



# What's New At Robelle



Daily Support Column  
Suprtool 5.9.10  
\$INRECNUM  
\$RESPACE

April 2017

## Daily Support Column

In early 2017, we started writing articles about some of the daily support calls, e-mails and issues that come up on regularly. Please note the following most popular Issues referenced.

### Defining or Accessing a Portion of Data

A customer asked recently how to test a portion of a field, for certain values, specifically a date field. This is relatively easy regardless of the type of the field, whether byte or numeric.

The specifics of the test the customer wanted was to determine the DD portion of a ccyyymmdd date field.

### Some Test Data

```

>in dates
>form
  File: dates      (SD version B.00.00)  Has linefeeds
  Entry:
    DUE-DATE-X      X8      1      <>
    DUE-DATE-NUM    I2      9      <>
  Entry Length: 12  Blocking: 1
>list
>xeg
>IN dates (0) >OUT $NULL (0)
DUE-DATE-X      = 20170131      DUE-DATE-NUM    = 20170131

>IN dates (1) >OUT $NULL (1)
DUE-DATE-X      = 20170131      DUE-DATE-NUM    = 20170131

```

## Byte Field Test

There are two methods for determining a portion of any field, the following isolates the DD portion of a date field:

### Directly

While this syntax is no longer documented, you may see this in an old job:

```
if due-date-x(1,7,2)="31"
```

The 1 is the first instance of a compound field, starting in the 7th byte for 2 bytes:

### Define Command

The syntax below allows you to document what it is that you are testing:

```
def dd,due-date-x[7],2,byte  
if dd="31"
```

By giving the field you want a test a more meaningful name it simply allows the script to be self-documenting.

### Numeric Field Test

Since numeric fields are a series of bits you can't easily define a portion of the data in a meaningful way, but you can use some simple math tricks to get at a portion of the data:

```
if $truncate(due-date-num mod 100) = 31
```

So the above statement simply takes the date, 20160131 and what mod does is take the remainder of a divide. So Suprtool, divides the date by 100, and takes the remainder, which in this case would be the last two digits of the date field, and then truncates that to a whole number, just in case of a rounding issue.

You can even look at the month with more math tricks:

```
if $truncate(due-date-num / 100 mod 100) = 01
```

The above, takes the date of 20160131, divides by 100, which becomes 201601 and then the mod 100 ends up with the MM portion, by taking the remainder of a divide by 100, which in this case is the 01.

## Using String Addition to Build a Link Key

In Suprlink there is a limit of two keys for a link operation, however, with some simple manipulation you can make to make a key out of a number of fields. The following takes some numeric fields and turns it in to a simple byte key with a length of 18 bytes. The technique is to re-define the space that the integer field occupies, as a Byte field, then you can use the String Addition feature to build up the key out of multiple fields.

```
input tmpnsf1  
{ Define Fields as Byte, with length of Field }
```

```

def x-acct,account,4,byte
def x-suffix,suffix,2,byte
def x-date,entry-date,4,byte
def x-sequence,sequence,4,byte
def x-beg-bal,beg-bal,4,byte
def targkey,1,18
ext account
ext suffix
ext entry-date
ext amount-sh
ext balance-sh
ext description
ext beg-bal
ext sequence
{ Use String Addition to build the key }
ext targkey = x-acct + x-suffix + x-date + x-beg-bal + x-sequence
out tmpnsf11,link
xeq
exit

```

In this case, we've made a key made up five fields, in one step.

## Exporting and Using a Variable

E-mail from a customer:

**You can call an environment variable from Suprtool on HP-UX? Trying to recall how to get the Hour from the Date Command and create a variable to use against the hour in the record. Any thoughts? It's an option on the execution of suprtool?**

How do you get the hour on HP-UX date command:

```

date "+%H"
06

```

Here is how you Put in a variable:

```

export sthour=`date "+%H"`
echo $sthour
06

```

Use in Suprtool:

```

suprtool
SUPRTOOL/UXia/Copyright Robelle Solutions Technology Inc. 1981-2017.
(Version 5.9.10 Internal) Fri, Mar 10, 2017, 6:59 AM
Build 2
{first set varsub on and you can reference the variable}
>set varsub on
>in x
>if newdate=$sthour

```

{Note the Set VarSubDebug shows the resolved variable}

```
>reset if
>set varsubdebug on
>if newdate=$sthour
vd:if newdate=06
>ea
vd:ea
```

Questions please contact Neil Armstrong [neil@robelle.com](mailto:neil@robelle.com)

---

## Suprtool 5.9.10

Suprtool version 5.9.10 was release in the first couple of days in March and is available as a pre-release. This version has six new functions, please see two new key functions that are directly from customer requests as always.

The first function exposes the Input Record Number of whatever input source is involved. The primary reason for this enhancement was to provide precise control over finding specific records. The specific instance of this was a process incorrectly added some duplicate records. Adding \$inrecnum provides the only basis of isolating the records, as in:

```
If key-field=12345 and $inrecnum >= 10000
```

Request your copy of 5.9.10 contact Tammy [tammy@robelle.com](mailto:tammy@robelle.com) Technical questions please contact Neil Armstrong [neil@robelle.com](mailto:neil@robelle.com)

---

## \$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.

The next new function is called \$respace, which essentially takes multiple spaces down to one in any given byte field. The primary reason for this was to fixup poorly formatted addresses, company names etc.

Questions please contact Neil Armstrong [neil@robelle.com](mailto:neil@robelle.com)

---

## \$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 *
>xeg
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
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 *
>xeg
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.
```

Questions please contact Neil Armstrong [neil@robelle.com](mailto:neil@robelle.com)

To unsubscribe from our newsletter, send us an e-mail at [wudunsub@robelle.com](mailto:wudunsub@robelle.com)