TH BINGEN
University of Applied Sciences

Interface to Excel via COM

Accessing Excel from APL2

Dieter Kilsch

eh. Technische Hochschule Bingen

APL Germany,
Berlin-Adlershof, 6. Mai 2022

What I need(ed)

Aims of the interfaces

1. Get Data from an Excel file (work book),
2. Write Data to an Excel file (work book),
3. Set properties of cells and (work) sheets.

What I need(ed)

Aims of the interfaces

1. Get Data from an Excel file (work book),
2. Write Data to an Excel file (work book),
3. Set properties of cells and (work) sheets.

What I need(ed)

Aims of the interfaces

1. Get Data from an Excel file (work book),
2. Write Data to an Excel file (work book),
3. Set properties of cells and (work) sheets.

What I need(ed)

Aims of the interfaces

1. Get Data from an Excel file (work book),
2. Write Data to an Excel file (work book),
3. Set properties of cells and (work) sheets.

Sources

1. APL2 User's Guide
2. VBA literature
3. https://docs.microsoft.com/en-us/office/vba/api/overview/excel
4. (MS-)internet search e.g. „Excel colour change VBA“

What I need(ed)

Aims of the interfaces

1. Get Data from an Excel file (work book),
2. Write Data to an Excel file (work book),
3. Set properties of cells and (work) sheets.

Sources

1. APL2 User's Guide
2. VBA literature
3. https://docs.microsoft.com/en-us/office/vba/api/overview/excel
4. (MS-)internet search e.g. „Excel colour change VBA“

What I need(ed)

Aims of the interfaces

1. Get Data from an Excel file (work book),
2. Write Data to an Excel file (work book),
3. Set properties of cells and (work) sheets.

Sources

1. APL2 User's Guide
2. VBA literature
3. https://docs.microsoft.com/en-us/office/vba/api/overview/excel
4. (MS-)internet search e.g. „Excel colour change VBA“

What I need(ed)

Aims of the interfaces

1. Get Data from an Excel file (work book),
2. Write Data to an Excel file (work book),
3. Set properties of cells and (work) sheets.

Sources

1. APL2 User's Guide
2. VBA literature
3. https://docs.microsoft.com/en-us/office/vba/api/overview/excel
4. (MS-)internet search e.g. „Excel colour change VBA“

Calling the routine

```
r←s XLSles dn

1 s[1] 0: read data
       1: connect to Excel, file and sheet, read
       2: read, disconnect sheet, file and Excel
       3: 1 and 2                                    [Def.: 3]
   [2] 1: read given sheet                           [Def.: 1]
       2: read previous sheet
       3: return number of sheets
       4: return names of sheets

2 dn[1] TV   file name (work book)
    [2] TV   name of (work) sheet to be read
             empty: active (work) sheet          [DEF.: '']
         S   number of (work) sheet to be read
    [3] V[2] left upper cell
         TV  '': read all                        [DEF.: '']
    [4] V[2] size of cell array
```

Calling the routine

```
r←s XLSles dn

1 s[1] 0: read data
       1: connect to Excel, file and sheet, read
       2: read, disconnect sheet, file and Excel
       3: 1 and 2                                    [Def.: 3]
   [2] 1: read given sheet                           [Def.: 1]
       2: read previous sheet
       3: return number of sheets
       4: return names of sheets

2 dn[1] TV   file name (work book)
    [2] TV   name of (work) sheet to be read
             empty: active (work) sheet          [DEF.: '']
         S   number of (work) sheet to be read
    [3] V[2] left upper cell
         TV  '': read all                        [DEF.: '']
    [4] V[2] size of cell array
```

The steps inside the routine

```
r←s XLSles dn
```

Connecting, only if `s[1]∈{1,3}`: Connect to COM

```
1 →(1≠r←3 11 ⎕NA 'COM')/F10
2
3
4
5
6
7
8
```

The steps inside the routine

```
r←s XLSles dn
```

Connecting, only if `s[1]∈{1,3}`: Connect to Excel

1 `→(1≠r←3 11 ⎕NA 'COM')/F10`

2 `→(~0 0≡↑(r em hex)←1 COM 'CREATE' 'Excel.Application')/F11`

`hex`: handle to Excel.

3

4

5

6

7

8

The steps inside the routine

```
r←s XLSles dn
```

Connecting, only if `s[1]∈{1,3}`: Connect to file

1 `→(1≠r←3 11 ⎕NA 'COM')/F10`

2 `→(~0 0≡↑(r em hex)←1 COM 'CREATE' 'Excel.Application')/F11`

3 `→(~0 0≡↑(r em hwb)←1 COM 'METHOD' hex 'Workbooks.Open'(1⊃dn))/F13`

`hwb`: handle to file.

4

5

6

7

8

The steps inside the routine

```
r←s XLSles dn
```

Connecting, only if `s[1]∈{1,3}`: Release old sheet, if handle exists

```
1 →(1≠r←3 11 ⎕NA 'COM')/F10
2 →(~0 0≡↑(r em hex)←1 COM 'CREATE' 'Excel.Application')/F11
3 →(~0 0≡↑(r em hwb)←1 COM 'METHOD' hex 'Workbooks.Open'(1⊃dn))/F13
4 →(2≠⎕NC 'hws')/1+⎕LC◇→(~0 0≡↑(r em res)←1 COM 'RELEASE' hws)/F41
5
6
7
8
```

The steps inside the routine

r←s XLSles dn

Connecting, only if s[1]∈{1,3}: Get number of work sheets

1 →(1≠r←3 11 ⎕NA 'COM')/F10

2 →(~0 0≡↑(r em hex)←1 COM 'CREATE' 'Excel.Application')/F11

3 →(~0 0≡↑(r em hwb)←1 COM 'METHOD' hex 'Workbooks.Open'(1⊃dn))/F13

4 →(2≠⎕NC 'hws')/1+⎕LC◇→(~0 0≡↑(r em res)←1 COM 'RELEASE' hws)/F41

5 →(0 0≡↑(r em ns)←1 COM 'PROPERTY' hwb 'WORKSHEETS.COUNT')/F19

ns: number of work sheets.

6

7

8

The steps inside the routine

r←s XLSles dn

Connecting, only if s[1]∈{1,3}: Cases to connect to a sheet

1 →(1≠r←3 11 ⎕NA 'COM')/F10

2 →(~0 0≡↑(r em hex)←1 COM 'CREATE' 'Excel.Application')/F11

3 →(~0 0≡↑(r em hwb)←1 COM 'METHOD' hex 'Workbooks.Open'(1⊃dn))/F13

4 →(2≠⎕NC 'hws')/1+⎕LC◇→(~0 0≡↑(r em res)←1 COM 'RELEASE' hws)/F41

5 →(0 0≡↑(r em ns)←1 COM 'PROPERTY' hwb 'WORKSHEETS.COUNT')/F19

6 cs←↑((0=↑0⍴2⊃dn),(0 1=0=⍴2⊃dn))/⍳3 ⍝ number, name, Active sheet
cs: way of addressing sheet.

7

8

The steps inside the routine

```
r←s XLSles dn
```

Connecting, only if `s[1]∈{1,3}`: Number of sheet correct ?

```
1 →(1≠r←3 11 ⎕NA 'COM')/F10
2 →(~0 0≡↑(r em hex)←1 COM 'CREATE' 'Excel.Application')/F11
3 →(~0 0≡↑(r em hwb)←1 COM 'METHOD' hex 'Workbooks.Open'(1⊃dn))/F13
4 →(2≠⎕NC 'hws')/1+⎕LC⋄→(~0 0≡↑(r em res)←1 COM 'RELEASE' hws)/F41
5 →(0 0≡↑(r em ns)←1 COM 'PROPERTY' hwb 'WORKSHEETS.COUNT')/F19
6 cs←↑((0=↑0⍴2⊃dn),(0 1=0=⍴2⊃dn))/⍳3 ⍝ number, name, Active sheet
7 →(cs>1)/1+⎕LC⋄→(ns<2⊃dn)/F22
8
```

The steps inside the routine

```
r←s XLSles dn
```

Connecting, only if `s[1]∈{1,3}`: Activate sheet

```
1 →(1≠r←3 11 ⎕NA 'COM')/F10
2 →(~0 0≡↑(r em hex)←1 COM 'CREATE' 'Excel.Application')/F11
3 →(~0 0≡↑(r em hwb)←1 COM 'METHOD' hex 'Workbooks.Open'(1⊃dn))/F13
4 →(2≠⎕NC 'hws')/1+⎕LC◇→(~0 0≡↑(r em res)←1 COM 'RELEASE' hws)/F41
5 →(0 0≡↑(r em ns)←1 COM 'PROPERTY' hwb 'WORKSHEETS.COUNT')/F19
6 cs←↑((0=↑0⍴2⊃dn),(0 1=0=⍴2⊃dn))/⍳3 ⍝ number, name, Active sheet
7 →(cs>1)/1+⎕LC◇→(ns<2⊃dn)/F22
8 r←cs⊃('WORKSHEETS.item()'(2⊃dn))('WORKSHEETS()'(2⌷dn))(⊂'ActiveSheet')
  →(~0 0≡↑(r em hws)←1 COM 'PROPERTY' hwb,r)/F20
```

`hws`: handle to sheet

The steps inside the routine

r←s XLSles dn

Connecting, only if s[1]∈{1,3}:

```
1  →(1≠r←3 11 ⎕NA 'COM')/F10
2  →(~0 0≡↑(r em hex)←1 COM 'CREATE' 'Excel.Application')/F11
3  →(~0 0≡↑(r em hwb)←1 COM 'METHOD' hex 'Workbooks.Open'(1⊃dn))/F13
4  →(2≠⎕NC 'hws')/1+⎕LC⋄→(~0 0≡↑(r em res)←1 COM 'RELEASE' hws)/F41
5  →(0 0≡↑(r em ns)←1 COM 'PROPERTY' hwb 'WORKSHEETS.COUNT')/F19
6  cs←↑((0=↑0⍴2⊃dn),(0 1=0=⍴2⊃dn))/⍳3 ⍝ number, name, Active sheet
7  →(cs>1)/1+⎕LC⋄→(ns<2⊃dn)/F22
8  r←cs⊃('WORKSHEETS.item()'(2⊃dn))('WORKSHEETS()'(2⌷dn))(⊂'ActiveSheet')
   →(~0 0≡↑(r em hws)←1 COM 'PROPERTY' hwb,r)/F20
```

Reading from active sheet: Partial or all

```
9  r←(1+''≡3⊃dn)⊃((⊂'Cells().Resize().Value'),2↓dn)(⊂'UsedRange.Value')
   →(~0 0≡↑(r em res)←1 COM 'PROPERTY' hws,r)/F25⋄tx←res
10
```

The steps inside the routine

r←s XLSles dn

Connecting, only if s[1]∈{1,3}:

```
1  →(1≠r←3 11 ⎕NA 'COM')/F10
2  →(~0 0≡↑(r em hex)←1 COM 'CREATE' 'Excel.Application')/F11
3  →(~0 0≡↑(r em hwb)←1 COM 'METHOD' hex 'Workbooks.Open'(1⊃dn))/F13
4  →(2≠⎕NC 'hws')/1+⎕LC⋄→(~0 0≡↑(r em res)←1 COM 'RELEASE' hws)/F41
5  →(0 0≡↑(r em ns)←1 COM 'PROPERTY' hwb 'WORKSHEETS.COUNT')/F19
6  cs←↑((0=↑0⍴2⊃dn),(0 1=0=⍴2⊃dn))/⍳3 ⍝ number, name, Active sheet
7  →(cs>1)/1+⎕LC⋄→(ns<2⊃dn)/F22
8  r←cs⊃('WORKSHEETS.item()'(2⊃dn))('WORKSHEETS()'(2⌷dn))(⊂'ActiveSheet')
   →(~0 0≡↑(r em hws)←1 COM 'PROPERTY' hwb,r)/F20
```

Reading from active sheet: Whole sheet: postprocessing

```
9  r←(1+''≡3⊃dn)⊃((⊂'Cells().Resize().Value'),2↓dn)(⊂'UsedRange.Value')
   →(~0 0≡↑(r em res)←1 COM 'PROPERTY' hws,r)/F25⋄tx←res
10  →(~''≡3⊃dn)/ENDIF3⋄→(~tx≡'')/1+⎕LC⋄tx←0 0⍴tx⋄→ENDIF3
    →((1≠≡tx)∨1≠⍴⍴tx)/1+⎕LC⋄tx←1 1⍴⊂tx
    tx←(+/∨\⌽∨/r)(+/∨\⌽∨≠r←↑¨0≠⍴¨,¨tx)↑tx
   ENDIF3:
```

The steps inside the routine

r←s XLSles dn

Reading from active sheet:

```
9  r←(1+''≡3⊃dn)⊃((⊂'Cells().Resize().Value'),2↓dn)(⊂'UsedRange.Value')
   →(~0 0≡↑(r em res)←1 COM 'PROPERTY' hws,r)/F25⋄tx←res
10  →(~''≡3⊃dn)/ENDIF3⋄→(~tx≡'')/1+⎕LC⋄tx←0 0⍴tx⋄→ENDIF3
    →((1≠≡tx)∨1≠⍴⍴tx)/1+⎕LC⋄tx←1 1⍴⊂tx
    tx←(+/∨\⌽∨/r)(+/∨\⌽∨≠r←↑¨0≠⍴¨,¨tx)↑tx
   ENDIF3:
```

Getting names of all sheets: Name of active sheet

```
11 →(~0 0≡↑(r em asn)←1 COM'PROPERTY' hwb 'ActiveSheet.Name')/F23
12
13
```

The steps inside the routine

r←s XLSles dn

Reading from active sheet:

```
9  r←(1+''≡3⊃dn)⊃((⊂'Cells().Resize().Value'),2↓dn)(⊂'UsedRange.Value')
   →(~0 0≡↑(r em res)←1 COM 'PROPERTY' hws,r)/F25⋄tx←res
10  →(~''≡3⊃dn)/ENDIF3⋄→(~tx≡'')/1+⎕LC⋄tx←0 0⍴tx⋄→ENDIF3
    →((1≠≡tx)∨1≠⍴⍴tx)/1+⎕LC⋄tx←1 1⍴⊂tx
    tx←(+/∨\⌽∨/r)(+/∨\⌽∨≠r←↑¨0≠⍴¨,¨tx)↑tx
   ENDIF3:
```

Getting names of all sheets:

All names

```
11 →(~0 0≡↑(r em asn)←1 COM'PROPERTY' hwb 'ActiveSheet.Name')/F23
12  i←0⋄tx←0⍴⊂''
   DO:→(ns<i←i+1)/UNDO
    →(~0 0=↑(r em hws)←1 COM 'PROPERTY' hwb 'WORKSHEETS.item()' i)/F20
    →(~0 0≡↑(r em res)←1 COM 'PROPERTY' hws 'Name')/F21⋄tx←tx,⊂res
    →(~0 0≡↑(r em res)←1 COM 'RELEASE' hws)/F41
    →DO
   UNDO:
13
```

The steps inside the routine

r←s XLSles dn

Reading from active sheet:

```
9  r←(1+''≡3⊃dn)⊃((⊂'Cells().Resize().Value'),2↓dn)(⊂'UsedRange.Value')
   →(~0 0≡↑(r em res)←1 COM 'PROPERTY' hws,r)/F25⋄tx←res
10  →(~''≡3⊃dn)/ENDIF3⋄→(~tx≡'')/1+⎕LC⋄tx←0 0⍴tx⋄→ENDIF3
    →((1≠≡tx)∨1≠⍴⍴tx)/1+⎕LC⋄tx←1 1⍴⊂tx
    tx←(+/∨\⌽∨/r)(+/∨\⌽∨≠r←↑¨0≠⍴¨,¨tx)↑tx
ENDIF3:
```

Getting names of all sheets:

Activate previously active sheet

```
11 →(~0 0≡↑(r em asn)←1 COM'PROPERTY' hwb 'ActiveSheet.Name')/F23
12  i←0⋄tx←0⍴⊂''
DO:→(ns<i←i+1)/UNDO
    →(~0 0=↑(r em hws)←1 COM 'PROPERTY' hwb 'WORKSHEETS.item()' i)/F20
    →(~0 0≡↑(r em res)←1 COM 'PROPERTY' hws 'Name')/F21⋄tx←tx,⊂res
    →(~0 0≡↑(r em res)←1 COM 'RELEASE' hws)/F41
    →DO
UNDO:
13  i←tx⍳⊂asn
    →(~0 0=↑(r em hws)←1 COM 'PROPERTY' hwb 'WORKSHEETS.item()' i)/F20
```

The steps inside the routine

r←s XLSles dn

Getting names of all sheets:

```
11 →(~0 0≡↑(r em asn)←1 COM'PROPERTY' hwb 'ActiveSheet.Name')/F23
12  i←0⋄tx←0⍴⊂''
DO:→(ns<i←i+1)/UNDO
 →(~0 0=↑(r em hws)←1 COM 'PROPERTY' hwb 'WORKSHEETS.item()' i)/F20
 →(~0 0≡↑(r em res)←1 COM 'PROPERTY' hws 'Name')/F21⋄tx←tx,⊂res
 →(~0 0≡↑(r em res)←1 COM 'RELEASE' hws)/F41
 →DO
UNDO:
13  i←tx⍳⊂asn
 →(~0 0=↑(r em hws)←1 COM 'PROPERTY' hwb 'WORKSHEETS.item()' i)/F20
```

Saving handles and closing file:

```
14  ⍎'→CB',⍕1⎕s
CB0:xlshws←hws⋄→CBend
CB1:xlshex←hex⋄xlshwb←hwb⋄→(2≠⎕NC 'hws')/CBend⋄xlshws←hws⋄→CBend
CB2:xlshws←hws⋄→(0≠≡r←XLSscl 0)/FZ⋄→CBend0
CB3:r←hex hwb⋄→(2≠⎕NC 'hws')/1+⎕LC⋄r←r,hws
 →(0≠≡r←XLSscl r)/FZ⋄→CBend0
```

Calling the routine

```
r←s XLSscr dn

1 s[1] 0:    write data
       1:    connect to Excel, file, sheet, write
       2:    Write, disconnect sheet, file, Excel
       3:    1 and 2                                  [Def.: 3]
   [2] 0/1:  clean sheet first                        [Def.: 0]
   [3]       if sheet does not exist:                 [Def.: 0]
       -1:   do not create                            [Def.: 0]
       0:    create as last sheet
       >0:   create before s[3]
             if too large: create at the end
       TV    create before s[3]
   [4] 0/1:  if file does not exist, create it        [Def.: 0]

2 dn[1] TV    file name
    [2] TV    name of sheet to be written to
              empty: active sheet                  [DEF.: '']
    [3] V[2]  left upper cell
    [4] AM    data matrix, defines size
```

Calling the routine

```
r←s XLSscr dn

1 s[1] 0:   write data
       1:   connect to Excel, file, sheet, write
       2:   Write, disconnect sheet, file, Excel
       3:   1 and 2                                  [Def.: 3]
   [2] 0/1: clean sheet first                        [Def.: 0]
   [3]      if sheet does not exist:                 [Def.: 0]
       -1:  do not create                            [Def.: 0]
       0:   create as last sheet
       >0:  create before s[3]
            if too large: create at the end
       TV   create before s[3]
   [4] 0/1: if file does not exist, create it        [Def.: 0]

2 dn[1] TV   file name
    [2] TV   name of sheet to be written to
             empty: active sheet                [DEF.: '']
    [3] V[2] left upper cell
    [4] AM   data matrix, defines size
```

The steps inside the routine

```
r←s XLSscr dn
```

Connecting, only if `s[1]∈{1,3}`: Connecting to COM

```
1 →(1≠r←3 11 ⎕NA 'COM')/F10
2
3
4
5
6
7
8
9
```

The steps inside the routine

r←s XLSscr dn

Connecting, only if s[1]∈{1,3}: Connecting to Excel

1 →(1≠r←3 11 ⎕NA 'COM')/F10

2 →(~0 0≡↑(r em hex)←1 COM 'CREATE' 'Excel.Application')/F11

hex: handle to Excel.

3

4

5

6

7

8

9

The steps inside the routine

r←s XLSscr dn

Connecting, only if s[1]∈{1,3}: Connecting to file

1 →(1≠r←3 11 ⎕NA 'COM')/F10

2 →(~0 0≡↑(r em hex)←1 COM 'CREATE' 'Excel.Application')/F11

3 ldat←0 0≡↑(r em hwb)←1 COM 'METHOD' hex 'Workbooks.Open'(1⊃dn)
→(~ldat∨(4⌷s)∧r≡1 5)/F13
hwb: handle to file, ldat: opening existing file?

4

5

6

7

8

9

The steps inside the routine

r←s XLSscr dn

Connecting, only if s[1]∈{1,3}:

1. →(1≠r←3 11 ⎕NA 'COM')/F10
2. →(~0 0≡↑(r em hex)←1 COM 'CREATE' 'Excel.Application')/F11
3. ldat←0 0≡↑(r em hwb)←1 COM 'METHOD' hex 'Workbooks.Open'(1⊃dn)
   →(~ldat∨(4⌷s)∧r≡1 5)/F13

New file: Create new file

4. →(~0 0≡↑(r em hwb)←1 COM 'METHOD' hex 'Workbooks.Add')/F12
   hwb: handle to file.
5.
6.
7.
8.
9.

The steps inside the routine

r←s XLSscr dn

Connecting, only if s[1]∈{1,3}:

1 →(1≠r←3 11 ⎕NA 'COM')/F10

2 →(~0 0≡↑(r em hex)←1 COM 'CREATE' 'Excel.Application')/F11

3 ldat←0 0≡↑(r em hwb)←1 COM 'METHOD' hex 'Workbooks.Open'(1⊃dn)
→(~ldat∨(4⍴s)∧r≡1 5)/F13

New file: Create new sheet, give name

4 →(~0 0≡↑(r em hwb)←1 COM 'METHOD' hex 'Workbooks.Add')/F12

5 →(~0 0≡↑(r em hws)←1 COM 'PROPERTY' hwb 'ActiveSheet')/F20
→(0=⍴r←2⊃dn)/1+⎕LC◇→(~0 0≡↑(r em res)←1 COM 'PROPERTY' hws 'Name' r)/F28
hws: handle to sheet

6

7

8

9

The steps inside the routine

r←s XLSscr dn

Connecting, only if s[1]∈{1,3}:

1 →(1≠r←3 11 ⎕NA 'COM')/F10

2 →(~0 0≡↑(r em hex)←1 COM 'CREATE' 'Excel.Application')/F11

3 ldat←0 0≡↑(r em hwb)←1 COM 'METHOD' hex 'Workbooks.Open'(1⊃dn)
→(~ldat∨(4⍴s)∧r≡1 5)/F13

New file:

4 →(~0 0≡↑(r em hwb)←1 COM 'METHOD' hex 'Workbooks.Add')/F12

5 →(~0 0≡↑(r em hws)←1 COM 'PROPERTY' hwb 'ActiveSheet')/F20
→(0=⍴r←2⊃dn)/1+⎕LC◇→(~0 0≡↑(r em res)←1 COM 'PROPERTY' hws 'Name' r)/F28

Existing file: Release active sheet

6 →(2≠⎕NC 'hws')/1+⎕LC◇→(~0 0≡↑(r em res)←1 COM 'RELEASE' hws)/F41

7

8

9

The steps inside the routine

r←s XLSscr dn

Connecting, only if s[1]∈{1,3}:

1 →(1≠r←3 11 ⎕NA 'COM')/F10

2 →(~0 0≡↑(r em hex)←1 COM 'CREATE' 'Excel.Application')/F11

3 ldat←0 0≡↑(r em hwb)←1 COM 'METHOD' hex 'Workbooks.Open'(1⊃dn)
→(~ldat∨(4⌷s)∧r≡1 5)/F13

New file:

4 →(~0 0≡↑(r em hwb)←1 COM 'METHOD' hex 'Workbooks.Add')/F12

5 →(~0 0≡↑(r em hws)←1 COM 'PROPERTY' hwb 'ActiveSheet')/F20
→(0=⍴r←2⊃dn)/1+⎕LC◇→(~0 0≡↑(r em res)←1 COM 'PROPERTY' hws 'Name' r)/F28

Existing file: Number of sheets

6 →(2≠⎕NC 'hws')/1+⎕LC◇→(~0 0≡↑(r em res)←1 COM 'RELEASE' hws)/F41

7 →(0 0≡↑(r em ns)←1 COM 'PROPERTY' hwb 'WORKSHEETS.COUNT')/F19
ns: number of work sheets.

8

9

The steps inside the routine

r←s XLSscr dn

Connecting, only if s[1]∈{1,3}:

1. →(1≠r←3 11 ⎕NA 'COM')/F10
2. →(~0 0≡↑(r em hex)←1 COM 'CREATE' 'Excel.Application')/F11
3. ldat←0 0≡↑(r em hwb)←1 COM 'METHOD' hex 'Workbooks.Open'(1⊃dn)
   →(~ldat∨(4⍴s)∧r≡1 5)/F13

New file:

4. →(~0 0≡↑(r em hwb)←1 COM 'METHOD' hex 'Workbooks.Add')/F12
5. →(~0 0≡↑(r em hws)←1 COM 'PROPERTY' hwb 'ActiveSheet')/F20
   →(0=⍴r←2⊃dn)/1+⎕LC◇→(~0 0≡↑(r em res)←1 COM 'PROPERTY' hws 'Name' r)/F28

Existing file:

Cases to connect to a sheet

6. →(2≠⎕NC 'hws')/1+⎕LC◇→(~0 0≡↑(r em res)←1 COM 'RELEASE' hws)/F41
7. →(0 0≡↑(r em ns)←1 COM 'PROPERTY' hwb 'WORKSHEETS.COUNT')/F19
8. cs←↑((0=↑0⍴2⊃dn),(0 1=0=⍴2⊃dn))/⍳3 ⍝ number, name, Active sheet
   cs: way of addressing sheet.
9.

The steps inside the routine

r←s XLSscr dn

Connecting, only if s[1]∈{1,3}:

1 →(1≠r←3 11 ⎕NA 'COM')/F10

2 →(~0 0≡↑(r em hex)←1 COM 'CREATE' 'Excel.Application')/F11

3 ldat←0 0≡↑(r em hwb)←1 COM 'METHOD' hex 'Workbooks.Open'(1⊃dn)
→(~ldat∨(4⍴s)∧r≡1 5)/F13

New file:

4 →(~0 0≡↑(r em hwb)←1 COM 'METHOD' hex 'Workbooks.Add')/F12

5 →(~0 0≡↑(r em hws)←1 COM 'PROPERTY' hwb 'ActiveSheet')/F20
→(0=⍴r←2⊃dn)/1+⎕LC◇→(~0 0≡↑(r em res)←1 COM 'PROPERTY' hws 'Name' r)/F28

Existing file:

Cases to connect to a sheet

6 →(2≠⎕NC 'hws')/1+⎕LC◇→(~0 0≡↑(r em res)←1 COM 'RELEASE' hws)/F41

7 →(0 0≡↑(r em ns)←1 COM 'PROPERTY' hwb 'WORKSHEETS.COUNT')/F19

8 cs←↑((0=↑0⍴2⊃dn),(0 1=0=⍴2⊃dn))/⍳3 ⍝ number, name, Active sheet

9 r←cs⊃('WORKSHEETS.item()'(2⊃dn))('WORKSHEETS()'(2⍴dn))(⊂'ActiveSheet')
→(0 0≡↑(r em hws)←1 COM 'PROPERTY' hwb,r)/ENDIF2

hws: handle to sheet

The steps inside the routine

r←s XLSscr dn

Existing file:

```
6  →(2≠⎕NC 'hws')/1+⎕LC◇→(~0 0≡↑(r em res)←1 COM 'RELEASE' hws)/F41
7  →(0 0≡↑(r em ns)←1 COM 'PROPERTY' hwb 'WORKSHEETS.COUNT')/F19
8  cs←↑((0=↑0⍴2⊃dn),(0 1=0=⍴2⊃dn))/⍳3 ⍝ number, name, Active sheet
9  r←cs⊃('WORKSHEETS.item()'(2⊃dn))('WORKSHEETS()'(2⌷dn))(⊂'ActiveSheet')
   →(0 0≡↑(r em hws)←1 COM 'PROPERTY' hwb,r)/ENDIF2
```

Existing file, new sheet: Add sheet: before, given by name

```
10 (r em hws)←1 COM 'PROPERTY' hex 'Worksheets()',(⊂s[3])
11
12
```

The steps inside the routine

r←s XLSscr dn

Existing file:

```
6  →(2≠⎕NC 'hws')/1+⎕LC◇→(~0 0≡↑(r em res)←1 COM 'RELEASE' hws)/F41
7  →(0 0≡↑(r em ns)←1 COM 'PROPERTY' hwb 'WORKSHEETS.COUNT')/F19
8  cs←↑((0=↑0⍴2⊃dn),(0 1=0=⍴2⊃dn))/⍳3 ⍝ number, name, Active sheet
9  r←cs⊃('WORKSHEETS.item()'(2⊃dn))('WORKSHEETS()'(2⌷dn))(⊂'ActiveSheet')
   →(0 0≡↑(r em hws)←1 COM 'PROPERTY' hwb,r)/ENDIF2
```

Existing file, new sheet: Add sheet: before, given by name

```
10 (r em hws)←1 COM 'PROPERTY' hex 'Worksheets()',(⊂s[3])
```

ok: `→(~0 0≡↑(r em hws)←1 COM 'METHOD' hex 'Worksheets.Add[]'('Before' hws))/F24`

```
11
12
```

The steps inside the routine

r←s XLSscr dn

Existing file:

```
6  →(2≠⎕NC 'hws')/1+⎕LC◇→(~0 0≡↑(r em res)←1 COM 'RELEASE' hws)/F41
7  →(0 0≡↑(r em ns)←1 COM 'PROPERTY' hwb 'WORKSHEETS.COUNT')/F19
8  cs←↑((0=↑0⍴2⊃dn),(0 1=0=⍴2⊃dn))/⍳3 ⍝ number, name, Active sheet
9  r←cs⊃('WORKSHEETS.item()'(2⊃dn))('WORKSHEETS()'(2⌷dn))(⊂'ActiveSheet')
   →(0 0≡↑(r em hws)←1 COM 'PROPERTY' hwb,r)/ENDIF2
```

Existing file, new sheet: Add sheet at end of file

```
10 (r em hws)←1 COM 'PROPERTY' hex 'Worksheets()',(⊂s[3])
```

ok: `→(~0 0≡↑(r em hws)←1 COM 'METHOD' hex 'Worksheets.Add[]'('Before' hws))/F24`

not ok: `→(~0 0≡↑(r em hws)←1 COM 'PROPERTY' hex 'Worksheets.Item()' ns)/F20`

`→(~0 0≡↑(r em hws)←1 COM 'METHOD' hex 'Worksheets.Add[]'('After' hws))/F24`

```
11
12
```

The steps inside the routine

r←s XLSscr dn

Existing file:

```
6  →(2≠⎕NC 'hws')/1+⎕LC◇→(~0 0≡↑(r em res)←1 COM 'RELEASE' hws)/F41
7  →(0 0≡↑(r em ns)←1 COM 'PROPERTY' hwb 'WORKSHEETS.COUNT')/F19
8  cs←↑((0=↑0⍴2⊃dn),(0 1=0=⍴2⊃dn))/⍳3 ⍝ number, name, Active sheet
9  r←cs⊃('WORKSHEETS.item()'(2⊃dn))('WORKSHEETS()'(2⌷dn))(⊂'ActiveSheet')
   →(0 0≡↑(r em hws)←1 COM 'PROPERTY' hwb,r)/ENDIF2
```

Existing file, new sheet: Add sheet: before, given by number

```
10 (r em hws)←1 COM 'PROPERTY' hex 'Worksheets()',(⊂s[3])
```

ok: `→(~0 0≡↑(r em hws)←1 COM 'METHOD' hex 'Worksheets.Add[]'('Before' hws))/F24`

not ok: `→(~0 0≡↑(r em hws)←1 COM 'PROPERTY' hex 'Worksheets.Item()' ns)/F20`

`→(~0 0≡↑(r em hws)←1 COM 'METHOD' hex 'Worksheets.Add[]'('After' hws))/F24`

```
11  ⍎(ns<3⌷s)/'s[3]←0'
    r←(0 1=0=3⌷s)/s[3],ns
    →(~0 0≡↑(r em hws)←1 COM 'PROPERTY' hex 'Worksheets.Item()' r)/F20
    r←∊(0 1=s[3]=0)/'Before' 'After'
    →(~0 0≡↑(r em hws)←1 COM 'METHOD' hex 'Worksheets.Add[]'(r hws))/F24
12
```

The steps inside the routine

r←s XLSscr dn

Existing file:

```
6  →(2≠⎕NC 'hws')/1+⎕LC◇→(~0 0≡↑(r em res)←1 COM 'RELEASE' hws)/F41
7  →(0 0≡↑(r em ns)←1 COM 'PROPERTY' hwb 'WORKSHEETS.COUNT')/F19
8  cs←↑((0=↑0⍴2⊃dn),(0 1=0=⍴2⊃dn))/⍳3 ⍝ number, name, Active sheet
9  r←cs⊃('WORKSHEETS.item()'(2⊃dn))('WORKSHEETS()'(2⌷dn))(⊂'ActiveSheet')
   →(0 0≡↑(r em hws)←1 COM 'PROPERTY' hwb,r)/ENDIF2
```

Existing file, new sheet: Give name

```
10 (r em hws)←1 COM 'PROPERTY' hex 'Worksheets()',(⊂s[3])
```

ok: `→(~0 0≡↑(r em hws)←1 COM 'METHOD' hex 'Worksheets.Add[]'('Before' hws))/F24`

not ok: `→(~0 0≡↑(r em hws)←1 COM 'PROPERTY' hex 'Worksheets.Item()' ns)/F20`

`→(~0 0≡↑(r em hws)←1 COM 'METHOD' hex 'Worksheets.Add[]'('After' hws))/F24`

```
11  ⍎(ns<3⌷s)/'s[3]←0'
    r←(0 1=0=3⌷s)/s[3],ns
    →(~0 0≡↑(r em hws)←1 COM 'PROPERTY' hex 'Worksheets.Item()' r)/F20
    r←∈(0 1=s[3]=0)/'Before' 'After'
    →(~0 0≡↑(r em hws)←1 COM 'METHOD' hex 'Worksheets.Add[]'(r hws))/F24

12 →(0=⍴r←2⊃dn)/1+⎕LC◇→(~0 0≡↑(r em res)←1 COM 'PROPERTY' hws 'Name' r)/F28
```

The steps inside the routine

```
r←s XLSscr dn
```

Writing data: Clean sheet

```
14 →(~0 0≡↑(r em tx)←1 COM 'PROPERTY' hws 'UsedRange.Value' '')/F25
   →(~0 0≡↑(r em tx)←1 COM 'PROPERTY' hws 'UsedRange.ClearFormats')/F26
15
16
```

The steps inside the routine

```
r←s XLSscr dn
```

Writing data: Write data

```
14 →(~0 0≡↑(r em tx)←1 COM 'PROPERTY' hws 'UsedRange.Value' '')/F25
   →(~0 0≡↑(r em tx)←1 COM 'PROPERTY' hws 'UsedRange.ClearFormats')/F26
15 r←'PROPERTY' hws 'Cells().Resize().Value',dn[3],(⊂⍴4⊃dn),dn[4]
   →(~0 0≡↑(r em res)←1 COM r)/F27
16
```

The steps inside the routine

```
r←s XLSscr dn
```

Writing data: Autofit columns

```
14 →(~0 0≡↑(r em tx)←1 COM 'PROPERTY' hws 'UsedRange.Value' '')/F25
   →(~0 0≡↑(r em tx)←1 COM 'PROPERTY' hws 'UsedRange.ClearFormats')/F26
15 r←'PROPERTY' hws 'Cells().Resize().Value',dn[3],(⊂⍴4⊃dn),dn[4]
   →(~0 0≡↑(r em res)←1 COM r)/F27
16 →(~0 0≡↑(r em res)←1 COM 'PROPERTY' hws 'UsedRange.Columns.Autofit')/F31
```

The steps inside the routine

r←s XLSscr dn

Writing data:

```
14 →(~0 0≡↑(r em tx)←1 COM 'PROPERTY' hws 'UsedRange.Value' '')/F25
   →(~0 0≡↑(r em tx)←1 COM 'PROPERTY' hws 'UsedRange.ClearFormats')/F26
15 r←'PROPERTY' hws 'Cells().Resize().Value',dn[3],(⊂⍴4⊃dn),dn[4]
   →(~0 0≡↑(r em res)←1 COM r)/F27
16 →(~0 0≡↑(r em res)←1 COM 'PROPERTY' hws 'UsedRange.Columns.Autofit')/F31
```

Closing file: Save sheet

```
17 (r em res)←1 COM 'METHOD' hwb,↑(1 0=ldat)/(⊂'Save')('SaveAs'(1⊃dn))
   →((1 0=ldat)∧~0 0≡r)/F42,F43
18
```

The steps inside the routine

r←s XLSscr dn

Writing data:

```
14 →(~0 0≡↑(r em tx)←1 COM 'PROPERTY' hws 'UsedRange.Value' '')/F25
   →(~0 0≡↑(r em tx)←1 COM 'PROPERTY' hws 'UsedRange.ClearFormats')/F26
15 r←'PROPERTY' hws 'Cells().Resize().Value',dn[3],(⊂⍴4⊃dn),dn[4]
   →(~0 0≡↑(r em res)←1 COM r)/F27
16 →(~0 0≡↑(r em res)←1 COM 'PROPERTY' hws 'UsedRange.Columns.Autofit')/F31
```

Closing file:

Save global var., release all

```
17 (r em res)←1 COM 'METHOD' hwb,↑(1 0=ldat)/(⊂'Save')('SaveAs'(1⊃dn))
   →((1 0=ldat)∧~0 0≡r)/F42,F43

18  ⍎'→CB',⍕↑s
   CB0:xlshws←hws◇→CBend
   CB1:xlshex←hex◇xlshwb←hwb◇xlshws←hws◇→CBend
   CB2:xlshws←hws◇→(0≠≡r←XLSscl 0)/FZ◇→CBend0
   CB3:→(0≠≡r←XLSscl hex hwb hws)/FZ
   CBend:
```

Calling the routine

```
r←hs XLSscl griff
```

```
1 hs     LS  0/1: screen output?
2 griff  V[1]   handle to Excel
          [2]   handle to file
          [3]   ≠0: handle to sheet, if connected
         S      0: use global handles
```

Calling the routine

```
r←hs XLSscl griff
```

1. `hs  LS  0/1: screen output?`
2. `griff  V[1]  handle to Excel`
   `[2]  handle to file`
   `[3]  ≠0: handle to sheet, if connected`
   `S  0: use global handles`

The steps inside the routine

```
r←hs XLSscl griff
```

`griff` contains the handles or 0 and gets `griff[4]←xlssav`

Releasing sheet:

1 `→(0=3⌷griff)/1+⎕LC◇→(~0 0≡↑(r em res)←1 COM 'RELEASE' griff[3])/F41`

The steps inside the routine

```
r←hs XLSscl griff
```

`griff` contains the handles or 0 and gets `griff[4]←xlssav`

Releasing sheet:

1. `→(0=3⌷griff)/1+⎕LC◇→(~0 0≡↑(r em res)←1 COM 'RELEASE' griff[3])/F41`

Saving and Releasing file: Save file

2. `→(~4⌷griff)/1+⎕LC◇→(0 0≠↑(r em res)←1 COM 'METHOD' griff[2]'Save')/F42`

The steps inside the routine

```
r←hs XLSscl griff
```

`griff` contains the handles or 0 and gets `griff[4]←xlssav`

Releasing sheet:

1 `→(0=3⌷griff)/1+⎕LC◇→(~0 0≡↑(r em res)←1 COM 'RELEASE' griff[3])/F41`

Saving and Releasing file: Allow releasing Excel

2 `→(~4⌷griff)/1+⎕LC◇→(0 0≠↑(r em res)←1 COM 'METHOD' griff[2]'Save')/F42`

3 `→(~0 0≡↑(r em res)←1 COM 'PROPERTY' griff[1]'Visible' 0)/F43`

The steps inside the routine

```
r←hs XLSscl griff
```

`griff` contains the handles or 0 and gets `griff[4]←xlssav`

Releasing sheet:

1 `→(0=3⌷griff)/1+⎕LC◇→(~0 0≡↑(r em res)←1 COM 'RELEASE' griff[3])/F41`

Saving and Releasing file: Hint at closing file ?

2 `→(~4⌷griff)/1+⎕LC◇→(0 0≠↑(r em res)←1 COM 'METHOD' griff[2]'Save')/F42`

3 `→(~0 0≡↑(r em res)←1 COM 'PROPERTY' griff[1]'Visible' 0)/F43`

4 `→(~0 0≡↑(r em res)←1 COM 'PROPERTY' griff[1]'DisplayAlerts' hs)/F43`

The steps inside the routine

```
r←hs XLSscl griff
```

`griff` contains the handles or 0 and gets `griff[4]←xlssav`

Releasing sheet:

1. `→(0=3⌷griff)/1+⎕LC◇→(~0 0≡↑(r em res)←1 COM 'RELEASE' griff[3])/F41`

Saving and Releasing file: Close file

2. `→(~4⌷griff)/1+⎕LC◇→(0 0≠↑(r em res)←1 COM 'METHOD' griff[2]'Save')/F42`
3. `→(~0 0≡↑(r em res)←1 COM 'PROPERTY' griff[1]'Visible' 0)/F43`
4. `→(~0 0≡↑(r em res)←1 COM 'PROPERTY' griff[1]'DisplayAlerts' hs)/F43`
5. `→(~0 0≡↑(r em res)←1 COM 'METHOD' griff[2]'Close')/F44`

The steps inside the routine

```
r←hs XLSscl griff
```

`griff` contains the handles or 0 and gets `griff[4]←xlssav`

Releasing sheet:

1. `→(0=3⌷griff)/1+⎕LC◇→(~0 0≡↑(r em res)←1 COM 'RELEASE' griff[3])/F41`

Saving and Releasing file: Release method (file)

2. `→(~4⌷griff)/1+⎕LC◇→(0 0≠↑(r em res)←1 COM 'METHOD' griff[2]'Save')/F42`
3. `→(~0 0≡↑(r em res)←1 COM 'PROPERTY' griff[1]'Visible' 0)/F43`
4. `→(~0 0≡↑(r em res)←1 COM 'PROPERTY' griff[1]'DisplayAlerts' hs)/F43`
5. `→(~0 0≡↑(r em res)←1 COM 'METHOD' griff[2]'Close')/F44`
6. `COM 'RELEASE' griff[2]`

The steps inside the routine

```
r←hs XLSscl griff
```

`griff` contains the handles or 0 and gets `griff[4]←xlssav`

Releasing sheet:

1. `→(0=3⌷griff)/1+⎕LC◇→(~0 0≡↑(r em res)←1 COM 'RELEASE' griff[3])/F41`

Saving and Releasing file: Release Excel

2. `→(~4⌷griff)/1+⎕LC◇→(0 0≠↑(r em res)←1 COM 'METHOD' griff[2]'Save')/F42`
3. `→(~0 0≡↑(r em res)←1 COM 'PROPERTY' griff[1]'Visible' 0)/F43`
4. `→(~0 0≡↑(r em res)←1 COM 'PROPERTY' griff[1]'DisplayAlerts' hs)/F43`
5. `→(~0 0≡↑(r em res)←1 COM 'METHOD' griff[2]'Close')/F44`
6. `COM 'RELEASE' griff[2]`
7. `COM 'RELEASE' griff[1]`

Setting properties

r←griff XLSprp par

Sets a property

```
→(0 0≠↑r←1 COM 'PROPERTY' griff,par)/FCOM
xlssav←1
```

Setting properties

r←griff XLSprp par — Sets a property

```
→(0 0≠↑r←1 COM 'PROPERTY' griff,par)/FCOM
xlssav←1
```

r←XLSpws par — ... of a sheet

```
→(0>↑r←xlshws XLSprp par)/FZ
```

Setting properties

r←griff XLSprp par — Sets a property

```
→(0 0≠↑r←1 COM 'PROPERTY' griff,par)/FCOM
xlssav←1
```

r←XLSpws par — ... of a sheet

```
→(0>↑r←xlshws XLSprp par)/FZ
```

r←XLSpwb par — ... of a file

```
→(0>↑r←xlshwb XLSprp par)/FZ
```

Translating Positions from matrix to Excel notation

r←XLSpos num — Positions

```
Calculates position in Excel-format
num     V         vector with two elements
        AV V      vectors with two elements
r       TV        position
        AV TV     positions
```

Example:

```
     XLSpos (2 2) (17 77) (1 120)

B2 BY17 DP1
```

Translating Positions from matrix to Excel notation

r←XLSpos num — Positions

Example:

```
      XLSpos (2 2) (17 77) (1 120)
B2 BY17 DP1
```

r←XLSspl num — Area

```
Column in Excel-format:
num     S/V         Nummern der Spalten        (0 1∨.=≡num)
r       AS/AV TV    kodierte Spaltennummern

Area in Excel-format:
num     AV[2] V[2] area in matrix format  2=∧.=∈(≡num) (⍴num) (⍴⍤¨num)
r       TV         area in Excel-format
```

Example:

```
      XLSspl (17 19)(27 117)
S17:DM27
```

Further methods of Excel

Delete a work sheet

```
1 COM 'METHOD' xlshex 'Worksheets().Delete' 2
```

Further methods of Excel

Delete a work sheet

```
1 COM 'METHOD' xlshex 'Worksheets().Delete' 2
```

Create a window asking for input

```
1 COM 'METHOD' xlshex 'InputBox' 'Enter your name' 'APL2 Sample'
```

Further Properties of a work book (file)

Select a work sheet

```
xlshws←COM 'PROPERTY' xlshwb 'WORKSHEETS()'(⊂'Tabelle1')
```

Further Properties of a work book (file)

Select a work sheet

```
xlshws←COM 'PROPERTY' xlshwb 'WORKSHEETS()'(⊂'Tabelle1')

xlshws←COM 'PROPERTY' xlshwb 'Sheets().Select' 1
```

Further Properties of a work book (file)

Select a work sheet

```
xlshws←COM 'PROPERTY' xlshwb 'WORKSHEETS()'(⊂'Tabelle1')
xlshws←COM 'PROPERTY' xlshwb 'Sheets().Select' 1
xlshws←COM 'PROPERTY' xlshwb 'ActiveSheet'
```

Further Properties of a work book (file)

Select a work sheet

```
xlshws←COM 'PROPERTY' xlshwb 'WORKSHEETS()'(⊂'Tabelle1')
xlshws←COM 'PROPERTY' xlshwb 'Sheets().Select' 1
xlshws←COM 'PROPERTY' xlshwb 'ActiveSheet'
```

Change font colour of worksheet Tabelle1 to green (RGB)

```
1 COM 'PROPERTY' xlshwb 'WORKSHEETS().[B2].Font.Color'(⊂'Tabelle1')65280
```

Further Properties of a work book (file)

Select a work sheet

```
xlshws←COM 'PROPERTY' xlshwb 'WORKSHEETS()'(⊂'Tabelle1')
xlshws←COM 'PROPERTY' xlshwb 'Sheets().Select' 1
xlshws←COM 'PROPERTY' xlshwb 'ActiveSheet'
```

Change font colour of worksheet Tabelle1 to green (RGB)

```
1 COM 'PROPERTY' xlshwb 'WORKSHEETS().[B2].Font.Color'(⊂'Tabelle1')65280
```

Change font shape in Tabelle1 to bold

```
1 COM 'PROPERTY' xlshwb 'WORKSHEETS().[B2].Font.Bold'(⊂'Tabelle1')1
```

Further Properties of a work sheet

Read content of a work sheet: row 1

```
1 COM 'PROPERTY' xlshws,'rows().Value' 1
```

Further Properties of a work sheet

Read content of a work sheet:

column 1

```
1 COM 'PROPERTY' xlshws,'rows().Value' 1
1 COM 'PROPERTY' xlshws,'columns().Value' 1
```

Further Properties of a work sheet

Read content of a work sheet: row 5, 3 cells

```
1 COM 'PROPERTY' xlshws,'rows().Value' 1

1 COM 'PROPERTY' xlshws,'columns().Value' 1

1 COM 'PROPERTY' xlshws,'rows().Resize().Value' 5(1 3)
```

Further Properties of a work sheet

Read content of a work sheet: row 5, 3 cells

```
1 COM 'PROPERTY' xlshws,'rows().Value' 1
1 COM 'PROPERTY' xlshws,'columns().Value' 1
1 COM 'PROPERTY' xlshws,'rows().Resize().Value' 5(1 3)
1 COM 'PROPERTY' xlshws,'Cells().Resize().Value' (5 1)(1 3)
```

Further Properties of a work sheet

Read content of a work sheet: complete work sheet

```
1 COM 'PROPERTY' xlshws,'rows().Value' 1
1 COM 'PROPERTY' xlshws,'columns().Value' 1
1 COM 'PROPERTY' xlshws,'rows().Resize().Value' 5(1 3)
1 COM 'PROPERTY' xlshws,'Cells().Resize().Value' (5 1)(1 3)
1 COM 'PROPERTY' xlshws,'UsedRange.Value'
```

Further Properties of a work sheet

Read content of a work sheet:

```
1 COM 'PROPERTY' xlshws,'rows().Value' 1
1 COM 'PROPERTY' xlshws,'columns().Value' 1
1 COM 'PROPERTY' xlshws,'rows().Resize().Value' 5(1 3)
1 COM 'PROPERTY' xlshws,'Cells().Resize().Value' (5 1)(1 3)
1 COM 'PROPERTY' xlshws,'UsedRange.Value'
```

Write content to a work sheet: (4 4)-matrix, starting at D2

```
1 COM 'PROPERTY' xlshws,'Cells().Resize().Value',(2 4),(4 4),⊂4 4⍴⍳16
```

Further Properties of a work sheet

Read content of a work sheet:

```
1 COM 'PROPERTY' xlshws,'rows().Value' 1
1 COM 'PROPERTY' xlshws,'columns().Value' 1
1 COM 'PROPERTY' xlshws,'rows().Resize().Value' 5(1 3)
1 COM 'PROPERTY' xlshws,'Cells().Resize().Value' (5 1)(1 3)
1 COM 'PROPERTY' xlshws,'UsedRange.Value'
```

Write content to a work sheet:

no error, but no effect

```
1 COM 'PROPERTY' xlshws,'Cells().Resize().Value',(2 4),(4 4),⊂4 4⍴⍳16
1 COM 'PROPERTY' xlshws,'rows().Value' 1(1 5⍴12)
```

Further Properties of a work sheet

Read content of a work sheet:

```
1 COM 'PROPERTY' xlshws,'rows().Value' 1
1 COM 'PROPERTY' xlshws,'columns().Value' 1
1 COM 'PROPERTY' xlshws,'rows().Resize().Value' 5(1 3)
1 COM 'PROPERTY' xlshws,'Cells().Resize().Value' (5 1)(1 3)
1 COM 'PROPERTY' xlshws,'UsedRange.Value'
```

Write content to a work sheet:

no error, but no effect

```
1 COM 'PROPERTY' xlshws,'Cells().Resize().Value',(2 4),(4 4),⊂4 4⍴⍳16
1 COM 'PROPERTY' xlshws,'rows().Value' 1(1 5⍴12)
1 COM 'PROPERTY' xlshws,'rows().Resize().Value' 5(1 3)(1 3⍴⍳3)
```

Further Properties of a work sheet

Change font or background colour in actual work sheet

```
1 COM 'PROPERTY' xlshws 'UsedRange.Font.Color' 65280
```

Further Properties of a work sheet

Change font or background colour in actual work sheet

```
1 COM 'PROPERTY' xlshws 'UsedRange.Font.Color' 65280
1 COM 'PROPERTY' xlshws 'Range().Font.Color'(⊂'A1:B6')65280
```

Further Properties of a work sheet

Change font or background colour in actual work sheet

```
1 COM 'PROPERTY' xlshws 'UsedRange.Font.Color' 65280
1 COM 'PROPERTY' xlshws 'Range().Font.Color'(⊂'A1:B6')65280
1 COM 'PROPERTY' xlshws 'Range().Interior.Color'(⊂'A1:B6')65280
```

Further Properties of a work sheet

Change font or background colour in actual work sheet

```
1 COM 'PROPERTY' xlshws 'UsedRange.Font.Color' 65280
1 COM 'PROPERTY' xlshws 'Range().Font.Color'(⊂'A1:B6')65280
1 COM 'PROPERTY' xlshws 'Range().Interior.Color'(⊂'A1:B6')65280
1 COM 'PROPERTY' xlshws 'Range().Columns.Interior.Color'(⊂'A:F')65280
```

Further Properties of a work sheet

Change font or background colour in actual work sheet

```
1 COM 'PROPERTY' xlshws 'UsedRange.Font.Color' 65280
1 COM 'PROPERTY' xlshws 'Range().Font.Color'(⊂'A1:B6')65280
1 COM 'PROPERTY' xlshws 'Range().Interior.Color'(⊂'A1:B6')65280
1 COM 'PROPERTY' xlshws 'Range().Columns.Interior.Color'(⊂'A:F')65280
1 COM 'PROPERTY' xlshws 'Range().Columns.Font.Color'(⊂'7:7')65280
```

Further Properties of a work sheet

Change font or background colour in actual work sheet

```
1 COM 'PROPERTY' xlshws 'UsedRange.Font.Color' 65280
1 COM 'PROPERTY' xlshws 'Range().Font.Color'(c'A1:B6')65280
1 COM 'PROPERTY' xlshws 'Range().Interior.Color'(c'A1:B6')65280
1 COM 'PROPERTY' xlshws 'Range().Columns.Interior.Color'(c'A:F')65280
1 COM 'PROPERTY' xlshws 'Range().Columns.Font.Color'(c'7:7')65280
1 COM 'PROPERTY' xlshws 'Range().Columns.Font.Color'(c'H:H')65280
```

Further Properties of a work sheet

Change font or background colour in actual work sheet

```
1 COM 'PROPERTY' xlshws 'UsedRange.Font.Color' 65280
1 COM 'PROPERTY' xlshws 'Range().Font.Color'(c'A1:B6')65280
1 COM 'PROPERTY' xlshws 'Range().Interior.Color'(c'A1:B6')65280
1 COM 'PROPERTY' xlshws 'Range().Columns.Interior.Color'(c'A:F')65280
1 COM 'PROPERTY' xlshws 'Range().Columns.Font.Color'(c'7:7')65280
1 COM 'PROPERTY' xlshws 'Range().Columns.Font.Color'(c'H:H')65280
1 COM 'PROPERTY' xlshws 'Range().Rows.Font.Color'(c'6:6')65280
```

Further Properties of a work sheet

Change font shape in actual work sheet to bold, italic, underline

```
1 COM 'PROPERTY' xlshws 'UsedRange.Font.Bold' 1
```

Further Properties of a work sheet

Change font shape in actual work sheet to bold, italic, underline

```
1 COM 'PROPERTY' xlshws 'UsedRange.Font.Bold' 1

1 COM 'PROPERTY' xlshws 'Range().Font.Bold'(c'A1:B6')1
```

Further Properties of a work sheet

Change font shape in actual work sheet to bold, italic, underline

```
1 COM 'PROPERTY' xlshws 'UsedRange.Font.Bold' 1
1 COM 'PROPERTY' xlshws 'Range().Font.Bold'(c'A1:B6')1
1 COM 'PROPERTY' xlshws 'Range().Font.Italic'(c'A1:B6')1
```

Further Properties of a work sheet

Change font shape in actual work sheet to bold, italic, underline

```
1 COM 'PROPERTY' xlshws 'UsedRange.Font.Bold' 1
1 COM 'PROPERTY' xlshws 'Range().Font.Bold'(c'A1:B6')1
1 COM 'PROPERTY' xlshws 'Range().Font.Italic'(c'A1:B6')1
1 COM 'PROPERTY' xlshws 'Range().Font.Underline'(c'A1:B6')2 asingle
```

Further Properties of a work sheet

Change font shape in actual work sheet to bold, italic, underline

```
1 COM 'PROPERTY' xlshws 'UsedRange.Font.Bold' 1
1 COM 'PROPERTY' xlshws 'Range().Font.Bold'(c'A1:B6')1
1 COM 'PROPERTY' xlshws 'Range().Font.Italic'(c'A1:B6')1
1 COM 'PROPERTY' xlshws 'Range().Font.Underline'(c'A1:B6')2 asingle
1 COM 'PROPERTY' xlshws 'Range().Font.Underline'(c'A1:B6')5 adouble
```

Further Properties of a work sheet

Change font shape in actual work sheet to bold, italic, underline

```
1 COM 'PROPERTY' xlshws 'UsedRange.Font.Bold' 1
1 COM 'PROPERTY' xlshws 'Range().Font.Bold'(c'A1:B6')1
1 COM 'PROPERTY' xlshws 'Range().Font.Italic'(c'A1:B6')1
1 COM 'PROPERTY' xlshws 'Range().Font.Underline'(c'A1:B6')2 asingle
1 COM 'PROPERTY' xlshws 'Range().Font.Underline'(c'A1:B6')5 adouble
1 COM 'PROPERTY' xlshws 'Range().Font.Underline'(c'A1:B6')-4119 adouble bold
```

Further Properties of a work sheet

Change font shape in actual work sheet to bold, italic, underline

```
1 COM 'PROPERTY' xlshws 'UsedRange.Font.Bold' 1
1 COM 'PROPERTY' xlshws 'Range().Font.Bold'(c'A1:B6')1
1 COM 'PROPERTY' xlshws 'Range().Font.Italic'(c'A1:B6')1
1 COM 'PROPERTY' xlshws 'Range().Font.Underline'(c'A1:B6')2 asingle
1 COM 'PROPERTY' xlshws 'Range().Font.Underline'(c'A1:B6')5 adouble
1 COM 'PROPERTY' xlshws 'Range().Font.Underline'(c'A1:B6')-4119 adouble bold
1 COM 'PROPERTY' xlshws 'Range().Font.Underline'(c'A1:B6')-4142 ano
```

Further Properties of a work sheet

Change font shape in actual work sheet to bold, italic, underline

```
1 COM 'PROPERTY' xlshws 'UsedRange.Font.Bold' 1
1 COM 'PROPERTY' xlshws 'Range().Font.Bold'(c'A1:B6')1
1 COM 'PROPERTY' xlshws 'Range().Font.Italic'(c'A1:B6')1
1 COM 'PROPERTY' xlshws 'Range().Font.Underline'(c'A1:B6')2 asingle
1 COM 'PROPERTY' xlshws 'Range().Font.Underline'(c'A1:B6')5 adouble
1 COM 'PROPERTY' xlshws 'Range().Font.Underline'(c'A1:B6')-4119 adouble bold
1 COM 'PROPERTY' xlshws 'Range().Font.Underline'(c'A1:B6')-4142 ano
1 COM 'PROPERTY' xlshws 'Range().Font.Size'(c'A1:B6')28
```

Further Properties of a work sheet

Merge and format cells: Merge cells

```
1 COM 'PROPERTY' xlshws 'Range().merge'(<'A1:B2')
```

Further Properties of a work sheet

Merge and format cells: Unmerge cells

```
1 COM 'PROPERTY' xlshws 'Range().merge'(<'A1:B2')
1 COM 'PROPERTY' xlshws 'Range().unmerge'(<'A1:B2')
```

Further Properties of a work sheet

Merge and format cells:

Format cells of columns as date

```
1 COM 'PROPERTY' xlshws 'Range().merge'(⊂'A1:B2')
1 COM 'PROPERTY' xlshws 'Range().unmerge'(⊂'A1:B2')
1 COM 'PROPERTY' xlshws 'Range().NumberFormat'(⊂'F1:G1')( ,⊂'???,0' )
```

Format must be vector or matrix, not scalar

Further Properties of a work sheet

Merge and format cells:

Format cells as decimal number

```
1 COM 'PROPERTY' xlshws 'Range().merge'(⊂'A1:B2')
1 COM 'PROPERTY' xlshws 'Range().unmerge'(⊂'A1:B2')
1 COM 'PROPERTY' xlshws 'Range().NumberFormat'(⊂'F1:G1')( ,⊂'???,0' )
1 COM 'PROPERTY' xlshws 'Range().NumberFormat'(⊂'F1:G1')( ,⊂'Standard' )
```

Format must be vector or matrix, not scalar

Further Properties of a work sheet

Merge and format cells:

Format cells as standard format

```
1 COM 'PROPERTY' xlshws 'Range().merge'(⊂'A1:B2')
1 COM 'PROPERTY' xlshws 'Range().unmerge'(⊂'A1:B2')
1 COM 'PROPERTY' xlshws 'Range().NumberFormat'(⊂'F1:G1')( ,⊂'???,0' )
1 COM 'PROPERTY' xlshws 'Range().NumberFormat'(⊂'F1:G1')( ,⊂'Standard' )
1 COM 'PROPERTY' xlshws 'Range().NumberFormat'(⊂'F:G')( ,⊂'TT.MM.JJJJ' )
```

Format must be vector or matrix, not scalar

Conclusion, final remarks

Further information, khxls.atf:

https://www.kilsch.eu/InternetLehre: Arbeitsgebiete - APL

Conclusion, final remarks

Thank you for listening to my talk !

Conclusion, final remarks

Thank you for listening to my talk !

?? ??

Questions?

Demonstration

To run the Demonstration:

```
)Load Demon
⎕pw←150
)in khxls
Demon 'Kilsch20220506ComXLS.dem'
```

Sources

- khxls
- Demon
- Kilsch20220506ComXLS.dem