

Processing non-APL Data Files (Part 2)

Richard Smith
Michael Baas



First: thank you!

- This would not have been possible without prior heavy-lifting done by
 - Morten Kromberg
 - Pierre Gilbert
 - Richard Procter



Health-Warning

- Work-shop
- „XL“ is a work-in-progress
- Parameters may be subject to changes
- Performance will change
- You may loose hair ;-)



Preparations

- Please install sfExcel and XL.dyalog
(links on Richards Homepage <http://broadmeadow.eu/Dyalog/Workshop.html>)
- Download samples from the same source



So, what do we have?

```
]load xl
#.XL
#.XL.Load'Book1.xlsx'
```

Hdr 1	Hdr 2	Hdr 3
Text	1.2	3.4
Text	5.6	7.8
[Null]	[Null]	11.2

```
MyFile← 'Book1.xlsx'
┌-data←1 1↓#.XL.Load MyFile
```

```
1.2 3.4
```

```
5.6 7.8
```

```
[Null] 11.2
```

```
1+data
```

```
DOMAIN ERROR
```

```
1+data
```

```
∧
```



Controlling import through α

```
↳ data ← ( 'GetAs' 'Number' ) #.XL.Load MyFile
```

Hdr 1	Hdr 2	Hdr 3
Text	1, 2	3, 4
Text	5, 6	7, 8
0	0	11, 2



Even more α

```
1+1 1↓('GetAs' 'Number')('TitleRows'  
1)#.XL.Load MyFile
```

```
2.2 4.4
```

```
6.6 8.8
```

```
1 12.2
```



Ok, I cheated. (With the 1 1↓)

```
opts←('GetAs' 'Number')('TitleRows' 1)
1+data←opts #.XL.Load MyFile
DOMAIN ERROR
    1+data←opts #.XL.Load MyFile
```

^
data

Hdr 1	Hdr 2	Hdr 3
Text	1.2	3.4
Text	5.6	7.8
0	0	11.2



We need a new option?!

```
opts←('GetAs' 'StrictNumber')('TitleRows' 1)
```

```
1+data←opts #.XL.Load MyFile
```

DOMAIN ERROR

```
1+data←opts #.XL.Load MyFile
```

^



Finally!

data

Hdr 1	Hdr 2	Hdr 3
0	1.2	3.4
0	5.6	7.8
0	0	11.2

1+1↓[1]data

1 2.2 4.4

1 6.6 8.8

1 1 12.2



Datatable (1)

```
MyFile←opendata_donations1,4m.xlsx‘  
opts←( 'GetAs' 'DatatableObject' )  
x←opts #.XL.Load MyFile  
x.Type
```

```
NetClient
```

```
x.GetType
```

```
System.Data.DataTable
```



Datatables into APL

```
a5←('GetAs' 'Datatableobject')#.XL. 'opendata_donations-50k.xlsx'  
]runtime "d1←2011 ⊍ a5" "d2←↑(⊍a5.Rows).ItemArray"
```

* Benchmarking "d1←2011 ⊍ a5"
(ms)

CPU (avg): 313

Elapsed: 343

* Benchmarking "d2←↑(⊍a5.Rows).ItemArray"
(ms)

CPU (avg): 377781

Elapsed: 501177



Datatable – more info

- <https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/dataset-datatable-dataview/datatables>
- <https://www.codeproject.com/Articles/6178/A-Practical-Guide-to-NET-DataTables-DataSets-and-D>
- <https://www.dotnetperls.com/datatable>



The sfExcel-Class

```
a5←('GetAs' 'sfexcel')#.XL.Load'opendata_donations999.xlsx'
```

```
a5.UsedRange
```

```
1 1 999 22
```

```
a5.(ConvertRCtoA1 UsedRange)
```

```
A1:V999
```

```
1 1 1 22 a5.SetFontBold 1
```

```
a5.Show
```

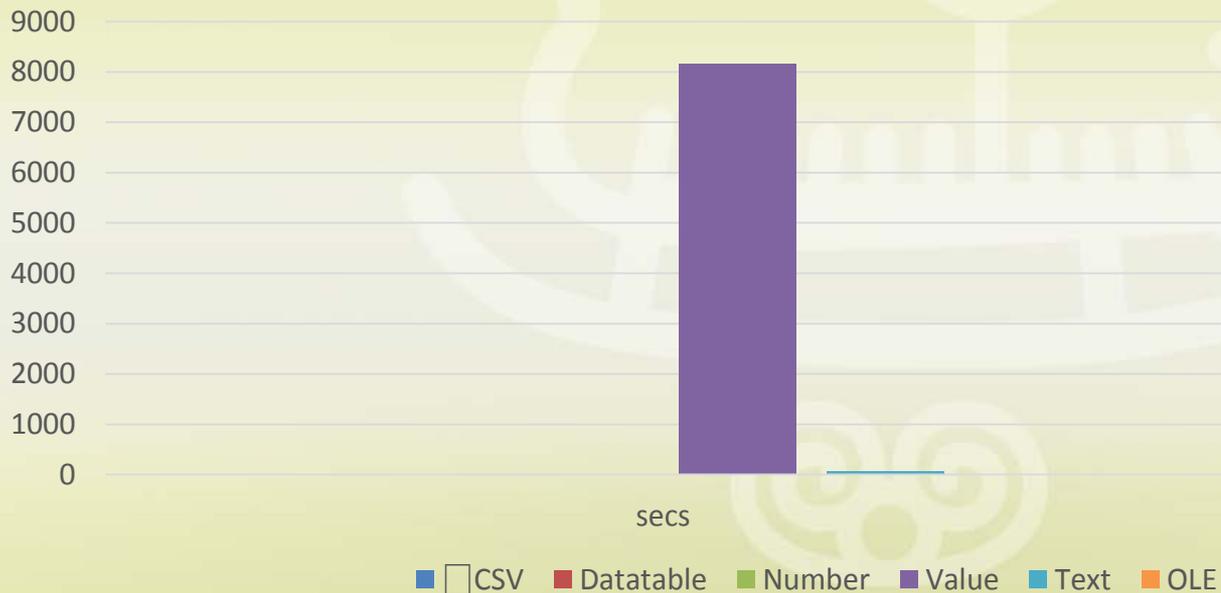
```
a5.Print 1
```

```
http://aplwiki.com/sfExcel
```



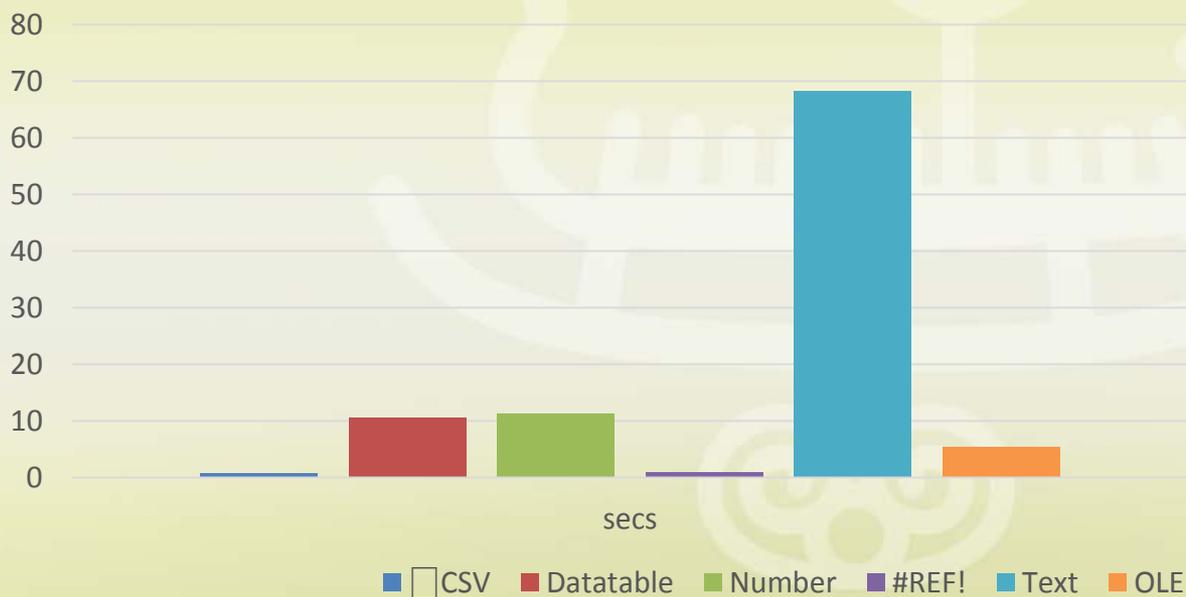
Performance

Reading a 50,000-row table



Performance

Reading a 50,000-row table



CSV	0.680
Datatable	10.547
Number	11.399
Value	8,155.002
Text	68.300
OLE	5.468



Save

- `{larg}XL.Save Data Filename`
[Sheetname or index]

Data = APL-Matrix

larg contains NameValue-Pairs:

UseOle: 0/1

ReadFile: charvec with filename (read, fill range, save as Filename)

Filename includes extension as one of „xls[x]“, „pdf“, „htm[l]“



Problems

- Importing .CSV should be transparent
 - Often fails. This is under investigation and will be fixed asap.
- Performance



Pierre Gilbert

- <http://aplwiki.com/netXML>: Dyalog has made a fantastic job with quadXML but the users are left alone to make all the others cover functions required around that quad. So I have put together on that Wiki what was important for me.
- <http://aplwiki.com/netCSV>: Those are my cover functions when using CSV files written before quadCSV
- <http://aplwiki.com/sfZip>: Those are my cover functions for zipping and unzipping files using Syncfusion dlls.



Library/Files

- `text←ReadText filename`
- `text AppendText filename`
- `text AppendTextU filename`
- `(tieno name)←{dcf>CreateTemp pattern`
- `r←filename Fopen tieno`



Spontaneous, non-representative polls

- Who would like inverted tables?
- The targets spreadsheets you need to import?

