Using Suprlink to Combine Data Files without Keys

A Robelle Tutorial

Paul Gobes, August

1

In this tutorial, you will learn how to combine data from several different files or TurboIMAGE datasets without using keys, or even needing them in the database. Suprtool is Robelle's high-performance tool for extracting and manipulating data on HP 3000 systems. Suprlink is the Suprtool component that combines the data. You will learn how to use Suprlink to attach customer names to invoice details, and to link from product numbers to customers who have ordered a product. You will also learn when to apply Suprlink for optimum performance.

Using Suprlink to Combine Data Files without Keys is taught by Paul Gobes, Robelle's senior technical manager and someone who has answered a lot of questions about Suprlink. Each participant will receive a *Suprtool Quick Reference Guide* and a tutorial workbook that contains a complete set of examples.

Robelle Solutions Technology Inc.
Suite 201, 15399 - 102A Avenue
Surrey, B.C. Canada V3R 7K1

Toll-free: 1.888.762.3553
Telephone: 604.582.1700
Fax: 604.582.1799
E-mail: support@robelle.com
WWW: http://www.robelle.com

For Techies

References

For further information on topics covered in this tutorial, please consult the *Suprtool User Manual*.

What's Inside

2

When most data tools combine information from several different files or TurboIMAGE datasets, they use indices built up for specific key or search fields. This tutorial will show you an alternate method of linking files that does not require existing key structures. With this method, you can link on almost any field. For batch reporting, if not for on-line inquiries, the method you will learn today often works faster than the keyed method.

Throughout this tutorial, we will digress into areas suggested by your questions, so please ***ask questions***!

For Techies

References

Let's print the customers in B.C.

```
>get m-customer
>if state-code="BC"
>extract cust-account,name-first,name-last
>list standard,title "Customers in BC"
>xeq
Apr 10, 1995 9:41  Customers in BC  Page 1
CUST-ACCO   NAME-FIRST  NAME-LAST
10010       Wayne       Humphreys
10014       Elizabeth   Welton
10011       William     Kirk
10012       Percy       Ferguson
10020       Walley      Nisbet
10002       Gordon      Lackner
10001       Darlene     Hamilton
10003       John        Melander
```

3

Suprtool can retrieve, select, rearrange, sort, and print data from IMAGE databases. To begin, you have to select a database and provide a password with the Base command. This is how you would open the demonstration Store database on the Suprtool installation tape:

```
:run suprtool.pub.robelle
>base store.demo.robelle,5,READER {mode and password}
```

As you can see, it is very easy to get a list of the B.C. customers. You can also sort the records by last name, or list their credit standing with the following commands:

```
>sort name-last
>extract credit-rating
```

So what's the catch? In this example, all the information we need is contained in a single dataset. In real world problems, the required information is usually dispersed among many datasets.

For example, what if you only want October 1993 purchases by B.C. customers?

For Techies

References

Finding dispersed information

```
>form sets
Database: STORE.DEMO.ROBELLE
                Set            Item              Entry  Load     Entry
Sets:           Num   Type     Count  Capacity   Count  Factor   Length
 M-CUSTOMER     1     M        9      211        20     9 %      55
 M-PRODUCT      2     M        3      307        13     4 %      24
 M-SUPPLIER     3     M        6      211        3      1 %      49
 D-INVENTORY    4     D        6      462        13     3 %      15
 D-SALES        5     D        8      602        8      1 %      19
```

Suprtool's FORM command can help you locate information within a database. After opening the database, use the FORM SETS command to display all the dataset attributes such as the size and type.

```
>form m-customer
Database: STORE.DEMO.ROBELLE
M-CUSTOMER    Master       Set 1
 Entry:                       Offset
    CITY                    X12   1
    CREDIT-RATING           J2    13
    CUST-ACCOUNT    Z8      17 <<Search Field>>
    CUST-STATUS             X2    25
    NAME-FIRST              X10   27
    NAME-LAST               X16   37
    STATE-CODE              X2    53
    STREET-ADDRESS         2X25   55
    POSTAL-CODE             X6    105
Capacity: 211 (7)  Entries: 20   Bytes: 110
```

You can use the command FORM M-CUSTOMER to see what fields are in this dataset. The d-sales dataset looks like it contains the purchase date information you need.

How would you display the fields available in d-sales?

How would you display all the items in the database?

How would you display the sets that contain a specific item?

Does d-sales contain purchase date information?

```
>form d-sales
Database: STORE.DEMO.ROBELLE
D-SALES   Detail  Set 5
  Entry:                Offset
      CUST-ACCOUNT     Z8     1   (!M-CUSTOMER)
      DELIV-DATE       J2     9
      PRODUCT-NO       Z8     13  (M-PRODUCT)
      PRODUCT-PRICE    J2     21
  ⇨   PURCH-DATE       J2     25
      SALES-QTY        J1     29
      SALES-TAX        J2     31
      SALES-TOTAL      J2     35
Capacity: 602 (14) Entries: 8 Bytes: 38
```

- Which field is also in the m-customer dataset?

5

The d-sales dataset does contain information about product purchases by customers. The purch-date field looks like it holds purchase date information. To verify this, you can print some entries with the List command.

```
>get d-sales; list; xeq
 CUST-ACCOUNT   = 10020      DELIV-DATE     = 19931005
 PRODUCT-NO     = 50511501   PRODUCT-PRICE = 9831
 PURCH-DATE     = 19931001   SALES-QTY      = 2
 SALES-TAX      = 2753       SALES-TOTAL    = 22415
 ...
```

You can see that the first entry in the purch-date field has a value of 19931001 which is probably October 1, 1993 in the YYYYMMDD format.

For Techies

References

The cust-account field is included in this dataset to identify the customer. Further information about the customer is available in the m-customer dataset using the cust-account field as a key. Cust-account is also a key to the d-sales dataset, as shown by the flag (!M-CUSTOMER) after the field name in the Form listing.

Let's list the October purchases

```
>get d-sales
>if purch-date >= 19931001 and purch-date <= 19931031
>extract purch-date,cust-account
>list standard,title "Purchases in October 93"
>xeq
Apr 11, 1995 8:31  Purchases in October 93  Page 1
PURCH-DATE   CUST-ACCO
19931001     10020
19931015     10003
19931015     10003
19931015     10003
19931021     10016
19931021     10016
19931028     10020
19931020     10010
```

6

Using the same Suprtool commands as you did to get the m-customer dataset, you have retrieved and printed all October purchases from the d-sales dataset. Keep in mind that some of these are not purchases by B.C. customers.

Now you have a listing of October purchases, as well as your previous listing of B.C. customers:

```
Customers in BC                              Page 1
CUST-ACCO       NAME-FIRST    NAME-LAST
  10010         Wayne         Humphreys
  10014         Elizabeth     Welton
  10011         William       Kirk
  10012         Percy         Ferguson
  10020         Walley        Nisbet
  10002         Gordon        Lackner
```

If both listings were sorted on the cust-account field, it might be easier to find the October purchases by B.C. customers.

For Techies

References

Here are the sorted listings

- Can you find the October purchases by B.C. customers?

```
Purchases in October 93  Customers in BC
PURCH-DATE CUST-ACCO    CUST-ACCO NAME-FIRST NAME-LAST
  19931015  10003---|      10001    Darlene    Hamilton
  19931015  10003---|      10002    Gordon     Lackner
  19931015  10003--------->10003    John       Melander
  19931020  10010---|      10008    Thomas     Serafin
  19931021  10016   |      10009    Gordon     Oxenbury
  19931021  10016   ------>10010    Wayne      Humphreys
  19931001  10020---|      10011    William    Kirk
  19931028  10020---|      10012    Percy      Ferguson
                    |      10013    Colin      Andersen
                    |      10014    Elizabeth  Welton
                    |      10019    Rupert     Hillstrom
                    ------>10020    Walley     Nisbet
```

7

Since both files are already sorted by cust-account, now it is easier to select the desired purchases. Although they are B.C. customers, you skip customers 10001 and 10002 because they don't have any purchases in October. You include B.C. customers 10003 and 10010 because they did make purchases in October. Although 10016 did make a purchase in October, he is not a B.C. customer, so you skip him. Similarly, you skip customers 10008, 10009, and 10011 through 10019 because they didn't make any purchases. Customer 10020 is a B.C. customer who purchased two items in October, so you keep him.

Ultimately, what you want is the intersection of the two reports you've already produced.

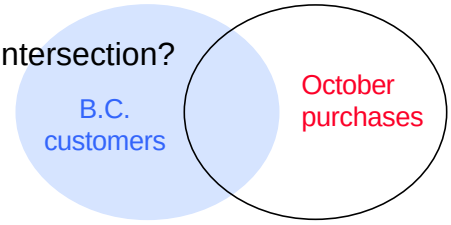
For Techies

References

The intersection of your two reports

PURCH-DATE	CUST-ACCO	CUST-ACCO	NAME-FIRST	NAME-LAST
19931015	10003	10003	John	Melander
19931015	10003	10003	John	Melander
19931015	10003	10003	John	Melander
19931020	10010	10010	Wayne	Humphreys
19931001	10020	10020	Walley	Nisbet
19931028	10020	10020	Walley	Nisbet

- How can the computer select the intersection?

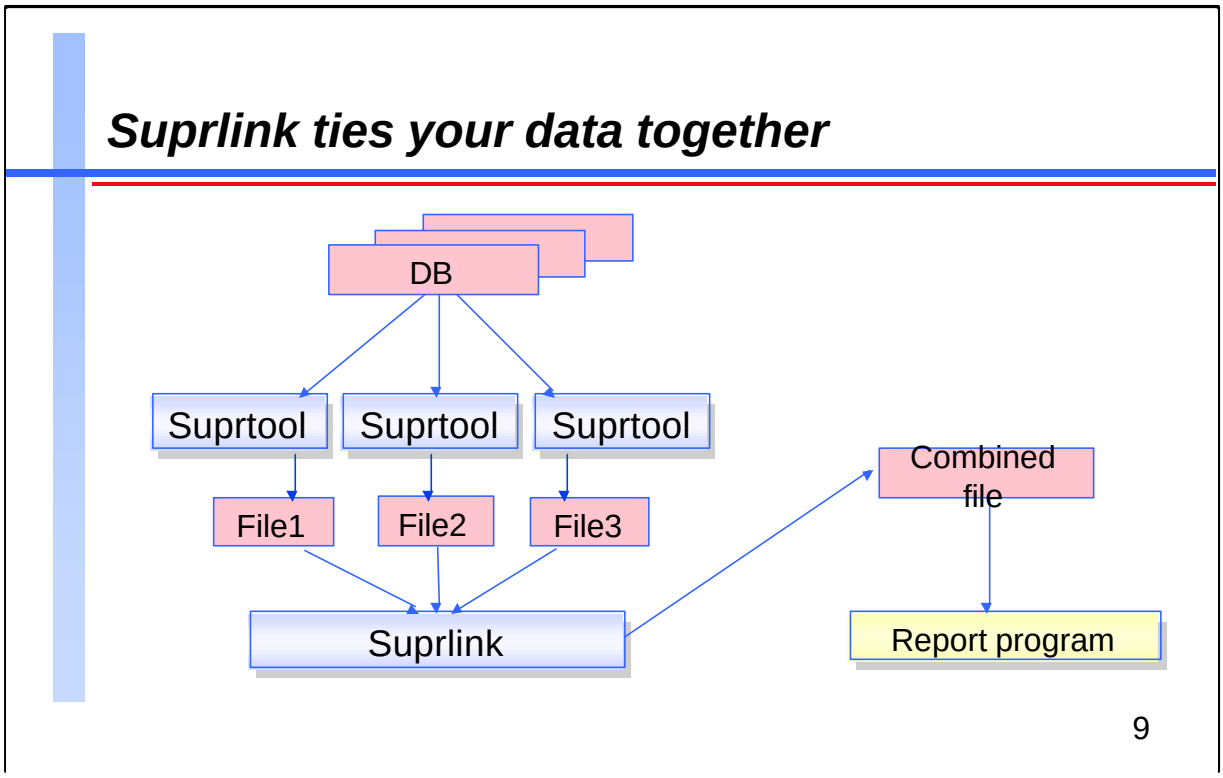


8

Still, this report is not all it could be. You can see who made the purchases, but not what they actually bought. In retrospect, you should have also extracted product-no from the sales dataset. What's the solution? You need to think about what data you want to collect, and how you want to display it in your report before you get the data and write the report.

For Techies

References



Once you've decided what data you need to display, Suprlink can be a big help to you. Suprlink is Suprtool's answer to your question: how do I combine data that is spread across many files?

The best way to understand Suprlink is to examine the process of writing a report. Eventually, you will write the report program in COBOL, RPG, PowerHouse or some other language. But before you can generate a report, the individual records must be collected. Imagine that instead of hunting all over the database with DBFIND and DBGET to collect individual records, you only have to read a sorted disc file with a big record that contains all the required data fields.

If you want this file to produce a sales report, it should contain records of sales transactions plus customer information. It would be sorted by customer number and by date. Then your report program would read the records, check for level breaks, and finally format and print the records.

For Techies

References

Extract October purchases into an SD file

```
>get d-sales
>if purch-date >= 19931001 and purch-date <= 19931031
>extract purch-date,product-no,cust-account

>sort cust-account
>output tempsale, temp, LINK

>xeq
```

- Now create second SD file containing B.C. customers

10

With a few exceptions, the commands in this example are similar to those used to produce the purchase listing. This time, you are not only extracting the product-no field, but also the cust-account field. Cust-account is needed as a sort key for a later part of this report writing process. Instead of using the List command, use the Output command with the Link option to create a self-describing (SD) file. The structure of your intermediate SD file would look like this:

```
>listftemp tempsale,2
ACCOUNT= ROBELLE  GROUP= DEMO
FILENAME CODE --------LOGICAL RECORD------  ----SPACE----
             SIZE   TYP  EOF   LIMIT  R/B   SECTORS X MX
TEMPSALE SD   20B   FA     7       7   64    16  1  1 (TEMP)
```

For Techies

References

Now extract the B.C. customers

```
>get m-customer
>if state-code="BC"
>extract cust-account,name-first,name-last
>sort cust-account
>output tempcust, temp, link
>xeq
```

11

Now you can extract the B.C. customers into their own SD file. You have to sort this file by cust-account, just as you did earlier with the October purchases.

For Techies

References

Final Step: Link the two SD files together

```
>link input tempsale
>link link tempcust
>link output tempfile, temp
>link xeq
```

October Purchases by BC Customers

PURCH-DATE	PRODUCT-N	CUST-ACCO	NAME-FIRST	NAME-LAST
19931015	50511501	10003	John	Melander
19931015	50512501	10003	John	Melander
19931015	50513001	10003	John	Melander
19931020	50533001	10010	Wayne	Humphreys
19931001	50511501	10020	Walley	Nisbet
19931028	50512501	10020	Walley	Nisbet

12

You can either run Suprlink on its own, or use it from within Suprtool, as in the example above. To run Suprlink, you use :Run Suprlink.Pub.Robelle and then use Suprlink commands. To work within Suprtool, you precede each Suprlink command with the word Link.

Once you have the final data in a temporary file, you can produce the above listing with these three Suprtool commands:

```
>input tempfile
>list standard,title "October Purchases by BC Customers"
>xeq
```

In summary, you did two Suprtool extracts into temporary SD files, linked the two SD files into a single combined file with Suprlink, and printed the final file. You did not have to tell Suprlink what was in the SD files or how it was sorted because Suprlink automatically figures out the structure of SD files. Suprtool purges your intermediate temporary files when your job logs off.

If you need an end-user report, you can feed Tempfile into a report writer and format it to your requirements. In fact, if you use Quiz as your report writer, you can feed the Suprlink output directly into a Quiz subfile. See the *Suprtool User Manual* for details.

For Techies

How you access Suprlink makes no difference to performance.

References

A review of the entire Suprlink task

- Here are the commands for all three steps of the Link task:

```
>base store.demo.robelle,5,READER

>comment Get purchases during October,Output to Tempsale
>get d-sales
>if purch-date>=19931001 and purch-date<=19931031
>extract purch-date,product-no,cust-account
>sort cust-account
>output tempsale, temp, LINK
>xeq

>comment Get customers in BC, Output to Tempcust
>get m-customer
>if state-code="BC"
>extract cust-account,name-first,name-last
>sort cust-account
>output tempcust, temp, LINK
>xeq

>comment Link Tempsale + Tempcust, Output to Tempfile
>link input tempsale
>link link tempcust
>link output tempfile, temp
>link xeq

>comment Print a simple listing of Tempfile
>input tempfile
>list standard,title "October Purchases by BC Customers"
>xeq
```

```
File: TEMPFILE.DEMO.ROBELLE (SD Version B.00.00)
  Entry:              Offset
    PURCH-DATE      I2    1
    PRODUCT-NO      Z8    5
    CUST-ACCOUNT    Z8    13 <<Sort 1 >>
    NAME-FIRST      X10   21
    NAME-LAST       X16   31
 Limit: 7 EOF: 5 Entry Length: 46 Blocking: 89
```

- Suprlink takes the fields of the input file, then appends the fields of the link file(s), dropping any duplicate key fields (E.g., cust-account). You can partially control the order of the fields in the output file by altering the sequence in which Suprtool extracts the fields.

What happens if we reverse the linking order?

- What if we link Tempsale to Tempcust?

```
>link input tempcust
>link link tempsale
>link output treverse, temp
>link xeq
```

```
Apr 11, 1995 14:43 October Purchases by BC Customers Page 1
CUST-ACCO                NAME-FIRST              NAME-LAST
                         PURCH-DATE              PRODUCT-N
 10003                   John       Melander     19931015
  50511501
 10010                   Wayne      Humphreys    19931020
  50533001
 10020                   Walley     Nisbet       19931001
  50511501
```

14

When you use Suprlink, you have to decide which file is the primary input file. The Suprlink output file cannot have more records than the primary input file.

If you are interested in transaction detail, you should select the transaction file as the input file. Since a customer usually has more than one transaction, in this case you link customer information into the transaction record, not vice versa.

As this example shows, if you start with one of the master files as your primary input file, you will have a smaller output file than the previous example. The Treverse file contains only three records, while the Tempfile in the last example contained six records. In fact, in such cases you end up with at most one record per master record, and each of these records is the *first* matching transaction record that Suprlink encountered.

For Techies

References

Student Exercise

- Enhance Tempfile by adding the product description to each purchase
- Hint: The command FORM M-PRODUCT gives you this information:

```
Database: STORE.DEMO.ROBELLE
M-PRODUCT   Master     Set 2
   Entry:                 Offset
   PRODUCT-DESC         X30     1
   PRODUCT-MODEL        X10     31
   PRODUCT-NO           Z8      41 <<Search Field>>
Capacity: 307 (12) Entries: 13 Bytes: 48
```

15

Starting with the existing Tempfile, produce a new output file that includes the product description for each purchase. Perhaps the report is for a user in the merchandising department.

Think about how many steps it will take, and whether you will need an intermediate temporary file? The final report should probably be sorted by product-no, since you are asking for additional product information.

Write out the commands to solve the problem. The solution is on the next page.

For Techies

References

Solution to Student Exercise

- If we want to link in the product-desc, we need to extract it from the m-product dataset, along with the product-no. The product-no is our key field for linking into the next file.

```
>get m-product
>extract product-no,product-desc
>sort product-no
>output tempprod, temp,link
>xeq
```

- If you want to link the product-desc to Tempfile, you need to re-sort the Tempfile by product-no because it is the key for product information.

```
>input tempfile
>sort product-no
>output =input  {sort back into the same file!}
>xeq
```

- Now you can use Suprlink to link Tempprod to Tempfile, producing Temptest.

```
>link input tempfile
>link link tempprod
>link output temptest,temp
>link xeq
```

- Now you have the final data in Temptest, which you can print with the Suprtool List command:

```
>input temptest
>list standard,title "Solution to Student Exercise"
>xeq
```

```
Apr 11, 1995 15:31  Solution to Student Exercise          Page 1
PURCH-DATE PRODUCT- CUST-  NAME-FIR NAME-LAST  PRODUCT-DESC
19931015   50511501 10003  John     Melander   3/8" Var Sp. Drill
19931001   50511501 10020  Walley   Nisbet     3/8" Var Sp. Drill
19931015   50512501 10003  John     Melander   8 1/4" Circ. Saw
19931028   50512501 10020  Walley   Nisbet     8 1/4" Circ. Saw
19931015   50513001 10003  John     Melander   1" Jigsaw
19931020   50533001 10010  Wayne    Humphreys  Skil Var Sp. Saw
```

What if there is no product description?

- By default, Suprlink drops records that fail any match
- To override default, use the OPTIONAL keyword on the LINK command
- Exercise Solution using the OPTIONAL description

```
>link input tempfile
>link link tempprod, optional
>link output temptest,temp
>link xeq
```

- Suprlink provides zero or blank values if link record is missing

17

What if there is no matching entry for your sort keys?

Suppose that in the previous example there was no matching entry for product description. This could happen through data entry error or because you are linking in a field that is optional, such as "special instructions". By default, Suprlink drops the input record if it doesn't find a match in the link file. You can override the default by adding the Optional keyword to the Link command. Then, if a match is not found, Suprlink initializes the link record fields to blanks or zeros.

In our previous example, some sales records would not have been included in the final output file if they didn't have a matching product description record. By using the Optional keyword, Suprlink includes every sales record even if it doesn't have a product description.

For Techies

References

How do you find the entries that don't match?

- Answer: Append a dummy field to each record
- Purchases missing a standard product description
  - Define Flag,1,2
  - Extract product-no, product-desc, flag = "xx"
  - Link optional; missing links have Flag = " "
  - Re-input the output file, apply selection:
    `If Flag <> "xx"` {product-desc not found}

18

If you append a dummy field with a known constant when you extract the optional fields, you can track the links that fail. The final link records that found a match will have a known value in the dummy field. The final link records that failed to link will also have the dummy field, but its value will be blanks (or zero for a binary field).

If you look at the file later, the dummy field makes it easy to distinguish the links that failed from the links that worked, especially if blanks are a valid value for the other extracted fields.

```
>input tfinal
>if flag <> "xx"            {did not find a match}
>extract product-no,cust-account,name-last,purch-date
>list standard,title "Purchases without a valid description"
>xeq
```

For Techies

References

Finding information from several datasets

- Pattern-match on m-product dataset; find customers who bought any drill
- Sort d-sales by product-no and cust-account; remove duplicates
- Link drills to sales; only select matches
- Sort all customer names by account number
- Re-sort drill purchases by account number
- Link customers to drill sales

19

For Techies

Reports frequently need information from more than just two datasets. For example, if you want a list of all the customers who have purchased a drill, you must look in three datasets:

1. m-customer dataset contains customer name and address
2. m-product dataset tells us if product is "drill" type
3. d-sales dataset tells us who bought what

In the sample database, the only way to select drills is to pattern-match on the description field to see if it contains the word "Drill".

```
>if product-desc == "@Drill@"
```

You sort all the d-sales entries by product number and by customer account. Then you can use the Duplicate command to discard multiple purchases of the same drill by the same customer.

```
>sort product-no; sort cust-account
>duplicate none keys
```

After you get a list of drill purchases, re-sort them by the customer account number so that you can link in the customer information.

References

Link from m-product to m-customer through d-sales

robelle
solutions technology

- Here is the complete Suprtool/Suprlink task to find all customers who have bought any drill product.
- Pattern-match on m-product dataset

```
>get m-product
>if product-desc == "@Drill@"
>extract product-no,product-desc
>sort product-no
>output tdrills, temp,link
>xeq
```

- Sort d-sales by product-no and cust-account; remove duplicates

```
>get d-sales
>sort product-no
>sort cust-account
>extract product-no,cust-account
>duplicate none keys
>output tsales, temp,link
>xeq
```

- Link drills to sales; select only matches

```
>link input tsales
>link link tdrills
>link output tdrsales,temp
>link xeq
```

- Sort all customer names by account number

```
>get m-customer
>extract cust-account,name-last
>sort cust-account
>output tcust, temp,link
>xeq
```

- Re-sort drill purchases by account number

```
>input tdrsales
>sort cust-account
>output =input
>xeq
```

- Link customers to drill purchases

```
>link input tdrsales
>link link tcust
>link output tfinal,temp
>link xeq
```

- Now you have a file, Tfinal, that contains a list of customers who have purchased a drill product. It includes the customer last name, the product purchased, and the customer number.

What is a self-describing file?

- Regular file plus a mini-dictionary
- Accepted in INPUT command
- Allows use of full field names in IF, EXTRACT, etc.
- Created by OUTPUT command in two flavors:

```
>output archfile,link
>output extrfile,query
```

21

Now you know that Suprtool creates a self-describing (SD) file when you use the Link option with the Output command. Suprlink only accepts SD files as input, so they are important. SD files also work well as archive files. An SD file is a standard MPE disc file with any record size and format, but with a special filecode and a mini-dictionary stored in some user labels. The mini-dictionary describes the name, size, type and position of all the fields in the SD file data record.

Suprtool can create and understand two flavors of SD file: Query and Link.

The Query-type SD files have been in Suprtool for years and follow a standard format that was invented by HP. This format retains the record's field names and data-types in some user labels.

The Link-type SD files are enhanced versions of the Query-type files and are recent additions to Suprtool. The enhanced user label format for Link-type SD files was developed by Robelle. It stores sort keys, compound items, and other information. Robelle has published the Link-type SD format as a standard that vendors can use.

For Techies

References

SD files have a record structure

```
>get m-customer
>if coststatus = 10,20
>extract custnumber,custname,coststatus
>output mysdfile,link
>xeq

>form mysdfile        {this file now has structure!}
>input mysdfile
>if coststatus = "10"
>sort custname
>extract custnumber
>list standard
>xeq
```

22

By applying the Form command against an SD file, you can see for yourself that SD files have a record structure. In this example, you want the account number, name, and status of the customer records that contain 10 or 20 in the status field. You write these selected and reformatted records to a Link-type SD file, which, of course, was newly created when you specified the Link option.

When you use the Form command against this new file, Suprtool sees that it is an SD file and prints the record structure in the same format as TurboIMAGE datasets.

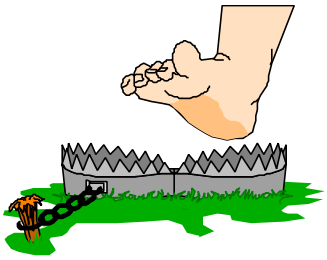
When you use the Input command against an SD file, Suprtool sees the mini-dictionary, and can recognize each field name and type. Once Suprtool knows the record structure, you can use the field names in your commands (E.g., If, Extract and Sort).

For Techies

References

Suprlink warnings

- Consumes a lot of disc space
- Extract files must be sorted
- Link fields of any type except real or long real
- File restrictions



23

Suprlink needs enough disc space to invert a large amount of your database, and then link it into an output file. Although all of the Suprlink files can be job temporary, you still need enough disc space for the original database, the final output file, the primary input file, each of the link files, and hidden Sortscr files (you can use Numrecs to reduce their size). Remember, you are trading disc space for speed.

In order to link the extract files, they must be sorted by the same key-field. In some cases, the time needed for sorting will be more than the time saved by not having to randomly search through the base.

For Techies

References

File restrictions

Maximum	Input file	Link files	Output file
No. of files	1	7	1
No. of fields	255	255	1023
Record size (words)	2048	2048	4096
Block size (words)	4096	2048	4096

24

The input file is the only file that should have multiple records with the same key values. The linking is many-to-one, not many-to-many. Therefore, if a link file has multiple records with the same sort key value, only the *first* one that matches will be linked to the current input record. Another way of saying this is you can never have more output records than input file records. Remember, you still have to do the thinking and planning for Suprlink.

For Techies

References

Performance guidelines for Suprlink

- Don't sort and merge if you must re-sort many times
- Keep record sizes small by only extracting needed fields
- Keep files small by only writing selected records
- Use serial scan with table lookup for sets of key values
- Use keyed access when number of key values is small
- It can be faster to sort the entire dataset

25

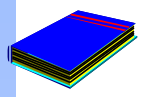
The sort/merge method uses a lot of sorts. On the HP 3000, sort time increases dramatically for large records and large datasets. You will need to think about when it is best to re-sort data for Suprlink.

There are many ways to reduce the sort time. Try using Suprtool's Extract command to reduce the record size. If the dataset has more than 100,000 records, consider using Suprtool's If command to reduce the number of records you select, perhaps with a table drawn from another link file. Chain access with a table is the fastest method to select a small percentage of records.

If you don't logically need selection in one dataset because the basic selection is done in another dataset, experiment with extracting and sorting the entire dataset. This eliminates the time for table lookups and other selection overhead. If the dataset is not too large, this may make the overall task faster than alternate methods because Suprlink can quickly skip over unwanted records.

For Techies

References



Summary of Linking without indexing

- Links sorted data by a common key field
- Drops records that do not match
- Combines records and eliminates duplicate key fields
- Reads and creates self-describing files
- SD files are like little databases
- Suprlink is a standard part of Suprtool

26

Suprlink helps you combine data that is spread across many files. It merges data with a minimum number of commands, and with opportunities for significant improvement in performance.

The procedure is always the same:

1. Use Suprtool to extract and sort from each file or dataset into intermediate disc files
2. Use Suprlink to link the intermediate files into one file
3. Use your favorite report tool to read the final combined file, check for level breaks, and format and print the records

Suprlink uses a special type of file called a self-describing file to hold the temporary, intermediate results. SD files retain a description of their internal data structure inside user labels. Because they are self-describing, subsequent handling of SD files is greatly simplified. You don't have to tell Suprtool or Suprlink anything about the format of the data fields and their positions, or the sorting sequence of the file.

For Techies

References