

APL64: Using XL

Contents

Overview	2
Syntax.....	3
 XL Actions.....	3
AddHyperlink	3
Hyperlink Types.....	3
AddHyperlink Example.....	4
AddWorksheet	5
AddWorksheet Example.....	6
Autofit	6
Autofit Specified Columns Example	7
Autofit All Columns Example	8
CreateWorkbook.....	9
CreateWorkbook Example	9
DeleteWorksheet	9
DeleteWorksheet Example	10
FromAPL.....	10
FromAPL Example #1	11
FromAPL Example #2	12
FromAPLToRange.....	14
FromAplToRange.....	15
GetA1AddrFromAbsR1C1Addr.....	15
GetA1AddrFromAbsR1C1Addr Example	16
GetA1ColNameFromColNum	16
GetA1ColNameFromColNum Example #1	17
GetA1ColNameFromColNum Example #2	17
GetR1C1AddrFromA1Addr.....	18
GetR1C1AddrFromA1Addr Example	18
GetUsedRange	19
GetUsedRange Example.....	19

GetWorksheetCount	20
GetWorkSheetCount Example	20
GetWorksheetNames.....	21
GetWorkSheetNames Example.....	21
Help	22
Help Example	23
NumberFormatRanges.....	23
NumberFormatRanges Example	24
RemoveHyperlink.....	25
RemoveHyperlink Example	25
RenameWorksheet	26
RenameWorksheet Example.....	26
SaveWorkSheetAsCsv	27
SaveWorksheetAsCsv with EN-US Decimal and Field Separators Example	28
SaveWorksheetAsCsv with non-EN-US Decimal and Field Separators Example.....	29
SortSelectedRange	31
SortSelectedRange Example	31
ToAPL	33
ToAPL Example.....	34
ToAPLFromRange.....	34
ToAPLFromRange Example	35
ToAPLFromCsv	36
ToAplFromCsv Example Csv with EN-US Delimiters and Date Information.....	37
ToAplFromCsv Example Csv with non-EN-US Delimiters and Date Information	38

Overview

The ☐XL system function in APL64 provides a convenient interface between APL64 and Microsoft Excel workbooks and worksheets. The APL64 ☐XL system function does not require the presence of Microsoft Excel on the target workstation. ☐XL operates on Microsoft Excel workbook files. ☐XL is available in the APL64 Developer version and the APL64 ‘runtime’ version. ☐XL is a cross-platform tool, so it is not designed to display a workbook.

Use ☐XL ‘?’ or ☐XL ‘Help’ in the APL64 Developer version to display a summary of ☐XL syntax.

Syntax

Result ← ☐xl Action ActionArgs

The Action argument is an APL64 text object indicate the action to be performed by ☐XL.

ActionArgs are the optional arguments necessary to carry out the action of ☐XL. At least one ActionArg is required.

Result is a two-element APL64 nested array. The first element of Result is a Boolean scalar indicating failure (0) or success (1). The second element of Result depends on the failure or success of the Action. If the Action fails, the second element of Result is an APL64 character vector describing the failure. If the Action succeeds, the second element of the Result depends on the selected Action.

Some ☐XL action names have short synonyms.

☐XL Actions

AddHyperlink

The AddHyperlink action will remove any previous hyperlink in a specified cell and add a new hyperlink.

(bRes aRes)←☐XL 'AddHyperlink' wbPath wsName cellId hIAddr hISubAddr hITip hIText

bRes: 0/Fail, 1/OK

aRes: ErrMsg (APL64 character vector)/Fail, "/OK

wbPath: Full path of the target Excel workbook. The workbook must exist.

wsName: Name of the worksheet in the target workbook. The worksheet must exist

cellId: Target cell for hyperlink, e.g. "A1"

hIAddr: url of hyperlink target or " if hISubAddr applies, e.g. https://myWebPage...

hISubAddr: hyperlink target location in workbook or " if hIAddr applies, e.g. "Sheet2!A1:D4"

hITip: Comment text which will be added to the specified cell or "

hIText: The text which will be displayed in the specified cell or "

Hyperlink Types

For all supported hyperlink types, APL64 cannot validate the hIAddr or hISubAddr arguments to the ☐XL AddHyperlink action, because the hyperlink can be created before the hIAddress or hISubAddr exists.

WebAddress

For this hyperlink type the hISubAddress must be ". The hIAddr must be text which the default browser on the end user's workstation will recognize as a valid [web url](#).

Examples:

<http://www.apl2000.com>

<http://APL2000.com>

www.apl2000.com

Email Address

For this hyperline type the hSubAddress must be ". The hAddress must be text, prefixed by 'mailto:', which the email software on the end user's workstation will recognize as a valid [email address](#).

Example: <mailto:support@apl2000.com>

Existing File

For this hyperline type the hSubAddress must be ". The hAddress must be text which is the name of an existing file accessible by the end user's credentials on the end user's workstation.

Examples:

c:\test\abc.xlsx

c:\test\abc.csv

c:\test\abc.xls

Place in the Current Workbook

For this hyperlink type the hAddr argument must be ". The hSubAddr must be a valid A1-format address in the workbook containing the hyperlink.

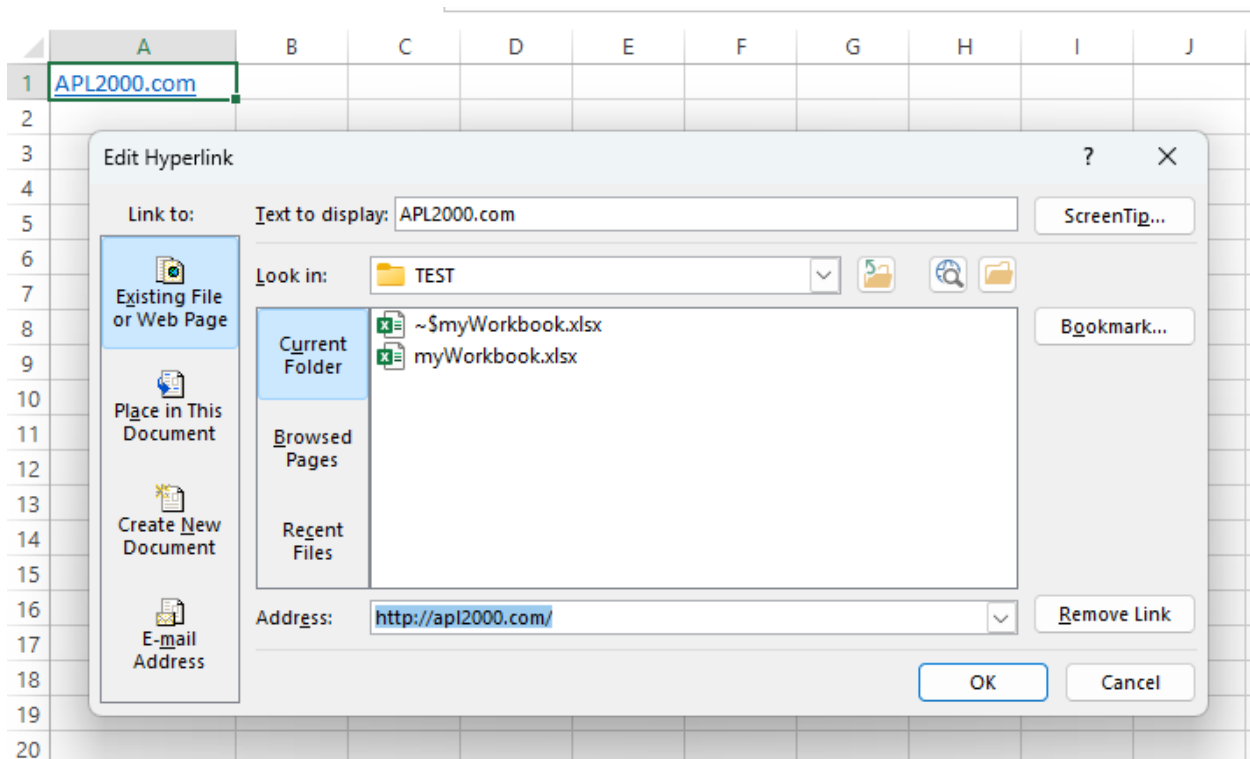
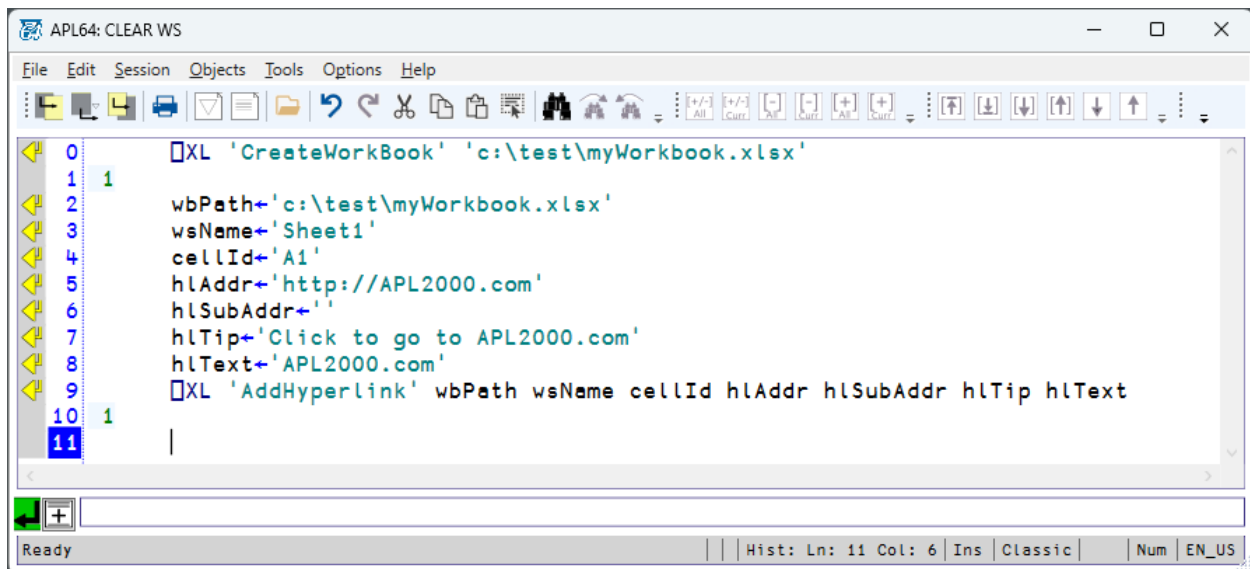
Examples:

G4: Cell in the worksheet containing the hyperlink

Sheet2!G4: Cell in the Sheet2 worksheet in the workbook containing the hyperlink

AddHyperlink Example

```
☐XL 'CreateWorkBook' 'c:\test\myWorkbook.xlsx'  
wbPath←'c:\test\myWorkbook.xlsx'  
wsName←'Sheet1'  
cellId←'A1'  
hAddr←'http://APL2000.com'  
hSubAddr←"  
hTip←'Click to go to APL2000.com'  
hText←'APL2000.com'  
☐XL 'AddHyperlink' wbPath wsName cellId hAddr hSubAddr hTip hText
```



AddWorksheet

The AddWorksheet action will add the specified worksheet to the specified Excel workbook.

(bRes aRes)←□XL 'AddWorksheet' wbPath wsName

bRes: 0/Fail, 1/OK

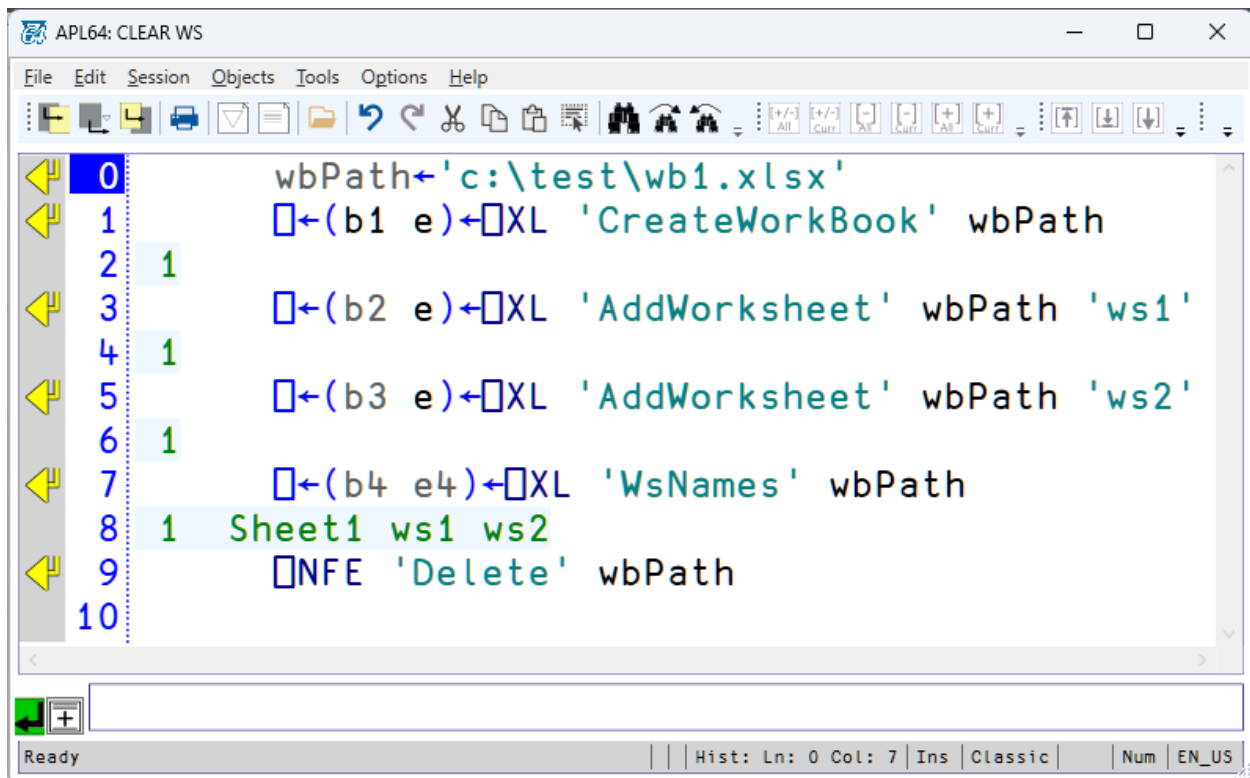
aRes: ErrMsg (APL64 character vector)/Fail, ''/OK

wbPath: Full path of the target Excel workbook. The workbook must exist.

wsName: Name of the worksheet in the target workbook.

AddWorksheet Example

```
wbPath←'c:\test\wb1.xlsx'  
⊞←(b1 e)←⊞XL 'CreateWorkBook' wbPath  
⊞←(b2 e)←⊞XL 'AddWorksheet' wbPath 'ws1'  
⊞←(b3 e)←⊞XL 'AddWorksheet' wbPath 'ws2'  
⊞←(b4 e4)←⊞XL 'WsNames' wbPath  
⊞NFE 'Delete' wbPath
```



Autofit

Autofit the specified ranges in an Excel worksheet

```
(bRes aRes)←⊞XL 'Autofit' wbPath wsName columnOrRow ranges
```

bRes: 0/Fail, 1/OK

aRes: ErrMsg (APL64 character vector)/Fail, ''/OK

wbPath: Full path of the target Excel workbook. If the workbook does not exist, it will be created.

wsName: Name of the worksheet in the target workbook. If the worksheet does not exist, it will be created.

columnOrRow: "Cols" or "Rows" (not case sensitive)

ranges specify the ranges for the Autofit action. If an element of the ranges is "", the Autofit action applies to all rows or columns in the worksheet, as specified by the columnOrRow argument.

Notes:

Ranges may be specified as character vectors, strings, or arrays (up to rank 2) of characters or strings

Autofit is valid only for rows or columns

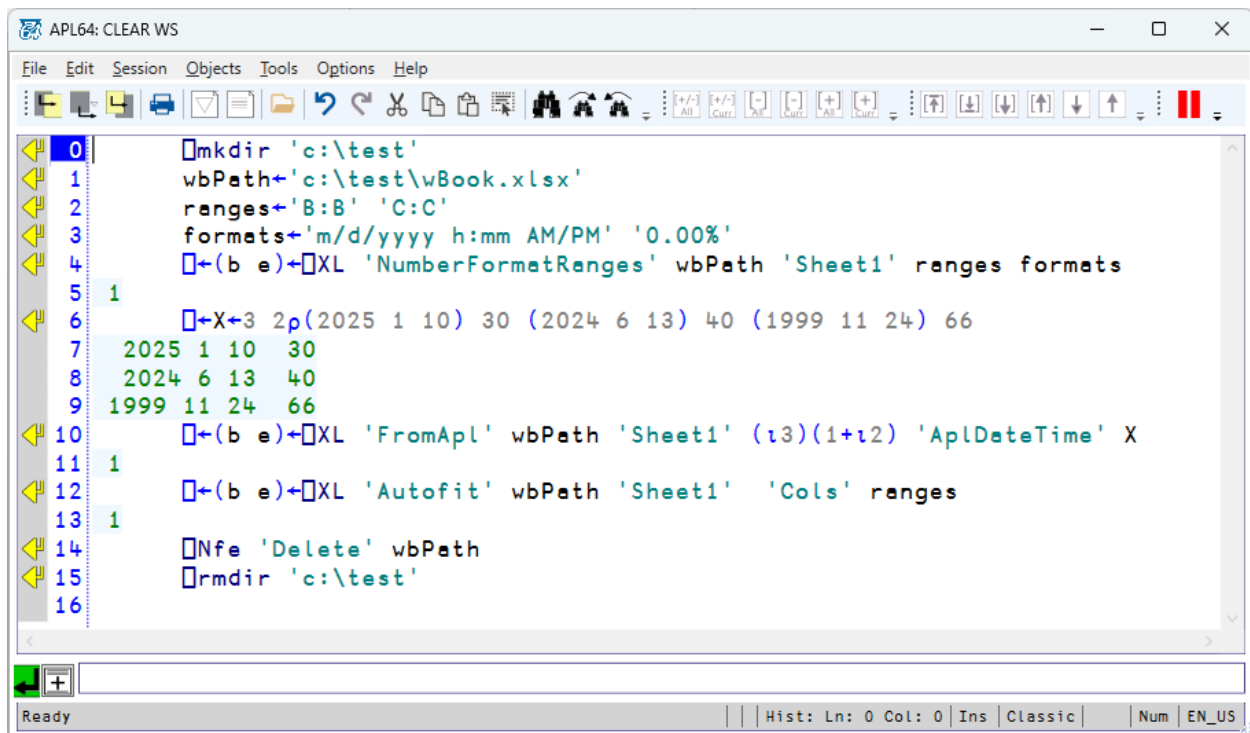
Autofit is applied to the entire range of each of the ranges

Typical row range: "7:9"

Typical column range: "B:C"

Autofit Specified Columns Example

```
mkdir 'c:\test'
wbPath←'c:\test\wBook.xlsx'
ranges←'B:B' 'C:C'
formats←'m/d/yyyy h:mm AM/PM' '0.00%'
←(b e)←XL 'NumberFormatRanges' wbPath 'Sheet1' ranges formats
←X←3 2p(2025 1 10) 30 (2024 6 13) 40 (1999 11 24) 66
←(b e)←XL 'FromApl' wbPath 'Sheet1' (i3)(1+i2) 'AplDateTime' X
←(b e)←XL 'Autofit' wbPath 'Sheet1' 'Cols' ranges
Nfe 'Delete' wbPath
rmdir 'c:\test'
```



Resulting worksheet:

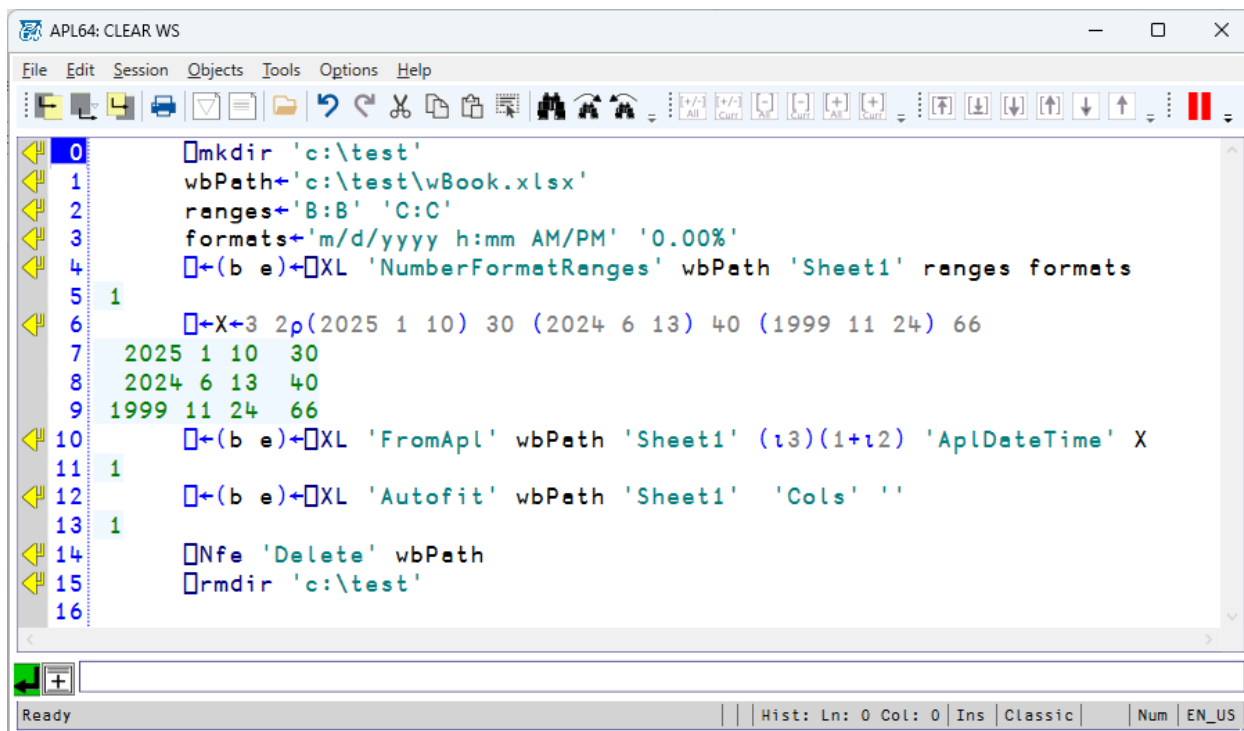
	A	B	C	D
1		1/10/2025 12:00 AM	3000.00%	
2		6/13/2024 12:00 AM	4000.00%	
3		11/24/1999 12:00 AM	6600.00%	
4				

Autofit All Columns Example

```

mkdir 'c:\test'
wbPath←'c:\test\wBook.xlsx'
ranges←'B:B' 'C:C'
formats←'m/d/yyyy h:mm AM/PM' '0.00%'
←(b e)←XL 'NumberFormatRanges' wbPath 'Sheet1' ranges formats
←X←3 2p(2025 1 10) 30 (2024 6 13) 40 (1999 11 24) 66
←(b e)←XL 'FromApl' wbPath 'Sheet1' (i3)(1+i2) 'AplDateTime' X
←(b e)←XL 'Autofit' wbPath 'Sheet1' 'Cols' ''
Nfe 'Delete' wbPath
rmdir 'c:\test'

```



1		1/10/2025 12:00 AM	3000.00%
2		6/13/2024 12:00 AM	4000.00%
3		11/24/1999 12:00 AM	6600.00%
4			

CreateWorkbook

The CreateWorkbook action will create a new Excel workbook of the type specified by the APL64 programmer-supplied workbook file extension, or if no extension is provided, the OpenXml (xlsx) format will be used. The new Excel workbook will contain one worksheet named 'Sheet1'.

(bRes aRes)←⎕XL 'FromApl' wbPath

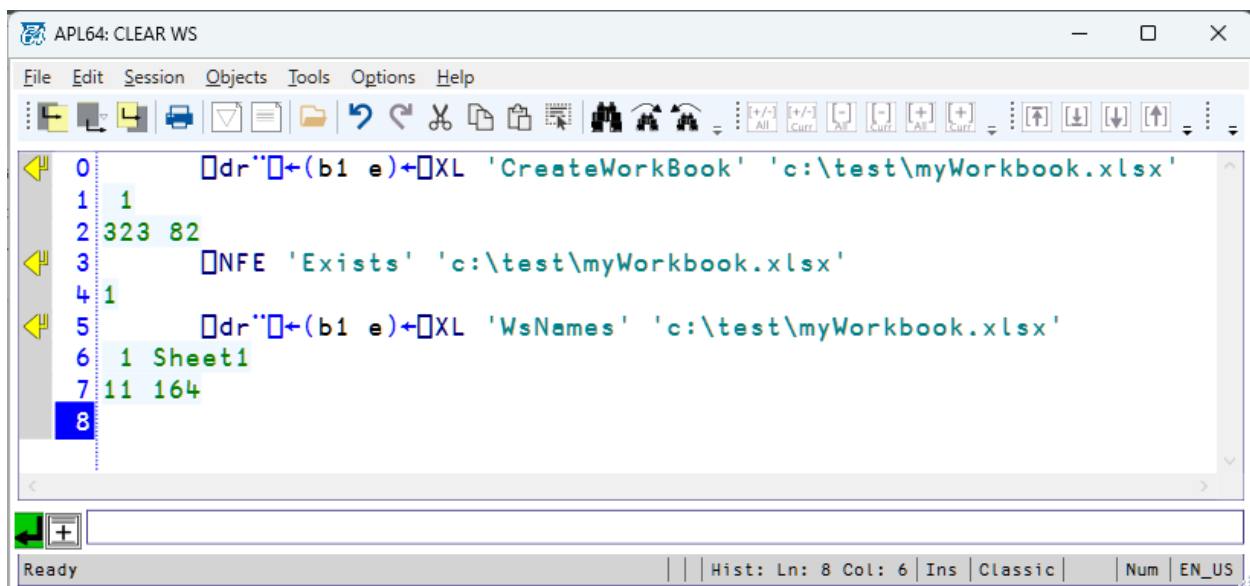
bRes: 0/Fail, 1/OK

aRes: ErrMsg (APL64 character vector)/Fail, ''/OK

wbPath: Full path of the target Excel workbook.

CreateWorkbook Example

```
⎕dr''←(b1 e)←⎕XL 'CreateWorkBook' 'c:\test\myWorkbook.xlsx'
⎕NFE 'Exists' 'c:\test\myWorkbook.xlsx'
⎕dr''←(b1 e)←⎕XL 'WsNames' 'c:\test\myWorkbook.xlsx'
```



DeleteWorksheet

The DeleteWorksheet action will delete the specified worksheet from the specified Excel workbook.

(bRes aRes)←⎕XL 'DeleteWorksheet' wbPath wsName

bRes: 0/Fail, 1/OK

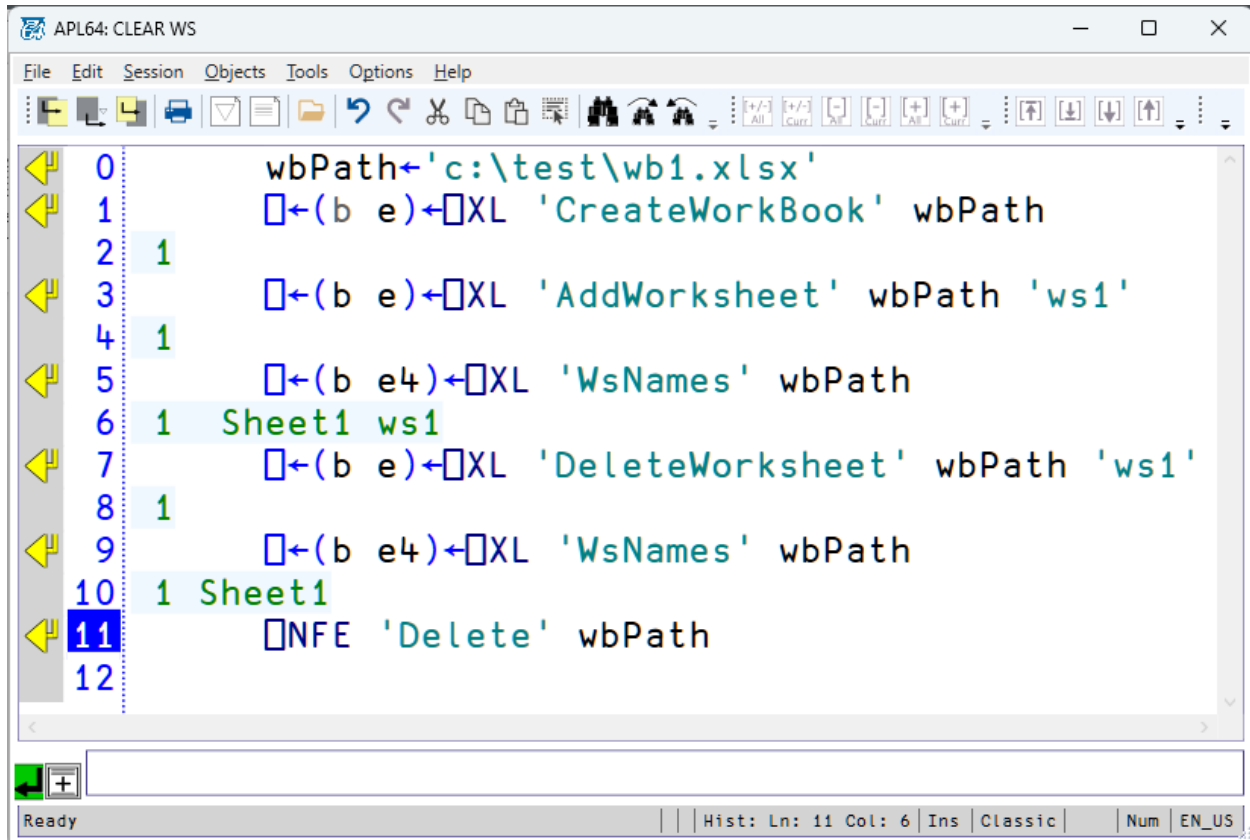
aRes: ErrMsg (APL64 character vector)/Fail, ''/OK

wbPath: Full path of the target Excel workbook. The workbook must exist.

wsName: Name of the worksheet in the target workbook. The worksheet must exist.

DeleteWorksheet Example

```
wbPath←'c:\test\wb1.xlsx'  
⎕←(b e)←⎕XL 'CreateWorkBook' wbPath  
⎕←(b e)←⎕XL 'AddWorksheet' wbPath 'ws1'  
⎕←(b e4)←⎕XL 'WsNames' wbPath  
⎕←(b e)←⎕XL 'DeleteWorksheet' wbPath 'ws1'  
⎕←(b e4)←⎕XL 'WsNames' wbPath  
⎕NFE 'Delete' wbPath
```



FromAPL

The FromAPL action will insert the elements of APL variable into an Excel worksheet. The number of elements in the APL variable must match the number of target cells in the Excel worksheet. The elements of the APL variable must be simple scalars.

```
(bRes aRes)←⎕XL 'FromApl' wbPath wsName inclRows inclCols dateTimeConv sourceAplValues
```

bRes: 0/Fail, 1/OK

aRes: ErrMsg (APL64 character vector)/Fail, ''/OK

wbPath: Full path of the target Excel workbook. If the workbook does not exist, it will be created.

wsName: Name of the worksheet in the target workbook. If the worksheet does not exist, it will be created.

The inclRows and inclCols arguments specify a worksheet range which will be the target for the APL64 source data. Order and ☐IO are significant.

- inclRows: APL64 integer vector of the rows of the target worksheet range.
- inclCols: APL64 integer vector of the columns of the target worksheet range.

dateTimeConv: Two options are available:

- 'AplDateTime': 7-element APL ☐TS date time will be converted to XL DateTime
- 'XlDateTime' : Floating point value relative to midnight 12/30/1899 (effectively no conversion)

sourceAplValues: The source APL64 elements are accessed in ravel order.

FromAPL Example #1

```
☐mkdir 'c:\test'  
☐← X←«Abc» 1.234 0 456 ☐OVER «qrst» ^5.33 1 ^334  
wbPath←'c:\test\wBook.xlsx'  
☐← (b a)←☐XL 'FromApl' wbPath 'Sheet1' (i2) (i4) 'XlDateTime' X  
☐← (b a)←☐XL 'ToApl' wbPath 'Sheet1' (i2) (i4) 'AplString' 'XlDateTime' '!'Empty' '!'Error'  
☐NFE 'DELETE' wbPath  
☐rmdir 'c:\test'  
X≡a  
☐IO
```

APL64: CLEAR WS

File Edit Session Objects Tools Options Help

0 `Ⓜmkdir 'c:\test'`
 1 `Ⓜ← X←«Abc» 1.234 0 456 ⓂOVER «qrst» -5.33 1 -334`
 2 `Abc 1.234 0 456`
 3 `qrst -5.330 1 -334`
 4 `wbPath←'wBook.xlsx'`
 5 `Ⓜ← (b a)←ⓂXL 'FromApl' wbPath 'Sheet1' (12) (14) 'XlDateTime' X`
 6 `1`
 7 `Ⓜ← (b a)←ⓂXL 'ToApl' wbPath 'Sheet1' (12) (14) 'AplString' 'XlDateTime' '!Empty' '!Error'`
 8 `1 Abc 1.234 0 456`
 9 `qrst -5.330 1 -334`
 10 `ⓂNFE 'DELETE' wbPath`
 11 `Ⓜrmdir 'c:\test'`
 12 `X←a`
 13 `1`
 14 `ⓂIO`
 15 `1`
 16

Ready | Hist: Ln: 16 Col: 6 | Ins | Classic | Num | EN_US

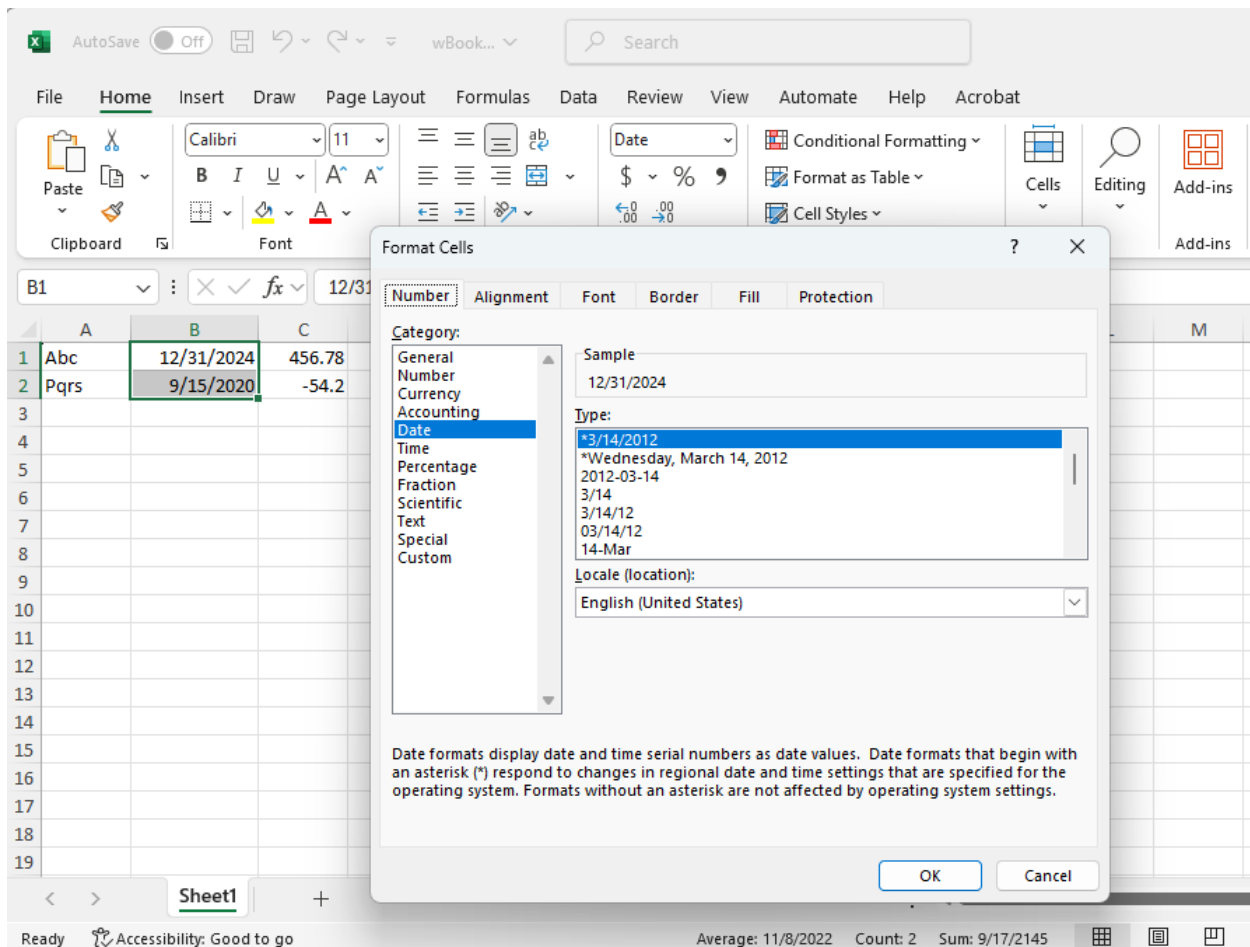
	A	B	C	D	E	F	G
1	Abc	1.234	0	456			
2	qrst	-5.33	1	-334			
3							
4							
5							

FromAPL Example #2

```

☐ mkdir 'c:\test'
wbPath←'c:\test\wBook.xlsx'
☐ ←(b a)←☐ XL 'NumberFormatRanges' wbPath 'Sheet1' 'B1:B2' 'm/d/yyyy'
☐ ←X←2 3p'Abc' (7↑2024 12 31) 456.78 'Pqrs' (7↑2020 9 15) ~54.2
☐ ←(b a)←☐ XL 'FromApl' wbPath 'Sheet1' (ι2) (ι3) 'AplDateTime' X
☐ ←(b a)←☐ XL 'ToApl' wbPath 'Sheet1' (ι2) (ι3) 'AplChar' 'AplDateTime' '!'Empty' '!'Error'
X≡a
☐ ←(b z)←☐ XL 'ToApl' wbPath 'Sheet1' (ι2) (ι3) 'AplChar' 'XlDateTime' '!'Empty' '!'Error'
☐ NFE 'DELETE' wbPath
☐ rmdir 'c:\test'
☐ io

```

FromAPLToRange

The FromAPLToRange action will insert the elements of an APL64 variable into an Excel worksheet. The number of elements in the APL variable must match the number of target cells in the Excel worksheet. The elements of the APL variable must be simple scalars.

(bRes aRes) ← ☐ XL 'FromAPLToRange' wbPath wsName range dateTimeConv sourceApIValues

bRes: 0/Fail, 1/OK

aRes: ErrMsg (APL64 character vector)/Fail, ''/OK

wbPath: Full path of the target Excel workbook. If the workbook does not exist, it will be created.

wsName: Name of the worksheet in the target workbook. If worksheet does not exist, it will be created.

range: Excel A1-format specification of target cells in the worksheet, e.g. 'A1', or 'B5:G23'

dateTimeConv: Two options are available:

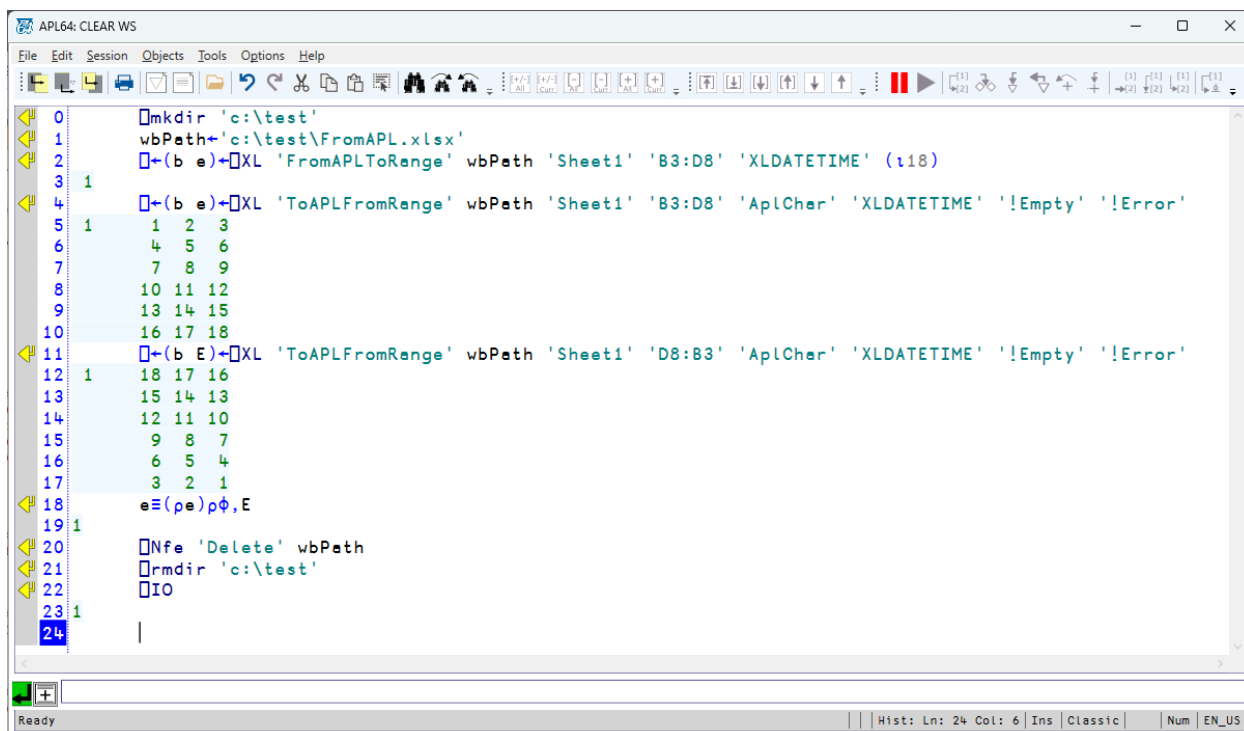
- 'ApIDateTime': 7-element APL ☐ TS date time will be converted to XL DateTime
- 'XIDatetime': Floating point value relative to midnight 12/30/1899 (effectively no conversion)

sourceApIValues: The source APL64 elements are accessed in ravel order.

FromAplToRange

```
☐ mkdir 'c:\test'
wbPath←'c:\test\FromAPL.xlsx'
☐ ←(b e)←☐ XL 'FromAPLToRange' wbPath 'Sheet1' 'B3:D8' 'XLDATE TIME' (118)
☐ ←(b e)←☐ XL 'ToAPLFromRange' wbPath 'Sheet1' 'B3:D8' 'AplChar' 'XLDATE TIME' '!'Empty' '!'Error'
☐ ←(b E)←☐ XL 'ToAPLFromRange' wbPath 'Sheet1' 'D8:B3' 'AplChar' 'XLDATE TIME' '!'Empty' '!'Error'
e≡(pe)p⊕,E
☐ Nfe 'Delete' wbPath
☐ rmdir 'c:\test'
☐ IO
```

The order of cells specifying the target range is significant.



GetA1AddrFromAbsR1C1Addr

The GetA1AddrFromAbsR1C1Addr action will convert an Excel row number and an Excel column number to an Excel A1-format absolute cell address.

```
(bRes aRes) ← ☐ XL 'GetA1AddrFromAbsR1C1Addr' r1 c1 [wbFmt]
```

Or synonymous action:

```
(bRes aRes) ← ☐ XL 'R1C1>A1' r1 c1 [wbFmt]
```

r1: Row# of source absolute R1C1-format address

c1: Column# of source absolute R1C1-format address

wbFmt: Optional target Excel format: 'Excel8' or 'OpenWb'

When the wbFmt argument is provided, the row and column numbers will be validated based on the target Excel format .

☐ IO is significant.

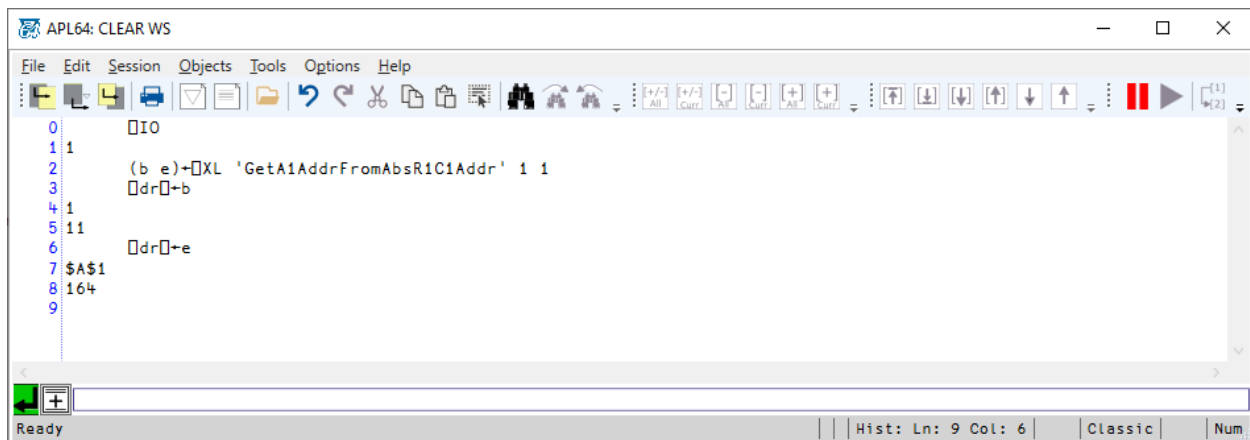
bRes: 0/Fail, 1/OK

aRes: Fail: ErrMsg (APL64 character vector)/Fail

aRes: OK: APL64 string containing the A1 address, e.g. '\$A\$1'

GetA1AddrFromAbsR1C1Addr Example

```
☐ IO
☐ ←(b e)←☐XL 'GetA1AddrFromAbsR1C1Addr' 1 1
☐ dr e
```



GetA1ColNameFromColNum

The GetA1ColNameFromColNum action will convert an Excel column number to an Excel column name.

(bRes aRes) ← ☐XL 'GetA1ColNameFromColNum' colNum [wbFmt]

Or synonymous action:

(bRes aRes) ← ☐XL 'COL#>COLNAME' colNum [wbFmt]

colNum: Integer scalar representing an Excel column number.

wbFmt: Optional target Excel format: 'Excel8' or 'OpenWb'

When the wbFmt argument is provided, the row and column numbers will be validated based on the target Excel format .

☐ IO is significant.

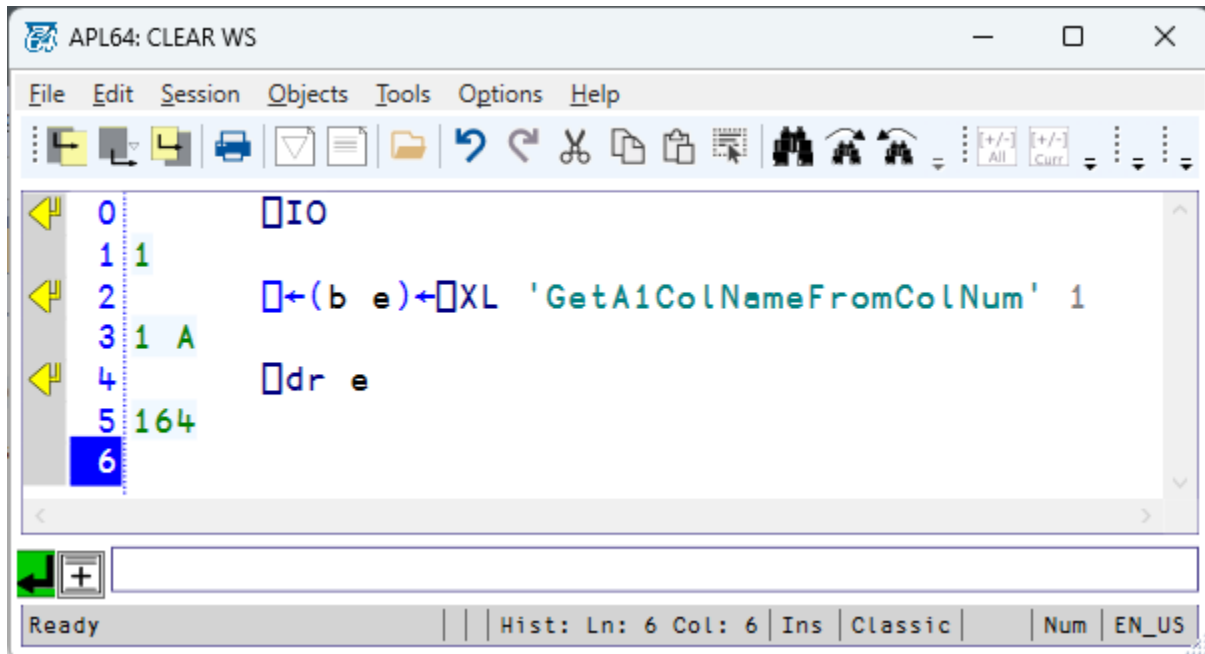
bRes: 0/Fail, 1/OK

aRes: Fail: ErrMsg (APL64 character vector)/Fail

aRes: OK: APL64 string containing Excel column name

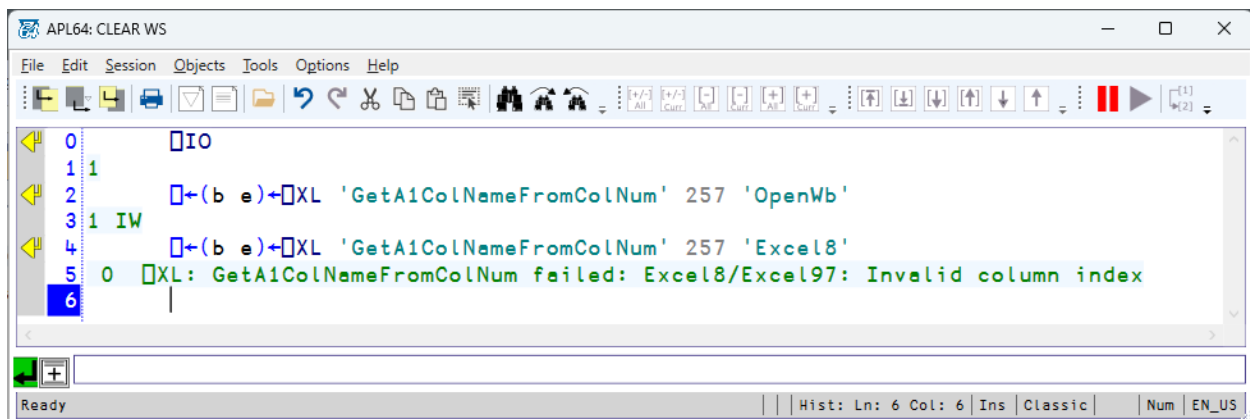
GetA1ColNameFromColNum Example #1

```
⎕IO
⎕←(b e)←⎕XL 'GetA1ColNameFromColNum' 1
⎕dr e
```



GetA1ColNameFromColNum Example #2

```
⎕IO
⎕←(b e)←⎕XL 'GetA1ColNameFromColNum' 257 'OpenWb'
⎕←(b e)←⎕XL 'GetA1ColNameFromColNum' 257 'Excel8'
```



GetR1C1AddrFromA1Addr

The GetR1C1AddrFromA1Addr action will convert an Excel A1-format address to an Excel row number and an Excel column number.

```
(bRes aRes) ← ☐XL 'GetR1C1AddrFromA1Addr' a1Addr [wbFmt]
```

Or synonymous action:

```
(bRes aRes) ← ☐XL 'A1>R1C1' a1Addr [wbFmt]
```

a1Addr: APL64 text object containing the A1-format address, e.g. '\$A\$1', '\$A\$1:\$B\$1', 'A1', 'A1:B1'

wbFmt: Optional target Excel format: 'Excel8' or 'OpenWb'

When the wbFmt argument is provided, the row and column numbers will be validated based on the target Excel format.

☐IO is significant.

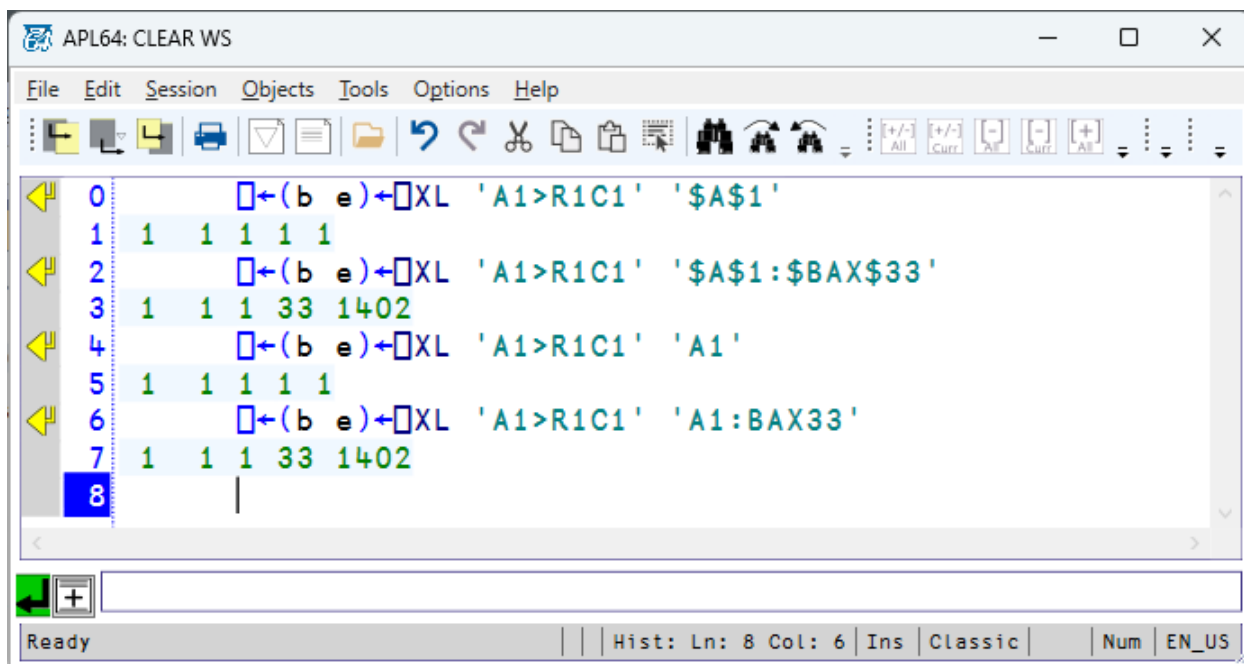
bRes: 0/Fail, 1/OK

aRes: Fail: ErrMsg (APL64 character vector)/Fail

aRes: OK: APL64 integer vector containing (row1 col1 row2 col2)

GetR1C1AddrFromA1Addr Example

```
☐←(b e)←☐XL 'A1>R1C1' '$A$1'  
☐←(b e)←☐XL 'A1>R1C1' '$A$1:$BAX$33'  
☐←(b e)←☐XL 'A1>R1C1' 'A1'  
☐←(b e)←☐XL 'A1>R1C1' 'A1:BAX33'
```



GetUsedRange

The GetUsedRange action will return the Excel used range of a worksheet in an Excel workbook file.

(bRes aRes) ← ☐XL 'GetUsedRange' wbPath wsName [ignoreEmptyCells]

Or synonymous action:

(bRes aRes) ← ☐XL 'UsedRange' wbPath wsName

wbPath is the path to the Excel workbook

wsName is the name of the worksheet in the workbook

ignoreEmptyCells (default value 0) is an optional Boolean argument indicating if only cells with values are to be considered in the used range of the specified worksheet. For example, a cell with a custom number format, but no value, will not be considered in the used range when ignoreEmptyCells is one, whereas if ignoreEmptyCells is zero, the cells with no value, but custom number formatting will be considered in the result.

If AutoFilters is enabled for the worksheet, hidden rows may not be considered in the result.

☐IO is significant.

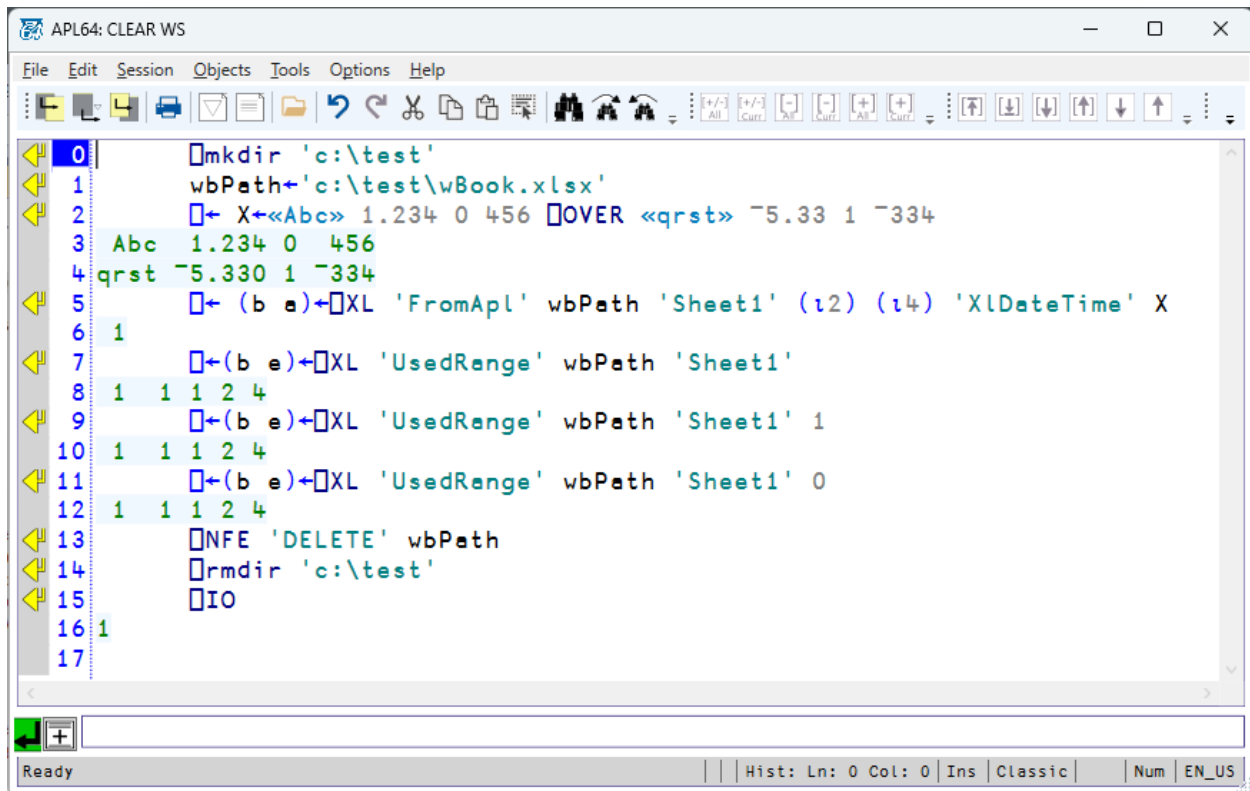
bRes: 0/Fail, 1/OK

aRes: Fail: ErrMsg (APL64 character vector)/Fail

aRes: OK: APL64 integer vector containing (row1 col1 row2 col2) of the used range in the worksheet

GetUsedRange Example

```
☐mkdir 'c:\test'
wbPath←'c:\test\wBook.xlsx'
☐← X←«Abc» 1.234 0 456 ☐OVER «qrst» 5.33 1 334
☐← (b a)←☐XL 'FromApl' wbPath 'Sheet1' (i2) (i4) 'XIDateTime' X
☐←(b e)←☐XL 'UsedRange' wbPath 'Sheet1'
☐←(b e)←☐XL 'UsedRange' wbPath 'Sheet1' 1
☐←(b e)←☐XL 'UsedRange' wbPath 'Sheet1' 0
☐NFE 'DELETE' wbPath
☐rmdir 'c:\test'
☐IO
```



GetWorksheetCount

The GetWorkSheetCount action will return the number of worksheets in the specified Excel workbook file.

(bRes aRes) ← ☐XL 'GetWorkSheetCount' wbPath

Or synonymous action:

(bRes aRes) ← ☐XL 'WsCount' wbPath

wbPath is the path to the Excel workbook

bRes: 0/Fail, 1/OK

aRes: Fail: ErrMsg (APL64 character vector)/Fail

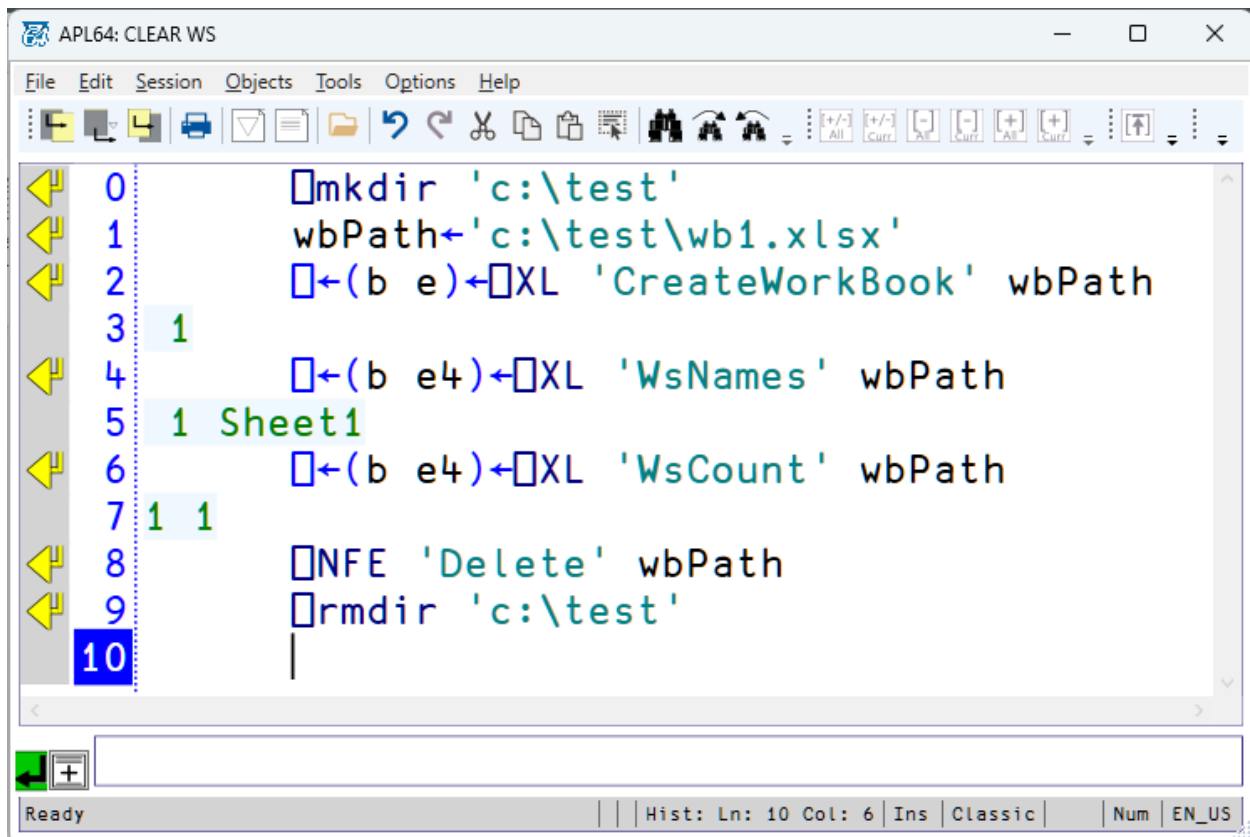
aRes: OK: Int32 value

GetWorkSheetCount Example

```

☐ mkdir 'c:\test'
wbPath←'c:\test\wb1.xlsx'
☐ ←(b e)←☐XL 'CreateWorkBook' wbPath
☐ ←(b e4)←☐XL 'WsNames' wbPath
☐ ←(b e4)←☐XL 'WsCount' wbPath
☐ NFE 'Delete' wbPath
☐ rmdir 'c:\test'

```



GetWorksheetNames

The GetWorkSheetNames action will return the worksheet names of worksheets in an Excel workbook file.

```
(bRes aRes) ← ⎕XL 'GetWorkSheetNames' wbPath
```

Or synonymous action:

```
(bRes aRes) ← ⎕XL 'WsNames' wbPath
```

wbPath is the path to the Excel workbook

bRes: 0/Fail, 1/OK

aRes: Fail: ErrMsg (APL64 character vector)/Fail

aRes: OK: Vector of APL64 strings containing the worksheet names. An Excel worksheet name may contain Unicode characters not in ⎕AV, so the 'GetWorksheetNames' action returns the APL64 string data type.

GetWorksheetNames Example

```

⎕mkdir 'c:\test'
wbPath←'c:\test\wBook.xlsx'
⎕← X←2 4πi8
⎕← (b a)←⎕XL 'FromApl' wbPath 'Sheet1' (i2) (i4) 'XIDateTime' X
⎕← (b a)←⎕XL 'FromApl' wbPath 'Sheet2' (i2) (i4) 'XIDateTime' X

```

```

⎕←(b e)←⎕XL 'WsNames' wbPath
⎕dr e
⎕NFE 'DELETE' wbPath
⎕rmdir 'c:\test'
⎕IO

```

```

APL64: CLEAR WS
File Edit Session Objects Tools Options Help
⎕mkdir 'c:\test'
wbPath←'c:\test\wBook.xlsx'
⎕← X←2 4p18
1 2 3 4
5 6 7 8
⎕← (b e)←⎕XL 'FromApl' wbPath 'Sheet1' (12) (14) 'XLDateTime' X
1
⎕← (b e)←⎕XL 'FromApl' wbPath 'Sheet2' (12) (14) 'XLDateTime' X
1
⎕←(b e)←⎕XL 'WsNames' wbPath
1 Sheet1 Sheet2
⎕dr e
164
⎕NFE 'DELETE' wbPath
⎕rmdir 'c:\test'
⎕IO
1

```

Ready | Hist: Ln: 0 Col: 0 Ins | Classic | Num | EN_US

Help

The Help or (?) action returns a summary of the ☐XL documentation

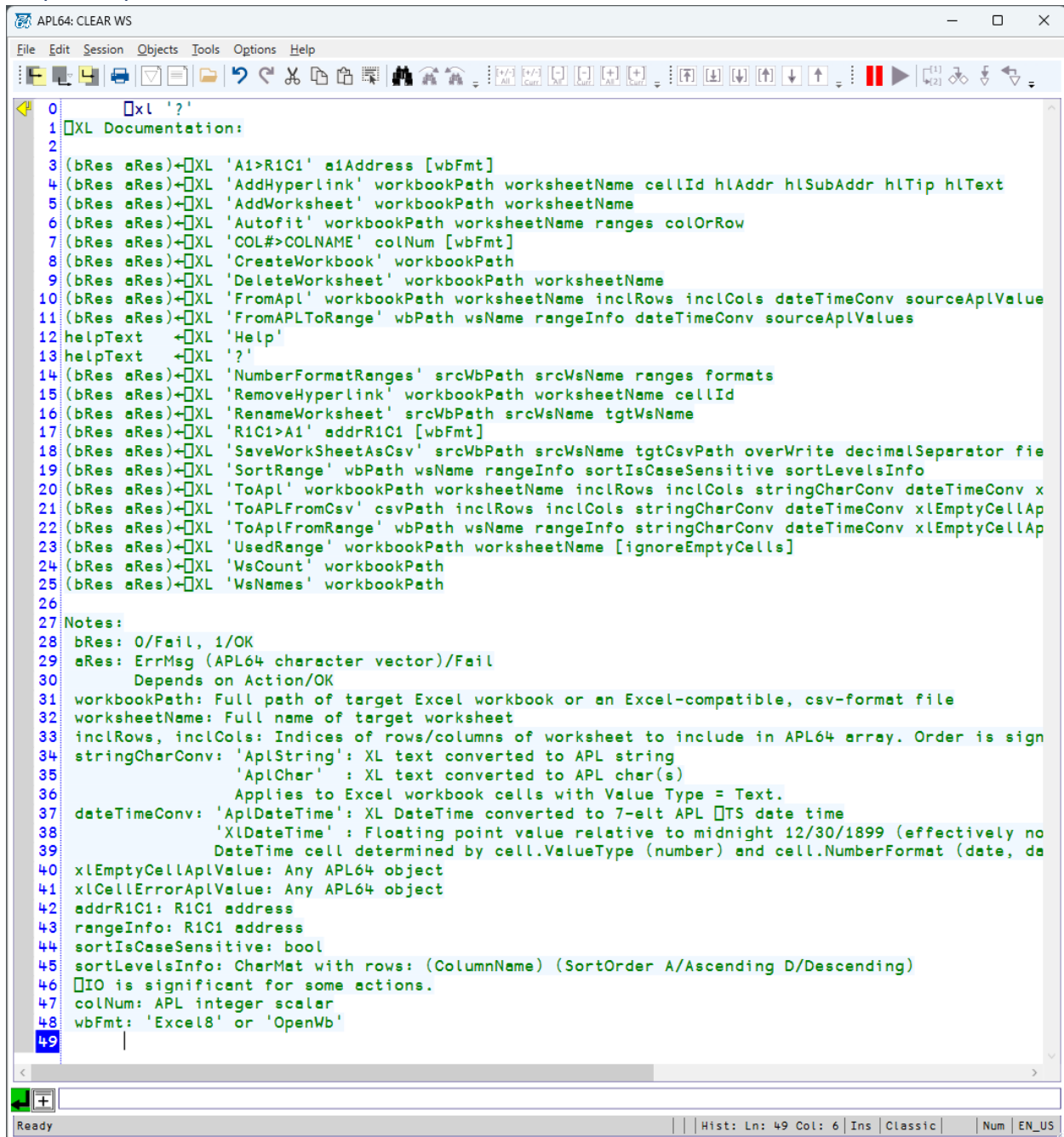
(bRes aRes) ← ☐XL 'Help' or (bRes aRes) ← ☐XL '?'

bRes: 0/Fail, 1/OK

aRes: Fail: ErrMsg (APL64 character vector)/Fail

aRes: OK: APL64 character vector containing the ☐XL documentation summary

Help Example



```
0  ?
1  XL Documentation:
2
3  (bRes aRes)←XL 'A1>R1C1' a1Address [wbFmt]
4  (bRes aRes)←XL 'AddHyperlink' workbookPath worksheetName cellId hlAddr hlSubAddr hlTip hlText
5  (bRes aRes)←XL 'AddWorksheet' workbookPath worksheetName
6  (bRes aRes)←XL 'Autofit' workbookPath worksheetName ranges colorRow
7  (bRes aRes)←XL 'COL#>COLNAME' colNum [wbFmt]
8  (bRes aRes)←XL 'CreateWorkbook' workbookPath
9  (bRes aRes)←XL 'DeleteWorksheet' workbookPath worksheetName
10 (bRes aRes)←XL 'FromApl' workbookPath worksheetName inclRows inclCols dateTimeConv sourceAplValue
11 (bRes aRes)←XL 'FromAPLToRange' wbPath wsName rangeInfo dateTimeConv sourceAplValues
12 helpText ←XL 'Help'
13 helpText ←XL '?'
14 (bRes aRes)←XL 'NumberFormatRanges' srcWbPath srcWsName ranges formats
15 (bRes aRes)←XL 'RemoveHyperlink' workbookPath worksheetName cellId
16 (bRes aRes)←XL 'RenameWorksheet' srcWbPath srcWsName tgtWsName
17 (bRes aRes)←XL 'R1C1>A1' addrR1C1 [wbFmt]
18 (bRes aRes)←XL 'SaveWorksheetAsCsv' srcWbPath srcWsName tgtCsvPath overWrite decimalSeparator file
19 (bRes aRes)←XL 'SortRange' wbPath wsName rangeInfo sortIsCaseSensitive sortLevelsInfo
20 (bRes aRes)←XL 'ToApl' workbookPath worksheetName inclRows inclCols stringCharConv dateTimeConv x
21 (bRes aRes)←XL 'ToAPLFromCsv' csvPath inclRows inclCols stringCharConv dateTimeConv xlEmptyCellAp
22 (bRes aRes)←XL 'ToAplFromRange' wbPath wsName rangeInfo stringCharConv dateTimeConv xlEmptyCellAp
23 (bRes aRes)←XL 'UsedRange' workbookPath worksheetName [ignoreEmptyCells]
24 (bRes aRes)←XL 'WsCount' workbookPath
25 (bRes aRes)←XL 'WsNames' workbookPath
26
27 Notes:
28 bRes: 0/Fail, 1/OK
29 aRes: ErrMsg (APL64 character vector)/Fail
30      Depends on Action/OK
31 workbookPath: Full path of target Excel workbook or an Excel-compatible, csv-format file
32 worksheetName: Full name of target worksheet
33 inclRows, inclCols: Indices of rows/columns of worksheet to include in APL64 array. Order is sign
34 stringCharConv: 'AplString': XL text converted to APL string
35                  'AplChar' : XL text converted to APL char(s)
36                  Applies to Excel workbook cells with Value Type = Text.
37 dateTimeConv: 'AplDateTime': XL DateTime converted to 7-elt APL TS date time
38                'XlDateTime' : Floating point value relative to midnight 12/30/1899 (effectively no
39                DateTime cell determined by cell.ValueType (number) and cell.NumberFormat (date, da
40 xlEmptyCellAplValue: Any APL64 object
41 xlCellErrorAplValue: Any APL64 object
42 addrR1C1: R1C1 address
43 rangeInfo: R1C1 address
44 sortIsCaseSensitive: bool
45 sortLevelsInfo: CharMat with rows: (ColumnName) (SortOrder A/Ascending D/Descending)
46 IO is significant for some actions.
47 colNum: APL integer scalar
48 wbFmt: 'Excel8' or 'OpenWb'
49
```

NumberFormatRanges

The NumberFormatRanges action will access the specified Excel source workbook worksheet, set the number formats for the specified ranges with the specified formats, and save the workbook.

(bRes aRes)←XL 'NumberFormatRanges' wbPath wsName ranges formats

srcWbPath: APL64 text object containing the path to the specified existing Excel workbook. If the workbook does not exist, it will be created.

srcWsName: APL64 text object containing the name of the specified existing worksheet in the source workbook. If the worksheet does not exist, it will be created.

ranges specify the ranges for the NumberFormatRanges action.

formats specify the number formats for the NumberFormatRanges action.

Notes:

Ranges and Formats may be specified as character vectors, strings, or arrays (up to rank 2) of characters or strings

The number of ranges must equal the number of numberformats

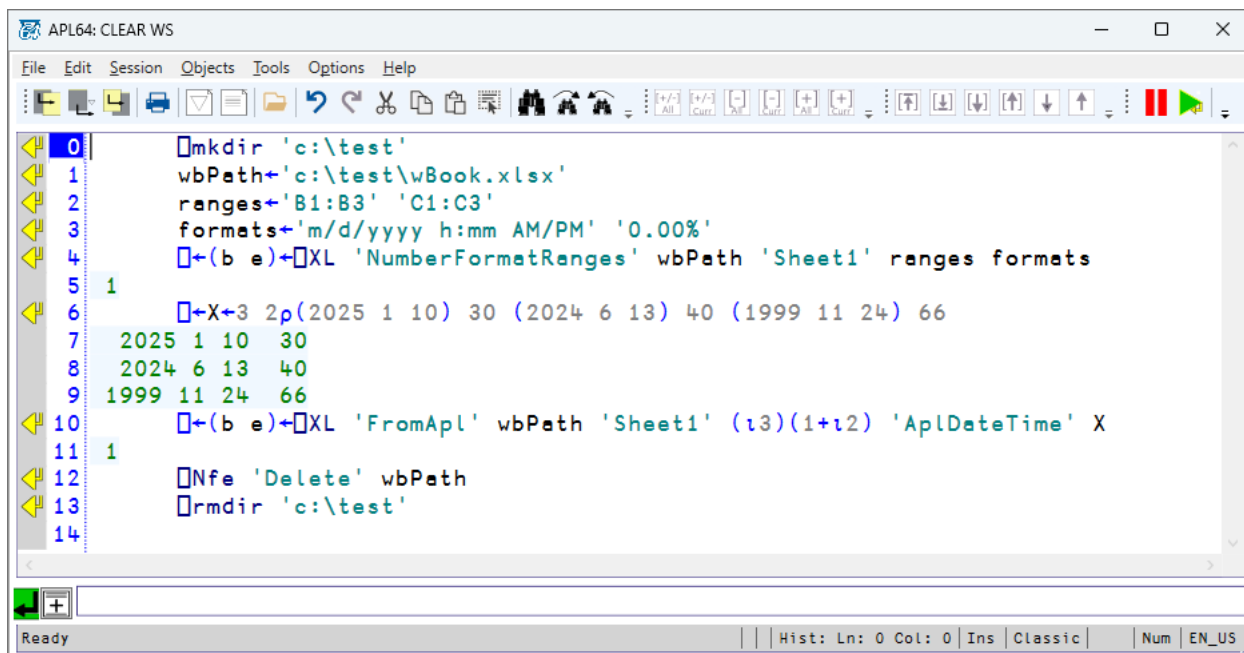
Each format is applied to the entire corresponding range

Typical ranges: "B3", "B1:F23"

Typical number formats: "General", "@", "0", "m/d/yyyy", "h:mm AM/PM", "0.00%", "0.00E+00"

NumberFormatRanges Example

```
☐ mkdir 'c:\test'
wbPath←'c:\test\wBook.xlsx'
ranges←'B1:B3' 'C1:C3'
formats←'m/d/yyyy h:mm AM/PM' '0.00%'
☐ ←(b e)←☐ XL 'NumberFormatRanges' wbPath 'Sheet1' ranges formats
☐ ←X←3 2p(2025 1 10) 30 (2024 6 13) 40 (1999 11 24) 66
☐ ←(b e)←☐ XL 'FromApl' wbPath 'Sheet1' (i3)(1+i2) 'AplDateTime' X
☐ Nfe 'Delete' wbPath
☐ rmdir 'c:\test'
```



Resulting Excel worksheet:

	A	B	C	D	E	F	G	H	I	J	K
1		1/10/2025 12:00 AM	3000.00%								
2		6/13/2024 12:00 AM	4000.00%								
3		11/24/1999 12:00 AM	6600.00%								
4											

RemoveHyperlink

The RemoveHyperlink action will remove the hyperlink from a cell, or remove the hyperlinks from a worksheet or workbook.

(bRes aRes)←☐XL 'RemoveHyperlink' workbookPath worksheetName [cellId]

bRes: 0/Fail, 1/OK

aRes: ErrMsg (APL64 character vector)/Fail, "/OK

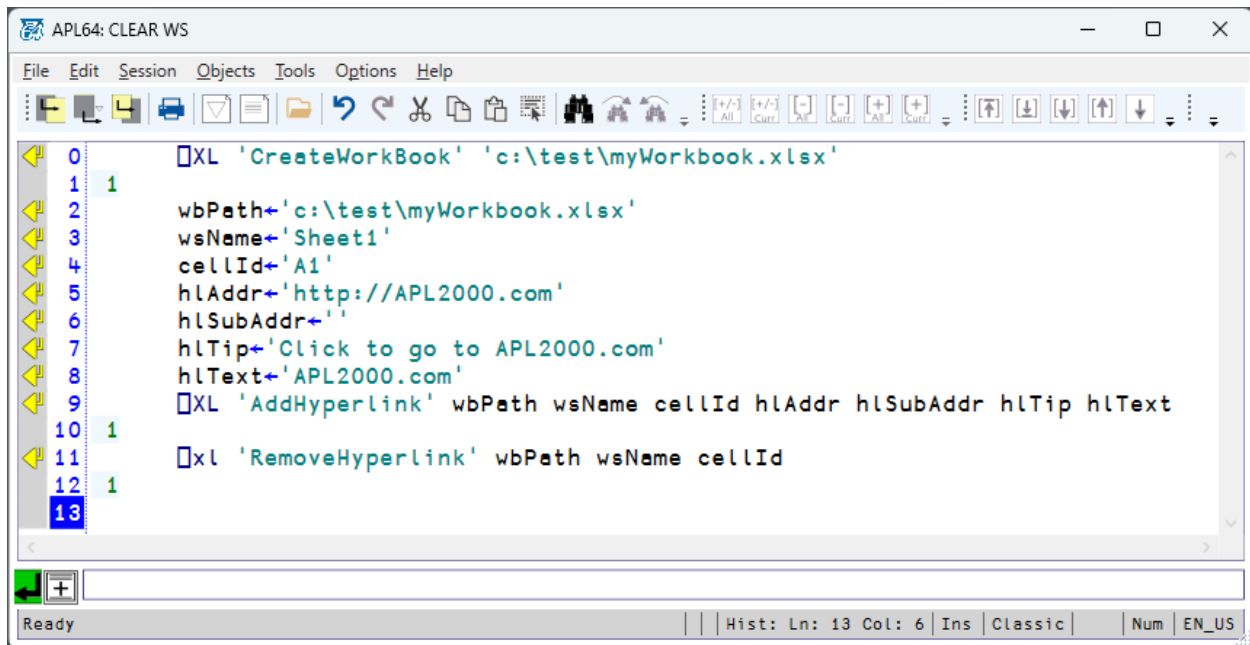
wbPath: Full path of the target Excel workbook. The workbook must exist.

wsName: Name of the worksheet in the target workbook. The worksheet must exist

cellId: Target cell for hyperlink, e.g. "A1". If the cellId is "", the hyperlinks in the worksheet will be deleted.

RemoveHyperlink Example

```
☐XL 'CreateWorkBook' 'c:\test\myWorkbook.xlsx'  
wbPath←'c:\test\myWorkbook.xlsx'  
wsName←'Sheet1'  
cellId←'A1'  
hlAddr←'http://APL2000.com'  
hlSubAddr←''  
hlTip←'Click to go to APL2000.com'  
hlText←'APL2000.com'  
☐XL 'AddHyperlink' wbPath wsName cellId hlAddr hlSubAddr hlTip hlText  
☐xl 'RemoveHyperlink' wbPath wsName cellId
```



RenameWorksheet

The RenameWorksheet action will rename the specified worksheet with the specified name.

(bRes aRes)←`⊞XL 'SaveWorkSheetAsCsv' srcWbPath srcWsName tgtCsvPath overwrite decimalSeparator fieldSeparator`

or the synonymous action:

(bRes aRes)←`⊞XL 'RenameWorksheet' srcWbPath srcWsName tgtWsName`

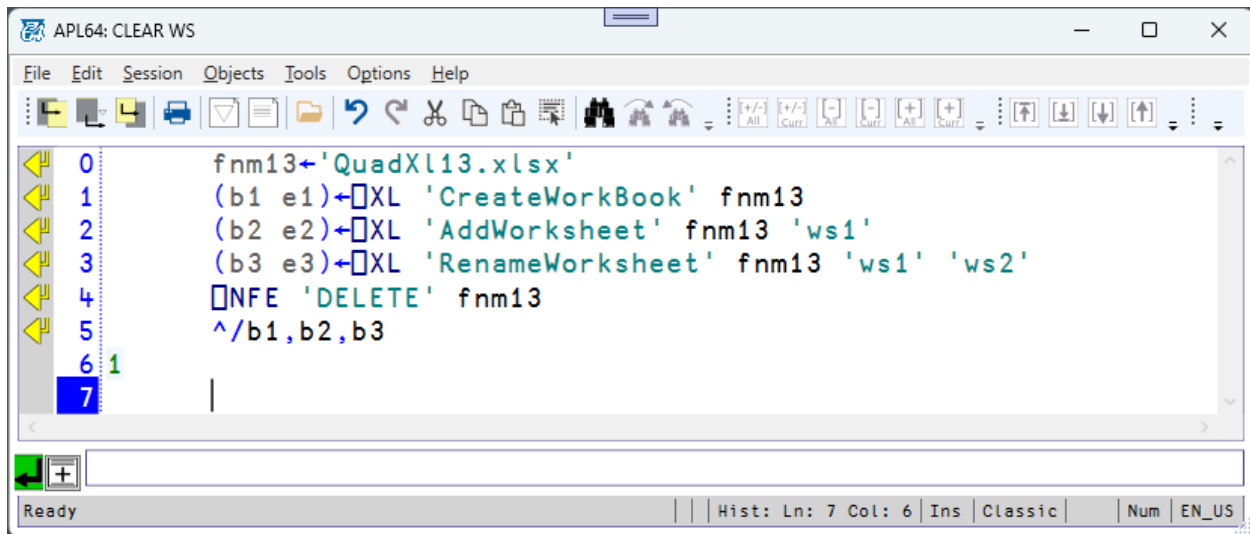
srcWbPath: APL64 text object containing the path to the specified existing Excel workbook.

srcWsName: APL64 text object containing the name of the specified existing worksheet in the source workbook.

tgtWsName: APL64 text object containing the new name for the worksheet.

RenameWorksheet Example

```
fnm13←'QuadXI13.xlsx'
(b1 e1)←⊞XL 'CreateWorkBook' fnm13
(b2 e2)←⊞XL 'AddWorksheet' fnm13 'ws1'
(b3 e3)←⊞XL 'RenameWorksheet' fnm13 'ws1' 'ws2'
⊞NFE 'DELETE' fnm13
^/b1,b2,b3
```



SaveWorkSheetAsCsv

The SaveWorkSheetAsCsv action will access the specified Excel source workbook and worksheet and save it in CSV format.t in csv format.

`(bRes aRes)←⊞XL 'SaveWorkSheetAsCsv' srcWbPath srcWsName tgtCsvPath overwrite decimalSeparator fieldSeparator`

or the synonymous action:

`(bRes aRes)←⊞XL 'SaveWsAsCsv' srcWbPath srcWsName tgtCsvPath overwrite decimalSeparator fieldSeparator`

srcWbPath: APL64 text object containing the path to the specified existing Excel workbook.

srcWsName: APL64 text object containing the name of the specified existing worksheet in the source workbook.

tgtCsvPath: APL64 text object containing the path for the new Excel workbook.

overWrite: If the target csv-format file exists is will be overwritten depending on the value of the overWrite argument (0/No 1/Yes)

decimalSeparator: Decimal separator used in the source file, typically period or comma

fieldSeparator: Field separator used in the target file, typically comma or semi-colon

The target csv-format file will be created if it does not exist.

Csv-format files do not support metadata such as Excel cell formats.

Excel Cell Format	Csv Format
Date	yyyy-MM-dd

DateTime	yyyy-MM-dd HH:mm:ss
Time	HH:mm:ss

SaveWorksheetAsCsv with EN-US Decimal and Field Separators Example

```

mkdir 'c:\test'
wbPath←'c:\test\wBook.xlsx'
← X←2 4p'Abc' 1.234 45 (2024 12 31) 'Efg' 3.45 ^99 (2025 01 01)
←(b e)←XL 'NumberFormatRanges' wbPath 'Sheet1' 'D1:D2' 'm/d/yyyy'
←(b a)←XL 'FromApl' wbPath 'Sheet1' (i2) (i4) 'AplDateTime' X
csvPath←'c:\test\csvData.csv'
←(b a)←XL 'SaveWsAsCsv' wbPath 'Sheet1' csvPath 1 '.' ','
Nfe 'Delete' wbPath
Nfe 'Delete' csvPath
rmdir 'c:\test'

```

```

APL64: CLEAR WS
File Edit Session Objects Tools Options Help
0 mkdir 'c:\test'
1 wbPath←'c:\test\wBook.xlsx'
2 ← X←2 4p'Abc' 1.234 45 (2024 12 31) 'Efg' 3.45 ^99 (2025 01 01)
3 Abc 1.234 45 2024 12 31
4 Efg 3.450 ^99 2025 1 1
5 ←(b e)←XL 'NumberFormatRanges' wbPath 'Sheet1' 'D1:D2' 'm/d/yyyy'
6 1
7 ←(b a)←XL 'FromApl' wbPath 'Sheet1' (i2) (i4) 'AplDateTime' X
8 1
9 csvPath←'c:\test\csvData.csv'
10 ←(b a)←XL 'SaveWsAsCsv' wbPath 'Sheet1' csvPath 1 '.' ','
11 1
12 Nfe 'Delete' wbPath
13 Nfe 'Delete' csvPath
14 rmdir 'c:\test'
15
Ready | Hist: Ln: 0 Col: 0 Ins Classic Num EN_US

```

Resulting Excel-format file:

	A	B	C	D	E	F
1	Abc	1.234	45	12/31/2024		
2	Efg	3.45	-99	1/1/2025		
3						
4						

Ready

Resulting csv-format file:

File	Edit	View			
Abc,1.234,45,2024-12-31 Efg,3.45,-99,2025-01-01					
Ln 1, Col 1	48 characters	100%	Window	UTF-8	

SaveWorksheetAsCsv with non-EN-US Decimal and Field Separators Example

```

❑ mkdir 'c:\test'
wbPath←'c:\test\wBook.xlsx'
❑ ← X←2 4p'Abc' 1.234 45 (2024 12 31) 'Efg' 3.45 `99 (2025 01 01)
❑ ←(b e)←❑XL 'NumberFormatRanges' wbPath 'Sheet1' 'D1:D2' 'm/d/yyyy'
❑ ← (b a)←❑XL 'FromApl' wbPath 'Sheet1' (i2) (i4) 'AplDateTime' X
csvPath←'c:\test\csvData.csv'
❑ ←(b a)←❑XL 'SaveWsAsCsv' wbPath 'Sheet1' csvPath 1 ',' ';'
❑ Nfe 'Delete' wbPath
❑ Nfe 'Delete' csvPath
❑ rmdir 'c:\test'

```


SortSelectedRange

The SortSelectedRange action will sort the specified range of a specified worksheet in a specified workbook according to specified 'sort levels'.

(bRes aRes)←☐XL 'SortSelectedRange' wbPath wsName rangeInfo sortIsCaseSensitive sortLevelsInfo

Or synonymous action:

(bRes aRes)←☐XL 'SortRange' wbPath wsName rangeInfo sortIsCaseSensitive sortLevelsInfo

wbPath: APL64 text object containing the path to the specified existing Excel workbook.

wsName: APL64 text object containing the name of the specified existing worksheet in the workbook.

rangeInfo: A text object containing the Excel 'A1:B1' format specified range in the worksheet to sort.

sortIsCaseSensitive: 0/No 1/Yes

sortLevelsInfo: A two column array of character vectors with one row for each 'sort level'. Each row contains these columns:

- Column Name: A text object containing the column name in Excel 'A1' format
- SortOrder: 'A'/Ascending, 'D'/Descending

bRes: 0/Fail, 1/OK

aRes: Fail: ErrMsg (APL64 character vector)/Fail

aRes: OK: ""

SortSelectedRange Example

APL64 Statements to perform a case-sensitive sort of the range "B2:D4" in the "X" worksheet of the "c:\junk\SortSelectedRange.xlsx" workbook using column "B" in descending order:

```
☐mkdir 'c:\test'  
wbPath←'c:\test\wBook.xlsx'  
☐←X←4 5p☐a  
☐←(b a)←☐XL 'FromApl' wbPath 'Sheet1' (i4) (i5) 'AplDateTime' X  
rangeInfo←"B2:D4"  
sortIsCaseSensitive←0  
sortLevelsInfo←1 2p'B' 'D'  
☐←(b e)←☐XL 'SortRange' wbPath 'Sheet1' rangeInfo sortIsCaseSensitive sortLevelsInfo  
☐Nfe 'Delete' wbPath  
☐rmdir 'c:\test'
```

Worksheet with selected range prior to sorting:

	A	B	C	D	E	F	G	H
1	A	B	C	D	E			
2	F	G	H	I	J			
3	K	L	M	N	O			
4	P	Q	R	S	T			
5								
6								

Worksheet with selected range sorted:

	A	B	C	D	E	F	G
1	A	B	C	D	E		
2	F	Q	R	S	J		
3	K	L	M	N	O		
4	P	G	H	I	T		
5							
6							

Sheet1

Ready Count: 9

100%

The screenshot shows the APL64: CLEAR WS window. The menu bar includes File, Edit, Session, Objects, Tools, Options, and Help. The toolbar contains various icons for file operations, editing, and execution. The main text area displays the following APL script:

```

0  mkdir 'c:\test'
1  wbPath←'c:\test\wBook.xlsx'
2  ⎵X←4 5p⎵a
3  ABCDE
4  FGHIJ
5  KLMNO
6  PQRST
7  ⎵←(b a)⎵XL 'FromApl' wbPath 'Sheet1' (14) (15) 'AplDateTime' X
8  1
9  rangeInfo←"B2:D4"
10 sortIsCaseSensitive←0
11 sortLevelsInfo←1 2p'B' 'D'
12 ⎵←(b a)⎵XL 'SortRange' wbPath 'Sheet1' rangeInfo sortIsCaseSensitive sortLevelsInfo
13 1
14 ⎵Nfe 'Delete' wbPath
15 ⎵rmdir 'c:\test'
16

```

The status bar at the bottom shows "Ready" and "Hist: Ln: 0 Col: 0 Ins Classic Num EN_US".

ToAPL

The ToAPL action will fill an APL64 array with values from selected cells of an Excel worksheet. The selected cells are specified by vectors of row and column numbers.

```
(bRes aRes)←⊞XL 'ToApl' wbPath wsName inclRows inclCols stringCharConv dateTimeConv  
xlEmptyCellAplValue xlCellErrorAplValue
```

wbPath: APL64 text object containing the path to the Excel workbook. The workbook will be created if it does not exist using an Excel format based on the file extension in the path.

wsName: APL64 text object containing the name of the worksheet.

The inclRows and inclCols arguments specify a worksheet range which will be the source of the data returned as an APL64 variable by this ⊞XL action. Order and ⊞IO are significant.

- inclRows: APL64 integer vector of the rows of the source worksheet range.
- inclCols: APL64 integer vector of the columns of the source worksheet range.

stringCharConv: Two options are available:

- 'AplString'
- 'AplChar'

dateTimeConv: Two options are available:

- 'AplDateTime': 7-element APL ⊞TS date time will be converted to XL DateTime
- 'XIDateTime': Floating point value relative to midnight 12/30/1899 (effectively no conversion)

xlEmptyCellAplValue: APL64 object

xlCellErrorAplValue: APL64 object

Row/Col Order and ⊞IO is significant.

bRes: 0/Fail, 1/OK

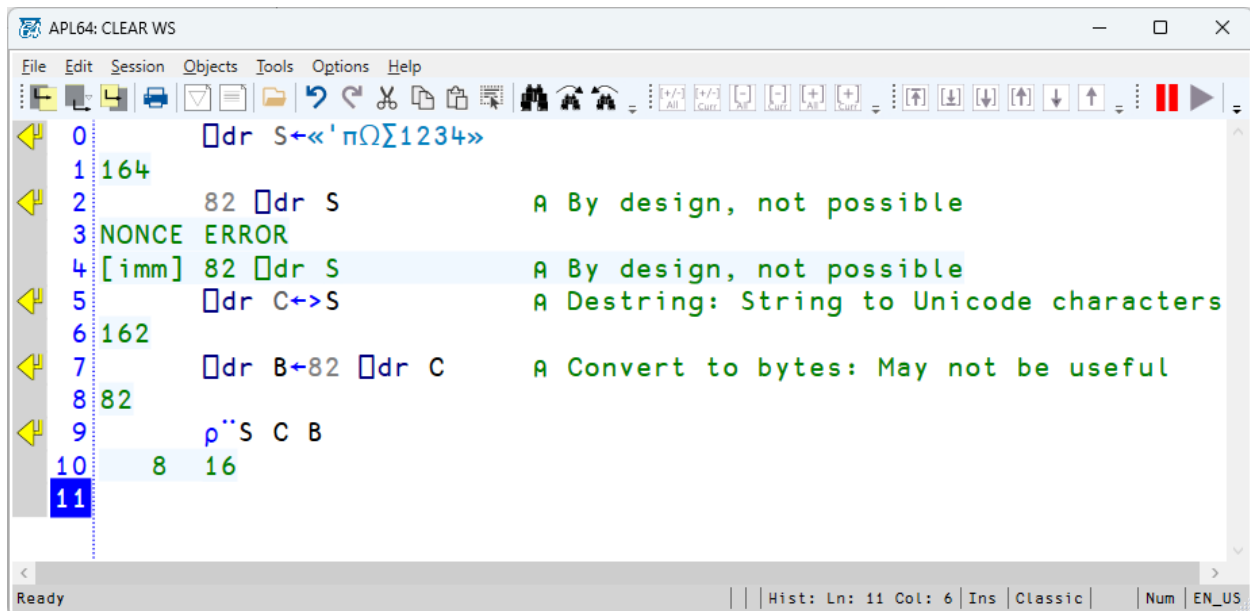
aRes: Fail: ErrMsg (APL64 character vector)/Fail

aRes: OK: APL64 data array containing information obtained from specified cells of the target Excel worksheet.

When the target Excel worksheet values are text information, the 'ToApl' action returns APL64 string or Unicode character text information. All possible text in an Excel workbook can be consumed by APL64 with these data types. The 'ToApl' action argument, stringCharConv, may be selected by the APL64 programmer to determine the returned data type when Excel text information is requested.

If string data is programmer-requested from the 'ToApl' action, the APL64 monadic destrng function '>' applied to an APL64 variable of string data type provides a conversion to a Unicode character array.

For example, a cell value or a worksheet named 'πΩΣ1234' contains Unicode code points not in ⊞AV, so the APL64 string or Unicode character data type is necessary to represent this data. Converting such text data to bytes, APL64 datatype code 82, may not be useful:



ToAPL Example

Refer to the FromAPL example [here](#) which illustrated the FromAPL and ToAPL XL actions.

ToAPLFromRange

The ToAPLFromRange action will fill an APL64 array with values from selected cells of an Excel worksheet. The selected cells are specified by an Excel A1-format range.

```
(bRes aRes)←XL 'ToAplFromRange' wbPath wsName range stringCharConv dateTimeConv
xlEmptyCellAplValue xlCellErrorAplValue
```

wbPath: APL64 text object containing the path to the Excel workbook. The workbook will be created if it does not exist using an Excel format based on the file extension in the path.

wsName: APL64 text object containing the name of the worksheet.

Range: Source range in the worksheet: Excel A1-style range, e.g. 'Z1' or 'A5:C8'

stringCharConv: Two options are available:

- 'AplString'
- 'AplChar'

dateTimeConv: Two options are available:

- 'AplDateTime': 7-element APL TS date time will be converted to XL DateTime
- 'XlDateTime': Floating point value relative to midnight 12/30/1899 (effectively no conversion)

xlEmptyCellAplValue: APL64 object

xlCellErrorAplValue: APL64 object

bRes: 0/Fail, 1/OK

aRes: Fail: ErrMsg (APL64 character vector)/Fail

aRes: OK: APL64 data array containing information obtained from specified cells of the target Excel worksheet.

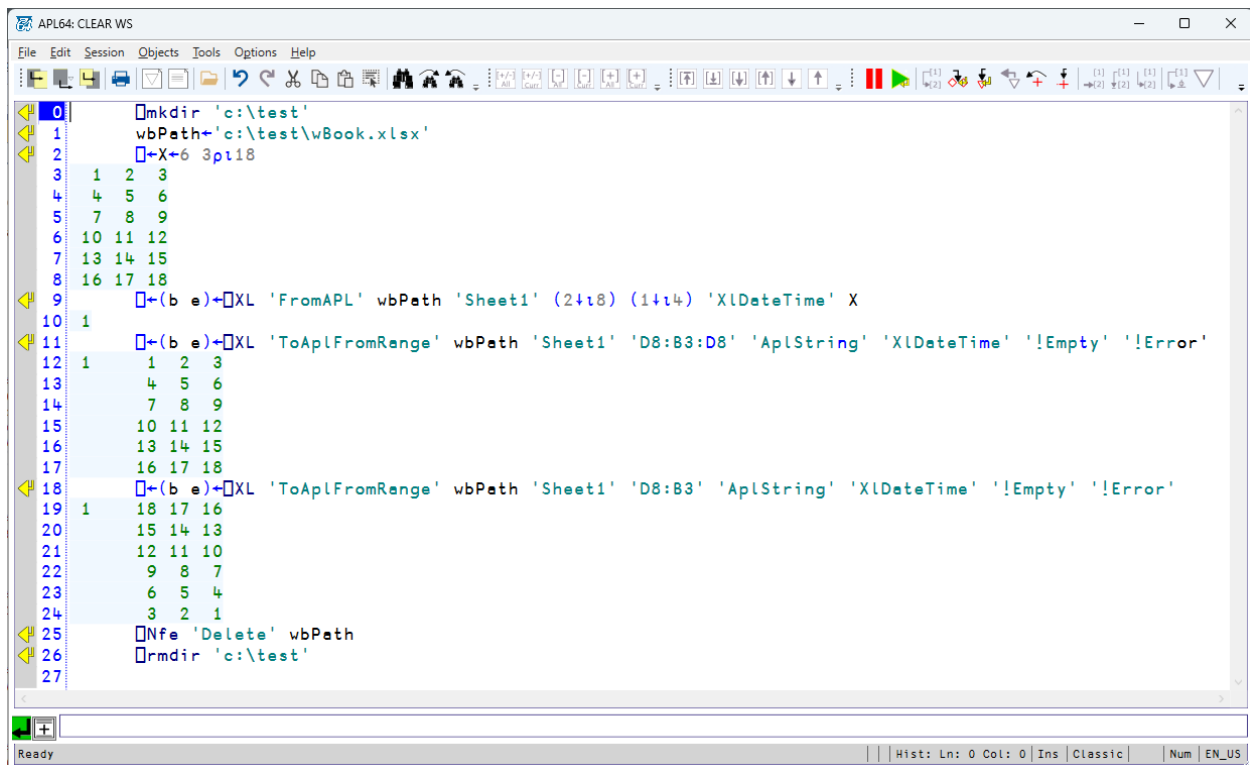
ToAPLFromRange Example

Create a sample worksheet using the FromApl action, and obtain selected values from the worksheet using the ToAPLFromRange action:

```
☐ mkdir 'c:\test'  
wbPath←'c:\test\wBook.xlsx'  
☐ ←X←6 3p18  
☐ ←(b e)←☐ XL 'FromAPL' wbPath 'Sheet1' (2↓18) (1↓14) 'XIDateTime' X  
☐ ←(b e)←☐ XL 'ToAplFromRange' wbPath 'Sheet1' 'D8:B3:D8' 'AplString' 'XIDateTime' '!'Empty' '!'Error'  
☐ ←(b e)←☐ XL 'ToAplFromRange' wbPath 'Sheet1' 'D8:B3' 'AplString' 'XIDateTime' '!'Empty' '!'Error'  
☐ Nfe 'Delete' wbPath  
☐ rmdir 'c:\test'
```

The source workspace:

	A	B	C	D	E	F	G	H
1								
2								
3		1	2	3				
4		4	5	6				
5		7	8	9				
6		10	11	12				
7		13	14	15				
8		16	17	18				
9								
10								



ToAPLFromCsv

The ToAPLFromCsv action will fill an APL64 array with values from a csv-format text file.

```
(bRes aRes)←⎕XL 'ToAPLFromCsv' csvPath inclRows inclCols stringCharConv dateTimeConv
xlEmptyCellAplValue xlCellErrorAplValue decimalSeparator fieldSeparator
```

csvPath: APL64 text object containing the path to the csv-format file.

The inclRows and inclCols arguments specify a range in the source csv-format file which will be the source of the data returned as an APL64 variable by this ⎕XL action. Order and ⎕IO are significant.

- inclRows: APL64 integer vector of the rows of the source csv-format range.
- inclCols: APL64 integer vector of the columns of the source csv-format range.

stringCharConv: Two options are available:

- 'AplString'
- 'AplChar'

dateTimeConv: Two options are available:

- 'AplDateTime': 7-element APL ⎕TS date time will be converted to XL DateTime
- 'XlDateTime': Floating point value relative to midnight 12/30/1899 (effectively no conversion)

xlEmptyCellAplValue: APL64 object

xlCellErrorAplValue: APL64 object

decimalSeparator: Decimal separator used in the source file, typically period or comma

fieldSeparator: Field separator used in the source file, typically comma or semi-colon

Row/Col Order and ☐IO is significant.

bRes: 0/Fail, 1/OK

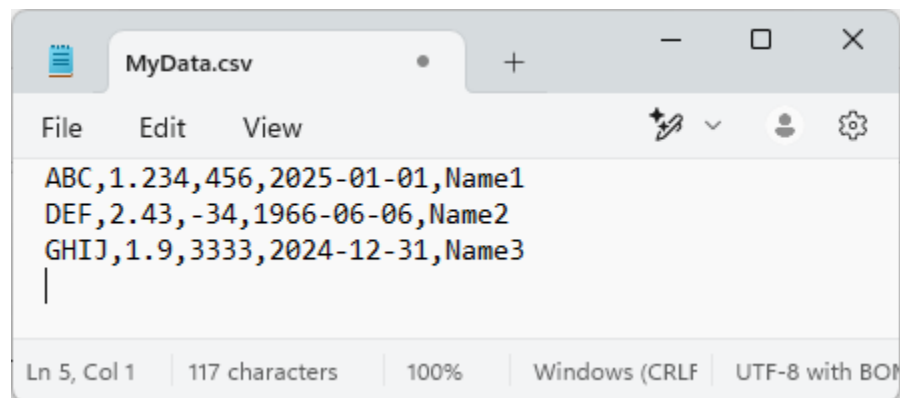
aRes: Fail: ErrMsg (APL64 character vector)/Fail

aRes: OK: APL64 data array containing information obtained from specified cells of the target Excel worksheet.

ToAplFromCsv Example Csv with EN-US Delimiters and Date Information

```
☐mkdir 'c:\test'
  csvPath←'c:\test\MyData.csv'
  X←'COL1,COL2,COL3,COL4,COL5',☐tcl
  X←X,'ABC,1.234,456,2025-01-01,Name1',☐tcl
  X←X,'DEF,2.43,-34,1966-06-06,Name2',☐tcl
  X←X,'GHIJ,1.9,3333,2024-12-31,Name3',☐tcl
  X
COL1,COL2,COL3,COL4,COL5
ABC,1.234,456,2025-01-01,Name1
DEF,2.43,-34,1966-06-06,Name2
GHIJ,1.9,3333,2024-12-31,Name3
'c:\test\MyData.csv'☐xcreate ~1
X☐nappend ~1
☐nuntie ~1
☐←(b e)←☐XL 'ToAPLFromCsv' csvPath (i4)(i5) 'AplString' 'AplDateTime' '!Empty' '!Error' ' ',''
☐←(b e)←☐XL 'ToAPLFromCsv' csvPath (i4)(i5) 'AplString' 'XlDateTime' '!Empty' '!Error' ' ',''
☐Nfe 'Delete' csvPath
☐rmdir 'c:\test'
```

Resulting Csv-format file with period for decimal separator and comma for field separator



```

APL64: CLEAR WS
File Edit Session Objects Tools Options Help
0  mkdir 'c:\test'
1  csvPath←'c:\test\MyData.csv'
2  X←'COL1,COL2,COL3,COL4,COL5',⎕tcnl
3  X←X,'ABC,1.234,456,2025-01-01,Name1',⎕tcnl
4  X←X,'DEF,2.43,-34,1966-06-06,Name2',⎕tcnl
5  X←X,'GHIJ,1.9,3333,2024-12-31,Name3',⎕tcnl
6  X
7  COL1,COL2,COL3,COL4,COL5
8  ABC,1.234,456,2025-01-01,Name1
9  DEF,2.43,-34,1966-06-06,Name2
10 GHIJ,1.9,3333,2024-12-31,Name3
11
12 'c:\test\MyData.csv'⎕xncreate ~1
13 X⎕nappend ~1
14 ⎕nuntie ~1
15
16 ⎕←(b e)←⎕XL 'ToAPLFromCsv' csvPath (i4)(i5) 'AplString' 'AplDateTime' '!Empty' '!Error' '.' ','
17 1 COL1 COL2 COL3 COL4 COL5
18 ABC 1.234 456 2025 1 1 0 0 0 0 Name1
19 DEF 2.430 -34 1966 6 6 0 0 0 0 Name2
20 GHIJ 1.900 3333 2024 12 31 0 0 0 0 Name3
21
22 ⎕←(b e)←⎕XL 'ToAPLFromCsv' csvPath (i4)(i5) 'AplString' 'XlDateTime' '!Empty' '!Error' '.' ','
23 1 COL1 COL2 COL3 COL4 COL5
24 ABC 1.234 456 45658 Name1
25 DEF 2.430 -34 24264 Name2
26 GHIJ 1.900 3333 45657 Name3
26 ⎕Nfe 'Delete' csvPath
27 ⎕rmdir 'c:\test'
28

```

ToAplFromCsv Example Csv with non-EN-US Delimiters and Date Information
Comma for decimal separator and semi-colon for field separator

```

mkdir 'c:\test'
csvPath←'c:\test\MyData.csv'
X← 'COL1;COL2;COL3;COL4 ;COL5' ,⎕tcnl
X←X,'ABC ;1,234;456;2025-01-01;Name1',⎕tcnl
X←X,'DEF ;2,43 ;-34 ;1966-06-06;Name2',⎕tcnl
X←X,'GHIJ;1,9 ;3333;2024-12-31;Name3',⎕tcnl
'c:\test\MyData.csv'⎕xncreate ~1
X⎕nappend ~1
⎕nuntie ~1
⎕←(b e)←⎕XL 'ToAPLFromCsv' csvPath (i4)(i5) 'AplString' 'AplDateTime' '!Empty' '!Error' ';';'
⎕←(b e)←⎕XL 'ToAPLFromCsv' csvPath (i4)(i5) 'AplString' 'XlDateTime' '!Empty' '!Error' ';';'
⎕Nfe 'Delete' csvPath
⎕rmdir 'c:\test'

```

Resulting Csv-format file:

