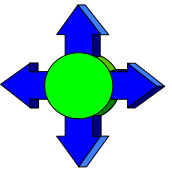


Inside Module 9

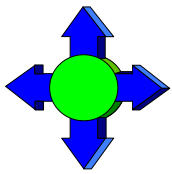
Latest Features in Suprtool

- Variable Substitution
- \$Counter
- \$Clean and \$FindClean
- \$Total and \$Subtotal
- \$SPLIT
- \$EDIT
- \$NUMBER
- Suprlink's Join Command
- Output,else
- FastRead (non-MPE Platforms)
- Dbedit (non-MPE Platforms)
- List Command (non-MPE Platforms)
- Dynamic Loading of Eloquence (non-MPE Platforms)



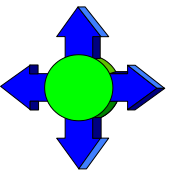
Development is a continuous process

- We are always working on new features
- New version every month or so



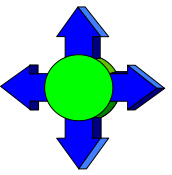
Variable Substitution

- MPE version has had Variable Substitution for two years
 - Internal MPE routine
 - Same as used by the CI
- HP-UX now has the same feature
 - Suprtool functions take precedence
 - \$stddate, \$total, \$date, \$today are not replaced
 - Variable must be set and exported prior to running program
 - Suprtool command line, \$read function
 - STExport and Suprlink as well
 - Must have Set Varsub On



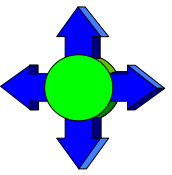
\$counter function

- Sequential number function
- Allows you to retain original output order
 - > get morder
 - > def mycount,1,4,double
 - > ext mycount=\$counter
 - > ext orderno
 - > out myfile,link
 - > xeq



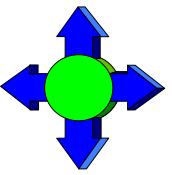
Clean my data

- \$clean function in Suprtool
 - clean “^9”
 - Set cleanchar “ “
 - update
 - extract mybytefield=\$clean(mybytefield)



Clean Example

```
>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
```



\$FindClean Example

- Users did not want to blindly \$clean all records
- Some wanted to track strange characters and corruptions
- \$FindClean will find fields that could be cleaned

>base membrs

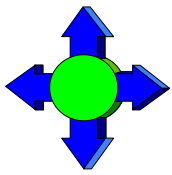
>get member-file

>Clean special

>If \$findclean(name)

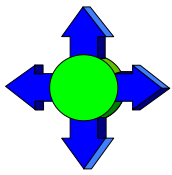
>output toclean,link

>xex



\$Total

- \$Total and \$subtotal functions
- Provide running grand total and subtotals
- Data is stored in a Packed field



\$Subtotal

- Syntax for the \$subtotal function in the extract command is:

- `extract target = $subtotal(field, sort-field)`

- Must specify a sort

- Sort fields must match

```
>def mytot,1,14,packed
```

```
>get orders
```

```
>sort ordnum
```

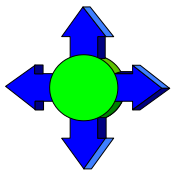
```
>ext ordnum
```

```
>ext part-number,description,sales-amount
```

```
>ext mytot=$subtotal(sales-amount,ordnum)
```

```
>out sales,link
```

```
>xeq
```



\$Split

- \$split function
 - Extracts out variable length strings from data
 - Extract from beginning to “/” character
- ```
>extract first-name=$split(name,first,"/")
>extract last-name=$split(name,"/",last)
```
- Can also split on multiple occurrences of a certain character

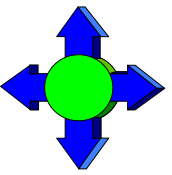
Consider the following data:

```
Armstrong/ Neil/ Patrick
```

```
Green/ Bob/ Miller
```

```
Edwards/ Janine/
```

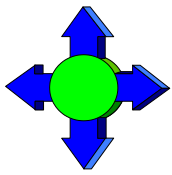
```
Armstrong/Arthur/Derek
```



## *\$split details*

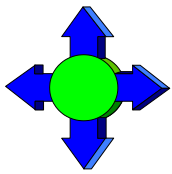
---

- Occurrence is honored
- No need to specify an occurrence of one
- Can nest inside a \$trim or similar string function
- \$split does check for overflow
- First and Last keywords available



# \$edit

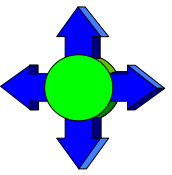
- \$edit function
- Converts from numeric or byte to formatted string of bytes
- Uses syntax and rules similar to Cobol Edit-Masks
- Placeholders and Format characters
- Two sets of rules byte type and numeric based on source data type
- `>ext formatdate=$edit(a, "xxxx/xx/xx")`
- `FORMATDATE = 2003/09/24`



# Numeric source \$edit masks

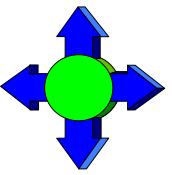
- Numeric source \$edit masks

```
>ext a=$edit(int-field,"$$,$$$.$99-")
>ext b=$edit(int-field,"99,999.$99-")
>ext c=$edit(int-field,"cr99999.$99")
>ext d=$edit(int-field,"-$9999.$99")
>ext e=$edit(int-field,"**,$**.$99+")
>ext f=$edit(int-field,"zz,$zz.$99+")
>list
>xeq
>IN FILE1SD.NEIL.GREEN (0) >OUT $NULL (0)
A = $11.11- B = 00,011.11-
C = CR00011.11 D = -$0011.11
E = ****11.11- F = 11.11
```



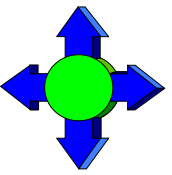
## *Handling the sign*

- +, -, CR and DB allowed
- Depends on state of the data
- Positive, negative, neutral



## *Rules for \$ sign*

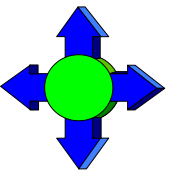
- Fixed \$ sign edit
- Floating \$ edit invoked if two “\$\$” appear in the edit mask
- Suprtool attempts to fixup most odd cases
  - \$,123.45 becomes \$123.45



## *Decimal places*

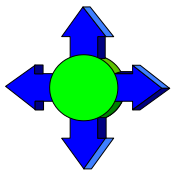
- Data is adjusted to number of decimal places
- Default Decimal symbol is “.”
- Can be changed to “,” or any other single character
- Source field decimal value is honored





## *Currency symbol and overflow*

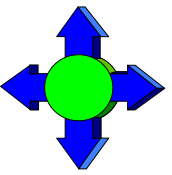
- Currency symbol can be up to four characters
- Set currencysymbol "\$"
- Suprtool by default will not stop if overflow occurs
- Set editstoperror on



# *\$number*

- \$number function
- Converts free-form numbers to numeric in one step
- Honors signs, decimal places and currency symbols
- This means numbers in New-Price can be read by \$number:

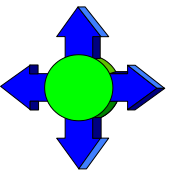
| Item-number | New-Price |
|-------------|-----------|
| 12345       | +123.45   |
| 34563       | + 27.5    |
| 21312       | + 1.545   |



## *\$number details*

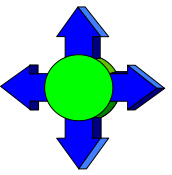
---

- Rounding and decimal places
- Error conditions
- Currency, Decimals and thousand symbols



# *Suprlink*

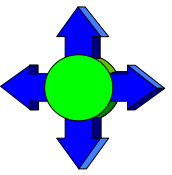
- Combines two files by common key
- Link command allows for many to one relationship
- Join command allows for many to many relationships
- SQL- like feature
- Inner Join
- Outer Join



## *SQL continued*

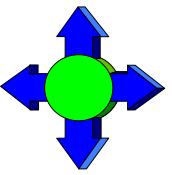
---

- Left Outer Join
- Right Outer Join
- Simple Join task



## *Join continued*

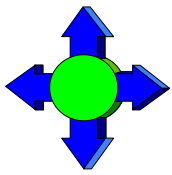
- Only one Join per task
- Can specify a secondary key to join on



## *Output,else*

---

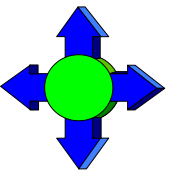
- One Process Output two files
- If condition
- Output,else = if NOT condition
- MPE Temp file
- Other Platforms filename.else



## *Set FastRead On*

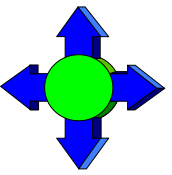
- MPE and MR Nobuf
- Set FastRead uses Eloquence Block read routines
- Two to Five times faster
- Less CPU and Wall Time





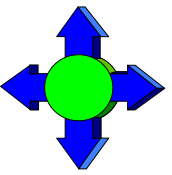
# *Dynamic Loading*

- Dynamically loads whatever version of Eloquence library you have
- Shlib\_path
- Most applications providers set this for you
- Enhancement for the future



# ***Dbedit***

- Edit Single Records
- Popular in MPE version for editing/fixing single records
- Now will work with Eloquence databases on HP-UX



# *Future*

- Reporting?
- More work in a single pass