

Files Contain Records

Records Contain Data

CC EQUATES
CC COMMON

TCL ↑ arrow to view previous commands.

;CT FILENAME * (All records)

ED T=top, P=Print lines, R=Replace, FI file

;COPY FILENAME [FILEID

* Allowed

TO: RECORD (FILENAME (IOD

I=internal

O=overwrite

D=delete original

UL=Upper/Lower

COMO

Use COMO to start or stop copying terminal output to a record in the &COMO& file. You can also use COMO to print records from the &COMO& file, delete them from the system, or list their names.

SYNTAX

COMO [action]

QUALIFIERS

action can be one of the following:

ON record [HUSH]

Creates a COMO record and starts copying terminal output to it.

HUSH suppresses terminal display. Output is copied to record but is not displayed on the screen. Be careful. Because prompts for input are not displayed, your program or paragraph can wait indefinitely for input.

OFF Stops copying terminal output to the COMO record.

DELETE {record|*} Deletes a COMO record. Asterisk (*) clears the &COMO& file.

LIST Lists COMO records in your account.

SPOOL record [T] Prints a COMO record on the line printer. T displays a COMO record on the screen.

DESCRIPTION

COMO is short for command output.

If you use COMO without any options, COMO prompts for the necessary information.

COMO records are records in a type 1 file named &COMO&. If you create a COMO record by using COMO ON, the COMO command creates an &COMO& file if one does not exist in your account.

In a COMO ON statement, if you specify the name of an &COMO& record that exists, the old contents of the record are overwritten. The records produced by COMO can be quite large. Remember to delete COMO records when you no longer need them (use COMO DELETE record).

The TANDEM command cannot capture output directed to an &COMO& file.

EXAMPLES

In the following example, COMO is used with no options. It establishes a COMO record of the terminal session. Everything that appears on the terminal is stored in the record SESSION1 in the file &COMO& until the command COMO OFF is issued. Terminal display is on, so output appears on the screen at the same time as it is copied to the COMO record.

```
>COMO
Enter action (ON, OFF, DELETE, LIST, SPOOL, QUIT) = ON
Enter COMO file name = SESSION1
Terminal display on? (Y/N) = Y
```

The next example lists all COMO records in the &COMO& file. The COMO record SPOOLLIST is deleted, and the user is prompted for the next action. QUIT returns the user to the system prompt.

```
>COMO LIST
COMO file listing 02:22:49pm 02 Jul 1995

01 SPOOLLIST
02 FILESTATS
03 DICTVOC
Enter number of file to be selected = 1
Enter action (ON, OFF, DELETE, LIST, SPOOL, QUIT) = DELETE
COMO SPOOLLIST deleted.
Enter action (ON, OFF, DELETE, LIST, SPOOL, QUIT) = QUIT
>
```

^ - special characters not drawn.

SP	System Programs
BP	Basic Programs
FBP	Field fixed Basic program – temporary will become part of a new release.
UBP	User Specific
CBP	Customer/Client edit
(7.0.3.0 BP/SP all same)	
OC	Object code
OCO	Machine code

Editor

Shift + F8	*-----*
Shift + F7	***
Shift + F9	Universe Help
Shift + F10	Subroutine
Shift + F11	Subroutine search
Alt+C Alt+C	Beginning and end of block
Alt+P	Paste
Alt+M	Move
Alt+Delete	
Alt+Insert	
Alt+L	Locate
Alt+S	Search from cursor
Alt+R	Replace

@(-4) clear line
 @(-1) clear screen
 @(-3) clear from cursor to end of screen

“L#15”

Common Variables

MENU.DRIVER – most others are subroutines

2 types

Process Specific
 Stack Level
 Subroutine

Named Common	Global All baby processes
Unnamed Common	Public – Baby Processes

CC COMMON Top Named
 Middle Unnamed

\$ at end is new format for common variables.

“EXEC.PGM” Protects common Variables.

EXECUTE *commands* [CAPTURING variable]

Spawns a child process to perform a

Command in TCL

EXECUTE 'SELECT PRODUCT'

- Returns control

- 1) Spawn

- 2) Executes

- 3) Return

CHAIN command

- 1) Spawns a new process

- 2) Terminates current process

- 3) Executes

Hashed Files

18 – Best type

Groups – Locked by group



Size: $(512 * \text{Separation} * \text{Modulo})$

Modulo:

Groups Default 512 Bytes (only 500 available)

Should be a prime number (;PRIME 1000)

Separation:

Size of Groups

Type:

Hash Algorithm

GROUP.STAT

HASH.HELP

LIST is GROUP ORDER

@ID L justification or R justification changes sort order.

Arrays

Dynamic – string with delimiter.

MY='JASON' no <1> → initialized

MY<2> <ATTRIB>

MY<3>

REC<-1> Append to end.

CHAR(255) RECORD
CHAR(254) AM
CHAR(253) VM
CHAR(252) SVM
CHAR(251) SSVM
↓

CHAR	#	Character
AM	254	□
VM	253	z
SVM	252	n
SSVM	251	√
SSSVM	250	l

Dimensioned

Typed

DIM NAME(SIZE)

MAT

CONVERT

FILES

OPEN FIEL TO FILEHANDLE THEN/ELSE

READ DYNARRAY FROM FILEHANDLE,REC.ID THEN/ELSE

WRITE DYNARRAY TO/ON FILEHANDLE, REC.ID THEN/ELSE

READV DYNARRAY FROM FILEHANDLE,REC.ID,ATTRIB, THEN/ELSE

WRITEV DYNARRAY TO/ON FILEHANDEL,REC.ID,ATTRIB THEN/ELSE

} 1 or 2 times

READVU

READU – Locks record

RELEASE FILEHANDLE REC.ID

WRITE

WRITEV

WRITEU

WRITEVU

} Releases Record
} Keeps Lock

MATREAD(V) DIM.ARRAY FROM FILEHANDLE,REC.ID THEN/ELSE

MATWRITE(V) DIM.ARRAY TO/ON FILEHANDLE,REC.ID THEN/ELSE

READLIST DYNARRAY ELSE DYNARRAY=

READNEXT DYNARRAY

NO.RECS=DCOUNT(DYNARRAY,AM)

STRING := -Append to string.

SORT DATA:

- 1) SELECT COMMAND
- 2) LOCATE
- 3) TEMPFILE

LOCATE *STRING* IN DYN.ARRAY BY 'AL'/'DR' SETTING VAR THEN/ELSE

INSERT(ARRAY,POS;NAME)

Screens

Report title on it's own line.

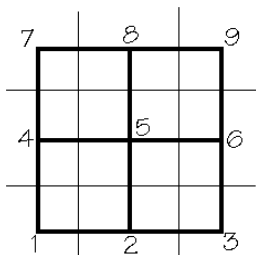
Maintenance screen title on top.

Name: almost always same as routine opening.

Program Editor:

SCREEN file.

S in program editor.



0 —

. |

Ctrl + Alt Double line

Ctrl + . (don't want to hit Ctrl + Alt + Del)

Alt + Key Double horizontal, single vertical.

Alt + Key Double vertical, single horizontal.

SCREEN 'NAME' Name not needed if same as subroutine.

WINDOW ,SCRN='DMM' instead of WINDOW ,,,,,,,DMM

INP

VARIABLE,HOR,VER,LENGTH,CNVR,STPOS,AUTORETURN,DFLT,VERIFY,TERMCHARS,HELPSCR,OUTSTRING

VARIABLE = User input

LABEL

INPNM:

HELP: SCRNLABEL

Divide line on INP (SHIFT+F8)

COMMON VARIABLE – MOVE

0 – Error

1 – Left

2 – Up

3 – Right

4 – Down

5 – Enter

Esc

F12

Mouse click

Pushing Level

Hot Key

On Move+1 goto , , , ,

COMMON VARIABLE – QUIT

0 – Not F12 or Esc

1 – Was F12 or Esc

COMMON VARIABLE – F12

0 – Not F12

1 – Was F12

CONFIRM.ABORT SURE

IF NOT()

CNVR

'MCU'

Upper case

'MCT'

Title

'MCL'

Lower case

'YN'

Yes/No 1/0

'D4/'

Date – F10 pulls colander

'VD4/'

Variable Dating

'TFILENAME;X;FIELD;FIELD'

'R#'

Number of decimal places.

VERIFY C: Control File
 F: File
 S: Subroutine
 D: Date
 Nothing – File

'C:VAILID.STATE'

'F:PRICE.LINE,1' 'F:PRICE.LINE,NEW'

'S:VERF.PRD.ID'

'S:VERF.CUS.ID'

'S:VERF.BT.ID'

'S:VERF.UD.-----' For user defined subroutines.

D:'X'

INP STATE,9,4,2,'MCU,V_ C:VALID.STATES' (Shortcut to parameter)

SUBROUTINE(X,Y,LGTH,ID,OPS,ORIG.VAL)

COMMON VARIABLE CHANGED 0 – No Change

 1 – Changed

COMMON VARIABLE HELP 0 – F10 Not Pressed

 1 – F10 Pressed

UT.OPEN.FILE FILENAME,FILE.HNDL,ERR.MSG,NO.ERR.DISP,HNDL.NUMBER

IF ERR.MSG THEN

 ID=ORIG.VAL

 RETURN

END

IF NOT(CHANGED) AND NOT(HELP) THEN

 ID=ORIG.VAL

 RETURN

END

IF ID[1,1]='.' THEN

 READY TST FROM FILEHNDL,ID[2,999],0 THEN

 ID=ID[2,999]

 RETURN

 END ELSE

 ID=""

 PRINT BELL

 RETURN

 END

DICT.SOUNDA SWORDS, OWORDS

FINDID

ID,OWORDS,FLID,CONV,LEFT,TOP,LENGTH,CWID,CONV.SUB,IDICT,AUTH.BRS,OPTS,TITLE,NOTLASTPASS

GET.NEW.ID CTRL.REC,ID,PRE,FMT,FILE,NO.CHK

Numeric

FILE is a handle not a name.

Control Record is automatically created.

FMTR%8

UPDATE.RECORD FILE.CODE,FILE,ID,OLDREC,NEWREC,UPD.ERR,CHANGE,DEL.VAL

Need to release.

Scrolling

Virtual scrolling region

VSCROLL.DEFINE

VSCROLL.SET

VPRINT

INPV

VDEL

VINS

Step 1

VSCROLL.DEFINE VIEW,COL,ROW,WIDTH,ROWS,SCREEN.ID

VIEW – ID for region

COL – X

ROW – Y

WIDTH

ROWS

SCREEN.ID – Optional – ‘1 of X’

Step 2

Make the region active

VSCROLL.SET VIEW

Step 3

Display Data

Vprint X,Y,DATA

Hot Keys

Step 1 – Define

Step 2 – Have an INP

Step 3 – Write code

MENU.LOAD HOR,VER,ILEN,HPOS,HKEY

Common Variable OPTION – in order of hot keys

INP -----; IF OPTION GOSUB SUBS

SUBS:

ON OPTION GOTO PRINT,COM

MENU.LOAD ,,,INSERT.KEY
MENU.LOAD ,,,DELETE.KEY

INPNO – No Hot Keys

CHECK.KEY KEY.ID,ENTRY.OK,LEVEL

CONTROL KEYS KEY,LEVEL RANGE,DEFAULT LEVEL

INP.PROMPT VAR,O.PROMPT,CONV,LGTH,VERIFY
INP.PROMPT ANS,'PROMT','YN',1

CONFIRM.DELETE SURE
MENU.CLEAR

Phantom

PH.EXE PROG.NAME,A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11,A12,A13,A14,A15
(No attrib marks)
Looks for 1 (Print),2 (Hold), 3 (Opts)

DRPT\$='TITLE'

DVR – Gather Data PHR – Produce report

PRINTER.ON SPEC,TITLE,DOC.ID,HDR,FTR,RPT.DFLT,NOTBASIC
PRINTER.OFF DOC.ID,PRINT.LOC

SUBS:
 ON OPTION GOTO DOIT,DOIT,DOIT

DOIT:
 PH.EXE
PHSTFILE –Phantom status file.
Common Variable PID\$

WRITE 'Selecting...' to PHSTFILE,PID\$
WRITE 'Spooling...' to PHSTFILE,PID\$

UT.PH.CLEANUP FLNM.ID

Binary Index

LIST.INDEX FILE ID (ALL)
CREATE.INDEX FILE ID NO.NULLS
BUILD.INDEX FILE ID (ALL)
DELET.INDEX FILE ID

Step 1 – Create dictionary

Step 2 – Create Index

Step 3 – Build

!!!QUIET SYSTEM!!!

If Build = Required or Null=Yes or In Dict = No then there is a problem.

CTM Seconds from midnight.

MTH

MTS

JLI.SELECT MESS,SD,ED,BRS,JTYPE,ASUB.FORMAT
JLI.READ.EMU IDS,ID

^ - part viewer

& - customer viewer

BSCAN

START MAY BE MORE THAN ONE REC.ID

SELECTINDEX

VIEW:##

C:/TMP

INP.BR HOR,VER,LENGTH,BR,NAME,BRCHS,TERR.OK,ALL.OK,HELP.ID

ALL.OK is not "ALL" but allows BR not authorized.

TERR.OK allows "ALL" but just authorized branches.

SDT=READATE(SD,DATE())

EDT=READATE(ED,DATE())

IF SDT>EDT THEN

ERR.MESS 2,3,'Start Date > End Date'

INPNO A,,,O

RETURN

END

LEDGER

S – Sale
P – Purchase Order
W – Work order
T – Transfer
A – Adjustment
C – Cash
Y – Payable
D – Discount
J – Journal entry
R – Rental

24 – 26 GL Posting

024 SLS~CTR~CORE"COGS~CTR~CORE"INVTYnTAXnTAX"AR

24 – Auto post – source product post id

Type-sales g/l account product type

025 7~183"7~63"7~235"7~74"7~75"7~11

25 – GL account BR~Account GL.ID

026 -2808"2016"-2016"-197"-197"3202

26 Amount

048 1²2

Generation specific list of line items.

049 1²2

List of all line items with an open Qty. Not shipped/rec'd

'LD' Array

'LEDGER.DET' Fake dictionary for line items as if records.

LD.GET LDID

LED(12) GID

LED(8) INVOICE# - Only Processed Generations

GEN → Value marked

GID, INVN <> GEN

ORDER.QUEUE – Contains a record for every open generation

S1928076.12819.1

OID.DATE.GID

(Short cut: Selecting by @ID – SELECT ORDER.QUEUE="S193"]")

REC.ID = 'S1928076.12819.1'

OID = FIELD(REC.ID,',',1)

GID = FIELD(REC.ID,',',3)

MATREAD LED FROM LEDFILE,OID

LOCATE GID IN LED(12)<1> SETTING GEN ELS

GID is 4 dec. .0001

ENTITY.DYNAM – Open orders Key = ENTITY ID

003 S1921190.0001²S1987399.0002²S1987557.0002²S1990492.0002

004 Bⁿ2253ⁿ-36200²Bⁿ2253ⁿ-75000²Bⁿ2253ⁿ-14570²Bⁿ2253ⁿ-5500

3 & 4 are bids

005 12911.S1972389.001²12917.S1975057.001

006 3ⁿ12965ⁿ1²3ⁿ12965ⁿ1

007 16536²2461

5 – 7 is open A/R

PRODUCT.DYNAM Key=PRODUCT ID

002 0000013002~13002~P1104416~37~1~E~~~0ⁿ0000013033~13033~P1104640~37~1~E~~~0ⁿ0

000013063~13063~P1104641~37~1~E~~~0ⁿ0000013094~13094~P1104643~37~1~E~~~0²²²
2222

2 – Priority~date~OID .BR 7,2↑RW*2

Open OID LDID GID

005 ~12889~S1963527~101~1~S~ⁿ~12909~S1971425~101~1~S~²²~12624~S1817021~1~3~S~²²
~12876~S1957737~47~1~S~²²²²

006 -1ⁿ-1²²-1²²-1²²²²

5 & 6 Bids OID LDID GID

007 197²²104²20

008 S~A10-1E²²S~R2-5²S~D2-C3²²²

7 & 8 Qty and location

SHIP.QTY = SUM(LD(5)<1,GEN>) + SUM(LD(6)<1,GEN>)

QSIGN = -1

AR GL Posting

ID=OID.INV# “R%3”

Open Bal

In TCL

;SS S1061869.002

ASUB ID : S1061869.002

001

002 ²11033.C0022122.001

003 11033²11033

004 3~83ü3~11²3~11

005 5410ⁿ-5410²5410

Alt+L jumps to LEDGER

READNEXT

GL.GET.JTYPE

```

1      SUBROUTINE (MODE,STAT,JTYPE)
2      ** Version# 5 - 07/20/1999 - 12:05pm - DONS - develop
3      BEGIN CASE
4      CASE STAT='$';           JTYPE=3
5      CASE MODE='S' AND STAT='Y'; JTYPE=6
6      CASE MODE='S';           JTYPE=1
7      CASE MODE='C';           JTYPE=2
8      CASE MODE='P';           JTYPE=4
9      CASE MODE='T';           JTYPE=5
10     CASE MODE='Y';           JTYPE=7
11     CASE MODE='D';           JTYPE=8
12     CASE MODE='J';           JTYPE=9
13     CASE MODE='A';           JTYPE=10
14     CASE MODE='W';           JTYPE=11
15     CASE MODE='R';           JTYPE=12
16     CASE OTHERWISE           JTYPE=9
17     END CASE
18
19     RETURN

```

ASUB.DATA.GET AGT,DTIDS,AINFO,AMTS,CN,AOD,AGD,BRS,GLAS,SCONLY,LME,XCUR

*** Subroutine: ASUB.DATA.GET

*** This routine will go out to the A/R file for the customer you supply
 *** (CN) and the asof date (AOD) you send in and pull back all open ar/ap
 *** items. It will also pull back all of the "aging" information
 *** for this customer/vendor in summary format within the AGT array
 *** (broken down by attb as to what age bucket it falls into). This
 *** routine can be used for a/r and a/p information so you should set GLAS
 *** appropriately for the information you want back.

*** If the Customer Number passed in is Attribute Mark delimited, the
 *** calling routine is only trying to get Open AR data for the ShipTo
 *** Customer in CN<2>. Another words, if the ShipTo Customer on the AR
 *** item does not match the ShipTo Customer passed in it will NOT be
 *** included in the data returned.

*** AGT - Aging summary information (attb 1 = bucket 1 etc.) (OUT)
 *** DTIDS - AR IDs (with dt. in front) that are open to this CN (OUT)
 *** AINFO - Add'l information (shipdate, status,St) about this ID (OUT)
 *** AMTS - Open a/r or a/p amount for this "invoice" (OUT)
 *** CN - Customer/Vendor (ST or SF) you want information for (IN)
 *** > When this value is Attribute Mark delimited, only those

*** Open AR items whose ShipTo MATCHES the Customer Number
 *** passed in Attribute 2 (CN<2>) will be included.
 *** AOD - Asof date you want the information for (" = current) (IN)
 *** AGD - Aging date to use when calculating AGT information (IN)
 *** BRS - Branches you want information for (" = ALL) (IN)
 *** GLAS - A/R or A/P items (1 = A/R, 2 = A/P) (IN)
 *** SCONLY - AM separated: 1 - Service Charges 2 - Exclude Credits (IN)
 *** 1 = You only want service changes passed back
 *** 1 = You want to exclude credits from Aging buckets
 *** LME - Last Month End, Cutoff dt for records to return ("=ALL) (IN)
 *** XCUR - Currency you want this to be displayed in (" = base) (IN)

 *** NOTE : CN & GLAS are the only req'd variables to pass in. If any other
 *** "IN" params are left null it will default to ALL or CURRENT
 *** information. To speed this routine up you should always pass in a null
 *** (") AOD if you want the current information so it can use ENTDFILE
 *** rather than looping through the AR file.

LD.GET LDID

PSUB

PN~SHPBR~DATE~OID~INV#~LDID~LOC~ALTLOC

LD 8 & 9 Override

;DISP PSUB

EXPLODE

Use with the SEARCH command to specify that an element be included in the select list for each value in a record that matches a specified string. The select list elements also contain the field and value number in which the string was found. When the EXPLODE keyword is used, the list elements are created according to the form:

record.idVvalueSfieldVO

record.id is the record ID where the string is found.

field and value are the field and value numbers within the record containing the string.

V represents a value mark, and S represents a subvalue mark.

PROD.INFO BRS,PN,SD,ED,SLS,SLS\$,COST,GP,AVEQOH,BASN,COST.TYPE,INCL

*** Subroutine : PROD.INFO

*** This routine goes through the PSUBFILE and returns sales, cost, GP
*** and onhand qtys info. If BASN is passed it will use GET.BASE to find
*** the specified cost basis. Data returned is branch specific.
*** If COST.TYPE is set, it will pass back the comm cost as COST,
*** otherwise, it will pass back cogs cost.

*** Parameters:

*** BRCHS - [IN] Branches
*** PN - [IN] Eclipse product number
*** SD - [IN] Start date; varies for desired data range
*** ED - [IN] End date; varies for desired data range
*** SLS - (OUT) Sales quantities
*** SLS\$ - (OUT) Sales dollars
*** COST - (OUT) Cost
*** GP - (OUT) Gross profit (calculated)
*** AVEQOH - (OUT) On hand quantities averaged over date range
*** BASN - (IN) Base number
*** COST.TYPE - [IN] Comm, COGS or None
*** INCL - [IN] Include display location types

GENLED.TRANS

Key=BR~ACCT*YEAR

Attribute = Julian Day

3~66*1998

002 ²²²3870551

003 ²²²²²²0

005 ²²²407773

006 ²²²4152331

007 ²²²1581500

008 ²²²2934870

009 ²²²699331

012 ²²²2013429

PROD.TRX PROD.SALES.HIST

Key=BR~PRODUCT ID*YEAR

Attribute = Julian Day

3~3794*1998

128 -1

161 ²¹

194 -1

236 -2²4

259 ²²-1

289 1

ENTITY.TRX

2314*1997

001 3~9L5²3~TOL²3~183²3~11²3~202²3~25²3~4²3~61²3~74²3~262²3~62²3~COG²3~COM²3~63²~235²3~364

002 2314²*

242 -12117²-12117²²12117²²-12117

245 -226496²-226496²²226496²²-226496

251 -146140²-146140²²146140²²-146140

252 -80903²-80903²²80903²²-80903

255 -8025²-8025²²8025²²-8025

256 5318²5318²²-5318²²5318

265 -15890²-15890²²15890²²-15890

ETRX.AMT.GET GAMT,CN.ID,SDT,EDT,BRS,SP.ID,TGT.CURR

CHECK.XREF

Bank Account~Check.id

Attrib<1> AR.ID

Attrib<2> Reconciled date

4~592

001 D1000657.001

002 10958

4~5920

001 D1006796.001

002 11292

OPEN.STANDARD.FILES

018 OPEN 'ABC.CODES' TO ABCCFILE ELSE

022 OPEN 'ABC.LOG' TO ABCLFILE ELSE

026 OPEN 'ABC.TRX' TO ABCTFILE ELSE

030 OPEN 'QUAL.CODES' TO QCFILE ELSE

034 OPEN 'QUAL.LOG' TO QLFILE ELSE

038 OPEN 'WHSE.QUEUE' TO WHQFILE ELSE

042 OPEN 'EDICT' TO EDICTFILE ELSE

046 OPEN 'BUY.GROUP' TO BGRPFILE ELSE

050 OPEN 'BUY.LINE' TO BLNEFILE ELSE

054 OPEN 'CHECK.XREF' TO CHKXFILE ELSE

058 OPEN 'COMBO.GROUP' TO CMBOFILE ELSE

062 OPEN 'CONTROL' TO CTRLFILE ELSE

066 OPEN 'DOCUMENTS' TO DPAFILE ELSE

070 OPEN 'ENTITY.DYNAM' TO ENTDFILE ELSE

074 OPEN 'ENTITY.LOG' TO ELOGFILE ELSE

078 OPEN 'ENTITY.PN.IDS' TO EPNFILE ELSE

082 OPEN 'ENTITY.TRX' TO ETFILE ELSE

086 OPEN 'ERR.LOGGER' TO ERRFILE ELSE

090 OPEN 'ERR.MSGS' TO EMESGFILE ELSE

094 OPEN 'FILE.DEFINE' TO FDFILE ELSE

```
098      OPEN 'GENLED'      TO GLFILE  ELSE
102      OPEN 'GENLED.TRANS' TO GLTFILE ELSE
106      OPEN 'HELP.INFO'   TO HELPFILE ELSE
110      OPEN 'HELP.DOC'    TO HDOCFILE ELSE
114      OPEN 'JLI.UPD'     TO JUPFILE  ELSE
118      OPEN 'LPRICE.INDEX' TO LPIFILE  ELSE
122      OPEN 'MAINT.LOG'   TO MLOGFILE ELSE
126      OPEN 'MD'          TO MDFILE   ELSE
130      OPEN 'MESSAGES'    TO MSGFILE  ELSE
134      OPEN 'MISC.DATA'   TO MISCFILE ELSE
138      OPEN 'MISC.INDEX'  TO MIDXFILE ELSE
142      OPEN 'ORDER.QUEUE' TO OQFILE   ELSE
146      OPEN 'OVERRIDES.LOG' TO OVRFILE ELSE
150      OPEN 'PHANTOM.STATUS' TO PHSTFILE ELSE
154      OPEN 'PLINE.BRCH'  TO PLNBFILE ELSE
158      OPEN 'PRICE-GRP'   TO PGRPFILE ELSE
162      OPEN 'PRICE.LINE'  TO PLNEFILE ELSE
166      OPEN 'PRINT.LOG'   TO PLOGFILE ELSE
170      OPEN 'PRINT.QUEUE' TO PQFILE   ELSE
174      OPEN 'PRINT.REVIEW' TO PRVWFILE ELSE
178      OPEN 'PRINTER.DEFINE' TO PDFNFILE ELSE
182      OPEN 'PROCURE.GROUP' TO PROCFILE ELSE
186      OPEN 'PROCURE.QUEUE' TO PROCQFILE ELSE
190      OPEN 'PROD.LOCS'   TO PLOCFILE ELSE
194      OPEN 'PROD.PRICE'   TO PPFILE   ELSE
198      OPEN 'PROD.SERIAL'  TO PRDSFILE ELSE
202      OPEN 'PRODUCT'     TO PRDFILE  ELSE
206      OPEN 'PROD.TRX'    TO PTXFILE  ELSE
210      OPEN 'PROGRAM.LOCKS' TO LOCKFILE ELSE
214      OPEN 'PU.IDX'      TO PUIFILE  ELSE
218      OPEN 'REPORTS'     TO RPTFILE  ELSE
222      OPEN 'SCREEN'      TO SCREEN.FILE ELSE
226      OPEN 'SVIEW'       TO SVIEWFILE ELSE
230      OPEN 'TAX.CODES'   TO TAXFILE  ELSE
234      OPEN 'TERMS'       TO TERMSFILE ELSE
238      OPEN 'TERRITORY'   TO TERRFILE ELSE
242      OPEN 'WORK.MISC'   TO WORKFILE ELSE
246      OPEN 'ZIP'         TO ZIPFILE  ELSE
250      OPEN 'MATRIX'     TO MXFILE   ELSE
254      OPEN 'MATRIX.DATES' TO MXDFILE  ELSE
258      OPEN 'GL.SYNC'     TO SYNCFILE ELSE
262      OPEN 'ENTITY'      TO CUSFILE  ELSE
266      OPEN 'PROD.DYNAM'   TO PRDDFILE ELSE
270      OPEN 'PSUB'        TO PSUBFILE ELSE
274      OPEN 'LEDGER'      TO LEDFILE  ELSE
278      OPEN 'LEDGER.LOG'   TO LEDLFILE ELSE
282      OPEN 'MENUS'       TO MENFILE  ELSE
286      OPEN 'INITIALS'    TO INIFILE  ELSE
```

```

290      OPEN 'LEDGER.RENTAL' TO RLEDFILE ELSE
294      OPEN 'AR' TO ARFILE ELSE
298      OPEN 'TRACKING.LOG' TO CLOGFILE ELSE

```

Tempfiles

```

UD.TEMPFILE CREATE
TYPE 30 SEP 4/8 18-LOW
MODULO FACTOR OF 2

```

MAINT.LOG

View log:

MAINT.LOG.VIEW PASSER

Basis

Basis Names	UoM	Curr	VLvl	Global	Basis
SSL-LIST	ea	0	1	DFLT-LIST.....	SSL-LIST
VDR-LIST	ea		1	DLFT-COST.....	REP-COST
				COGS-COST.....	LND-COST
LT-BRK	ea		9	COMM-COST.....	LND-COST
GT-BRK	ea		9	REBAT-COST....	REP-COST
REP-COST	ea		9	SELL-BREAK....	REP-COST
LND-COST	ea		9	PURC-BREAK....	REP-COST
AVG-COST	ea		9	DISP-COST.....	REP-COST
LASTCOST	ea		9		

GBASN	BASN
Global basis	Local basis name
	Local to price line

GET.BASE PN,BR,OBASN,PRC.DATE,BASE,PER.QTY,OKOPTS,OKQTYS,OKPNS,OKSPLS,XCURR (Needs Local)

GLOBAL.BNAME.GET GBASN,GBASIS,ORDER.COSTS

LOCATE BASIS-NAME IN GBNAMES SETTING GN

Converts to local:

GLOBAL.BASN.GET GBASN,BASN

CMP

@ID	A	0	@ID	L	M
DESC	A	1	Desc	40L	M
PROMPT	A	2	PROMPT	30L	M
CTRL.ID	A	3	CTRL.ID	15L	M
CONV	A	4	CONV	10L	M
FORMAT	A	5	FORMAT	10L	M
BR.SPEC	A	6	BR.SPEC	5L	M
VAL.PGM	A	7	VAL.PGM	15L	M
VAL.FILE	A	8	VAL.FILE	15L	M
CTRL.SUBR	A	10	CTRL.SUBR	9L	M
VALID.SUBR	A	11	VALID.SUBR	50L	M
INP.WIDTH	A	12	INP.WIDTH	6R	M
INP.CONV	A	13	INP.CONV	10L	M
CATEGORY	A	14	CATEGORY	12L	M
AREA	A	15	AREA	12L	M

CREDIT.MANAGER

001 User To Get Tracker When Order On Credit Hold

002 User To Get Tracker When Order On Credit Hold

003 CREDIT.MANAGER.ID

005 L#10

006 1

008 INITIALS

014 EMSG

015 CR