Suprtool 4.6 for HP-UX:

Change Notice

by Robelle Solutions Technology Inc.

Contents

Bugs Fixed   17

Introducing Suprtool Version 4.6

Overview

Use Suprtool/UX to read, select, and sort data from HP Eloquence, Oracle and Allbase databases and from 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 Version 4.6

- Support for HP Eloquence databases has been added.

- Suprtool incorrectly rounded Real and Long target data in arithmetic expressions.

Highlights in Version 4.5

- 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.

Highlights in Version 4.4

- Suprtool now supports string expressions in both the Extract and If commands. String expressions allow users to select and format their byte-type data in numerous new ways. This includes combining byte-type fields together, upshifting and downshifting data, and trimming blanks. These new features combine to reduce the number of tasks needed to select and format the data the way you need it.

- Suprtool now supports 5,100 bytes of constants to be extracted. (only NM and HP-UX)

- Suprtool, Suprlink and STExport now have the ability to create their own persistent redo stacks. The contents of these stacks can be recalled not only in the current session, but also in a future session.

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*, 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.

Suprtool for HP-UX typically comes with two versions in two different directories on your tape. The version of Suprtool in /opt/robelle is compatible with HP-UX 10.20 and later. The version of Suprtool in /usr/robelle is compatible with versions earlier than HP-UX 10.20.

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, call Robelle for assistance.

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.

Online Documentation in HTML Help Format

The Suprtool user manuals and change notice are available in HTML Help format. Support for compiled HTML help (CHM files) is built into recent versions of Windows.

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 of 4.6

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 7 a.m. to 4 p.m. Pacific time at 1.604.582.1700. Technical support can also be obtained via e-mail at: support@robelle.com

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 4.6

Overview

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

HP Eloquence

Suprtool for HP-UX now supports HP Eloquence which is a database similar to IMAGE on the HP 3000, but is available on HP-UX, Linux and Windows. The only version that Suprtool 4.6 has been tested with is A.06.31. HP 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/
```

Suprtool for HP-UX now supports the standard Suprtool for MPE Image 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.

Enhancements in Version 4.5

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.

Set Warnings Off

Suprtool, STExport and Suprlink have a new command called Set Warnings which when turned off will no longer print warning messages if you are running in batch.

In Suprtool, you can simulate batch mode with the command Set Interactive off. If Set Warnings is off, Suprtool will honour the value of the Interactive setting, provided Set Warnings off is set after the Set Interactive command.

XML Command

STExport will now generate "well-formed" XML output with just a few commands.

```
/opt/robelle/bin/stexport
$input file1sd
$xml
$output myxml
$xeq
```

These four simple commands will generate the following file that packages such as XMLSpy will consider to be "well-formed" XML. The result of such an STExport task will look as follows:

```
<?xml version='1.0'?>
<file>
<record>
<CITY>Los Altos</CITY>
<CREDIT-RATING>100000</CREDIT-RATING>
<CUST-ACCOUNT>4003302</CUST-ACCOUNT>
<CUST-STATUS>20</CUST-STATUS>
<NAME-FIRST>Ralph</NAME-FIRST>
<NAME-LAST>Perkins</NAME-LAST>
<STATE-CODE>CA</STATE-CODE>
<STREET-ADDRESS>Room 655</STREET-ADDRESS>
<STREET-ADDRESS>Los Altos     040033022</STREET-ADDRESS>
<ZIP-CODE>93002</ZIP-CODE>
</record>


</file>
```

By default STExport will add the simplest version tag at the beginning of the file, then it inserts a <file> and matching </file> at the beginning and the end of the file. Then STExport encloses each record from the input file in a <record> and </record> tag. Finally, the Self-Describing tags are added around each field's data values and edited appropriately.

Naturally users would want options to customize and specify the various options and tags themselves, in order to generate a file that is acceptable to their tools.

You can specify the "version" tag at the beginning of the XML file with the following command:

```
xml version "?xml version='1.0' encoding='ISO-8859-1'?"
```

STExport will put the "<" and ">" around what is specified in the version string. In addition, you can customize the "file" and "record" tags with the following simple commands:

```
xml file "orders" record "orderdetail"
```

You can enter multiple XML commands per task to set the XML options you require.

```
$in file1sd
$xml version "?xml version='1.0' encoding='ISO-8859-1'?"
$xml file "Orders" record "Details"
$out myfile
```

An example of the output generated by the above commands is as follows:

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<Orders>
<Details>
<CITY>Los Altos</CITY>
<CREDIT-RATING>100000</CREDIT-RATING>
<CUST-ACCOUNT>4003302</CUST-ACCOUNT>
<CUST-STATUS>20</CUST-STATUS>
<NAME-FIRST>Ralph</NAME-FIRST>
<NAME-LAST>Perkins</NAME-LAST>
<STATE-CODE>CA</STATE-CODE>
<STREET-ADDRESS>Room 655</STREET-ADDRESS>
<STREET-ADDRESS>Los Altos     040033022</STREET-ADDRESS>
<ZIP-CODE>93002</ZIP-CODE>
</Details>
</Orders>
```

In XML the tags that surrounded the data can not have any special characters other than hyphen, underscore and period ("-","_", "."). So STExport replaces any of the invalid special characters with a "." by default. You can change the default character to be something else with the following set command:

```
$Set xmltagchar "_"
```

STExport will only allow the hyphen, underscore and period to be set with this command.

A Document Type Declaration can be made at the beginning of the file via the !DOCTYPE specification. This typically tells whatever tool that is parsing the xml file where the DTD for the file resides.

In STExport you can specify simple one-line doctype specs with the following command:

```
xml doctype '!DOCTYPE address-book SYSTEM "address-book.dtd"'
```

This will write the doctype specification at the top of the output file, directly after the XML version specification.

All of STExport's XML command options (version, doctype, file and record) allow for a string to be passed via surrounding quotes. The quotes may be either single or double, but keep in mind that if the string is to contain double quotes, then you should surround the entire string with single quotes.

So to summarize this new feature the following commands have been added for XML support. The XML command has the following options:

XML
      VERSION *"string"*
      DOCTYPE *"string"*
      FILE *"string"*
      RECORD *"string"*

The following set command has been added to aid in the conversion of special characters in the tag:

```
set xmltagchar "."
```

Enhancements in Version 4.4

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.

Extracting Constants

Previously Suprtool was limited to 1,530 bytes of constants for MPE/iX and HP-UX versions of Suprtool. This limit has now been increased to 5,100 bytes. The MPE/V version of Suprtool is still limited to 1,275 bytes.

Persistent Redo

Commands entered at the Suprtool prompt are saved in something called the redo stack. You can recall commands from this stack by using other commands such as Before, Do and Redo. By default, the redo stack is stored in a temporary file and discarded as soon as you exit Suprtool. This temporary stack is not preserved across Suprtool invocations.

The new Set Redo command assigns a permanent file as the redo stack, allowing the stack to become available for future Suprtool invocations. For example, to assign the myredo file to a persistent redo stack, enter

```
>set redo myredo
```

If the file does not exist, Suprtool creates it. Otherwise, Suprtool uses the existing file. All subsequent commands are written to the persistent redo stack. This setting is valid for the duration of the Suprtool session. As soon as you exit Suprtool, the setting is discarded. Next time you run Suprtool, you will get the temporary stack.

If the file name is not qualified, the redo stack is created in the current working directory. This may be desirable if you want to have separate stacks. If you want to always use the same persistent stacks, you should qualify the name.

The Verify command shows which stack is currently in use. If it shows <temporary>, it means Suprtool is using the default stack. Anything else is the name of the file used on the Set Redo command.

Concurrency

When Suprtool uses the default temporary stack, it is accessible only to that particular instance of Suprtool. You can run as many Suprtool instances as you need and each one gets its own redo stack. With temporary stacks, you will never have concurrency problems.

If you start using a persistent redo stack, however, you might start running into concurrency problems. A persistent redo stack can be used by only one Suprtool instance at a time. If you try to use a persistent redo stack that is already in use, you will get the following message:

```
>set redo myredo
The redo file is already in use.
Unable to open file for REDO stack
```

In this situation, Suprtool continues to use the redo stack active at the time and lets you continue working as normal.

Qedit can also have permanent redo stacks. To prevent products from writing to each other's redo stack, it is advisable to use separate stacks for each product by giving them different file names. For example if you use

```
>set redo myredo
```

you will have a redo stack called myredo for your Suprtool commands. If you exit Suprtool, then run Qedit and supply the same Set Redo command, your Qedit commands will be written to the same file that was used for your Suprtool commands.

String Expressions

Both the Extract and If commands have been enhanced to allow string expressions, which can be used to:

1. Combine two fields together (using the + operator).

2. Remove spaces (using the built-in trim functions).

3. Upshift or downshift characters.

String expressions reduce the number of tasks required for many common operations involving byte-type fields. Fewer tasks means that Suprtool delivers data to your applications faster than ever before.

These changes are so extensive that string expressions are described separately for the Extract and If Commands. See the sections below for specific details and examples of string expressions.

Extract Command

You can now use string expressions in the Extract command. String expressions allow you to combine byte-type fields together (using the + operator) or operate on byte-type fields with special functions. In many cases, you can now reduce the number of tasks required to format the data the way you want. Fewer tasks means that Suprtool delivers the data where you need it faster than ever before. To extract a string expression, use this syntax:

EXTRACT *target-field* = *expression*

Target-Field

The *target-field* determines the byte-length for the expression. The data-type must be Byte or Char. The expression is extracted during the output phase and cannot be used by other Suprtool commands that accept fields (e.g., Sort).

Examples

```
>extract id-no = warehouse-no + bin-no

>extract full-name = first-name + last-name
```

Constants vs. Expressions

If you have an string expression that starts with a string, Suprtool assumes that you are attempting to extract a single string value and not a string expression. To specify a string expression that starts with a constant, surround the expression with parentheses. For example,

Incorrect

```
>extract name = " " + product-desc

Error:  Missing comma or invalid arithmetic expression
```

Correct

```
>extract name = (" " + product-desc)
```

Variable Length Strings

String expressions use variable-length strings. Suprtool keeps track of the length of every string, and all operations are done using the actual string length. For fields, the length of the string is the length of the field. If you do not want to retain all of the spaces in a field, use one of the built-in trimming functions.

String constants are created with the exact length of the constant. For example, the string "abc" is three characters long and the string "a" is one.

When assigning the string expression to the target field, Suprtool pads the final string value with spaces to fill out the target field. String expressions longer than the target field generate an error.

```
>in testfile
>def a,1,10,byte
>ext a="I'm too long for this container"

Error:  String is too long for the specified item
```

String Truncation

Suprtool produces an error if the string expression is longer than the target field. You cannot override this error with Set Ignore On. To help avoid the error, you may want to trim trailing spaces from the expression before assigning it to the target field. For example,

```
>extract new-field = $trim(a + b + c)
```

Upshifting Strings ($Upper)

Use the built-in function $upper to upshift all of the characters of a string expression into uppercase characters. This function can be used to upshift a single field, a complicated string expression, or any subpart of an expression. Both ASCII and Roman-8 characters are upshifted by $upper. For example,

```
>extract city-up = $upper(city)

>extract full-name = $upper(first + last)

>extract desc = desc-1 + $upper(desc-2)
```

Downshifting Strings ($Lower)

If you want to downshift all characters of a string expression to lowercase, use the built-in function $lower. This function can be used to downshift a single field, a complicated string expression, or any subpart of an expression. Both ASCII and Roman-8 characters are downshifted by $lower. For example,

```
>extract city-lower-case = $lower(city)

>extract city-state = $lower(city + state)

>extract desc = desc-1 + $lower(desc-2)
```

Trimming Spaces ($Trim, $Ltrim, $Rtrim)

Use one of three built-in string functions to remove leading or trailing spaces from a string expression. The three functions are:

$Trim:	Remove leading and trailing spaces from the string expression.
$Ltrim:	Remove leading spaces.
$Rtrim:	Remove trailing spaces.

If Command

You can do comparisons with byte-type fields in numerous ways using Suprtool. These powerful features minimize the number of tasks you must execute in order to select the data you need. The fewer the number of tasks, the faster your data is delivered to the users and applications that need it.

You can combine byte-type fields together and use the built-in string functions to create string expressions. String expressions involve the + operator and any of the built-in string functions, which are $lower, $upper, $trim, $ltrim and $rtrim.

Fixed vs. Variable Length Strings

String comparisons are done using fixed- and variable-length strings. For most users, there should be no difference between the two types of strings. When doing string comparisons, Suprtool always pads shorter strings with spaces, with the one exception of comparing two fixed-length fields (see "Byte Fields" below).

String expressions involving the + operator or the $lower, $upper, $trim, $ltrim and $rtrim built-in functions are done using variable-length strings. Suprtool keeps track

of the length of every string, and all operations are done using the actual string length. For fields, the length of the string is the length of the field. If you do not want to retain all the spaces in a field, use one of the built-in trimming functions.

When creating string expressions, string constants are created with the exact length of the constant. For example, the string "abc" is three characters long and the string "a" is one.

Byte Fields

For historical reasons, comparing two byte-type fields to each other is a special case. If the two fields are exactly the same length, Suprtool compares them completely. If one field is shorter, the comparison is done for the length of the shortest field. Suprtool does not check for spaces in the trailing characters of the longer field. For example,

```
>define  short, 1,10                        {10-character field}
>define  long ,11,15                        {15-character field}
>if      short = long
```

In this example, Suprtool compares the 10 bytes in the short field with the first 10 bytes of the long field, but ignores the last five bytes of the long field. If the expression on either side of the equal sign consisted of more than one field (using the + operator) or involved any of the built-in string functions, such as $lower, $upper, $trim, $ltrim and $rtrim, Suprtool would have compared both sides of the equal sign by padding the shorter field with spaces. It is only the case where you are directly comparing one byte-type field to another that Suprtool uses the length of the shortest field for the comparison.

Trimming Spaces ($Trim, $Ltrim, $Rtrim)

Use one of three built-in string functions to remove leading or trailing spaces from a string expression. The three functions are:

$Trim: Remove leading and trailing spaces from the string expression.

$Ltrim: Remove leading spaces.

$Rtrim: Remove trailing spaces.

Because Suprtool pads shorter strings with spaces when doing comparisons, trimming spaces is most useful when creating a combined string with several fields. For example, you might want to combine a person's first and last name (including a space between the two):

```
>if $trim(first) + " " + $trim(last) = "Joe Smith"
```

Mixed Case ($Upper and $Lower)

By default, Suprtool does an exact match when comparing two string expressions. If the expressions vary in the capitalization of characters, Suprtool will find them unequal. To do caseless string comparisons or pattern matches, use the $upper or $lower functions. Both ASCII and Roman-8 characters are shifted by $upper and $lower. For example,

```
>if $upper(city) = "VANCOUVER"

>if $lower(city) = "edmonton"
```

Note that if you use the $upper or $lower functions, Suprtool does not shift any
constants in the comparison. You must explicitly specify the constants in the correct
case or you can use $upper or $lower with the constant:

```
>if $upper(city) = $upper("vancouver")
```

Use the $upper or $lower functions for caseless pattern matching. As with other
comparison operators, you must specify constants in the correct case when doing
pattern matching:

```
>if $upper(city) == "VAN@"

>if $lower(city) == "ed@"
```

You can use $upper and $lower with string expressions that combine many fields
and string functions. In the following example, we create

```
>if $read
-    $upper($trim(first) +
-            " "                +
-            $trim(last))
-    = "JOE SMITH"
-
```

Bugs Fixed

Bugs Fixed in Version 4.6

Real and Long Arithmetics. Suprtool incorrectly rounded Real and Long target data in arithmetic expressions

Bugs Fixed in Version 4.5

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.

Extract Character Constant. If you had a extract of a character constant, and if the constant specified was not completely filled in for all of the positions in the length of the field, Suprtool would would produce strange results.

Define Command. A warning message has been added in the case where fields defined as non-standard integers, will be treated as strings.

Extract Command. The error message that prints when the expression specified cannot be coerced/converted into the target has been improved.

If $Read Operations. Suprtool would incorrectly put commands entered thru the $read facility into the redo file or stack when executed thru a Use file.

Table Command. Suprtool would not hold on to a table in some cases when the hold option was specified in the table command. The CM and HP-UX versions of Surptool would not hold on to the correct table in some cases.

Verify Command. Verify Define will now show the correct information when defining non-standard integers.

Bugs Fixed in Version 4.4

Add Command. The Add command no longer fails in the following cases:

Extracting byte-type fields with constants and the literal string values are shorter than the length of the field.

A self-describing file with more than one field is used as input.

Arithmetic Operations. Suprtool now has a better error message when an arithmetic operation has an invalid decimal operand length.

Extract Command. When extracting constants, Suprtool no longer incorrectly reports that the constant limit has been reached.

Extract Command. When extracting one field into another, invalid data is no longer produced if the two fields are exactly the same length and type, but have a different number of implied decimal places.

Extract Command. When the maximum limit of extracted constants is reached, Suprtool no longer incorrectly reports the following message:

```
Error:  Number of constants in >EXTRACT exceeds maximum
```

This error message has now been changed to:

```
Error:  Total size of constants in >EXTRACT exceeds maximum
```

Set Ignore On. Suprtool now honors the Set Ignore On flag when encountering an illegal packed decimal number in the $stddate function.