Suprtool 4.7 for HP e3000:

Change Notice

by Robelle Solutions Technology Inc.

Robelle Solutions Technology Inc.
13711 - 72nd Avenue, Suite 372
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

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. This change notice provides you with detailed installation steps and a description of Suprtool's new features.

Highlights in Suprtool 4.7

- Suprtool would report an incorrect record number when encountering an Illegal ascii digit, if the Suprtool task involved the Duplicate command.
- Suprtool now has a $Number function which will allow Suprtool to use a freeform ascii number with signs, decimal places and currency symbol as a display field.
- 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.
- The Clean command in STExport and Suprtool has improved syntax to define a range of characters to replace.
- 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.
- Suprlink and STExport now report the number of output records in a manner similar to Suprtool.
- The $counter function was not reset in between tasks.
- Extract from a table would incorrectly report an error in some cases.
- 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 and STExport now support features to Clean your data.
- STExport now supports an Escape command which will escape out certain characters.
- 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 total a specified field.

Highlights in Suprtool 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 were 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 Suprtool 4.6

- Suprtool incorrectly rounded Real and Long target data in arithmetic expressions.
- The Table command incorrectly gave an error message in some cases when data was loaded into a Table.

Highlights in Suprtool 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.
- CM to NM switches in Suprtool have been decreased significantly.
- Suprtool now has the ability to load and extract data from a Table.
- Loading a Table is now up to 28% faster than previous versions.

- Table lookup operations are now up to 33% faster than previous versions.

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.

Suprtool has been changed to no longer save an output file if no records have been written and an error has occurred.

CPU Serial Number and CPU Name (HPSUSAN & HPCPUNAME)

Suprtool and it's associated products runs 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.

Compatibility Mode vs. Native Mode

Throughout this change notice and in all of the Suprtool documentation, there are references to compatibility mode and native mode. If you are on a Classic HP e3000 (i.e., not MPE/iX), you can use the compatibility-mode version of Suprtool. Our job streams automatically install the correct version for your machine. We use the terms MPE XL and MPE/iX interchangeably in our documentation.

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

If you need to install this upgrade in an account *other* than the Robelle account, use the installation instructions in the appendix at the end of this change notice.

The following installation instructions are for existing Suprtool users who are installing a new version in the Robelle account. Please use the instructions from this change notice instead of the user manual because the change notice always has the latest version. The whole process should take about 30 minutes, assuming no one uses Suprtool or its components during the installation.

To install this update of Suprtool, follow these steps:

1. Restore the new files from tape.
2. Upgrade the structure of the Robelle account.
3. Install the proper program files (NM or Classic).
4. Install Speed Demon in the system SL (only Classic).
5. Install the Suprtool2 interface in the system SL (only Classic).
6. Limit Suprtool's run time priority (optional).
7. Install QLIB and Bonus programs.
8. Purge the installation files (optional).

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/

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.

Important Note about Passwords

None of the jobs that we supply have passwords in them. Before streaming a job, you might have to add your system's passwords to the first line. Users of MPE XL 3.0 and later do not have to do this because the operating system prompts for missing passwords. The same is true for some MPE V users who have security software that inserts passwords. Most MPE V users have to edit the jobs. For example, if the system manager logon password is Qwerty, you would do the following:

```
:editor
HP32201A.07.22 EDIT/3000
/text robelle.job.robelle
FILE UNNUMBERED
/modify 1
MODIFY     1
!job robelle,manager.sys,pub;hipri
                         i/qwerty
!job robelle,manager.sys/qwerty,pub;hipri

/keep robtemp
/exit
END OF SUBSYSTEM
:stream robtemp
:purge robtemp
```

STREAMX Users

Users of STREAMX, a part of SECURITY/3000 from VESOFT Inc., must set a Job Control Word before streaming jobs. This prevents STREAMX from generating an error if the Robelle account does not yet exist. For example,

```
:setjcw streamxtempnonest 1
:stream robelle.job.robelle
```

Combined Qedit/Suprtool Tapes

If the tape you received has more than one product on it (e.g., Qedit and Suprtool), you can take a shortcut instead of following the complete instructions for both products.

Steps that are common to both installations need only be done once. That is, there is no need to restore twice, or setup the Robelle account twice, or install the Qlib and Bonus programs twice, or clean up unneeded files twice.

Note that if you do *not* use this shortcut it is not a problem; it will just take you a few minutes longer, and you will have to mount the restore tape a second time.

Say you install Qedit first, followed by Suprtool:

1. Do all of the Qedit steps, except for the final *Purge Installation Files* step.

2. Then do the Suprtool installation starting with the *:stream Install.Suprjob* step. Skip the initial *Restore* and *Robelle Accoount* steps, and skip the *Qlib/Bonus* step. Do the final *Purge Installation Files* step.

Step 1: Restore the Files

You begin by restoring all the files from the tape.

```
:hello manager.sys
:file rtape; dev=tape                    {use appropriate device}
:restore *rtape; @.@.robelle; create     {restore everything}
```

The files on the tape require approximately 120,000 sectors. If you are upgrading Suprtool, most of these files will replace existing files, because the new version of Suprtool does not take up much more space than the old version.

The Suprtool program files are located in the Pub group of the Robelle account. At this point in the installation process, the files do not yet have the correct names. The install job assigns the proper names to the program files.

Step 2: Set Up the Robelle Account

Even if you already have the Robelle account, you must stream the Robelle job to ensure that the Robelle account and all its groups have the proper security and capabilities.

```
:stream robelle.job.robelle
```

It's a bad idea to use the password suggested in the instructions. I.e., "hardpass".

When the job stream is finished, it will send you a message. The job removes the password from the Robelle account. Stay logged on as Manager.Sys and put a password on the Robelle account. If you are upgrading your Robelle account, use the old password.

```
:altacct robelle;pass=hardpass           {something hard to guess}
```

Note that during installation we add OP capability to the Robelle account. After installing Suprtool, you have the option to remove OP capability.

Step 3: Install Suprtool into Production

Our installation job installs Suprtool, STExport, Suprlink, Speed Demon, and the Suprtool2 interface. No one can use these programs during the installation. Before you stream this installation job, warn people not to use these programs for a while.

```
:hello mgr.robelle
:warn @;please exit from Suprtool and Suprlink NOW!
:stream install.suprjob
```

Once Suprtool has been installed, check the installation job $stdlist. If anyone was either using Suprtool, STExport, Suprlink, or Speed Demon, or attempting to back them up, the installation job will fail. Chase away any users and ensure that a backup is not in progress, then stream the installation job again.

The installation job renames your current versions of Suprtool and its components to the PubOld group of the Robelle account. If you need to move these versions back into production, use the Previous.Suprjob job stream.

You can now access Suprtool by entering

```
:run suprtool.pub.robelle
```

Optional Installation Steps

On MPE/iX your installation should be complete. If you have a copy of Speed Demon or the Suprtool2 interface in your own XL files, you will have to update them with the new versions (use the instructions in the Speed Demon and Calling Suprtool user manuals).

To use the new versions on MPE V, you still need to install Speed Demon and Suprtool2 in the system SL. MPE/iX users must also do this if they are calling Speed Demon or Suprtool2 from compatibility-mode programs. With so few changes to Speed Demon and Suprtool2, you can safely ignore them for this version of Suprtool.

Step 4: Speed Demon/V Installation (optional)

To use the new features of Speed Demon/V, you must install the latest version in the system SL. You should follow the installation instructions in the Speed Demon user manual. These instructions assume that you have used the Install.Suprjob job stream to move Suprtool and its components into the Pub group of the Robelle account.

```
:stream demon.suprjob.robelle
```

Step 5: Suprtool2 Interface Installation (only Classic)

The Suprtool2 interface has been revised. On MPE V you must re-install the interface. If you do not have the latest Suprtool2 interface, programs that attempt to use totals from Suprtool2 will fail. You can install this interface without recompiling any of your existing programs.

System SL Installation

To install the latest Suprtool2 interface in the system SL, use our standard installation job:

```
:stream suprcall.suprjob.robelle
```

User SL or XL Files

If you have a copy of the Suprtool2 interface in your own SL or XL, you will need to update it. For instructions on how to do this, consult the user manual for Calling Suprtool.

Step 6: Limit Suprtool's Run Time Priority (optional)

Some system managers like to limit the amount of resources that Suprtool consumes. One way to do this is by running Suprtool in the D queue. You can configure this using HP's Linkedit program.

```
:linkedit
>altprog suprtool;maxpri=ds
>exit
```

By forcing Suprtool to always run in the lower queue (even when Suprtool runs on-line), it competes less with on-line users, because most users run in the C queue. System managers should restrict the priority of Suprtool only if they are familiar with MPE process queues and the Tune command.

Step 7: Install QLIB and Bonus Programs

Suprtool comes with an array of contributed software in the QLIB library (in the QLIB groups of the Robelle account). QLIB programs may be used on any CPU and may be given away freely. Your Robelle license may also entitle you to receive our Bonus programs: Compare/iX, HowMessy, Select, Spell, and Xpedit. These programs reside in the Pub group of the Robelle account. Bonus programs can only be used on authorized machines, and you may not distribute them to anyone.

If you received Bonus programs with this version of Suprtool, use the job stream called Bonus.Job.Robelle to install both the QLIB and Bonus programs. If you did not receive Bonus programs, use the job stream Qlib.Job.Robelle to install the QLIB programs.

```
:hello mgr.robelle
:stream bonus.job.robelle                    {or Qlib.Job.Robelle}
```

If you skip this step, you might end up running old versions of these programs.

Step 8: Purge Installation Files (optional)

The easiest way to clean up after an installation is to stream the Cleanup.Purgejob.Robelle job. This job saves you disc space by purging the installation files. It does not purge the files that are necessary to run the software.

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.

$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 and 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
 >xeq
```

The $number function 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 know 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" 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 wholefield into three separate fields. This 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 now 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
>xeq
```

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
>xeq
```

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.

$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)
>xeq
```

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 would generate an output file called myfile, however, you would loose 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
>xeq
```

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.

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 and 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 clean 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. The decimal notation is indicated by the "^" character.

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))
>xeq
```

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
```

Clean Command Syntax

The Clean command has improved syntax to specify which characters to look to replace. You can specify 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"
```

This enhancement makes it much easier to define characters to search for and clean in both Suprtool and STExport.

Outcount and Fullcount

For years Suprtool has had a jcw and a variable to communicate how many records have been selected. The SuprtoolOutCount JCW and SuprtoolFullCount Variable, are set at the end of each task with how many records have been selected. Suprlink and STExport now have their own variables and jcw's.

They are:

```
SUPRLINKOUTCOUNT
SUPRLINKFULLCOUNT
STEXPORTOUTCOUNT
STEXPORTFULLCOUNT
```

On MPE/ix, it is recommended that you use the Fullcount variables when checking the number or records.

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 of the import.

Some import programs, will treat the next character as data as opposed to a delimeter 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 delimeter quote eol "/"
```

The above command will take the defined delimeter, 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.

Enhancements in Version 4.6

Introduction

This version is released mainly to fix bugs and to release the functionality in Suprtool for HP-UX which now includes support for HP Eloquence databases which are databases that are similar to Image on MPE.

Enhancements in Version 4.5

Introduction

Every year we provide Suprtool users with new features. The following section describes the new enhancements to Suprtool since Suprtool 4.4.

CM to NM Switches Decreased

Suprtool has dramatically decreased the number of CM to NM switches. In our internal testing we have seen reductions from between 8 and 24 times, depending on the task.

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 honor the value of the Interactive setting, provided Set Warnings off is set after the Set Interactive command. This feature is for those customers that have software packages that do analysis of your $STDLISTs. By preventing the "Warning:" messages from being printed, these Suprtool scripts will not get flagged as jobs that failed.

Table Improvements

Loading Tables Performance Improvement

Suprtool has the ability to load key-values into a table via the Table command. The loading of the key-values into the table has been improved by up to 28%.

If $Lookup Performance

The performance of tasks with an if $lookup operation have had a performance boost of up to 33%.

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 (MPE/iX only) 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.

XML Command

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

```
$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, when 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.

More complicated and lengthy doctype specifications can be implemented by using the filename option, whereby STExport will append the contents of a named file directly after the version specification. The syntax can be simply:

```
xml doctype myfile.xml
```

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@|2"string"|
##DOCTYPE@|2"string" &| <filename>|
##FILE@@@@|2"string"|
##RECORD@@|2"string"|
```

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

```
set xmltagchar "."
```

We are, as always, keenly interested in getting feedback on this new feature. Particularly what your company is doing with XML files and what applications or tools you are using to parse the xml files with. As always please send feedback to support@robelle.com.

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 Suprtool 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 "&&gt" and "<" to "&&lt".

The XML and HTML commands failed to convert "&&" to "&&amp".

Item Command. 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.

Bugs Fixed In Suprtool 4.6

Table Command. The Table command incorrectly gave an error message in some cases when data was loaded with key values.

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

Bugs Fixed In Suprtool 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 produce strange results.

Error Message. The error message on a failure to open the $stdin file has been improved.

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 Suprtool 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.

Variable Substitution. Suprtool did not handle resolving of system variables that are defined with no value.

Variable Substitution. Suprtool did not properly handle a variable that was only a comment.

Appendix: Non-Robelle Installation

Introduction

Suprtool is usually installed on your HP e3000 in the Robelle account (see the installation instructions earlier in this change notice). However, you can easily install Suprtool into another account by following these general steps:

1. Restore the files into your account instead of the Robelle account.
2. Stream the account upgrade job (with modifications).
3. Stream the installation jobs (with modifications).

For example, to install Suprtool into an account called Tools and into the same groups as Suprtool would have used in the Robelle account (Pub, Doc, Help, etc.), follow these steps.

Step 1: Restore Files

Restore the files into your account (Tools in this example):

```
:restore *rtape;@.@.@;create;account=tools
```

Step 2: Stream the Account Upgrade Job (with modifications)

With a text editor, change all account references in the Robelle.Job.Tools job from "robelle" to "tools." Don't worry about changes to some of the comments in the job. Note that this job will change the capabilities, access flags, and logon password of the Tools account. Examine the job carefully to ensure that the changes do not cause problems for other software in the Tools account.

```
:qedit                                          {or editor/quad/etc.}
/text robelle.job.tools
/change "robelle"(upshift) "tools" all
```

Step 3: Stream the Installation Jobs (with modifications)

Now log on to the Tools account instead of the Robelle account.

```
:hello mgr.tools
:qedit                                          {or editor/quad/etc.}
```

Before streaming any jobs, you must alter them so that they log on to the Tools account instead of the Robelle account. Make the following change to each job in the Suprjob, Job, and Purgejob groups:

```
/text install.suprjob
/change "robelle"(upshift smart) "tools" all
/keep
/exit
:stream install.suprjob
```

In the example above, we have used Robelle's Qedit text editor to make the required changes to the account name, but you can use any editor. Just make sure that all occurrences of "robelle" are changed, regardless of whether "robelle" is in uppercase, lowercase, or mixed case. But change "robelle" only when it is a separate word, not when it is embedded in a longer word.