P O R C

PROCESSOR ORIENTED REALITY CONTROL LANGUAGE

REFERENCE MANUAL

PREPARED FOR:

WILLARD OLSON & PAUL DESJARDINS
INFACT
3123 STEVELY AVENUE
LONG BEACH, CALIFORNIA 90808

COPY IDENTIFICATION:

0005 -- 17 JUN 1974

THE PORC PROCESSOR IS AVAILABLE FOR USE AS A CONTROL LANGUAGE WITHIN THE ´REALITY´ SYSTEM. IT IS AVAILABLE FROM AND SUPPORTED BY:

VIRTUAL-DATA-SERVICES
7204 CLAIREMONT MESA BOULEVARD
SAN DIEGO CA 92111
TELEPHONE: (714) 560-0790

POSSESSION OF THIS MANUAL DOES NOT IMPLY AUTHORIZATION OR LICENSE TO USE, DISTRIBUTE, OR IMPLEMENT THE CAPABILITIES DESCRIBED HEREIN.

VIRTUAL-DATA-SERVICES

SUBJECT: DOCUMENTATION FOR "PROCESSOR ORIENTED REALITY CONTROL" LANGUAGE

AN INTERPRETATIVE CONTROL LANGUAGE HAS BEEN IMPLEMENTED USING THE PROCEDURE CONCEPT AS IS DESCRIBED IN THE "PROC" SECTION OF REALITY SOFTWARE DOCUMENTATION. THIS NOTE DESCRIBES ENHANCEMENTS TO CURRENTLY EXISTING CAPABILITIES AND PROVIDES ADDITIONAL DOCUMENTATION FOR SOME PROCEDURE FUNCTIONS NOT CURRENTLY AVAILABLE.

THE "PROCESSOR ORIENTED REALITY CONTROL" (PORC) LANGUAGE COMBINES ADVANTAGES OF THE "PROC" LANGUAGE WITH A GENERALIZED CONTROL STRUCTURE, ABILITY TO RETRIEVE AND MODIFY DATA FROM FILES UTILIZING MULTI-VALUE AND MULTI-SUBVALUE STRUCTURES, AND TO ACCESS THE "ENGLISH" LANGUAGE FUNCTIONS WITHIN THE PORC PROCEDURES.

AS EXAMPLES; IT IS POSSIBLE TO TEST FOR EXISTENCE OF A FILE, ITEM WITHIN THE FILE, ATTRIBUTE WITHIN THE ITEM, VALUE WITHIN THE ATTRIBUTE, AND FINALLY SUB-VALUES WITHIN THE VALUE. PORC CONTROL PROCEDURES PERMIT PORCS TO BE ACCESSED FROM FILES OTHER THAN THE MASTER DICTIONARY THUS ALLOWING PORCS TO BE LOCATED IN ACCOUNTS OTHER THAN THE USERS.

PORC PROCEDURES UTILIZE A CONTROL STACK FOR SAVING RETURN ADDRESSES, STORING COUNTER VALUES, AND HOLDING STATUS FLAGS.

THE PORC LANGUAGE USES A SYNTAX FORM SIMILIAR TO THE PROC WITH THE INCLUSION OF THE ABILITY TO DEFINE PRIMITIVE INSTRUCTIONS WITHIN THE PORC PROCESSOR. USER EXITS MAY ALSO BE DEFINED IN A SYMBOLIC FORMAT RATHER THAN LIMITED TO THE MODAL ENTRY POINT.

THE FOLLOWING SECTIONS PROVIDE DETAILED DESCRIPTIONS OF EACH OF THE FUNCTIONAL EXTENSIONS TO THE PORC LANGUAGE.

SECTION 1

*** PORC LANGUAGE ELEMENTS ***

THE ELEMENTS WITHIN THE PORC LANGUAGE CONSIST OF PRIMITIVES WHICH ARE PROCESSED IN CONJUNCTION WITH MODIFIERS. THE BASIC LANGUAGE ELEMENTS ARE EXAMINED AND ANALYZED AT WHICH TIME CONTROL IS PASSED TO THE APPROPRIATE FUNCTION. EXAMPLES OF PRIMITIVES ARE: ´H´ - MOVE HOLLERITH LIKE STRING; ´0´ - OUTPUT DATA TO THE TERMINAL; AND ´IF´ - MAKE A LOGICAL DECISION BASED ON ARGUMENTS WHICH FOLLOW. A MODIFIER TO A LANGUAGE PRIMITIVE IS ILLUSTRATED BY THE ´IFN´ WHICH IS ´IF´ WITH A NEGATION MODIFIER. THE RESULT OF THE IF STATEMENT IS INVERTED.

TO PERMIT USER EXTENSIONS TO THE PORC PROCESSOR, THE PRIMITIVE FUNCTION ´U´ HAS BEEN EXPANDED TO INCLUDE SYMBOLIC USER EXITS. THE STANDARD FORM OF THE ´U´ IS ´UXXXX´ WHERE ´XXXX´ IS THE MODAL-ID ADDRESS IN HEXADECIMAL. THE SYMBOLIC FORM OF THE USER EXIT IS ´U NAME´ WHERE NAME IS ANY SYMBOLIC NAME WHICH IS DEFINED IN THE USER FUNCTION TABLE.

THE FOLLOWING LIST IS A SUMMARY OF THE PORC PRIMITIVES AND THEIR FUNCTIONS. IN SOME CASES, PRIMITIVES MAY APPEAR TO BE SIMILIAR, THE PARSING OF THE STATEMENTS IS DONE USING AN ORDERED TABLE TO MINIMIZE CONFLICTS OF NAMES.

PRIMITIVE NAME		FUNCTIONAL DESCRIPTION
A	-	MOVE ELEMENT FROM ACTIVE INPUT TO ACTIVE OUTPUT BUFFER
ADD	-	ADD VALUE TO A WORK COUNTER
B	-	MOVE INPUT BUFFER POINTER BACKWARDS
CALL	-	MARK PLACE AND TRANSFER CONTROL
C	-	COMMENT - PROCESS NO INFORMATION IN THE LINE
DI	-	DISPLAY DATA WITHIN THE INPUT BUFFER
EXIT	-	EXIT FROM SUBROUTINE
F	-	MOVE INPUT BUFFER POINTER FORWARD
F-	-	FILE ACCESS PRIMITIVE FUNCTIONS (SEE SECTION 6)
G	-	PROCESS GO TO FUNCTION
H	-	MOVE HOLLERITH STRING TO ACTIVE OUTPUT BUFFER
IF	-	PROCESS CONDITIONAL TEST
I	-	INPUT TO ONE OF THE INPUT BUFFERS
MARK	-	MARK PLACE IN THE CONTROL STACK AND CONTINUE
MDEL	-	DELETE LAST SAVED PLACE IN THE CONTROL STACK
MOV	-	MOVE DATA PRIMITIVES (SEE SECTION 4. A)
O	-	OUTPUT MESSAGE TO TERMINAL
P	-	PROCEED TO PROCESSOR
READ	-	READ AN ITEM FROM A FILE TO THE DESIGNATED WORK AREA
RETN	-	RETURN FROM SUBROUTINE
RI	-	RESET INPUT BUFFERS
RO	-	RESET OUTPUT BUFFERS
SET	-	PLACE A VALUE OF 1 OR 0 IN THE INDICATED WORK BIT
SKIP	-	MOVE FORWARD OR BACKWARD SPECIFIED NUMBER OF LINES
ST	-	TURN THE STACK ON OR OFF
SUB	-	DECREASE THE CONTENTS OF A WORK COUNTER
S	-	SET POINTER WITHIN THE INPUT BUFFERS
TRANS	-	TRANSFER CONTROL TO ANOTHER PORC PROCEDURE
T-ON	-	TURN ON THE DISPLAY TERMINAL
T-OFF	-	TURN OFF THE DISPLAY TERMINAL
U	-	PROCESS USER EXIT FUNCTION

WRAP	-	WRAP-UP CURRENT HISTORY STRING AND RETURN
WRI	-	WRITE AN ITEM FROM A WORK SPACE TO A FILE
X	-	EXIT FROM THE PORC TO TCL VIA WRAP-UP
*	-	COMMENT - TOTALLY IGNORE THE LINE DURING EXECUTION

CERTAIN PRIMITIVES HAVE SEVERAL FORMS AS PREVIOUSLY MENTIONED. THE SPECIFICS OF THESE FORMS ARE DESCRIBED AS FOLLOWS:

-- A -- THE A FUNCTION USED WITH THE (M,N) OPTION WILL OPERATE WITHIN THE ACTIVE BUFFER. THE ´SP´ OR ´SS´ DIRECTIVES WILL MOVE THE POINTER INTO THE PRIMARY OR SECONDARY BUFFER RESPECTIVELY.

-- ADD -- INCREMENTS THE CONTENTS OF A WORK COUNTER BY A LITERAL VALUE OR BY THE CONTENTS OF ANOTHER WORK COUNTER. STATEMENT SYNTAX IS: ADD ´TO´ ´FROM´. ´TO´ IS A WORK COUNTER DESIGNATOR AND ´FROM´ IS EITHER A STRING OR A WORK COUNTER DESIGNATOR.

-- B -- THE B FUNCTION WILL MOVE THE POINTER WITHIN THE INPUT BUFFER BACK ANY NUMBER OF ELEMENTS. THE VALUE IS SPECIFIED IN THE ELEMENT FOLLOWING THE B. A MOVE OF 0 ELEMENTS IS THE SAME AS A MOVE OF 1 ELEMENT AND IS IMPLIED WHEN NO MODIFIER IS PRESENT.

-- DI -- THE DI FUNCTION WILL DISPLAY ALL OR PART OF EITHER INPUT BUFFER. THE INPUT BUFFER POINTER IS NOT MOVED WHEN THE DI FUNCTION IS PROCESSED. VALID FORMS OF THE DI AND THEIR INTERPRETATION ARE-

DIP	DISPLAY FROM THE PRIMARY BUFFER
DIS	DISPLAY FROM THE SECONDARY BUFFER
DI	DISPLAY FROM THE ACTIVE INPUT BUFFER
DI1	DISPLAY THE FIRST ELEMENT IN THE ACTIVE INPUT BUFFER
DI0	DISPLAY ALL OF THE ACTIVE INPUT BUFFER
DIP2	DISPLAY THE SECOND ELEMENT IN THE PRIMARY INPUT BUFFER

IF THE ELEMENT SPECIFIED IS NOT PRESENT, ALL OF THE SPECIFIED INPUT BUFFER WILL BE DISPLAYED.
IF THE LAST CHARACTER OF THE COMMAND IS A ´<´ THE CURSOR WILL BE RETURNED TO THE BEGINNING OF THE LINE ON WHICH THE DISPLAY WAS GENERATED. IF THE LAST CHARACTER IS A ´+´ THE CURSOR WILL BE LEFT AT THE FINAL POSITION.

-- F -- THE F FUNCTION WILL MOVE THE POINTER WITHIN THE INPUT BUFFER FORWARD ANY NUMBER OF ELEMENTS. THE FORWARD POSITION WILL NEVER MOVE BEYOND THE END OF THE ACTIVE INPUT BUFFER.
THE NUMBER OF ELEMENTS MOVED WILL BE 1 IF EITHER A BLANK OR A 0 IS FOUND.

-- G -- THE GO TO PROCESSOR WILL OPERATE WITH EITHER A SYMBOLIC OR NUMERIC LABEL. IF A NUMERIC LABEL IS USED, IT MUST BE GREATER THAN 0 AS LEADING ZEROS ARE SUPPRESSED FROM NUMERIC LABELS. SYMBOLIC LABELS MAY CONSIST OF ANY CONTIGUOUS STRING OF CHARACTERS NOT INCLUDING BLANKS. THE DESIGNATED PORC WILL BE SCANNED FOR THE LITERAL STRING AND CONTROL WILL BE PASSED TO THE LINE CONTAINING THAT STRING. IN THE CASE OF LITERAL LABELS THE FIRST CHARACTER MUST BE AN ´L´ TO DESIGNATE A LABEL.

-- IF --
-- IFN -- THE LOGICAL IF FUNCTION MAY BE INVERTED OR NEGATED BY USING THE ´N´ MODIFIER. IF THE ´IFN´ FORM IS USED, THE IF WILL BE

PROCESSED IN THE NORMAL WAY; AT THE TIME THE DECISION IS MADE TO RETURN EITHER TO THE NEXT ELEMENT IN THE CURRENT LINE OR TO THE NEXT LINE, THE RETURN WILL BE TO THE OPPOSITE PLACE. DETAILED DESCRIPTION OF THE VARIANTS OF THE IF PRIMITIVE ARE CONTAINED IN SECTION 4. B.

-- I -- THE INPUT PRIMITIVE HAS THREE BASIC FORMS: IN, IP, AND IT. THE ´IN´ WILL ALWAYS RESET AND INPUT TO THE BEGINNING OF THE SECONDARY INPUT BUFFER. THE ´IP´ WILL INPUT AT THE CURRENT POSITION OF THE ACTIVE INPUT BUFFER AND WILL REPLACE THE ELEMENT WHICH IS POINTED TO BY THE STRING OF ELEMENTS INPUT. BY USING THE ´IP´ FORM, THE POSITION OF ELEMENTS FOLLOWING THE ELEMENT POINTED TO MAY CHANGE. IN THE CASE OF THE ´IT´ INPUT IS OBTAINED FROM THE MAGNETIC TAPE IF IT IS ATTACHED. INPUT IS ALWAYS TO THE SECONDARY INPUT BUFFER (NOTE-DIFFERENT FROM THE PROC IN WHICH ´IT´ IS ALWAYS TO PRIMARY BUFFER). IN ALL CASES, WBO IS SET IF DATA WAS ENTERED AND WCO CONTAINS THE NUMBER OF CHARACTERS WHICH WERE INPUT.

-- MOV - THE MOV PRIMITIVE IS DESCRIBED IN SECTION 4. A. IT MAY BE USED TO MOVE VARIOUS DATA ELEMENTS TO OTHER DATA STORAGE AREAS.

-- P -- THE PROCEED FUNCTION HAS SEVERAL VARIANTS. THE BASIC FORM IS THE ´P´ WHICH TRANSFERS CONTROL FROM THE PORC TO ANOTHER PROCESSOR. IN THIS FORM, THE DATA STORED IN THE WORKSTRINGS AND THE PRIMARY OUTPUT BUFFER IS DESTROYED. THIS FUNCTION IS A NORMAL PART OF THE TRANSFER OF CONTROL AND IS RELATED TO THE LOGICAL EXCHANGE OF THE ´IS´ AND ´OS´ WORKSPACE REGIONS. THE ´PP´ IS SIMILIAR IN FUNCTION TO THE ´P´ EXCEPT THE PRIMARY AND SECONDARY OUTPUT BUFFERS ARE DISPLAYED AT THE TERMINAL AND THE PORC PAUSES BEFORE TRANSFERING CONTROL TO THE PROCESSOR. IF THE CHARACTER ´N´ IS TYPED FOLLOWED BY A CARRIAGE RETURN, THE PORC IS ABORTED AND CONTROL IS RETURNED TO TCL WITHOUT PROCESSING ANY OF THE FUNCTIONS WHICH MAY HAVE BEEN REQUESTED. AN OPTION EXISTS FOR THE ´P´ AND ´PP´ TO PASS CONTROL TO A USER EXIT OR MODAL ENTRY POINT BY USING THE FORM ´PU´ OR ´PPU´. IN THIS CASE, THE PROCESSING IS NOT PASSED TO TCL BUT TO THE USER EXIT DEFINED EITHER BY SYMBOLIC NAME OR MODAL ENTRY POINT.

-- PA -- THE ´PA´ FUNCTION IS A PAUSE WITHOUT MODIFICATION OF THE USERS DATA STORAGE ELEMENTS. IT IS OF USE IN WAITING FOR RESPONSE TO MOUNT FORMS ON A TERMINAL OR DURING CHECKOUT. IT WILL DISPLAY THE REMAINDER OF THE COMMAND LINE AT THE TERMINAL. A CARRIAGE RETURN WILL CONTINUE EXECUTION OF THE PORC; A ´N´ WILL ABORT AS IN THE CASE OF THE ´PP´ FUNCTION.

-- PE -- THE ´PE´ FUNCTION EXISTS TO PEEK AT ANY DATA STORAGE ELEMENTS AT THE TERMINAL. IT IS SIMILIAR IN FORM TO THE ´O´ FUNCTION EXCEPT THAT INDIRECT ADDRESS POINTERS MAY BE USED.

-- READ - READ THE DESIGNATED ITEM FROM THE DESIGNATED FILE TO THE DESIGNATED WORK SPACE. STATEMENT SYNTAX IS:

```
   READ ´WORK FILE´ ´TO´  ´ITEM´
```

´WORK FILE´ IS A FILE POINTER DESIGNATOR OF A FILE WHICH HAS BEEN OPENED BY A PREVIOUS STATEMENT. ´TO´ IS A STRING

STORAGE DESIGNATOR. FOR THE READ PRIMITIVE ONLY AN OUTPUT BUFFER OR A (LONG) WORK STRING ARE LEGAL STRING DESIGNATORS. ´ITEM´ IS EITHER AN EXPLICIT STATEMENT (LITERAL) OF THE ITEM NAME OR A POINTER TO THE ITEM NAME.
IF THE READ REQUEST COULD NOT BE CONSUMATED, WORK BIT ZERO IS SET TO ZERO. FAILURE TO READ COULD BE DUE TO INCORRECT SPECIFICATION OF PARAMETERS OR THEIR VALUE OR ABSENCE OF ITEM NAME FROM THE FILE.

-- RI -- THIS FUNCTION RESETS THE INPUT BUFFERS TO AN EMPTY STATE. OR DELETES A PORTION OF THE BUFFER.
SPECIFIC FORMS ARE:

RI	-	RESET BOTH INPUT BUFFERS
RIP	-	RESET THE PRIMARY INPUT BUFFER
RIS	-	RESET THE SECONDARY INPUT BUFFER
RIXN	-	X = P OR S. RESET THE INPUT BUFFER FROM THE SPECIFIED ELEMENT NUMBER (N) ONWARD. I. E. IF THE PRIMARY INPUT BUFFER CONTAINS: A B C RIP3 WILL SET TO THE BEGINNING OF ELEMENT ´C´ AND MARK THAT AS THE END OF THE BUFFER (LEAVING ONLY ELEMENTS ´A´ AND ´B´).
RIX(N)	-	X = P OR S. RESET TO COLUMN ´N´ IN THE INDICATED BUFFER. COLUMNS LESS THAN ´N´ WILL BE RETAINED.

-- RO -- THIS FUNCTION RESETS THE SELECTED OUTPUT BUFFER. VALID FORMS ARE- ´ROP´ AND ´ROS´ TO RESET THE PRIMARY AND SECONDARY OUTPUT BUFFERS. THE STACK ON/OFF STATUS IS NOT AFFECTED.

-- SET -- PLACE A ONE OR A ZERO IN THE SPECIFIED WORK BIT (INDICATOR).
STATEMENT SYNTAX IS: SET ´TO´ VALUE
´TO´ IS A WORK BIT DESIGNATOR. IF ´VALUE´ IS OMITTED, A ONE IS IMPLIED.

-- SKIP - THE SKIP FUNCTION IS USED TO SKIP FORWARD OR BACKWARD A SPECIFIED NUMBER OF LINES RELATIVE TO THE CURRENT LINE. A ´SKIP 1´ IS THE SAME AS A COMMENT. A ´SKIP 0´ WILL HANG UP THE TERMINAL SINCE IT IS A SKIP TO ITSELF.

-- SUB -- DECREMENT THE DESIGNATED WORK COUNTER BY THE DESIGNATED VALUE. STATEMENT SYNTAX IS: SUB ´TO´ ´BY´
´TO´ IS A WORK COUNTER DESIGNATOR AND ´BY´ IS EITHER A WORK COUNTER DESIGNATOR OR A LITERAL INTEGER VALUE.

-- S -- THIS FUNCTION WILL SET THE POINTER TO EITHER THE PRIMARY OR SECONDARY INPUT BUFFER AT ANY SPECIFIC COLUMN OR ELEMENT NUMBER. VALID FORMS ARE:

S	-	SET TO BEGINNING OF ACTIVE BUFFER
SP	-	SET TO BEGINNING OF PRIMARY BUFFER
SS	-	SET TO BEGINNING OF SECONDARY BUFFER
SP(3)		SET TO COLUMN THREE OF THE PRIMARY BUFFER
SP3		SET TO ELEMENT THREE OF THE PRIMARY BUFFER
SS0		SET TO BEGINNING OF THE SECONDARY BUFFER

-- T-ON TURN ON THE TERMINAL SO ALL MESSAGES ARE DISPLAYED. THIS FUNCTION WILL NOT TOGGLE BUT WILL FORCE ON THE DISPLAY.

-- T-OFF TURN OFF THE TERMINAL SO ALL MESSAGES WILL NOT BE DISPLAYED. THIS FUNCTION WILL NOT TOGGLE BUT WILL TURN OFF THE DISPLAY.

-- U -- THE USER EXIT MAY BE EITHER A MODAL-ID WHICH IS SPECIFIED AS FOUR HEX DIGITS IMMEDIATELY FOLLOWING THE 'U' OR ELSE A SYMBOLIC EXIT SPECIFIED BY A BLANK IMMEDIATELY FOLLOWING THE 'U'. IN THIS CASE, THE NEXT ELEMENT WILL BE EXAMINED TO DETERMINE THE NAME OF THE USER FUNCTION AND OPTIONALLY INFORMATION SPECIFYING THE NUMBER OF ARGUMENT LINES AND RETURN POINTS. THE FOLLOWING FORMS ARE VALID:

```
            U FUNCTION          -   FUNCTION ONLY
            U FUNCTION,ARGS     -   FUNCTION AND ARGUMENTS
            U FUNCTION,ARGS,RETS    FUNCTION, ARGUMENTS AND
                                    RETURN POINTS
            U FUNCTION,,RETS    -   FUNCTION AND RETURN POINTS
```

THE ABOVE COMBINATIONS ARE ALSO VALID FOR THE SYMBOLIC 'P' AND 'PP' ENTRIES DESCRIBED IN THE 'P' SECTION.

-- WRITE- UPDATE THE ITEM IN THE DESIGNATED WORK SPACE TO THE DESIGNATED FILE. STATEMENT SYNTAX IS:

```
   WRITE  'WORK FILE'  'FROM'
```

'WORK FILE' IS A VALID FILE POINTER DESIGNATOR AS FOR THE READ PRIMITIVE. 'FROM' IS ANY STRING STORAGE DESIGNATOR. THE CONTENTS OF THE FIRST ATTRIBUTE (ATTRIBUTE ZERO) IN ITEM WILL BE USED AS THE ITEM NAME. THE WRITE WILL ABORT THE PORC PROCESS IF THE WORK FILE SPECIFICATION IS FOR LEVEL ZERO (DICTIONARY LEVEL) OF THE FILE.

SECTION 2

*** ELEMENTS FOR TRANSFER OF CONTROL ***

SIX PRIMITIVE FUNCTIONS ARE AVAILABLE FOR TRANSFER OF CONTROL WITHIN AND BETWEEN PORC PROCESSES. THE PORC LANGUAGE IS INTERPRETATIVE AND EACH PORC IS CONTAINED WITHIN AN ITEM. THE ITEMS MAY EXIST WITHIN THE USERS MASTER DICTIONARY, A FILE DICTIONARY, A DATA SECTION OF A FILE, OR WITHIN ANOTHER USERS DATA STRUCTURES.

TRANSFERS OF CONTROL CONSIST OF TWO BASIC TYPES, TRANSFERS TO NEW ITEMS OR POSITIONS WITHIN ITEMS, AND RETURNS TO KNOWN POSITIONS. A CONTROL STACK IS USED TO SAVE KNOWN POSITIONS AND HAS THE CAPABILITY TO SAVE AT LEAST 11 POSITIONS BEFORE THE STACK OVERFLOWS.

RETURNS TO KNOWN POSITIONS CONSIST OF RESTORING THE INFORMATION SAVED IN THE CONTROL STACK. A STACK STRUCTURE IS USED TO ALLOW RECURSIVE CALLS TO A ROUTINE AS WELL AS TO PERMIT MANIPULATION OF THE STACK WHEN PROCESSING ERROR RETURNS. RETURNS MAY CONTAIN A MODIFIER WHICH WILL CAUSE CONTROL TO BE RETURNED TO THE SAVED POSITION MODIFIED BY A VALUE REPRESENTING THE NUMBER OF LINES RELATIVE TO THE LINE AT WHICH THE POSITION WAS SAVED. THIS ALSO FACILITATES THE PROCESSING OF ERROR RETURNS AND THE CONDITIONAL BRANCHING WHICH MAY BE REQUIRED.

TRANSFERS OF CONTROL CONSIST OF SPECIFYING THE NAME OF THE ITEM TO WHICH CONTROL IS TO BE PASSED AND AN OPTIONAL ENTRY POINT WITHIN THE ITEM WHICH IS ANALOGUS TO THE 'GO' STATEMENT. THE NAME OF THE ITEM MUST EXIST IN THE USERS MASTER DICTIONARY. THE CONTENTS OF THE ITEM WITHIN THE MASTER DICTIONARY MAY BE A PORC PROCEDURE OR A LINKAGE TO A PORC PROCEDURE. A SPECIAL CASE EXISTS WHEN THE ITEM NAME IS "*" WHICH IMPLIES THAT THE CURRENTLY ACTIVE PORC IS THE DESTINATION PORC. IN CONJUNCTION WITH THE 'TRANS' FUNCTION, AN EQUIVALENT FORM OF THE 'GO' FUNCTION IS THUS CREATED.

TRANSFER OF CONTROL PRIMITIVES ARE:

```
   --  MARK             : MARK CURRENT PLACE AND CONTINUE WITH THE NEXT
                        : LINE OF THE PORC

   --  TRANS ITEM LABEL  : TRANSFER TO ANOTHER ITEM WITH THE NAME
                        : SPECIFIED IN ELEMENT CALLED 'ITEM'  GO TO THE
                        : LABEL SPECIFIED IN THE ELEMENT CALLED 'LABEL' IF
                        : IT EXISTS.  'ITEM' IS OBTAINED FROM THE USERS
                        : M/DICT.  THE ITEM MAY BE ANY OF THE FOLLOWING
                        : TYPES; D, Q, PQC, PQCL, FQC, FQCL.  IF THE
                        : 'ITEM' IS AN *, THEN THE TRANSFER REMAINS IN THE
                        : CURRENT ITEM.  IF THE 'ITEM' IS ENCLOSED IN
                        : PARENTHESIS, THE FOLLOWING FORMS ARE POSSIBLE;
                        : (FILE ITEM) OR (DICT FILE ITEM).  IT IS ALSO
                        : POSSIBLE FOR 'ITEM' TO BE AN INDIRECT POINTER
                        : TO ONE OF THE DEFINED WORKSPACES.

   --  CALL ITEM LABEL  : MARK CURRENT PLACE AND TRANSFER CONTROL AS
                        : IN THE 'TRANS' FUNCTION.

   --  RETN VALUE       : RETURN TO POSITION SAVED IN THE CONTROL STACK
                        : OPTIONALLY MODIFIED BY THE NUMBER OF LINES SPECIFED
```

: BY VALUE. IF VALUE IS NEGATIVE, MOVE BACKWARD.
: IF THE CONTROL STACK CONTAINS NO VALUES, AN ERROR
: CONDITION EXISTS AND THE PORC IS ABORTED.

-- EXIT VALUE : RETURN TO POSITION SAVED IN THE CONTROL STACK AS
: DESCRIBED FOR RETN. IF THE CONTROL STACK IS
: EMPTY, EXIT FROM THE PORC IN NORMAL WAY TO WRAP-UP
: AND RETURN CONTROL TO THE TERMINAL.

-- MDEL : THIS PRIMITIVE IS USED TO DELETE A MARKED PLACE
: FROM THE STACK. THE LAST SAVED VALUE IS DELETED.

WITHIN THE TRANSFER OF CONTROL TO ANOTHER PORC, THE FOLLOWING FORMS OF AN ITEM MAY BE ENCOUNTERED AND PROCESSED. THE DESCRIPTION WITH EACH ITEM SHOULD PROVIDE SOME INSIGHT TO ITS PURPOSE.

001 PQC - DESIGNATOR FOR PORC PROCEDURES AS OPPOSED TO PROC PROCEDURES. THIS FORM IMPLIES THE PROCEDURE IS CONTAINED BEGINNING AT THE SECOND ATTRIBUTE.
IF THE FIRST CHARACTER OF THE SECOND ATTRIBUTE IS AN * THE PORC PROCESSOR IGNORES THE LINE. ANY NUMBER OF * LINES MAY BE INCLUDED AT THE BEGINNING OF A PORC TO PROVIDE A SECTION FOR COMMENTS. THIS AREA IS NOT CONSIDERED A SECTION OF THE PORC AND DOES NOT CONTRIBUTE TO THE OVERHEAD WHEN SCANNING THE PORC FOR 'GO TO' TYPE STATEMENTS.

001 PQCL - DESIGNATES THE PORC PROCEDURE IS NOT CONTAINED WITHIN
002 FILE THIS ITEM BUT TO LOOK WITHIN THE FILE SPECIFIED IN
003 ITEM ATTRIBUTE 002 AND THE ITEM SPECIFIED BY ATTRIBUTE 003.
IF ATTRIBUTE 003 CONTAINS AN '*' AS THE FIRST CHARACTER, THE ITEM WILL BE FOUND IN THE DATA SECTION OF THE FILE, OTHERWISE THE TRANSFER OF CONTROL PROCESSOR WILL LOOK IN THE DICTIONARY SECTION OF THE FILE. IF THE FIRST CHARCTER OF ATTRIBUTE 002 IS AN *, THE FILE NAME WILL BE LOOKED FOR IN THE DATA SECTION OF THE CURRENTLY ACTIVE FILE. IF EITHER ATTRIBUTE 002 OR 003 IS OF THE FORM (FILE ITEM) OR (DICT FILE ITEM) THE APPROPRIATE TRANSFER TO THE FILE AND ITEM WILL OCCUR. THUS, IF ATTRIBUTE 002 HAS THE FORM (X Y), ATTRIBUTE 003 IS IGNORED.

001 FQC - DESIGNATOR FOR PORC PROCEDURES WHICH MAY BE CONTAINED WITHIN THE USERS MASTER DICTIONARY BUT ARE NOT TO BE CONSIDERED AS VALID VERBS BY THE TERMINAL CONTROL LANGUAGE INTERPRETER. THIS FORM IS COMMONLY USED FOR SUBROUTINE ITEMS WHICH ARE NOT DIRECTLY EXECUTED.
THIS FORM IS PROCESSED THE SAME AS PQC.

001 FQCL - THIS FORM IS A LINKAGE SIMILIAR TO THE PQCL WITH THE ABILITY TO NOT BE CONSIDERED AS A VERB BY TCL.

001 D - THIS IS THE STANDARD DEFINITION ITEM FOR A FILE
002 BASE INCLUDING THE BASE, MODULE, AND SEPARATION AS WELL AS
003 MODULO THE FILE LOCKS WHICH ARE CHECKED IN THE STANDARD WAY.
004 SEPARATION

```
    001  Q      -  THIS IS THE STANDARD EQUIVALENCE ITEM TO ANOTHER FILE
    002  USER      WITHIN ANOTHER USERS ACCOUNT.  IF THIS FORM IS USED,
    003  FILE      THE ITEM NAME IS IMPLIED AND INTERPRETED AS THE ITEM
                   NAME SPECIFIED IN THE ORIGINAL TRANSFER OR ELSE WITHIN
                   ATTRIBUTE 003 OF AN 'PQCL' ITEM OR A 'FQCL' ITEM.
```

THE VARIOUS FORMS OF THE TRANSFER OF CONTROL PROVIDE TECHNIQUES WHICH ALLOW VERY COMPLEX LINKAGES. FOR MOST APPLICATIONS, THE BASIC FORMS WILL BE SUFFICIENT.

SECTION 3

*** PORC STORAGE ELEMENTS ***

VARIOUS STORAGE REGIONS AND DATA ELEMENTS ARE DIRECTLY ACCESSABLE BY EACH PORC PROCESS. FOUR CLASSES OF STORAGE ARE AVAILABLE:

DATA STRING TYPE STORAGE
ADDRESSABLE WORK COUNTERS
ADDRESSABLE INDICATORS (WORK BITS)
FILE ACCESS POINTERS

STRING STORAGE:

THERE ARE FOUR CLASSES OF STRING STORAGE AVAILABLE

1. PRIMARY AND SECONDARY INPUT BUFFERS
2. PRIMARY AND SECONDARY OUTPUT BUFFERS
3. VARIABLE WORK STRINGS
4. SHORT WORK STRINGS

WORK COUNTERS:

TEN (16 BIT) WORK COUNTERS MAY BE MANIPULATED. WORK COUNTER ZERO IS USED TO COMMUNICATE THE RESULTS OF THE OPERATION OF SOME PORC PRIMITIVES.

INDICATORS (WORK BITS):

ONE HUNDRED ADDRESSABLE INDICATORS ARE AVAILABLE. WORK BIT ZERO IS USED TO COMMUNICATE THE RESULTS OF THE OPERATION OF SOME PORC PRIMITIVE FUNCTIONS.

FILE ACCESS POINTERS:

FOUR FILES MAY BE MANIPULATED USING THE STANDARD FILE LINKAGES. EACH OF THE FILES IS HANDLED AS A MULTI-LEVEL FILE WITH LEVEL 0 BEING THE DICTIONARY AND LEVEL 1 BEING THE DATA SECTION. IF THE FILE IS OPENED TO THE 'DICT' SECTION, LEVEL 0 WILL POINT TO THE FILE SECTION WHICH WAS USED TO FIND THE 'DICT' DEFINITION ITEM.

*** ADDRESSING CONVENTIONS ***

ADDRESSES ARE OF THE GENERAL FORM:

```
@ABC,X,Y,Z
```

```
WHERE A = PRINCIPAL STORAGE ADDRESS IDENTIFIER
      B = SECONDARY STORAGE ADDRESS IDENTIFIER
      C = TERTIARY STORAGE ADDRESS IDENTIFIER
      X = FIRST INDEX
      Y = SECOND INDEX
      Z = THIRD INDEX
```

THE '@' SYMBOL IS THE UNIVERSAL INDIRECT ADDRESS SENTINEL. ADDRESS SPECFICATION MAY BE A MULTI-LEVEL INDIRECT CHAIN.

SPECIFIC ADDRESS FORMS ARE:

```
I.   FOR INPUT BUFFERS
           @IP'N',X - SELECT PRIMARY INPUT BUFFER AT ELEMENT N+X
           @IS'N',X - SELECT SECONDARY INPUT BUFFER AT ELEMENT N+X
           @IA'N',X - SELECT CURRENTLY ACTIVE INPUT BUFFER AT
                      ELEMENT N+X

II.  FOR OUTPUT BUFFERS
           @OP'N',X - SELECT PRIMARY OUTPUT BUFFER AT ELEMENT N+X
           @OS'N',X - SELECT SECONDARY OUTPUT BUFFER AT ELEMENT N+X
           @OA'N',X - SELECT CURRENTLY ACTIVE OUTPUT BUFFER AT
                      ELEMENT N+X
```

IN THESE BUFFERS, ELEMENTS ARE SEPARATED BY BLANKS. IF 'N' IS OMITTED, THE CURRENT END OF THE BUFFER IS IMPLIED AND ANY INDEX FIELD WILL BE IGNORED. FOR PURPOSES OF ELEMENT POSITIONING, VALUES OF N=0 AND N=1 ARE EQUIVALENT. N=0 CARRIES THE ADDED SIGNIFICANCE OF TERMINATING AN INDIRECT ADDRESS CHAIN. FOR NONZERO VALUES OF N, INDIRECT ADDRESS SPECIFICATION MAY CONTINUE FOR ANY NUMBER OF LEVELS. FOR MULTI-LEVEL INDIRECT SPECIFICATION, THE VALUE OF THE INDEX WILL BE SAVED AND USED ONLY AFTER THE LAST INDIRECT LEVEL. ADDITIONALLY, THE INDEX SPECIFICATION MAY OCCUR AT ANY LEVEL OF THE INDIRECT CHAIN. FOR MULTI-LEVEL INDIRECT SPECIFICATION, ONLY THE FIRST OCCURRENCE OF AN INDEX FIELD WILL BE RETAINED.

EXAMPLES OF INPUT AND OUTPUT BUFFER ADDRESS SPECIFICATION:

SUPPOSE THE PRIMARY INPUT BUFFER CONTAINS:

```
TEST-FILE-NAME  TEST-ITM-NAME
```

THE PORC FUNCTION

```
F-CITEM @WF1,1 @IP2
```

WOULD USE THE PARAMETER 'TEST-ITM-NAME' AS ITS ARGUMENT. EQUIVALENT ADDRESS SPECIFICATIONS WOULD BE

```
F-CITEM @WF1,1 @IP0,1  AND
F-CITEM @WF1,1 @IP1,1
```

NOTE THAT INDEX FIELD VALUES ALWAYS DECLARE THE NUMBER OF DATA ELEMENTS TO DISPLACE TO FROM THE BASE ADDRESS SPECIFICATION. AN INDEX VALUE OF ZERO IMPLIES USING THE ELEMENT AT THE BASE ADDRESS. AN INDEX OF ONE IMPLIES A SKIP FORWARD OF ONE ELEMENT FROM THE BASE ADDRESS ELEMENT.

TO ILLUSTRATE MULTI-LEVEL INDIRECTION AND INDEXING, CONSIDER THE FOLLOWING EXERCISE:

```
PRIMARY INPUT BUFFER: 8  10.05  3.14159
SECONDARY INPUT BUFFER: @IS3  @IP1,2  @OP1,1  @IP3
PRIMARY OUTPUT BUFFER:  @IS2
```

THE PORC USER FUNCTION SEQUENCE

```
U CALC
(2.1*@IS1 +(85/@OP1 +2))*@IP0 ?
```

WOULD EVALUATE THE EXPRESSION

```
(2.1*10.05+(85/3.14159 +2))*8  , GIVING A VALUE OF 401.29090536
```

THE DETERMINATION OF THE LOCATIONS OF THE VARIABLES IS AS FOLLOWS:

```
FIRST VARIABLE: @IS1 : @IS3 : @OP1,1 (SAVE INDEX OF 1) : @IS2 :
                @IP1,2 (IGNORE INDEX) : 8 : 10.05 (SINCE INDEX = 1)
SECOND VARIABLE: @OP1 : @IS2 : @IP1,2 (SAVE INDEX OF 2) : 8
                 3.14159 (SINCE INDEX = 2)
THIRD VARIABLE: @IP0 : 8
```

III. FOR OUTPUT TO TERMINAL OR PRINT SPOOLER:

```
@T
@P
```

IV. FOR WORK ACCUMMULATOR (SHORT STRING), WORK BITS AND WORK COUNTERS:

```
@WA´N´   WHERE  N = 0-9
@WB´N´   WHERE  N = 0-99
@WC´N´   WHERE  N = 0-9
```

WORK ACCUMMULATORS ARE LIMITED TO A MAXIMUM LENGTH OF 50 CHARACTERS.

V. FOR WORK STRINGS:

```
@WS´N´,X,Y,Z  WHERE N = 1-4
```

ONLY SINGLE LEVEL INDIRECT ADDRESSING IS USED FOR THE WORK STRINGS, ELEMENT DETERMINATION WITHIN THE WORK STRINGS IS DIFFERENT THAN FOR THE INPUT AND OUTPUT BUFFERS IN THAT, THE DATA ELEMENT HIERARCHY OF THE REALITY FILE SYSTEM IS SUPPORTED IN THE WORK STRINGS. HERE, ´N´ IDENTIFIES WHICH OF THE FOUR WORK STRINGS IS TO BE SELECTED. DEFAULT WORK STRING SIZES ARE 750 CHARACTERS EACH. USERS WITH LARGER LOGON WORK SPACES MAY HAVE LARGER WORK STRINGS.

OMISSION OF THE INDICES X,Y AND Z PERMITS ACCESSING THE CURRENT END POSITON OF THE SELECTED WORK STRING. THE INDICES X,Y AND Z HAVE THE FOLLOWING SIGNIFICANCE FOR EFFECTIVE ADDRESS DETERMINATION:

X = ATTRIBUTE NUMBER WITHIN THE STRING
Y = VALUE NUMBER WITHIN THE ATTRIBUTE
Z = SUBVALUE NUMBER WITHIN THE VALUE

IN ALL CASES, INDEX VALUES ARE INTERPRETED FROM A BASE OF ZERO. THE ADDRESS SPECIFICATION @WS2,0 IS EQUIVALENT TO @WS2,0,0,0 (BUT NOT TO @WS2) AND WOULD PERMIT ACCESSING THE BEGINNING OF WORK STRING TWO.

TO ILLUSTRATE WORK STRING ADDRESSING, CONSIDER THE FOLLOWING EXAMPLE:

SUPPOSE WORK STRING TWO CONTAINED THE FOLLOWING DATA

```
ABC(VM)DEF(SVM)GHI(AM)JKL(AM)MNO(VM)PQR(VM)STU(VM)VWX(AM)YZ(AM)(SM)
```

THIS DATA HAS THE LOGICAL STRUCTURE:

```
. . . . .  A0 . . . . . . . . . . *.  A1 . *. . . . . . . . . . . . . .  A2 . . . . . . . . . . *.  A3 . **
    OR EQUIVALENTLY:

. V0. *. . . . . . V1. . . . . . *. . V0. . *. . V0. . . *. . V1. . *. . V2. . *. . V3. . *. . V0. . **
    OR EQUIVALENTLY:

. SV0*. . SV0. . *. . SV1. *. . SV0. *. . SV0. . *. . SV0. *. . SV0. *. . SV0. *. . SV0. **
```

IN THE ABOVE EXPLOSION, A0 = ´ATTRIBUTE ZERO´, V0 = ´VALUE ZERO´, SV0 = ´SUBVALUE ZERO´, ETC. ´*´ INDICATES A STRING ELEMENT DELIMITER.

THAT IS, THE PRIMITIVE FORM OF AN ATTRIBUTE MAY BE DESCRIBED AS CONSISTING OF SINGLE SUBVALUE OR EQUIVALENTLY, A SINGLE VALUE. SIMILARLY, THE PRIMITIVE FORM OF A VALUE CONSISTS OF A SINGLE SUBVALUE. FOR THE PRIMITIVE FORMS, THE VALUE AND SUBVALUE SUB-STRUCTURE NOTIONS ARE NOT NORMALLY BROUGHT INTO PLAY.

EXAMPLE:

ADDRESS SPECIFICATION	CHARACTER STRING ADDRESSED
@WS2,0,1	DEF
@WS2,0,1,0	DEF
@WS2,0,1,1	GHI
@WS2,1	JKL
@WS2,2,2	STU

INDEX VALUES

INDEX VALUES MAY BE EITHER A LITERAL SPECIFICAION (AS IN THE EXAMPLES ABOVE) OR MAY SPECIFY THE CONTENTS OF ANY OF THE TEN WORK COUNTERS. TO ILLUSTRATE, IF WORK COUNTER FIVE CONTAINS THE VALUE 2,

```
    @WS2,@WC5,@WC5   IS EQUIVALENT TO   @WS2,2,2
```

SIMILARLY,

```
    @IP1,@WC5    IS EQUIVALENT TO    @IP1,2
```

NEGATIVE INDEX VALUES ARE TREATED AS ZERO.

VI.   FOR ACCESS TO FILES:

```
    @WF'N',X      WHERE N = 0-4 AND X = 0-1
```

FOR EACH OF THE FILES, THE BASE, MODULO, AND SEPARATION IS SAVED FOR THE DICTIONARY DEFINING THE FILE AND THE DATA SECTION OF THE FILE. THE INDEX VALUE OF 0 WILL REFERENCE THE DICTIONARY SECTION OF THE FILE AND THE INDEX VALUE OF 1 WILL REFERENCE THE DATA SECTION OF THE FILE. WORK FILE ZERO IS USED TO SAVE THE CURRENTLY ACTIVE FILE POINTERS WHEN EXECUTING THE 'WRAP' PRIMITIVE OR EXITING TO OTHER REALITY SYSTEM PROCESSORS VIA THE P OR PP PRIMITIVES. IT MAY OTHERWISE BE USED AS A FIFTH OPEN FILE DESIGNATOR.

SECTION 4

4. A MOV

THE PORC "MOV" PRIMITIVE PERMITS MOVEMENT OF DATA FROM REALITY DATA FILES TO PORC WORKSPACES AND BETWEEN WORK SPACE ELEMENTS. SEVERAL VARIANTS OF THE MOV PRIMITIVE ARE AVAILABLE IN ORDER TO FACILITATE DATA MANIPULATION WHICH WILL MAINTAIN THE INTEGRITY OF THE DATA ELEMENT HIERARCHY SUPPORTED BY REALITY.

MOVA TO FROM COUNT

MOVE 'ATTRIBUTES' BETWEEN ELEMENTS SPECIFIED IN THE 'TO' AND 'FROM' FIELDS. THESE MUST BE POINTERS TO PORC STORAGE ELEMENTS. THE 'COUNT' FIELD IS OPTIONAL AND MAY CONTAIN A LITERAL NUMERIC VALUE OR IT MAY BE A POINTER TO A PORC WORK COUNTER ELEMENT FROM WHICH THE COUNT VALUE IS TO BE RETRIEVED. THIS PRIMITIVE WILL MOVE THE CONTENTS OF ALL ATTRIBUTES BEGINNING AT THE SUBVALUE ADDRESSED IN THE 'FROM' FIELD FOR A MAXIMUM OF 'COUNT' ATTRIBUTES. IF 'COUNT' IS OMITTED, ALL DATA TO THE FIRST X'FF' (SEGMENT MARK) WILL BE MOVED. NORMALLY THIS WOULD BE ALL THE DATA REMAINING IN THE ITEM.

MOVV TO FROM COUNT

MOVE 'VALUES' BETWEEN STORAGE ELEMENTS SPECIFIED. THE DATA MOVE WILL COMMENCE WITH THE SUBVALUE ADDRESSED BY THE 'FROM' FIELD AS FOR THE MOVA PRIMITIVE. 'COUNT' VALUES WILL BE MOVED.
IF COUNT FIELD IS OMITTED, THE MOVE WILL TERMINATE AT THE NEXT HIGHER LEVEL IN THE DATA HIERARCHY, I. E. AT THE END OF THE CURRENTLY ADDRESSED ATTRIBUTE BOUNDARY.

MOVS TO FROM COUNT

MOVE 'SUBVALUES' BETWEEN STORAGE ELEMENTS; 'COUNT' SUBVALUES WILL BE MOVED IF POSSIBLE. OMMISSION OF THE COUNT FIELD WILL CAUSE MOVEMENT OF ALL SUBVALUES TO THE END OF THE CURRENTLY ADDRESSED VALUE.

MOVE TO FROM COUNT

MOVE ELEMENTS USING BLANKS AS THE ELEMENT DELIMITER. COUNT USED AS IN ALL THE VARIATIONS ABOVE. MOVEMENT WILL STOP ON EITHER A COUNT RUN OUT OR ENCOUNTERING ANY OF THE HIGHER DELIMITERS.

MOVC TO CHARACTER

MOVE ONE CHARACTER TO THE 'TO' LOCATION. THE CHARACTER CANNOT BE A BLANK.

MOVD TO DELIMITER

REPLACE THE CHARACTER POSITIONED TO WITH THE DELIMITER SPECIFIED. VALID DELIMITERS ARE: SVM SUBVALUE MARK, VM VALUE MARK,

AM ATTRIBUTE MARK, SM SEGMENT MARK,
SB SUBSTITUE BLANK, BL BLANK.

MOVX TO HEX-CHARACTER

MOVE ONE HEXADECIMAL CHARACTER TO THE 'TO' LOCATION. THE HEX CHARACTER MAY BE ANY CODE.

EXCEPTION CIRCUMSTANCES FOR THE 'MOV' PRIMITIVE:

```
1 THE FORM: MOVA 'TO' LITERAL
  EITHER WITH OR WITHOUT 'COUNT' IS IGNORED
  WITH THE EXCEPTION OF SETTING WORK BIT  ZERO TO
  ZERO. HOWEVER, THE FORMS
        MOVV 'TO' LITERAL
  AND   MOVS 'TO' LITERAL
  ARE EXECUTABLE.
```

2 FOR FORMS WHICH PERMIT A COUNT FIELD, THE ACTUAL NUMBER OF ELEMENTS MOVED WILL BE PLACED IN WORK COUNTER ZERO UPON COMPLETION OF THE MOVE.

3 IF THE ELEMENT IDENTIFIED IN THE 'FROM' SPECIFICATION DOES NOT EXIST (E.G. ATTEMPTING TO ADDRESS THE SECOND VALUE IN AN ATTRIBUTE HAVING ONLY ONE VALUE), NO DATA MOVEMENT WILL OCCUR AND WORK BIT ZERO WILL BE SET TO ZERO AS WILL WORK COUNTER ZERO.

4.B IF

THE 'IF' PRIMITIVE PERMITS TESTING OF THE CONTENTS OF ADDRESSABLE PORC STORAGE ELEMENTS. THE SYNTAX OF THE TEST STATEMENT IS:

```
IF 'SOURCE' 'CONDITIONAL' 'TEST' 'FUNCTION'
OR
IFN 'SOURCE' 'CONDITIONAL' 'TEST' 'FUNCTION'
```

'FUNCTION' IS ANY EXECUTABLE PORC STATEMENT.
SPECIFIC LEGAL FORMS ARE (IN ALL CASES, IF COULD BE REPLACED BY IFN):

1. STRING VARIABLE TESTS

```
IF 'STRINGID1'                    FUNCTION (PRESENCE TEST)
IF 'STRINGID1' = 'STRINGID2'      FUNCTION (STRING MATCH)
IF 'STRINGID1' = ('STRINGID2')    FUNCTION (TEMPLATE MATCH)
```

2. INDICATOR TEST FORMS

```
IF @WB'N' = 0                     FUNCTION (TEST FOR BIT RESET)
IF @WB'N' = ANY NONZERO CHAR.     FUNCTION (TEST FOR BIT SET)
IF @WB'N' = @WB'M'                FUNCTION (TEST FOR SAME STATE)
```

3. VALUE TEST FORMS

```
IF @WC'N' 'RELATIONAL' LITERAL    FUNCTION (COMPARE CONTENTS
                                                 OF WORK CTR TO LITERAL VALUE)
   IF @WC'N' 'RELATIONAL' @WC'M'      FUNCTION (COMPARE CONTENTS OF
                                                 TWO WORK COUNTERS)
   IF @WC'N' 'RELATIONAL' 'STRINGID2' FUNCTION (COMPARE WORK COUNTER
                                                 WITH STRING ELEMENT VALUE)
```

RELATIONAL OPERATORS MAY BE =#><][(I.E. EQ,NEQ,GT,LT,GTE, OR LTE).

'STRINGID1' MAY BE ANY ONE OF THE FOLLOWING:

```
   A            -(OR ANY OF THE LEGAL VARIANTS AS FOR PROC IF
                  STATEMENTS) PARAMETER IS IN THE CURRENTLY ACTIVE
                  INPUT BUFFER.
   @IPN,X       - PRIMARY INPUT BUFFER ELEMENT SELECTION
   @ISN,X       - SECONDARY INPUT BUFFER ELEMENT SELECTION
   @IAN,X       - CURRENTLY ACTIVE INPUT BUFFER ELEMENT SELECTION
   @OPN,X       - PRIMARY OUTPUT BUFFER ELEMENT SELECTION
   @OSN,X       - SECONDARY OUTPUT BUFFER ELEMENT SELECTION
   @OAN,X       - CURRENTLY ACTIVE OUTPUT BUFFER ELEMENT SELECTION
   @WSN,X,Y,Z   - VALUE FROM WORK STRING SELECTION
   @WAN         - WORK ACCUMULATOR SELECTION (AS A STRING)
```

'STRINGID2' MAY HAVE ANY ONE OF THE 'STRINGID1' FORMS WITH THE EXCEPTION OF THE 'A' FORM OR IT MAY BE A LITERAL EXPRESSION.

IF A NON-LITERAL FORM IS USED FOR TEMPLATE MATCH, THE FINAL CHARACTER IN THE ULTIMATE LITERAL STRING (I.E THE TEMPLATE) MUST BE THE CHARACTER ')'. IN SUCH INSTANCES, THE CLOSING PARENTHESIS MAY BE OMMITTED IN THE IF STATEMENT.

THE VALUE RELATION FORM OF THE IF STATEMENT TREATS ALL ELEMENTS AS INTEGERS. I.E. THE CORRESPONDING VALUE FOR A STRING ELEMENT BEGINNING WITH A NON-NUMERIC CHARACTER IS ZERO.

SECTION 5

*** STANDARD SYMBOLIC USER EXITS ***

THE FOLLOWING USER EXITS ARE AVAILABLE FOR USE WITHIN THE PORC PROCEDURES. THE SYMBOLIC USER EXIT LIST MAY BE MODIFIED TO INCLUDE THOSE FUNCTIONS WHICH A PARTICULAR SYSTEM UTILIZES. THE EXITS LISTED HERE HAVE BEEN FOUND TO BE USEFUL IN SOME OF THE APPLICATIONS WHICH ARE BEING PROCESSED ON THE REALITY SYSTEM.

FUNCTION DESCRIPTION

-- CP

CURSOR POSITIONING FOR THE ADDS TERMINAL
THE STANDARD CURSOR POSITIONING COMMANDS ARE AVIALABE WITH THE COMMANDS INCLUDED ON THE LINE WITH THE FUNCTION. TEXT MAY ALSO BE OUTPUT IN ANY DIRECTION USING THE FORM TN/TEXT/ WHERE N IS IN RANGE 0 - 7. THE DIRECTION CORRESPONDS TO ONE OF THE 8 POINTS OF THE COMPASS WITH 0 AS EAST AND 2 AS NORTH.

```
U CP CLEAR,BELL,TO*MESSAGE*,T6/DOWNWARD/,T4%BACKWARD%
```

-- CALC

ARITHMETIC CALCULATOR PACKAGE
THIS FUNCTION PROVIDES CAPABILITY TO DO FLOATING POINT ARITHMETIC FUNCTIONS AND STORE THEM IN A FORMATED FORM.

-- STRIP

STRIP TRAILING BLANKS FROM DATA STORED IN THE SECONDARY OUTPUT BUFFER. THE CURRENT POSITION OF THE INPUT BUFFER POINTER IS NOT CHANGED. THE END OF THE SECONDARY INPUT BUFFER IS FOUND AND BACKSCANED UNTIL NO BLANKS EXIST.

-- DISP

DISPLAY DATA FROM ONE OF THE WORK STORAGE AREAS AT THE TERMINAL. THE GENERAL FORM OF THE FUNCTION IS:

```
U DISP XYNN @ABC @DEF
```

X IS AN ONE OF THE FOLLOWING:
- I - DISPLAY AN ITEM WITH 1 ATTRIBUTE PER LINE
- A - DISPLAY AN ATTRIBUTE WITH 1 VALUE PER LINE
- V - DISPLAY A VALUE WITH 1 SUBVALUE PER LINE
- S - DISPLAY A SUBVALUE WITH 1 ELEMENT PER LINE

YNN MAY BE ANY OF THE FOLLOWING COMBINATIONS:
- N - INSERT COUNT NUMBERS PRECEEDING EACH LINE
- L - INSERT LINE NUMBERS PRECEEDING EACH LINE
- SNN - SKIP NN BLANKS BEFORE STARTNG LINE
- SCN - SKIP COLUMNS SPECIFIED IN WORKCOUNTER N
- LNN - INSERT LINE NUMBERS PRECEEDING EACH LINE AND POSITION DATA STARTING AT COLUMN NN

LINE NUMBERS START WITH 1 AND COUNTS WITH 0
BOTH COUNT AND LINE NUMBERS ARE PLACED STARTING IN COLUMN 1 WITH THE FOLLOWING FORMAT '(NN) '.
IN ALL CASES AT LEAST ONE BLANK FOLLOWS THE NUMBER.
THE NUMBER OF COLUMNS TO SKIP IS INDEPENDENT OF THE

NUMBERS UNLESS THE NUMBERS FLOW INTO THE AREA WHERE THE DATA IS TO BE PLACED.

@ABC IS THE LOGICAL POSITION AT WHICH TO START DISPLAY

@DEF IS THE LOGICAL POSITION AT WHICH THE FORMAT IS TO BE FOUND. IF THE FORMAT IS SPECIFED IN THE FORM (FILE ITEM), THAT FILE AND ITEM WILL BE USED FOR THE FORMAT, OTHERWISE THE FORMAT WILL BE FOUND DIRECTLY.

SECTION 6

*** FILE ACCESS USING PORC PRIMITIVES ***

A PORC PRIMITIVE EXISTS TO EXAMINE FILE STRUCTURES AND TO MANIPULATE ITEMS. THE PREFIX ´F-´ IS USED IN COMBINATION WITH COMMAND ABBREVIATIONS TO FORM ALL OF THE PRIMITIVES FOR FILE ACCESS. THE ´READ´ AND ´WRI´ (WRITE) PRIMITIVES COMPLEMENT THE FILE PRIMITIVE AND ALLOW ITEMS TO BE READ AND WRITTEN. ALL ITEMS ARE WRITTEN BY PLACING THE ITEM IN THE HISTORY STRING AND THEN ALLOWING WRAP-UP TO PROCESS THE HISTORY STRING. THE PRIMITIVE ´WRAP´ ALLOWS THE PORC USER TO PROCESS THE HISTORY STRING AT HIS DISCRETION.

EACH OF THE PRIMITIVES ARE DESCRIBED AND A SMALL EXAMPLE IS INCLUDED FOR THE USE OF THE FILE ACCESS PRIMITIVES.

FIVE FILES, EACH CONSISTING OF TWO LEVELS, MAY BE PROCESSED USING THE LOGICAL DESIGNATORS @WF0 - @WF4.

EACH WORK FILE WHICH HAS BEEN OPENED HAS TWO SECTIONS, DESIGNATED BY @WFX,0 AND @WFX,1. THE LEVEL 0 SECTION IS TYPICALLY THE DICTIONARY SECTION OF A FILE AND THE LEVEL 1 SECTION IS THE DATA SECTION. IF THE FILE IS OPENED TO THE DICTIONARY SECTION, THE LEVEL 0 SECTION WILL CONTAIN THE FILE WHERE THE DESCRIPTOR FOR THE FILE WAS FOUND, I. E. THE MASTER DICTIONARY FOR THE USERS ACCOUNT.

THE FOLLOWING CHART DESCRIBES THE TEN FILE STORAGE LOCATIONS:

WORK FILE 0, LEVEL 0	WORK FILE 0, LEVEL 1
WORK FILE 1, LEVEL 0	WORK FILE 1, LEVEL 1
WORK FILE 2, LEVEL 0	WORK FILE 2, LEVEL 1
WORK FILE 3, LEVEL 0	WORK FILE 3, LEVEL 1
WORK FILE 4, LEVEL 0	WORK FILE 4, LEVEL 1

IN THE FOLLWING DESCRIPTIONS, THE FOLLOWING CONVENTIONS ARE USED TO DEFINE THE WORK FILES.

```
   @WFB   -   IMPLIES WORK FILE 0 - 4 AT BOTH LEVEL 0 AND 1
   @WFM   -   IMPLIES WORK FILE 0 - 4 AT LEVEL 0 OR 1  (SOURCE)
```

@WFN - IMPLIES WORK FILE 0 - 4 AT LEVEL 0 OR 1 (DESTINATION)

WORK BIT 0 (@WB0) BEING SET IMPLIES IT IS EQUAL TO 1
WORK BIT 0 (@WB0) BEING CLEARED IMPLIES IT IS EQUAL TO 0

THE STANDARD TWO LEVEL FILE STRUCTURE OF REALITY MAY BE EXPANDED USING THESE PRIMITIVES TO A ´N´ LEVEL STRUCTURE. TO ACHIEVE THIS AND MAINTAIN THE FILE SAVE/RESTORE A PSEUDO-MULTILEVEL FILE IS HANDLED VIA A FILE ITEM WITHIN A DICTIONARY CONTAINING AS THE FIRST ATTRIBUTE AN "FL" FOR FILE LINK; THE SECOND ATTRIBUTE CONTAINS THE NAME USED TO REFERENCE THE FILE IN THE MASTER DICTIONARY.

PRIMITIVES AVAILABLE ARE:

F-OP @WFB ´NAME´ FILE-OPEN -- OPEN THE FILE ´NAME´ WITH WORK FILE B DESIGNATED AS THE STORAGE FOR BASE, MODULO, AND SEPARATION FOR THE FILE. ´NAME´ MAY BE ANY FILE NAME DEFINED IN THE USERS ACCOUNT OR ANY FILE WHICH IS LINKED VIA A Q ENTRY IN THE USERS MASTER DICTIONARY. WORK BIT 0 (@WB0) IS SET TO DESIGNATE THE FILE WAS SUCCESSFULLY OPENED.
IT IS ALSO POSSIBLE TO OPEN ONLY THE DICTIONARY SECTION OF A FILE BY USING THE FORM ´DICT NAME´ INDIRECT POINTERS OF THE FOLLOWING FORMS ARE ALSO ACCEPTABLE: ´@WS1,1´ OR ´@IS2´ WHERE @IS2 MAY CONTAIN ´DICT @IP2´, ETC.

F-OL @WFB @WFM ´NAME´ FILE-OPEN-LINKED -- THIS IS SIMILIAR TO THE F-OP FORM EXCEPT THE FILE WHICH IS USED TO FIND THE FILE IS NOT NECESSARILY THE MASTER-DICTIONARY. THE SEARCH FOR THE FILE DESCRIPTOR IS BEGUN AT THE FILE DESIGNATED BY @WFM.

F-OM @WFN ´NAME´ OPEN-MASTER-DICTIONARY -- THIS FUNCTION WILL OPEN THE FILE SPECIFED IN THE MASTER DICTIONARY. THE OPEN IS LIMITED TO THE FIRST LEVEL OF THE FILE. IF THE FILE CANNOT BE OPENED OR IF THE FILE DOES NOT EXIST, WORK BIT 0 IS CLEARED. THIS FORM OF THE OPEN USES ONLY ONE OF THE 10 FILE STORAGE CELLS RATHER THAN TWO.

F-OF @WFN @WFM ´NAME´ OPEN-FILE-FROM-LEVEL -- THIS VERSION OF THE OPEN IS SIMILIAR TO F-OM EXCEPT THAT IT BEGINS THE OPEN PROCEDURE AT THE FILE DESIGNATED BY @WFM. AN ITEM DESIGNATED BY ´NAME´ IS LOOKED FOR. IF THIS ITEM DOES NOT EXIST OR IF A FILE CANNOT BE FOUND BY USING THIS ITEM AS A BASE, WORK BIT 0 IS CLEARED. THE FIRST ATTRIBUTE OF AN ITEM MAY BE A ´D´, ´Q´, OR ´FL´ TYPE TO SUCCESSFULLY OPEN THE FILE.

F-C @WFM ´NAME´ FILE-CHECK -- THIS FUNCTION CHECKS TO DETERMINE IF AN ITEM EXISTS IN THE DESIGNATED FILE.

F-M @WFN @WFM FILE-MOVE -- THIS FUNCTION WILL MOVE THE BASE, MODULO, AND SEPARATION FROM ONE FILE SPACE

TO ANOTHER FILE SPACE. IT IS USED TO OBTAIN ACCESS TO MORE THAN FIVE FILES SIMULTANEOUSLY.

F-IS @WFM — FILE-INITIALZIE-SEQUENTIAL -- THIS FUNCTION WILL SET-UP A TEMPORARY FILE TO BE USED TO ACCESS ALL ITEMS WITHIN THE DESIGNATED FILE. THE ITEMS ARE DETERMINED BY USING THE RELATED FUNCTION F-NS.

F-XS — FILE-EXTERNAL-SEQUENTIAL -- THIS FUNCTION WILL UTILIZE THE SEQUENTIAL LIST OF ITEMS WHICH IS GENERATED BY THE "ENGLISH" VERBS ´SELECT´ AND THE SORT VERSION OF ´SELECT´ CALLED ´SSELECT´. IT ALSO USES THE RELATED FUNCTION F-NS.

F-NS @ABC — MOVE THE NEXT ITEM IN THE SEQUENTIALIZED LIST TO THE WORKSPACE DESIGNATED BY @ABC. WORK BIT 0 IS SET IF THE ITEM WAS MOVED INTO THE SPACE. IF THE LIST WAS EXHAUSTED OR NOT INITIALIZED, WORK BIT 0 IS CLEARED.

READ @WFN @ABC ´NAME´ — READ-ITEM -- THIS FUNCTION WILL READ AN ITEM FROM THE FILE DESIGNATED BY @WFN INTO THE STORAGE SPACE @ABC. ´NAME´ IS THE NAME OF THE ITEM. WORK BIT 0 WILL BE SET IF THE ITEM WAS SUCCESSFULLY MOVED INTO THE WORK SPACE.

WRI @WFN @ABC — WRITE-ITEM -- THIS FUNCTION WILL WRITE AN ITEM DESIGNATED BY ATTRIBUTE 0 OF THE WORKSPACE @ABC. THE FILE DESCRIBED @WFN IS USED AS THE BASIS FOR THE UPDATE. THIS FILE DESIGNATION MUST BE OF THE FORM @WFN,1. THE WRITE FUNCTION IS NOT ALLOWED TO OCCUR INTO THE ZERO´TH LEVEL OF THE FILE USING THE WRITE COMMAND.

THE FOLLOWING EXAMPLE IS AN ILLUSTRATION OF HOW THE PORC FILE ACCESS ROUTINES MAY BE USED TO OPEN AND PROCESS DATA FROM FILES.

THE FILE ´CUSTOMERS´ CONTAINS CUSTOMER ACCOUNT NUMBERS. IT IS DESIRED TO DETERMINE IF A CUSTOMER NUMBER (ITEM IN THE FILE) EXISTS AND IF IT DOES TO READ THE ITEM INTO WORK SPACE 1 STARTING AT THE BEGINNING OF THE WORK SPACE. WITHIN THE ITEM, ATTRIBUTE TWO CONTAINS THE NAME OF THE ITEM IN THE FILE ´HISTORY´ WHICH DEFINES THE NAME OF THE FILE CONTAINING THE CUSTOMERS HISTORY. (THIS IS EFFECTIVELY A MULTI-LEVEL LINKED FILE) THE CUSTOMERS HISTORY FILE IS OPENED AND THE FILE POINTER IS STORED AT WORK FILE 3, LEVEL 1.

```
F-OPEN @WF1 CUSTOMERS
IF @WB0 = 0 XEXIT - FILE ´CUSTOMERS´ DOES NOT EXIST
READ @WF1,1 @WS1,0 @IP2  READ THE ITEM FROM THE ´CUSTOMER´ FILE
IF @WB0 = 0 XEXIT - CUSTOMER DOES NOT EXIST
F-OPEN @WF2 HISTORY
IF @WB0 = 0 XEXIT - FILE ´HISTORY´ DOES NOT EXIST
F-OF @WF3,1 @WF2,1 @WS1,2 OPEN LINKED FILE SPECIFIED TO THE FILE
*                         CONTAINING THE HISTORY FOR THE CUSTOMER.
IF @WB0 = 0 XEXIT - CUSTOMERS HISTORY FILE DID NOT EXIST
```

SECTION 7

*** EXAMPLES OF PORC PROCEDURES ***

THE FOLLOWING EXAMPLES ILLUSTRATE SOME OF THE WAYS PORC PROCEDURES MAY BE USED. ALL ARE ACTUAL EXAMPLES AND WILL OPERATE AS IS, INCLUING THE COMMENTS.

```
**********************************************************************
*
*  BEGINNING OF PORC EXAMPLE 1  -  CONSISTS OF 3 ITEMS IN THE
*  FILE 'TSP' WITH ITEMS NAMED 'EX-1', 'EX-2', AND 'EX-3'.
*
*  AN ENTRY IN THE MASTER DICTIONARY OF THE FOLLOWING FORM
*  IS REQUIRED TO TRANSFER CONTROL TO THE 'TSP' FILE.
*
* FILE: M/DICT
* ITEM: P-TS
*
*  001 PQCL
*  002 TSP
*  003 *@IP2
*
*  THE THIRD ATTRIBUTE SAYS TO GO TO SECOND COMMAND PARAMETER
*  TO FIND THE NAME OF THE ITEM TO TRANSFER TO.
*
*  TO EXECUTE ANY PORC IN THE DATA SECTION OF THE TSP FILE,
*  THE FOLLOWING COMMAND MAY BE USED.  THE SECOND PARAMETER IS THE
*  NAME OF THE ITEM CONTAINING THE PORC.
*
* :P-TS EX-1
*
**********************************************************************
```

```
PQC
*
* THIS IS A COMMENT AREA WHICH IS INVISIBLE TO THE PORC PROCESSOR
* IT MAY BE USED TO DESCRIBE THE PORC AND DOES NOT EFFECT THE TIME
* REQUIRED FOR LABEL SEARCHES.  THE PORC PROCESSOR DOES NOT SCAN THIS
* AREA BECAUSE THE PORC REALLY STARTS ON THE FIRST LINE WHICH DOES
* NOT HAVE AN ´*´ SYMBOL IN COLUMN 1.
*
001 C             THIS IS LABEL LINE 1
0DEMONSTRATION OF BASIC PORC CAPABILITIES
C
C COMMENTS MAY OCCUR FOLLOWING MOST STATEMENTS
C THEY ARE TABBED OVER IN THESE EXAMPLES FOR CLARITY
C
IN+                INPUT A MESSAGE
IF A GO L.SECT-2   GO TO SECTION 2 IF SOMETHING WAS ENTERED
*
* ASTERISK LINES WITHIN A PORC ARE SIMILAR TO COMMENT
* LINES EXCEPT THEY ARE IGNORED BY FUNCTIONS WHICH
* SKIP FORWARD OR BACKWARD A NUMBER OF LINES.
*
02 OPLEASE ENTER SOMETHING
SKIP -8            SKIP BACK TO PREVIOUS OUTPUT STATEMENT
C
C IN THE ABOVE SKIP STATEMENT, THE 5 LINES WITH AN ASTERISK WERE IGNORED
C
L.SECT-2 C         LABEL FOR GO FUNCTIONS MAY BE EITHER LITERALS
C                  OR NUMERICS.  IF NUMERICS ARE USED, LEADING ZEROS
C                  ARE SUPPRESSED.  IF LITERALS ARE USED, THE LITERAL
C                  STRINGS MUST MATCH TO A BLANK OR AM DELIMITER.
C
IF A2 SKIP 4       SKIP NEXT 4 LINES IF TWO ELEMENTS WERE ENTERED
*
OPLEASE ENTER TWO ELEMENTS NEXT TIME
O
GO 00001           GO TO LINE 1 OF PORC
*
IFN A = (ON) OFORGOT TO MENTION, THE SECOND ELEMENT SHOULD BE NUMERIC
C
C THE PREVIOUS STATMENT USED THE NEGATED VERSION OF THE IF FUNCTION
C
IF A = (ON) EXIT   EXIT IS SIMILIAR TO ´X´ EXCEPT IT MAY BE USED
G 1                AS A SUBROUTINE RETURN AS WELL AS A DIRECT EXIT.
C
```

```
PQC
*
* THIS PORC CONTAINS EXAMPLE OF TRANSFER OF CONTROL
* THESE FUNCTIONS ARE:
*                           TRANS  (TRANSFER WITH NO RETURN)
*                           CALL   (SUBROUTINE CALL CAPABILTIY)
*                           RETN   (RETURN TO SAVED POSITION)
*                           EXIT   (RETURN IF POSITION SAVED)
*                           MARK   (SAVE CURRENT POSITION)
*                           MDEL   (DELETE CURRENT POSITION)
*
CALL (TSP EX-1) 2  ENTER EXAMPLE 1 AT LINE 2 TO PROCESS IT
*                  CALLS AND TRANSFERS MAY BE TO A LABELED LINE
*                  WITHIN A PORC AS WELL AS AT THE BEGINNING
*
CALL * 200         CALL LOCAL SUBROUTINE AT LABEL LINE 200
*                  THE '*' DESIGNATES IN THE SAME PORC
*
XFINISHED
C
C THE RETURN FROM A CALL (EITHER RETN OR EXIT) MAY BE RELATIVE TO
C THE CALL BY USING A RETURN OF THE FORM 'RETN N' WHERE N IS A
C POSITIVE OR NEGATIVE VALUE OR N MAY BE A WORK COUNTER DESIGNATOR
C
00200 C            THIS IS THE LOCAL SUBROUTINE
*
OENTER YOUR AGE IN YEARS
*
CALL * L.INPUT     GO TO INPUT SUBROUTINE
IFN A = (ON) IFN A = (-ON) SKIP -2
*
MOVE @WC1 @IS1
IF @WC1 [ 0 RETN   IF ZERO OR NEGATIVE, IT IS TIME TO EXIT
IF @WC1 > 120 ODID YOU USE METRIC YEARS ???
*
MARK               SAVE THIS POSITION
O*+
SUB @WC1 1         DECREMENT VALUE
IF @WC1 > 0 RETN 0
MDEL               DELETE LAST RETURN POINT
O
RETN 0             GO BACK AND DO IT AGAIN
*
* SUBROUTINE TO GET INPUT DATA
*
L.INPUT C
IN#
IF A RETN
RETN -1            GO BACK AND ASK AGAIN
C
```

```
PQC THIS IS A VERY SIMPLE PORC WHICH IS MAY BE USED TO EXAMINE ITSELF
*
* IT OPEATES BY USING THE INDIRECT ADDRESS POINTERS AND THE FILE
* ACCESS ROUTINES.  IT ALSO DEMONSTRATES THE ABILITY TO BE BOTH
* A PORC AND A PORC SUBROUTINE AS EXAMPLE 1 ALSO DID
* THE PORC WHICH IS DISPLAYED IS ACTUALLY THE ONE THAT WAS CALLED
* WHEN THE ORIGINAL VERB WAS ENTERED UNLESS THE PRIMARY OUTPUT
* BUFFER HAS BEEN MODIFIED.
*
F-OPEN @WF1 TSP    OPEN THE FILE CONTAINING THE PORCS
READ @WF1,1 @OPO @IP2    READ THE ITEM NAMED IN THE PRIMARY
*                        INPUT BUFFER AT ELEMENT 2
IF @WBO = 1 SKIP 3       SKIP 3 LINES IF ITEM WAS READ
OITEM DID NOT EXIST
EXIT
*
* DISPLAY THE ITEM USING THE PRIMITIVE EXAMINE FUNCTION
*
PEEK @OPO          LOOK AT THE ITEM - ONE ATTRIBUTE PER LINE
*
PAUSE NOTE: ALL STRING CONTROLS ARE DISPLAYED TO FIND TRAILING BLANKS
*
EXIT               FINISHED - EITHER EXIT OR SUBROUTINE RETURN
```

*** EXAMPLES OF DATA MOVEMENT ***

THE FOLLOWING EXAMPLES ILLUSTRATE SOME OF THE DATA HANDLING FEATURES AVAILABLE THRU THE USE OF THE PORC PRIMITIVES. CENTRAL TO THE UNDERSTANDING OF MUCH OF THE DATA MANIPULATION IN REALITY IS THE RECOGNITION OF THE DATA HIERARCHY STRUCTURE. SOME ELEMENTS IN THE HIERARCHY MAY NOT OCCUR EXPLICITLY IN THE TYPICAL FILE STRUCTURE BUT ARE NEVERTHELESS IMPORTANT WHEN MANIPULATING DATA WITHIN PORCS.

THE PREMIER DELIMITER IS A 'SEGMENT MARK' (X'FF'), WITH AN 'ATTRIBUTE MARK' (X'FE') NEXT IN THE PECKING ORDER FOLLOWED BY A 'VALUE MARK' (X'FD') AND A 'SUBVALUE MARK' (X'FC'). BECAUSE OF THE WIDESPREAD USE OF BLANKS AS DELIMITERS, THE DELIMITER HIERARCHY IN PORCS IS EXTENDED TO INCLUDE BLANKS AT THE LOWEST RUNG IN THE LADDER.

THE IMPORTANCE HERE IS THAT ELEMENTS OF A GIVEN LEVEL IN THE STRUCTURE MAY BE DELIMITED BY A CHARACTER OF THEIR OWN LEVEL OR ANY OF A HIGHER LEVEL. WITHOUT EXCEPTION, ALL STRING STORAGE LOCATIONS UNDER THE CONTROL OF THE PORC PROCESSORS WILL CONTAIN A SEGMENT MARK AT THEIR CURRENT END POSITION. HENCE EXECUTION OF ANY PORC DATA MOVEMENT FUNCTION WILL, AT MOST, MOVE DATA THRU THE SEGMENT MARK UNLESS THERE ARE PARAMETERS SPECIFIYING EARLIER TERMINATION OF THE OPERATION.

TO ILLUSTRATE, A 'MOV' OF STRING WITH THE COUNT PARAMETER OMITTED, WILL MOVE ALL DATA FROM ONE CHARACTER STRING TO ANOTHER UNTIL THE END OF THE IMPLIED DATA HIERARCHY WHICH IS DEFINED AS THE OCCURENCE OF A DELIMITER OF ANY HIGHER LEVEL. A 'MOVA' (MOVE ATTRIBUTES) WITH NO COUNT WILL TRANSFER ALL DATA THRU THE SEGMENT MARK INCLUDING ANY DELIMITERS.

FOR STRING MOVES, THE LAST DATA CHARACTER TRANSFERRED WILL BE THE DELIMITER WHICH FORCED TERMINATION OF THE MOVEMENT.

MOVEMENTS TO AND FROM WORK COUNTERS CARRY WITH THEM AN IMPLIED CONVERSION FROM CHARACTER TO BINARY REPRESENTATIONS OF THE DATA AND VICE VERSA.

****

*** DATA MOVEMENT EXAMPLE 1 ***

THIS PORC EXISTS IN THE FILE 'SAMPLES' WITH ITEM NAME 'MOV-1'

THE DIALOGUE IS:

:MOV-1 TYPE ANYTHING ELSE YOUR HEART DESIRES ON THE INPUT LINE

IT WILL PROMPT YOU FOR MORE AND THEN DISPLAY WHAT YOU TYPED IN A SLIGHTLY SCRAMBLED FORM, FIRST DISPLAYING THE NUMBER OF ELEMENTS YOU TYPED (BLANKS AS SEPARATORS) IN THE INITIAL INPUT LINE AND IN THE PROMPT LINE AND THEIR TOTAL.

```
PQC
*
* COPY PARAMETERS FROM INPUT BUFFERS TO PRIMARY OUTPUT BUFFER
*
OINPUT ANY THING
IN#
*
MOVE @OP0 @IS1 1      MOVE ONLY FIRST ELEMENT FROM SECONDARY INPUT
MOVV @WC2 @WC0        KEEP ELEMENT COUNT FOR SECONDARY IN WRK CTR TWO
MOVD @OP BL           MAKE SURE BLANK FOLLOWS
MOVE @OP @IP1 1       MOVE ONLY FIRST ELEMENT FROM PRIMARY INPUT
MOVV @WC1 @WC0        KEEP ELEMENT COUNT FOR PRIMARY IN WORK COUNTER ONE
*
* NOW MOVE REMAINDER OF INPUT BUFFERS
*
MOVE @OP @IS2         NO COUNT MEANS TO MOVE TILL END OF THIS HIERARCHY
*
* IF SOME DATA WAS MOVED, REPLACE THE END OF INPUT BUFFER DELIMITER
* WITH AN ELEMENT DELIMITER (BLANK) BECAUSE WE'RE GOING TO ADD
* DATA FROM THE PRIMARY INPUT BUFFER.
*
IF @WB0 = 1 MOVD @OP BL
ADD @WC2 @WC0         WC0 HAS ACTUAL COUNT OF ELEMENTS MOVED
MOVE @OP @IP2         GET BALANCE OF PRIMARY INPUT BUFFER
ADD @WC1 @WC0
*
* NOW LETS WRITE IT ALL BACK ON THE SCREEN
*
MOVA @WS1,0 @WC1      MOVE AND CONVERT THE ELEMENT COUNT FROM PRIMARY INPUT
MOVA @WS1,1 @WC2      MOVE AND CONVERT THE ELEMENT COUNT FROM SECONDARY
ADD @WC1 @WC2
MOVA @WS1,2 @WC1      MOVE AND CONVERT THE TOTAL ELEMENT COUNT
*
U CP CLEAR,TO/ELEMENT COUNTS DISPLAYED  WITH LINE NUMBERS/,(0,2)
*
* DISPLAY THE ITEM (ALL THE ATTRIBUTES) IN WORK STRING 1 PREFACING
* EACH ATTRIBUTE WITH A LINE NUMBER AND INDENTING 10 SPACES FROM
* THE MARGIN.
*
U DISP IL10 @WS1,0
*
```

```
* DISPLAY THE PRIMARY OUTPUT BUFFER AS A SUBVALUE SO THAT EACH
* ELEMENT DELIMITED BY A BLANK WILL APPEAR AS A SEPARATE LINE.
* EACH ELEMENT IS PREFACED WITH A COUNT NUMBER AS DISTINCT FROM
* A LINE NUMBER AS ABOVE.  COUNT NUMBERS BEGIN WITH ZERO RATHER
* THAN ONE.  ALSO INDENT 10 SPACES FOR THIS EXAMPLE.
*
U CP (0,6),TO/ELEMENTS IN PRIMARY OUTPUT BUFFER ARE:/,(0,8)
U DISP SN10 @OPO
EXIT
```

*** DATA MOVEMENT EXAMPLE 2 ***

THE FOLLOWING EXAMPLE EXISTS IN THE FILE ´SAMPLES´ WITH THE ITEM NAME OF ´MOV-2´. THIS EXAMPLE IS NOT COMPLETE AND EXECUTABLE BY ITSELF IN ORDER THAT IT MAY SERVE AS A COMPACT EXAMPLE.

FOR THE EXAMPLE, IT HAS BEEN ASSUMED THAT IT HAS BEEN CALLED BY A MAIN PROGRAM WITH THE FIRST LINE FOLLOWING THE CALL SERVING AS AN ERROR RETURN LINE AND THE SECOND LINE THE NORMAL RETURN POINT.

THIS ROUTINE CONTROLS TWO SETS OF THREE CO-RELATED ATTRIBUTES WHICH ARE ´LEDGER´, ´AMOUNT´ AND ´COMMENTS´ EACH FOR DEBIT AND CREDIT ENTRIES INTO A JOURNAL. THE DEBITS AND CREDITS MUST BALANCE IN ORDER THAT THE NORMAL RETURN MAY BE TAKEN.

```
FQC                          START OF THE EXAMPLE
GO L.DEBITS                  GET DEBIT ENTRIES FIRST
*
* THE ESSENCE OF THE EXAMPLE IS PERFORMED IN THE FOLLOWING
* INTERNAL SUBROUTINE.
*
L.INPUT C
MOVV @WA1 O                  INITIALIZE DECIMAL ACCUMULATOR
MOVV @WC1 O                  INITIALIZE VALUE INDEX
*
OLEDGER  AMOUNT  COMMENTS
MARK
IN#
IFN A GO L.MERGE             IF NO ENTRY, MERGE DATA INTO ITEM
MOVE @WS1,O,@WC1 @IA1 1      GET LEDGER NUMBER
MOVD @WS1 VM                 PUT PROPER DELIMITER
IF A2 SKIP 4
MOVV @WS2,O,@WC1 O           SUBSTITUTE ZERO AMOUNT FOR NO AMT ENTRY
MOVX @WS3,O,@WC1 FD          CREATE A NULL VALUE IN ´COMMENTS´
RETN O                       GO BACK TO THE MARK ABOVE
MOVE @WS2,O,@WC1 @IA2 1      GET AMOUNT ENTERED
MOVD @WS2 VM                 DELIMIT IN STRING
U CALC
(@IA2 + @WA1 )?              SUM THE NEW DOLLAR AMOUNT TO ACCUMULATION
I @WA1                       RETAIN INTERNAL FORMAT..RESULT BACK TO ACCUM.
IFN A3 SKIP -7               DO SKIP IF NO COMMENT
MOVE @WS3,O,@WC1 @IA3 1      ADD VALUE TO COMMENT ATTRIBUTE
MOVD @WS3 VM
RETN O                       GET NEXT ENTRIES
*
* MERGE THE THREE SCRATCH STRINGS EACH AS AN ADDITIONAL ATTRIBUTE
* IN THE TRANSACTION ITEM.
*
L.MERGE MOVA @OP @WS1,O      GET STRING OF LEDGER NUMBERS
MOVD @OP AM                  FORCE IN ATTRIBUTE MARK
MOVA @OP @WS2,O              GET SEQUENCE OF AMOUNTS
MOVD @OP AM
MOVA @OP @WS3,O              GET COMMENTS
MDEL                         POP MY RETURN STACK
RETN
```

```
*
L.DEBITS 0 -ENTER DEBITS-
CALL * L.INPUT
MOVV @WA2 @WA1               SAVE DEBIT ACCUMULATION
MOVV @WC2 @WC1               SAVE COUNT OF NUMBER OF DEBIT ENTRIES
*
* NOW GET THE CREDIT ENTRIES
*
CALL * L.INPUT
IF @WA1 = @WA2 RETN 2        NORMAL RETURN IF CREDITS AND DEBITS BALANCE
*
O** BALANCE ERROR **
MOVV @ISO @WC2               DISPLAY SOME INFORMATION FOR THE OPERATOR
MOVV @IS @WC1
ONUMBER OF DEBITS/CREDITS
U DISP A15 @ISO
RETN                         TAKE ERROR RETURN
```