

Super SUPRTOOL Solutions



A Kubler Consulting Presentation at HP 3000 Solutions Symposium

Copyright 2002, Kubler Consulting, Inc

What's Inside

- Introduction
- Newer Features
 - STEXPORT – Webpage creation
 - CI Variables
 - Updating a entries from TABLE
- Data Validation with SUPRTOOL
- Fixing data
- Defining data items
- Selecting on dates
- Adding a record count and subtotaling

Announcements

- SUPRTOOL now ready for HP ELOQUENCE database use.
 - HP Eloquence is an IMAGE-like database that runs on HP-UX, LINUX and Windows NT/2K. More information can be found at:www.hp-eloquence.com
 - HP Eloquence is owned by HP Germany and supported by a 3rd party, Markmeier Software

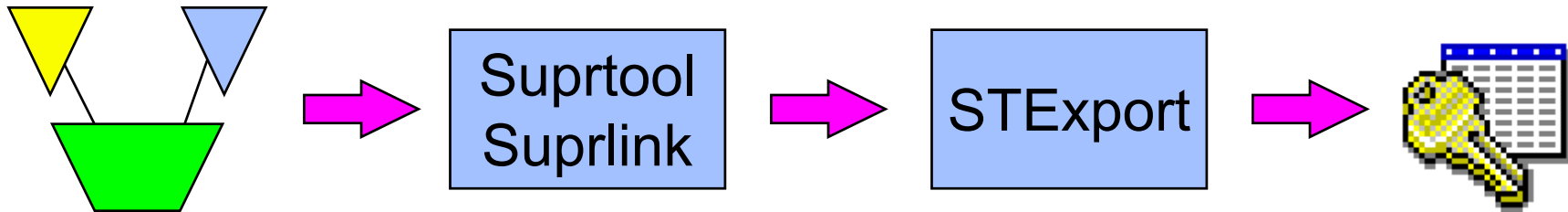
- SUPRTOOL for HP-UX already supports ORACLE.

Introduction

- **SPEED Stories**
 - **SUPRTOOL replaces Query**
 - **SUPRTOOL2 interfaces with COBOL**
- **Great Uses**
 - **Archival application**
 - **HTML presentations of data**
 - **Data validation/Data fixing**
 - **Date work**
- **One of the most Prevalent tools**
 - **AMISYS, ECOMETRY, etc.**

STEXPORT - Exporting IMAGE/SQL Data to other Applications

- Extract the IMAGE data using Suprtool and Suprlink
- Convert the files using STExport
- Transfer the file to the PC
- Import the delimited file



Data needs to be converted

- Image data has:
 - Fixed-width fields
 - binary storage formats (J2, K2, P28, etc)
 - Structure defined in Root File.

- PC Applications require:
 - variable-length fields
 - Ascii values for numerics
 - field delimiters
 - Field name declarations

STExport converts the data

- STExport reads self-describing files
- Outputs ascii files
- Allows you to specify:
 - field delimiters to use
 - date format
 - fieldnames in first record
 - numeric format
 - fixed or variable length
 - quotes on character fields
 - HTML - *table* or *preformatted*

3 ways to run STExport

- From the MPE prompt:
:run stexport.pub.robelle

- From Suprtool:

 - >export

 - STExport/iX/Copyright Robelle Consulting Ltd. 1995-1998 Type H for Help.
(Version 4.0.17 Pre-Release) MON, MAR 9, 1998, 1:50 PM
Licensee: Robelle Consulting Ltd. [0]

- From *inside* Suprtool:

 - >export input custsd

 - >export output custexp

 - >export exit

 - In=20. Out=20. CPU-Sec=1. Wall-Sec=1.

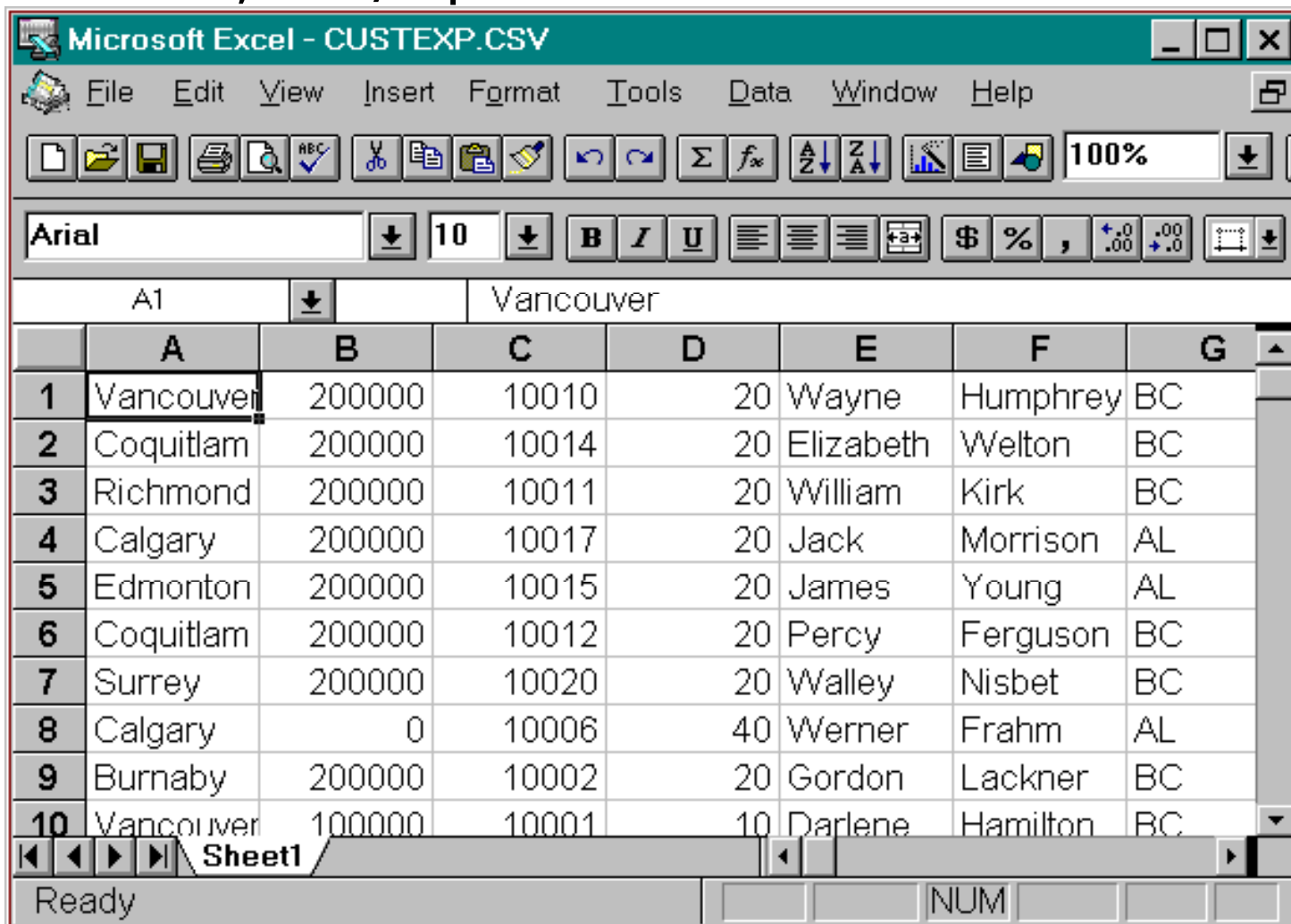
For example

```
>export
$in custsd
$out custexp
$xeq
In=19. Out=19. CPU-Sec=1. Wall-Sec=2.
$print custexp
```

```
"Vancouver",200000,10010,"20","Wayne","Humphreys","BC",....
"Coquitlam",200000,10014,"20","Elizabeth","Welton","BC",...
"Richmond",200000,10011,"20","William","Kirk","BC",.....
"Calgary",200000,10017,"20","Jack","Morrison","AL",.....
"Edmonton",200000,10015,"20","James","Young","AL",.....
"Coquitlam",200000,10012,"20","Percy","Ferguson","BC",.....
"Surrey",200000,10020,"20","Walley","Nisbet","BC",.....
```

In MS Excel

- Transfer to PC, File/Open in Microsoft Excel:



Microsoft Excel - CUSTEXP.CSV

File Edit View Insert Format Tools Data Window Help

Arial 10 B I U \$ % , .00 .00

A1 Vancouver

	A	B	C	D	E	F	G
1	Vancouver	200000	10010	20	Wayne	Humphrey	BC
2	Coquitlam	200000	10014	20	Elizabeth	Welton	BC
3	Richmond	200000	10011	20	William	Kirk	BC
4	Calgary	200000	10017	20	Jack	Morrison	AL
5	Edmonton	200000	10015	20	James	Young	AL
6	Coquitlam	200000	10012	20	Percy	Ferguson	BC
7	Surrey	200000	10020	20	Walley	Nisbet	BC
8	Calgary	0	10006	40	Werner	Frahm	AL
9	Burnaby	200000	10002	20	Gordon	Lackner	BC
10	Vancouver	100000	10001	10	Darlene	Hamilton	BC

Sheet1

Ready NUM

Dates and Decimals

- Use Suprtool's ITEM command to qualify the fields:

```
>get d-sales
```

```
>item deliv-date,date,YYYYMMDD
```

```
>item product-price,decimal,2
```

```
>out salesd,link
```

```
>x
```

```
IN=8, OUT=8. CPU-Sec=1. Wall-Sec=1.
```

```
>form salesd
```

```
File: SALESD.HANS.TRAINING (SD Version B.00.00)
```

Entry:		Offset	
CUST-ACCOUNT	Z8	1	
DELIV-DATE	I2	9	<<YYYYMMDD>>
PRODUCT-NO	Z8	13	
PRODUCT-PRICE	I2	21	<< .2 >>
PURCH-DATE	I2	25	...etc

..... continued

■ Specify date format in STEXPOR:

```
>export
$in salesd
$date DDMMYY "/"
$output *
$x
10020,04/10/97,50511501,98.31,19971000,2,2753,22415
10003,16/10/97,50511501,98.31,19971016,1,1376,11207
10003,16/10/97,50512501,145.62,19971016,1,2039,16600
10003,16/10/97,50513001,192.20,19971016,1,2691,21910
10016,20/10/97,50521001,24.59,19971020,3,1033,8411
10016,20/10/97,50532001,139.85,19971020,1,1958,15942
10020,28/10/97,50512501,146.60,19971028,1,2052,16713
10010,20/10/97,50533001,69.92,19971020,1,979,7970
In=8. Out=8. CPU-Sec=1. Wall-Sec=1.
```

Specifying field names

- Use HEADING command to add fieldnames in the first record:

```
$heading ' "Description", "Model" '
```

```
$heading add ' , "Product Code" '
```

```
$output *
```

```
$xeg
```

```
 "Description", "Model", "Product Code"
```

```
 "Skil 3/8 Variable Speed Drill", "#6523", 50531501
```

```
 "B&D Router", "#7613-04", 50522001
```

```
 "Skil Var. Sp. Auto-Scroll Saw", "#4560", 50533001
```

```
 "Skil 8 1/2 Circular Saw", "#5665", 50532501
```

```
 .....etc.....
```

- HEADING FIELDNAMES uses Image field names.

Fixed-length output

- Force fixed-length with COLUMNS command

```
$input prodsd
```

```
$columns fixed
```

```
$out *
```

```
$x
```

```
"Description", "Model", "Product Code"
```

```
"Skil 3/8 Variable Speed Drill", "#6523" , 50531501
```

```
"B&D Router" , "#7613-04" , 50522001
```

```
"Skil Var. Sp. Auto-Scroll Saw" , "#4560" , 50533001
```

```
"Skil 8 1/2 Circular Saw" , "#5665" , 50532501
```

```
"B&D Cordless Screwdriver" , "#9018-04" , 50521001
```

- Also see SPACES and ZERO commands

Preparing Data For The Web

- STExport can create HTML files
- Data can be formatted in a table
 - HTML TABLE command
- Or it can be formatted like a List Standard listing
 - HTML PREFORMATTED command
- Formatting is applied by STExport
 - Numeric data is right justified, with decimal points
 - Alpha data is left justified
 - Dates are formatted as you specify

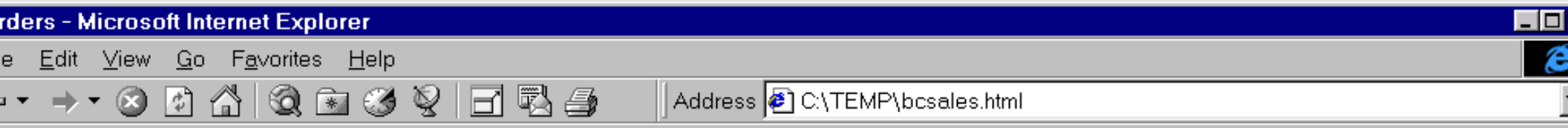
Preparing HTML Tables

- Use the HTML TABLE command

```
$input reptfile
$heading none
$heading column "Account #"
$heading column "Amount"
$heading column "Date"
$heading column "Product #"
$heading column "Last Name"
$heading column "First Name"
$html table title "Orders" heading "BC Sales over $100"
$output bcsales
$xeq
```


Table With Column Headings

- The table has one column per field, and one row per record



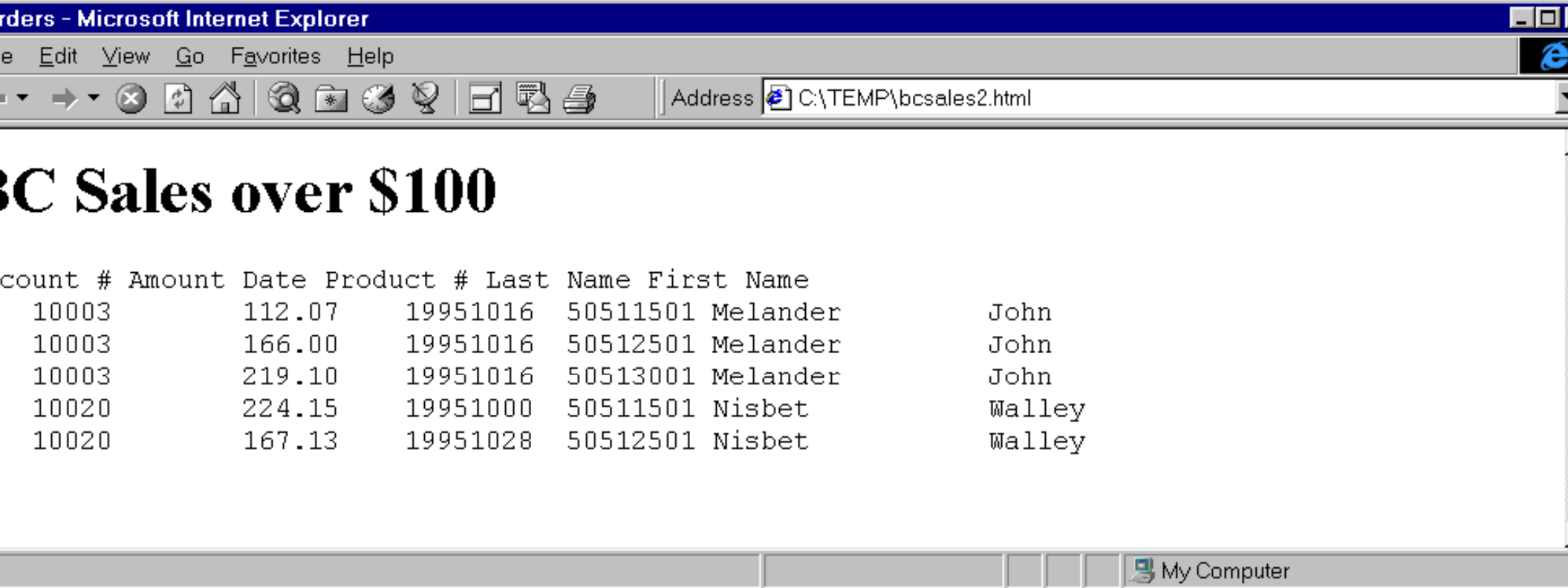
BC Sales over \$100

Account #	Amount	Date	Product #	Last Name	First Name
10003	112.07	19951016	50511501	Melander	John
10003	166.00	19951016	50512501	Melander	John
10003	219.10	19951016	50513001	Melander	John
10020	224.15	19951000	50511501	Nisbet	Walley
10020	167.13	19951028	50512501	Nisbet	Walley

My Computer

Listing-style Data

- Use the PREFORMATTED option instead of TABLE



The screenshot shows a Microsoft Internet Explorer window with the title "Orders - Microsoft Internet Explorer". The address bar displays "C:\TEMP\bcsales2.html". The main content area shows a heading "BC Sales over \$100" followed by a table of sales data. The table has columns for count, amount, date, product ID, last name, and first name. The data is presented in a listing style, with each row containing a count, amount, date, product ID, last name, and first name.

count #	Amount	Date	Product #	Last Name	First Name
10003	112.07	19951016	50511501	Melander	John
10003	166.00	19951016	50512501	Melander	John
10003	219.10	19951016	50513001	Melander	John
10020	224.15	19951000	50511501	Nisbet	Walley
10020	167.13	19951028	50512501	Nisbet	Walley

Calling Command Interpreter Variables

- Command Interpreter Variables can be a very useful way of adding to the selection criteria for a SUPRTOOL task. CI Variables can also be nifty items to add into the headings of reports, etc.
 - However, SUPRTOOL could not replace these variables into a SUPRTOOL task until version 4.3 of SUPRTOOL.
 - Certain users did find a way to do this using the echo command as the following slide illustrates.

TIPS - USING THE ECHO COMMAND to FEED VARIABLES TO SUPRTOOL (old).

- !FILE MACORD=MACORD.MACSDATA
- !ECHO BASE MACORD,1,DOALL >> TEMP4040
- !ECHO GET FINANCIAL-ORDER >> TEMP4040
- !ECHO DEF A,BIG-STATUS:0,1 >> TEMP4040
- !ECHO DEF B,BIG-STATUS:1,1 >> TEMP4040
- !ECHO DEF C,BIG-STATUS:2,1 >> TEMP4040
- !ECHO DEF ORD,FULL-ORDER-NO[1],8 >> TEMP4040
- !ECHO IF (A = "N","P" AND B <> "V" AND C = "1","3") AND & >> TEMP4040
- !ECHO (DATE >= "!MONTHSTRT" AND DATE <= "!MONTHEND") AND & >> TEMP4040

TIPS - USING THE ECHO COMMAND to FEED VARIABLES TO SUPRTOOL.

- !ECHO (DIVISION = "01") >> TEMP4040
- !ECHO EXT ORD,'0000',FULL-ORDER-NO >> TEMP4040
- !ECHO OUT FO1,LINK >> TEMP4040
- !ECHO X >> TEMP4040
- !ECHO IN FO1 >> TEMP4040
- !ECHO DEF FON,1,12 >> TEMP4040
- !ECHO EXT FON,FULL-ORDER-NO >> TEMP4040
- !ECHO SORT FON >> TEMP4040
- !ECHO OUT FOTEMP,LINK >> TEMP4040
- !ECHO X >> TEMP4040
- !ECHO EXIT >> TEMP4040
- USE TEMP4040

Tip - inserting the current time

- There's no built-in Suprtool function for inserting the current time. However, you can use HP variables and command I/O redirection.

```
> define timestamp,1,8
```

```
> echo extract timestamp = "!HPTIMEF" > foo
```

```
> use foo
```

- This will insert an X8 field called "timestamp" into each output record. The timestamp contains the time the data was extracted.

CI VARIABLES

- A new set command has been added to turn on a Variable Substitution at the command line for Suprtool, Suprlink and STExport.
- The set command:
 - >set VarSub On
- Turning this option on tells Suprtool, Suprlink and STExport to resolve any CI variables to be resolved on the command line.
 - NEW at 4.2.53

CI Variables

- After setting VARSUB ON
 - Get m-customer
 - Set varsub on
 - Ext "!hpsusan"
 - Ext cust-account
 - L s
 - X
- If statement can now call directly as in IF (A = "N","P" AND B <> "V" AND C = "1","3") AND & (DATE >= "!MONTHSTRT" AND DATE <= "!MONTHEND")

TIPS - USING THE ECHO COMMAND to FEED VARIABLES TO SUPRTOOL (new).

- FILE MACORD=MACORD.MACSDATA
- BASE MACORD,1,DOALL
- GET FINANCIAL-ORDER
- DEF A,BIG-STATUS:0,1
- DEF B,BIG-STATUS:1,1
- DEF C,BIG-STATUS:2,1
- DEF ORD,FULL-ORDER-NO[1],8
- IF (A = "N","P" AND B <> "V" AND C = "1","3") AND &
- (DATE >= "!MONTHSTRT" AND DATE <= "!MONTHEND") AND &

TIPS – USING CI VARIABLE IN SUPRTOOL.

- !ECHO (DIVISION = "01")
- EXT ORD,'0000',FULL-ORDER-NO
- OUT FO1,LINK
- X
- IN FO1
- DEF FON,1,12
- EXT FON,FULL-ORDER-NO
- SORT FON
- OUT FOTEMP,LINK
- X
- EXIT

New – Updating from a Table

- *Suprtool added the Update-From-Table feature in version 4.4.10*
- Many applications keep a running total in the master record of all the detail transactions. You can use the latest feature in Suprtool to easily do this. The following example reads the D-SALES detail set and summarizes the total sales for each customer.
 - >base store,1,writer
 - >get d-sales
 - >sort cust-account
 - >duplicate none keys total sales-total
 - >extract cust-account
 - >output summary,link
 - >xeq

New – Updating from a Table

- >input summary;list standard;x
- Aug 23, 2001 11:56
- CUST-ACCO ST-TOTAL-1
 - 123 136193+
 - 4566 723309+
- IN=2, OUT=2. CPU-Sec=1. Wall-Sec=1.
-

New – Updating from a Table

- Now let's update the master dataset records. Notice the new syntax on the TABLE and the EXTRACT commands.
- `>table t-sales,cust-account,file,summary,data(st-total-1)`
- `>get m-customer`
- `>if $lookup(t-sales,cust-account)`
- `>update`
- `>extract sales-history = &`
- `>>$lookup(t-sales,cust-account,st-total-1)`
- `>xeq`

New – Updating from a Table

- Finally, here is what the Master set records look like:
- >g m-customer;ext cust-account,sales-history;l;x
- >GET M-CUSTOMER (134) >OUT \$NULL (0) CUST-ACCOUNT = 4566 SALES-HISTORY = 723309
- >GET M-CUSTOMER (187) >OUT \$NULL (1) CUST-ACCOUNT = 123 SALES-HISTORY = 136193

Data Validation using SUPRTOOL

- Another important capability SUPRTOOL provides is the ability to test and check data for proper form/type and range within the data.
- SUPRTOOL has a number of capabilities built into it that make it essential for any one concerned that their data is what it should be.
- Data validation capabilities within SUPRTOOL include ensuring that data is of numeric type, ensuring that a date is valid, ensuring that specific characters are in the record, subtotaling to ensure that totals are correct, etc.
- SUPRTOOL allows for pattern matching.
- SUPRTOOL can check dates to ensure they are valid.

Data Validation - Comparing fields

- You can compare one field to another
>**if delivery-date = purchase-date**
- You can compare a numeric field to a calculation
>**if sales-total <> product-price * sales-qty**
- You can compare a field to a constant
>**if customer-status = "OK", "DEAC"**

Data Validation - Identifying a field as a date

- First use the ITEM command to identify a field as a date:

```
>item transaction-date, date, mmddy  
>item date-of-birth, date, phdate  
>item disbursement-date, date, ccyyymmdd
```

- Then use the IF command to select records:

```
>if transaction-date = $today and &  
date-of-birth < $date(1950/01/01) &  
and disbursement-date >= &  
$date(*+5/*/*)
```

Data Validation - Verify that dates are valid

- Use \$INVALID to select records with invalid dates

```
>item entry-date,date,yyyymmdd
>if $invalid(entry-date)
>list standard title "Records with bad dates"
```
- Or use it to deselect invalid dates

```
>if not $invalid(entry-date) and &
entry-date > $date(*/*-6/*)
```

Data Validation – Looking for duplicates

- Duplicates in the data can occur because of a number of human errors. Once they exist, how can you identify and get rid of them?
- Using SUPRTOOLS power DUP command duplicates can easily be identified and removed.
 - The command DUP NONE KEYS to remove duplicates.
 - A sort is a key to proper function.
 - The command DUP ONLY KEYS will identify only those records that are duplicates.

Data Validation – Checking the Pattern

- SUPRTOOL's powerful pattern matching capability to be of great help in checking data for the correct pattern of use. Is the supposed to be a numeric value in a specific position? Is there supposed to be a special character? These checks and many more are available using pattern matching.
- An example from ECOMETRY deals with the storage of email addresses:
 - Base macord,5,password
 - Get CUSTOMER-ADDL
 - Def EMAIL,ADDL-DATA[9],50
 - If email == “@&@@” (this gets you anything with an @ embedded)

Fixing Bad Data

- For small fixes use DBEDIT

- To get into DBEDIT, you first run Suprtool, open the database, and use the EDIT command:

```
EDIT      {LIST Command is great for looking at data!}
```

```
– #L STYLE-TABLE;RELATED
```

```
– List in File: STYLE-TABLE
```

- ```
 STYLE-TBL >COLOR-2____
 STYLE-TYPE >CO
 STYLE-DESC >_____
```

```
– STYLE-TBL = COLOR-2 STYLE-DESC = COLOR TABLE 2
```

```
 STYLE-MAX-CODE = 4
```

```
STYLE-MAX-DESC = 10
```

```
STYLE-TBL-ENTRY = ALPI ALPINE
```

# Fixing Bad Data

```
#modify d-inventory : unit-cost
```

```
Modify within File: D-INVENTORY
```

```
SUPPLIER-NAME >STD Ribbons
```

```
PRODUCT-NO >105391
```

```
Enter new values (or <Return> to leave as is):
```

```
SUPPLIER-NAME = STD Ribbons
```

```
PRODUCT-NO = 105391
```

```
UNIT-COST = 500
```

```
_____ {enter new unit cost}
```

# Fixing Bad Data

- For large fixes use either SUPRTOOL or DBEDIT
  - DBEDIT Change/Add/Delete Commands

## #change m-supplier

Enter existing key value to find:

SUPPLIER-NAME >ACME

Enter new key value to replace with:

SUPPLIER-NAME >ACME SUPPLY

```
SUPPLIER-NAME = ACME CITY = Los
Angeles
STATE-CODE = CA STREET(1) = 100 Main
STREET(2) = ZIP-CODE = 91201
OK to change this entry[no]:Y
```

# Fixing Data

- Or SUPRTOOLS Update or ADD (PUT to) command
  - **>get d-sales**
  - >item purch-date, date, yymmdd**
  - >if purch-date < \$date (\*-1/\*/\*)**
  - >update**
  - >extract purch-status = "OLD"**
  - >xeq**



# Defining Data Items

- Sometimes data item names are not reflective of content, or there are more than one data item e.g. BIG-DATA. A define statement can be used to specify the new item.
- BIG-DATA X200
  - DEF INS,BIG-STATUS[2],3
    - means create a new field called INS in suprtool that is 3 bytes long by taking the data in the BIG-STATUS field starting in the 2 position.

# *Positions can be referred to directly*

- BA MACORD,5,READALL
- GET PROCESSING
- IF ACTION-SEARCH = "P7" AND BIG-DATA(1,160,2) = "Z1","Z2","Z3"
- EXT FULL-ORDER-NOSO FULL-ORDER-NO
- O PROCTEMP,L,T
- XEQ

# Selecting on Dates

- The \$function capability of SUPRTOOL allows for easy checking of dates.
  - `IF CUST-STATUS = "IA" AND & ORIG-DATE >= $TODAY(-14) AND ORIG-DATE <= $TODAY(+14)`
    - The \$TODAY can be used provide the ORIG-DATE is first declared as a date via an ITEM statement.
      - `ITEM ORIG-DATE,DATE,YYYYMMDD`
  - The \$DATE allows for adding to and subtracting from the YEAR/MONTH and DATE.
    - `GET SERVICE`
    - `ITEM YMDEFF, DATE, YYYYMMDD`
    - `IF YMDEFF > $DATE(*-1/*-1/01)`

# *Selecting on Dates*

- The \$DATE function also allows for the use of the words "Last" .
- Additional DATE handling is done by \$DAYS, and \$STDLIST.

# Getting the Day of the Week

- "Is there a way to pick dates that will include only the dates between the last day of the month (`$date(*/*-1/last)`) and the prior Monday? This is for a report that looks at only the days from a Monday to the last day of the month for partial week. "
- Solution: Realize the Julian day 0 is a Monday
- Convert any date to a Julians value (via Suprtool's "`$days`" function), and divide by 7, the modulus would tell you how many days that date is from the previous Monday.

# *Getting the Day of the Week*

- Steps:
  1. Get the date of the last day of the month
  2. Convert that date to a juliandays value
  3. Divide the juliandays by 7, the modulus shows how many days the date was beyond the previous Monday.
  4. Deduct that number of days from the date in 1) (above), to calculate the date of the last Monday

# Getting the Day of the Week

- purge dtfile
- purge dtfile2
- purge dtfile3
- input catalog.pub.sys {any file with at least 1 record}
- def lastdate,1,8,display
- ite lastdate,date,yyyymmdd
- ext lastdate = \$date(\*/\*-1/last) {output the date for last day of last month}
- num 1 {only need 1 output record}
- out dtfile,link
- x
- in dtfile
- def lastmonday,1,4,int
- extract lastdate
- ext lastmonday = (\$days(lastdate) - (\$days(lastdate) mod 7)) {calculate juliandays value for the previous Monday}
- out dtfile2,link
- x

# Getting the Day of the Week

- in dtfile2
- extract 'setvar lastdayoflastmonth,', lastdate {create a "setvar" command for the last day of previous month}
- :file dtfile3;rec=-80
- out dtfile3,ascii
- num 2 {leave space for a second record}
- set squeeze off
- xeq
- in dtfile2 {create a "setvar" command for the previous Monday}
- item lastmonday,date,julianday
- extract 'setvar previousmonday,'
- ext lastmonday = \$stddate(lastmonday)
- out dtfile3,ascii,append
- x
- use dtfile3



# Getting the Day of the Week

- DTFILE3 now contains 2 setvar commands:
- /I dtfile3 1 setvar lastdayoflastmonth,20001231 2 setvar previousmonday, 20001225 ... and the file has been "use"d in Suprtool, so the variables have been set. They can then be referenced further down in the jobstream, as follows:
- >set varsub on
- >if mydate >= !previousmonday and mydate <= !lastdayoflastmonth  
>verify if
- IF mydate >= 20001225 and mydate <= 20001231

# ***Include a Record Count with Total***

- List Standard Device XXX prints a simple report on device XXX
- Total \$File \$List prints the total of a field on the same List device
  - get D-SALES
  - ext CUST-ACCOUNT, PRODUCT-PRICE, SALES-TOTAL
  - Total PRODUCT-PRICE, SALES-TOTAL
  - total \$file \$list
  - list standard
  - device LP
  - xeq

# Counting Records

- with two passes. The first pass adds a field with a value of 1, and the second pass totals that field, effectively providing a count of the number of records
  - get dataset
  - ext id, zone, amount
  - define count,1,4,int
  - ext count = 1
  - output tempfile,link
  - xeq

# *Counting Records*

---

- input tempfile
- ext id\amount
- total count
- total amount
- total \$file \$list
- list standard device LP
- xeq

# ***Sub-Totaling with Suprtool***

- >base store,5,reader
- >get d-sales {open a dataset}
- >sort product-no {define a sort key}
- >duplicate none keys count total sales-qty sales-total
- >out salessum,link {Output to a link file}
- >xeq

---

# *The End*

For more information on this presentation go to  
[www.kublerconsulting.com](http://www.kublerconsulting.com)

Or

[www.robelle.com](http://www.robelle.com)