
Suprtool/Open 6.3.50

Change Notice

by Robelle Solutions Technology Inc.



Program and manual copyright © 1981-2021 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 6.3.50	5
Overview	5
Highlights in Suprtool 6.3.50	5
Highlights in Suprtool 6.3	6
Highlights in Suprtool 6.2	6
Highlights in Suprtool 6.1	7
Highlights in Suprtool 6.0	7
Compatibility	8
Documentation.....	8
Installation	9
Overview	9
Installation Instructions.....	9
Installation Assistance.....	9
Enhancements in Version 6.3.50	10
Introduction	10
Set SDItemNoExt	10
Set INSDBypass	10
Enhancements in Version 6.2	11
Introduction	11
Rport Set Pagelen.....	11
Rport Subtotal.....	11
Rport Set Maxlen.....	11
Rport Comma	11
Rport Hide	12
System Commands	12
\$Month(\$stddate())	12
Rport.....	12
\$BOM and \$EOM.....	14
String Function Checks	14
Sdlinux for Linux.....	15
Set Limits Override On Off	15
Enhancements in Version 6.1	17
ENDIANIEEE	17
Eloquence B.08.30.....	17
STEXPORT	18
SUPRLINK.....	18
\$Length	18
\$LENGTH (Works on Byte-type and numeric fields).....	18
If Usage:	18

Extract Usage:.....	18
\$ROBELLE variable.....	18
Set EditSignNeutral	18

Enhancements in Version 6.0 **20**

HP-UX Data Files	20
ENDIANINT	20
ENDIANLOG.....	20
FFISBE.....	20
Sdlinux Utility	21
Set SDOutBE.....	23
BackwardChain.....	23
\$INRECNUM	23
\$LEADZEROZ.....	24
\$LEADZEROB.....	25
\$JUSTIFYL	25
\$JUSTIFYR.....	26
\$RESPACE	26
ORACLE and MYSQL Dynamic Loading	27

Bugs Fixed **29**

Bugs Fixed In Suprtool 6.3.01.....	29
------------------------------------	----

Introducing Suprtool Version 6.3.50

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. Rport created simple reports with subtotals and totals from self-describing files.

Highlights in Suprtool 6.3.50

- An input of a non-sd file with defines and extracts would assume for BigEndian to Little Endian as a Self-Describing File. (6.3.50 Build 2)
- Rport spacing and alignment has been improved for subtotal and total lines.
- Rport spacing of detail lines has been improved to allow for a space after each byte field.
- A subsequent Suprtool task with large key sorts would fail when the previous step was a step that accessed an Oracle database and the Oracle database was still open. (Suprtool 6.3.01 Build 2)
- Suprtool no longer requires that you need to extract a field from a Flat file (ffisbe on) or Self-Describing file in order for comands like if, sort, dup none keys total field, and total, to correctly convert the data from Big Endian to Little Endian. (Build 14)
- Suprtool / Open didn't handle double logical numbers over 2 billion when converting to ascii in the list command. (Linux Only Fixed in Build 2)
- Suprmgr files wouldn't get read properly in Suprtool/Open 6.3 (Build 2)
- Sorting with certain files would not allocate enough memory space to complete the sort. (Fixed in Build 5)

- An if \$lookup would not work all times with certain data fields if the data field was Big Endian. (Fixed in Build 5)
- A Chain with multiple values specified inline with byte or logical field types would fail. (Fixed in Build 5)
- Input files with Bigendian data are converted to Little endian about three times faster in all cases. (Build 5)
- Output,ascii and Stexport, conversion of double integers with decimal places would incorrectly convert, numbers where the value was less than the number of decimal places. (Build 14)
- Output, ascii and Stexport would incorrectly handle single logical conversions for some values.
- Output,ascii would incorrectly pad uneven byte output files with a null on the first record. (Build 14)
- Suprtool has a new command called set sдинbypass on, allows the reclen, If syntax on an input file if it is self describing. This was done to allow some customers to not have to change a number of scripts.

Highlights in Suprtool 6.3

- STExport Heading fieldnames limit has been increased to 14336 bytes.
- The Suprtool suite of products has had more 2027 mitigation work done to prepare for the end of 20271231.
- Suprtool would incorrectly do a conversion on input of a file with Big Endian data if there was an extract of a constant into an integer field (Linux Only)
- Non-MPE versions of Suprtool didn't update the Item information, decimals and date like the MPE version. When an extract command wasn't entered but an Item command referenced a specific field. You can turn this on with set sдitemnoext on, but the default is off to have it be compatible with all versions of Suprtool/Open and HP-UX. You can put it in your suprmgr file, to enable it globally.
- Suprtool would incorrectly think it could sort a file in memory, as opposed to using sort scratch files in an extremely rare case. (Linux)

Highlights in Suprtool 6.2

- Set Pagelen Command has been added to Rport.
- The subtotal feature of Rport will now break on multiple sort levels in the Self-Describing file.
- Rport has a new command called Hide which is intended to not report on a field in an SD file.
- Rport has a new command called Comma, which will place appropriate commas in an ascii field that was converted from a numeric field.
- There is a new setting in RPORT that is called set maxlen on, which calculates the maximum size that a field can be, including commas and dollar signs.

- Suprtool/Open now defaults to /bin/bash when a users shell is not found in the user information.
- The \$eom/\$bom function wouldn't work properly when \$stddate was nested in a \$month function and the to month had 31 days.
- Sdlinux version is available for Linux to convert the .sd file for use on Linux is now available on Linux. You can now directly copy a data file and sfile from HP-UX and convert it for use on Linux.
- Two new sub-functions of \$BOM and \$EOM are now available in the \$stddate and \$month functions. The \$BOM function returns the starting day of the month and the \$EOM returns the last date of the month for a given month.
- String functions in Suprtool now have better length and bounds checking of the 4095-byte string limit.
- Rport a new simple report writer has been added to the Suprtool suite of products.
- Set Limits Override has been added to the Suprtool for MPE version. (6.1.01)

Highlights in Suprtool 6.1

- Suprtool has a new function called \$length that will return a double integer value of the length of a byte and/or numeric field.
- The \$ROBELLE variable has been expanded to 58 characters.
- Suprtool has a new option called Set EditSignNeutral which tells the \$edit function to treat Neutral zoned and packed fields to be positive and treated as such by the \$edit function.
- 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.

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, Rport 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 6.3.50

Introduction

Suprtool is constantly being updated with new features. The following section describes the new enhancements to Suprtool since Suprtool 6.2

Set SDItemNoExt

Non-MPE versions of Suprtool didn't update the Item information, decimals and date like the MPE version does, when an extract command wasn't entered. You can turn this off with set sditmnoext off, but the default is on to have it behave in the same manner as the MPE version. (Suprtool 6.3 Build 1)

Set INSDBypass

If this option is turned on, Suprtool will bypass the check on the Recsize and LF parameters on the input command. Normally, Suprtool will throw an error if the Recsize and LineFeed are specified on the Input command **and** the Input file is a Self-Describing file.

If you have Set InSDBypass on, Suprtool will not throw an error on and use the Self-Describing information. This was done to allow people to add ,link output to certain Suprtool Tasks and not have to adjust the input command of a subsequent step. (Suprtool 6.3 Build 12)

Enhancements in Version 6.2

Introduction

Suprtool is constantly being updated with new features. The following section describes the new enhancements to Suprtool since Suprtool 6.1.10.

Rport Set Pagelen

Rport will Page Break by default at 24 lines, you can set a Page Break to any integer value greater than 10. When Rport reaches the number of lines for a page the Titles and headings are re-printed and Rport continues on printing, detail and subtotal lines.

Rport Subtotal

Rport will follow all levels of sort specified in the Self-Describing file, and will produce a subtotal line when each level break is made.

Rport Set Maxlen

Rport will by default calculate the maximum size of a field, including, commas, decimals and dollar signs. The previous version just took into account, decimals. This is on by default, you can just use the old calculation by adding set maxlen off to your rportmgr file.

Rport Comma

Rport has a new command called comma, where you can specify a fieldname or list of fieldnames to tell Rport to add commas to a given numeric field, when it is reported on and converted to ascii. The same rules will apply for a given fields subtotal and total field. This command can only be entered after the input command.

Rport Hide

Rport has a new command called Hide which when given a fieldname will remove that field from the report. The report will be adjusted accordingly and also the Hide command must be entered after the input command.

System Commands

If the user id isn't found or a shell name isn't returned the Open version of Suprtool suite of products will assume a shell of /bin/bash.

\$Month(\$stddate())

The combination of nested \$stddate within \$Month would handle months with 31 days incorrectly if the from date was a month with less than 31 days and the to month was a month with 31 days.

Rport

All versions of Suprtool, MPE, HP-UX (PA-Risc and Itanium), and Suprtool / Open have a new component called Rport, which is a simple report writer.

The code below represents one of the more complicated reports and shows some of the attributes that you can control. Rport works off of self-describing files and is available on MPE, HP-UX (PA-Risc and Itanium) and on all Linux platforms that Suprtool/Open runs on.

The Title command divides the lines into quadrants, primarily, left, center and right and two lines are supported. The heading column option prints the Heading information specified in each column. You can also specify Heading Fieldnames, where Rport will divide the field name into each column, and you can do your own layout for each Heading Line:

head 1	Add	"	CharIntege	Double	Packed	Packed*	Quad"
head 1	Add	"	IdLogica	DbfLog	Zoned"		
head 2	add	"	Field Field	Field	Field	Field	Field"
head 2	add	"	Field Field	Field Field"			

The Size command tells Rport the maximum size of the column for the data. By default, Rport picks the maximum size that a field will be when converted to Ascii, as shown in the following table:

Field Format	Output Size
I1, J1	6 bytes
I2, J2	11 bytes
I3, J3	16 bytes
I4, J4	20 bytes
K1	5 bytes
K2	10 bytes
E2	12 bytes

E4	23 bytes
R2	12 bytes
Zn	n+1 bytes
Pn	n bytes

The integer specified beside the fieldname is the length for the field to use for the report. If the data overflows the field will be filled with asterisks. The Subtotal and Total commands are pretty self-explanatory.

```

in file1.rporttst
title line 1 left RPNAME "SampReport"
title line 1 center RPCOMPANY "Robelle Solutions"
title line 1 right rppageno tag "Page No : "
title line 2 left rpdate tag "Date : "
title line 2 right rpruntime tag "Run Time : "
head 1 col "Char"
head 2 col "Field"
head 1 col "Integer"
head 2 col "Field"
head 1 col "Double"
head 2 col "Field"
head 1 col "Packed"
head 2 col "Field"
head 1 col "Packed*"
head 2 col "Field"
head 1 col "Quad"
head 2 col "Field"
head 1 col "Id"
head 2 col "Field"
head 1 col "Logical"
head 2 col "Field"
head 1 col "DblLog"
head 2 col "Field"
head 1 col "Zoned"
head 2 col "Field"
size int-field 8
size dbl-field 8
size packed-field 7
size packed*-field 7
size quad-field 7
size logical-field 7
size dbllog-field 7
size zoned-field 8
subtotal int-field dbl-field packed-field packed*-field quad-field id-field
subtotal logical-field dbllog-field zoned-field
total int-field dbl-field packed-field packed*-field quad-field id-field
total logical-field dbllog-field zoned-field
out REPORT01
xeq

```

The above code creates the following report. The report shows that it can handle most data types (the field names are the data types) and provides a simple layout and automatically aligns the subtotals and the total lines. The subtotalling works off the sorted field, (in this case char-field) and prints a subtotal line each time the char-field changes.

SampReport	Robelle Solutions	Page No : 1
Date : 20191024	Run Time : 12:45	
Char Integer Double Packed Packed* Quad Id Logical DblLog Zoned		
Field Field Field Field Field Field Field Field Field		

```

11111 11.11 111.11 +111.11 +111.11 111.11 1 11.11 111.11 111.11
11.11 111.11 +111.11 +111.11 111.11 1 11.11 111.11 111.11
22222 22.22 222.22 +222.22 +222.22 222.22 1 22.22 222.22 222.22
22222 22.22 222.22 +222.22 +222.22 222.22 2 22.22 222.22 222.22
44.44 444.44 +444.44 +444.44 444.44 3 44.44 444.44 444.44
33333 33.33 333.33 +333.33 +333.33 333.33 1 33.33 333.33 333.33
33333 33.33 333.33 +333.33 +333.33 333.33 2 33.33 333.33 333.33
33333 33.33 333.33 +333.33 +333.33 333.33 3 33.33 333.33 333.33
99.99 999.99 +999.99 +999.99 999.99 6 99.99 999.99 999.99
44444 44.44 444.44 +444.44 +444.44 444.44 1 44.44 444.44 444.44
44444 44.44 444.44 +444.44 +444.44 444.44 2 44.44 444.44 444.44
44444 44.44 444.44 +444.44 +444.44 444.44 3 44.44 444.44 444.44
44444 44.44 444.44 +444.44 +444.44 444.44 4 44.44 444.44 444.44
177.76 1777.76 +1777.76 +1777.76 1777.76 10 177.76 1777.76 1777.76
55555 55.55 555.55 +555.55 +555.55 555.55 1 55.55 555.55 555.55
55555 55.55 555.55 +555.55 +555.55 555.55 2 55.55 555.55 555.55
55555 55.55 555.55 +555.55 +555.55 555.55 3 55.55 555.55 555.55
55555 55.55 555.55 +555.55 +555.55 555.55 4 55.55 555.55 555.55
55555 55.55 555.55 +555.55 +555.55 555.55 5 55.55 555.55 555.55
277.75 2777.75 +2777.75 +2777.75 2777.75 15 277.75 2777.75 2777.75
611.05 6111.05 +6111.05 +6111.05 6111.05 35 611.05 6111.05 6111.05

```

\$BOM and \$EOM

Suprtool has two new subfunctions called \$BOM and \$EOM which are now available in the \$stddate and \$month functions. The \$bom returns the beginning of the month date, "01" and the \$eom, returns the end of the month for the given date.

For Example, the \$stddate function will return the converted curdate, with either the beginning of the month for the day or the end of the month for the given curdate.

```

in file131
item curdate,date,ddmmyyyy
ext curdate
ext curbom=$stddate(curdate,$bom)
ext cureom=$stddate(curdate,$eom)

```

You can also use \$stddate to convert a date that is already in ccymmd format to the beginning of the month or the end of the month respectively.

You can also use \$bom and \$eom as part of the \$month function even if the date is not in ccymmd format.

```

Item curdate,date,ddmmyyyy
ext curdate
ext curbom=$month($stddate(curdate),+6,$bom)
ext cureom=$month($stddate(curdate),+6,$eom)

```

String Function Checks

Some of the String functions in Suprtool did not have proper length checks in place to insure that an overflow wouldn't occur for specific functions. The length for most string fields to be checked is 4095 bytes. Suprtool will now throw a more generalized error of:

```
Error: Field in function is greater than 4095 bytes
```

If a function is longer than the 4095, the field is potentially corrupted and/or truncated.

Sdlinux for Linux

SDLinux is a utility that will convert the Self-Describing information, such that Suprtool/Open can read the file. Specifically, it is meant to convert the .sd file from an HP-UX system to be read on Linux natively.

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.

Set Limits Override On | Off

Set Limits OverRide tells Suprtool for MPE to ignore any subsequent Set Limits TableSize command. This was added since the new MPE

table size is now 2 GB and works well without any limit. It was also added as Suprtool 4.4 and lower did not calculate the Set Limits Tablesize limit properly. So, if a user had Set Limits Tablesize 25, Suprtool would allow for a table greater than that old limit, while the new version of the table would stop with "Table Full" as prescribed by the command.

Overriding the TableSize, allows for customers to set in a global suprmgr file and not have to search thru code to remove the specific command.

Enhancements in Version 6.1

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.

\$Length

Suprtool has a new function called \$length which returns a double integer number that is the length of a given byte or numeric field. For byte type fields Suprtool starts at the right most byte position to look for any non-space character and calculates the length of the data. For numeric fields Suprtool will convert the number to display and starting on the left will search for any non-zero character to establish the length.

Below is an excerpt from the manual, that describes how it can be used:

\$LENGTH (Works on Byte-type and numeric fields)

Purpose of the \$length function is to return the length of the specified byte field as if the field has had the \$rtrim function applied or the position of the rightmost non-space character.

If Usage:

```
if $length(NAME) >= 15
```

Extract Usage:

```
Define LenField,1,4,double  
extract LenField=$length(byte-field)
```

\$ROBELLE variable

The \$ROBELLE variable on HP-UX and Open platforms has been expanded to 55 characters for the \$ROBELLE directory name when you are installing outside of /opt/robelle. You use the \$robelle variable to tell suprtool and all of the programs that come with suprtool where to find where it is installed.

Set EditSignNeutral

The \$edit function converts data from numeric to byte and applies an edit mask to format the data. Neutral data is not considered to be signed so if you have neutral data, a positive sign will not be applied to the edit mask. With Set EditSignNeutral turned on, Suprtool will consider the Neutral data to be positive and therefore print out the “+” or “DR”, depending on the edit-mask used.

```
Set EditSignNeutral On
```

Enhancements in Version 6.0

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
>xeg
>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 *
>xeq
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
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.

Bugs Fixed

Bugs Fixed In Suprtool 6.3.01

Sort Command. Suprtool would sometimes think it could sort a file in memory in very rare conditions. (Suprtool 6.3 Build 2)

Sort Command. Suprtool would not allocate enough space to sort certain files. (Suprtool 6.3 Build 5)

Sort Command. Suprtool would allocate too much space when trying to sort a file if the previous step still had an Oracle database open. (Suprtool 6.3.01 Build 2)

List Command. Suprtool did not correctly convert double logicals over 2 billion, when converting to ascii in the list command. (Suprtool 6.3 Build 2)

Output,ascii. Suprtool did not correctly convert double integers if the value had decimal places and the value was less than the number of decimal places. (Suprtool 6.3 Build 14)

Output,ascii. Suprtool would incorrectly pad an uneven output record size on the first record with a NUL. (Suprtool 6.3 Build 14)

Input Command. Input files with Bigendian data, converts the data about three times quicker. (Suprtool 6.3 Build 5)

If Command. The if command, tracks and converts BigEndian data on input files when ffisbe is on more effieciently and in the proper cases. (Suprtool 6.3 Build 5)

Chain Command. A chain command with multiple values inline with byte or logical fields would fail. (Suprtool 6.3 Build 5)

If \$Lookup. An if \$lookup would fail on certain data fields if the fields were Big Endian. (Suprtool 6.3 Build 5)