Suprtool 4.9.03 for HP-UX:

Change Notice

by Robelle Solutions Technology Inc.

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

Phone: 604.501.2001
Fax: 604.501.2003

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

Contents

Introducing Suprtool Version 4.9.03

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 4.9.03

- Suprtool for HP-UX would not set the return code variable $?, if it failed with some syntax errors.
- Suprtool had some debug messages were left over in the Table module.
- Suprtool would have a stack overflow when the input source was an Oracle table, the if command was used and no records were selected.
- Suprcall for AMXW would not build a tempfile even if the Build command was specified in the Suprtool task.
- Suprcall would not always print the stdlist if the print-state was set to ER.
- Suprcall now returns an error status if Suprtool run has an error, in both the AMXW version and the regular HP-UX version.

Highlights in Suprtool 4.9.02

- A stack overflow error would occur when sorting with two sort keys and the input source was an Oracle Table.

- Suprtool would output 100 blank records when reading an empty Eloquence dataset with set fastread on.
- Suprtool2 fixed to work better with certain cobol compilers and compiler options.
- Command lines that were not terminated with a null are now handled correctly in Suprtool2.
- Print state and execution of print of stdout now works properly in all cases in Suprtool2.
- Tmpnam call would occasionally abort on Itanium due to ineffective space allocation in Suprtool2.

Highlights in Suprtool 4.9.01

- The Total command now supports the $file and $list options.
- Output=input now writes out the sort information if the file involved is self-describing.

Highlights in Suprtool 4.9

- The $CLEAN function in Suprtool and Clean command in STExport can now remove a character that needs to be cleaned but it does not need to replace with a space, but rather shifts the text to the left and adds a space to the end.
- Suprlink's Join command would not build the output file large enough if the input file flimit was significantly smaller than the Link file.
- Suprtool 4.8.10 would abort on a duplicate none key operation on a file greater than 5,000 bytes (approx)
- Suprtool would incorrectly retrieve Oracle Null dates when with a real Date value, instead of Null values in certain versions of Oracle.
- Suprtool would not be able to extract the last field of a table when the extracted record size was an uneven number of bytes and the set Ifouroutput option was being used with Oracle tables.
- Suprtool would return Illegal constant error when Set Varsub was on and the $Counter function was being used.
- Suprtool would return Bad Address or stack overflow on some sort operations.
- The Table command would previously not release memory used for Tables in all cases.
- Output=input is now supported in Suprtool for HP-UX.
- The $number function would either add garbage onto the end of a conversion or two zeroes if the field did not have a decimal point.
- Suprtool has all new code to connect to Oracle databases which fixes a number of issues with connecting to recently released Oracle patches.

- Suprtool now dynamically loads all Oracle calls that it requires which also helps with several issues with connecting to recently release Oracle patches.
- The FILE option of the List command did not handle absolute file names properly.
- A number of changes have been made to Suprcall for HP-UX, commonly referred to as Suprtool2. Please see the manual for all the details.

Highlights in Suprtool 4.8.02

- The get command would not get end of file signal properly if the dataset was empty and Set FastRead was on.
- The information about the data loaded in a Table would be lost if the table data being referenced was the second held table and the previous task involved a chain command.
- The chain command would not return the correct record when used with Set FastRead On
- The information on the data in a Table would be lost if the table data being referenced was the second held table and the previous task involved a chain command.
- The Clean command in Suprtool would incorrectly upshift lower case alpha characters.
- The $edit function will now work when nested within other string functions
- The List command on HPUX now has the RECLEN parameter.
- Suprtool now has the command Set Oracle ZeroNull On or Off which optionally turns null values into zeros.
- Suprtool for HP-UX now can read Eloquence datasets up to five times faster than the default DBGET serial reads.
- Eloquence routines are now dynamically loaded, this allows new features and bug fixes to be available as soon as the new version of Eloquence is loaded.
- The $subtotal command has been re-written to use less resources and some bugs have been fixed.
- Suprtool now has a $findclean function to identify records with specific characters in it
- Dbedit for HP-UX now works on HP Eloquence databases.
- The List command now has new options for listing to a file on HP-UX.
- The $subtotal function would not work if the previous task used the Dup None Keys feature.
- The $total function would appear to total data incorrectly when sorting on the field that was being totalled.

- Variables that resolved to all spaces for the entire command line would not work.
- Suprtool now allows up to 255 $split functions per task.
- Suprtool now has a $edit function for formatting data.
- Suprtool's Open command can now connect to a remote Oracle database.
- Numrecs 100% would come up with the wrong output file size when reading very large files.
- The new $split function would put random characters at the point where the split would occur in some cases.
- The new $split function would incorrectly report an error in a second task with multiple $split operations.
- The new $number function did not handle numbers that consisted of only a decimal place followed by any number of zeroes and a number, as in .01 thru .09.
- Suprlink would abort if the Join file was empty.
- Suprlink would hold the Join file open after the task was completed.
- Suprtool would total incorrectly when using a $subtotal function.
- The form command would display the percentage full as a very large strange number if the capacity and the entries were both zero for Eloquence databases.
- Update ciupdate would not be effective for Eloquence databases in the case where Ciupdate was allowed.
- We have worked around an Oracle patch issue that stopped Suprtool from being able to connect to databases with the Open command.

Known Problems

Occasionally and Add to an Oracle Table aborts, typically after multiple large adds. Currently the work around is to just start a fresh copy of Suprtool for the particular Add task that the abort occurs on.

If you have any questions or concerns or feedback on this or any other issue, please feel free to e-mail me at: neil@robelle.com

Compatibility

Suprtool/UX 4.9 is now compatible with,all versions of HP-UX 11.2x, as well as Oracle version 8 and higher.

Suprtool/UX version 4.8.02 and lower is compatible with HP-UX 9.0 and higher and Oracle 7.

The fix to the $number bug, creates an incompatibility if you used a work around by dividing by 100 and truncating. We therefore added set numbug on, to retain the behaviour that has the bug, you will need to turn on set numbug on. Therefore it is the default behaviour that is incompatible. See >set numbug on for details.

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 1-800-453-8970.

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 PDF format or HTMLHelp format (.CHM) and even 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 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 1.800.453.8970. 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 1.800 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/

Enhancements in Version 4.9.03

Introduction

Every year we provide Suprtool users with new features. The following section describes the new enhancements to Suprtool.

Total $File $List

The Total command has been enhanced in version 4.9.01 to allow the $file and $list options. This feature which has been on MPE for some time, is now available in Suprtool for HP-UX. The same syntax in valid, however, the default size for the file created needs to be opened with 48 bytes wide and with linefeeds:

```
  >in file1sd
  >tot int-field
  >total $file file82a
  >xeq
  >in file82a,rec 48,lf
```

You can also send the totals to a list file for simple reports:

```
  >in file1sd
  >tot int-field
  >list standard file file82a
  >total $file $list
  >xeq
```

Oracle Integers

Suprtool by default maps certain numeric fields into packed-decimal data types when they have more than one decimal place:

Precision	Decimal Places	Suprtool Data-Type
None	Any	8-byte IEEE
1-4	Zero	2-byte Integer
5-9	Zero	4-byte Integer
1-9	Non-zero	Packed-decimal
10-27	Any	Packed-decimal
28-38	Any	8-byte IEEE

The setting:

```
Set Oracle Integer on
```

changes the Suprtool format from packed-decimal to Integer based on the size of the Number:

Precision	Decimal Places	Suprtool Data-Type
1-4	Any	2-byte Integer
5-9	Any	4-byte Integer
10-27	Any	8-byte Integer

$Clean Function and Clean Command

The $Clean function in Suprtool and Clean command in STExport now have the ability to replace a character to be cleaned with null or nothing. If you set the cleanchar as being:

```
   >Set Cleanchar “<null>”
```

Suprtool will remove the character specified in the clean command and effectively shift the text to the left and blank out the portion at the end.

Oracle Dynamic Load

We have further revised the Oracle Dynamic Load to do the following:

1) Attempt to load libclntsh.sl using the dynamic path feature. See man dlopen for details.

2) Manually load $ORACLE_HOME/lib32/libclntsh.sl

Suprtool now dynamically loads all of the Oracle routines that it needs at startup. This is only done if you the Oracle interface is enabled. This insures that Suprtool is using calls only for your version of Oracle and provides a more stable environment.

If Suprtool fails to load the Oracle library, you will see the message:

```
  Warning: Your oracle library could not be loaded.
```

This means that any reference to Oracle calls will not function. If you do not have the Oracle option enabled in Suprtool you will not see the errors. It also may mean you need to add the path for the Oracle libraries either to your LD_LIBRARY_PATH or SHLIB_PATH variables. Suprtool will search for the library first in LD_LIBRARY_PATH and then SHLIB_PATH.

Eloquence Dynamic Load

Previously, Suprtool would attempt to load the Eloquence libraries and report an error message if the libraries were not loaded. This is no longer the default behaviour. Suprtool will try to load the libraries function, however Suprtool will not report an error by default. To check if your libraries were loaded you can run Suprtool with the –lw option:

```
 ./suprtool -lw
```

Oracle Connections

Due to recent patches and changes in Oracle Security, older versions of Suprtool could not connect to some Oracle databases. We have therefore re-written all of the code to connect to databases to use more up to date OCI functions.

By default, Suprtool will now use this new method of connecting. This version also improves the parsing of the open command, allowing for 30 characters each for the username, password and dbname. This also allows for connection to databases on other servers.

```
   >Open Oracle username/password@dbname
```

Set Oracle OpenFix On. Set Oracle OpenFix forces Suprtool to use the olog call, when connecting to Oracle databases. This option should no longer be necessary, now that the default is to use the new connection method.

Set Oracle OpenOld On Set Oracle OpenOld forces Suprtool to use the orlon call, when connecting to Oracle databases. This should only be used if connection to Oracle 7 databases.

$Number Function

The $number function had a bug whereby it would add on two zeroes and or bad data if the input number did not have a decimal point. We have fixed the bug so that the number function no longer adds the two digits on the end in error.

However, some users worked around this issue by doing the following:

```
 >extract target = $truncate($number(conv-field) / 100)
```

Since some users, used this work around, the fix to the $number function will then return incorrect results. Therefore, we added the set numbug command to have Suprtool revert from the correct behaviour to continue to have the bug.

By default, Suprtool will just convert the number and not add on the data at the end, however, if you have used the work around then you can add the command:

```
  >set numbug on
```

to the script directly or globally in your suprmgr file.

Table Command

The Table command is now capable of reading files greater than 4Gb.

Output=input

Suprtool now supports the Output=input command, whereby Suprtool sorts a file onto itself. [4.9]

Suprtool now writes out the sort information when using output=input and the file is self-describing. [4.9.01]

Enhancements in Version 4.8.02

Introduction

Every year we provide Suprtool users with new features. The following section describes the new enhancements to Suprtool since Suprtool 4.7.

Set Oracle ZeroNull

Suprtool for HP-UX has a new option to turn on changing null fields to zeroes. Set Oracle ZeroNull On will change any fields that are "null" (in the SQL sense of the word), will become zeroes for the appropriate number type.

In order to turn this feature on for all accesses you can put the command:

```
   >Set Oracle ZeroNull On
```

into the file /opt/robelle/suprmgr.

Previous to version 4.6 in Suprtool used to return nulls, but Suprtool 4.6 and Suprtool 4.7 would return zeroes. We have decided to make this optional and make the default to return Nulls, due to problems when using the to_char function in the select statement.

Reclen on List command

Suprtool for HP-UX can output data from the List command to a discfile. Although the concept of record size is not the same on HP-UX as it is on MPE it is still important in some areas within Suprtool.

In this case the RECLEN parm merely tells the List command where to fold the lines. The Reclen parm can be a value from 56 to 256 and defaults to 80 bytes.

```
   >List FILE myreport RECLEN 132
```

Set FastRead On

Suprtool for HP-UX by default calls dbget to do serial reads, now with Suprtool you can utilize faster reads with the Set FastRead On command. This command invokes more efficient large reads. Testing has shown that the CPU time can be improved by anywhere from two to five times and Wall time has improved anywhere from two to six times faster. In order to turn this feature on for all accesses you can put the command:

```
  >Set FastRead On
```

into the file /opt/robelle/suprmgr. This means that Suprtool will use the faster reads for all runs of Suprtool.

Eloquence Dynamic Loading

Suprtool for HP-UX now attempts to dynamically load the Eloquence routines. Suprtool requires two Eloquence libraries, namely: libimage3k.sl and libeqdb.sl. What Suprtool now does is look for these libraries in two separate ways.

First Suprtool looks for libeqdb.sl and libimage3k.sl in any of the directories named in the 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:

```
# add SHLIB_PATH for Eloquence library search
    export SHLIB_PATH=/opt/eloquence6/lib/pa11_32
```

The line proceeding the export command is a comment line and does not need to be in the file, but is just a reference to indicate what it is used for. If you do not have the SHLIB_PATH variable set to a value where libeqdb.sl and libimage3k.sl can be found, Suprtool will then try to load libimage3k.sl in the directory /opt/eloquence6/lib/pa11_32 and libeqdb.sl from the same directory. If the libraries still fail to be loaded then Suprtool will print two warnings, however, it will still continue to function, just any of the Eloquence features will fail when called:

```
SUPRTOOL/UX/Copyright Robelle Solutions Technology Inc. 1981-2004.
(Version 4.9 Internal)  MON, MAR 08, 2004,  7:34 AM  Type H for help.
Warning: Could not load Eloquence image library
Warning: Could not load Eloquence scan library.
```

$FindClean Function

We recently added the $Clean function to primarily clean "bad" characters in text fields. This has been extremely popular enhancement but many wanted to do investigative work and try to figure out what records had these bad characters, to hopefully find out where the "bad" data was coming from. For this reason we have created the $FindClean function. $FindClean will return true if it finds a character defined using the Clean command.

```
>in cleansd
>clean "^9","^10"
>if $findclean(myfield)
>list
>xeq
```

The above task will list the record if the field nonprint has a Tab (Decimal 9) or a Line Feed (Decimal 10) anywhere in the field. You can Find and clean the "bad" characters from a field at the same time:

```
>in cleansd
>clean "^9","^10"
>if $findclean(myfield)
>extract myfield=$clean(myfield)
>list
>xeq
```

Dbedit Module

The Dbedit Module has been added to Suprtool for HP-UX. It works with Eloquence databases and with Oracle databases with the IMAXSoft version of Suprtool.

List Command

The List command has been enhanced with a new File keyword that allows output to be directed to a file. The List command also has a new option to Append to an existing file. The File option takes the next parameter as being the filename:

```
 >in test/file1sd
 >list stan file myslist
>xeq
```

If the file myslist exists it will be over-written, unless you specify the Append option. If you specify the append option the new report will be added to the file.

So if you want to incorporate multiple reports you just need to do the following:

```
>in test/file1sd
>list stan file myslist
>xeq
>in test/file2sd
>list stan file myslist append
>xeq
```

$Split Function

Suprtool now allows up to 255 $split functions per task. The previous limit was 16, and the limit has been changed to assist in reading data from "PRN" files.

$Edit Function

Suprtool can format fields using edit-mask features similar to edit-mask features of Cobol. Suprtool employs two distinct types of edit-masks: one for byte type fields and the other for numeric fields.

The type of mask utilized depends on the source type of the field. If the source field is numeric, then the numeric edit-mask logic is applied, if the source field is byte type, then the byte edit-mask logic and characters apply.

The target field must always be a byte type field.

Placeholders and Format Characters

An edit-mask consists of "placeholder" characters, such as "9" for a numeric column, and "format" characters, such as "." for the decimal place. Sometimes an edit-mask character acts as both a placeholder and a format character, such as the "$" in floating dollar signs.

Byte-Type Formatting

For Byte type fields there are two placeholder characters. These are:

X ~ place the data in the matching column for the X in the edit-mask

Z ~ place the data in the matching column unless the data is a zero; if the data is a zero, then replace with a space

The format characters are as follows:

B (space) / (slash) , (comma) . (period) + (plus) - (minus) * (asterisk)

and a Space. Please note that you can denote a space using two methods, either by putting a "B" in the mask or a space itself. For example, suppose you have data that is in ccyymmdd format in an X8 field. Here is how you would use a "xxxx/xx/xx" mask to format the data:

```
>in mydate
>form
   File: MYDATE.TEST.NEIL (SD Version B.00.00)
      Entry:             Offset
         A       X8      1  <CCYYMMDD>
   Limit: 10000  EOF: 2  Entry Length: 8
>def formatdate,1,10
>ext formatdate=$edit(a,"xxxx/xx/xx")
>list
>xeq
>IN MYDATE.NEIL.GREEN (0) >;OUT $NULL (0)
FORMATDATE   = 2003/09/24

>IN MYDATE.NEIL.GREEN (1) >;OUT $NULL (1)
FORMATDATE   = 2003/09/24
```

As you see in the example above, the placeholder character is the "x" and the "/" is the format character. You insert a space either by specifying a "B" or by putting an actual Space character in the edit-mask. An example of inserting a space might be the formatting of Canadian postal codes (e.g., V3R 7K1):

```
>in postal
>form
   File: POSTAL.NEIL.GREEN
      Entry:                     Offset
         POSTAL-CODE          X6      1
   Limit: 10000  EOF: 2  Entry Length: 6
>def post1,1,7,byte
>def post2,1,7,byte
>ext post1=$edit(postal-code,"xxx xxx")
>ext post2=$edit(postal-code,"xxxbxxx")
>list
>xeq

>IN POSTAL.NEIL.GREEN (0) >OUT $NULL (0)
POST1     = L2H 1L2     POST2     = L2H 1L2

>IN POSTAL.NEIL.GREEN (1) >OUT $NULL (1)
POST1     = L2H 1L2     POST2     = L2H 1L2
```

Z-placeholder for byte-fields

The Z-placeholder character works differently for byte-fields than for numeric fields. For byte type fields, if the Z placeholder and the corresponding data is "0", then the zero is suppressed, regardless of the position. This is primarily for suppression of zeroes in byte type date fields:

```
ext a=$edit(date-field,"xxxx/zx/zx")
```

The above edit mask would then edit a byte type date of 20031005, to be:

```
2003/10/ 5
```

Overflow and limits

An edit mask is limited to 32 characters in total for both numeric and byte type fields. If data overflows the edit-mask, by default Suprtool will fill that field with asterisks. There is an option to have Suprtool stop when it encounters a formatting overflow:

```
 >set editstoperror on
```

will force Suprtool to stop if there is data left over after applying the edit-mask. With byte-type fields, leading spaces do not cause overflow. Therefore if your data consists of:

```
"    L2H1L2"
```

and your edit mask is:

```
"xxxBxxx"
```

It is not an overflow since there are only spaces to the left of the "L". If the data was:

```
"   JL2H1L2"
```

an overflow exception would occur.

Numeric field edit-masks

Our edit-masks for numeric fields are patterned after those in COBOL. We provide four placeholder characters, each with a slightly different effect:

"9" - insert a digit from 0 to 9 in this position

"$" - if you specify more than one dollar sign, you get a floating dollar sign. This means that there can be as many numeric positions as there are dollar signs, but if some positions are not needed because the value is small, the $ floats to the right next to the first digit and the preceding positions are blank.

"*" - if there are enough digits in the value, the * position is replaced by a numeric digit; if not, an asterisk is printed. Leading asterisks are often used for check writing, so that no one can insert a different value.

"z" - insert a numeric digit at this position; if the rest of the data to the left is a zero then a space will be placed at this position. For example:

```
>ext a=$edit(int-field,"$$,$$$.99-")
>ext b=$edit(int-field,"99,999.99-")
>ext c=$edit(int-field,"cr99999.99")
>ext d=$edit(int-field,"-$9999.99")
>ext e=$edit(int-field,"**,***.99+")
>ext f=$edit(int-field,"zz,zzz.99+")
>list
>xeq
>IN FILE1SD.NEIL.GREEN (0) >OUT $NULL (0)
A      =    $11.11-     B   = 00,011.11-
C      = CR00011.11     D   =  -$0011.11
E      = ****11.11-     F   =      11.11-

>IN FILE1SD.NEIL.GREEN (1) >OUT $NULL (1)
A      =    $22.22-     B   = 00,022.22-
C      = CR00022.22     D   =  -$0022.22
E      = ****22.22-     F   =      22.22-
```

Signs

As shown in the example above, there are also numerous format characters for numeric edits, including four ways to specify the sign. You can specify a sign, with +, -, or the typical accounting specification of "CR" and "DB". You will note in the example above that the "cr" in the mask was up-shifted to be "CR". This is because the entire mask is up-shifted as the mask is being parsed.

You can specify more than one sign in a numeric field edit, although Suprtool will give you a warning that having two sign edit-mask characters does not really make sense. Cobol gives a Questionable warning when compiling an edit-mask with two sign characters. Suprtool, will apply the sign in both places.

Keep in mind that most data has three states:

1) Postive
2) Negative
3) Neutral

Any neutral data will not display the sign. If you specify a "+" sign in the edit-mask and the data is negative, it will of course display a "-" sign.

Decimal Places

For numeric-type edits, Suprtool attempts to adjust the data according to the number of decimal places in the edit-mask, when compared to the number of decimal places defined in the field.

For example if the data field has one decimal place, and the edit mask has two decimal places, then the data is adjusted:

Data and Edit mask:

```
  102.3   ZZZZ.99
```

will result in the final data being:

```
  102.30
```

Similarly, if the data has three decimal places and the edit-mask only has two, then the data will be rounded appropriately with the same rules as outlined in the $number function.

You can specify more than one decimal place in an edit-mask. However, Suprtool will print a warning and it will utilize the right-most decimal place for data alignment. The decimal place character is defined by a set command:

```
   >set decimalsymbol "."
```

If you define another character as the decimal symbol, Suprtool will use that character as the point to align the decimals. If you define a decimal symbol that is not an allowed edit-mask character with Set Decimalsymbol, Suprtool will assume that the field has zero decimal places and adjust the data accordingly.

Currency and Dollar signs

Suprtool edit-masks support both fixed and floating dollar signs. Logic for floating dollar-signs will be invoked if more than two dollar signs are defined in the edit-mask.

A floating-dollar edit mask attempts to put the dollar sign at the left most position of the significant data. For example if you have the following data and edit mask:

```
0001234.54 $$$$$$.$$
```

the data would end up as:

```
   $1234.54
```

Suprtool will not however, put the dollar sign to the right of the decimal place. If you had the same edit mask and the data was, .09, the data would end up being formatted as:

```
       $.09
```

Similarily, the $edit function will attempt to place the dollar sign correctly in most cases. For example Suprtool will not format data in the form of:

```
   $,123.50
```

Suprtool, does attempt to fixup these cases and would format the data in the following manner:

```
    $123.50
```

Overflow and floating dollar

If the number of digits in the data is equal to the number of placeholder dollar signs, then the dollar sign is dropped and not added to the edited field.

```
12345.50 $$$$$.99
```

would result in:

```
12345.50
```

Set CurrencySymbol

If Set CurrencySymbol is not equal to "$", then after the formatting has been applied, whatever symbol(s) are defined within the set command, are used to replace the "$" symbol in the data. For example, if you have the Currency symbol set as "CDN".

```
  >set currencysymbol "CDN"
```

Suprtool will replace the "$" after the edit-mask has been applied with CDN, provided there is room to the left of the dollar-sign.It is recommended that if you are using multiple characters for the dollar symbol that you leave enough characters to the left of the symbol.

For example if the CurrencySymbol is defined as CDN, then you should leave two spaces to the left of a fixed dollar sign definition. If there is not enough room, to put in the currency symbol, then the dollar symbol is blank.

Overflow and limits

An edit mask is limited to 32 characters in total for both numeric and byte type fields. If data overflows the edit-mask, by default Suprtool will fill that field with asterisks. There is an option to have Suprtool stop when it encounters a formatting overflow:

```
  >set editstoperror on
```

will force Suprtool to stop if there is data left over to place when applying the edit-mask. With numeric-type fields, leading zeroes do not cause overflow.

Oracle Open

The Open command has been enhanced to allow connections to a remote Oracle database. In order to invoke this feature the username/password@machine, need to be specified on a single line.

```
  >open oracle suprtest/suprpass@remote
```

Bugs Fixed

Bugs Fixed In Suprtool 4.9.03

OS Communication. Suprtool for HP-UX would not set the return code variable $?, if it failed with some syntax errors.

Table Command. Suprtool had some debug messages were left over in the Table module.

Oracle Access. Suprtool would have a stack overflow when the input source was an Oracle table, the if command was used and no records were selected.

Suprtool2. Suprcall for AMXW would not build a tempfile even if the Build command was specified in the Suprtool task.

Suprtool2. Suprcall would not always print the stdlist if the print-state was set to ER.

Suprtool2. Suprcall now returns an error status if Suprtool run has an error, in both the AMXW version and the regular HP-UX version.

Bugs Fixed In Suprtool 4.9.02

Oracle Access. A stack overflow error would occur when sorting with two sort keys and the input source was an Oracle Table.

Set FastRead On. Suprtool would output 100 blank records when reading an empty Eloquence dataset with set fastread on.

Suprtool2. Command lines that were not terminated with a null are now handled correctly in Suprtool2.

Suprtool2. Print state and execution of print of stdout now works properly in all cases in Suprtool2.

Suprtool2. Tmpnam call would occasionally abort on Itanium due to ineffective space allocation in Suprtool2.

Bugs Fixed In Suprtool 4.9

Table Command. The Table command would not release memory space after the task was completed in all cases.

Table Command. The Table command would not release memory space after the task was completed in all cases.

Join Command. The Join command would fail with in certain cases

```
Call to ffileinfo-1
MPE File system unavailable
```

This only occurred in Suprlink 4.8.10.

Dup Command. Suprtool 4.8.10 would abort on a duplicate none key operation on a file greater than 5,000 bytes (approx)

Oracle Select. Suprtool would incorrectly retrieve Oracle Null dates when with a real date value, instead of Null values in certain versions of Oracle.

Set Ifourouput On. Suprtool would not be able to extract the last field of a table when the extracted record size was an uneven number of bytes and the set Ifouroutput option was being used with Oracle tables.

$Counter function. Suprtool would return Illegal constant error when Set Varsub was on.

Sort Operation. Suprtool would return Bad Address or stack overflow on some sort operations.

$Number Function. The $number function would either add garbage onto the end of a conversion or two zeroes if the field did not have a decimal point.

List Command. The FILE option of the List command did not handle absolute file names properly. (HP-UX, specific)

Table Command The Table command now can read files greater than 4 Gbs. (HP-UX specific, MPE version already does this)

Bugs Fixed In Suprtool 4.8.01

Chain Command. The chain command would not return the correct record when used with Set FastRead On.

Bugs Fixed In Suprtool 4.8

$Edit Function. The $edit function now returns a proper result when nested within another string function such as $ltrim.

Clean Command. The clean command was improperly upshifting lower case alpha characters.

Bugs Fixed In Suprtool 4.7.12

SubTotal Function. The $subtotal function would fail if the size of the Output buffer was larger than the size of the Input buffer.

Bugs Fixed In Suprtool 4.7.11

Variable Substitution. Suprtool would report the error:

```
  Error:  >KEY has 2-4 parms: pos,len[,type][,DESC].
```

when resolving a variable that resolved to a blank line. This is now fixed in Suprtool 4.7.11.

Total Function. The $total function would appear to accumulate incorrectly when sorting in the same task.

SubTotal Function. The $subtotal function would not work if run in the same copy of Suprtool if the previous task used the Duplicate command.

Bugs Fixed In Suprtool 4.7.10

Incorrect Flimit. Suprtool would incorrectly calculate the flimit on an output file when using Numrecs 100%, and if the input file was very large.

$Split generating random characters. The new $split function would put random characters at the point where the split would occur in some cases.

$Split reported bogus error on repeated task. The new $split function would incorrectly report an error in a second task with multiple $split operations.

$Number decimal only numbers incorrect. The new $number function did not handle numbers that consisted of only a decimal place followed by any number of zeroes and a number, as in .01 thru .09.

Empty Join File caused abort. Suprlink would abort if the Join file was empty.

Join File held open. Suprlink would hold the Output file open after the task was completed.

$Subtotal incorrect. Suprtool would give incorrect numbers for a $subtotal function in certain cases.