Suprtool/Open 6.0

Change Notice

by Robelle Solutions Technology Inc.

Robelle Solutions Technology Inc.
Suite 372, 7360 137 Street
Surrey, BC Canada V3W 1A3

Phone: 604.501.2001
Support: 289.480.1060

E-mail:sales@robelle.com
E-mail:support@robelle.com
Web: www.robelle.com

Contents

Introducing Suprtool Version 6.0 5

Overview 5
Highlights in Suprtool 6.0 5
Highlights in Suprtool 5.9 6
Highlights in Suprtool 5.8 6
Highlights in Suprtool 5.7 7
Highlights in Suprtool 5.6 7
Compatibility 7
Documentation 7

Installation 9

Overview 9
Installation Instructions 9
Installation Assistance 9

Enhancements in Version 6.0 10

Introduction 10
HP-UX Data Files 10
ENDIANINT 10
ENDIANLOG 10
FFISBE 11
Sdlinux Utility 11
Set SDOutBE 13
BackwardChain 13
$INRECNUM 13
$LEADZEROZ 14
$LEADZEROB 15
$JUSTIFYL 15
$JUSTIFYR 16
$RESPACE 16

Enhancements in Version 5.9 19

Introduction 19
$Month 19
Excel Command 19
Json Output 21
Multiple Json Commands 22
MySQL Access 22

Enhancements in Version 5.8 24

Input ($first/$last) 24
Linkmgr 25
Stexpmgr 25

Set ComLog 25
$Proper 27
$Translate 28

Enhancements in Version 5.6 31

Extract Command 31
Data Items Support 32
$SubCount 32

Bugs Fixed 33

Bugs Fixed In Suprtool 5.9 33
Bugs Fixed In Suprtool 5.6.11 33
Bugs Fixed In Suprtool 5.6.10 33
Bugs Fixed In Suprtool 5.6 33

Introducing Suprtool Version 6.0

Overview

Suprtool/Open is a new version of Suprtool designed to be platform independant. Suprtool/Open is designed to read, select, and sort data from Oracle, and Eloquence databases and data files with fixed-length records. Suprtool/Open is designed to be similar to Suprtool for MPE and Suprtool for HP-UX.

Suprlink/Open provides high-speed data-file linking based on a sort key. Use STExport to convert fields in a self-describing input file into an output file that can be imported into different applications.

Highlights in Suprtool 6.0

- The List command now has a NOSAMETO option to turn off the SAMETO feature.
- Set Backwardchain On, will cause the Chain command to do a backward chained read.
- STExport has a new set command called Set Excel Leadzero On which tells the Excel command to add leading zeroes to the fields specified in the Excel Preserve command.
- Suprtool has a new function available to the if/extract commands called $inrecnum, which expects a double integer result.
- Suprtool has a new function called $leadzeroz, which will add leading zeroes to a display field and will optionally justify the field.
- Suprtool has a four new string handling functions, specifically, $justifyl, $justifyr, $leadzerob and $respace.
- Suprtool / Open now has the ability to read Self-Describing files from HP-UX natively on Linux. Suprtool / Open can also create Self-Describing files for use on HP-UX with Set SDOutIsBe On.
- Suprtool / Open now has the ability to read Flat files from HP-UX and natively handle Big Endian Integer and Logical fields with Set FFISBE On.
- Suprtool / Open can handle and create BigEndian Logical fields with Set EndianLog On.

- A new utility to convert the version number and field information for HP-UX Self-Describing files to be used on Linux natively called sdlinux.
- Output,num would not work if the input source was a self-describing or a flat file.
- An Output file would have a null written in the last byte incorrectly if the output file was an uneven byte length and if a sort was specified.
- Sorts with multiple keys and large records would fail.
- Suprtool, STexport and Suprlink no longer prints a message if there is an exported JCW variable, that is greater than >32767 or less that -32767.
- Output,num on a flat file wouldn't output the numbers in BE format if set Endianint was set to BE.
- Output=Input, did not parse properly and created an output file called =Input (or similar). The parsing was fixed and is now functioning properly on Linux and other Small Endian platforms. (Build 12)
- Set SDOUTISBE, was mistakenly set to have a default value of on, the default is now off as of Build 13.
- Two new variables allow for easier setup of Dynamic Loading of the Oracle Libraries and MYSQL Libraries.
- Writing of an Output file to an NFS device would fail with Permission Denied when over-writing to an existing file.
- Suprtool would fail with File System err 13, Permission Denied reading from the Rsortscr file in certain circumstances. (Build 18)

Highlights in Suprtool 5.9

- A new $month function in the if / extract option allows you to add or subtract a number of months from a given date.
- STExport has a new command called Excel command which allows you to format a field in a format that allows leading zeroes or spaces to be preserved.
- A new option called JSON will output SD data in Java Script Object Notation.
- Suprtools new banner would show the incorrect day and day of week in certain circumstances. This was cosmetic only.
- The Add command would fail if a Table in an Oracle database accessible by a given username had more than 2.1 billion entries on Oracle 11 and higher.
- Suprtool/Open now has the option to read MySQL databases. We are looking for feedback on where to take this feature.

Highlights in Suprtool 5.8

- Set CleanChar to a single character, after a Set CleanChar "<null>" would not come into effect due to the special null flag not being reset. This was in both Suprtool and STExport.

- The input command now accepts the keywords($first/$last) when inputting a range of records.

Highlights in Suprtool 5.7

- Suprlink now has it's own "Suprmgr" file. Suprlink will process all the commands in /opt/robelle/linkmgr on startup.
- STExport now has it's own "Suprmgr" file. STExport will process all the commands in /opt/robelle/stexpmgr on startup.
- Set Comlog On has been added to Suprtool, Suprlink and STExport to log all commands entered in Suprtool, Suprlink or STExport are logged in it's own file.
- Suprtool would potentially have problems with system commands in variable-length scripts if the command ended in a number that had a length of eight numbers.
- Suprtool would fail with a write error in the case of an Input SD file with BigEndian Fields, Set EndianInt BE and if the output was to an SQL Table via the Add command and the output was sorted.
- Suprtool would lose track of Table data field information, on tables held with the hold option in subsequent tasks.
- Suprtool now has the $proper function which will shift the first character in a string and any first character after a space or ampersand.
- Suprtool now has the Translate command and a $translate function to obfuscate test data or any byte field from being readable.
- Suprtool/Open 5.7 Build 6 and higher supports the Oracle 12 client.
- Suprtool/Open would not parse a negative number into a quad integer container.
- Suprtool for Itanium and Suprtool/Open would not properly convert negative, single and double integers when output/,display is invoked.
- Suprtool for Itanium and Suprtool/Open incorrectly reallocated if/extract code space on subsequent tasks, which would eventually cause Suprtool to fail with the error, "Unable to allocate heap space."

Compatibility

Suprtool/Open first release is designed to be compatible with Suprtool for HP-UX.

Documentation

The user manual contains the full description of all the Suprtool suite of products including Dbedit, Suprlink, STExport, and Suprtool2, as well as usage tips and commands for each. The manuals are up-to-date with all the latest changes. To see only the changes in the latest version, see the "What's New" section of the manual.

You can download our manuals and Change Notices in various formats and order printed (hardcopy) manuals from our web site at:

http://www.robelle.com/library/manuals/.

Installation

Overview

The following instructions describe the installation process of a new Suprtool release. The new version overwrites an existing version of Suprtool on your system.

Installation Instructions

There are typically two main types of installations. The first and most often utilized is the Download instructions. You can find the download install instructions here:

http://www.robelle.com/downloads/install-soprod.html

Installation Assistance

If you have any questions or run into any problems, please call us. Technical support is available on weekdays from 7 a.m. to 3 p.m. Pacific time at 1.289.480.1060. Technical support can also be obtained via e-mail at: support@robelle.com

Enhancements in Version 6.0

Introduction

Suprtool is constantly being updated with new features. The following section describes the new enhancements to Suprtool since Suprtool 5.9

HP-UX Data Files

Suprtool/Open now has the ability to read self-describing files that come directly from HP-UX. Suprtool/Open will depending on the version of the SD file format will when necessary, convert all of the SD information from BigEndian data to Little Endian numbers such that Suprtool/Open will be able to natively read the files.

Sdlinux is a new utility available for HP-UX that will help aid in allowing Suprtool/Open to know whether or not an SD file is bigendian or smallendian.

ENDIANINT

Suprtool/Open now has the ability to read and write Self-Describing files with Integers that are in BigEndian format.

Set EndianInt BE

Will tell Suprtool to write out any Integer in BigEndian byte order on Linux on a small endian machine.

ENDIANLOG

Suprtool/Open now has the ability to read and write Self-Describing files with Logicals that are in BigEndian format.

Set EndianLog BE

Will tell Suprtool to write out any Logical in BigEndian byte order on Linux on a small endian machine.

FFISBE

Suprtool/Open now has the ability to read Flat Files with BigEndian Data. If SetFFISBE is turned On, Suprtool will assume that Integer and Logical fields are in BigEndian format.

Sdlinux Utility

Suprtool now has a utility called sdlinux, which will help convert an HP-UX self-describing file which will have Big Endian numbers, in the Self-Describing information and in the integers in the data file. The sdlinux utility is available in the Suprtool 6.0 HP-UX download or you can ftp the utility from our ftp site by doing the following from you HP-UX system:

```
cd /opt/robelle/bin
ftp ftp.robelle.com
anonymous
youremail@domain.com
binary
get sdlinux
quit
```

Please note that you need Suprtool 6.0 version of Suprtool/Open, but there is no dependency, on what version of Suprtool you are using on HP-UX. The utility was just distributed with the 6.0 version as it was the next available release vehicle.

Sdlinux, will change the sd version field which tells Suprtool/Open the endianness of the sd file.

SD Version	Endianness	Extended Names
B.00.00	BIG (HP-UX) LITTLE(LINUX)	NO
B.00.01	BIG (HP-UX) LITTLE(LINUX)	YES
B.00.02	BIG	NO
B.00.03	BIG	YES

The sdlinux utility has four options, -f, -r, -h and –d. Only one option can be specified at any given time:

Option	SD Version	Integer Field
-f	B.00.00 becomes B.00.02 and B.00.01 becomes B.00.03	Any integer/logical field gets flagged as BE.
-r	B.00.02 becomes B.00.00 and B.00.03 becomes B.00.01	Any integer/ logical field gets the Endianness flag turned off.
-h	B.00.02 becomes B.00.00 and B.00.03 becomes B.00.01	No effect.

-d	No effect.	Any integer/logical field gets the Endianness flag turned off.

The sdlinux utility gets run with the above options and a filename for an argument as in:

```
./sdlinux '-ffilename'
./sdlinux '-rfilename'
./sdlinux '-hfilename'
./sdlinux '-dfilename'
```

The filename specified just needs to be the data file name not the sd filename. Regardless, sdlinux will figure out what to do if the .sd extension is in the filename argument.

Please see the following example how you can move data from HP-UX, to Linux natively without having to export to ascii. On HP-UX, you simply create a self-describing file as you normally would:

```
Base mydb
Get mydataset
Out outfile,link
Xeq
```

Then you can use sdlinux to make some quick changes to the Self-describing information, specifically the version in the header and the integers and logicals get updated with the big endian flag being turned on:

```
sdlinux '-foutfile'
```

You can then ftp the data file and the sd file over to your Linux box:

```
ftp linuxbox.robelle.com
user
password
put outfile outfile
put outfile.sd outfile.sd
quit
```

You can then reverse the effects of the –f option with the –r option on the files after you have transferred:

```
sdlinux -routfile
```

Then on Linux you can read the sd file natively even though the sd information has big endian information and the data file can be read with bigendian integers and logicals.

```
>in outfile
>form
    File: outfile     (SD Version B.00.02)  Has linefeeds
       Entry:                    Offset
          CHAR-FIELD           X5      1
          INT-FIELD            I1      6                                BE
          DBL-FIELD            I2      8                                BE
          PACKED-FIELD         P12    12
          PACKED*-FIELD        P12    18
          QUAD-FIELD           I4     24                                BE
          ID-FIELD             I1     32                                BE
          LOGICAL-FIELD        K1     34                                BE
          DBLLOG-FIELD         K2     36                                BE
          ZONED-FIELD          Z5     40
          FILLER               X36    45
    Entry Length: 80  Blocking: 1
>num 3
>list
>xeq
>IN outfile (0) >OUT $NULL (0)
CHAR-FIELD      = 11111          INT-FIELD       = 1111
DBL-FIELD       = 11111          PACKED-FIELD    = +11111
PACKED*-FIELD   = +11111         QUAD-FIELD      = 11111
ID-FIELD        = 1              LOGICAL-FIELD   = 1111
DBLLOG-FIELD    = 11111          ZONED-FIELD     = 11111
FILLER          =
```

Set SDOutBE

The Set command, SDOutBE when turned on Suprtool/Open on a Little Endian Linux box will create an SD file with BigEndian data such that it can be transferred to an HP-UX box and read natively on HP-UX.

When introduced, this feature was turned on, but has since been turned off in Build 13.

BackwardChain

The Set command, Backwardchain when turned on will tell the Chain command to do a Backward Chained read.

$INRECNUM

The if / extract commands can now utilize a new function called $INRECNUM, which allows you to use the input record number in certain tasks. For example, the task below would find record number 11.

```
    In somefile
    If $inrecnum=11
```

You can also, utilize the $inrecnum function in the extract command:

```
    In somefile
    Def recnum,1,4,double
    Ext recnum=$inrecnum
```

The $inrecnum function was designed to find records especially in the instance where duplicate records are in a particular data source and cannot be isolated by any other means.

$LEADZEROZ

The if / extract commands can now utilize a new function called $LEADZEROZ, which allows you to add leading zeroes to a specific display field. This was designed specifically for the extract command and fixing up data but can be used in the if command as well.

$NUMBER is capable of fixing up numbers, but the new $LEADZEROZ function is more lightweight and simply adds leading zeroes, and has an option to justify right as shown below. The source data looks like this:

```
/PRINT LEADZERO
1
 2
  3
   4
    5
     6
      7
       8
      9
10
12
12345
220
```

You can format with the following:

```
>IN LEADZERO.SUPRTEST
>DEF A,1,12,DISPLAY
>EXT A=$LEADZEROZ(A,J)
>out *
>xeq
000000000001
000000000002
000000000003
000000000004
000000000005
000000000006
000000000007
000000000008
000000000009
000000000010
000000000012
000000012345
000000000220
IN=13, OUT=13. CPU-Sec=1. Wall-Sec=1.
```

The $LEADZEROZ function cannot fix issues like commas and decimal places in a display field, this can be handled by the $number function.

$LEADZEROB

The if / extract commands can now utilize a new function called $LEADZEROB, which allows you to add leading zeroes to a specific byte field. This was designed specifically for the extract command and fixing up data but can be used in the if command as well. The data looks like this:

```
/PRINT LEADZERO
1
 2
  3
   4
    5
     6
      7
       8
      9
10
12
12345
220
```

You can clean it up with the following:

```
>IN LEADZERO.SUPRTEST
>DEF A,1,12,byte
>EXT A=$LEADZEROB(A,J)
>out *
>xeq
000000000001
000000000002
000000000003
000000000004
000000000005
000000000006
000000000007
000000000008
000000000009
000000000010
000000000012
000000012345
000000000220
IN=13, OUT=13. CPU-Sec=1. Wall-Sec=1.
```

$JUSTIFYL

The if / extract commands can now utilize a new function called $JUSTIFYL, which allows you to left justify text to the left side of a field.

```
>in leadzero
>def b,1,12,byte
>ext b=$justifyl(b)
>out *
>xeq
1
2
3
4
5
6
7
8
9
10
12
12345
220
IN=13, OUT=13. CPU-Sec=1. Wall-Sec=1.
```

$JUSTIFYR

The if / extract commands can now utilize a new function called $JUSTIFYR, which allows you to right justify text to the right side of a field.

```
>in leadzero.suprtest
>def a,1,12,byte
>ext a=$justifyr(a)
>out *
>xeq
           1
           2
           3
           4
           5
           6
           7
           8
           9
          10
          12
       12345
         220
IN=13, OUT=13. CPU-Sec=1. Wall-Sec=1.
```

$RESPACE

The if / extract commands can now utilize a new function called $RESPACE, which allows you to fixup byte data that has multiple spaces in between text. For example your data looks like this:

```
>in respace.suprtest
>def text,1,40
>ext text
>out *
>xeq
this  is  a  test
 this is a        test
this  is  a   test
this         is a test
this   is   a   test
this is a test
 please note this   is   a     test
this  is   a   test
this    is      a      test
this is a test
       this   is   a   test
 this is a test
this    is    a    test
this is a test
dummy   record
this is a  not   test
silly   record
IN=30, OUT=30. CPU-Sec=1. Wall-Sec=1.
```

It can easily be fixed up and converted to what is shown below:

```
>in respace.suprtest
>ext text=$respace(text,J)
>out *
>xeq
this is a test
this is a test
this is a test
this is a test
this is a test
this is a test
please note this is a test
this is a test
this is a test
this is a test
this is a test
this is a test
this is a test
this is a test
dummy record
this is a not test
silly record
IN=30, OUT=30. CPU-Sec=1. Wall-Sec=1.
```

ORACLE and MYSQL Dynamic Loading

Two new variables have been implemented to enhance the dynamic loading of the Oracle and MySQL client libraries. ROBELLE_ORACLE_LIB and ROBELLE_MYSQL_LIB can now be set to the directory where the libraries reside:

```
export ROBELLE_ORACLE_LIB=/home/neil/myoracle/lib32/
export ROBELLE_MYSQL_LIB=/home/neil/mysql/lib32/
```

When these variables are set Suprtool will take the variable and add on the respective necessary client name and dynamically load what is needed. Suprtool first will try

the LD_LIBRARY_PATH, but if that fails it will specifically use the above variables if set to dynamically load the library.

Enhancements in Version 5.9

Introduction

Suprtool is constantly being updated with new features. The following section describes the new enhancements to Suprtool since Suprtool 5.8.10

$Month

The if / extract commands can now utilize a new function called $Month, which will add a given number of months to a given date in the format of ccyymmdd or yyyymmdd.

For Example:

```
In somefile
Item mydate,date,ccyymmdd
Def targetdate,1,4,double
Ext targetdate=$month(mydate,+4)
```

The above task will take the field mydate and add four months to it. Suprtool will check if the date is valid and adjust the date within reason. For example if the given month for mydate has 31 days and the day is 31, and the month mydate becomes when the date is added to has only 30 days. The date will be adjusted to have the 30th for the day.

Excel Command

The Excel command can be used to produce columns of data that when imported will preserve spaces or leading zeroes.

EXCEL PRESERVE <*fieldname*>

Example

STExport can generate columns that are imported into Excel in such a way that leading zeroes are preserved. While the format produced is not traditional CSV, the format will produce a field in the form:

```
="00055555"
```

This form when imported into Excel will preserve the leading zeroes. In order to invoke this format the Excel command has very simple syntax:

```
$in filexcel
$col fixed
$quote double
$zero leading
$excel preserve newchar int-field
$out *
$xeq
```

These simple commands will generate a file that will have the usually formatted fields as well as some fields formatted specifically for preserving spaces and leading zeroes in Excel.

The result of such an STExport task will look as follows:

```
="    11111 ",=" 01111", 0000011111,+00000011111
="    11111 ",=" 02222", 0000022222,+00000022222
```

Json Output

The JSON command specifies STExport to generate Json output. Use the JSON to produce Java Script Object Notation documents for either Internet or Intranet applications.

JSON

OBJECT "string"

ONEPERLINE

Example

STExport can generate JSON output with just a few commands.

```
$input file1sd
$JSON
$output myJSON
$xeq
```

These four simple commands will generate a file that can be read by various applications. The result of such an STExport task will look as follows:

[{"CHAR-FIELD":"11111","INT-FIELD":1111,"ZONED-FIELD":11111}]

Object

The Object option allows the JSON data to be wrapped in a specific Object description.

```
JSON Object "Json object"
```

Looks like this:

```
{"Json object":
[{\"CHAR-FIELD":\"11111",
\"INT-FIELD"\:1111,
\"DBL\-FIELD"\:11111,
\"PACKED\-FIELD"\:+1111,
\"PACKED\.-FIELD"\:+11111,
\"QUAD\-FIELD"\:11111,
\"ID\-FIELD"\:1,
\"LOGICAL\-FIELD"\:111,
\"DBLLOG\-FIELD"\:11111,
\"ZONED\-FIELD"\:11111
}]
}
```

Note that the example of the Output has one field per line with data. Normally this would have to be specified via the command line but the data is shown this way simply due to space constraints.

OnePerLine

For files that have many fields you may want to consider using the OneLine option of the JSON command:

```
JSON OnePerLine
```

STExport will put each field and data on one line with the appropriate beginning and end notation.

```
[{\"CHAR\-FIELD"\:"11111",
\"INT\-FIELD"\:1111,
\"DBL\-FIELD"\:11111,
\"PAC\KED\-FIELD"\:+11111,
\"PAC\KED\.-FIELD"\:+11111,
\"QUAD\-FIELD"\:11111,
\"ID\-FIELD"\:1,
\"LOG\ICAL\-FIELD"\:1111,
\"DBL\LOG\-FIELD"\:11111,
\"ZONED\-FIELD"\:11111
}]
```

Multiple Json Commands

You can enter multiple JSON commands per task to set the JSON options you require.

```
$in file1sd
$JSON Object "Json object"
$JSON OnePerLine
$out *
$xeq
```

An example of the output generated by the above commands is as follows:

```
{"Json object":
[{"CHAR\-FIELD"\:"11111",
\"INT\-FIELD"\:1111,
\"DBL\-FIELD"\:11111,
\"PACKED\-FIELD"\:+11111,
\"PACKED\.-FIELD"\:+11111,
\"QUAD\-FIELD"\:11111,
\"ID\-FIELD"\:1,
\"LOGICAL\-FIELD"\:1111,
\"DBLLOG\-FIELD"\:11111,
\"ZONED\-FIELD"\:11111
}]
}
```

MySQL Access

Suprtool/Open now has the ability to read MySQL databases via the Open and Select commands. We are looking for alpha, beta testers and feedback specifically on how

to treat certain data items. Please e-mail Neil Armstrong at neil@robelle.com to request a trial of this software.

Enhancements in Version 5.8

Input ($first/$last)

Suprtool now has $first and $last mnemonics, which can be used on a range selection of records on the input command. It was designed to handle a request to list the last N number of records in a file as in:

```
Input somefile($last-10/$last)
```

Suprtool will parse the Range selection and semantically check if the record range entered is logical. For instance, $first-2 and $last+10, which do not make logical sense would throw and error. Similarly if a record only has 5 records in it then $last-10 or $first+7, would also throw an error.

Enhancements in Version 5.7

Linkmgr

Suprlink now has a “suprmgr” file similar to Suprtool. Suprlink will process all the commands found in /opt/robelle/linkmgr or $ROBELLE/linkmgr.

Stexpmgr

STExport now has a “suprmgr” file similar to Suprtool. STExport will process all the commands found in /opt/robelle/stexpmgr or $ROBELLE/stexpmgr.

Set ComLog

Suprtool, Suprlink and STExport can now log all commands in each product. Every command will be logged if the setting Set ComLog, is on. Commands will not be logged if the user does not have write access to the comlog file. There are three separate log files, one for each product. The stlog file for Suprtool, sllog for Suprlink and sxlog for STExport. The files are by default located in /opt/robelle/log/suprtool directory. If the $ROBELLE variable is set, the log files would be in the directory $ROBELLE/log/suprtool.

The format of the log file is as follows:

DateTime, UID, PID, command entered

The Date and Time is 16 bytes in the format Year, Month, Day, Hours, Minutes, Seconds.

UID/PID are six digit numbers, followed by the command with a line feed at the end.

Please note that if this command is global the log files can expand quickly, taking up disc space by default in the /opt/robelle directory. You can redirect where the commands are logged by using linked files.

```
ln -s /log/robelle/suprtool/stlog /opt/robelle/log/suprtool/stlog
ln -s /log/robelle/suprtool/sxlog /opt/robelle/log/suprtool/sxlog
ln -s /log/robelle/suprtool/sllog /opt/robelle/log/suprtool/sllog
```

The user running any of the products must have write access to the log files whether Link files or directly.

$Proper

Suprtool now has the $proper function which will shift to upper case the first character of a byte type field and after any space or ampersand. It will also shift to lower case any other characters in the byte-field.

```
>in mprod
>list stan
>xeq
Apr 29, 2014 13:00                File: mprod                Page 1

PRODUCT-DESC

skil 3/8" variable speed drill
b&d router
skil var. sp. auto-scroll saw
skil 8 1/2" circular saw
b&d cordless screwdriver
makita 8 1/4" circular saw
b&d variable speed jigsaw
makita 1/2" router
makita 3/8" var. speed drill
skil router
b&d 7 1/4" circular saw
b&d 3/8" variable speed drill
makita 1" jigsaw
```

Considering the following data, you can fix all of the product names with one simple task:

```
>in mprod
>ext product-desc=$proper(product-desc)
>list stan
>xeq
May 01, 2014 11:40                File: MPROD                Page 1

PRODUCT-DESC

Skil 3/8" Variable Speed Drill
B&D Router
Skil Var. Sp. Auto-Scroll Saw
Skil 8 1/2" Circular Saw
B&D Cordless Screwdriver
Makita 8 1/4" Circular Saw
B&D Variable Speed Jig Saw
Makita 1/2" Router
Makita 3/8" Var. Speed Drill
Skil Router
B&D 7 1/4" Circular Saw
B&D 3/8" Variable Speed Drill
Makita 1" Jig Saw
```

Note that any character after a space, “&”, or “-“ is upshifted for a proper name. Suprtool will also downshift those characters that do not qualify as needing proper capitalization and it is a capital character, the proper function will downshift those characters. See an example below:

```
>IN NAME
>LIST
>XEQ
>IN NAME.NEIL.GREEN (0) >OUT $NULL (0)
NAME            = NEIL ARMSTRONG

>IN NAME
>EXT NAME=$PROPER(NAME)
>LIST
>XEQ
>IN NAME.NEIL.GREEN (0) >OUT $NULL (0)
NAME            = Neil Armstrong
```

The $proper function only works on byte type fields.

$Translate

Suprtool now has a $translate function which in conjunction with the translate command allows you to build a translation table, whereby you can translate from any byte character to any character. We have also added a method to a supplied translate table which will allow you to obscure the data such that it can't be read.

```
>in newprod
>list
>xeq
>IN NEWPROD.NEIL.GREEN (0) >OUT $NULL (0)
PRODUCT-DESC    = Skil 3/8" Variable Speed Drill

>IN NEWPROD.NEIL.GREEN (1) >OUT $NULL (1)
PRODUCT-DESC    = B&D Router

>in newprod
>translate tounread
>ext product-desc=$translate(product-desc)
>out unread,link
>xeq
IN=13, OUT=13. CPU-Sec=1. Wall-Sec=1.

>in unread
>num 1
>list
>xeq
>IN UNREAD.NEIL.GREEN (0) >OUT $NULL (0)
PRODUCT-DESC    = Hzxo .2)? Epcxpqot Hatts Rcxoo
Warning:  NUMRECS exceeded; some records not processed.
IN=2, OUT=1. CPU-Sec=1. Wall-Sec=1.

>in unread
>translate toread
>ext product-desc=$translate(product-desc)
>list
>xeq
>IN UNREAD.NEIL.GREEN (0) >OUT $NULL (0)
PRODUCT-DESC    = Skil 3/8" Variable Speed Drill

>IN UNREAD.NEIL.GREEN (1) >OUT $NULL (1)
PRODUCT-DESC    = B&D Router
```

You can make your own Translate table using the Translate command, where you can specify the character you want to translate and what you want to translate to, using Decimal Notation. So if you want to translate "A" to "Z", you would type the command:

```
Translate "^65:^90"
```

So you specify the from character on the left in decimal which is the capital-A and the to-character is also in decimal format which is capital-Z, which is decimal 90. If you want to reverse the translation you can simply do the following command and translate the field back with:

```
Translate "^90:^65"
```

This is not meant to be an encryption solution, but it will help obfuscate test date really quickly.

Enhancements in Version 5.6

Extract Command

Suprtool's extract command now has three new keywords, which can be used for extract range feature. You can now say extract $all, extract $first / $last on an SD file or Image/Eloquence dataset. The intention is to make your scripts more easily maintained. If you had a script that you wanted to put a sequence number at the beginning and then extract the rest of the dataset you previously had to specify the starting field and the ending field. For example if the first field in a dataset was order-no and the last field was pst-code you may have a script that looked like this.

```
base orddb
get customers
def seq-no,1,4,double
ext seq-no=$counter
ext order-no / pst-code
out newfile,link
xeq
```

If you added any fields to the beginning or end of the dataset you would have to re-write the script. Now you can write the script as being:

```
base orddb
get customers
def seq-no,1,4,double
ext seq-no=$counter
ext $all
out newfile,link
xeq
```

You can also write the script using $first / $last as your preference, but $first and $last are also useful if you need to add data into the middle of the fields you extract:

```
base orddb
get customers
def seq-no,1,4,double
ext $first / zip
ext seq-no=$counter
ext tax-code / $last
out newfile,link
xeq
```

Please note that if a self-describing file has a fieldname that is a duplicate field and one of the duplicate fields is the last field in the file, then $first / $last and $all, will only extract up to the first occurrence of the duplicate fieldname. This may seem as an issue but it is consistent with what Suprtool does currently with extract from a range. Currently and prior to the $first / $last enhancement Suprtool would have

extracted only up to the first occurrence of the field if you had a file such as this:

```
>form
File: newfile     (SD Version B.00.00)  Has linefeeds
       Entry:                      Offset
          CHAR-FIELD           X5      1
          INT-FIELD            I1      6
          DBL-FIELD            I2      8
          PACKED-FIELD         P12    12
          PACKED*-FIELD        P12    18
          QUAD-FIELD           I4     24
          ID-FIELD             I1     32
          LOGICAL-FIELD        K1     34
          DBLLOG-FIELD         K2     36
          ZONED-FIELD          Z5     40
          FILLER               X36    45
          FILLER               X36    81
    Entry Length: 116  Blocking: 1
```

Notice that FILLER is a duplicate named field, so if you entered, extract char-field / filler, Suprtool would only extract up to and including the first FILLER field. For consistency, extract $first / $last behaves the same way.

Data Items Support

Suprtool now supports 512 data items in both Eloquence datasets and SD files and Oracle fields. STExport and Suprlink have also been updated to support the increased number of data items in the SD fields that they read, but please note that other limitations still exist. [5.5.10]

$SubCount

$SubCount has been added to provide a counter that only gets reset at a given sort break.

```
In file1sd
Sort char-field
Def control-count,1,4,double
Ext $first / $last
Ext control-count=$subcount(char-field)
Out newfile,link
xeq
```

What Suprtool will do in this case is start incrementing a number starting with 1, and increase by 1 for any given char-field value. This way you can add a counter based on a sort break for a given field.

Bugs Fixed

Bugs Fixed In Suprtool 5.9

Add Command. The Add command would fail if a Table in an Oracle database accessible by a given username had more than 2.1 billion entries on Oracle 11 and higher.

Bugs Fixed In Suprtool 5.6.11

Output , Display Command. Suprtool for Itanium and Suprtool/Open would not properly convert negative, single and double integers when output/,display is invoked

Quad Integer Input Parsing. Suprtool/Open would not parse a negative number into a quad integer container.

Bugs Fixed In Suprtool 5.6.10

If Command. Suprtool for Itanium and Suprtool/Open incorrectly reallocated if/extract code space on subsequent tasks, which would eventually cause Suprtool to fail with the error, "Unable to allocate heap space."

Bugs Fixed In Suprtool 5.6

Arithmetic Expressions. Suprtool/Open had problems with arithmetic expressions with brackets for Packed-Decimal and Zoned-Decimal data types.

Arithmetic Expressions. Suprtool/Open had problems with rounding for arithmetic expressions with Quad Integers, Packed-Decimal and Zoned-Decimal data types.