

## Suprtool Tables, Keys, and Third-Party Indexing

### A Robelle Tutorial

Neil Armstrong, August 1995

Copyright 1995, Robelle Solutions Technology Inc.



1

Although it is usually thought of as a fast tool that serially scans a single file, Suprtool can also manipulate data from multiple files and access search keys. This tutorial is aimed at Suprtool users who want to learn the following tasks:

- Using data from one file as the search criteria for another file (E.g., how to select invoices when the customer is from California, but that field is stored in another file)
- Finding all the records that don't match certain criteria
- Using the Chain command to retrieve records via their key value; finding out why reading the entire dataset is often faster than indexed reads
- Accessing Omnidex or Superdex third-party indexing (TPI) from within Suprtool

*Suprtool Tables, Keys, and Third-Party Indexing* does not cover Suprlink, which is another method of linking files. Among the many Robelle presentations at Interex this year, Suprlink is the subject of a separate tutorial. You can attend the whole series, or select only those topics that interest you.

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](mailto: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

	<u>Page</u>
■ What is an index?	3
■ Pattern-matching	4
■ Table lookup	6
■ Using the \$Lookup function	7
■ Following an index instead of serial scanning	11
■ The Chain command	12
■ Using Chain instead of Get	13
■ Third-party indexing	16

This tutorial will teach you how to use Suprtool to solve a number of common, non-trivial problems.

By showing you real examples, we will point out the Suprtool features used to obtain their results. For instance, the first example scans entries from a TurboIMAGE dataset, and looks for a specific value or word in the product description field.

### For Techies

All sample output was generated using version 3.7 of Suprtool/iX running on an HP 3000 Series 927 (MPE/iX version 5.0).

### References

For complete information on all the options of Suprtool commands (Base, Get, If, Output and Xeq), refer to the *Suprtool User Manual*.

## *What is an index?*

- Work done ahead to make some key searches faster
- TurboIMAGE keys and sort fields, Omnidex/Superdex keys
- Costs = disc space + disc accesses + CPU time
- Index is essential when a user is waiting at a screen
- Index is not necessarily a benefit for batch reports

3

You cannot use an index to answer every conceivable question that users might ask of a database.

For example, what if you want to find all the products that contain the word "Drill" in the product description? Usually, you can install third-party indexing software (Omnidex or Superdex), and keyword index on the description field. This builds a "B-tree" that contains all the words in the product description field, and includes pointers to their entries. Since each additional index adds overhead, this technique is not recommended for fields that are seldom queried.

Alternately, you can serially scan the dataset on the description field, look at each value and keep the records you want. Suprtool is flexible and can search either way. It can take advantage of any existing keys when the circumstances are appropriate, or it can serially scan entire files at high speed.

### **For Techies**

#### **References**

We will examine third-party indexing more closely on page 16.

## How to isolate the products that are drills?

```
>get m-product
>if product-desc=="@Drill@" {pattern-matching}
>extract product-no, product-desc
>sort product-no
>output drills
>list standard
>xeq
May 07, 1994 17:02 Base STORE.DEMO.ROBELLE Set M-PRODUCT
PRODUCT-N PRODUCT-DESC
50511501 Makita 3/8" Var. Speed Drill
50521501 BD 3/8" Variable Speed Drill
50531501 Skil 3/8" Variable Speed Drill
```

4

To find all the drills, you need to search the product description string for the word "Drill". In Suprtool, you can do this by pattern-matching. The pattern-match function is denoted by a double equal sign (==). Patterns are specified in the normal MPE style, and can use the following special characters:

@	zero or more characters of any type
#	exactly one digit
?	exactly one letter or digit
~	zero or more blanks
&	escape character, next character is literal E.g., &@ looks for the @ character
^ and !	reserved for future use; they must be escaped

The m-product dataset holds reference information about each product. In this dataset, there are three records that contain "Drill" in the product description field. Suprtool stores this list in the Drills file. How can we use this list as the selection criteria for another dataset such as the d-sales dataset which contains customer purchase information?

### For Techies

### References

## Use the drill list to breakdown sales by drill

```
>get d-sales
>item sales-total, decimal, 2
>sort product-no
>extract product-no
>table drill-table, product-no, file, drills
>if $lookup(drill-table, product-no)
>duplicate none keys count total sales-total
>list standard
>xeq
May 07, 1994 17:06 Base STORE.DEMO.ROBELLE Set D-SALES
PRODUCT-N    ST-COUNT                ST-TOTAL-1
50511501          2                    336.22+
```

5

In this example, Suprtool produces a list of drills that also shows the number of sales by product number and their total dollar amount.

To only select the drills from d-sales, we access our drill list from the previous slide by using a Table command followed by the \$Lookup function. The Get command reads serially through an entire dataset. To retrieve all the sales records, we use `>get d-sales`. The Item command provides Suprtool with supplemental field information to the TurboIMAGE schema. To format the sales total field with two decimal places, we use `>item sales-total, decimal, 2`. The Sort command rearranges the selected records into sorted order based on the product number. To combine the sales information for the same drill into a single result, we use `>duplicate none keys count total sales-total`.

The Extract command determines which input fields appear in the output record. If no fields are specified, Suprtool extracts all of them. By using `>extract product-no`, Suprtool discards all the fields from the sales record (E.g., customer number, date) except the product number. The List command converts the output records into a printer format listing. It can either replace or supplement an Output command which sends the binary records to an output file. To produce a formatted listing with column headings and page numbers, we use `>list standard`.

The Xeq command checks all the preceding commands. If they define a valid task, Suprtool launches the task into execution.

### For Techies

### References

## Put the drill list into a table

```
>get d-sales
>item sales-total, decimal, 2
>sort product-no
>extract product-no
>table drill-table, product-no, file, drills
>if $lookup(drill-table, product-no)
>duplicate none keys count total sales-total
>list standard
>xeq
May 07, 1994 17:06 Base STORE.DEMO.ROBELLE Set D-SALES
PRODUCT-N    ST-COUNT                ST-TOTAL-1
50511501      2                      336.22+
```

6

The Table command builds a memory table that contains a list of selection values for a specific field. You can have many tables with different fields, and together they cannot exceed 15 megabytes.

For this task, we tell Suprtool to take the values in the product-no field of the Drills file and load them into drill-table. Suprtool takes one value from each record, and uses as many bytes as it needs from the beginning of the record. The memory table stores these values in sorted order for very fast binary searches. Although product-desc is also in the record, Suprtool ignores this field. These are the default settings of the Table command.

### For Techies

### References

Additional features of the Table command are described on page 9.

## Search the drill table on each input sales record

```
>get d-sales
>item sales-total, decimal, 2
>sort product-no
>extract product-no
>table drill-table, product-no, file, drills
>if $lookup(drill-table, product-no)
>duplicate none keys count total sales-total

>list standard
>xeq
May 07, 1994 17:06 Base STORE.DEMO.ROBELLE Set D-SALES
PRODUCT-N    ST-COUNT                ST-TOTAL-1
50511501          2                    336.22+
```

7

By using the Table command, we created and populated the drill-table with a sorted list of drill product-no values.

If you want to select a subset of the input dataset, use the If command as a filter. The \$Lookup function of the If command searches an existing table for the values in the specified input field.

```
>if $lookup(tablename,fieldname)
```

The \$Lookup function returns a True value if the input field is in the table, otherwise it returns a False value. To select input that does not match values in the memory table, you can precede the \$Lookup function with NOT. \$Lookup can also be combined with other tests, or other table lookups by using parentheses or the standard AND/OR operators.

**PERFORMANCE TIP:** If there are multiple conditions in the If statement, put the \$Lookup tests last because they take more CPU time than simple comparisons (E.g., date=\$today). If you do this, Suprtool can skip the time-consuming \$Lookup test when one of the initial tests fail.

### For Techies

### References

## Data consolidation via duplicate removal

```
>get d-sales
>item sales-total, decimal, 2
>sort product-no
>extract product-no
>table drill-table, product-no, file, drills
>if $lookup(drill-table, product-no)
>duplicate none keys count total sales-total
>list standard
>xeq
May 07, 1994 17:06 Base STORE.DEMO.ROBELLE Set D-SALES
PRODUCT-N ST-COUNT ST-TOTAL-1
50511501 2 336.22+
```

8

The Duplicate command is essential when you are dealing with records that have the same key value. Without it, you would get a list of all the drill sales, but it would include many duplicate product numbers instead of a consolidated list. The Duplicate command works with the Sort command to combine records with the same sort value into a single output record, and then appends new fields that count the duplicates and compute numeric sub-totals.

To combine all the sales records for each product into one output line, you can use the None Keys option of the Duplicate command, and remove duplicate key values from the report.

To calculate the number of sales by each drill type, use the Count option with the Duplicate command. This automatically appends a new field (ST-COUNT) to the end of each output record, and inserts an integer value that represents the number of consolidated records.

To calculate the total sales dollars for each drill product, use the Total option of the Duplicate command on the sales-total field. This automatically appends another new field (ST-TOTAL-1) to the output record, and inserts the sum of all the sales-total values from the consolidated records.

### For Techies

### References



## ***Added features of the Table command***

- OUTPUT,LINK creates self-describing (SD) file
- TABLE command can use any field of SD file to fill table
- TABLE <tablename>,<field>,FILE,<filename>,HOLD  
HOLD option retains table in memory at end of current task
- TABLE <tablename>,<field>,SORTED,<filename>,HOLD  
SORTED option tells Suprtool that the file is already sorted

9

If you want to build the table from a field that is not the first field in a record, use a self-describing file (see the *Suprtool User Manual* for details).

If you want to use the same table for several tasks, append the Hold option to the Table command.

If your file is already sorted, specify the Sorted option in the Table command instead of the File option (see the example on page 10).

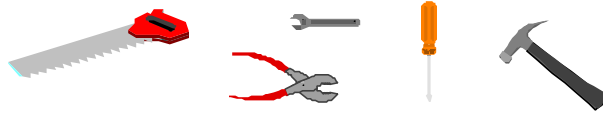
### **For Techies**

### **References**

## What if you want all sales except drills?

- Use NOT \$LOOKUP

```
>get d-sales  
>table drill-table, product-no, sorted, drills  
>sort product-no  
>if not $lookup(drill-table, product-no)  
>output nondrill  
>xeq
```



10

You can select entries that do not match values in the table by putting NOT in front of the \$Lookup function of the If command. This tells Suprtool to discard the input entries whose key values are in the table, and to select those that are not in the table.

### For Techies

### References

## What if there are many sales, but only a few are drills?

- If product-no is a TurboIMAGE index into d-sales, it MAY be faster to retrieve records using the index

```
>table drill-table,product-no,sorted,drills  
>chain d-sales,product-no=drill-table  
>output drillsle  
>xeq
```

- Output file is automatically sorted by product-no

11

Scanning millions of sales records to find 100 drill sales is not very efficient, especially if product-no is an index to the d-sales dataset.

When the Chain command accepts a table name as a parameter, it retrieves each table entry from the dataset. For a master dataset, Chain uses Dbget Mode-7 (hashed), and cannot have more than one entry per key value. For a detail dataset, Chain does a Dbfind and then chained Dbgets (Mode-5) to retrieve all the entries with the same key value. When Chain replaces Get in a Suprtool task, it processes the retrieved entries the same way as Get (E.g., Sort, If, Extract, etc.). The only difference between Chain and Get is the initial method of entry retrieval.

With master datasets or details that only have one entry per key value, the Chain command has another nice benefit. *Since the table is already sorted, so is the extract file!*

### For Techies

### References

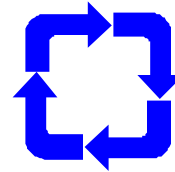
## *The Chain command*

- CHAIN <dataset>, <keyfield> = <values>

```
>chain d-sales, product-no=drill-table
```

```
>chain m-product, product-no=50511501,50521501
```

```
>chain m-product, product-no=drill-table
```



12

The Chain command works on both detail and master datasets, and can use any of the search keys defined for the dataset.

You can either specify the list of required key values directly in the command, or indirectly by referring to a table of sorted values. If you use the command method, it limits you to a relatively short list of values.

### **For Techies**

### **References**

## Part 1 of Get versus Chain

- What if the table or the selected subset is large?

- GET versus CHAIN

```
>table clist,cust,file,custfile
>get d-lines                                >chain d-lines,cust=clist
>if $lookup(clist,cust)
>sort cust
>output myfile
>xeq
```

- Which method is faster? What determines the answer?

13

Suprtool has two commands that read records from datasets: Get and Chain. The Get command reads the entire dataset serially at high speed, and selects records based on the criteria in your If command. The Chain command uses regular IMAGE calls to read records based on the IMAGE index values. It can also use the If command to specify the selection criteria for non-index fields.

If you want to select records based on index values, the Chain command may seem like the obvious choice. It may not necessarily be the right choice because the Get command is more efficient in some situations. How do you decide which command to use?

Let's look at the example on this slide, and compare the Get command to the Chain command. In this task, we can use the list of customer numbers in Custfile to select line item entries from the d-line dataset. The Get and Chain methods use the same logic, and achieve the same results except the sort order of detail records with the same customer number is different. With Chain, the records are in the order defined for the IMAGE path, either chronological or sorted. With Get, they are in the same order as they appear physically in the dataset, effectively random. If you need a second sort field in the Get example, you should specify it explicitly with another Sort command.

### For Techies

### References

## Part 2 of Get versus Chain

- The number of records makes a difference in performance
- Assume the following:
  - d-line has 2.3 million records of 308 bytes each
  - Ordfile contains 162,000 customer numbers
  - There are 261,000 matching d-line records
- How many disc reads for the CHAIN method?

14

Chain uses the standard IMAGE calls of Dbfind and Dbget (Mode-5, chained). Since there are 162,000 customer numbers, there will be 162,000 calls to Dbfind, and each call will probably need at least one disc read because Dbfind uses a pseudo-random hashing method to find master records. Unless you have just "packed" the dataset along this path, the Dbget calls will also use one disc read per record retrieved. Packing a dataset means putting records with the same key value in adjacent physical locations. It only provides a temporary improvement in efficiency until you add or delete more records.

The Chain method uses the following number of disc reads:

162,000	disc reads for Dbfinds
+ 261,000	disc reads for Dbgets
423,000	total disc reads

Since Suprtool sorted the customer numbers when it loaded them into the table, the Chain command reads them in the correct order. The time required to write the output file is negligible because Suprtool transfers the records into large buffers, and performs a single file write for many records.

### For Techies

### References

## Part 3 of Get versus Chain

- How many disc reads for the GET method?
  - 50,000 bytes per disc read
  - 308 bytes per record and 2.3 million records
  - 14,197 disc reads versus 423,000 for CHAIN
  
- Surprisingly, GET is much faster than CHAIN

15

In this example, the Get command retrieves many contiguous records when it reads 50,000 bytes with each disc access. More precisely, it reads 162 records with each disc read (50,000 bytes divided by 308 bytes), and only requires 14,197 disc accesses to read the entire 2.3 million record dataset.

Compared to the 423,000 disc reads with the Chain method, that's a 98.7 % saving in I/O. It's true that the Get method must sort the records and do table lookups, but even with these additional steps, it is still much more efficient than the Chain method.

While it may be more intuitive to do a chained read when there are index values to the required records, it is often more efficient to read the whole dataset sequentially and simply discard the unwanted records. When in doubt, use the Get command with Set Statistics On to read the dataset sequentially. If the number of reported Input Fread Calls is less than the number of records selected, Get will be faster than Chain.

### For Techies

### References

## *Two ways to find customers whose names start with "M"*

- Suprtool can search two ways:

Without TPI

```
>get m-customer
```

```
>if name-last=="M@"
```

```
>sort name-last
```

```
>out mcust
```

```
>xeg
```

With TPI (Omnidex or Superdex)

```
>chain m-customer,name-last="M@"
```

```
>out mcust
```

```
>xeg
```

- TPI adds advanced keys such as Indexed Sequential keys
- TPI is an HP interface to third-party products

16

Omnidex is a product of Dynamic Information Systems Corporation (DISC), and Superdex is a product of Bradmark Technologies, Inc. Both of these products supplement the regular hashed and chained indexing of TurboIMAGE by providing Sorted Sequential and Keyword search capabilities for TurboIMAGE data. MPE/iX 4.0 has added an interface that updates the Omnidex and Superdex indexes. If you enabled third-party indexing (TPI) using DBUTIL and are adding, deleting or modifying records, IMAGE automatically calls the TPI intrinsics to update the indexes. TPI also allows extended search capabilities through the Dbfind intrinsic.

```
:run dbutil.pub.sys  
>>enable store for indexing  
>>exit
```

Suprtool is TPI-aware and provides extra features to take advantage of the standard interface to Omnidex and Superdex. In this example, we can scan the entire dataset, and check each last name for a pattern that starts with the letter "M" followed by any character. If name-last is a TPI search field, we can also use the pattern with the Chain command.

### For Techies

### References



## ***Benefits of TPI in Suprtool***

- CHAIN command passes search expression to TPI
- CHAIN accepts TPI keys, even though not IMAGE keys
- CHAIN accepts TPI keys, even though not IMAGE fields
- PUT, DELETE, UPDATE, and EDIT commands update TPI indexes
- FORM shows TPI keys and product identification

17

Since Suprtool is TPI-aware, it provides extra functionality. You can use TPI indexes to retrieve records the same way as standard IMAGE indexes. Remember that the presence of a TPI index on a field does not necessarily mean that every TPI retrieval is faster than a Suprtool serial Get. If you are retrieving over 25% of the records, a simple serial scan and sort may actually be faster.

TPI restrictions in Suprtool:

1. The Chain command accepts only keys of type X or U (byte type, not numeric). TPI itself can also handle numeric types of keys.
2. The maximum size of a Chain search value is the declared size of the search field. Some retrievals may exceed this length (E.g., using AND/OR logic).
3. The Suprtool Chain command uses Dbfind in Mode-1. TPI provides other search capabilities that use different modes, but they are not supported in Suprtool version 3.7.

**For Techies**

**References**

## Retrieving with non-IMAGE search fields

- TPI allows you to add hidden fields to sets for indexing
- These fields are not part of the official IMAGE entry
- These fields do not appear in the Dbschema or many other tools
- These fields are not stored in the official IMAGE dataset
- Suprtool recognizes these non-IMAGE index fields  
`>chain d-inventory,si-item-desc = "D@"`

18

Omnidex and Superdex have capabilities that go beyond those of TurboIMAGE. They allow you to construct dataset indexes from any part of the record without restricting you to the official IMAGE fields.

TPI provides extra Dbinfo calls that recognize these indexes. Suprtool's Form command uses these calls to display the non-IMAGE index fields after the regular IMAGE fields. Since Dbfind also accepts these non-IMAGE index fields, Suprtool's Chain command has been enhanced to recognize these fields, and pass them on to Dbfind.

### For Techies

### References

## Can I use TPI to find all the drills?

- Earlier, we did a serial GET with pattern-matching  
`>if prod-desc == "@Drill@"`
- Can a TPI CHAIN command access a keyword index?
  - Superdex: Yes, if prod-desc is a Keyword path.  
`>chain m-product,prod-desc="Drill@"`
  - Omnidex: For non-key items, yes. For key items, it depends on the position of the word and the number of wildcard characters.

19

The TPI interface is standardized for only some of the Omnidex and Superdex functions. While the framework may be standardized, what you put in the search string value may differ between products. To do the equivalent of a pattern-match search, you have to use a Keyword path in TPI.

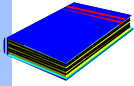
Can Suprtool do keyword searches using the TPI interface?

- With Superdex: Yes, if the field is defined as a Keyword path.
- With Omnidex: For non-key items, yes. For key items, it depends on the following:

Search Pattern	First Word	Other Words
string	no	no
string@	yes	no
@string@	yes	yes

### For Techies

### References



## ***Summary of Suprtool Tables, Keys, and Third-Party Indexing***

- Tables use data from one file to select records from another file
- CHAIN uses IMAGE keys to retrieve specific key values
- CHAIN can accept a table as the list of selection criteria
- CHAIN is slower than serial GET if you want a large percent of records
- CHAIN supports third-party indexing (TPI)

20

Thank you for attending our Suprtool tutorial. If you have any comments or suggestions about this tutorial, or suggestions for future tutorials, please pass them on to Mr. Green.

This tutorial explored the ordinary ways of dealing with multiple files and multiple search requirements. Suprtool provides another way to select and link multiple files. It is called Suprlink and it is extraordinary. Suprlink does not depend on tables, IMAGE indexes, or TPI indexes, but it can use them when it is appropriate.

### **For Techies**

### **References**