Suprtool/Open 5.9.10

Change Notice

by Robelle Solutions Technology Inc.

Program and manual copyright © 1981-2017 Robelle Solutions Technology Inc.

Contents

Bugs Fixed   27

Introducing Suprtool Version 5.9.10

Overview

Suprtool/Open is a new version of Suprtool designed to be platform independant. Suprtool/Open is designed to read, select, and sort data from Oracle, and Eloquence databases and data files with fixed-length records. Suprtool/Open is designed to be similar to Suprtool for MPE and Suprtool for HP-UX.

Suprlink/Open 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 5.9.10

- STExport has a new set command called Set Excel Leadzero On which tells the Excel command to add leading zeroes to the fields specified in the Excel Preserve command.

- Suprtool has a new function available to the if/extract commands called $inrecnum, which expects a double integer result.

- Suprtool has a new function called $leadzeroz, which will add leading zeroes to a display field and will optionally justify the field.

- Suprtool has a four new string handling functions, specifically, $justifyl, $justifyr, $leadzerob and $respace.

Highlights in Suprtool 5.9

- A new $month function in the if / extract option allows you to add or subtract a number of months from a given date.

- STExport has a new command called Excel command which allows you to format a field in a format that allows leading zeroes or spaces to be preserved.

- A new option called JSON will output SD data in Java Script Object Notation.

- Suprtools new banner would show the incorrect day and day of week in certain circumstances. This was cosmetic only.

- The Add command would fail if a Table in an Oracle database accessible by a given username had more than 2.1 billion entries on Oracle 11 and higher.

- Suprtool/Open now has the option to read MySQL databases. We are looking for feedback on where to take this feature.

Highlights in Suprtool 5.8

- Set CleanChar to a single character, after a Set CleanChar "<null>" would not come into effect due to the special null flag not being reset. This was in both Suprtool and STExport.

- The input command now accepts the keywords($first/$last) when inputting a range of records.

Highlights in Suprtool 5.7

- Suprlink now has it's own "Suprmgr" file. Suprlink will process all the commands in /opt/robelle/linkmgr on startup.

- STExport now has it's own "Suprmgr" file. STExport will process all the commands in /opt/robelle/stexpmgr on startup.

- Set Comlog On has been added to Suprtool, Suprlink and STExport to log all commands entered in Suprtool, Suprlink or STExport are logged in it's own file.

- Suprtool would potentially have problems with system commands in variable-length scripts if the command ended in a number that had a length of eight numbers.

- Suprtool would fail with a write error in the case of an Input SD file with BigEndian Fields, Set EndianInt BE and if the output was to an SQL Table via the Add command and the output was sorted.

- Suprtool would lose track of Table data field information, on tables held with the hold option in subsequent tasks.

- Suprtool now has the $proper function which will shift the first character in a string and any first character after a space or ampersand.

- Suprtool now has the Translate command and a $translate function to obfuscate test data or any byte field from being readable.

- Suprtool/Open 5.7 Build 6 and higher supports the Oracle 12 client.

- Suprtool/Open would not parse a negative number into a quad integer container.

- Suprtool for Itanium and Suprtool/Open would not properly convert negative, single and double integers when output/,display is invoked.

- Suprtool for Itanium and Suprtool/Open incorrectly reallocated if/extract code space on subsequent tasks, which would eventually cause Suprtool to fail with the error, "Unable to allocate heap space."

Highlights in Suprtool 5.6

- Suprtool for Itanium and Suprtool/Open did not properly handle sorting of display type with over punch.

- Suprtool, Suprlink and STExport now handle 512 fields on HP-UX and Open platforms.

- Suprtools' extract command now has three special keywords to help with extract using a range, which is designed to make your Suprtool scripts more easily maintained. Suprtool now has $all, $first and $last which respectively means all fields, the first field or the last field. Note that $first and $last may only be used in a range extract.

- $subcount has been added to keep a running count for a given sort break.

- Suprtool had problems with brackets and arithmetic expressions for packed-decimal and display data-types. (Build 12)

- Suprtool had problems with improper rounding for Packed-decimal, Quad and Zoned-Decimal data types and arithmetic expressions. (Build 12)

- Suprtool would use an incorrect field definition when copying a file where the input file had a duplicate field name with different field definition, the second field would utilize the first definition.

- Suprtool/Open would potentially incorrectly set the decimal or date attribute on a straight copy of an SD file. This was introduced by the fix to copying SD files with duplicate fieldnames and was in 5.5.04 pre-release and 5.5.11 and 5.6 Build 1. This is fixed in Suprtool/Open 5.6 Build 3.

Compatibility

Suprtool/Open first release is designed to be compatible with Suprtool for HP-UX.

Documentation

The user manual contains the full description of all the Suprtool suite of products including Dbedit, Suprlink, STExport, and Suprtool2, as well as usage tips and commands for each. The manuals are up-to-date with all the latest changes. To see only the changes in the latest version, see the "What's New" section of the manual.

You can download our manuals and Change Notices in various formats and order printed (hardcopy) manuals from our web site at:

http://www.robelle.com/library/manuals/.

Installation

Overview

The following instructions describe the installation process of a new Suprtool release. The new version overwrites an existing version of Suprtool on your system.

Installation Instructions

There are typically two main types of installations. The first and most often utilized is the Download instructions. You can find the download install instructions here:

http://www.robelle.com/downloads/install-soprod.html

Installation Assistance

If you have any questions or run into any problems, please call us. Technical support is available on weekdays from 8 a.m. to 4 p.m. Pacific time at 1.800.453.8970. Technical support can also be obtained via e-mail at: support@robelle.com

Enhancements in Version 5.9.10

Introduction

Suprtool is constantly being updated with new features. The following section describes the new enhancements to Suprtool since Suprtool 5.9

$INRECNUM

The if / extract commands can now utilize a new function called $INRECNUM, which allows you to use the input record number in certain tasks. For example, the task below would find record number 11.

```
In somefile
If $inrecnum=11
```

You can also, utilize the $inrecnum function in the extract command:

```
In somefile
Def recnum,1,4,double
Ext recnum=$inrecnum
```

The $inrecnum function was designed to find records especially in the instance where duplicate records are in a particular data source and cannot be isolated by any other means.

describes the new enhancements to Suprtool since Suprtool 5.9

$LEADZEROZ

The if / extract commands can now utilize a new function called $LEADZEROZ, which allows you to add leading zeroes to a specific display field. This was designed specifically for the extract command and fixing up data but can be used in the if command as well.

$NUMBER is capable of fixing up numbers, but the new $LEADZEROZ function is more lightweight and simply adds leading zeroes, and has an option to justify right as shown below:

```
/PRINT LEADZERO
1
 2
  3
   4
    5
     6
      7
       8
       9
10
12
12345
220
RUN SUPRTOOL.PUB.ROBELLE

SUPRTOOL/Copyright Robelle Solutions Technology Inc. 1981-2017.
(Version 5.9.10 Internal)  Mon, Feb 27, 2017,  3:48 PM  Type H for
help.
>IN LEADZERO.SUPRTEST
>DEF A,1,12,DISPLAY
>EXT A
>OUT *
>XEQ
1
 2
  3
   4
    5
     6
      7
       8
       9
10
12
12345
220
IN=13, OUT=13. CPU-Sec=1. Wall-Sec=1.

>IN LEADZERO.SUPRTEST
>DEF A,1,12,DISPLAY
>EXT A=$LEADZEROZ(A,J)
>out *
>xeq
000000000001
000000000002
000000000003
000000000004
000000000005
000000000006
000000000007
000000000008
000000000009
000000000010
000000000012
000000012345
000000000220
IN=13, OUT=13. CPU-Sec=1. Wall-Sec=1.
```

The $LEADZEROZ function cannot fix issues like commas and decimal places in a
display field, this can be handled by the $number function.

$LEADZEROB

The if / extract commands can now utilize a new function called $LEADZEROB,
which allows you to add leading zeroes to a specific byte field. This was designed

specifically for the extract command and fixing up data but can be used in the if command as well.

```
/PRINT LEADZERO
1
 2
  3
   4
    5
     6
      7
       8
       9
10
12
12345
220
RUN SUPRTOOL.PUB.ROBELLE

SUPRTOOL/Copyright Robelle Solutions Technology Inc. 1981-2017.
(Version 5.9.10 Internal)  Mon, Feb 27, 2017,  3:48 PM  Type H for
help.
>IN LEADZERO.SUPRTEST
>DEF A,1,12,byte
>EXT A
>OUT *
>XEQ
1
 2
  3
   4
    5
     6
      7
       8
       9
10
12
12345
220
IN=13, OUT=13. CPU-Sec=1. Wall-Sec=1.

>IN LEADZERO.SUPRTEST
>DEF A,1,12,byte
>EXT A=$LEADZEROB(A,J)
>out *
>xeq
000000000001
000000000002
000000000003
000000000004
000000000005
000000000006
000000000007
000000000008
000000000009
000000000010
000000000012
000000012345
000000000220
IN=13, OUT=13. CPU-Sec=1. Wall-Sec=1.
```

$JUSTIFYL

The if / extract commands can now utilize a new function called $JUSTIFYL, which allows you to left justify text to the left side of a field.

```
>in leadzero
>def b,1,12,byte
>ext b=$justifyl(b)
>out *
>xeq
1
2
3
4
5
6
7
8
9
10
12
12345
220
IN=13, OUT=13. CPU-Sec=1. Wall-Sec=1.
```

$JUSTIFYR

The if / extract commands can now utilize a new function called $JUSTIFYR, which allows you to right justify text to the right side of a field.

```
>in leadzero.suprtest
>def a,1,12,byte
>ext a=$justifyr(a)
>out *
>xeq
           1
           2
           3
           4
           5
           6
           7
           8
           9
          10
          12
       12345
         220
IN=13, OUT=13. CPU-Sec=1. Wall-Sec=1.
```

$RESPACE

The if / extract commands can now utilize a new function called $RESPACE, which allows you to fixup byte data that has multiple spaces in between text.

```
>in respace.suprtest
>def text,1,40
>ext text
>out *
>xeq
this  is  a   test
 this is a         test
this  is  a    test
this         is a test
this   is   a    test
this is a test
 please note this   is   a      test
this  is   a    test
this    is        a       test
this is a test
       this    is    a    test
 this is a test
this    is    a       test
this is a test
dummy    record
this is a   not    test
silly    record
IN=30, OUT=30. CPU-Sec=1. Wall-Sec=1.

>in respace.suprtest
>ext text=$respace(text,J)
>out *
>xeq
this is a test
this is a test
this is a test
this is a test
this is a test
this is a test
please note this is a test
this is a test
this is a test
this is a test
this is a test
this is a test
this is a test
this is a test
dummy record
this is a not test
silly record
IN=30, OUT=30. CPU-Sec=1. Wall-Sec=1.
```

Enhancements in Version 5.9

Introduction

Suprtool is constantly being updated with new features. The following section describes the new enhancements to Suprtool since Suprtool 5.8.10

$Month

The if / extract commands can now utilize a new function called $Month, which will add a given number of months to a given date in the format of ccyymmdd or yyyymmdd.

For Example:

```
In somefile
Item mydate,date,ccyymmdd
Def targetdate,1,4,double
Ext targetdate=$month(mydate,+4)
```

The above task will take the field mydate and add four months to it. Suprtool will check if the date is valid and adjust the date within reason. For example if the given month for mydate has 31 days and the day is 31, and the month mydate becomes when the date is added to has only 30 days. The date will be adjusted to have the 30th for the day.

Excel Command

The Excel command can be used to produce columns of data that when imported will preserve spaces or leading zeroes.

> EXCEL PRESERVE *<fieldname>*

Example

STExport can generate columns that are imported into Excel in such a way that leading zeroes are preserved. While the format produced is not traditional CSV, the format will produce a field in the form:

```
="00055555"
```

This form when imported into Excel will preserve the leading zeroes. In order to invoke this format the Excel command has very simple syntax:

```
$in filexcel
$col fixed
$quote double
$zero leading
$excel preserve newchar int-field
$out *
$xeq
```

These simple commands will generate a file that will have the usually formatted fields as well as some fields formatted specifically for preserving spaces and leading zeroes in Excel.

The result of such an STExport task will look as follows:

```
="    11111 ",=" 01111", 0000011111,+00000011111
="    11111 ",=" 02222", 0000022222,+00000022222
```

Json Output

The JSON command specifies STExport to generate Json output. Use the JSON to produce Java Script Object Notation documents for either Internet or Intranet applications.

JSON

OBJECT "string"

ONEPERLINE

Example

STExport can generate JSON output with just a few commands.

```
$input file1sd
$JSON
$output myJSON
$xeq
```

These four simple commands will generate a file that can be read by various applications. The result of such an STExport task will look as follows:

[{"CHAR-FIELD":"11111","INT-FIELD":1111,"ZONED-FIELD":11111}]

Object

The Object option allows the JSON data to be wrapped in a specific Object description.

```
JSON Object "Json object"
```

Looks like this:

```
{"Json object":
[{\"CHAR-FIELD":\"11111",
\"INT-FIELD"\:1111,
\"DBL\-FIELD"\:11111,
\"PACKED\-FIELD"\:+1111,
\"PACKED\.-FIELD"\:+11111,
\"QUAD\-FIELD"\:11111,
\"ID\-FIELD"\:1,
\"LOGICAL\-FIELD"\:111,
\"DBLLOG\-FIELD"\:11111,
\"ZONED\-FIELD"\:11111
}]
}
```

Note that the example of the Output has one field per line with data. Normally this would have to be specified via the command line but the data is shown this way simply due to space constraints.

OnePerLine

For files that have many fields you may want to consider using the OneLine option of the JSON command:

```
JSON OnePerLine
```

STExport will put each field and data on one line with the appropriate beginning and end notation.

```
[{\"CHAR\-FIELD"\:"11111",
\"INT\-FIELD"\:1111,
\"DBL\-FIELD"\:11111,
\"PAC\KED\-FIELD"\:+11111,
\"PAC\KED\.-FIELD"\:+11111,
\"QUAD\-FIELD"\:11111,
\"ID\-FIELD"\:1,
\"LOG\ICAL\-FIELD"\:1111,
\"DBL\LOG\-FIELD"\:11111,
\"ZONED\-FIELD"\:11111
}]
```

Multiple Json Commands

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

```
$in file1sd
$JSON Object "Json object"
$JSON OnePerLine
$out *
$xeq
```

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

```
{"Json object":
[{"CHAR\-FIELD"\:"11111",
\"INT\-FIELD"\:1111,
\"DBL\-FIELD"\:11111,
\"PACKED\-FIELD"\:+11111,
\"PACKED\.-FIELD"\:+11111,
\"QUAD\-FIELD"\:11111,
\"ID\-FIELD"\:1,
\"LOGICAL\-FIELD"\:1111,
\"DBLLOG\-FIELD"\:11111,
\"ZONED\-FIELD"\:11111
}]
}
```

MySQL Access

Suprtool/Open now has the ability to read MySQL databases via the Open and Select commands. We are looking for alpha, beta testers and feedback specifically on how to treat certain data items. Please e-mail Neil Armstrong at neil@robelle.com to request a trial of this software.

Enhancements in Version 5.8

Input ($first/$last)

Suprtool now has $first and $last mnemonics, which can be used on a range selection of records on the input command. It was designed to handle a request to list the last N number of records in a file as in:

```
Input somefile($last-10/$last)
```

Suprtool will parse the Range selection and semantically check if the record range entered is logical. For instance, $first-2 and $last+10, which do not make logical sense would throw and error. Similarly if a record only has 5 records in it then $last-10 or $first+7, would also throw an error.

Enhancements in Version 5.7

Linkmgr

Suprlink now has a "suprmgr" file similar to Suprtool. Suprlink will process all the commands found in /opt/robelle/linkmgr or $ROBELLE/linkmgr.

Stexpmgr

STExport now has a "suprmgr" file similar to Suprtool. STExport will process all the commands found in /opt/robelle/stexpmgr or $ROBELLE/stexpmgr.

Set ComLog

Suprtool, Suprlink and STExport can now log all commands in each product. Every command will be logged if the setting Set ComLog, is on. Commands will not be logged if the user does not have write access to the comlog file. There are three separate log files, one for each product. The stlog file for Suprtool, sllog for Suprlink and sxlog for STExport. The files are by default located in /opt/robelle/log/suprtool directory. If the $ROBELLE variable is set, the log files would be in the directory $ROBELLE/log/suprtool.

The format of the log file is as follows:

DateTime, UID, PID, command entered

The Date and Time is 16 bytes in the format Year, Month, Day, Hours, Minutes, Seconds.

UID/PID are six digit numbers, followed by the command with a line feed at the end.

Please note that if this command is global the log files can expand quickly, taking up disc space by default in the /opt/robelle directory. You can redirect where the commands are logged by using linked files.

```
ln -s /log/robelle/suprtool/stlog /opt/robelle/log/suprtool/stlog
ln -s /log/robelle/suprtool/sxlog /opt/robelle/log/suprtool/sxlog
ln -s /log/robelle/suprtool/sllog /opt/robelle/log/suprtool/sllog
```

The user running any of the products must have write access to the log files whether Link files or directly.

$Proper

Suprtool now has the $proper function which will shift to upper case the first character of a byte type field and after any space or ampersand. It will also shift to lower case any other characters in the byte-field.

```
>in mprod
>list stan
>xeq
Apr 29, 2014 13:00                    File: mprod                  Page 1

PRODUCT-DESC

skil 3/8" variable speed drill
b&d router
skil var. sp. auto-scroll saw
skil 8 1/2" circular saw
b&d cordless screwdriver
makita 8 1/4" circular saw
b&d variable speed jigsaw
makita 1/2" router
makita 3/8" var. speed drill
skil router
b&d 7 1/4" circular saw
b&d 3/8" variable speed drill
makita 1" jigsaw
```

Considering the following data, you can fix all of the product names with one simple task:

```
>in mprod
>ext product-desc=$proper(product-desc)
>list stan
>xeq
May 01, 2014 11:40                    File: MPROD                  Page 1

PRODUCT-DESC

Skil 3/8" Variable Speed Drill
B&D Router
Skil Var. Sp. Auto-Scroll Saw
Skil 8 1/2" Circular Saw
B&D Cordless Screwdriver
Makita 8 1/4" Circular Saw
B&D Variable Speed Jig Saw
Makita 1/2" Router
Makita 3/8" Var. Speed Drill
Skil Router
B&D 7 1/4" Circular Saw
B&D 3/8" Variable Speed Drill
Makita 1" Jig Saw
```

Note that any character after a space, "&", or "-" is upshifted for a proper name. Suprtool will also downshift those characters that do not qualify as needing proper capitalization and it is a capital character, the proper function will downshift those characters. See an example below:

```
>IN NAME
>LIST
>XEQ
>IN NAME.NEIL.GREEN (0) >OUT $NULL (0)
NAME            = NEIL ARMSTRONG

>IN NAME
>EXT NAME=$PROPER(NAME)
>LIST
>XEQ
>IN NAME.NEIL.GREEN (0) >OUT $NULL (0)
NAME            = Neil Armstrong
```

The $proper function only works on byte type fields.

$Translate

Suprtool now has a $translate function which in conjunction with the translate command allows you to build a translation table, whereby you can translate from any byte character to any character. We have also added a method to a supplied translate table which will allow you to obscure the data such that it can't be read.

```
>in newprod
>list
>xeq
>IN NEWPROD.NEIL.GREEN (0) >OUT $NULL (0)
PRODUCT-DESC    = Skil 3/8" Variable Speed Drill

>IN NEWPROD.NEIL.GREEN (1) >OUT $NULL (1)
PRODUCT-DESC    = B&D Router

>in newprod
>translate tounread
>ext product-desc=$translate(product-desc)
>out unread,link
>xeq
IN=13, OUT=13. CPU-Sec=1. Wall-Sec=1.

>in unread
>num 1
>list
>xeq
>IN UNREAD.NEIL.GREEN (0) >OUT $NULL (0)
PRODUCT-DESC    = Hzxo .2)? Epcxpqot Hatts Rcxoo
Warning:  NUMRECS exceeded; some records not processed.
IN=2, OUT=1. CPU-Sec=1. Wall-Sec=1.

>in unread
>translate toread
>ext product-desc=$translate(product-desc)
>list
>xeq
>IN UNREAD.NEIL.GREEN (0) >OUT $NULL (0)
PRODUCT-DESC    = Skil 3/8" Variable Speed Drill

>IN UNREAD.NEIL.GREEN (1) >OUT $NULL (1)
PRODUCT-DESC    = B&D Router
```

You can make your own Translate table using the Translate command, where you can specify the character you want to translate and what you want to translate to, using Decimal Notation. So if you want to translate "A" to "Z", you would type the command:

```
Translate "^65:^90"
```

So you specify the from character on the left in decimal which is the capital-A and the to-character is also in decimal format which is capital-Z, which is decimal 90. If you want to reverse the translation you can simply do the following command and translate the field back with:

```
Translate "^90:^65"
```

This is not meant to be an encryption solution, but it will help obfuscate test date really quickly.

Enhancements in Version 5.6

Extract Command

Suprtool's extract command now has three new keywords, which can be used for extract range feature. You can now say extract $all, extract $first / $last on an SD file or Image/Eloquence dataset. The intention is to make your scripts more easily maintained. If you had a script that you wanted to put a sequence number at the beginning and then extract the rest of the dataset you previously had to specify the starting field and the ending field. For example if the first field in a dataset was order-no and the last field was pst-code you may have a script that looked like this.

```
base orddb
get customers
def seq-no,1,4,double
ext seq-no=$counter
ext order-no / pst-code
out newfile,link
xeq
```

If you added any fields to the beginning or end of the dataset you would have to re-write the script. Now you can write the script as being:

```
base orddb
get customers
def seq-no,1,4,double
ext seq-no=$counter
ext $all
out newfile,link
xeq
```

You can also write the script using $first / $last as your preference, but $first and $last are also useful if you need to add data into the middle of the fields you extract:

```
base orddb
get customers
def seq-no,1,4,double
ext $first / zip
ext seq-no=$counter
ext tax-code / $last
out newfile,link
xeq
```

Please note that if a self-describing file has a fieldname that is a duplicate field and one of the duplicate fields is the last field in the file, then $first / $last and $all, will only extract up to the first occurrence of the duplicate fieldname. This may seem as an issue but it is consistent with what Suprtool does currently with extract from a range. Currently and prior to the $first / $last enhancement Suprtool would have extracted only up to the first occurrence of the field if you had a file such as this:

```
>form
File: newfile      (SD Version B.00.00)  Has linefeeds
        Entry:                   Offset
            CHAR-FIELD          X5      1
            INT-FIELD           I1      6
            DBL-FIELD           I2      8
            PACKED-FIELD        P12    12
            PACKED*-FIELD       P12    18
            QUAD-FIELD          I4     24
            ID-FIELD            I1     32
            LOGICAL-FIELD       K1     34
            DBLLOG-FIELD        K2     36
            ZONED-FIELD         Z5     40
            FILLER              X36    45
            FILLER              X36    81
     Entry Length: 116  Blocking: 1
```

Notice that FILLER is a duplicate named field, so if you entered, extract char-field /
filler, Suprtool would only extract up to and including the first FILLER field. For
consistency, extract $first / $last behaves the same way.

Data Items Support

Suprtool now supports 512 data items in both Eloquence datasets and SD files and
Oracle fields. STExport and Suprlink have also been updated to support the
increased number of data items in the SD fields that they read, but please note that
other limitations still exist. [5.5.10]

$SubCount

$SubCount has been added to provide a counter that only gets reset at a given sort
break.

```
In file1sd
Sort char-field
Def control-count,1,4,double
Ext $first / $last
Ext control-count=$subcount(char-field)
Out newfile,link
xeq
```

What Suprtool will do in this case is start incrementing a number starting with 1, and
increase by 1 for any given char-field value. This way you can add a counter based
on a sort break for a given field.

Bugs Fixed

Bugs Fixed In Suprtool 5.9

Add Command. The Add command would fail if a Table in an Oracle database accessible by a given username had more than 2.1 billion entries on Oracle 11 and higher.

Bugs Fixed In Suprtool 5.6.11

Output , Display Command. Suprtool for Itanium and Suprtool/Open would not properly convert negative, single and double integers when output/,display is invoked

Quad Integer Input Parsing. Suprtool/Open would not parse a negative number into a quad integer container.

Bugs Fixed In Suprtool 5.6.10

If Command. Suprtool for Itanium and Suprtool/Open incorrectly reallocated if/extract code space on subsequent tasks, which would eventually cause Suprtool to fail with the error, "Unable to allocate heap space."

Bugs Fixed In Suprtool 5.6

Arithmetic Expressions. Suprtool/Open had problems with arithmetic expressions with brackets for Packed-Decimal and Zoned-Decimal data types.

Arithmetic Expressions. Suprtool/Open had problems with rounding for arithmetic expressions with Quad Integers, Packed-Decimal and Zoned-Decimal data types.