
Suprtool 6.0 for HP-UX:

Change Notice

by Robelle Solutions Technology Inc.



Program and manual copyright © 1981-2018 Robelle Solutions Technology Inc.

Permission is granted to reprint this document (but not for profit), provided that copyright notice is given.

Qedit and Suprtool are trademarks of Robelle Solutions Technology Inc. Oracle is a trademark of Oracle Corporation, Redwood City, California, USA. Other product and company names mentioned herein may be the trademarks of their respective owners.



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	5
Highlights in Suprtool 5.8	6
Highlights in Suprtool 5.7	6
Highlights in Suprtool 5.6	7
Itanium vs PA_RISC	7
Compatibility	9
Itanium Compatibility	10
CPU Serial Number (uname)	10
Documentation	10
Installation	11
Overview	11
Installation Instructions	11
Installation Assistance	11
Enhancements in Version 6.0	12
Introduction	12
\$List NOSAMETO	12
BackwardChain	12
\$INRECNUM	12
\$LEADZEROZ	13
\$LEADZEROB	13
You can clean it up with the following:	14
\$JUSTIFYL	14
\$JUSTIFYR	15
\$RESPACE	15
SDLINUX UTILITY	16
Enhancements in Version 5.9	19
Introduction	19
\$Month	19
Excel Command	19
Json Output	20
Multiple Json Commands	21
Enhancements in Version 5.8	23
Introduction	23
Input (\$first/\$last)	23
Introduction	24

Linkmgr.....	24
Stexpmgr	24
Set ComLog	24
\$Proper.....	25
\$Translate.....	26

Bugs Fixed **29**

Bugs Fixed In Suprtool 5.8.11	29
Bugs Fixed In Suprtool 5.8.10	29
Bugs Fixed In Suprtool 5.7.11	29
Bugs Fixed In Suprtool 5.7	29
Bugs Fixed In Suprtool 5.6.11	29
Bugs Fixed In Suprtool 5.6	30
Bugs Fixed In Suprtool 5.5.11	30
Bugs Fixed In Suprtool 5.5.04	30
Bugs Fixed In Suprtool 5.5.03	30
Bugs Fixed In Suprtool 5.5.02	30
Bugs Fixed In Suprtool 5.5.01	30
Bugs Fixed In Suprtool 5.5	30

Introducing Suprtool Version 6.0

Overview

Use Suprtool/UX to read, select, and sort data from Oracle, Allbase and Eloquence databases and data files with fixed-length records. Suprtool/UX is designed to be similar to Suprtool for MPE while providing necessary HP-UX features.

Suprlink/UX 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.
- The Add command to add data to an Oracle table would fail on some tables with an abort if there were a large number of tables in the database.

Highlights in Suprtool 5.9

- A new function in Suprtools if/extract engine called \$month is capable of adding or subtracting months for 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.

- 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.
- Suprtools new banner would show the incorrect day and day of week in certain circumstances. This was cosmetic only.

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.
- The list command Reclen verb will now impact the Record size for stdout, where previously it was ignored unless a **FILE** was specified.
- Suprtool would lose track of Table data field information, on tables held with the hold option in subsequent tasks.

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 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 on both HP-UX platforms now correctly calculates how many records are in a file loaded via the Table command when the file is a greater than 2Gb.
- 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.”
- Suprtool is now compatible with Oracle 12 version of the client. The support begins with Suprtool 5.7 Build 6.

Highlights in Suprtool 5.6

- Suprtool for Itanium and Suprtool/Open did not properly handle Display type sorts with over punch characters.
- Suprtool, Suprlink and STExport now handle 512 fields on HP-UX and Open platforms.
- Suprtools' extract command now has three special keywords to help with extract using a range, which is designed to make your Suprtool scripts more easily maintained. Suprtool now has \$all, \$first and \$last which respectively means all fields, the first field or the last field. Note that \$first and \$last may only be used in a range extract.
- \$subcount has been added to keep a running count for a given sort break.
- The limit of the Select command in Suprtool has been increased to 8192 bytes. (Suprtool 5.6 Build 11)
- Suprtool on Itanium had trouble with Arithmetic expressions with multiple sets of brackets when dealing with Packed-Decimal and Display data.
- Suprtool would use an incorrect field definition when copying a file where the input file had a duplicate field name with different field definition, the second field would utilize the first definition.
- Suprtool for HP-UX would potentially incorrectly set the decimal or date attribute on a straight copy of an SD file. This was introduced by the fix to copying SD files with duplicate fieldnames and was in 5.5.04 pre-release and 5.5.11 and 5.6 Build 1. This is fixed in Suprtool 5.6 Build 2.
- Suprtool would give an incorrect result when doing an arithmetic expression with brackets and the data involved is Packed-Decimal. (Itanium only Fixed Suprtool 5.6 Build 11)
- Suprtool and STExport would incorrectly assume a positive sign for a packed-decimal field that had an invalid sign value, when converting from Decimal to Ascii. (Itanium only)
- Suprtool for AMXW did not allocate enough space to support the same number of defined fields and items as previous versions.
- Set Ifouroutput on extracts would fail at xeq time with an "Unknown if execute phase" error in the PA_RISC version, when an alignment issue would occur in some circumstances.

Itanium vs PA_RISC

There are two types of machines that HP has that run HP-UX, the older machines are based on the PA_RISC chip and the new machines that use the Itanium (and Itanium 2 chips). There are also program files that are specific to each chip. PA_RISC programs can run on both PA_RISC and Itanium machines, however, Itanium programs can only run on Itanium machines.

Any program that runs on Itanium must utilize libraries that match its own program type. Therefore if a PA_RISC program runs on Itanium and looks for a library, it

must find/load the PA_RISC version of the libraries that it needs and conversely an Itanium program must load the libraries specific to its type.

How to Tell what Machine I am on?

There are many ways to tell what kind of machine you are on, here are a couple:

```
uname -m
getconf MACHINE_MODEL
```

In each case the Itanium machine will typically say ia64 somewhere in the result of the above two commands. PA_RISC machines typically say "9000" somewhere in the string.

What is the Program File Type?

Since there are two types of machines there are also at least two types of program files. Suprtool is primarily either PA_RISC 1.1 or ELF-32 / IA64 which is also known as Itanium. To determine the type of program file you can use the file command:

Itanium

```
file suprtool
suprtool:      ELF-32 executable object file - IA64
```

PA_RISC

```
file suprtool
suprtool:      PA-RISC1.1 executable dynamically linked -not stripped
```

The PA_RISC binary can run on both platforms but you need to have PA_RISC libraries for Eloquence or Oracle if you are using the Oracle module. An Itanium binary will not run on a PA_RISC machine and will fail with "cannot execute".

Of course we've made a provision to identify if you are running the ia version of Suprtool in the banner and verify command with the ia after the UX :

```
SUPRTOOL/UXia/Copyright Robelle Solutions Technology Inc. 1981-2009.
(Version 5.2 Internal) MON, JUN  8, 2009, 11:26 AM Type H for help.
```

PA_RISC Loading

Suprtool by default looks in the libraries in any of the directories named in SHLIB_PATH. For example to insure that Suprtool resolves the library loads you can set the SHLIB_PATH system wide in your /etc/profile file in the following manner:

```
export SHLIB_PATH=/opt/eloquence6/lib/pa11_32:/opt/oracle/lib
```

If you do not have the SHLIB_PATH variable set to a value where the libraries can be found, Suprtool will then try to load from their default names locations. For Eloquence the default location is :/opt/eloquence6/lib/pa11_32.

The Oracle interface will first try to load libclntsh.so and then attempt to load libclntsh.sl from \$ORACLE_HOME/lib32/libclntsh.sl.

Itanium Loading

You can change Suprtool to pay attention to LD_LIBRARY_PATH and use dlopen to load libraries by setting the environment variable:

```
export ROBELLE_DYN_LOAD =Y.
```

It is generally advisable to use DLOPEN or the ROBELLE_DYN_LOAD option when using the Itanium version. The PA_RISC version of Suprtool can use either type of loading, the default or the dlopen method invoked with ROBELLE_DYN_LOAD.

The Itanium version will make one more attempt at loading the Eloquence libraries from `hpux32/libimage3k.sl` and `hpux32/libeqdb.sl` as most users are setting `SHLIB_PATH` and `LD_LIBRARY_NAME` pointing to the `lib` directory.

Running the PA_RISC version on Itanium:

You can of course run the `PA_RISC` version of `Suprtool`, however, given that the `PA_RISC` version requires `SHLIB_PATH` to point to `PA_RISC` libraries may go against the configuration necessary for other programs that may need `SHLIB_PATH` to point to Native Itanium libraries.

The solution to this is fairly simple, you can create a command file (let's say `/opt/robelle/cmd/suprtool`) that runs `suprtool` for you, of course you need to insure your path variable has `/opt/robelle/cmd` before `/opt/robelle/bin`.

You can then have the following commands in the `/opt/robelle/cmd/suprtool` command file:

```
export SAVED_SHLIB_PATH=$SHLIB_PATH
export SHLIB_PATH=/opt/oracle/pa_risc/lib:/opt/eloquence/lib/pa11_32
/opt/robelle/bin/suprtool $@
export SHLIB_PATH=$SAVED_SHLIB
```

Essentially all it does save the current `SHLIB_PATH` setting, sets it to point to `PA_RISC` libraries, runs `suprtool` and then resets the library back.

Suprtool for Itanium

Now that you know all about `PA_RISC` and Itanium you do need to be aware of what `Suprtool` for Itanium is in terms of new code and risk for potential issues. `Suprtool` for Itanium is completely in C code. This does not mean that it is all new C code however, the majority of the `Suprtool` code is still based on `SPL`, but is converted to C, using Allegro's `SPLASH` compiler.

So while the C code is new it is based on the original `SPLash` code, however, given differences between C and `SPLash` not all code can be automatically converted in some cases. We worked closely with Allegro to find ways to have the source code work and address the various issues that came up. The entire time Allegro improved the `SPLash` to C converter.

So while we are based on the original `Suprtool` we do have new code paths in the Native Itanium version and other potential issues where the C code might not work in exactly the same manner as the `SPLash` code.

Having said that we have tested this against our test suites, and improved and added a large number of tests to our `HP-UX` test suites.

Compatibility

`Fastread` option is now set to `Off` for `Suprtool` for `HP-UX`. The Dynamic loading version of `Suprtool` the option is set to `off`.

`Suprtool` production versions 5.1 thru 5.4.01 would incorrectly double the record size for an `output,data` file when either `Set Ifouroutput` or `Set Oracle Integer` were on. This has been fixed but if your script accounted for a doubled record size this may cause an issue as we no longer incorrectly write out double the bytes. If you want to have the old bug behaviour to be re-instated you can `Set Oracle DataBug On`. The default for this setting is `off`.

Itanium Compatibility

There are a couple of differences in the Itanium version of Suprtool, which you need to be aware of.

The DBEdit module is not available in the Itanium versions of Suprtool. This module is invoked by the EDIT command.

The layout for the Verify command is slightly different then the PA-RISC versions of Suprtool.

CPU Serial Number (uname)

This program runs only on CPUs whose serial numbers have been encoded (the "uname" on HP-UX). If it fails to run and you get an "invalid HPSUSAN" error message, contact Robelle for assistance, via support@robelle.com or the support number at 1.289.480.1060.

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.

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 HP-UX 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 HP-UX download install instructions here:

<http://www.robelle.com/downloads/install-sxprod.html>

Tape installation instructions can be found here:

<http://www.robelle.com/support/install/tape/sxprod.html>

Installation Assistance

If you have any questions or run into any problems, please call us. Technical support is available on weekdays from 8 a.m. to 4 p.m. Pacific time at 1.289.480.1060.

Technical support can also be obtained via e-mail at: support@robelle.com If your new version of software will not run, you can page someone from technical support by calling the support number, or you can typically easily run extend with the disaster option to tide you over until business hours. Instructions for this are available at:

<http://www.robelle.com/disaster/>

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/>.

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

\$List NOSAMETO

The List command now has a NOSAMETO option to turn off the SAMETO feature.

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
00000000220
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 *
>xeg
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 *
>xeg
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.suptest
>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.suptest
>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
  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.suptest
>ext text=$respace(text,J)
>out *
>xeg
this is a test
please note this is a test
dummy record
this is a not test
silly record
IN=30, OUT=30. CPU-Sec=1. Wall-Sec=1.

```

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.

`Sdlinux`, is a small utility that 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
<code>-f</code>	B.00.00 becomes B.00.02 and B.00.01 becomes B.00.03	Any integer/logical field gets flagged as BE.
<code>-r</code>	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.
<code>-h</code>	B.00.02 becomes B.00.00 and B.00.03 becomes B.00.01	No effect.
<code>-d</code>	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:

```
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
  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
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          =

```

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

\$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 ccyyymmdd or yyyyymmdd.

For Example:

```
In somefile
Item mydate,date,ccyyymmdd
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 fileexcel
$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 filelsd
$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,
\ "PACKED-FIELD"\": +11111,
\ "PACKED.-FIELD"\": +11111,
\ "QUAD-FIELD"\": 11111,
\ "ID-FIELD"\": 1,
\ "LOGICAL-FIELD"\": 1111,
\ "DBLLOG-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  
  }]  
}
```

Enhancements in Version 5.8

Introduction

Suprtool is constantly being updated with new features. The following section describes the new enhancements to Suprtool since Suprtool 5.7.11

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 an 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

Introduction

Suprtool is constantly being updated with new features. The following section(s) describes the new enhancements to Suprtool since Suprtool 5.6.

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, Username, UID, PID, command entered

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

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
>xeg
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
>xeg
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. (5.6.12)

\$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)? Epcxpgot 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 data really quickly. (5.6.12)

Bugs Fixed

Bugs Fixed In Suprtool 5.8.11

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 5.8.10

Banner. The opening banner at runtime would display the wrong day and day of month. This was only an issue in Suprtool 5.8.

Bugs Fixed In Suprtool 5.7.11

Set CleanChar. 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.

Bugs Fixed In Suprtool 5.7

STExport/Suprlink. PA_RISC versions of Suprlink and STExport would fail when attempting to report an error on an OUTPUT file if the user specified a directory that didn't exist or the user didn't have write permission to.

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

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

Bugs Fixed In Suprtool 5.5.11

Over Punch Sort. Suprtool would not properly sort display type data with Over punch characters. (Itanium and Open platforms only)

Bugs Fixed In Suprtool 5.5.04

SD File Copy. Suprtool would use an incorrect field definition on a straight copy of an SD file if the input file had a duplicate field name with different byte lengths. (HP-UX only)

Bugs Fixed In Suprtool 5.5.03

Extract Command. Suprtool would give an incorrect result when doing an Arithmetic Expression that has more than one set of brackets as in: `extract amt-p=(packed-field + packed-field) - (packed-field *2)` (Itanium only)

Bugs Fixed In Suprtool 5.5.02

Decimal to Ascii. Suprtool and STExport would incorrectly assume a positive sign for a packed-decimal field that had an invalid sign value, when converting from Decimal to Ascii. (Itanium only)

Bugs Fixed In Suprtool 5.5.01

Ifouroutput. Set `Ifouroutput` on extracts would fail at xeq time with an "Unknown if execute phase" error in the PA_RISC version, when an alignment issue would occur in some circumstances.

Bugs Fixed In Suprtool 5.5

\$edit function. The `$edit` function did not work well with alpha based edit masks and if the target was greater than 32 characters.

Output,data. Suprtool would write out double the calculated record size when set `oracle integer` was on or set `ifouroutput` was on, and `output,data` option was used and no extract commands were specified and the input source was an Oracle table.

List Command. The List command on Suprtool for HP-UX on PA_RISC would not print the specified headings and title for list command to a file.

Sorting. Suprtool for HP-UX on PA_RISC would abort when sorting a file with an output record size of 39 bytes and only on some machines, with `fastread` on.