



SCULPTOR

SAMPLE PROGRAMS

SAMPLE PROGRAMS

APPENDIX A. SAMPLE PROGRAMS

Appendix A contains sample programs designed to demonstrate some of the features and techniques used in SCULPTOR programming. The programs should not be taken as an example of a complete stock system, although individual techniques shown could well be incorporated in such a system.

The programs are as follows:

Program	Page
Menu program: menu.m	A-2
Data Dictionaries: order.d and stock.d	A-3
Orders file program: order.f	A-4
Stock file program: stock.f	A-14
Orders report program: order.r	A-17
Pricelist report program: prlist.r	A-18
Stock report program: stock.r	A-19
Supplier report program: supplier.r	A-20

The menu program was written as a straightforward text file. The Data Dictionaries were written using the **describe** program. The orders file program was written using the SCULPTOR screen form language and demonstrates the use of "scroll" fields and simple file cross-referencing. The stock file program was created using the SCULPTOR program **sg**. All the report programs were generated using the SCULPTOR program **rg**, but the suppliers report program was then modified to include the technique of passing parameters from the menu program. (Refer to option 4 on the menu and the first two **!init** statements in the supplier.r program.)

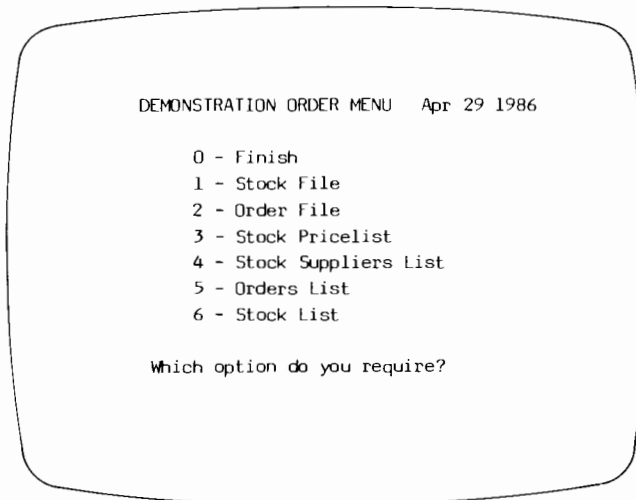
MENU TEXT FILE

DEMONSTRATION ORDER MENU

```
1,Stock File
sagerep stock
2,Order File
sagerep order
3,Stock Pricelist
sagerep prlist pvdu |more ;pause
4,Stock Suppliers list
sagerep supplier pvdu % |more ;pause
Enter Supplier .. ( or RETURN for all) .....
5,Orders list
sagerep order printer |spooler
6,Stock List
sagerep stock printer |spooler
```

NOTE: In options 3 and 4, **pvdu** is the driver file used to control **sagerep** output to the screen instead of a printer. The **|more** command is a standard UNIX command to allow page at a time display on the screen: this command will differ under other operating systems. The **pause** command prevents the final display from being erased by a return to the menu. In options 5 and 6, the output is directed to the printer.

DEMONSTRATION ORDER MENU



DATA DICTIONARIES

Descriptors for order.d

KEY FIELDS

1:o_ordno,Order No.,a10,u

DATA FIELDS

2:o_date,Date,d4,

3:o_cust,Customer,a30,u

4:o_add1,Address,a30,

5:o_add2, ,a30,

6:o_add3, ,a30,

7:o_pcode,Code,a7,u

8:o_qty,Qty,i4(10),#####

9:o_unit,Unit,a6(10),

10:o_part,Part No.,a10(10),u

11:o_desc,Description,a20(10),

12:o_uprice,Unit Price,m4(10),###.##

13:o_deldate,Delivery Date,d4,

Descriptors for stock.d

KEY FIELDS

1:s_part,Part No.,a10,u

DATA FIELDS

2:s_desc,Description,a20,

3:s_unit,Unit,a6,

4:s_ucp,Unit Cost,m4,

5:s_uprice,Unit Price,m4,

6:s_qty,Quantity,i4,

7:s_min,Min level,i2,

8:s_backord,Qty on order,i4,

9:s_lead,Lead time,i2,

10:s_supp,Supplier,a30,u

11:s_deldate,Exp delivery date,d4,

ORDERS FILE PROGRAM: order.f

This is a demonstration program showing some of the features of SCULPTOR such as scrolling and file interrelations. It is not meant to be a complete order program as a number of features required have been omitted for clarity in the example.

ORDER FILE

```
-----FILES-----  
!file order order  
!file stock stock
```

```
-----TEMPORARY FIELDS-----  
!temp date,Date,d4  
!temp scrip, ,il  
  
!temp func, ,a10  
!temp totpr,Total Price,m4  
!temp price,Price,m4  
!temp item, ,il  
!temp qty,,a9  
!temp oqty,,i4
```

```
-----BOX LIST-----  
!scroll 9,10
```

```
+ func,,1,2  
+ date,,1,71  
+ o_ordno,,3,15  
+ o_date,,5,15  
+ o_cust,,3,40  
+ o_add1,,4,40  
+ o_add2,,5,40  
+ o_add3,,6,40  
+ o_pcode,,7,40  
+ item,,9,2, ##  
+ o_qty,,9,6  
+ o_part,,9,15  
+ o_desc,,9,28  
+ o_unit,,9,50  
+ o_uprice,,9,60, ##### . ##  
+ price,,9,72, ##### . ##  
+ totpr,,21,70, ##### . ##
```

. INIT - Initialisation routine

```
.  
INIT \  
    display date  
    gosub UNLOCK  
IN99 \  
    end
```

----- OPTIONS -----

```
*i = insert  
    func = "INSERT": display func  
    clear o_ordno - o_pcode  
    message "Backspace to finish inserting ... "  
    gosub CLEARSCROLL  
I10 \  
    input o_ordno bs = I99  
    if (o_ordno = "") then \  
        error "Must enter order number": \  
        goto I10  
    testkey order nsr = I14  
    error "This order number already used": goto I10
```

... Set order date to system date as default.

```
I14 \  
    o_date = date: display o_date  
I17 \  
    input o_date bs = I10  
I20 \  
    input o_cust bs = I17  
    if (o_cust = "") then \  
        error "Must enter customer ... ": \  
        goto I20  
I30 \  
    input o_add1 - o_pcode bs = I20
```

(cont.)

... Enter up to 10 items in the order

```
I40 \  
    scroll 1: totpr = 0  
I45 \  
    item = scline: display item  
I50 \  
    input o_qty bs = I65  
    if (o_qty = 0) then goto I70  
I55 \  
    input o_part bs = I50  
    if (o_part = "") then \  
        error "Must enter part no. ... ": \  
        goto I55
```

... Read the stock file to get the description, unit of issue
... and unit price

```
    read stock key = o_part nsr = I60
```

... If the quantity in stock is less than the quantity ordered
... then make the order for the actual amount available

```
    if (o_qty > s_qty) then \  
        qty = s_qty: \  
        error "Only " + qty + " in stock" : \  
        o_qty = s_qty
```

... Adjust the stock record for the amount ordered

```
    s_qty = s_qty - o_qty  
    write stock  
    o_desc = s_desc: o_uprice = s_uprice  
    price = (o_uprice * o_qty) :o_unit = s_unit  
    totpr = totpr + price  
    display item - totpr  
    scroll  
    if (scline > 10) then \  
        message "All lines full ...": \  
        goto I70
```

goto I45

```

I60 \
    error "No such part in stock file ... "
    prompt "Do you want enter this in the stock file ... " no = I55

... Run the stock program to allow entry of a new item

    exec "sage stock"

... Return to the order program , redraw the screen and display
... the valid data. Return to the prompt line for next action

    newform
    clear item - price
    gosub DISPLAY: goto I99

I65 \
    if (scrline = 1) then goto I20
    scroll -1
    goto I45

I70 \
    if (item = 1) and (o_qty = 0) and (o_part = "") then \
        message "Entry aborted ... ": \
        goto I99
    prompt "Post this order ... " no = I10
    insert order
    goto I10

I99 \
    gosub UNLOCK
    end

```

.....

(cont.)

*f = find

```
func = "FIND": display func  
message ""  
input o_ordno bs = F99  
find order nsr = F80  
gosub DISPLAY  
goto F99
```

F80 \

```
error "No order found ... "  
goto F99
```

F99 \

```
gosub UNLOCK  
end
```

*n = next

```
func = "NEXT": display func  
message ""  
next order nsr = N80  
gosub DISPLAY  
goto N99
```

N80 \

```
error "No more orders on file ..."  
goto N99
```

N99 \

```
gosub UNLOCK  
end
```

```

*a = amend
    func = "AMEND": display func
    message ""
    if (o_ordno = "") then \
        error "Use FIND or NEXT to select order... ": \
        goto A99
    read order
A10 \
    input o_ordno bs = A99
A15 \
    input o_date bs = A10
A20 \
    input o_cust bs = A15
    if (o_cust = "") then \
        error "Must enter customer ... " : \
        goto A20
A30 \
    input o_add1 - o_pcode bs = A20
A40 \
    scroll 1: totpr = 0
A45 \
    item = scrline: display item

... Keep the original quantity to allow for stock file amendment

    oqty = o_qty
A50 \
    input o_qty bs = A70
    if (o_qty = 0) and (o_part = "") then goto A75
A55 \
    input o_part bs = A50
    if (o_part = "") then \
        error "Must enter part no. ... " : \
        goto A55
    read stock key = o_part nsr = A60

```

(cont.)

```

... If the quantity is different then check the stock file
... and amend if possible
    o_desc = s_desc: o_uprice = s_uprice
    price = (o_uprice * o_qty): o_unit = s_unit
    totpr = totpr + price
    display item - totpr
    scroll
    if (scline > 10) then \
        message "All lines full ...": \
        goto A75
    goto A45
A60 \
    error "No such part in stock file ..."
    prompt "Do you want enter this in the stock file ... " no = A55

... Run the stock program to allow entry of a new item
    exec "sage stock"

... Return to the order program , redraw the screen and display
... the valid data. Return to the prompt line for next action
    newform
    clear item - price
    gosub DISPLAY: goto A99

A70 \
    if (scline = 1) then goto A20
    scroll -1
    goto A45

A75 \
    if (item = 1) and (o_qty = 0) and (o_part = "") then \
        message "Entry aborted ... " : \
        goto A99
    prompt "Post this order ... " no = A10
    write order
    goto A99
A99 \
    gosub UNLOCK
    end

```

*e = exit

exit

----- SUBROUTINES -----

. ASTOCK - Amend quantity in stock file
.

ASTOCK \

oqty = (oqty - o_qty)

... oqty now holds the variation between the original order
... and the amended order

if (s_qty + oqty >= 0) then \
 s_qty = s_qty + oqty: \
 write stock : \
 goto AS99

qty = s_qty
error "Only " + qty + " in stock"
o_qty = o_qty + (oqty + s_qty)
s_qty = 0
write stock

AS99 \
.

return

. CLEARSCROLL - Clear scroll area
.

CLEARSCROLL \
.

scroll 1

CS10 \
.

clear item - price
scroll
if (scrline > 10) then clear totpr: goto CS99
goto CS10

CS99 \
.

return

(cont.)

. DISPLAY - Display order record details
.

DISPLAY \
 display o_ordno - o_pcode

... display the scroll area . If there is no valid data for a line
... then clear the line

 scroll 1
 totpr = 0

DI10 \
 item = scline
 if (o_qty = 0) and (o_part = "") then \
 clear item - price : \
 goto DI20
 price = (o_qty * o_uprice)
 totpr = (totpr + price)
 display item - totpr

DI20 \
 scroll
 if (scline > 10) then goto DI99
 goto DI10

DI99 \
 return

. UNLOCK - Unlock all files in program
.

... The use of unlock commands at the end of each option helps to
... alleviate locked record problems in multi-user mode.

UNLOCK \
 unlock order
 unlock stock

UL99 \
 return

----- END OF PROGRAM -----

STOCK FILE PROGRAM: stock.f

"STOCK" FILE MAINTENANCE

!file stock stock

```
!temp date,,d4
+ date,Today's date,2,70
+ s_part,,4,40
+ s_desc,,6,35
+ s_unit,,7,35
+ s_ucp,,8,35
+ s_uprice,,9,35
+ s_qty,,10,35
+ s_min,,11,35
+ s_backord,,12,35
+ s_lead,,13,35
+ s_supp,,14,35
+ s_deldate,,15,35
```

```
    display date
```

```
END\
```

```
    end
```

```
*i = insert
```

```
    clear : display date
```

```
i1\
```

```
    message "Use BACKSPACE to finish inserting"
```

```
    input s_part bs = i4
```

```
    read stock nsr = i2
```

```
    gosub DISPLAY
```

```
    error "Already recorded"
```

```
    end
```

```
i2\
```

```
    input s_desc-s_deldate bs = i1 eoi = i3
```

```
i3\
```

```
    insert stock
```

```
    clear : display date
```

```
    goto i1
```

```
i4\
```

```
    clear : display date
```

```
    end
```

```

*f = find
    clear : display date
    input s_part bs = END
    find stock
    gosub DISPLAY
    end

*n = next
    next stock
    gosub DISPLAY
    end

*m = match
    match stock nsr = m1
    gosub DISPLAY
    end

m1\
    error "No further matching records"
    end

*a = amend
    check stock
    goto a1

a0\
    input s_part bs = END

a1\
    input s_desc-s_deldate bs = a0 eoi = a2

a2\
    prompt "All correct" no = a1
    write stock
    clear : display date
    message "Record amended"
    end

```

(cont.)


```

*d = delete
    check stock
    prompt "Are you sure" no = END
    delete stock
    clear : display date
    end

```

```

*e = exit
    exit

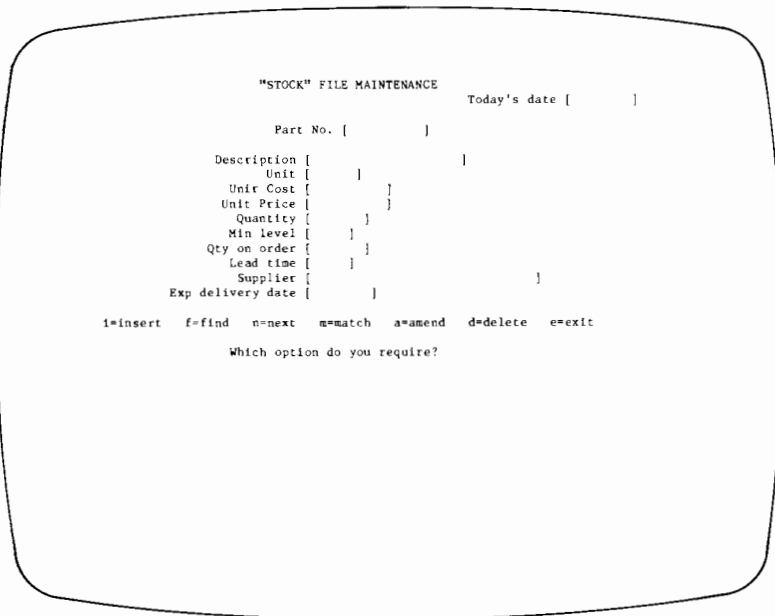
```

```

DISPLAY\
    display s_part-s_deldate
    return

```

STOCK FILE MAINTENANCE SCREEN



ORDERS REPORT PROGRAM: order.r

```
!file order
```

```
!temp totpr,Value,m4
```

```
!title gosub TITLE
```

```
!heading gosub HEAD
```

```
!final gosub TAIL
```

```
    keep 14  
    scroll 1  
    printh o_ordno,o_date,o_cust,  
    totpr=0  
    scroll 1
```

```
LOOP\  
    if scline > 10 then goto LOOP_END
```

```
    totpr = totpr + o_qty*o_uprice  
    scroll  
    goto LOOP
```

```
LOOP_END\  
    print totpr
```

```
    end
```

```
TITLE\  
    print #dw; "Demonstration" ;#sw
```

```
    return
```

```
HEAD\  
    print #dw
```

```
    print tab(28);'ORDERS LISTING'  
    print #sw  
    print tab(2);'Page: ';pageno;tab(56);' Date: ';date  
    print  
    printh *o_ordno,*o_date,*o_cust,*totpr,  
    print:print  
    return
```

TAIL\

```
print
printh [o_ordno],[o_date],[o_cust],total(totpr),
print:print tab(10);'END OF REPORT'
return
```

PRICELIST REPORT PROGRAM: prlist.r

!file stock

```
!title gosub TITLE
!heading gosub HEAD
!final gosub TAIL
```

```
keep 4
printh s_part,s_desc,s_unit,s_uprice
end
```

TITLE\

```
width 59
print #dw;"Demonstration"; #sw
return
```

HEAD\

```
print #dw
print tab(22);'STOCK PRICELIST'
print #sw
print tab(2);'Page: ';pageno;tab(44);' Date: ';date
print
printh *s_part,*s_desc,*s_unit,*s_uprice,
print:print
return
```

TAIL\

```
print:print tab(10);'END OF REPORT'
return
```

STOCK REPORT PROGRAM: stock.r

```
!file stock
```

```
!temp new_s_part,,il
```

```
!title gosub TITLE
```

```
!heading gosub HEAD
```

```
!final gosub TAIL
```

```
    keep 4
    printh s_part,
    printh s_desc,s_unit,s_ucp,s_uprice,s_qty,s_min,
    printh s_backord,s_lead,s_supp,s_deldate
    new_s_part = 0
    end
```

```
TITLE\
```

```
    width 168
```

```
    return
```

```
HEAD\
```

```
    print #dw
```

```
    print tab(74);"STOCK" FILE REPORT'
```

```
    print #sw
```

```
    print tab(2);'Page: ';pageno;tab(153);' Date: ';date
```

```
    print
```

```
    printh *s_part,*s_desc,*s_unit,*s_ucp,*s_uprice,*s_qty,
```

```
    printh *s_min,*s_backord,*s_lead,*s_supp,*s_deldate,
```

```
    print:print
```

```
    return
```

```
TAIL\
```

```
    print
```

```
    printh [s_part],[s_desc],[s_unit],[s_uprice],
```

```
    printh [s_qty],[s_min],[s_backord],[s_lead],[s_supp],
```

```
    printh [s_deldate],
```

```
    print:print tab(10);'END OF REPORT'
```

```
    return
```

SUPPLIER REPORT PROGRAM: supplier.r

```
!file stock
```

```
!temp supp,,a30
```

...The following init lines get the required supplier's
...name from the response to the menu prompt.
...This is passed via the special temp field 'arg'.

```
!init scroll 4 : supp = arg
```

```
!init if supp = "" then supp = "all suppliers"
```

```
!title gosub TITLE
```

```
!heading gosub HEAD
```

```
!exclude if supp <> "all suppliers" and supp <> s_supp
```

```
!final gosub TAIL
```

```
    keep 4
```

```
    printh s_part,s_desc,s_unit,s_ucp,s_supp
```

```
    end
```

```
TITLE\  
    print "Report for ";supp
```

```
    return
```

```
HEAD\  
    print #dw
```

```
    print tab(35);'STOCK SUPPLIERS LIST'
```

```
    print #sw
```

```
    print tab(2);'Page: ';pageno;tab(60);' Date: ';date
```

```
    print
```

```
    printh *s_part,*s_desc,*s_unit,*s_ucp,*s_supp,
```

```
    print:print
```

```
    return
```

```
TAIL\  
    print:print tab(10);'END OF REPORT'
```

```
    return
```

INDEX

	Page
A	
automatic program generators.....	3-1
B	
B-tree index.....	1-4, 2-2
box definitions.....	4-3
box delimiters.....	4-8
!box	4-5
C	
cf	1-6, 4-17
!cfile	4-5, 5-10
commands.....	4-10, 5-14
comments.....	4-4, 5-3
compiling programs.....	4-17, 5-16
!constant	5-9
cr	1-6, 5-16
cross-reference files.....	5-10
D	
database.....	1-5
data dictionaries.....	1-6, 2-4, 2-6
data storage.....	1-3
date fields.....	2-11
declarations (sage).....	4-3, 4-5
declarations (sagerep).....	5-2, 5-8
decprinter	1-7, 8-2
decvdu	1-7, 8-2
default values.....	1-9
!depth	4-5, 5-13
describe	1-6, 2-6
development tools.....	1-2
!display	5-9
driving file.....	5-3
E	
!endrec	5-9
enquiries.....	7-1
errors.....	4-16
!exclude	5-10
expressions and operators.....	4-15, 5-15

	Page
F	
field format.....	2-9, 4-7
field headings.....	2-8, 4-7
field names.....	2-7, 4-7
field type & size.....	2-8
fields.....	1-3, 2-4
file names.....	2-3
!file	4-5, 5-8
files.....	1-3, 2-2
!final	5-13
!footnote	5-12
format definitions.....	5-2, 5-7
fourth generation programming.....	1-2
G	
!gap	5-13
H	
!heading	5-12
I	
index sequential files.....	2-2
indexes.....	1-4
!init	5-8
!input	5-9
K	
key fields.....	1-4, 2-2, 2-4, 2-5
keyed file utilities.....	1-7, 8-1
keyed files.....	1-4, 2-1
kfcheck	1-7, 8-1
kfcopy	1-7, 8-1
kfdet	1-7, 8-1
kfri	1-7, 8-1
L	
language configuration.....	1-7, 8-2
lcf	1-7, 8-2
M	
menu	1-6, 6-1
money fields.....	2-11
N	
newkf	1-6, 2-13

	Page
O	
!on ending	5-12
!on starting	5-12
P	
print commands.....	5-14
printer parameter files.....	8-2
program options.....	4-3, 4-9
program statements.....	5-3
program structure (sage).....	4-2
program structure (sagerep).....	5-2
programming standards.....	1-9
programming techniques.....	4-17, 5-16
Q	
query	1-6, 7-1
R	
!read	5-9
!record	4-5, 5-10
records.....	1-3
reformat	1-7, 8-2
report programs.....	3-4, 5-1 & ff
rg	1-7, 3-4
running programs.....	4-18, 5-17
S	
sage	1-6, 4-1
sageform	1-7, 8-3
sagerep	1-6, 5-1
sagerep processing cycle.....	5-5
screen form design.....	4-7
screen form programs.....	3-2, 4-1 & ff
!scroll	4-6
SCULPTOR programs.....	1-6
selecting records.....	5-11
!select	5-11
setprinter	1-7, 8-2
setvdu	1-7, 8-2
sg	1-7, 3-2
!startrec	5-9

	Page
T	
temporary fields.....	4-6, 5-8
!temp	4-6, 5-8
terminals.....	8-2
text editor.....	4-1, 5-1
!title	5-10
trap codes.....	4-16
trapping errors.....	4-16
U	
utility programs.....	1-7, 8-1
V	
validation.....	2-10, 4-18
VDU parameter files.....	8-2
W	
!width	4-5, 5-13
X	
!xfile	5-10