
Suprtool/Open 6.0.10

Change Notice

by Robelle Solutions Technology Inc.



Program and manual copyright © 1981-2019 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 6.0.10	5
Overview	5
Highlights in Suprtool 6.0.10.....	5
Highlights in Suprtool 6.0.....	5
Highlights in Suprtool 5.9.....	6
Compatibility	7
Documentation.....	7
Installation	8
Overview.....	8
Installation Instructions.....	8
Installation Assistance	8
Enhancements in Version 6.0.10	9
Introduction.....	9
ENDIANIEEE.....	9
Eloquence B.08.30.....	9
STEXPORT	10
SUPRLINK.....	10
Enhancements in Version 6.0	11
Introduction.....	11
HP-UX Data Files	11
ENDIANINT.....	11
ENDIANLOG.....	11
FFISBE	12
Sdlinux Utility.....	12
Set SDOutBE	14
BackwardChain.....	14
\$INRECNUM.....	14
\$LEADZEROZ.....	15
\$LEADZEROB.....	16
\$JUSTIFYL.....	16
\$JUSTIFYR	17
\$RESPACE.....	17
ORACLE and MYSQL Dynamic Loading.....	18
Enhancements in Version 5.9	20
Introduction.....	20
\$Month.....	20
Excel Command.....	20
Json Output	22
Multiple Json Commands	23

MySQL Access	23
--------------------	----

Bugs Fixed	24
-------------------	-----------

Bugs Fixed In Suprtool 5.9	24
Bugs Fixed In Suprtool 5.6.11	24
Bugs Fixed In Suprtool 5.6.10	24
Bugs Fixed In Suprtool 5.6	24

Introducing Suprtool Version 6.0.10

Overview

Suprtool/Open is a new version of Suprtool designed to be platform independent. 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 6.0.10

- Suprlink now supports 8192 bytes on both the input file and the link file.
- Suprlink now supports 16384 bytes on the output file in Suprlink
- Stexport now supports 8192 bytes on both the input file and output files.
- Suprtool now supports Eloquence B.08.30, and the following options for BYTEORDER are also supported, NATIVE, COBOL, BE, BE2, BEDATA, BEDATA2

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.

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.

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 8 a.m. to 4 p.m. Pacific time at 1.800.453.8970.

Technical support can also be obtained via e-mail at: support@robelle.com

Enhancements in Version 6.0.10

Introduction

Suprtool is constantly being updated with new features. The following section describes the new enhancements to Suprtool since Suprtool 6.0.

ENDIANIEEE

Suprtool now supports IEEE fields to optionally be written out in Big Endian Byteorder when the setting:

Set EndianIEEE BE

will tell Suprtool to write out any IEEE field in BigEndian byte order on Linux on a small endian machine. This setting automatically gets turned on when accessing an Eloquence B.08.30, database when the BYTEORDER is set to BE2 or BEDATA2.

Eloquence B.08.30

Suprtool now supports Eloquence B.08.30 database which has a new feature whereby you can pick the BYTEORDER of any I/J/K and IEEE fields. There are several options for the BYTEORDER and they are NATIVE, COBOL, BE, BE2, BEDATA AND BEDATA2. See the table below to:

BYTEORDER	I/J/K Data	Ieee Data	Image Area
NATIVE	NATIVE	NATIVE	NATIVE
COBOL	BE	NATIVE	BE
BE	BE	NATIVE	BE
BE2	BE	BE	BE
BEDATA	BE	NATIVE	NATIVE
BEDATA2	BE	BE	NATIVE

What Suprtool does to support this is takes note of the option and sets it's own internal flag(s) appropriately and then turns off the translation from Eloquence so the data isn't translated twice. These options in both Suprtool and Eloquence is to support certain COBOL compilers that have Integers and Logicals of all sizes in Big Endian ByteOrder.

STEXPORT

STExport now supports 8196 bytes on the input, and the output file.

SUPRLINK

Suprlink now supports 8196 bytes on the input file, link file and 16384 bytes on the output file.

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
  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          =

```

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 *
>xeg
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 *
>xeg
000000000001
000000000002
000000000003
000000000004
000000000005
000000000006
000000000007
000000000008
000000000009
000000000010
000000000012
000000012345
00000000220
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 *
>xeg
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 *
>xeg
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
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_PATH and ROBELLE_MYSQL_PATH 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 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:

```
= " 1111 ",=" 0111", 00001111,+000001111
=" 1111 ",=" 0222", 00002222,+000002222
```

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,
  "PAC\KED\FIELD": +11111,
  "PAC\KED\.-FIELD": +11111,
  "QUAD\FIELD": 11111,
  "ID\FIELD": 1,
  "LOGICAL\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 filelsd
$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.

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.