
Suprtool 4.7 for HP-UX:

Change Notice

by Robelle Solutions Technology Inc.



Program and manual copyright © 1981-2003 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, 13711 – 72nd Avenue,
Surrey, B.C. Canada V3W 2P2

Phone: 604.582.1700
Fax: 604.582.1799

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

Contents

Introducing Suprtool Version 4.7	2
Overview	2
Highlights in Suprtool 4.7.....	2
Highlights in Version 4.6.01.....	3
Highlights in Version 4.6.....	3
Known Problems	3
Compatibility	4
CPU Serial Number (uname).....	4
Documentation.....	4
User Manuals.....	4
Change Notice	4
Printed Documentation.....	4
CD or Web Download.....	4
Installation	5
Overview	5
Who Should Use These Instructions?.....	5
Summary of Installation Steps.....	5
Installation Assistance	5
Step 1: Log In as Root	6
Step 2: Create Robelle Directory.....	6
Step 3: Restore Files	6
Step 4: Set the ROBELLE Variable.....	6
Bourne and Korn Shells.....	6
C Shell	6
Enhancements in Version 4.7	7
Introduction	7
Eloquence 7	7
Base Command.....	7
Put Command.....	7
Expanded Database Limits	8
Variable Substitution	8
Tables on HP-UX Improvements.....	8
Table Sizes	8
Extract from a Table.....	9
\$Number Function	10
Suprlink Join Command	11
\$\$split Function.....	12
\$SubTotal Function	12
Cleaning your Data	13
Suprtool	13
Defining a Clean Character	13

Setting the Clean Character	14
Cleaning a Field.....	14
Cleaning your data.....	14
STExport	14
Outcount and Fullcount	15
STExport Escape Command.....	15
Table Filename	16
\$Total Function.....	16
\$Counter Function	16

Enhancements in Version 4.6 **17**

Overview	17
Support for Eloquence	17

Bugs Fixed **19**

Bugs Fixed In Suprtool 4.7	19
Bugs Fixed in Version 4.6.01	19
Bugs Fixed in Version 4.6	20

Introducing Suprtool Version 4.7

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 4.7

- Suprtool now has a \$Number function which will allow Suprtool to read a freeform ascii number with signs, decimal places and currency symbol as a display field.
- Suprtool now has a \$SubTotal function.
- Suprtool can now split byte strings into multiple fields via the \$split function.
- Suprlink can now do many-to-many links via the Join command.
- Suprtool and STExport now support features to Clean your data.
- STExport now supports an Escape command, which will escape out certain characters when used with certain import programs.
- Suprtool now supports a \$counter function which will increment a field by one for every record selected.
- Suprtool now supports a \$total function which will keep a running total for a specified field.
- Suprtool, STExport and Suprlink now support environment variables.
- Suprtool has been enhanced to support Eloquence 7.0 features.
- The Base and Put commands have been changed to support new features in Eloquence 7.0.
- Suprtool now supports the expanded database limits in Eloquence 7.0.

- Suprlink and STExport now report the number of output records in a manner similar to Suprtool.
- The Table command now supports filenames up to 80 characters.
- The Table command truncated filenames at the limit of 36 characters.
- The number of defines allowed in Suprtool has been increased to 768.
- Suprtool would report an incorrect record number when encountering an Illegal ascii digit, if the Suprtool task involved the Duplicate command.
- Suprtool would incorrectly report an error on a second set tablesize command for a given task.
- STExport would in some cases attempt to format data in XML and HTML format at the same time.
- Extract from a table would incorrectly report an error in some cases.

Highlights in Version 4.6.01

- Suprtool would incorrectly coerce large negative numbers from a Packed field to another Packed or Display field.
- Suprtool would fail with prefetch point failure if the file had an uneven record size and an uneven blocking factor.
- STExport now properly converts the Roman 8 universal monetary symbol to the Euro-Symbol in the HTML and XML commands.
- The Item command would fail if a definition or Item command was already issued for a field, if the input source was a self-describing file and an extract range was used or all items from a dataset were extracted.
- The XML command would fail to convert ">" to ">" and "<" to "<".
- The XML and HTML commands failed to convert "&" to "&".

Highlights in Version 4.6

- Support for "well-formed" XML in STExport.
- Suprtool, STExport and Suprlink can now have warnings turned off when run from batch with the set warnings off command.
- Some division operations would not work properly if the decimal portion was large and the target type was packed or zoned.

Known Problems

There are no known problems at this time.

Compatibility

Suprtool/UX is compatible with HP-UX 9.0, all versions of HP-UX 10.x,11.x as well as Oracle version 7.1.3.2.0.

On HP-UX 10.x, Suprtool/UX creates all of its temporary and scratch files in /var/tmp, unless you have overridden the temporary directory with the `TMPDIR` environment variable.

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

Suprtool comes with User Manuals and a Change Notice. You may have received printed copies of these.

They are also available as PDF or HTML Help files. You may have received a documentation CD with these files, or you can download the files from the Robelle web site.

User Manuals

The user manuals contain the full description of all the Suprtool commands, as well as usage tips. The manuals are up-to-date with all the latest changes incorporated in Suprtool. To see only the changes in the latest version, see the "What's New" section of the manual, or see the change notice.

Change Notice

For a complete description of the latest changes made to Suprtool, the installation instructions, and any compatibility issues, see the change notice that was included with the release.

Printed Documentation

The latest user manuals and change notice are available in Adobe PDF format. If you do not already have the Adobe Acrobat Reader, you can get a copy from <http://www.adobe.com/prodindex/acrobat/readstep.html>.

CD or Web Download

If you received a documentation CD with this version of Suprtool, you will find the documentation files on the CD. The installation program will start automatically when you insert the CD into the drive. If it does not start automatically, run `D:\setup.exe`, where *D* is the letter assigned to your CD-ROM drive.

If you do not have a documentation CD, you can download the files from the Robelle 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.

Who Should Use These Instructions?

The system manager should use the following instructions to install Suprtool/UX. During the installation, which should take only a few minutes, no one can use Suprtool/UX.

Summary of Installation Steps

To install Suprtool, follow these steps:

1. Log in as root.
2. Create the correct directory structure.
3. Restore Suprtool/UX and its associated files from the distribution tape.
4. Set up variables to let Suprtool know where its supporting files are located.

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/>

Step 1: Log In as Root

There are two ways you can log in as root:

1. Exit from HP-UX and log in with root as the user name.
2. If you are already logged in, you can execute this command:

```
su -
```

In either case, you have to know and supply the user password for root.

Step 2: Create Robelle Directory

Before restoring files, you must first create the directory where Suprtool/UX will reside:

```
mkdir /opt/robelle
```

Step 3: Restore Files

Use the following command to restore the Suprtool/UX files from the distribution tape:

```
tar xv /opt/robelle
```

This command assumes your tape device is /dev/rmt/0m. If it is not, you need to specify your tape device using the "f" option in the tar command. For example, if your tape device is /dev/rmt/1m, you need to use the following command to restore the files:

```
tar xvf /dev/rmt/1m /opt/robelle
```

Step 4: Set the ROBELLE Variable

You must set the ROBELLE environment variable so that Suprtool can find its Help and Suprmgr files.

Bourne and Korn Shells

```
export ROBELLE=/opt/robelle
```

C Shell

```
setenv ROBELLE /opt/robelle
```

Enhancements in Version 4.7

Introduction

Every year we provide Suprtool users with new features. The following section describes the new enhancements to Suprtool since Suprtool 4.6.01.

Eloquence 7

Suprtool has been enhanced to work with Eloquence 7. It takes advantage of many new features and supports the new limits and data types.

Base Command

Suprtool's Base command has been enhanced to allow the new syntax supported in Eloquence 7. Eloquence now allows the server name and service to be specified in dbopen. To support this new syntax Suprtool's Base command has been changed to allow the server name, service and database name be specified. Suprtool uses the same syntax as Eloquence whereby the database name consists of the following elements:

```
[[host][:service]/]database
```

Examples of using this syntax within Suprtool using the sample database that Eloquence provides.:

```
base sample,5
base :eloqdb/sample,5
base hostname.robelle.com:eloqdb/sample,5
```

Put Command

The Put command has always supported for a database name to be specified. Since the Base command allows the new syntax for Eloquence version 7 support, the Put command also allows this new syntax.

```
put dataset,sample
put dataset,:eloqdb/sample
put dataset,hostname.robelle.com:eloqdb/sample
```

Expanded Database Limits

Eloquence version 7 now has new database limits that follow or extend the current Turbo IMAGE limits.

Data Items per database	2048
Data Sets per database	500
Paths per dataset detail	16
Paths per dataset master	64

Variable Substitution

Suprtool now supports environment variable substitution. To use this enhancement you must do a:

```
>set varsub on
```

Due to how HP-UX processes work with environment variables any variables must be exported prior to running Suprtool, STExport or Suprlink. All of these programs support HP-UX variable Substitution.

```
export tablefile='abcdefghijklmnopqrstuvwxyabcd'
export infile='file1sd'
./suprtool -oc << \!EOD
set varsub on
in $infile
table mytable,char-field,file, &
/users/robdev/suprtool/test/$tablefile
if $lookup(mytable,char-field)
out file05,link
exit
!EOD
```

Suprtool examines a command line and looks for variables denoted by the "\$" sign. Since Suprtool has some functions that begin with a \$-sign, these will take precedence regardless of the value set in the variable of the same name.

Tables on HP-UX Improvements

We have re-written the underlying mechanisms for the Table and \$Lookup functions in Suprtool for HP-UX, to allow more data in the table and have similar functionality to the Table command on MPE. This includes Larger tables and Extract from a table functionality described below.

Table Sizes

On HP-UX you can control the size of a table with the Set Limits TableSize command. By default an individual Table will be 50 Megabytes in size and you can have up to 10 tables. The Global limit for all tables is up to 500 Megabytes. You can control the size of a given table with the command:

```
>Set Limits TableSize n
```

If you enter the command `Set Limits TableSize 100` and the next table command that you build will have a limit of 100 Megabytes. Previously the Limits on Tables were 15Mb in total and defaulted to 1Mb in size.

Extract from a Table

Suprtool now has the ability to load data into a table via the `Table` command, and extract that data out of the table using the `Extract` command. The `Table` command now allows for data to be loaded along with matching key values.

```
>table table-name, key-field, file, filename, &
    data(field1, field2, ...)
```

An example of loading two data fields called `cost` and `desc` along with the key field of `part` into a table would be:

```
>table partab, part, file, partin, data(cost, desc)
```

You can specify up to 20 data fields as long as the total size of the key fields and data does not exceed 256 bytes. The `Table` file must be Self-Describing (Link) in order to use the `data` option. When loading data into a table, Suprtool will eliminate the duplicate entries based on the key value, so the associated data values may not be loaded into the table.

The `Extract` command can utilize the `$lookup` function to return data. The syntax for the `$lookup` function would look as follows:

```
>extract target = $lookup(table-name, key-field, data-field)
```

The `Table` name, `key-field` and `data-field` are all defined by the `Table` command, which must be input before the `Extract` command.

A classic example: your boss comes to you with a list of new prices and descriptions for certain parts for your Part-Master dataset.

The basic steps to do this are to load the new prices and descriptions into a `Table`, index by the product number (`prodno`), then `Extract` the price field from each record and replace it with a `$lookup` on the table.

Here is the Suprtool code:

```
>table newprices, prodno, file, bosslist, data(price, desc)
>get part-master
>if $lookup(newprices, prodno)
>update
>extract price = $lookup(newprices, prodno, price)
>extract desc = $lookup(newprices, prodno, desc)
>xeq
```

We do the `If $lookup` to select only the parts which have new prices, then do `Extract` with `$lookup` to replace the existing price with a new one. The `Update` command forces a database update on each selected record and must come before the `Extract` command.

If you did not specify the `If $lookup`, then records that did not qualify under the `$lookup` function in the `extract` field, will result in zeroes for any numeric field and spaces for any byte type fields.

\$Number Function

Suprtool now has the ability to accept free-form "numbers" as display data types. This means numbers in the form:

```
1234.45-  
-12345  
-123.2134  
12343  
$123.45
```

can now be accepted and converted to any other numeric data type. Consider the following data:

Item-number	New-Price
12345	+123.45
34563	+ 27.5
21312	+ 1.545

Suprtool can now read and convert the data in New-Price using the number function. Let's say we want New-Price to be a double integer but currently occupies eight bytes starting in position six. Here is the task you would use to convert the New-Price free-format number into a double integer.

```
>in mynums  
>def item-number,1,5,byte  
>def new-price-ascii,6,8,display  
>def new-price,1,4,double  
>item new-price-ascii,dec,2  
>item new-price,dec,2  
>ext item-number  
>ext new-price=$number(new-price-ascii)  
>out somefile,link  
>xeg
```

The \$Number function will take the free-format number and make it a valid display number. It will determine the decimal, sign and add leading zeroes. It will round the number to the defined number of decimal places.

In the case of the 1.545 number, Suprtool will round the value to be 1.55, since the given number of decimal places is two and the following value is five or greater. If you have a whole number such as 54, with no decimal point the value becomes 54.00.

Suprtool will not accept data that has:

```
More than one sign.  
More than one decimal place.  
Spaces in between numbers.  
Signs that are in between numbers.  
Characters that are not over punch characters.  
Fields that when edited do not fit in the defined space for the  
display field.
```

You can control the character that defines the currency, thousand and decimal symbol for other currencies and formats using the following commands:

```
>set decimalsymbol "."  
>set thousandsymbol ","  
>set currencysymbol "$"
```

Suprtool in the above case will strip the currency and thousand symbols and use the decimal symbol to determine the number of decimal places. You can set these characters to any values you want but the defaults for each are used in the above set commands. The decimal and thousand symbols are only single characters. The currency symbol allows for four characters.

Suprlink Join Command

Suprlink can now join files together that have multiple key records in each file, what has been come to be known as a many-to-many link. Suprlink has traditionally been able to link an Input file with many records with the same key to a Link file that has a single record with the same key value.

The Join command, will now link two files with many key records in both the input file and the "Linking" or join file. The syntax of the Join command is exactly the same as the Link command so a sample task would look as follows:

```
+input ordhist
+join orders
+output custord
+xeq
```

The above task will link multiple records of the file ordhist, to the multiple records of the file in orders. This assumes that the files are sorted by a common key. In SQL terms this is known as an Inner Join. An Outer Join, one where the keys do not necessarily have a match can be achieved by adding the optional keyword to the Join command:

```
+input ordhist
+join orders optional
+output joined
+xeq
```

In SQL parlance, once again you can achieve both a Left Outer Join and Right Outer Join by reversing the order of the files, between the input and the join commands.

To give you an example of how the Join operation would work consider the following data. First we have an inventory file with multiple records for the same product-no. This data is stored in the file dinv:

```
50512001 {Rest of data}
50512001 {Rest of data}
50512003 {Rest of data}
```

The next file will have sales records, once again with multiple key values, this data is stored in the file dsales:

```
50512001 {Rest of data}
50512001 {Rest of data}
```

If you did the following task assuming both files are sorted by the product-no:

```
+in dinv
+join dsales
+out invsales
+xeq
```

The resulting file would have four records, with the multiple matching dinv and dsales records. The record layout would have the dinv information first followed by the dsales information. If you add the optional keyword on the join command the resulting file would have 5 records. The matching 4 records from dinv and dsales as well as the dinv record that did not match with the numeric fields set to zero and the byte fields set to spaces.

Only one Join operation is allowed per task.

By default, Suprlink will join files based on the primary sorted key in the self-describing file. You can specify a secondary key for the files to be joined on in a similar manner to how the Link command did:

```
+in orders
+join dsales by order-no product-no
+out ordsales
+xeq
```

\$Split Function

Suprtool can extract portions of a byte field based on the occurrence of certain characters.

Consider the following data:

```
Armstrong/ Neil/ Patrick
Green/ Bob/ Miller
Fritshaw/ Elizabeth/
Edwards/ Janine/
Armstrong/Arthur/Derek
```

The \$split function can extract each token into separate fields. The syntax for the \$split function is:

```
$split (Field, Start Character, Occurrence, End Character, Occurrence)
```

The following task will split the data in the wholename field into three separate fields. This task assumes that the file namefile is a self-describing file with a field called wholename.

```
>in namefile
>define lastname,1,30
>define firstname,1,20
>define middlename,1,20
>extract lastname = $split(wholename,first,"/")
>extract firstname=$trim($split(wholename,"/","/"))
>extract middlename=$trim($split(wholename,"/",2," ",2))
>out names,link
>xeq
```

The first extract statement tells Suprtool extract the bytes from the field wholename, starting at the beginning (first keyword), and stopping at the "/" character. The second extract statement, tells Suprtool to extract the bytes between the first occurrence of the "/" character to the next occurrence of the "/" character, and then that string is trimmed of spaces as it is nested within the \$trim function.

The third and final extract statement tells Suprtool to extract the bytes beginning with the second occurrence of the "/" character to the second occurrence of the space character.

If the target field is not long enough to hold the data Suprtool will abort with an error. You can easily prevent this from happening on blank fields by nesting the \$split statement within a \$trim or \$rtrim function.

\$SubTotal Function

Suprtool has the ability to keep a running subtotal for any numeric field based on a given sort key. The target data must be a packed field with 28 digits, in order to help avoid overflow issues.

A sample use of the \$subtotal function could be:


```
>def mytotal,1,14,packed
>get orders
>sort order-number
>ext order-number
>ext part-number
>ext description
>ext sales-amount
>ext mytotal = $subtotal(sales-amount,order-number)
>out sales,link
>xeg
```

This would result in a file containing a running subtotal in the field mytotal for a given order-number. You could then generate a simple report with the simple Suprtool commands:

```
>in sales
>list standard
>xeg
```

The basic syntax for the \$subtotal function in the extract command is:

```
extract targetfield = $subtotal(field,sort-field)
```

You must specify the sort command before referencing the sort-field in the \$subtotal function. You can subtotal up to ten fields per pass and the \$subtotal function is also available in the if command, however, is of limited use.

Cleaning your Data

In this day and age of migrations we were looking at issues that customers have run into when importing data into new databases. What came from this investigation where ways to clean up your data in any given byte type field.

We have added two methods to clean your data, you can use Suprtool to clean an individual byte type field, or STExport to clean all of the byte-type fields for a given file that you are exporting.

Suprtool

Sometimes un-printable or extraneous characters get stored in files or databases that have no business being there. This may be some tab characters in an address field or perhaps an embedded carriage return or line-feed. Suprtool now supports the \$Clean function which will replace individual characters for a given byte field.

There are three things that Suprtool needs to know in order to "clean" a field. Suprtool needs to know which characters it needs to clean, what character it needs to change the "bad" characters to, and also what field does it need to clean.

Defining a Clean Character

The Clean command is used to tell Suprtool what characters it needs to look for in a given byte type field. For example:

```
clean "^9", "^10", "."
```

will tell Suprtool to replace the tab character (Decimal 9), Line Feed (Decimal 10), and a period to whatever the CleanChar character is set to. The Clean command takes both, decimal notation and the character itself, however, it is probably most convenient to use the decimal notation for the characters that you wish to clean, since most are un-printable. The decimal notation is indicated by the "^" character.

The Clean command also has a special keyword which defines special characters Decimal 0 thru Decimal 31 via the command:

```
Clean special
```

You can also specify a range or characters by using the following syntax:

```
Clean "^0:^31", "^240:^255"
```

Setting the Clean Character

By default, Suprtool will replace any of the characters specified in the clean command with a space. You can specify what character to use to replace any of the characters that qualify with the following set command:

```
>set CleanChar "."
```

This will set the character to replace any of the qualifying "to be cleaned" characters to be a period.

Cleaning a Field

You call the clean function, the same way you normally use other functions available to if and extract. For example:

```
ext address1=$clean(address1)
```

shows how to clean the field address1. You do not necessarily need to have the target field be the same as the source field.

```
def new-address,1,30  
ext new-address=$clean(address1)
```

Cleaning your data

An example of how easy it would be to clean your database of certain "bad" characters in byte-type fields would be as follows:

```
>base mydb,1,  
>get customer  
>clean "^9", "^10", "^0", "^7"  
>set cleanchar " "  
>update  
>ext address(1) = $clean(address(1))  
>ext address(2) = $clean(address(2))  
>ext address(3) = $clean(address(3))  
>xex
```

The above task will look at the three instances of address and replace the tab, linefeed, null and bell characters with a space.

STExport

This same feature has been added to STExport, except that STExport will automatically clean all the byte type fields for a given SD file. The commands are very similar, except STExport just needs to know what the replace character should be and what characters it needs to look for.

```
$ in mysdfile
$clean "^9","^10","^0","^7"
$set cleanchar " "
$out myexport
$xeq
```

Since the Cleanchar is by default set to space, the above task could simply be:

```
$in mysdfile
$clean "^9","^10","^0","^7"
$out myexport
$xeq
```

Outcount and Fullcount

Suprlink and STExport on HP-UX now have a method for reporting how many records have been output. In Suprtool, we wrote the number of records to a file called .stoutcount. In Suprlink and STExport we used the files, .sloutcount and .sxoutcount respectively.

For Suprlink:

```
&#!/bin/sh
&#
suprlink -oc << !EOD
+in orders
+join ordhist
+out ordcomb
exit
!EOD
if [ `cat .sloutcount` -ge 10 ]; then
    echo "More than 10 records found"
fi
```

For STExport:

```
&#!/bin/sh
&#
stexport -oc << !EOD
$in orders
$heading fieldnames
$out ordprn
exit
!EOD
if [ `cat .sxoutcount` -ge 11 ]; then
    echo "More than 10 records found"
fi
```

STExport Escape Command

Many SQL importers allow you to add an escape character in front of characters that may mean something else to the import program. For example if the import program thinks that the delimiter character is a comma, the importer may treat a comma in an address field as an indication to move to the next field, which will throw off the import.

Some import programs, will treat the next character as data as opposed to a delimiter if the character is preceded by an escape character, such as a slash. Thus when the field is analyzed by STExport the data that originally started as:

```
"Niagara Falls,Ontario, Canada"
```

would be transformed to be:

```
"Niagara Falls/,Ontario/, Canada"
```

This function will not work on fixed columns and can be invoked with the escape command:

```
escape delimiter quote eol "/"
```

The above command will take the defined delimiter, quote and Eol and escape with a "/", if found in any byte type field.

Table Filename

Previously the permitted length for the filename for the Table command was 36 characters. This has been increased to 80 characters.

\$Total Function

Suprtool now has the ability to keep a running total for any numeric field. The target data must be a packed field with 28 digits, in order to help avoid overflow issues. A sample use of the total function could be:

```
>def mytotal,1,14,packed
>get orders
>ext mytotal = $total(sales-amount)
>xeg
```

You can total up to ten fields per pass and the \$total function is also available in the if command, however, is of limited use.

\$Counter Function

For years Suprtool has had the ability to output a record number to an output file with the num option of the output command:

```
>in mysdfile
>out myfile,num,data
```

The above task would generate an output file called myfile, however, you would lose the SD information and you can only put the number at the beginning or the end of the data. Suprtool now has a counter function that allows you to place a \$counter at any spot as well as preserve the SD information.

```
>in mysdfile
>def mycount,1,4,double
>ext field1
>ext field2
>ext mycount=$counter
>out myfile,link
>xeg
```

The file myfile will be self-describing and contain the fields field1, field2 and mycount. The field mycount is defined as a double integer, since this is the only field type that the \$counter function can use. Each record will have a unique ascending number starting with one.

Enhancements in Version 4.6

Overview

Every year we provide Suprtool/UX users with new features. The following section describes the new enhancements to Suprtool since the last major release.

Support for Eloquence

Suprtool for HP-UX now supports Eloquence. The only version that Suprtool 4.6 has been tested with is A.06.31.

Eloquence is a database that runs on HP-UX, which has an Image like intrinsic layer. For more information on this database please visit their web site at:

<http://www.hp-eloquence.com/>

We are interested in people wanting to have their old familiar Suprtool for MPE commands, such as Base, Get, Update, Delete and Put.

The syntax for the commands in the HP-UX version is identical to those commands on MPE.

Bugs Fixed

Bugs Fixed In Suprtool 4.7

Incorrect record Number. Suprtool would incorrectly report the record number when an Illegal Ascii digit was encountered on a Duplicate operation. We no longer attempt to show the record number from the Input source.

Set Limits Tablesize. Suprtool would report an error on a second Set Limits TableSize command if the value given was larger than the previous.

Extract from a Table. Suprtool would incorrectly report an error in some cases when doing an update from a table. The error "Field offset is beyond the input record length" was incorrectly hit if the input file was smaller than the actual length of the table file.

\$Counter Function. Suprtool did not correctly reset the \$counter variable in between tasks.

Define Commands. The number of Define commands allowed has been increased to 768 defines. The number of defines allowed used to be a variable number dependant on various system settings, this was changed in Suprtool 4.4.10 to be a fixed structure.

Bugs Fixed in Version 4.6.01

Packed and Display Coercion. Suprtool would incorrectly coerce large negative numbers from one packed or display field to another packed or display field.

Input Filename (start/end). Suprtool would fail with prefetch point failure if the file had an uneven record size and an uneven blocking factor.

HTML and XML commands. STExport now properly converts the Roman 8 universal monetary symbol to the Euro-Symbol in the HTML and XML commands.

The XML command would fail to convert ">" to ">" and "<" to "<".
The XML and HTML commands failed to convert "&" to "&".

Bugs Fixed in Version 4.6

Dividing Zero by Zero. Some division operations would result in a large number when dividing zero by zero if the target type was zoned, packed or quad integer.

Divide Operations. Suprtool could not properly convert the result of a division operation if the target was zoned or packed, in one specific case.