

## Section VI

### A working accounting system

The following pages are command files written by a customer using almost all of the dBASE II commands, following the instructions in this manual.

The system may be fairly elementary from an accountant's viewpoint, but includes some programming techniques that you might find useful.

Besides illustrating individual commands, it shows you how to set up menus, how to find data and merge files, different ways of setting up the inputs to the system, and a number of interesting solutions to some of the problems involved in keeping a cash journal, doing payrolls, and managing databases so that information is available and data integrity is not compromised.

The programs are self-documented, with comments sprinkled liberally throughout.

## CONTENTS

	Page
STRUCTURES.....	117
REPORT FILES.....	122
SYSTEM CONSTANTS.....	124
ACCOUNTS.....	125
COSTMENU.....	127
USETAX.....	130
COSTBILLS.....	133
COSTTIME.....	135
COSTUPDATE.....	138
PAYMENU.....	139
PAYBILLS.....	140
PAYFIND.....	144
PAYEMPS.....	146
PAYROLL.....	148
DEPMENU.....	155
DEPOSITS.....	156
DEPPRINT.....	158
DEPTRANS.....	159
IOMENU.....	160
IOPOST.....	162
IOREVIEW.....	165
INVMENU.....	167
INVOICES.....	169
REPORTMENU.....	172
JOB COSTS.....	174
JOBSINDX.....	178
FINDBILLS.....	179
REVIEW.....	182
REVDHR.....	188
REVMRGN.....	189
SALESTAX.....	190
TIMECALC.....	193
PRINTOUT.....	195
GETDATE.....	196
DATETEST.....	197
NAMETEST.....	198
CHECKSTUB.....	199

## STRUCTURE FOR FILE: B:COSTBASE.DBF

FLD	NAME	TYPE	WIDTH	DEC
001	CHECK:DATE	C	007	
002	CHECK:NMBR	C	005	
003	CLIENT	C	003	
004	JOB:NMBR	N	003	
005	AMOUNT	N	009	002
006	-	C	001	
007	NAME	C	020	
008	-	C	001	
009	DESCRIP	C	020	
010	-	C	001	
011	BILL:DATE	C	007	
012	BILL:NMBR	C	007	
013	HOURS	N	006	002
014	EMP:NMBR	N	003	
** TOTAL **			00094	

(Indexed on NAME to B:\$SUPP.NDX)

## STRUCTURE FOR FILE: B:POSTFILE.DBF

FLD	NAME	TYPE	WIDTH	DEC
001	CHECK:DATE	C	007	
002	CHECK:NMBR	C	005	
003	CLIENT	C	003	
004	JOB:NMBR	N	003	
005	-	C	001	
006	NAME	C	020	
007	-	C	001	
008	DESCRIP	C	020	
009	AMOUNT	N	009	002
010	-	C	003	
011	BILL:DATE	C	007	
012	BILL:NMBR	C	007	
013	HOURS	N	006	002
014	EMP:NMBR	N	003	
** TOTAL **			00096	

## STRUCTURE FOR FILE: B:BILLINGS.DBF

FLD	NAME	TYPE	WIDTH	DEC
001	INV:NMBR	C	006	
002	CLIENT	C	003	
003	JOB:NMBR	N	003	
004	-	C	001	
005	INV:DATE	C	006	
006	TAXABLE	N	009	002
007	SALES:TAX	N	009	002
008	TAXFREE	N	009	002
009	-	C	001	
010	PO:NMBR	C	008	
011	DESCRIP	C	027	
012	MORE	L	001	

\*\* TOTAL \*\* 00084

(Indexed on INV:NMBR to B:BILLINGS.NDX)

## STRUCTURE FOR FILE: B:INVOICES.DBF

FLD	NAME	TYPE	WIDTH	DEC
001	INV:NMBR	C	006	
002	CLIENT	C	004	
003	INV:DATE	C	007	
004	TAXABLE	N	009	002
005	SALES:TAX	N	009	002
006	TAXFREE	N	009	002
007	-	C	001	
008	AMOUNT	N	009	002
009	AMT:RCD	N	009	002
010	-	C	001	
011	DATE:RCD	C	007	

\*\* TOTAL \*\* 00072

(Indexed on INV:NMBR to B:INVOICES.NDX)

## STRUCTURE FOR FILE: B:DEPOSITS.DBF

FLD	NAME	TYPE	WIDTH	DEC
001	DEP:DATE	C	007	
002	PAYER	C	020	
003	-	C	001	
004	PAY:NMBR	C	007	
005	DEPOSIT	N	009	002
006	-	C	001	
007	INV:NMBR	C	006	
008	COMMENTS	C	021	

\*\* TOTAL \*\* 00073

## STRUCTURE FOR FILE: B:CHECKFIL.DBF

FLD	NAME	TYPE	WIDTH	DEC
001	CHECK:DATE	C	007	
002	CHECK:NMBR	C	005	
003	AMOUNT	N	009	002
004	-	C	001	
005	BILL:NMBR	C	007	
006	NAME	C	020	
007	EMP:NMBR	N	003	
008	-	C	001	
009	CLIENT	C	003	
010	JOB:NMBR	N	003	
011	DESCRIP	C	020	
012	BALANCE	N	009	002
** TOTAL **			00089	

## STRUCTURE FOR FILE: B:INSERTS.DBF

FLD	NAME	TYPE	WIDTH	DEC
001	IO:NMBR	C	005	
002	MAGAZINE	C	014	
003	-	C	001	
004	ISSUE	C	006	
005	-	C	001	
006	CLIENT	C	003	
007	JOB:NMBR	N	003	
008	-	C	001	
009	AD	C	015	
010	SPACE	C	013	
011	GROSS:COST	N	009	002
012	NET:COST	N	009	002
013	-	C	001	
014	TIMES	C	003	
015	IO:DATE	C	006	
** TOTAL **			00091	

(Indexed on IO:NMBR to B:INSERTS.NDX)

## STRUCTURE FOR FILE: B:HOLD81.DBF

FLD	NAME	TYPE	WIDTH	DEC
001	CHECK:DATE	C	004	
002	-	C	001	
003	MARKER	C	001	
004	PAYROLL	N	009	002
005	FICA	N	008	002
006	FICASAL	N	009	002
007	FIT	N	009	002
008	SDI	N	007	002
009	SDISAL	N	009	002
010	SIT	N	009	002
011	UISAL	N	009	002
** TOTAL **			00076	

## STRUCTURE FOR FILE: B:PERSONNE.DBF

FLD	NAME	TYPE	WIDTH	DEC
001	EMP:NMBR	N	003	
002	NAME	C	020	
003	ADDRESS	C	024	
004	CITY:STATE	C	020	
005	ZIP	C	005	
006	PH:NMBR	C	013	
007	SS:NMBR	C	009	
008	M:S:H	C	001	
009	DEDUCTS	N	002	
010	PAY:RATE	N	007	002
011	FICA	N	008	002
012	YTDFICA	N	008	002
013	FIT	N	009	002
014	YTDFIT	N	009	002
015	SDI	N	007	002
016	YTDSDI	N	007	002
017	SIT	N	009	002
018	YTDSIT	N	009	002
019	NET:PAY	N	009	002
020	QTDSAL	N	009	002
021	YTDSAL	N	009	002
022	PAID	L	001	
023	START:DATE	C	006	
024	RATIO	N	005	003
** TOTAL **			00209	

## STRUCTURE FOR FILE: B:SUPPLIER.DBF

FLD	NAME	TYPE	WIDTH	DEC
001	SUPPLIER	C	030	
002	ADDRESS	C	024	
003	CITY	C	016	
004	STATE	C	002	
005	ZIP	C	005	
006	PHONE:NMBR	C	008	
007	AREA:CODE	C	003	
** TOTAL **			00089	

(Indexed on SUPPLIER to B:SUPPLIER.NDX)

The agency accounting system uses three standard report forms. The first one is for media and is filled out completely. The remaining two are skeletons, showing only the answers to the questions asked by dBASE II.)

ENTER REPORT FORM NAME: Media  
 ENTER OPTIONS, M=LEFT MARGIN, L=LINES/PAGE, W=PAGE WIDTH  
 M=50 <enter>  
 PAGE HEADING? (Y/N) n  
 DOUBLE SPACE REPORT? (Y/N) n  
 ARE TOTALS REQUIRED? (Y/N) y  
 SUBTOTALS IN REPORT? (Y/N) n  
 COL WIDTH,CONTENTS  
 001 6,IO:NMBR  
 ENTER HEADING: IO #  
 002 15,MAGAZINE  
 ENTER HEADING: MAGAZINE  
 003 7,ISSUE  
 ENTER HEADING: ISSUE  
 004 6,CLIENT+STR(JOB:NMBR,3)  
 ENTER HEADING: JOB #  
 005 15,AD  
 ENTER HEADING: AD  
 006 9,GROSS:COST  
 ENTER HEADING: \$GROSS  
 ARE TOTALS REQUIRED? (Y/N) Y  
 007 <enter>

The above dialog generate this report form:

PAGE NO. 00001

IO #	MAGAZINE	ISSUE	JOB #	AD	\$GROSS
2787	EDN	JAN 7	SPI678	FLAT MAN	3225.00
2788	MICROWAVES	JAN	POM772	FISHERMAN GUNNS	2500.00
2789	MICROWAVES	MAR	POM639	COP: GUNNS	2500.00
2790	ELECTRONICS	JAN	PSS754	NICE LITTLE BK	5900.00
2791	BYTE	FEB	SFT789	BILGE PUMP	2932.00

...(etc.)...



The other two report forms are:

JOB COSTS.FRM

n  
n  
y  
n  
9,BILL:DATE  
DATE  
22,NAME  
SUPPLIER.....  
17,DESCRIP  
DESCRIPTION.....  
12,AMOUNT  
AMOUNT  
Y

BILLED.FRM

n  
n  
Y  
n  
8,INV:DATE  
DATE  
8,INV:NMBR  
INVOICE  
17,DESCRIP  
DESCRIPTION  
10,TAXABLE  
TAXABLE  
Y  
10,SALES:TAX  
SALES TAX  
Y  
10,TAXFREE  
TAX-FREE  
Y

## SYSTEM CONSTANTS

System constants are kept in a file called B:Constant.MEM.

These are called out, used and updated where appropriate from within a number of programs within this accounting system. Constants are kept in a single file so that when any of them change (new tax rates, new year, etc.), they need only be changed in one location to update the entire system.

NEXTCHECK	(C)	3565	Keep checkbook current in PAYBILLS, PAYROLL and DEPOSITS command files.
MBALANCE	(N)	23921.18	
THISYEAR	(N)	81	Used by GETDATE, DATECHECK and PAYROLL files.
MINYEAR	(N)	79	
NEXT:IO	(C)	2885	Next insertion order number (IOPOST.CMD).
NEXT:INV	(C)	10623	Next invoice number (INVOICES.CMD).
FICACUT	(N)	0.0665	Entire grouping is used in the PAYROLL file. Easy to update every year because the values are not sprinkled throughout the programs.
FICAEND	(N)	29700.00	
MAXFICA	(N)	1975.05	
SDICUT	(N)	0.006	
SDIEND	(N)	14900.00	
MAXSDI	(N)	89.40	
UIEND	(N)	6000	
COMPLETED	(C)		
PREVDATE	(C)	810814	Highest employee number. Also used in POSTTIME and TIMECALC programs.
MAXEMPL	(N)	14	

\*\*\*\*\* ACCOUNTS COMMAND FILE \*\*\*\*\*  
 \* THIS IS THE CONTROL MODULE FOR ALL THE PROCEDURES USED IN THE ACCOUNTING  
 \* FUNCTIONS. (MENU DRIVEN). The operator is given a choice of major  
 \* functions. The menu selection here calls up menus of sub-functions  
 \* to as many levels as necessary. A package of utility functions is  
 \* also provided as part of the overall system for file maintenance, etc.  
 \*\*\*\*\*

SELECT PRIMARY  
 CLEAR  
 SET TALK OFF  
 SET EJECT OFF  
 SET MARGIN TO 38  
 SET RAW ON

\*\*\*\*\*  
 \* This group of commands is hardware-dependent, may be eliminated  
 \* down to the next row of asterisks.  
 SET PRINT ON  
 \* Sets (8-1)≅7 lines/inch and 89 lines page, Diablo 1650/sheet feeder  
 ?? CHR(27)+CHR(30)+CHR(8)  
 ?? CHR(27)+CHR(12)+CHR(89)  
 SET PRINT OFF  
 \* End printer setup commands.  
 \*\*\*\*\*

STORE T TO Accounting  
 DO WHILE Accounting

ERASE

?

?

? '1> ENTER BILLS & TIME SHEETS

? '

? '2> PAY BILLS & SALARIES

? '

? '3> DEPOSITS & CHECKBOOK

? '

? '4> MEDIA INSERTION ORDERS'

? '

? '5> CLIENT BILLINGS & INVOICES'

? '

? '

? ' Pick a number (Q to QUIT)'

WAIT TO Action

IF ! (Action) = 'Q'

\* This sequence homes the cursor and erases the screen on the IBM 3101  
 STORE chr(27) + chr(76) + chr(27) + chr(89) + chr(0) TO q  
 QUIT TO '&q'

ELSE

IF Action = '1'

DO CostMenu

ELSE

IF Action = '2'

DO PayMenu

ELSE

IF Action = '3'

DO DepMenu

ELSE

IF Action = '4'

DO IOMenu

ELSE

IF Action = '5'

DO InvMenu

```
ELSE
  IF Action = '6'
    DO ReportMe
  ELSE
    If Action = '7'
      ERASE
      @ 5,10 SAY '      HELP file not ready yet.'
      ? ' <Return> to continue.'
      WAIT
    ENDIF 7
  ENDIF 6
ENDIF 5
ENDIF 4
ENDIF 3
ENDIF 2
ENDIF 1
ENDIF Q
STORE T TO Accounting
ENDDO Accounting
```

\*\*\*\*\* COSTMENU COMMAND FILE. \*\*\*\*\*

- \* This is one level down from the Accounts.Cmd control module.
  - \* Selections are refinements that relate to costs for client-related jobs or agency overhead.
  - \* The main database is called CostBase.Dbf and is kept on disk B.
  - \* Costs are not entered directly into the CostBase, however, because this leads to data contamination and all sorts of problems fixing the errors. Instead, supplier bills and agency time sheets are posted into an interim file called PostFile.Dbf. In here, they can be reviewed and edited as necessary.
  - \* When all the cost entries are confirmed as being correct, they are transferred to the CostBase by using the update procedure (selection 5).
- \*\*\*\*\*

STORE T TO Posting  
DO WHILE Posting

```
ERASE
@ 2,20 SAY ' 1> UNTAXED ITEMS USED BY AGENCY'
@ 4,20 SAY ' 2> ENTER SUPPLIER BILLS'
@ 6,20 SAY ' 3> ENTER EMPLOYEE TIME SHEETS'
@ 8,20 SAY ' 4> EDIT the POSTFILE'
@ 10,20 SAY ' 5> REVIEW/PRINT the POSTFILE'
@ 12,20 SAY ' 6> UPDATE THE COSTBASE'
@ 14,20 SAY ' 7> WIPE OUT DELETED RECORDS IN POSTFILE'
@ 16,20 SAY ' <RETURN>'
WAIT TO Action
ERASE
```

IF Action = '1'

```
ERASE
@ 4,10 SAY 'This program accepts bills for items that the agency bought,
? ' without paying sales tax, but will use internally, rather'
? ' than for a job that will be billed to a client. This would'
? ' include equipment bought out of state and locally bought'
? ' materials NOT used in client jobs and NOT taxed.'
?
? ' DO NOT ENTER ANY OTHER BILLS!'
?
? 'Do you want to continue (Y or N)?'
WAIT TO GoAhead
IF !(GoAhead) = 'Y'
DO UseTax
ELSE
RELEASE ALL
ENDIF
```

ELSE

```
IF Action = '2'
ERASE
@ 4,10 SAY ' CHECK ALL THE BILLS BEFORE ENTERING THEM.'
? ' If any of the bills are for items used by the agency'
? ' but sales tax was not paid, select OPTION 1 from the'
? ' entry menu. <Return> to continue.'
?
WAIT
DC CostBills
ELSE
IF Action = '3'
DO CostTime
ELSE
IF Action = '4'
STORE "Y" TO Changing
DO WHILE !(Changing)='Y'
USE B:PostFile
IF EOF
? 'There are no entries in the POSTING file.'

```

```

? '<Return> to continue.'
WAIT
STORE "N" TO Changing
ELSE
GO BOTTOM
ERASE
@ 3,10 SAY 'EDITING BILLS ENTERED.'
@ 5,10 SAY 'There are '+STR(%,5)+' file entries.'
@ 6,10 SAY 'Which entry do you want to EDIT?'
ACCEPT TO Number
IF VAL(Number) <= 0 .OR. VAL(Number) > #
?
? 'Out of range: do you want to continue (Y or N)?'
WAIT TO Changing
ELSE
Edit &Number
REPLACE Name WITH !(Name), Descrip WITH !(Descrip),;
Client WITH !(Client), Bill:Nmbr WITH !(Bill:Nmbr)
?
? 'Do you want to edit any other entries (Y or N)?'
WAIT TO Changing
ENDIF
ENDIF
ENDDO Changing
RELEASE All
ELSE
IF Action = '5'
STORE 'Y' TO Reviewing
DO WHILE !(Reviewing)='Y'
USE B:Postfile
COUNT FOR .NOT. * TO Any
IF Any = 0
? 'No unposted entries in the POSTING file.'
? '<Return> to continue.'
WAIT
STORE "N" TO Reviewing
ELSE
ERASE
? 'There are '-STR(Any,5)+' unposted entries.'
? 'Do you want to print them, too (Y or N)?'
WAIT TO Output

IF !(Output)='Y'
SET PRINT ON
ENDIF
? '
+ 'TION          JOB      NAME          DESCRIP';
?          AMOUNT      DATE      NUMBER'
STORE 'OFF' TO Condition
STORE '0' TO Number
DO Printout

? "That's all the unposted entries."
? 'Want to see them again (Y or N)?'
? '(To see deleted records, choose "Edit".)'
WAIT TO Reviewing
ENDIF
ENDDO Reviewing
RELEASE all
ELSE
IF Action = '6'
DO CostUpdate
ELSE

```

```
IF Action = '7'  
  ? 'This destroys all records in the PostFile.'  
  ? 'Do you want to do this (Y or N)?'  
  WAIT TO WipeOut  
  IF !(WipeOut) = 'Y'  
    USE B:PostFile  
    PACK  
  ENDIF  
  RELEASE ALL  
ELSE  
  RELEASE ALL  
  RETURN  
ENDIF 7  
ENDIF 6  
ENDIF 5  
ENDIF 4  
ENDIF 3  
ENDIF 2  
ENDIF 1  
STORE T TO Posting  
ENDDO Posting
```

## \*\*\*\*\* USETAX COMMAND FILE \*\*\*\*\*

```

* This file accepts inputs for supplier bills when the agency has bought
* an item without paying a use tax on it.
*   The item or items are added to the Invoices file (not Billings),
* then are used by the SalesTax program so that the Quarterly Sales Tax
* report can be prepared by the computer.
*
* A temporary file called GetBills is used for data entry because the operator
* can decide to quit on an incomplete entry, which is marked for deletion.
* When the data is APPENDED to the PostFile, these entries are eliminated (the
* APPEND command does not transfer records marked for deletion). An entry must
* include at least the name of a supplier and the amount of the bill. If these
* are not both supplied, the entry is flagged for correction or deletion.
*****

```

ERASE

@ 5,20 SAY 'AGENCY USE TAX PROCEDURE'

?

USE B:PostFile

COPY STRUCTURE TO GetBills

USE GetBills

STORE 'Y' TO Bills

DO WHILE !(Bills) &lt;&gt;

APPEND BLANK

STORE STR(5) TO Number

REPLACE Client WITH 'QFC'

STORE T TO Entering

DO WHILE Entering

ERASE

@ 1,0 SAY 'ENTER ONLY UNTAXED ITEMS NOT USED FOR CLIENT JOBS.'

@ 3,0 SAY 'RECORD NUMBER:' + Number

@ 4,0 SAY 'CLIENT:' + Client + ':'

@ 5,0 SAY 'JOB NUMBER' GET Job:Nmbr

@ 6,0 SAY 'AMOUNT' GET Amount

@ 7,0 SAY 'BILL NUMBER' GET Bill:Nmbr

@ 8,0 SAY 'BILL DATE' GET Bill:Date

@ 9,0 SAY 'SUPPLIER NAME' GET Name

READ

REPLACE Name WITH !(Name), Descrip WITH 'USE TAX ENTRY';

Bill:Nmbr WITH !(Bill:Nmbr)

@ 7,17 SAY Bill:Nmbr

@ 9,17 SAY Name

@ 10,17 SAY Descrip

STORE ' ' TO Getting

IF Job:Nmbr &lt;= 0 .OR. Job:Nmbr &gt; 99

@ 12,0

? ' The JOB NUMBER entry is wrong.'

? ' Agency jobs are from 1 through 99.'

? ' F if FINISHED,'

ACCEPT ' &lt;Return&gt; to change.' TO Getting

ELSE

IF Amount = 0 .OR. Name &lt;= ' '

?

?

? ' AMOUNT or NAME missing.'

? ' F if FINISHED,'

ACCEPT ' &lt;Return&gt; to change.' TO Getting

ELSE

@ 12,5 SAY ' C to CHANGE.'

@ 13,5 SAY ' F if FINISHED,'

ACCEPT ' &lt;Return&gt; to continue.' TO Bills



```

        IF !(Bills)='C'
            STORE T TO Entering
        ELSE
            STORE F TO Entering
        ENDIF
    ENDIF amount or name
ENDIF client or job number

    IF !(Getting)='F'
        DELETE RECORD &Number
        STORE F TO Entering
        STORE 'F' TO Bills
    ENDIF
ENDDO Entering
ENDDO Bills

COUNT FOR .NOT. * TO Any
IF Any = 0
    ?
    ? '          No valid entries to add to the files.'
    ? ' <Return> to the menu.'
    WAIT
ELSE

    RESTORE FROM B:Constant
    STORE 'Bill:Date' TO Date
    DO DateTest

    * Checks names against a list of suppliers to catch spelling and
    * abbreviation inconsistencies.
    DO NameTest

    ERASE
    @ 3,25 SAY ' *** DO NOT INTERRUPT ***'
    @ 5,25 SAY ' UPDATING THE POSTING FILE'
    USE B:PostFile
    APPEND FROM GetBills

    * The following loop transfers the bills just entered into the Invoices
    * file. The amount of the bill is entered in the "Taxable" column. The
    * job number is entered into the Invoice Number column. Since invoice
    * have 5 digits, while job numbers are under 1000, we use this to sepa-
    * rate the two types of entries later in the SalesTax.CMD file.
    * PRIMARY and SECONDARY work areas are used to step through the GetBills
    * file one entry at a time.
    USE GetBills
    SELECT SECONDARY
    USE B:Invoices
    SELECT PRIMARY
    DO WHILE .NOT. EOF
        IF *
            SKIP
        ELSE
            SELECT SECONDARY
            APPEND BLANK
            REPLACE Inv:Nmbr WITH STR(Job:Nmbr,3), Inv:Date WITH Bill:Date,;
                Taxable WITH P.Amount, Date:Rcd WITH 'USE TAX'
            SELECT PRIMARY
            SKIP
        ENDIF
    ENDDO
ENDIF

```

ENDIF

USE  
DELETE FILE GetBills  
RELEASE All  
RETURN

## \*\*\*\*\* COSTBILLS COMMAND FILE \*\*\*\*\*

\* This file accepts inputs for supplier bills.  
 \* A temporary file called GetBills is used for data entry because the operator  
 \* can decide to quit on an incomplete entry, which is marked for deletion.  
 \* When the data is APPENDED to the PostFile, these entries are eliminated (the  
 \* APPEND command does not transfer records marked for deletion). An entry must  
 \* include at least the name of a supplier and the amount of the bill. If these  
 \* are not both supplied, the entry is flagged for correction or deletion.  
 \*\*\*\*\*

ERASE

@ 5,20 SAY 'SUPPLIER BILLS'

USE B:PostFile

COPY STRUCTURE TO GetBills

USE GetBills

STORE 'Y' TO Bills

DO WHILE !(Bills) &lt;&gt; 'F'

APPEND BLANK

STORE STR(5) TO Number

STORE T TO Entering

DO WHILE Entering

ERASE

@ 1,0 SAY ' RECORD NUMBER: '-Number

@ 3,0 SAY ' CLIENT' GET Client

@ 4,0 SAY ' JOB NUMBER' GET Job:Nmbr

@ 5,0 SAY ' AMOUNT' GET Amount

@ 6,0 SAY ' BILL NUMBER' GET Bill:Nmbr

@ 7,0 SAY ' BILL DATE' GET Bill:Date

@ 8,0 SAY ' SUPPLIER NAME' GET Name

@ 9,0 SAY ' DESCRIPTION' GET Descrip

READ

REPLACE Client WITH !(Client),Name WITH !(Name),Descrip;  
WITH !(Descrip), Bill:Nmbr WITH !(Bill:Nmbr)

@ 3,17 SAY Client

@ 8,17 SAY Name

@ 9,17 SAY Descrip

STORE ' ' TO Getting

IF \$(Client,1,1) = ' ' .OR. \$(Client,2,1) = ' ' .OR. \$(Client,3,1) = ' '  
.OR. Job:Nmbr <= 0

@ 12,0

? ' CLIENT or JOB NUMBER wrong.'

? ' F if FINISHED,'

ACCEPT ' &lt;Return&gt; to change.' TO Getting

ELSE

IF Amount = 0 .OR. Name &lt;= ' '

?

?

? ' AMOUNT or NAME missing.'

?

? ' F if FINISHED,'

ACCEPT ' &lt;Return&gt; to change.' TO Getting

ELSE

@ 12,5 SAY ' C to CHANGE,'

@ 13,5 SAY ' F if FINISHED,'

ACCEPT ' &lt;Return&gt; to continue.' TO Bills

IF !(Bills)='C'

STORE T TO Entering

ELSE

STORE F TO Entering

ENDIF

```
        ENDIF amount or name
    ENDIF client or job number

    IF !(Getting)= 'F'
        DELETE RECORD &Number
        STORE F TO Entering
        STORE 'F' TO Bills
    ENDIF
    ENDDO Entering
    ENDDO Bills

    COUNT FOR .NOT. * TO Any
    IF Any = 0
        ? 'No entries to add to the Cost Base.'
        ? '<Return> to the menu.'
        USE
        WAIT
    ELSE

        RESTORE FROM B:Constant
        STORE 'Bill:Date' TO Date
        DO DateTest

        * Checks names against a list of suppliers to catch spelling and
        * abbreviation inconsistencies.
        DO NameTest

        ERASE
        @ 3,25 SAY ' *** DO NOT INTERRUPT ***'
        @ 5,25 SAY ' UPDATING THE POSTING FILE'
        USE B:PostFile
        APPEND FROM GetBills
    ENDIF

    USE
    DELETE FILE GetBills
    RELEASE All
    RETURN
```

## \*\*\*\*\* COSTTIME COMMAND FILE \*\*\*\*\*

\* Accepts time sheet entries for employees using a temporary  
 \* file called GetTime. For data entry.  
 \* GetTime is used because the operator can decide to quit on an incomplete  
 \* entry. In that case, the entry is marked for deletion, and when the data is  
 \* APPENDED to the PostFile, these entries are eliminated (the APPEND command  
 \* does not transfer records marked for deletion).  
 \* After all entries are made, entries are checked for the  
 \* correct range of employee numbers and to see that hours have  
 \* been entered. Using GetTime, we can check the entries without  
 \* having to go through the entire PostFile.  
 \* After verifying that the dates are in the right format and  
 \* checking the names against our Suppliers file, the billing amounts  
 \* are computed.  
 \* The records are then transferred to the CostFile and the  
 \* temporary file GetTime is deleted.  
 \*\*\*\*\*

@ 0,25 SAY ' TIME SHEETS '

RESTORE FROM B:Constant  
 USE B:PostFile  
 COPY STRUCTURE TO GetTime

USE GetTime  
 STORE 'Y' TO Time  
 DO WHILE !(Time) <>'F'  
 APPEND BLANK  
 STORE STR(%,5) TO Number

STORE T TO Entering  
 DO WHILE Entering  
 ERASE

STORE F TO Entering  
 @ 1,0 SAY ' RECORD NUMBER: '-Number  
 @ 3,0 SAY ' DATE WORKED' GET Bill:Date  
 @ 4,0 SAY ' CLIENT' GET Client  
 @ 5,0 SAY ' JOB NUMBER' GET Job:Nmbr  
 @ 6,0 SAY ' HOURS WORKED' GET Hours  
 @ 7,0 SAY ' EMPLOYEE NUMBER' GET Emp:Nmbr  
 @ 8,0 SAY ' EMPLOYEE NAME' GET Name  
 READ

REPLACE Check:Nmbr WITH '----', Check:Date WITH Bill:Date,  
 Client WITH !(Client), Name WITH !(Name)  
 @ 4,17 SAY Client  
 @ 8,17 SAY Name

\* The following sequence of IF statements flags all entry errors, then  
 \* gives the operator the choice of fixing them or ending the procedurs.

?  
 IF \$(Client,1,1) = ' ' .OR. \$(Client,2,1) = ' ' .OR. \$(Client,3,1) = ' '  
 ? ' CLIENT must have three letters.'  
 STORE T TO Entering  
 ENDIF

IF Job:Nmbr < 100  
 ? ' JOB # is not for a client job.'  
 ? ' Is this right (Y or N)?'  
 WAIT TO Ask  
 IF !(Ask) <> 'Y'  
 STORE T TO Entering  
 ENDIF

ENDIF

```
IF .NOT. (Hours > 0)
  ? '    HOURS must be entered.
  STORE T TO Entering
ENDIF
```

```
IF .NOT.(Emp:Nmbr>0 .AND. Emp:Nmbr<=MaxEmpl)
  ? '    EMPLOYEE # out of range.
  STORE T TO Entering
ENDIF
```

```
IF $(Name,1,1) = ' '
  ? '    NAME must not start with a blank.'
  STORE T TO Entering
ENDIF
```

```
IF Entering
  ?
  ?
  ? '    F if FINISHED,'
  ACCEPT '    <Return> to change' TO Time

  * If the operator decides to quit on an incomplete entry, it is
  * marked for deletion so that it is not transferred to the PostFile.
  IF I(Time) = 'F'
    DELETE RECORD &Number
    STORE F TO Entering
  ENDIF
ELSE
  ?
  ? '    C to CHANGE,'
  ? '    F if FINISHED,'
  ACCEPT '    <Return> to continue' TO Time
  IF I(Time) = 'C'
    STORE T TO Entering
  ENDIF
ENDIF
ENDDO Entering
ENDDO Time
```

COUNT FOR .NOT. \* TO Any

```
IF Any = 0
  ERASE
  @ 3;0 SAY '    No entries to add to the CostFile. '
  ? ' <Return> to the menu. '
  USE
  WAIT
ELSE
```

```
* The test for the date needs the name of the date field to be tested.
STORE 'Bill:Date' TO Date
DO DateTest
```

```
* Checks names against a list of suppliers to catch spelling and
* abbreviation inconsistencies.
DO NameTest
```

```
* Verifies match between employee name and number, then computes the amount
* to be billed for the employee's time based on his salary.
DO TimeCalc
```

ERASE

```
@ 3,25 SAY " *** DO NOT INTERRUPT ****"  
@ 5,25 SAY " UPDATING THE POSTING FILE"  
USE B:PostFile  
APPEND FROM GetTime  
ENDIF  
  
DELETE FILE GetTime  
RELEASE All  
RETURN
```

## \*\*\*\*\* COSTUPDATE COMMAND FILE \*\*\*\*\*

```

*      Records from the COSTFILE are added to the COSTBASE.
* This step is so critical to data integrity that we: use a password
* to prevent accidental access; verify dates; check the names of suppliers;
* and compute time charges if necessary. Notice that these are done by
* simply calling the utility command files.
*      The PostFile has all its records marked for deletion after they
* have been posted (can still be recovered).
*****

```

SET TALK OFF

```

@ 4,12 SAY '*****'
@ 6,12 SAY 'MAKE CERTAIN EVERYTHING IN THE POSTFILE IS CORRECT'
@ 8,12 SAY 'BEFORE ENTERING THE CODE TO CONTINUE'
@ 10,12 SAY '*****'

```

SET CONSOLE OFF

ACCEPT TO Lock

SET CONSOLE ON

IF !(Lock) <> 'H'

```

  @ 12,12 SAY '      UNAUTHORIZED ACCESS ATTEMPTED.'
  @ 14,12 SAY 'YOU HAVE 6 SECONDS BEFORE THE EXPLOSION.'
  STORE 1 TO X
  DO WHILE X < 150
    STORE X + 1 TO X
  ENDDO
  RELEASE Lock
  RETURN

```

ELSE

```

  ERASE
  @ 5,20 SAY 'Checking bills in the POSTING File:'
  USE B:PostFile
  COUNT FOR .NOT. * TO None
  IF None = 0
    @ 6,20 SAY 'No new entries in the POSTING file.'
    @ 7,20 SAY '<Return> to continue.'
    WAIT
  ELSE
    GO TOP
    RESTORE FROM B:Constant
    STORE 'Bill:Date' TO Date
    DO DateTest
    DO NameTest
    DO TimeCalc
    ERASE
    @ 5,20 SAY ' *** DO NOT INTERRUPT ***'
    @ 6,20 SAY 'Posting COSTS to the Costbase.'
    * Save the number of the last record in Costbase
    USE B:CostBase
    GO BOTTOM
    STORE # TO LastReco
  
```

```

  USE B:Costbase INDEX B:$Supp
  APPEND FROM B:PostFile

```

```

  USE B:PostFile
  DELETE ALL

```

ENDIF

ENDIF

RELEASE All

RETURN



\*\*\*\*\* PAYMENU COMMAND FILE \*\*\*\*\*

\* This is a sub-module of the Accounts.CMD file and provides choices  
 \* as to which checks are to be prepared for posting and printing.  
 \* Paying salaries has another menu level to allow partial payments  
 \* to selected employees (e.g., leave of absence, when an employee does not  
 \* work a full two week stretch, etc.)  
 \* The checkbook balance and next check number must be confirmed before  
 \* either of the procedures can be performed.  
 \*\*\*\*\*

RESTORE FROM B:Constant

ERASE

@ 3, 0 SAY 'CHECK NUMBER: '+NextCheck+'  
 ?  
 ? ' Do these match the checkbook?'  
 ? ' C to CONTINUE,'  
 ? ' <Return> to change.'  
 ?

BALANCE: '+STR(MBalance,9,2)

WAIT TO Continue

IF !(Continue) <> 'C'

RELEASE All

RETURN

ENDIF

STORE T TO Paying

DO WHILE Paying

ERASE

@ 5,20 SAY ' 1> PAY BILLS

@ 7,20 SAY ' 2> PAY SALARIES

@ 10,20 SAY ' <RETURN>'

WAIT TO Action

IF Action = '1'

USE B:PostFile

\* Can abort if any entries in the Postfile.

COUNT FOR .NOT. \* TO Any

IF Any = 0

DO PayBills

ELSE

?

? 'The POSTING file has '-STR(Any,5)+' bills in it.'

? 'Do you still want to pay bills now (Y or N)?'

WAIT TO Continue

IF !(Continue) = 'Y'

DO PayBills

ELSE

RELEASE All

ENDIF

ENDIF

ELSE

IF Action = '2'

DO PayEmps

ELSE

RELEASE All

RETURN

ENDIF 2

ENDIF 1

STORE T TO Paying

ENDDO Paying

## \*\*\*\*\* PAYBILLS COMMAND FILE \*\*\*\*\*

```

* Before this procedure can be accessed, the check number and balance must
* be verified in the PAYMENU command file.
* This is one of the longer files, but the individual portions of it are
* not too complicated. Repetitive procedures in the main loop (controlled
* by the variable "Finished") could have been put in separate command files
* to make this file easier to understand and maintain, but this way it
* minimizes disk accesses and increases speed.
* This file finds bills to be paid in the CostBase, generates the next
* check number, writes a check in the CheckFil and maintains the checkbook
* balance.
* The next check number and checkbook balance are recalled from a file
* called Constant.MEM. The final values for both of these are stored in the
* same file after all the bills have been paid.
* The date is entered once at the start of the procedure, then
* is automatically inserted into each entry. The date is checked to
* see that it is in the YYMMDD format, and that the values are within
* possible limits (month from 1 to 12, day from 1 to 31, year=ThisYear).
* Entries must include at least the name of the party being paid.
* Balances are automatically computed and shown to the operator.
* Check numbers are automatically assigned by the computer.
* If several entries are made against a single check number (the
* operator has this option), these are added and shown as a single
* item in the printout.
*****

```

```

RESTORE FROM B:Constant
DO GetDate

```

```

SELECT PRIMARY
USE B:CostBase INDEX B:$Supp

```

```

* Initialize. "New" is used to determine whether the program should generate
* a new check number or use the old one (where several bills to a single
* supplier are being paid). "Finished" is the control variable that determines
* whether we should run through the procedure again, or are done paying bills.

```

```

STORE 'N' TO New
STORE 'N' TO Finished
DO WHILE !(Finished) <> 'F'
  STORE "C" TO Entering
  DO WHILE !(Entering) = 'C'
    ERASE
    @ 3, 0 SAY 'CHECK NUMBER: '+NextCheck+ ' BALANCE: '+STR(MBalance,9,2)
    ? CHR(7)
    @ 4,C
    ACCEPT ' MAKE CHECK TO ' TO MName
    ACCEPT ' INVOICE NUMBER ' TO MBill:Nmbr
    ACCEPT ' ENTER AMOUNT ' TO Temp
    STORE !(MName) TO MName
    STORE !(MBill:Nmbr) TO MBill:Nmbr
    STORE VAL(Temp) TO MAmount
    STORE MAmount*1.00 TO MAmount
    @ 6,19 SAY MName
    @ 7,19 SAY MBill:Nmbr
    @ 8,19 SAY MAmount
    @ 11, 0 SAY ' C to CHANGE,
    ? ' <Return> to continue.'
    WAIT TO Entering
  ENDDO Entering

IF LEN(MName) > 10
  STORE $(MName,1,10) TO Key
ELSE
  STORE MName TO Key

```

ENDIF

```

IF Key > ' '
STORE T TO Looking
@ 11,0 SAY "I'M LOOKING, I'M LOOKING!!"
@ 12,0
@ 13,0
STORE 0 TO Start
FIND &Key
IF # = 0
?
? "• GEE, I CAN'T FIND THE NAME. Please check the spelling."
? " Or maybe it hasn't been posted to the COSTBASE yet."
? '<Return> to continue.'
WAIT
ERASE
ELSE
DO PayFind
ENDIF there is an unpaid bill for the supplier

* "Start" is brought in from PayFind.CMD. If we started at the first
* entry for a name (had only the name), Start=0. If we had more than
* the name, Start contains the record number we started on. Since this
* could be in the middle of the listing, we use "Counter" so that we can
* come back to the top of the listing for the name once.
IF Start > 0
STORE 0 TO Counter
ELSE
STORE 1 TO Counter
ENDIF

STORE ' ' TO Confirm
DO WHILE !(Confirm) <> 'P' .AND. .NOT. Looking
@ 9,0
? 'RECORD      NAME                      AMOUNT BILL #';
? '  +  DATE'
?
DISPLAY '      '+Name, Amount, Bill:Nmbr, Bill:Date
?
? CHR(7)
? '          P to PAY this bill,'
? '          Q to QUIT without paying,'
? '          <Return> to continue.'
ACCEPT '          ' TO Confirm

IF !(Confirm) = 'Q'
IF !(New) = 'S'
STORE STR(VAL(NextCheck)+1,4) TO NextCheck
ENDIF
STORE ' ' TO New
STORE T TO Looking
ELSE
IF !(Confirm) = 'P'
STORE STR(#,5) TO Found
REPLACE Check:Date WITH Date, Check:Nmbr WITH NextCheck
STORE (MBalance-Amount) TO MBalance

SELECT SECONDARY
USE B:Checkfil
APPEND BLANK
REPLACE Check:Date WITH P.Check:Date, Name WITH P.Name,;
Check:Nmbr WITH P.Check:Nmbr, Balance WITH MBalance,;
Amount WITH P.Amount, Bill:Nmbr WITH P.Bill:Nmbr
SELECT PRIMARY

```

```

ERASE
@ 3, 0 SAY 'CHECK WRITTEN: '+NextCheck+;
          ' NEW BALANCE: '+STR(MBalance,9,2)
?
DISPLAY 'PAYMENT MADE: '+Check:Date, Name, Amount, Bill:Nmbr;;
          Bill:Date OFF
?
? ' S for SAME SUPPLIER (Repeats check #)'
? CHR(7)
ACCEPT ' <Return> to continue.' TO New
IF ! (New) <> 'S'
  STORE STR(VAL(NextCheck)+1,4) TO NextCheck
ELSE
  STORE ' ' TO Confirm
ENDIF
ENDIF
ENDIF

IF ! (New) = 'S' .OR. ! (Confirm) <> 'P'
* If Confirm <> 'P', we rejected the first unpaid bill that was
* shown. Rather than going back to the beginning, the loop
* below SKIPs to the next INDEXed name until we find an unpaid
* bill, or go beyond the records for the name we are paying.
* The same applies if we want to pay another bill to the
* same supplier (New='S'). Since we are in the file on the name
* we want we SKIP to the next record until we find an unpaid
* bill or run out of records for that name.
* If we had only the name and started with the first unpaid
* bill we stop now since we have looked at all the unpaid bills
* for that supplier.
* If we could have entered the list of records for this
* supplier in the middle (more than the name provided), we look
* at the unpaid bills between where we are and the end of the
* list, then go up to the first entry for that name and check
* all of the unpaid bills that we had previously skipped past.
* This is controlled by Counter.
* After the second FIND in the command file (below), we
* stop looking when the record number we are on is greater than
* or equal to the number of the record we start on (Start).

SKIP
DO WHILE Check:Nmbr <> ' ' .AND. Name=Key .AND. .NOT. EOF
  SKIP
ENDDO

* We enter this loop when we reach the end of the records with
* names that match the one we are looking for. If we started
* with the first unpaid bill, the record number is greater than
* Start (because Start=0) and Counter=1 (because we set it to
* that value). The second IF below is True and we terminate the
* search.
* If Start>0, Counter=0 the first time we run out of
* records with a matching name, so the program does the ELSE
* commands below.
* Start is still >0 and Count is now 1, so the last term in
* the first IF applies. On this second pass when we get to a
* record number >=Start, we drop into the loop and do the IF to
* terminate the search because we have now looked at all the
* unpaid bills for the name we entered.
IF EOF .OR. Name <> Key .OR. (# >= Start .AND. Start <> 0;
          .AND. Counter > 0)
  IF (# >= Start .AND. Counter > 0)
    STORE T TO Looking
    @ 4, 0
    ? chr(27)+chr(74)

```

```

? '      We have now looked at all the entries for '+ MName
? ' <Return> to continue.'
? CHR(7)
IF !(New)='S'
  STORE STR(VAL(NextCheck)+1,4) TO NextCheck
  STORE 'N' to New
ENDIF
WAIT
ELSE
  STORE Counter + 1 TO Counter
  @ 13, 0
  @ 16, 0 SAY "I'M WORKING AS FAST AS I CAN -- HANG ON! "
  FIND &Key
  DO WHILE Check:Nmbr <> ' '
    SKIP
  ENDDO
ENDIF
ENDIF
ENDIF is it the right record
ENDIF
ENDDO Confirm the record
ENDIF
IF !(New) <> 'S'
  @ 4, 0
  ? chr(27)+chr(74)
  ? '      F if FINISHED,
  ? CHR(7)
  ACCEPT '      <Return> to continue.' TO Finished
ENDIF
ENDDO Finished
RELEASE MName, MBill:Nmbr, Key, MAmount, Start, Found, Looking, New, Change,;
      Entering, Counter, Temp, Abort, Continue, Finished, Confirm, Date
SAVE TO B:Constant
USE B:Checkfil
COUNT FOR .NOT. * TO Any
ERASE
@ 3,0
IF Any=0
  ? '      No new checks in the checkfile.'
  ? '      <Return> to continue.'
  WAIT
ELSE
  ? 'There are '-STR(Any,5)+' new checks in the CheckFile.'
  ? 'Do you want to print the checkstubs now (Y-or N)?'
  ?
  WAIT TO Hardcopy
  IF !(Hardcopy) = 'Y'
    DO NameTest
    DO CheckStub
  ENDIF
ENDIF
RELEASE All
RETURN

```

\*\*\*\*\* PAYFIND COMMAND FILE \*\*\*\*\*

```

* This file is called by the PAYBILLS command file after we have found at least
* one cost entry for the supplier that we are looking for.
*   This file now looks for either the first unpaid bill for the supplier
* (if only the name was specified) or looks for a complete match (if more than
* the name was specified.
*   If an unpaid bill meeting the criteria is found, Looking is
* set to False. Other wise it remains True.
*   If only the name was used, at this point we are at the first
* unpaid bill for the supplier name.
*   If more than the name was specified for the search, we could be anywhere
* in the indexed list of records for this supplier. If we do not want to pay
* this particular bill, or we want to pay more bills for this supplier, we use
* a short cut in the PAYBILLS command file so that we do not have to start at
* the first record for the name every time. To do this, we store the record
* number that we start at to a variable called Start if we have more than the
* name to look for. Otherwise, Start =0
*****

```

STORE T TO Looking

IF Mbill:Nmbr > ' ' .OR. MAmount > 0

```

* If we have more than the name, we first check for the bill number.
* If this is not found or if the bill has already been paid,
* the confirming procedure is skipped (Looking set TRUE).
*   In this case, we may have entered the list of supplier bills in
* middle of the indexed list. In a later procedure, we may need to go
* back to the top and look at the names we skipped. To do this, if we
* find a record here, we store its number to "Start".

```

IF Mbill:Nmbr > ' '

```

DO WHILE Name=Key .AND. .NOT. EOF .AND. Looking
  IF Bill:Nmbr <> Mbill:Nmbr
    SKIP
  ELSE
    STORE F TO Looking
  ENDIF
ENDDO

```

```

* If we're on a new name or the end of the file, Looking is TRUE
* because we have not found the supplier we were looking for.
* Otherwise, we have a matching bill number to confirm.

```

IF Looking

```

? ' This BILL NUMBER is not in the costbase.'
? '<Return> to continue.'
WAIT

```

ELSE

```

IF Check:Nmbr <> ' '
  STORE T TO Looking
  ? ' This bill paid on '+Check:Date+', check '+Check:Nmbr
  ? '<Return> to continue.'
  WAIT

```

ENDIF

ENDIF

ELSE

```

* If no bill number, look for the amount and an unpaid bill.
* If not found, skip the confirmation procedure.

```

```

DO WHILE Name=Key .AND. .NOT. EOF .AND. Looking
  IF Amount <> MAmount .OR. Check:Nmbr <> ' '
    SKIP
  ELSE

```

STORE F TO Looking

ENDIF

ENDDO

```

* If we're on a new name or the end of the file, Looking is TRUE
* Otherwise, we have an unpaid bill to confirm.
IF Looking
  ? ' No unpaid bill for this amount and this supplier.
  ? '<Return> to continue.'
  WAIT
ENDIF
ENDIF

* If we found a matching record, store its number to Start
IF .NOT. Looking
  STORE # TO Start
ENDIF

ELSE
* If we have only the name, find the next unpaid bill
DO WHILE Name=Key .AND. .NOT. EOF .AND. Looking
  IF Check:Nmbr <> ' '
    SKIP
  ELSE
    STORE F TO Looking
  ENDIF
ENDDO

* If we're on a new name or the end of the file, Looking is TRUE
* because we did not find the supplier we were looking for.
* Otherwise, we have an unpaid bill to confirm.
IF Looking
  ? ' There are no unpaid bills for this supplier.'
  ? '<Return> to continue.'
  WAIT
ENDIF
ENDIF

RETURN

```

```
***** PAYEMPS COMMAND FILE *****
* Does normal payroll processing or exceptions.
*****
```

```
SET TALK OFF
```

```
STORE T TO Salaries
```

```
DO WHILE Salaries
```

```
ERASE
```

```
@ 3,20 SAY ' PAYROLL FUNCTIONS '
```

```
@ 6,20 SAY ' 1> NORMAL PAYROLL '
```

```
@ 7,20 SAY ' 2> PARTIAL PAYMENT(S) '
```

```
@ 8,20 SAY ' 3> SKIP EMPLOYEE(S) '
```

```
@ 10,20 SAY ' <RETURN>'
```

```
WAIT TO, Action
```

```
IF Action = '1'
```

```
DO Payroll
```

```
ELSE
```

```
IF Action = '2'
```

```
ERASE
```

```
?
```

```
?
```

```
?
```

```
? 'This procedure allows you to pay less than a full salary if'
```

```
? 'for some reason an employee skipped days of work that are '
```

```
? 'not to be paid for. Do you want to continue (Y or N)?'
```

```
WAIT TO Continue
```

```
IF !(Continue) = 'Y'
```

```
RESTORE FROM B:Constant
```

```
USE B:Personne
```

```
? 'Select the employee number for partial payment:'
```

```
? ' (Type 0 to quit.)'
```

```
?
```

```
? 'NO. NAME % OF PAY'
```

```
LIST Name, Ratio*100 FOR .NOT. *
```

```
?
```

```
INPUT 'Which number(0 to quit)? ' TO Wipe
```

```
STORE INT(Wipe) TO Wipe
```

```
DO WHILE Wipe <> 0
```

```
GO Wipe
```

```
? 'How many days were worked'
```

```
? 'since the last regular payday?'
```

```
? 'Use decimals if needed (1 hour = 0.1333.)'
```

```
?
```

```
INPUT TO Worked
```

```
STORE Worked/11.0000 TO NewRatio
```

```
REPLACE Ratio WITH NewRatio
```

```
?
```

```
DISP Name, Ratio*100
```

```
?
```

```
INPUT 'Next (0 to quit)? ' TO Wipe
```

```
STORE INT(Wipe) TO Wipe
```

```
ENDDO
```

```
ENDIF
```

```
RELEASE All
```

```
?
```

```
? 'Do you want to SKIP any employees (Y or N)?'
```

```
WAIT TO Skip
```

```
IF !(Skip) <> 'Y'
```

```
DO Payroll
```

```
ENDIF
```

```
RELEASE Skip
```

```
ELSE
```



```

IF Action = '3'
  ERASE
  ?
  ?
  ?
  ? 'This procedure allows you to skip a paycheck in the payroll'
  ? 'procedure. Do you want to continue (Y or N)?'
  WAIT TO Continue
  IF I(Continue) = 'Y'
    RESTORE FROM B:Constant
    USE B:Personne
    ? 'Select the number of the employee to skip:'
    ? ' (Type 0 to quit.)'
    ?'NO.    NAME                SKIP'
    ?
    LIST Name, Paid FOR .NOT. *
    ?
    INPUT 'Which number (0 to quit)? ' TO Wipe
    STORE INT(Wipe) TO Wipe
    DO WHILE Wipe <> 0
      GO Wipe
      REPLACE Paid WITH T
      ?
      ?'NO.    NAME                SKIP'
      ?
      DISP Name, Paid
      ?
      INPUT 'Next? ("0"to quit) ' TO Wipe
      STORE INT(Wipe) TO Wipe
    ENDDO
  ENDIF
  RELEASE All

  ?
  ? 'Do you want to pay a partial salary'
  ? 'to any employees (Y or N)?'
  WAIT TO Part
  IF I(Part) <> 'Y'
    DO Payroll
  ENDIF
  RELEASE Part
ELSE
  IF Action = '4'
    ? 'Something 4'
    WAIT
  ELSE
    RELEASE All
    RETURN
  ENDIF 4
ENDIF 3
ENDIF 2
ENDIF 1
STORE T TO Salaries
ENDDO Salaries

```

```

***** PAYROLL COMMAND FILE *****
* This command file generates payroll check stubs showing all deductions; gets
* the next check number and writes a check in the CheckFile, showing the new
* balance; and stores the salaries and deductions in a database called Hold81.
* This file is used to store the monthly, quarterly and annual FIT, FICA, SDI
* and SIT deductions. The deductions are not picked up from tax tables because
* there are so few employees. Instead, they are obtained from the individual
* employee records in the Personnel database.
* Constants.MEM keeps track of the FICA and SDI percentages and their
* maximums, as well as the the constant for ThisYear. Changes can be thus
* made in a single spot and will be correct in all the programs in the
* accounting system.
* The file is quite long, but breaks down into simpler modules:
*   I: Get the date and End of Month, Quarter and Year flags.
*   II: Compute all deductions and net pay for an individual employee, then
*       place this in the employee record in Personne.DBF
*   III: Operator verifies deductions and payroll stub is printed.
*   IV: Paycheck is written to the Checkfil and all amounts are placed into
*       the Hold81 summary file.
*   V: When all individuals have been paid, the Hold81 summary file is
*      updated if it is the end of month, quarter or year.
*   VI: Print out the summary file and data so that the physical checkbook
*       can be updated (computer does not print our checks).
*   VII: Delete transient constants, save others back to Constant.MEM for
*        system use.
*****

```

```

*****
***** I: Get date and pay period flags *****

```

```

RESTORE FROM B:Constant
DO GetDate

```

```

STORE 'Y' TO GetWhen
DO WHILE !(GetWhen) = "Y"
  ERASE
  @ 1,18 SAY "PAYROLL PROCESSING"
  STORE " " TO EOY
  @ 4,8 SAY 'Want to change the date?' GET Date
  @ 5,8 SAY '(Press <Return> if okay.)'
  READ
  @ 7, 6 SAY "Is it the end of the YEAR?" GET EOY
  @ 7,35 SAY "(Y or N)"
  ? CHR(7)
  READ
  STORE !(EOY) TO EOY
  IF EOY = "Y"
    STORE "Y" TO EOQ
    STORE "Y" TO EOM
  ELSE
    STORE "N" TO EOY
    STORE " " TO EOQ
    @ 8, 3 SAY "Is it the end of the QUARTER?" GET EOQ
    @ 8,35 SAY "(Y or N)"
    ? CHR(7)
    READ
    STORE !(EOQ) TO EOQ
    IF EOQ = "Y"
      STORE "Y" TO EOM
    ELSE
      STORE "N" TO EOQ
      STORE " " TO EOM
      @ 9, 5 SAY "Is it the end of the MONTH?".GET EOM
      @ 9,35 SAY "(Y or N)"

```

```

? CHR(7)
READ
STORE ! (EOM) TO EOM
IF EOM <> "Y"
    STORE "N" TO EOM
ENDIF
ENDIF quarter
ENDIF year

ERASE
@ 4,25 SAY $(Date,1,2)+'/'+$(Date,3,2)+'/'+$(Date,5,2)
@ 6,0 SAY "End of YEAR: "+EOY+"      End of QUARTER: "+EOQ+;
      "      End of MONTH: "+EOM
STORE " " TO GetWhen
?
? '
@ 8,6 say 'The above information MUST be correct.
? CHR(7)
* 2nd chance at date and flags
ACCEPT '          Any CHANGES (Y or N)?' TO GetWhen

STORE 'B:Hold'+STR(ThisYear,2) TO Header
* Computer now does a date and flag check
IF !(GetWhen) <> 'Y'
    IF $(Date,5,2)<'26' .AND. EOM = 'Y'
        ?
        ?
        ? "CHECK THE INFO AGAIN. It's the end of the month, but the"
        ? 'date is '+Date-'. Do you want to make changes (Y or N)?'
        ? CHR(7)
        WAIT TO GetWhen
    ENDIF
    IF EOY = 'Y'
        SELECT SECONDARY
        USE &Header
        GO BOTTOM
        IF Marker = 'Y'
            ? CHR(7)
            ? 'You blew it--the end of the year has been done!'
            WAIT
            RELEASE All
            STORE T TO Paying
            RETURN
        ENDIF
    ENDIF
ENDIF
ENDIF
ENDDO GetWhen
RELEASE GetWhen

```

```

*****
***** II: Calculate deductions and net pay for each individual *****
* Compute deductions. Deductions for FICA, FIT, SDI and SIT are kept in the
* individual employee's Personnel record, rather than getting them from tax
* tables, because there are so few employees. (You have to decide what should
* and should not be computerized.) The "YTDxxx" variables are the year-to-date
* totals for these items. Limits and percentages for FICA and SDI are obtained
* from a file called Constant.MEM. These are the variables FICACut, FICAMax,
* FICAEnd, SDICut, SDIMax and SDIEnd.

```

```

SELECT PRIMARY

USE B:Personne
REPLACE All FICA WITH (Pay:Rate*FICACUT+0.005);

```

SDI WITH (Pay:Rate\*SDICUT+0.005)

STORE 0 TO Count

GO TOP

DO WHILE .NOT. EOF

IF Paid .OR. \*

SKIP

ELSE

STORE Count + 1 TO Count

\*\*\* Save the employee record in case the procedure is ended \*\*\*

STORE STR(5) TO Payee

COPY Record &Payee TO Bak

\*\*\* Deductions for partial salary based on number of days worked \*\*\*

\*\*\* Ratio is computed in PayMenu.CMD

IF Ratio < 1.0000

REPLACE Pay:Rate WITH Pay:Rate\*Ratio, FICA WITH FICA\*Ratio, FIT;  
WITH FIT\*Ratio, SDI WITH SDI\*Ratio, SIT WITH SIT\*Ratio

ENDIF

\* Deductions and totals are computed then stored in the employee record  
\* FedTemp, Statemp and EmpTemp are used to carry forward values for  
\* salaries subject to FICA, SDI and state unemployment insurance to  
\* Hold81, the summary file.

IF YTDSAL > FICAEnd

STORE 0 TO FedTemp

REPLACE FICA WITH 0

ELSE

IF (YTDSal + Pay:Rate) <= FICAEnd

REPLACE YTDFICA WITH (YTDFICA + FICA)

STORE Pay:Rate TO FedTemp

ELSE

REPLACE FICA WITH (MAXFICA - YTDFICA), YTDFICA WITH MAXFICA

STORE (FICAEnd - YTDSal) TO FedTemp

ENDIF

ENDIF

IF YTDSal > SDIEnd

STORE 0 TO StaTemp

REPLACE SDI WITH 0

ELSE

IF (YTDSAL + Pay:Rate) <= SDIEnd

REPLACE YTDSDI WITH (YTDSDI + SDI)

STORE Pay:Rate TO StaTemp

ELSE

REPLACE SDI WITH (MAXSDI - YTDSDI), YTDSDI WITH MAXSDI

STORE (SDIEnd-YTDSal) TO StaTemp

ENDIF

ENDIF

\* In California, the employer pays an Unemployment Insurance contribution  
\* on employee salary up to the amount of UIEnd. There is nothing  
\* deducted from the employee salary for this, so we keep track only of  
\* the employer obligation as UISal.

IF YTDSal > UIEnd

STORE 0 TO EmpTemp

ELSE

IF (YTDSal + Pay:Rate) <= UIEnd

STORE Pay:Rate TO EmpTemp

ELSE

STORE (UIEnd - YTDSal) TO EmpTemp

ENDIF  
ENDIF

REPLACE Net:Pay WITH (Pay:Rate-FICA-FIT-SDI-SIT)  
REPLACE YTFIT WITH (YTFIT + FIT)  
REPLACE YTDSIT WITH (YTDSIT + SIT)  
REPLACE QTDSal WITH (QTDSal + Pay:Rate)  
REPLACE YTDSal WITH (YTDSal + Pay:Rate)

\*\*\*\*\*  
\*\*\*\*\* III: Print employee stub \*\*\*\*\*

```
ERASE
SET PRINT ON
? '      ' +$(Date,3,2)+'/'+$(Date,5,2)+'/'+$(Date,1,2)+'': '+Name;
? '      ' +$(SS:Nmbr,1,3)+'-' +$(SS:Nmbr,4,2)+'-' +$(SS:Nmbr,6,4)
? '      GROSS PAY: $'-STR(Pay:Rate,7,2)+'      NET PAY: $';
? '      -STR(Net:Pay,7,2)
?
? '
? '      THIS CHECK:      FICA      FIT      SDI      SIT'
? '      '+STR(FICA,6,2)+'      '+STR(FIT,7,2);
? '      '+STR(SDI,5,2)+'      '+STR(SIT,7,2)
? '      THIS YEAR:      '+STR(YTFICA,7,2)+'      '+STR(YTFIT,8,2);
? '      '+STR(YTDSI,6,2)+'      '+STR(YTDSIT,7,2)
? '      TOTAL SALARY THIS QUARTER: $'-STR(QTDSal,9,2)
? '      TOTAL SALARY THIS YEAR: $'-STR(YTDSal,9,2)
?
?
?
```

\* Pagefeed after every six employee stubs  
IF Count >= 6  
? CHR(12)  
STORE 0 TO Count  
ENDIF  
SET PRINT OFF

IF EQ = 'Y' .AND. Paid  
REPLACE QTDSal WITH 0  
ENDIF

\*\*\*\*\*  
\*\*\*\*\* IV: Record paycheck in Checkfil and Hold81 \*\*\*\*\*

```
* Now a check is "written" in the Checkfil.
SELECT SECONDARY
USE B:Checkfil
APPEND BLANK
REPLACE Check:Nmbr WITH NextCheck, Check:Date WITH Date,;
      Name WITH P.Name, Amount WITH Net:Pay, Emp:Nmbr;
      WITH P.Emp:Nmbr, Client WITH 'OFC', Job:Nmbr WITH 31,;
      Descrip WITH 'SALARY', Balance WITH (MBalance - Amount)
STORE (MBalance - Amount) TO MBalance
STORE STR(VAL(NextCheck)+1,4) TO NextCheck
```

```
ERASE
@ 3,25 SAY "*** DO NOT INTERRUPT ***"
@ 5,25 SAY "UPDATING MASTER RECORD"
? CHR(7)
* We keep an aggregate record of payroll and deductions. The amounts
* for each employee are added to the amounts already in the last
* record in the file represented by "Header". (This was set up at the
* start of the "GetWhen" loop earlier, and has the name "B:Hold81" or
* "B:Hold82" or whatever "ThisYear" is.)
* This last record is either a blank (if this is the first
```

```
* payroll of the month), or has data from previous salary payments
* made during the current month. At the end of the month, quarter and
* year, totals and a new blank record (except at the end of the year)
* are added. This is done in the next loop.
```

```
USE &Header
```

```
* If this is a new year, there are no records in the file so we add a
* blank record. Otherwise, we go to the last record in the file.
```

```
IF EOF
  APPEND BLANK
ELSE
  GO BOTTOM
ENDIF
```

```
REPLACE Check:Date WITH Date, Payroll WITH (Payroll+Pay:Rate),;
  FICA WITH (FICA+P.FICA), FICASal WITH (FICASal + FedTemp),;
  FIT WITH (FIT + P.FIT), SDI WITH (SDI+P.SDI),;
  SDISal WITH (SDISal + Statemp), SIT WITH (SIT + P.SIT),;
  UISal WITH (UISal + EmpTemp)
SELECT PRIMARY
```

```
*** Reset the employee record if he was paid for part time. ***
*** The Bak file is not deleted here, as each copy command ***
*** above wipes out the previous contents. ***
IF Ratio <> 1.0000
  REPLACE Ratio WITH 1.0000
  UPDA FROM Bak ON Emp:Nmbr REPL Pay:Rate,FICA,FIT,SDI,SIT,Net:Pay
ENDIF
ENDIF
```

```
SKIP
LNDDO personnel file
```

```
*****
***** V: Personnel records are reset and Holdxx is updated *****
STORE ' ' TO Completed
REPLACE All Paid WITH F
```

```
USE &Header
GO BOTTOM
IF EOM = 'Y'
```

```
  REPLACE Marker WITH 'M'
```

```
* If it's the end of the quarter, we total the amounts for the
* previous three months to a new record and mark it with a 'Q'.
```

```
IF EQQ = 'Y'
  STORE STR(,5) TO Number
  TOTAL ON Marker TO Quarter FOR # >= (VAL(Number)-2)
  APPEND FROM Quarter
  DELETE FILE Quarter
```

```
IF $(Date,3,2) = '03'
  REPLACE Check:Date WITH '1ST'
ELSE
  IF $(Date,3,2) = '06'
    REPLACE Check:Date WITH '2ND'
  ELSE
    IF $(Date,3,2) = '09'
      REPLACE Check:Date WITH '3RD'
    ELSE
      IF $(Date,3,2) = '12'
        REPLACE Check:Date WITH '4TH'
```

```

      I ENDIF
    ENDIF
  ENDIF
ENDIF

REPLACE Marker WITH 'Q',

* If it's the end of the year, we total all the quarterly amounts to
* a new record and mark it with a 'Y'.
IF EOY = 'Y'
  TOTAL ON Marker TO Annual FOR Marker = 'Q'
  APPEND FROM Annual
  REPLACE Marker WITH 'Y', Check:Date WITH 'END'
  DELETE FILE Annual
ENDIF
ENDIF

* If it's the end of a month but not the end of the year, we add a new
* blank record for next month's payroll records.
IF EOY <> 'Y'
  APPEND BLANK
ENDIF
ENDIF

*****
***** VI: Print payroll summary, transfer checks to costbase *****
USE B:Checkfil
COUNT FOR .NOT. * TO Any
IF Any=0
  ? '      No new checks written.'
  ? '      <Return> to continue.'
  WAIT
ELSE
  USE &Header
  ERASE
  @ 12,25 SAY "CHECK THE PRINTER, THEN PRESS <RETURN>."
  ? CHR(7)
  WAIT
  ERASE
  SET PRINT ON
  SET MARGIN TO 45
  ? '      MASTER PAYROLL FILE SUMMARY: '+$(Date,3,2)+'/'+';
      +$(Date,5,2)+'/'+'+$(Date,1,2)

  ?
  ?
  ?'DATE   PAYROLL   FICA   FICASAL   FIT   SDI   SDISAL   '+';
      'SIT     UISal'

  ?
  LIST OFF
  SET MARGIN TO 38
  ? CHR(12)
  SET PRINT OFF

  ERASE
  @ 3,25 SAY "**** DO NOT INTERRUPT ****"
  @ 5,25 SAY " UPDATING THE COSTBASE"
  ? CHR(7)

  USE B:CostBase INDEX B:$Supp
  APPEND FROM B:Checkfil

  DO CheckStub

```

ENDIF

```
*****6
***** VII: Dump transient variables, save necessary ones *****
RELEASE Payee,Number,Date,Ratio,Aborted,Printed,EOY,EOQ,EOM,Any,Header;;
      Count, FedTemp, StaTemp, EmpTemp, Marker, Paying, Salaries
SAVE TO B:Constant
```

```
USE
RELEASE All
DELETE FILE Bak
RETURN
```



```
***** DEPMENU COMMAND FILE *****
* Select deposits or perform houskeeping on the checkbook.
*****
```

```
STORE T TO Incoming
DO WHILE Incoming
  ERASE
  @ 5,20 SAY ' 1> ENTER MONEY COMING IN '
  @ 7,20 SAY ' 2> CHANGE OUR CHECK NUMBER '
  @ 9,20 SAY ' 3> CHANGE CHECKBOOK BALANCE '
  ?
  ?
  ? ' <RETURN>'
  WAIT TO Action

  IF Action = '1'
    DO Deposits
  ELSE
    IF Action = '2'
      RESTORE FROM B:Constant
      ERASE
      @ 5,0 SAY 'This is the next check number' GET NextCheck
      @ 6,0 SAY 'To leave it unchanged, use the <return>.'
      @ 7,0 SAY 'To change it, just type in the new number.'
      READ
      SAVE TO B:Constant
      RELEASE All
    ELSE
      IF Action = '3'
        RESTORE FROM B:Constant
        STORE 'Y' TO Change
        DO WHILE !(Change) = 'Y'
          ERASE
          @ 5,0 SAY ' The current balance is:' GET MBalance
          ? 'To leave it unchanged, use the <return>.'
          ? 'To change it, just type in the new value.'
          READ
          ?
          ?
          ? ' Want to change your mind (Y or N)?'
          WAIT TO Change
        ENDDO
        RELEASE Change
        SAVE TO B:Constant
        RELEASE All
      ELSE
        RELEASE All
        RETURN
      ENDIF 3
    ENDIF 2
  ENDIF 1
  ERASE
  STORE T TO Incoming
ENDDO Incoming
```

## \*\*\*\*\* DEPOSITS COMMAND FILE \*\*\*\*\*

\* This file records any money coming in in a file called Deposits. If the  
 \* the money is in payment of an invoice, the amount and date of payment are  
 \* entered against that invoice in the Invoice file.  
 \* The checkbook balance is kept current for each entry.  
 \* At the end of the session, deposits are printed out individually, then  
 \* the total of deposits plus the new checkbook balance are printed.  
 \*\*\*\*\*

RESTORE FROM B:Constant

ERASE

@ 5,20 SAY ' ENTERING INCOME'  
 @ 7, 5 SAY 'The STARTING BALANCE is '+STR(MBalance,9,2)

?  
 ? ' If this does not match the checkbook,'  
 ? ' <Return> to the main menu to change.'  
 ?  
 ? ' C to CONTINUE.'  
 ?

WAIT TO Continue

IF !(Continue) <> 'C'

RELEASE All

RETURN

ENDIF

RELEASE Continue

DO GetDate

SELECT PRIMARY

USE B:Deposits

COPY STRUCTURE TO GetDep

USE GetDep

STORE 'Y' TO Depositing

DO WHILE !(Depositing) <> 'F'

APPEND BLANK

STORE STR(#,5) TO Number

REPLACE Dep:Date WITH Date

ERASE

\* Next loop is used when there has been an error in the entry  
 \* (defined as no client or no rate). The operator is shown the  
 \* previous entries and can make any changes required.

STORE 'T' TO Incorrect

DO WHILE !(Incorrect) <> 'F'

@ 3, 0 SAY ' If a check covers more than one agency invoice,'

@ 4, 0 SAY ' enter each invoice and amount separately.'

?

@ 6,0 SAY ' RECORD NUMBER: '-Number

@ 7,0 SAY ' HOW MUCH' GET Deposit

@ 8,0 SAY 'OUR INVOICE NO' GET Inv:Nmbr

@ 9,0 SAY ' CHECK FROM' GET Payer

@ 10,0 SAY 'THEIR CHECK NO' GET Pay:Nmbr

@ 11,0 SAY ' Comments' GET Comments

? CHR(7)

READ

REPLACE Payer WITH !(Payer), Comments WITH !(Comments)

@ 9,15 SAY Payer

@ 11,15 SAY Comments

IF Payer <> ' ' .AND. Deposit > 0

@ 17,5 SAY ' C to CHANGE,'

```

@ 18,5 SAY '<Return> to continue.'
?
? CHR(7)
WAIT TO Depositing
IF !(Depositing)='C'
  STORE 'T' TO Incorrect
  ERASE
ELSE
  STORE (MBalance + Deposit) TO MBalance
  @ 17, 5 SAY ' F if FINISHED,'
  ?
  ? '
  ? CHR(7)
  WAIT TO Depositing
  STORE 'F' TO Incorrect
ENDIF
ELSE
@ 15,5 SAY 'CHECK WRITER or AMOUNT missing.'
?
? ' F if FINISHED,'
? ' <Return> to correct the record.'
? CHR(7)
WAIT TO Depositing
ERASE
IF !(Depositing)='F'
  DELETE RECORD &Number
  STORE 'F' TO Incorrect
ELSE
  ERASE
  STORE 'T' TO Incorrect
ENDIF
ENDIF
ENDDO Incorrect
ENDDO Depositing
RELEASE Change, Date, NoDate, Depositing, Number, Update, New, Incorrect
SAVE TO B:Constant

COUNT FOR .NOT. * TO Any
ERASE
IF Any = 0
? 'No deposits to add to the file.'
? 'Press any key to continue.'
? CHR(7)
USE
WAIT
ELSE
DO DepPrint

* The next portion of this program uses the Primary and Secondary work
* areas to record payments received against agency invoices in the record
* for that invoice in the Invoices file. Both work areas are necessary so
* that we can compare each record in the GetDep file against all of the
* records in the Invoices file.
DO DepTrans
USE B:Deposits
APPEND FROM GetDep
ENDIF there are deposits to add to the file

DELETE FILE GetDep
RELEASE All
RETURN

```

```

***** DEPPRINT COMMAND FILE *****
* Prints valid deposits in the GetDep file as part of the Deposits file.
*****

@ 5,10 SAY 'To print the deposits you just entered,'
@ 6,10 SAY 'press <Return>.'
? CHR(7)
WAIT
SET PRINT ON
? ' DATE      PAID BY                AMOUNT  INV #  COMMENTS:'
?
GO TOP
STORE 0 TO Count
DO WHILE .NOT. EOF
  DISPLAY OFF Dep:Date, Payer, Deposit, Inv:Nmbr, Comments FOR .NOT. *
  SKIP
  STORE Count+1 TO Count
  IF Count=10
    STORE 0 TO Count
    WAIT
  ENDF
ENDDO
SUM Deposit TO Temp
?
? ' The total deposit is ' + STR(Temp,9,2)
?
? ' The final balance is ' + STR(Mbalance,9,2)
?
SET PRINT OFF

RELEASE Count, Temp
RETURN

```

## \*\*\*\*\* DEPTRANS COMMAND FILE \*\*\*\*\*

\* Applies deposits from the GetDep file against the matching invoices in the  
 \* Invoices file as payments are received against them.  
 \*\*\*\*\*

```

GO TOP
ERASE
DO WHILE .NOT. EOF
  STORE STR(#,5) TO Number
  @ 6,20 SAY 'RECORD NUMBER '+Number
  ? CHR(7) + CHR(27) + CHR(74)

  IF Inv:Nmbr=' ' .OR. *
    SKIP
  ELSE
    @ 7,20 SAY 'INVOICE NUMBER '+Inv:Nmbr
    STORE Inv:Nmbr TO Key

    SELECT SECONDARY
    USE B:Invoices INDEX B:Invoices
    FIND &Key
    STORE T TO Again
    STORE 'T' TO Decision
    IF # = 0
      DO WHILE Again
        @ 9,15 SAY 'THIS INVOICE NUMBER IS NOT IN THE INVOICE FILE.
        @ 11,15 SAY '      E to EDIT it.
        @ 12,15 SAY '      C to CONTINUE.
        ?
        ? CHR(7)
        WAIT TO Decision

        IF !(Decision) = 'E'
          SELECT PRIMARY
          EDIT &Number
          SELECT SECONDARY
          STORE F TO Again
        ELSE
          IF !(Decision) = 'C'
            STORE F TO Again
          ELSE
            STORE T TO Again
          ENDIF C
        ENDIF E
      ENDDO
    ELSE
      REPLACE Amt:Rcd WITH (Amt:Rcd + Deposit), Date:Rcd WITH Dep:Date
    ENDIF 0
    SELECT PRIMARY

    * We do not skip to the next record if the record was edited. This
    * allows us to run the edited record through the process again.
    IF !(Decision) <> 'E'
      SKIP
    ENDIF
  ENDIF no invoice number or record deleted
ENDDO the transfer

```

## \*\*\*\*\* IOMENU COMMAND FILE \*\*\*\*\*

\* Selects the appropriate action to be taken with insertion orders  
 \* (instructions from our ad agency to magazine publishers).  
 \*\*\*\*\*

```

STORE T TO Inserting
DO WHILE Inserting
  ERASE
  @ 7,20 SAY ' 1> ENTER INSERTION ORDERS'
  @ 9,20 SAY ' 2> EDIT INSERTION ORDERS'
  @ 11,20 SAY ' 3> REVIEW/PRINT INSERTION ORDERS'
  @ 12,20 SAY ' BY CLIENT & MAGAZINE'
  @ 14,20 SAY ' <RETURN>'
  WAIT TO Action

  IF Action = '1'
    DO IOPost
  ELSE
    IF Action = '2'
      STORE "Y" TO Changing
      DO WHILE !(Changing)='Y'
        USE B:Inserts
        IF EOF
          ? 'There are no entries in the INSERTION ORDER file.'
          STORE "N" TO Changing
        ELSE
          STORE IO:Nmbr TO First
          GO BOTTOM
          STORE IO:Nmbr TO Last
          ERASE
          @ 3,15 SAY 'EDITING INSERTION ORDERS '+First+'thru '+Last
          @ 5,15 SAY '~W to SAVE, ^Q to CANCEL changes you make.'
          @ 6,15 SAY '^R for PREVIOUS, ^C for NEXT record if MORE = T'
          ?
          ?
          ACCEPT 'Which ORDER NUMBER do you want to EDIT?' TO Order
          USE B:Inserts INDEX B:Inserts
          FIND &Order
          IF #=0
            ?
            ?
            ? 'That insertion order is not in the file.'
            ? 'Do you want to continue (Y or N)?'
            WAIT TO Changing
          ELSE
            STORE STR(%,5) TO Number
            Edit &Number
            REPLACE Client WITH !(Client), Ad WITH !(Ad),Magazine WITH;
              !(Magazine)
            ?
            ? 'Do you want to edit any other insertion orders (Y or N)?'
            WAIT TO Changing
          ENDIF
        ENDIF
      ENDDO Changing
      RELEASE All
    ELSE
      IF Action = '3'
        DO IOReview
      ELSE
        RELEASE All
        RETURN
      ENDIF 3
    ENDIF 2
  ENDIF

```

ENDIF 1  
STORE T TO Inserting  
ENDDO Inserting

\*\*\*\*\* IOPOST COMMAND FILE \*\*\*\*\*

\* Gets information for insertion orders (instructions to magazine  
 \* publishers from our ad agency). Works much like Postbills and  
 \* Posttime.  
 \*\*\*\*\*

RESTORE FROM B:Constant

DO GetDate

USE B:Inserts  
 COPY STRUCTURE TO GetInserts  
 USE GetInserts

STORE ' ' TO New  
 STORE 'Y' TO Inserting  
 DO WHILE !(Inserting) <> 'F'  
 APPEND BLANK  
 STORE STR(%,5) TO Number  
 REPLACE IO:Date WITH Date, IO:Nmbr WITH Next:IO

ERASE  
 \* Next loop is used when there has been an error in the entry  
 \* (defined as no client or no rate).  
 STORE 'I' TO Incorrect  
 DO WHILE !(Incorrect) <> 'F'

ERASE  
 @ 4,0 SAY ' INSERTION ORDER: '+IO:Nmbr  
 @ 4,30 SAY ' DATE:'+Date  
 ?  
 @ 6,0 SAY ' RECORD NUMBER: '-Number  
 IF !(New) = 'S'  
 @ 7,0 SAY ' OUR CLIENT: ' + MClient  
 ELSE  
 @ 7,0 SAY ' OUR CLIENT ' GET MClient  
 STORE !(MClient) TO MClient  
 ENDIF  
 @ 8,0 SAY ' JOB NUMBER ' GET Job:Nmbr  
 @ 9,0 SAY ' AD DESCRIPTION ' GET Ad  
 @ 10,0 SAY ' HOW MUCH SPACE ' GET Space  
 @ 11,0 SAY ' WHICH MAGAZINE ' GET Magazine  
 @ 12,0 SAY ' WHICH ISSUE ' GET Issue  
 @ 13,0 SAY 'GROSS SPACE COST ' GET Gross:Cost  
 @ 14,0 SAY ' DISCOUNT RATE ' GET Times  
 READ

REPLACE Net:Cost WITH Gross:Cost\*0.8500, Client WITH MClient,  
 Ad WITH !(Ad), Magazine WITH !(Magazine), Issue WITH !(Issue)

@ 7,18 SAY Client  
 @ 9,18 SAY Ad  
 @ 11,18 SAY Magazine  
 @ 12,18 SAY Issue  
 @ 15,0 SAY ' NET SPACE COST ' GET Net:Cost

IF Client <> ' ' .AND. Gross:Cost > 0 .AND. Job:Nmbr > 99  
 @ 18,5 SAY ' C to CHANGE,'  
 @ 19,5 SAY '<Return> to continue.'  
 ?

WAIT TO New  
 IF !(New)='C'  
 STORE 'I' TO Incorrect  
 ELSE  
 @ 17, 5 SAY ' -F if FINISHED,'  
 @ 18, 5 SAY ' S for SAME insertion order,'



```

@ 19, 5 SAY '<Return> for NEXT insertion order.'
@ 21, 0 SAY '
ACCEPT TO New

IF !(New) <> 'S'
  IF VAL(Next:IO) < 9999
    STORE STR(VAL(Next:IO)+1,4) TO Next:IO
  ELSE
    STORE '1001' TO Next:IO
  ENDIF
ENDIF
STORE 'F' TO Incorrect
ENDIF
STORE New TO Insertinz
ELSE
  ?
  ?
  ?
  ?
  ?
  ? ' CLIENT, JOB or RATE missing.'
  ?
  ? ' F if FINISHED,'
  ? ' <Return> to correct the record.'
  ?
  WAIT TO Inserting
  IF !(Inserting)= 'F'
    DELETE RECORD &Number
    STORE 'F' TO Incorrect
  ELSE
    STORE 'T' TO Incorrect
  ENDIF
ENDIF
ENDDO Incorrect
ENDDO Inserting

RELEASE Date, NoDate, Inserting, Number, Update, New, Incorrect
SAVE TO B:Constant

COUNT FOR .NOT. * TO Any
ERASE
IF Any = 0
  ? 'No insertions to add to the file.'
  ? 'Press any key to continue.'
  USE
  WAIT
ELSE
  @ 5,10 SAY 'To print the insertions you just entered,'
  @ 6,10 SAY 'press <Return>.'
  WAIT TO Number
  *"Number" determines the starting record number for the printou

SET PRINT ON
? 'IO # MAGAZINE      ISSUE   JOB   AD      ';
  + 'SPACE          GROSS   NET X DATE'
?

* "Output" and "Condition" needed in the Printout Command file
STORE 'Y' TO Output
STORE 'OFF' TO Condition
DO Printout

ERASE

```

```
@ 5,20 SAY 'UPDATING THE INSERTION ORDER FILE'  
USE B:Inserts INDEX B:Inserts  
APPEND FROM GetInserts  
ENDIF
```

```
DELETE FILE GetInserts  
RELEASE All  
RETURN
```

```

***** IOREVIEW COMMAND FILE *****
* Provides insertion order displays and printout.
*   The operator can select all the insertions for the client,
*   or can select only those for a particular magazine.
*****

```

```

SET TALK OFF
USE B:Inserts

```

```

STORE ' ' TO Again
DO WHILE !(Again) <> 'F'
  STORE ' ' TO MClient
  STORE ' ' TO MMagazine
  STORE ' ' TO Hardcopy
  STORE ' ' TO Other
  ERASE
  @ 2,11 SAY ' MEDIA SUMMARY:'
  @ 4,11 SAY 'ENTER CLIENT CODE' GET MClient
  @ 5,11 SAY ' MAGAZINE NAME?' GET MMagazine
  @ 6,11 SAY ' P to PRINT' GET Hardcopy
  READ
  IF MClient = ' '
    @ 9, 0 SAY ' '
    ? ' CLIENT missing.'
    ? ' F if Finished,'
    ? ' <Return> to continue.'
    WAIT TO Again
  ELSE
    STORE !(MClient) TO MClient
    STORE !(MMagazine) TO MMagazine
    STORE !(Hardcopy) TO Hardcopy
    @ 4,29 SAY MClient
    @ 5,29 SAY MMagazine
    @ 6,29 SAY Hardcopy
    @ 9, 0 SAY ' '
    ?
    ?
    ACCEPT 'Type C to CHANGE any entries' TO Changes
    IF !(Changes) = 'C'
      STORE '~' TO again
      ERASE
    ELSE
      IF MMagazine >'
        STORE TRIM(MMagazine) TO MMagazine
        STORE '.AND. Magazine=MMagazine' TO Condition
      ELSE
        STORE CHR(0) TO Condition
      ENDIF
      IF !(Hardcopy) = 'P'
        STORE 'TO PRINT' TO Hardcopy
      ELSE
        STORE CHR(0) TO Hardcopy
      ENDIF Hardcopy

      _SET HEADING TO MEDIA SUMMARY FOR &MClient &MMagazine
      REPORT FORM Media &Hardcopy FOR Client=MClient.&Condition
      ?
      ? ' F if Finished,'
      ? ' <Return> to continue.'
      WAIT TO Again
      ERASE
    ENDIF okay to do the report
  ENDIF

```

ENDDO Again

ERASE  
RELEASE All  
RETURN

## \*\*\*\*\* INVMENU COMMAND FILE \*\*\*\*\*

\* Functions are selected by the menu. This procedure works with two data  
 \* files, BILLINGS and INVOICES. BILLINGS keeps track of the amount  
 \* billed to a client by individual job number, while INVOICES is a  
 \* summary of the total billed on any given invoice. This latter file can  
 \* be used to set up an accounts receivable system, as it has fields for  
 \* storing how much has been received in payment against an invoice and  
 \* when that amount was received (filled in by the Deposits.CMD file).  
 \*\*\*\*\*

ERASE

STORE T TO Invoicing

DO WHILE Invoicing

```
@ 5,20 SAY ' 1> BILL CLIENTS BY JOB'
@ 7,20 SAY ' 2> EDIT INVOICES and BILLINGS'
@ 9,20 SAY ' 3> REVIEW/PRINT INVOICES and BILLINGS'
@ 12,20 SAY ' <RETURN>'
WAIT TO Action
```

IF Action = '1'

DO Invoices

ELSE

IF Action = '2'

STORE 'Y' TO Changing

DO WHILE !(Changing) = 'Y'

ERASE

? ' J to edit individual job billings,'

? ' &lt;Return&gt; to edit the summary invoices.'

WAIT TO Which

IF !(Which) = 'J'

STORE 'Billings' TO Database

ELSE

STORE 'Invoices' TO Database

ENDIF

USE B:&amp;Database

STORE Inv:Nmbr TO First

GO BOTTOM

STORE Inv:Nmbr TO Last

ERASE

@ 3,10 SAY 'EDITING '+!(Database)

@ 3,35 SAY First+'thru '+Last

@ 5,10 SAY '^W to SAVE, ^Q to CANCEL changes you make.'

@ 6,10 SAY '^R for PREVIOUS, ^C for NEXT record.'

@ 8,10 SAY 'Which INVOICE NUMBER do you want to EDIT?'

IF !(Which) = 'J'

@ 9,10 SAY 'This takes you to the FIRST ENTRY for that number.'

@ 10,10 SAY 'Use ^C to look at the rest of them.'

ENDIF

ACCEPT TO Invoice

USE B:&amp;Database INDEX B:&amp;Database

FIND &amp;Invoice

IF #=0

?

?

? 'That invoice number is not in the file.'

? 'Do you want to continue (Y or N)?'

WAIT TO Changing

ELSE

STORE STR(#,5) TO Number

Edit &amp;Number

REPLACE Sales:Tax WITH 0.06\*Taxable

REPLACE Client WITH !(Client)

```

IF !(Which) = 'J'
  REPLACE Descrip WITH !(Descrip),PO:Nmbr WITH !(PO:Nmbr)
ENDIF
?
? 'Do you want to edit any other invoices (Y or N)?'
WAIT TO Changing
ENDIF
ENDDO Changing
RELEASE All
ELSE
IF Action = '3'
  ERASE
  @ 4, 0 SAY ' '
  ? '          J to see individual job billings,'
  ? '          <Return> to see the summary invoices.'
  WAIT TO Which
  IF !(Which) = 'J'
    STORE 'Billings' TO Database
  ELSE
    STORE 'Invoices' TO Database
  ENDIF

  USE B:&Database
  STORE 'Y' TO Reviewing
  DO WHILE !(Reviewing)='Y'
    GO BOTTOM
    STORE STR(#,5) TO Last
    ERASE
    @ 5,10 SAY 'The '+!(Database)+' file has '-Last-' entries.'
    @ 7,10 SAY '<Return> to see the entire file, or'
    @ 8,10 SAY 'enter the record number to start on.'
    ACCEPT TO Number

    ? 'Do you want to print the file now (Y or N)?'
    WAIT TO Output

    IF !(Output)='Y'
      SET PRINT ON
    ENDIF

    STORE CHR(0) TO Condition
    DO Printout
    ?
    SET PRINT OFF
    ? 'Do you want to see it again (Y or N)?'
    WAIT TO Reviewing
    ERASE
  ENDDO Reviewing
  RELEASE All
ELSE
  RELEASE All
  RETURN
ENDIF 3
ENDIF 2
ENDIF 1
ERASE
STORE T TO Invoicing
ENDDO Invoicing

```

\*\*\*\*\* INVOICES COMMAND FILE \*\*\*\*\*

\* This file accepts inputs for invoices to clients. Individual projects  
 \* and items are stored in the Billings data file. Any number of items  
 \* may be entered using a single invoice number. Invoice numbers are  
 \* automatically generated by the computer and stored in the Constant.Mem  
 \* file.  
 \* After all the job billings have been entered, they are summarized b  
 \* invoice number and the data is stored in the Invoices file.  
 \* A printout of items billed and invoice totals is provided.  
 \*\*\*\*\*

RESTORE FROM B:Constant

DO GetDate

USE B:Billings

COPY STRUCTURE TO GetCosts

USE GetCosts

STORE ' ' TO Billing

DO WHILE !(Billing) <> 'F'

APPEND BLANK

STORE STR(#,5) TO Number

REPLACE Inv:Date WITH Date. Inv:Nmbr WITH Next:Inv

ERASE

STORE 'T' TO Entering

DO WHILE !(Entering) <> 'F'

ERASE

@ 3, 0 SAY 'INVOICE NUMBER '+Next:Inv

@ 3,30 SAY ' DATE '+Inv:Date

@ 5,0 SAY ' RECORD NUMBER: '-Number

IF !(Billing) = 'S'

@ 7,0 SAY ' CLIENT:'+ MClient

REPLACE Client WITH MClient

ELSE

@ 7,0 SAY ' CLIENT ' GET Client

ENDIF

@ 8,0 SAY ' JOB NUMBER ' GET Job:Nmbr

@ 9,0 SAY 'TAXABLE AMOUNT ' GET Taxable

@ 10,0 SAY 'TAXFREE AMOUNT ' GET TaxFree

@ 11,0 SAY ' P. O. NUMBER ' GET PO:Nmbr

@ 12,0 SAY ' DESCRIPTION ' GET Descrip

READ

STORE !(Client) TO MClient

REPLACE Client WITH MClient, Descrip WITH !(Descrip),;

PO:Nmbr WITH !(PO:Nmbr)

@ 7,16 SAY Client

@ 11,16 SAY PO:Nmbr

@ 12,16 SAY Descrip

IF Taxable > 0

REPLACE Sales:Tax WITH 0.06\*Taxable

@ 13,0 SAY ' SALES TAX'GET Sales:Tax

ENDIF

IF Job:Nmbr < 100

@ 16,0 SAY ' JOB not 3 digits.'

ENDIF

IF MClient <> ' ' .AND. (Taxable > 0 .OR. TaxFree > 0)

@ 17,0 SAY ' C to CHANGE this entry.'

? \* <Return> to continue.'

```

WAIT TO New
IF !(New)='C'
  STORE 'T' TO Entering
ELSE
  @ 16, 0 SAY '          F if FINISHED,'
  @ 17, 0 SAY '          S for SAME invoice number,'
  @ 18, 0 SAY '    <Return> for NEXT invoice number.'
  @ 19, 0 SAY '
  ACCEPT TO New

  IF !(New) <> 'S'
    STORE STR(VAL(Next:Inv)+3,5) TO Next:Inv
  ENDIF
  STORE 'F' TO Entering
ENDIF
STORE New TO Billing
ELSE
  @ 17,0 SAY '          CLIENT or AMOUNT missing.'
  ?
  ? '          F if FINISHED,'
  ? '    <Return> to correct the record.'
  WAIT TO Billing
  IF !(Billing)='F'
    DELETE RECORD &Number
    STORE 'F' TO Entering
  ELSE
    STORE 'T' TO Entering
  ENDIF
ENDIF
ENDDO Entering
ENDDO Billing

RELEASE Billing, Entering, MClient, Task, Number, Date, New
SAVE TO B:Constant

PACK

GO TOP
ERASE
IF EOF
  ? 'No invoices to add to the file.'
  ? 'Press any-key to continue.'
  WAIT
ELSE
  @ 5,20 SAY '**** DO NOT INTERRUPT ****'
  @ 7,20 SAY 'UPDATING BILLINGS AND INVOICES'

  * Costs entered are totalled by invoice number to Scratch because several
  * job costs can be entered against each invoice number. Amounts are adusted
  * for one client who always pays promptly and takes a 2% discount. Each
  * invoice is totalled. Temp has only summary data needed for a printout.

  USE B:Invoices
  COPY STRUCTURE TO Scratch

  USE GetCosts
  ERASE
  @ 5,10 SAY 'When ready to print the billings you just added,'
  @ 6,10 SAY 'press <Return>'
  TOTAL ON Inv:Nmbr TO Scratch FIELDS Taxable, Sales:Tax, TaxFree
  WAIT TO Nmber

  SET PRINT ON
  ? 'ENTRIES BY JOB NUMBER:'

```



```
?
? 'INV # JOB DATE TAXABLE TAX TAXFREE P.O.# DESCRIPTION'
?
```

```
* "Output" is needed in the Printout Command file
STORE 'Y' TO Output
STORE 'OFF' TO Condition
DO Printout
```

```
* One of our clients always pays promptly and takes a 2% discount.
* We do this after the original entries were printed out:
REPLACE Taxable WITH 0.980*Taxable, TaxFree WITH 0.980*TaxFree, Sales:Tax;
WITH 0.980*Sales:Tax FOR Client = 'SPI'
```

```
?
? 'Updating the BILLINGS database now.'
USE B:Billings INDEX B:Billings
APPEND FROM GetCosts
```

```
USE Scratch
REPLACE All Amount WITH (Taxable + Sales:Tax + TaxFree)
COPY TO Temp FIELDS Inv:Date, Inv:Nmbr, Taxable, Sales:Tax,;
TaxFree, Amount
REPLACE Taxable WITH 0.980*Taxable, TaxFree WITH 0.980*TaxFree, Sales:Tax;
WITH 0.980*Sales:Tax, Amount WITH 0.980*Amount FOR Client = 'SPI'
```

```
USE Temp
STORE 'Y' TO Output
SET PRINT ON
?
?
? 'TOTALS BY INVOICE NUMBER:'
?
? 'DATE INV# TAXABLE TAX TAXFREE TOTAL'
?
```

```
DO Printout
?
? 'Updating the INVOICES database now.'
USE B:Invoices INDEX B:Invoices
APPEND FROM Scratch
```

```
ENDIF
```

```
USE
DELETE FILE Scratch
DELETE FILE Temp
DELETE FILE GetCosts
RELEASE All
RETURN
```

\*\*\*\*\* REPORTMENU COMMAND FILE \*\*\*\*\*

- \* This command file is a sub-module of the ACCOUNTS.COMD control
- \* module. It provides detailed choices that relate to reports
- \* that the user might choose to see or print from the cost
- \* database. The functions are set up as sub-sub-procedures
- \* under the control of this module.

\*\*\*\*\*

ERASE

STORE T TO Reporting

DO WHILE Reporting

```

@ 3,20 SAY ' 1> COSTS BY JOB'
@ 5,20 SAY ' 2> FIND & EDIT BILLS'
@ 7,20 SAY ' 3> REVIEW A DATABASE'
@ 9,20 SAY ' 4> Quarterly Sales Tax Summary'
@ 11,20 SAY ' 5> RE-INDEX THE COSTBASE ON JOB NUMBERS'
@ 12,20 SAY ' Make sure you won't need the computer'
@ 13,20 SAY ' for a while: this takes a long time.'
```

```

@ 17,20 SAY ' <RETURN>'
```

WAIT TO Action

IF Action = '1'

USE B:Postfile

COUNT FOR .NOT. \* TO Any

IF Any > 0

```

@ 15, 0 SAY CHR(27)+CHR(74)
? 'There are '+STR(Any,5)+' entries in the Postfile.'
? 'Do you still want to do the Job Costs (Y or N).'
```

WAIT TO Continue

IF !(Continue) = 'Y'

DO JobCosts

ENDIF

ELSE

DO JobCosts

ENDIF

RELEASE Any

ELSE

IF Action = '2'

DO FindBills

ELSE

IF Action = '3'

ERASE

DISPLAY FILES ON B

?

?

? 'Which file do you want to review?'

ACCEPT TO Database

IF FILE("B:" + DATABASE) > 0

USE B:&Database

DO Review

ELSE

\* Erases IBM 3101 to end of screen

```

@ 17,0 SAY CHR(27)+CHR(74)
```

```

@ 17,0 SAY !(Database) + " isn't on the list, is it? Check ";
+ 'your spelling, then hit <Return>'
```

? 'and try again. Or not, as the case may be.'

WAIT

ENDIF

ELSE

IF Action = '4'

DO SalesTax

ELSE

IF Action = '5'

```
DO JobsIndx
ELSE
  RELEASE All
  RETURN
ENDIF 5
ENDIF 4
ENDIF 3
ENDIF 2
ENDIF 1
ERASE
STORE T TO Reporting
ENDDO Reporting
```

## \*\*\*\*\* JOBCOSTS COMMAND FILE \*\*\*\*\*

\* Provides summaries of costs by client and job number. This can  
 \* also be used to summarize all office categories, since they fall  
 \* into these fields.  
 \* REPORTS ARE BY JOB NUMBER. Client code is used only in the heading.  
 \* The report is actually prepared based on the job number, so accuracy is  
 \* critical.  
 \* This file works with a partially indexed costbase, so "Unindexed" is  
 \* used to keep track of how many records are not in the index. If this gets  
 \* beyond a specific number, the operator is prompted to reindex the Costbase.  
 \*\*\*\*\*

SET TALK OFF

RESTORE FROM B:Constant  
 DO GetDate

STORE 0 TO Unindexed  
 STORE ' ' TO Again  
 DO WHILE !(Again) <> 'F'  
 STORE ' ' TO MClient  
 STORE ' ' TO MJob:Nmbr  
 STORE ' ' TO Hardcopy  
 STORE 'N' TO Number  
 ERASE  
 @ 2,11 SAY ' JOB COST SUMMARY :'  
 @ 4,11 SAY 'ENTER CLIENT CODE ' GET MClient  
 @ 5,11 SAY ' ENTER JOB NUMBER ' GET MJob:Nmbr  
 @ 6,11 SAY ' P to PRINT ' GET Hardcopy  
 @ 7,11 SAY 'SHOW BILL NUMBERS ' GET Number  
 READ  
 ?  
 IF MClient = ' ' .OR. MJob:Nmbr= ' '  
 @ 9, 0  
 ? ' CLIENT or JOB NUMBER missing.'  
 ? ' F if Finished,'  
 ? ' <Return> to continue.'  
 WAIT TO Again  
 ELSE  
 @ 8,0 SAY CHR(27)+CHR(74)  
 ACCEPT ' OPTIONAL JOB DESCRIPTION ' TO Message  
 STORE TRIM!(Message) TO Message  
 STORE !(MClient) TO MClient  
 STORE !(Hardcopy) TO Hardcopy  
 STORE !(Number) TO Number  
 @ 4,30 SAY MClient  
 @ 6,30 SAY Hardcopy  
 @ 7,30 SAY Number  
 @ 9,30 SAY Message  
 ?  
 ?  
 ACCEPT 'Type C to CHANGE any entries' TO Changes  
 IF !(Changes) = 'C'  
 STORE ' ' TO Again  
 ERASE  
 ELSE  
 ERASE  
 IF !(Hardcopy) = 'P'  
 STORE "TO PRINT" TO Hardcopy  
 SET PRINT ON  
 ENDIF Hardcopy  
 IF Number = 'Y'  
 STORE 'Bill #' TO Other

```

ELSE
  STORE CHR(0) TO Other
ENDIF

? $(Date,3,2)+'/'+$(Date,5,2)+'/'+$(Date,1,2)+': COST SUMMARY FOR ';
+ '&MClient-&MJob:Nmbr'
? '
? ' + Message
?
? 'DATE    NAME                DESCRIPTION                AMOUNT';
+ ' &Other'
?
USE B:CostBase INDEX B:$Jobs
IF Number = 'Y'
  STORE ',Bill:Nmbr' TO Other
ELSE
  STORE CHR(0) TO Other
ENDIF

STORE 0 TO Sum
STORE 0 TO HowMany
STORE 0 TO LineCnt
STORE 0 TO Spacer
FIND &MJob:Nmbr
IF # <> 0
  DO WHILE Job:Nmbr = VAL(MJob:Nmbr) .AND. .NOT. EOF
    DISPLAY Next 1 Bill:Date,Name,Descrip+' ',Amount &Other OFF
    STORE Sum + Amount TO Sum
    STORE LineCnt + 1 TO LineCnt
    STORE Spacer + 1 TO Spacer

    IF Spacer = 10
      ?
      STORE 0 TO Spacer
    ENDIF

    IF LineCnt = 50
      ? CHR(12)
      STORE 0 TO LineCnt
      STORE 0 TO Spacer
      ? 'DATE    NAME                DESCRIPTION';
      + '                AMOUNT'
      ?
    ENDIF
    SKIP
  ENDDO
ENDIF

GO TOP
STORE VAL(Name) TO LastReco
USE B:Costbase
STORE 0 TO Unindexed
GO LastReco
SKIP
DO WHILE .NOT. EOF
  DISPLAY Next 1 Bill:Date, Name, Descrip+' ', Amount;
  FOR Job:Nmbr = VAL(MJob:Nmbr) OF:
  IF Job:Nmbr = &MJob:Nmbr
    STORE Sum + Amount TO Sum
    STORE LineCnt + 1 TO LineCnt
    STORE Spacer + 1 TO Spacer

    IF Spacer = 10
      ?
      STORE 0 TO Spacer
    
```



```

      SKIP
    ENDDO
    ?
    ? '
    ? '          SUB-TOTALS : '+ STR(T,9,2) + ' ' ;
    ? ' + STR(S,9,2)+' ' + STR(F,9,2)
    ?
    ? '          TOTAL BILLED TO DATE: ' -;
    ? '          STR(Sum,9,2)

    ? CHR(12)
    SET PRINT OFF

    ? '      F if Finished,'
    ? ' <Return> to continue.'
    WAIT TO Again
    ENDIF okay to do the report
  ENDIF
ENDDO Again

IF Unindexed > 50
  ERASE
  @ 5,0
  ? ' There are ' - STR(Unindexed,9) + ' unindexed records'
  ? ' in the Costbase. To speed up the Job Costs procedure,'
  ? ' please reindex from the next menu.'
  ? '<Return> to continue.'
  WAIT
ENDIF

RELEASE All
RETURN

```

## \*\*\*\*\* JOBSINDX COMMAND FILE \*\*\*\*\*

```

* Indexes the costbase on job numbers to B:Jobs.NDX.
*   The method of indexing here allows us to use the index to help.
* find job numbers for the Job Costs command files, but allows us to
* do so without having to index the Costbase every time we add a bill.
*   The strategy is: before we index the Costbase on job numbers,
* we first store the number of the last record in a record with a job
* number of zero. When the file is indexed, this record is at the top
* of the indexed file ($Jobs) so that we can find it whenever we want to.
*****

```

```

USE B:Costbase
GO BOTTOM
STORE STR(#,5) TO Temp
GO TOP
IF Job:Nmbr = 0
  REPLACE Name WITH Temp.
ELSE
  DO WHILE !(Code) <> 'H'
    ? "Uh, Oh--trouble. Don't touch anything"
    ACCEPT 'and call Hal.' TO Code
  ENDDO
ENDIF

DELETE FILE B:$Jobs.NDX
ERASE
@ 5,0 SAY 'There are ' + Temp + ' records to index.'
SET TALK ON
INDEX ON Job:Nmbr TO B:$Jobs
SET TALK OFF

RELEASE Temp
RETURN

```



## \*\*\*\*\* FINDBILLS COMMAND FILE \*\*\*\*\*

\* This procedure finds specific bills that we are looking for, then allows  
 \* us to edit them.  
 \* The bill can be specified by bill number and/or amount. If you decide  
 \* not to pay a bill that was found specifying more than one item, you will be  
 \* presented the rest of the entries for the supplier based on name only.  
 \*\*\*\*\*

```
SELECT PRIMARY
USE B:CostBase INDEX B:$Supp
```

```
STORE 'N' TO Finished
DO WHILE !(Finished) <> 'F'
```

\* "Entering" controls a closed loop that allows the operator to change  
 \* the entry if he or she spots an error.

```
STORE "C" TO Entering
DO WHILE !(Entering) = 'C'
  ERASE
  @ 4,0
  ACCEPT ' NAME OF SUPPLIER ' TO MName
  ACCEPT ' INVOICE NUMBER ' TO MBill:Nmbr
  ACCEPT ' ENTER AMOUNT ' TO Temp
  STORE !(MName) TO MName
  STORE !(MBill:Nmbr) TO MBill:Nmbr
  STORE VAL(Temp) TO MAmount
  STORE MAmount*1.00 TO MAmount
  @ 6,19 SAY MName
  @ 7,19 SAY MBill:Nmbr
  @ 8,19 SAY MAmount
  @ 11, 0 SAY ' C to CHANGE,'
  ? ' <Return> to continue.'
```

\* OneByOne is used so that we look at the entire listing for a name once  
 \* if we could have started in the middle of the list and the bill is not  
 \* the one we want, we go up to the first listing then go through all the  
 \* entries for the name, one by one. Used in the last loop in this file.  
 IF Bill:Nmbr > ' ' .OR. Amount <> 0

```
  STORE 0 TO OneByOne
ELSE
  STORE 1 TO OneByOne
ENDIF
```

```
  WAIT TO Entering
ENDDO Entering
```

```
STORE T TO Looking
@ 11, 0 SAY "I'M LOOKING, I'M LOOKING!!"
@ 12,0
@ 13,0
```

\* Now look for a match on the first 10 characters of the name. This finds  
 \* the first entry for that supplier, then looks for bill number or amount  
 \* if we specified them. If not specified, it skips through all the entries  
 \* for the name.

```
IF LEN(MName) > 10
  STORE $(MName,1,10) TO Key
ELSE
  STORE MName TO Key
ENDIF
```

```
FIND &Key
```

```

@ 11, 0
IF # = 0
?
? " GEE, I CAN'T FIND THE NAME. Please check the spelling."
? " Or maybe it hasn't been posted to the COSTBASE yet."
? '<Return> to continue.'
WAIT
ERASE
ELSE
* Found at least one entry with a matching name.
STORE T TO Looking
IF MBill:Nmbr = ' ' .AND. MAmount = 0
STORE F TO Looking
ELSE
* If we have more than the name, we first check for the bill number.
IF MBill:Nmbr > ' '
DO WHILE Name=Key .AND. .NOT. EOF .AND. Looking
IF Bill:Nmbr <> MBill:Nmbr
SKIP
ELSE
STORE F TO Looking
ENDIF
ENDDO

* If we're on a new name or the end of the file, Looking is TRUE
* because we have not found the supplier we were looking for.
* Otherwise, we have a matching bill number to confirm.
IF Looking
? ' This BILL NUMBER is not in the costbase.'
? '<Return> to continue.'
WAIT
ENDIF
ELSE

* If no bill number, look for the amount.
DO WHILE Name=Key .AND. .NOT. EOF .AND. Looking
IF Amount <> MAmount
SKIP
ELSE
STORE F TO Looking
ENDIF
ENDDO

* If we're on a new name or the end of the file, Looking is TRUE
* Otherwise, we have an unpaid bill to confirm.
IF Looking
? ' No bill for this amount and this supplier.'
? '<Return> to continue.'
WAIT
ENDIF
ENDIF we have the bill number
ENDIF we have only the name
ENDIF there is an unpaid bill for the supplier

STORE 'N' TO Changing
DO WHILE !(Changing) <> 'Y' .AND. .NOT. Looking
@ 12,0
DISPLAY
? CHR(7)
? '
? ' E to EDIT this record,'
? ' Q to QUIT this supplier,'
ACCEPT ' <Return> to continue.' TO Changing
?

```

```

IF !(Changing) = 'Q'
  STORE T TO Looking
ELSE
  IF !(Changing) = 'E'
    STORE STR(5) TO Found
    EDIT &Number
    ERASE
  ELSE
    * If the first record is not the one we want, we skip through the
    * rest of the entries for the name. We first go on from where we
    * were in the listing (if we had more than the name), then go back
    * to the first entry and look at those we had skipped. If we had
    * only the name, OneByOne = 1 and we go through the list only once.
    SKIP
    IF EOF .OR. Name <> Key
      IF OneByOne = 0
        FIND &Key
        STORE 1 TO OneByOne
      ELSE
        @ 11, 0 SAY CHR(27) + CHR(74)
        ? "We've gone through all the entries for " + MName+'.'
        ? '<Return> to continue.'
        STORE T TO Looking
        WAIT
      ENDIF
    ENDIF we've gone through the list
  ENDIF is it the right record
ENDIF
ENDDO Changing the record
?
? '          F if FINISHED finding bills,'
? '    <Return> to continue.'
? CHR(7)
WAIT TO Finished
ENDDO Finished

```

## \*\*\*\*\* REVIEW.CMD FILE \*\*\*\*\*

\* This is used to list entries in any .DBF file. The database must be named in  
 \* the command file calling the procedure. Records may be listed conditionally,  
 \* with or without the record numbers.  
 \* Records are listed in groups of 10 with a line space between each group.  
 \* Processing can be continuous, or can stop after every group of 10.  
 \* The listing can start on a specified record number.  
 \* The files can be re-listed as many times as desired.  
 \* Printing is optional. The "CHR(X)" commands are for a Diablo 1650  
 \* printer.  
 \*\*\*\*\*

```

STORE 'Y' TO Reviewing
DO WHILE !(Reviewing)='Y'
  COPY STRUCTURE EXTENDED TO Temp
  GO BOTTOM
  STORE STR(5) TO Last
  ERASE
  ?
  ? 'The '+!(Database)+' database has '-Last+' entries. They will be shown'
  ? 'in groups of 10 records, 50 records to a page if printed.'
  ? 'Enter new values for defaults or press <Return>:'
  ?
  ? '*** DISPLAY [Field list] [FOR <expression>] [OFF] ***'
  ?
  STORE 1 TO First
  STORE 1 TO PageCnt
  STORE VAL(Last) TO RecoCnt
  STORE 'N' TO Pause
  STORE 'N' TO Partial
  STORE 'N' TO Conditions
  STORE 'N' TO Tally
  STORE 'C' TO Changing
  DO WHILE !(Changing) = 'C'
    @ 8,10 SAY 'START ON RECORD NUMBER ' GET First
    @ 9,10 SAY ' STOP ON RECORD NUMBER ' GET RecoCnt
    @ 10,10 SAY ' START PAGE NUMBERS ON ' GET PageCnt
    @ 11,10 SAY 'PAUSE EVERY 10 RECORDS ' GET Pause
    @ 12,10 SAY ' SHOW SELECTED FIELDS ' GET Partial
    @ 13,10 SAY 'DISPLAY FOR EXPRESSION ' GET Conditions
    @ 14,10 SAY ' SHOW RECORD NUMBERS ' GET Tally
    ?
    ? ' C to CHANGE the defaults,'
    ? ' <Return>.to continue.'
    WAIT TO Changing
  IF !(Changing) = 'C'
    * Clear to end of screen on IBM 3101
    @ 15,0 SAY CHR(27)+CHR(74)
    READ
  ELSE
    IF First > VAL(Last) .OR. First <= 0 .OR. RecoCnt > VAL(Last);
      .OR. RecoCnt <= 0
      @ 15,0 SAY CHR(27)+CHR(74)
      @ 16,0 SAY 'Sorry, wrong number: '-!(Database)+' contains '+';
        'records 1 through'+Last+'.'
      ? '<Return> to correct your entry.'
      WAIT
      @ 15,0 SAY CHR(27)+CHR(74)
      STORE 'C' TO Changing
      STORE 1 TO First
      STORE VAL(Last) TO RecoCnt
    ENDIF
  ENDIF

```

```

ENDIF
* Clears to end of screen on IBM 3101
@ 15,0 SAY CHR(27)+CHR(74)
ENDDO
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
IF !(Partial)= 'Y'
@ 11,0 SAY-CHR(27)+CHR(74)
@ 11,0 SAY 'The '+!(Database)+' database consists of these FIELDS:'
USE Temp
?
STORE ' ' TO Choices
DO WHILE .NOT. EOF
    STORE Choices+TRIM(Field:Name)+' , ' TO Choices
    SKIP
ENDDO
STORE $(Choices,2,LEN(Choices)-3) TO Choices
STORE 'Y' TO Unfinished
DO WHILE !(Unfinished) = 'Y'
@ 13, 0 SAY Choices

USE B:&Database
?
? 'List FIELDS to display (<return> to show all).'
?
ACCEPT '          DISPLAY ' TO Partial
STORE !(Partial) TO Partial
STORE Partial TO String
STORE LEN(String) TO Size

IF Size =0 .OR. (Size=1 .AND. Partial=' ')
    STORE CHR(0) TO Partial
    STORE 'N' TO Unfinished
ELSE
?
? 'Want to change it (Y or N)?'
WAIT TO Unfinished
IF !(Unfinished) = 'Y'
    @ 12, 0 SAY CHR(27) + CHR(74)
ELSE
@ 10,0 SAY CHR(27) + CHR(74)
? '*** Checking fields ['+Partial+'] : '
?
STORE 0 TO F
STORE 0 TO Counter
DO WHILE Size >0
    STORE Counter + 1 to Counter
    ?? ' *'+STR(Counter,2)
    STORE @(',',, String) TO Mark
    IF Mark = 1 .OR. Mark = Size
        ? 'Uh, oh--trouble: comma cannot be at the ' ;
        +'start or end of a list of values.'
    
```



```

?
? 'DISPLAY &Partial FOR '
ACCEPT TO Expression
?
? 'Do you want to change the expression (Y or N)?'
WAIT TO Unfinished
ENDDO

IF Expression > '
  STORE 'FOR '+ Expression TO Conditions
ELSE
  STORE CHR(0) TO Conditions
ENDIF
ELSE
  STORE CHR(0) TO Conditions
ENDIF

IF!(Tally) <> 'Y'
  STORE 'OFF' TO Tally
ELSE
  STORE CHR(0) TO Tally
ENDIF

STORE [DISPLAY Next 1 &Partial &Conditions &Tally] TO Command
@ 11, 0 SAY CHR(27)+CHR(74)
@ 11, 0 SAY '*** '+[DISPLAY &Partial &Conditions &Tally]+' ***'
?
? 'is the command that will be performed on the '+!(Database)+' database.'
? '      C to CHANGE *t,'
? '      Q to QUIT with no action,'
? '      <Return> to review the database.'
WAIT TO Abort

IF !(Abort) = 'Q'
  STORE CHR(0) TO Reviewing
ELSE
  IF !(Abort) <> 'C'
    ERASE
    ? 'Enter a one-line heading or press <Return> to skip.'
    ACCEPT TO Message
    STORE !(Message) TO Message
    ?
    STORE 0 TO Count
    STORE 0 TO PageMark
    STORE STR(First,5) TO Number
    GO &Number

    ERASE
    ? 'Do you want to print the listing now (Y or N)?'
    ACCEPT TO Hardcopy

    IF !(Hardcopy)='Y'
      SET PRINT ON
      DO RevMrgn
    ENDIF

    ERASE
    ? Message
    ? 'Page '+ STR(PageCnt,3)

    IF Tally = 'OFF'
      ?? ' starts on Record #'-STR(#,5)
      ?
      IF .NOT.( Partial > ' ' .OR. Conditions > ' ')

```

```

DO RevHdr
ENDIF
ENDIF
?
DO WHILE .NOT. EOF .AND. # <= RecoCnt
&Command

IF !(Conditions) > CHR(0)
IF &Expression
STORE (Count + 1) TO Count
ENDIF
ELSE
STORE (Count + 1) TO Count
ENDIF
ENDIF
SKIP

IF Count=10
STORE 0 TO Count
* Inserts a space every ten records, then waits. The printer
* is turned off so that "WAIT" does not print on the hardcopy.
?
SET PRINT OFF
IF !(Pause) = 'Y'
WAIT
ENDIF

IF !(Hardcopy) = 'Y'
SET PRINT ON
ENDIF

* The following routine prints 50 entries to a page,
* then moves to the next page and prints a heading

-STORE (PageMark + 1) TO PageMark
IF PageMark = 5
? CHR(12)
STORE (PageCnt + 1) TO PageCnt

IF .INT(PageCnt/7) = PageCnt/7
?
ENDIF

? Message
? 'Page '+STR(PageCnt,3)

IF Tally = 'OFF'
?? ' starts on Record #' -STR(5)
?
IF .NOT.( Partial > ' ' .OR. Conditions > ' ' )
DO RevHdr
ENDIF
ENDIF
?
STORE 0 TO PageMark
ENDIF
ENDIF
ENDDO

* Formfeed on Diablo 1650 printer
? CHR(10)
SET PRINT OFF
SET RAW ON
SET MARGIN TO 38
? 'Do you want to see the '+!(Database)+' again (Y or N)?'

```



```
    WAIT TO Reviewing
  ELSE
    STORE 'Y' TO Reviewing
  ENDIF
ENDIF
?
ENDDO Reviewing

USE
DELETE FILE Temp
RELEASE All
RETURN
```

\*\*\*\*\* REVHDR COMMAND FILE \*\*\*\*\*  
 \* Used by Review.CMD to print headings for different database listings.  
 \*\*\*\*\*

```

IF !(Database)='INSERTS'
  ? 'IO# MAGAZINE      ISSUE  JOB  AD          SPACE
    '  GROSS      NET  X  DATE'
ELSE
  IF !(Database)='BILLINGS'
    ? 'INV# JOB      DATE  TAXABLE      TAX  NO:TAX  PO#  DESCRIPTION'
  ELSE
    IF !(Database)='INVOICES'
      ? 'INV#  CLT  DATE  TAXABLE      TAX  NO:TAX      '+';
        'TOTAL  AMT:RCD  DATE'
    ELSE
      IF !(Database)='COSTBASE'
        ? 'DATE  CHECK  JOB      AMOUNT  NAME      '+';
          'DESCRIPTION      DATE  BILL#  HOURS  EMP'
      ELSE
        IF !(Database)='DEPOSITS'
          ? 'DATE  RECEIVED FROM      CHECK      AMOUNT  '+';
            'INV#  COMMENTS'
        ENDIF
      ENDIF
    ENDIF
  ENDIF
ENDIF
ENDIF
ENDIF
ENDIF

```

```
***** REVMRGN COMMAND FILE *****  
* Used by Review.CMD to set margins for different database listings.  
*****
```

```
IF !(Database)='INSERTS'  
  SET MARGIN TO 38  
ELSE  
  IF !(Database)='COSTBASE'  
    SET MARGIN TO 36  
  ELSE  
    SET MARGIN TO 45  
  ENDIF  
ENDIF  
RETURN
```

```
***** SALES TAX COMMAND FILE *****
* This file summarizes the invoice file for a specified period.
* It shows the invoices and the type of billing (taxable or
* service) along with the totals for the two types and the total
* sales tax liability for the period.
* It also includes materials and equipment subject to a use tax
* that has not been paid. These are entered in the invoices database
* when they come in as well as in the Postfile.
*****
```

USE B:Invoices

ERASE

```
? 'This file summarizes the data you need to prepare the End-of-Quarter'
? 'report to the State Board of Equalization for SALES TAX collected by'
? 'the agency. It includes use tax on materials bought out of state or'
? 'bought with our resale number without paying a use tax.'
```

STORE 'C' TO Dating

DO WHILE !(Dating) = 'C'

STORE 'YMMDD' TO Start

STORE 'YMMDD' TO Finish

@ 7, 0 SAY 'This summary is for the period FROM ' GET Start

@ 7,45 SAY ' TO ' GET Finish

READ

@ 9,0 SAY ' '

? ' C to CHANGE,'

? '<Return> to continue.'

WAIT TO Dating

@ 7,0

? CHR(27) + CHR(74)

ENDDO Dating

ERASE

@ 5,10 SAY '\*\*\*\*\* DO NOT INTERRUPT \*\*\*\*\*'

@ 7,10 SAY 'COMPUTING THE QUARTERLY SALES TAX REPORT'

?

COPY TO Temp FIELDS Inv:Nmbr, Inv:Date, Taxable, Sales:Tax, TaxFree, Amount;

FOR Inv:Date >= Start .AND. Inv:Date <= Finish'

USE Temp

SORT ON Inv:Nmbr TO Temp2

USE Temp2

REPLACE Inv:nmbr WITH ' USED' FOR VAL(Inv:Nmbr) < 1000

STORE \$(Start,3,2)+'/'+\$(Start,5,2)+'/'+\$(Start,1,2) TO Start

STORE \$(Finish,3,2)+'/'+\$(Finish,5,2)+'/'+\$(Finish,1,2) TO Finish

@ 5,0

SET MARGIN TO 45

SET PRINT ON

STORE 1 TO PageCnt

? 'SALES TAX SUMMARY FROM '+Start+' TO '+Finish+' Page '+STR(PageCnt,3)

?

? 'INV# DATE TAXABLE TAX SERVICE TOTAL'

?

STORE 0 TO Count

STORE 0 TO PageMark

GO TOP

DO WHILE .NOT. EOF

DISPLAY Inv:Nmbr, Inv:Date, Taxable, Sales:Tax, TaxFree, ' '+STR(Amount,9,2) OFF

STORE (Count + 1) TO Count

SKIP

IF Count=10

STORE 0 TO Count

\* Inserts a space every ten records, then waits. The printen

```

* is turned off so that "WAIT" does not print on the hardcopy.
?
* The following routine prints 50 entries to a page,
* then moves to the next page and prints a heading

STORE (PageMark + 1) TO PageMark
IF PageMark = 5
  STORE 0 TO PageMark
  ? CHR(12)
  STORE (PageCnt + 1) TO PageCnt

  * Compensates for an offset caused by the 7 lines/inch printing
  IF INT(PageCnt/7) = PageCnt/7
    ?
  ENDIF

  ? 'SALES TAX SUMMARY FROM ' + Start + ' TO ' + Finish+' : Page ' + ;
    STR(PageCnt,3)
  ?
  ? 'INV#    DATE    TAXABLE    TAX    SERVICE    TOTAL'
  ?
  ENDIF
ENDIF
ENDDO
?
SET PRINT OFF
?
? '          COMPUTING TOTALS NOW.'
?
REPLACE All Inv:Nmbr WITH ' ' FOR VAL(Inv:Nmbr) > 1000
TOTAL ON Inv:Nmbr TO Other
USE Other
REPLACE All Inv:Date WITH 'TOTAL'
REPLACE All Inv:Nmbr WITH 'SALES' FOR Inv:Nmbr = '
SUM Taxable TO Used FOR Inv:Nmbr = ' USED'
SUM Amount TO Sold
STORE Sold + Used TO Gross
SUM Sales:Tax TO Collected
SUM TaxFree TO Service
STORE Collected + Service TO Exempt
STORE Gross - Exempt TO Subject
STORE 0.06*Subject + 0.005 TO Payable

* Print totals of all the invoices
GO TOP
SET PRINT ON
DO WHILE .NOT. EOF
  DISPLAY Inv:Nmbr,Inv:Date,Taxable,Sales:Tax,TaxFree,' '+STR(Amount,9,2) OFF
  STORE Count + 1 TO Count
  SKIP
ENDDO
IF PageMark > 3
  * Formfeed if not enough room to print the following list
  ? CHR(12)
ENDIF
?
?
?
? 'ENTER THE FOLLOWING DATA ON THE BOARD OF EQUALIZATION FORM:'
?
* The following segment is not the final, but the state auditor is in right now
* and I've got to get the info out to him and to the state for this month.
* The final version will include all lines in the form, to allow for changes
* in the way we do our business. Obviously, this is also the place to
* print the form if you want to do that. Since the form is used only once

```

\* every three months, we won't automate it entirely.

```
? '   LINE 1>   TOTAL GROSS SALES: ' + STR(Sold,9,2)
? '   LINE 2>   SUBJECT TO USE TAX: ' + STR(Used,9,2)
? '   LINE 3>   TOTAL TRANSACTIONS: ' + STR(Gross,9,2)
?
? '   LINE 9>   SALES TAX INCLUDED: ' + STR(Collected,9,2)
? '   LINE 10>  ADVERTISING SERVICES: ' + STR(Service,9,2)
? '   LINE 11>  TOTAL EXEMPTIONS: ' + STR(Exempt,9,2)
? '   LINE 12>  SUBJECT TO STATE TAX: ' + STR(Subject,9,2)
? '   LINE 13>  AMOUNT OF STATE TAX: ' + STR(0.05*Subject+0.005,9,2)
? '   LINE 14>  SUBJECT TO LOCAL TAX: ' + STR(Subject,9,2)
?
? '   LINE 19>  AMOUNT OF LOCAL TAX: ' + STR(0.01*Subject+0.005,9,2)
?
? '   LINE 21>          TOTAL TAXES: ' + STR(Payable,9,2)
?
? '   LINE 28> TOTAL DUE AND PAYABLE: ' + STR(Payable,9,2)
? CHR(12)
SET MARGIN TO 38
SET PRINT OFF
```

```
RELEASE All
USE
DELETE FILE Temp
DELETE FILE Temp2
DELETE FILE Other
RETURN
```

\*\*\*\*\* TIMECALC COMMAND FILE \*\*\*\*\*  
 \* Verifies that employee name and number match, then  
 \* calculates billing charges for employee time.  
 \*\*\*\*\*

SET TALK OFF  
 ERASE  
 SELECT PRIMARY  
 RESTORE FROM B:Constant

GO TOP  
 DO WHILE .NOT. EOF

ERASE  
 @ 4,20 SAY ' \*\* DO NOT INTERRUPT \*\*  
 @ 5,20 SAY ' PROCESSING TIME CHARGES

IF \*.OR. Job:Nmbr = 31 .OR. Check:Nmbr <> '---'  
 SKIP

ELSE  
 REPLACE Client WITH !(Client),Name WITH !(Name)  
 STORE STR(#,4) TO Number  
 @ 7,20 SAY ' Record # '+Number  
 @ 8,20 SAY ' '+Name  
 ? CHR(7)  
 IF Emp:Nmbr<=0 .OR. Emp:Nmbr>MaxEmpl .OR. Hours = 0

ERASE  
 REPLACE Hours WITH Hours\*1.00  
 REPLACE Emp:Nmbr WITH Emp:Nmbr\*1  
 @ 4,0 SAY ' '

DISPLAY  
 @ 6,3 SAY 'HOURS='  
 @ 6,18 SAY '=EMPLOYEE NUMBER'

?  
 ? 'Press ANY KEY to correct the EMPLOYEE NUMBER,'  
 ? 'or press H to correct the HOURS.'

WAIT TO Decision  
 IF !(Decision) <> 'H'  
 @ 6,14 GET Emp:Nmbr

ELSE  
 @ 6,8 GET Hours

ENDIF  
 READ

ELSE  
 SELECT SECONDARY  
 USE B:Personne  
 STORE T TO Looking  
 DO WHILE Looking .AND. .NOT. EOF  
 IF \$(Name,1,10)=\$(P.Name,1,10)

IF Emp:Nmbr=P.Emp:Nmbr  
 SELECT PRIMARY  
 \* Formula optimistically assumes 65 billable hours out  
 \* of 75 hours possible in two weeks. Eff. mult.=3.23  
 REPLACE Amount WITH Pay:Rate\*2.8\*Hours/65  
 SELECT SECONDARY  
 STORE F TO Looking

ELSE  
 SELECT PRIMARY  
 STORE T TO Fixing  
 DO WHILE Fixing  
 ERASE  
 @ 4,0 SAY ' '  
 DISPLAY  
 @ 6,16 SAY '=EMPLOYEE NUMBER'  
 ?

```

? 'The correct Employee Number is'
?? S.Emp:Nmbr
?? ' For '+S.Name
? 'Press ANY KEY to change the EMPLOYEE NUMBER'
? 'press N to change the NAME.'
WAIT TO Choice
IF !(Choice) <> 'N'
    @ 6,12 GET Emp:Nmbr
    READ
    STORE F TO Fixing
ELSE
    @ 5,25 GET Name
    REPLACE Name WITH !(Name)
    READ
    STORE F TO Fixing
ENDIF Employee number
ERASE
ENDDO Fixing
SELECT SECONDARY
GO TOP
ENDIF Numbers match
ELSE
    SKIP
ENDIF
IF EOF
    ERASE
    SELECT PRIMARY
    @ 4,0 SAY ' '
    DISPLAY
    @ 6,16 SAY '-EMPLOYEE NUMBER'
    ?
    ? 'This name is not listed in the Personnel file,'
    ? 'so time charges were not calculated.'
    ? 'Press any key to change the name, or write the'
    ? 'record number down and press D to DELETE.'
    WAIT TO Change
    IF !(Change)<> 'D'
        @ 5,25 GET Name
        REPLACE Name WITH !(Name)
        READ
        SKIP-1
    ELSE
        ERASE
        DELETE
        DISPLAY
        ?
        ? 'THIS RECORD HAS BEEN DELETED.'
        WAIT
        ENDIF Change
    SELECT SECONDARY
    ENDIF no name
ENDDO Looking
SELECT PRIMARY
SKIP
ENDIF
ENDIF deleted
NDDO billing calculations

RELEASE All
RETURN

```



## \*\*\*\*\* PRINTOUT COMMAND FILE \*\*\*\*\*

\* This file is used by several other command files. It prints out a  
 \* listing of the records in a file without the record number. The  
 \* output is spaced every 10 records and the printer is positioned back  
 \* at the left margin after the printout.  
 \* The calling command file determines where the printout starts by  
 \* specifying a value for the variable "Number".  
 \* This does not show the record numbers. To do so, use the  
 \* Review.Cmd file.  
 \*\*\*\*\*

```

IF VAL(Number) > 0
  GOTO RECORD &Number
ELSE
  GO TOP
ENDIF

STORE 0 TO Count
DO WHILE .NOT. EOF
  IF *
    SKIP
  ELSE
    DISPLAY &Condition
    SKIP
    STORE Count+1 TO Count
    IF Count=10
      STORE 0 TO Count
      * Spaces one line every 10 records, then waits. Turns the printer
      * off so that "WAIT" does not print.
      ?
      SET PRINT OFF
      WAIT
      IF !(Output)= 'Y'
        SET PRINT ON
      ENDIF
    ENDIF
  ENDIF
ENDIF

ENDDO
* The next 2 lines reposition the printer at the
* left margin.
?
SET PRINT OFF

RELEASE Count, Output
RETURN

```

## \*\*\*\*\* GETDATE COMMAND FILE \*\*\*\*\*

\* Confirms that the date is entered as YYMMDD by checking to see that  
 \* the entries for each item are in the correct range. The year is  
 \* checked against a constant stored in the B:Constant.MEM file.  
 \*\*\*\*\*

```

STORE "T" TO NoDate
DO WHILE I(NoDate) <> 'F'
  ERASE
  STORE 'YYMMDD' TO Date
  @ 5,10 SAY "Enter TODAY'S date" GET Date
  ? CHR(7)
  READ

  IF VAL($(Date,1,2)) <> ThisYear;
    .OR. VAL($(Date,3,2)) < 1 .OR. VAL($(Date,3,2)) > 12;
    .OR. VAL($(Date,5,2)) < 1 .OR. VAL($(Date,5,2)) > 31
    @ 10,25 SAY 'DATE ERROR'
    STORE 0 TO X
    DO WHILE X < 50
      STORE X + 1 TO X
    ENDDO
  ELSE
    ?
    @ 10,0 SAY ' C to CHANGE the date,'
    ? '<Return> to continue.'
    WAIT TO Change
    IF !(Change) <> 'C'
      STORE 'F' TO NoDate
    ENDIF
  ENDIF
ENDDO NoDate

RELEASE NoDate, Change, X
RETURN

```

\*\*\*\*\* DATETEST COMMAND FILE \*\*\*\*\*

\* This file verifies the Bill:Date and Check:Date to see that they are  
 \* in the right format. If incorrect, the operator may edit them.  
 \*\*\*\*\*

ERASE  
 GO TOP

\* The variable DATE brings in the NAME of the date field to be checked  
 \* from the command files where this is used.  
 DO WHILE .NOT. EOF

  @ 6,30 SAY ' VERIFYING '+Date+' '

IF \*  
 SKIP  
 ELSE

  IF &Date <> '  
 STORE STR(#,5) TO Found  
 STORE T TO NoDate  
 DO WHILE NoDate

    @ 8,30 SAY ' RECORD '+Found  
 @ 9,30 SAY ' '+&Date,1,2)+'/'+&Date,3,2)+'/'+&Late,5,2)  
 ? CHR(7)

    \* The macro symbol is used to get the contents of the date field  
 \* being checked without creating a new variable.

  IF VAL(\$(&Date,1,2)) > ThisYear .OR. VAL(\$(&Date,1,2)) < MinYear;  
     .OR. VAL(\$(&Date,3,2)) < 1 .OR. VAL(\$(&Date,3,2)) > 12;  
     .OR. VAL(\$(&Date,5,2)) < 1 .OR. VAL(\$(&Date,5,2)) > 31

    ?  
 ?  
 ? '

  DATE ERROR: Must be YYYYMMDD '

    ACCEPT 'Enter new Date' TO Temp  
 REPLACE &Date WITH Temp  
 ERASE

  ELSE  
 STORE F TO NoDate  
 SKIP

ENDIF  
 ENDDO NoDate  
 RELEASE Temp, NoDate

ELSE  
 SKIP  
 ENDIF date is not blank

  \* Delay to allow date being checked to be read (quickly)  
 STORE 0 TO X  
 DO WHILE X < 5  
 STORE (X + 1) TO X  
 ENDDO

ENDIF deleted or posted  
 ENDDO  
 RELEASE All  
 RETURN

```

***** NAMETEST COMMAND FILE *****
* Checks names in the file in USE against the Suppliers file and gives
* the operator the options of editing, adding them to the Suppliers file
* or ignoring them. If a name is edited, it is presented again.
*****
GO TOP
DO WHILE .NOT. EOF
  IF *
    SKIP
  ELSE
    STORE STR(#,5) TO Number
    STORE !(Name) TO Name
    ERASE
    @ 4,25 SAY 'CHECKING NAMES '
    @ 6,25 SAY 'RECORD '+Number
    @ 7,25 SAY Name
    ? CHR(7)
    STORE $(Name,1,10) to Key
    SELECT SECONDARY
    USE B:Supplier INDEX B:Supplier
    FIND &Key
    STORE T TO Again
    STORE 'T' TO Decision
    IF # = 0
      DO WHILE Again
        @ 9,20 SAY 'THIS SUPPLIER NAME IS NOT IN THE SUPPLIERS FILE.
        @ 11,20 SAY '      E to EDIT it.
        @ 12,20 SAY '      A to ADD it to the SUPPLIERS file.
        @ 13,20 SAY '      C to CONTINUE.
        ?
        WAIT TO Decision
        IF !(Decision) = 'A'
          APPEND
          SKIP-1
          REPLACE Name WITH !(Name),Address WITH !(Address),City WITH;
            !(City)
          STORE F TO Again
        ELSE
          IF !(Decision) = 'E'
            SELECT PRIMARY
            EDIT &Number
            REPLACE Name WITH !(Name)
            SELECT SECONDARY
            STORE F TO Again
          ELSE
            IF !(Decision) = 'C'
              STORE F TO Again
            ELSE
              STORE T TO Again
            ENDIF C
          ENDIF E
        ENDIF A
      ENDDO
    ENDIF 0
    SELECT PRIMARY
    IF !(Decision) <> 'E'
      SKIP
    ENDIF
  ENDIF deleted
ENDDO
RELEASE All
RETURN

```

```

***** CHECKSTUB COMMAND FILE *****
* Prints out check numbers, amounts, and balances from the CheckFile when
* SALARIES and BILLS are paid. When more than one bill is paid by a
* single check, the program totals all the bills against that check if they
* are entered in consecutive order (which they are in the two command files)
* Records are marked for deletion, but can be reviewed and retrieved.
*****

```

```

ERASE
@ 5,10 SAY '*** DO NOT INTERRUPT ***'
@ 6,10 SAY ' JUST GETTING ORGANIZED'

```

```

USE B:Checkfil
TOTAL ON Check:Nmbr TO Scratch FOR .NOT. *
COUNT FOR .NOT. * TO Entries

```

```

USE Scratch
UPDATE FROM B:Checkfil ON Check:Nmbr REPLACE Balance
COUNT FOR .NOT. * TO Checks

```

```

IF Entries > Checks

```

```

  USE B:Checkfil

```

```

  ERASE

```

```

  @ 4,0 SAY ' '

```

```

  SET PRINT ON

```

```

  ?

```

```

  THESE INDIVIDUAL BILLS WERE PAID:'

```

```

  ?

```

```

  ?

```

```

  Date Check Name Amount

```

```

  ?

```

```

  LIST ' '+Check:Date, Check:Nmbr, Name, Amount, Bill:Nmbr OFF;

```

```

  FOR .NOT.*

```

```

  ?

```

```

  ?

```

```

  ?

```

```

ENDIF

```

```

USE Scratch

```

```

STORE 'Y' TO Doing

```

```

DO WHILE !(Doing)='Y'

```

```

  ERASE

```

```

  SET PRINT ON

```

```

  ?

```

```

  MAKE THE FOLLOWING ENTRIES IN THE CHECK BOOK:'

```

```

  ?

```

```

  ?

```

```

  Date Check Name Amount Balance'

```

```

  ?

```

```

  ?

```

```

  ?

```

```

  ?

```

```

  ?

```

```

  ?

```

```

  ?

```

```

  ?

```

```

  ?

```

```

  ?

```

```

  ?

```

```

  ?

```

```

  ?

```

```

  ?

```

```

  ?

```

```

  ?

```

```

  ?

```

```

  ?

```

```

  ?

```

```

  ?

```

```

  ?

```

```

  ?

```

```

  ?

```

```

  ?

```

```

  ?

```

```

  ?

```

```

  ?

```