



What's New At Robelle



Jan 2017



Qedit Server 6.2 Security Fix

We have made an important Security fix to the login portion of the Qedit Server.

The Qedit server would not properly honour an administrative lock on a user account, during login from the Qedit for Windows client.

The Qedit server now confirms if a user account is under administrative lock or not. The server now checks for this and honours these locks and will stop the user from logging in as appropriate. No change is required in the client.

Check out what's else new with Qedit & Qedit Server 6.2
<http://www.robelle.com/products/whatsnew.html#Qedit>

Suprtool \$Month Function

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 yyyyymmdd.

For Example:

```
In somefile
Item mydate,date,ccyymmdd
Def targetdate,l,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.

Check out what's else new with Suprtool 5.9

<http://www.robelle.com/products/whatsnew.html#Suprtool>

STExports Json Output

While working on some new iPhone test applications, I needed some JSON data to play with that I was familiar with. I had played with some simple JSON data but it wasn't enough, so I wrote the JSON command to format larger amounts of data to play with on my iPhone.

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 filelsd
$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":"IIIII","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":\ "IIIII",  
    \ "INT-FIELD"\ :1111,  
    \ "DBL\FIELD"\ :11111,  
    \ "PACKED\FIELD"\ :+1111,  
    \ "PACKED\FIELD"\ :+11111,  
    \ "QUAD\FIELD"\ :11111,  
    \ "ID\FIELD"\ :!,  
    \ "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\":"||||",
  \INT\FIELD\:.|.|.|,
  \DBL\FIELD\:.|.|.|.|,
  \PACKED\FIELD\:.+|.|.|.|,
  \PACKED\.-FIELD\:.+|.|.|.|,
  \QUAD\FIELD\:.|.|.|.|,
  \ID\FIELD\:.|,
  \LOGICAL\FIELD\:.|.|.|,
  \DBLLOG\FIELD\:.|.|.|.|,
  \ZONED\FIELD\:.|.|.|}|]
```

Multiple Json Commands

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

```
$in file|sd
$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\":"||||",
    \INT\FIELD\:.|.|.|,
    \DBL\FIELD\:.|.|.|.|,
    \PACKED\FIELD\:.+|.|.|.|,
    \PACKED\.-FIELD\:.+|.|.|.|,
    \QUAD\FIELD\:.|.|.|.|,
    \ID\FIELD\:.|,
    \LOGICAL\FIELD\:.|.|.|,
    \DBLLOG\FIELD\:.|.|.|.|,
    \ZONED\FIELD\:.|.|.|}]}]
```

Glueing Lines

Occasionally we get requests to summarize data or manipulate reports to be re-organized. One common manipulation is to stick the contents of one line onto the end of another based on the a value in the first line. The commands to do this are fairly simple:

```
Append "~" "Transaction"  
Glue "~"  
ch "~" "@"
```

Here is the data:

```
qux/lq all  
Transaction Date: 20160908  
Balance: 1245.85  
Transaction Date: 20151213  
Balance: 908.54  
Transaction Date: 20121221  
Balance 125.44
```

Here is the code in action:

```
qux/append "~" "Transaction"  
1 Transaction Date: 20160908~  
3 Transaction Date: 20151213~  
5 Transaction Date: 20121221~  
3 lines appended to  
qux/glue "~"  
1 Transaction Date: 20160908~Balance: 1245.85  
3 Transaction Date: 20151213~Balance: 908.54  
5 Transaction Date: 20121221~Balance 125.44  
3 lines Glue-updated  
qux/ch "~" "@"  
1 Transaction Date: 20160908 Balance: 1245.85  
3 Transaction Date: 20151213 Balance: 908.54  
5 Transaction Date: 20121221 Balance 125.44  
3 lines changed
```

The Append command sticks a Tilde on the end of the line, and the glue command, glues the line after the Transaction line onto the end. Tilde is used since it isn't often in normal files, you can look to insure that the tilde is not in the file with a simple: ch "~"@"@, to see if there are any occurrences.

Interesting Hewlett-Packard article

Ole Nord former Robelle dealer sent us an interesting article; Hewlett-Packard's management history offers cautionary tales for today's leading tech companies

<https://www.technologyreview.com/s/602961/four-lessons-for-silicon-valley-from-its-first-startup/>

Robelle Book Club

We Robellians like to read. In our last newsletter, we listed a few of the books we have shared and enjoyed.

We have listed a few more of our recent "good reads" below.

[Medicine Walk](#) by [Richard Wagamese](#)

[The Lake House](#) by [Kate Morton](#)

[Precious Cargo: My Year of Driving the Kids on School Bus 3077](#) by [Craig Davidson](#)
(nice light read)

[A Robot in the Garden](#) by [Deborah Install](#) (strange & quirky)

[The Lewis Man \(The Lewis Trilogy, #2\)](#) by [Peter May](#)

[The Gilded Hour](#) by [Sara Donati](#)

Jody recommended [The Smartest Guys in the Room: The Amazing Rise and Scandalous Fall of Enron](#) by [Bethany McLean](#), [Peter Elkind](#)

We are always looking for book recommendations. Send your book recommendation to Fran Glasgow fran@robelle.com

To unsubscribe from our newsletter, send us an e-mail at wudunsub@robelle.com

*May the New Year bring fresh bout of excellence
in your life*