Suprtool 4.9.02 for HP e3000:

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 4.9.02

Introduction

Suprtool provides fast access to your data on the HP e3000. With Suprtool, you can perform many necessary DP functions easily, with just a few simple commands. At Robelle we are constantly working on your enhancement requests so that we can include them every year when we release an updated version of Suprtool.

Highlights in Suprtool 4.9.02

- Suprtool would incorrectly give an error with the Set Buffer command with some values.

Highlights in Suprtool 4.9.01

- Suprtool would leave a KSAM/iX file open if the task had an if command.
- The $CLEAN function in Suprtool and Clean command in STExport can now remove a character that needs to be cleaned and it does not 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 not select the correct records for Ksam64 and KSAM/XL files when using high speed access.
- 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.

Highlights in Suprtool 4.8.02

- 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 Clean command in Suprtool would incorrectly upshift lower case alpha characters.
- The $edit function will now work when nested within other string
- The $subtotal command has been re-written to use less resources and to fix a bug in the recognition of a key value change.
- Suprtool now has a $findclean function to identify records with specific characters in it.
- 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.
- Numrecs 100% would come up with the wrong output file size when reading very large files.
- Suprtool now has a $edit function for formatting data.
- 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.
- Suprtool would write corrupt data at the end of each block if the record size of the input file and output file was over 10,000 bytes.

Known Problems

There are no known problems at this time.

Compatibility

Suprtool/iX is compatible with MPE V, MPE XL, and MPE/iX, including MPE/iX 6.5, 7.0 and MPE/iX 7.5.

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 details further on in this change notice.

CPU Serial Number and CPU Name (HPSUSAN & HPCPUNAME)

Suprtool and its associated products run only on CPUs whose serial numbers have been encoded ("showvar hpsusan" on MPE/iX). They also may be encoded with your HPCPUNAME values. If it fails to run and you get an "invalid HPCPUNAME" or "invalid HPSUSAN" error message, contact Robelle for assistance.

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, or see the change notice.

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

We have two types of Installations, Tape and Download. We provide links to the Installation instructions on our web site for each type of Install.

Ecometry Users

Users of the Ecometry application need to use the installation instructions supplied by Ecometry Corporation. Those instructions have extra steps for updating the Ecometry code accounts.

You will find the Ecometry version of these instructions at http://www.robelle.com/ecometry/

Types of installations, tape, download combined qedit and suprtool tape.

Instructions

If you received a production release tape from Robelle, then you can install using the instructions on this web page:

http://www.robelle.com/support/install/tape/stprod.html

If you downloaded the Suprtool production release from our web site then the instructions can be found here for the ftp download method:

http://www.robelle.com/downloads/install-stprod-ftp.html

and here for the WRQ download method:

http://www.robelle.com/downloads/install-stprod-wrq.html

Installation Assistance

If you have any questions about the upgrade process 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 800.453.8970 or you can e-mail your question to support@robelle.com.

Enhancements in Version 4.9.02

Introduction

Every year we provide Suprtool users with new features. The following section describes the new enhancements to Suprtool since Suprtool 4.8.10.

$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.

$Number Function

The $number 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.

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 4.7.02.

$Subtotal Function

The $subtotal function now uses less resources than in previous versions. There was also a bug where the $subtotal function would not correctly determine if a field entered was a sort key or not.

$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(nonprint)
  >list
```

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 $findclean(nonprint)
  >extract nonprint=$clean(nonprint)
  >list
```

$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) &gt;OUT $NULL (0)
FORMATDATE   = 2003/09/24

>IN MYDATE.NEIL.GREEN (1) &gt;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.

Bugs Fixed

Bugs Fixed In Suprtool 4.9.02

Set Buffer. Suprtool would incorrectly give an error with the Set Buffer command with some values.

Bugs Fixed In Suprtool 4.9.01

Join Command. When the Join command was used in Suprlink and the Input file has a considerably smaller flimit than the Join file, the Output file would not be built large enough.

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

Chain Command. Suprtool's Chain command with a key value from a Table would not find some entries in Master Datasets for an Omnidex key in Omnidex 3.07.55.

KSAM files. Suprtool and FCOPY for that matter, would skip some records when not accessing records with a ;nocopy file equation and using an if command, or in the case of FCOPY, using the ;exclude option. Suprtool now automatically detects when using an if command and re-opens with ;nocopy access. Suprtool 4.9 would leave KSAM/iX files open when the task had an if command in it, this is now fixed in Suprtool 4.9.01.

$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.

Bugs Fixed In Suprtool 4.8.02

Get Command. The get command would not get end of file signal properly if the dataset was empty and Set FastRead was on.

Table Command. 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.

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.

Data Corruption. Suprtool write incorrect data at the end of each block if the record size was greater than 10,000 bytes.