Using ⎕xl

Contents

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 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 hlAddr hlSubAddr hlTip hlText

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”

hlAddr: url of hyperlink target or '' if hlSubAddr applies, e.g. https://myWebPage...

hlSubAddr: hyperlink target location in workbook or '' if hlAddr applies, e.g. "Sheet2!A1:D4"

hlTip: Comment text which will be added to the specified cell or ''

hlText: The text which will be displayed in the specified cell or ''

Hyperlink Types

For all supported hyperlink types, APL64 cannot validate the hlAddr or hlSubAddr arguments to the ⎕XL AddHyperlink action, because the hyperlink can be created before the hlAddress or hlSubAddr exists.

WebAddress

For this hyperlink type the hlSubAddress must be ''. The hlAddr 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 hlSubAddress must be ''. The hlAddress 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 hlSubAddress must be ''. The hlAddress 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 hlAddr argument must be ''. The hlSubAddr must be a valid A1-format address in the workbook containing the hyperlink.

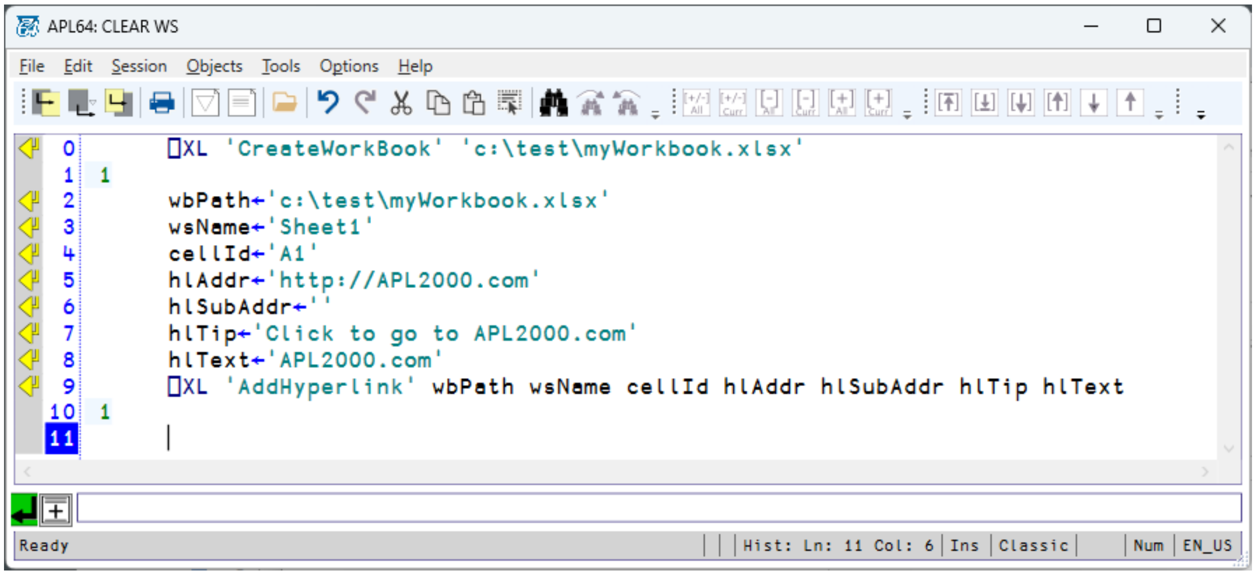
Examples:

G4: Cell in the worksheet containing the hyperlink

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

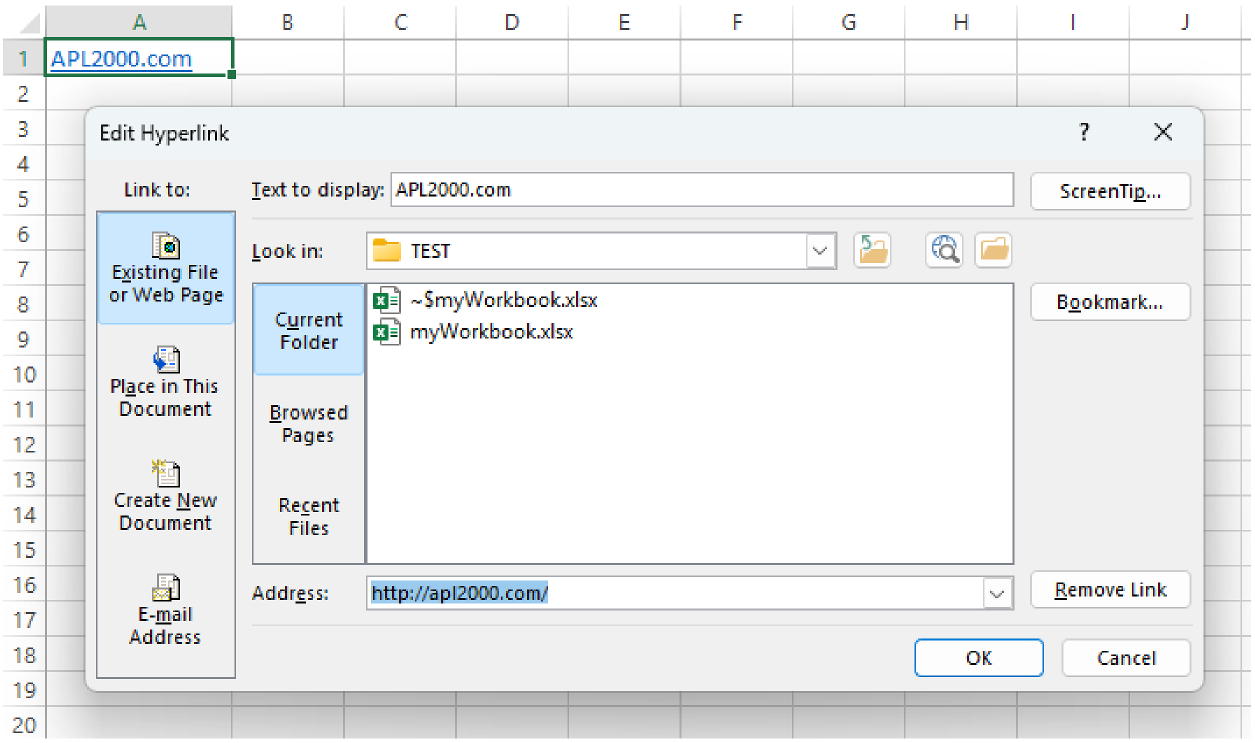
AddHyperlink Example

```
⎕XL 'CreateWorkBook' 'c:\test\myWorkbook.xlsx'
wbPath←'c:\test\myWorkbook.xlsx'
wsName←'Sheet1'
cellId←'A1'
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 'CreateWorkBook' 'c:\test\myWorkbook.xlsx'
1
      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
1
```

APL64: CLEAR WS
File Edit Session Objects Tools Options Help
Ready
Hist: Ln: 11 Col: 6 Ins Classic Num EN_US



Text to display: APL2000.com

~$myWorkbook.xlsx
myWorkbook.xlsx

Address: http://apl2000.com/

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

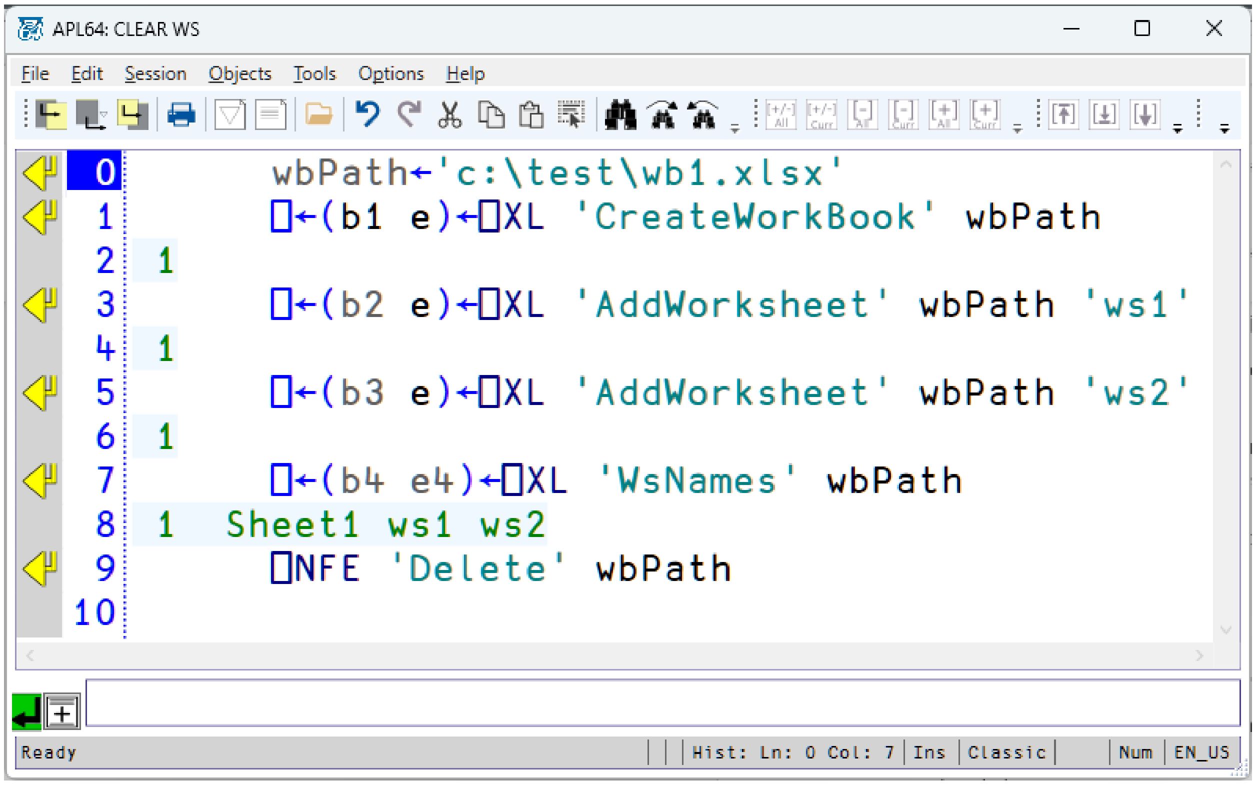
aRes: ErrMsg (APL64 character vector)/Fail, ''/OK

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
```


APL64: CLEAR WS
File Edit Session Objects Tools Options Help

```
      wbPath←'c:\test\wb1.xlsx'
      ⎕←(b1 e)←⎕XL 'CreateWorkBook' wbPath
 1
      ⎕←(b2 e)←⎕XL 'AddWorksheet' wbPath 'ws1'
 1
      ⎕←(b3 e)←⎕XL 'AddWorksheet' wbPath 'ws2'
 1
      ⎕←(b4 e4)←⎕XL 'WsNames' wbPath
 1  Sheet1 ws1 ws2
      ⎕NFE 'Delete' wbPath
```

Ready
Hist: Ln: 0 Col: 7 Ins Classic Num EN_US

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

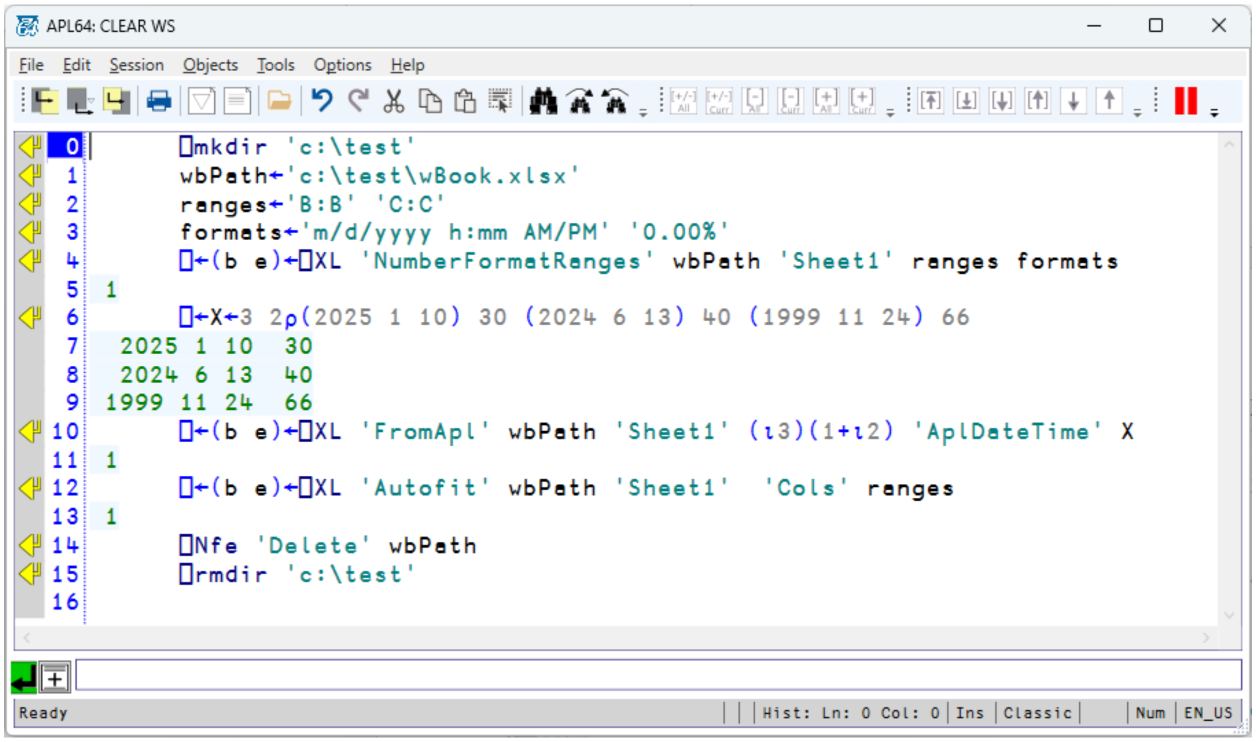
Autofit is applied to the entire range of each of the ranges

Typical row range: "7:9"

Typical column range: "B:C"

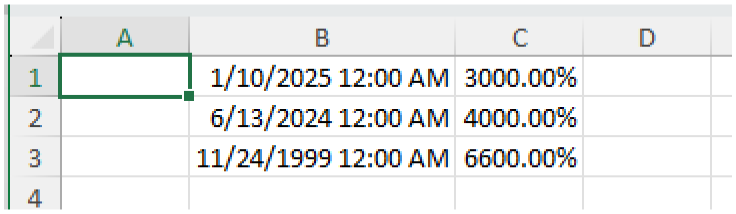
Autofit Specified Columns Example

```
⎕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 2⍴(2025 1 10) 30 (2024 6 13) 40 (1999 11 24) 66
⎕←(b e)←⎕XL 'FromApl' wbPath 'Sheet1' (⍳3)(1+⍳2) 'AplDateTime' X
⎕←(b e)←⎕XL 'Autofit' wbPath 'Sheet1'  'Cols' ranges
⎕Nfe 'Delete' wbPath
⎕rmdir 'c:\test'
```



```
      ⎕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
1
      ⎕←X←3 2⍴(2025 1 10) 30 (2024 6 13) 40 (1999 11 24) 66
  2025 1 10  30
  2024 6 13  40
 1999 11 24  66
      ⎕←(b e)←⎕XL 'FromApl' wbPath 'Sheet1' (⍳3)(1+⍳2) 'AplDateTime' X
1
      ⎕←(b e)←⎕XL 'Autofit' wbPath 'Sheet1'  'Cols' ranges
1
      ⎕Nfe 'Delete' wbPath
      ⎕rmdir 'c:\test'
```

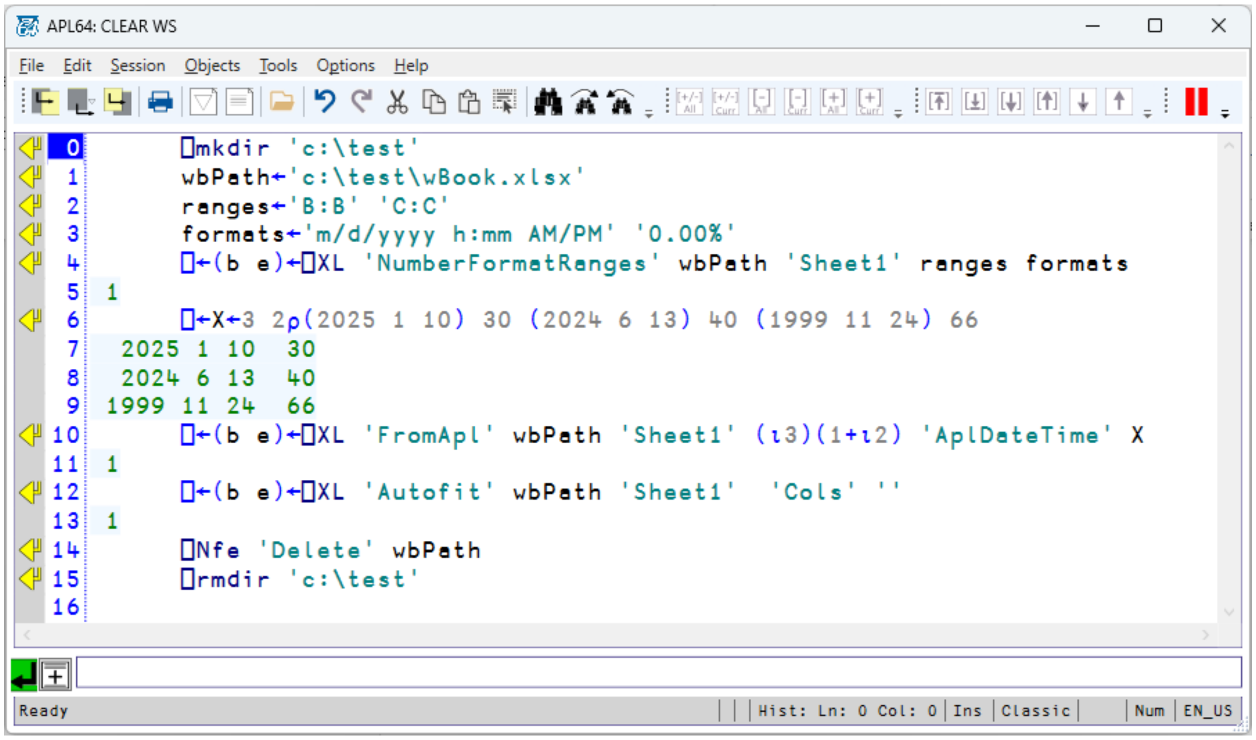
Resulting worksheet:



	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				

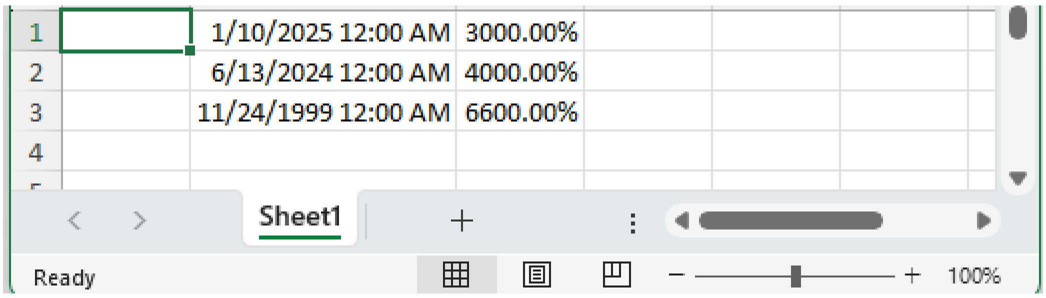
Autofit All Columns Example

```
⎕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 2⍴(2025 1 10) 30 (2024 6 13) 40 (1999 11 24) 66
⎕←(b e)←⎕XL 'FromApl' wbPath 'Sheet1' (⍳3)(1+⍳2) 'AplDateTime' X
⎕←(b e)←⎕XL 'Autofit' wbPath 'Sheet1'  'Cols' ''
⎕Nfe 'Delete' wbPath
⎕rmdir 'c:\test'
```



```
      ⎕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
1
      ⎕←X←3 2⍴(2025 1 10) 30 (2024 6 13) 40 (1999 11 24) 66
 2025 1 10  30
 2024 6 13  40
1999 11 24  66
      ⎕←(b e)←⎕XL 'FromApl' wbPath 'Sheet1' (⍳3)(1+⍳2) 'AplDateTime' X
1
      ⎕←(b e)←⎕XL 'Autofit' wbPath 'Sheet1'  'Cols' ''
1
      ⎕Nfe 'Delete' wbPath
      ⎕rmdir 'c:\test'
```

APL64: CLEAR WS
File Edit Session Objects Tools Options Help
Ready
Hist: Ln: 0 Col: 0 Ins Classic Num EN_US



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

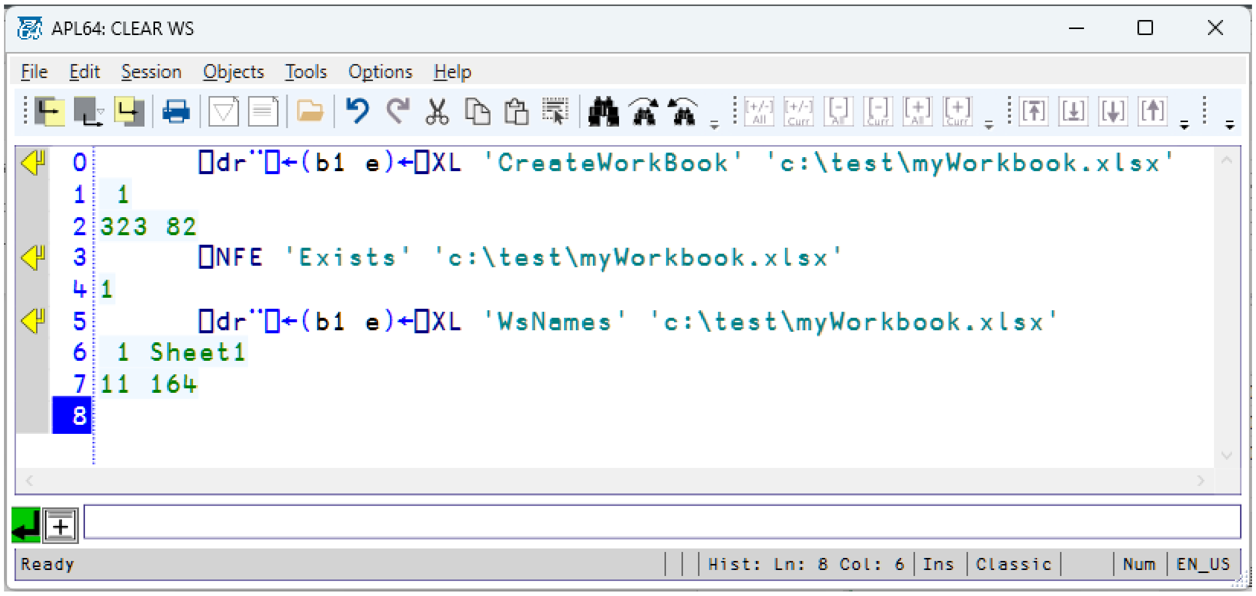
bRes: 0/Fail, 1/OK

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'
```



APL64: CLEAR WS
File Edit Session Objects Tools Options Help

```
      ⎕dr¨⎕←(b1 e)←⎕XL 'CreateWorkBook' 'c:\test\myWorkbook.xlsx'
 1
323 82
      ⎕NFE 'Exists' 'c:\test\myWorkbook.xlsx'
1
      ⎕dr¨⎕←(b1 e)←⎕XL 'WsNames' 'c:\test\myWorkbook.xlsx'
 1 Sheet1
11 164
```

Ready | Hist: Ln: 8 Col: 6 | Ins | Classic | Num | EN_US

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

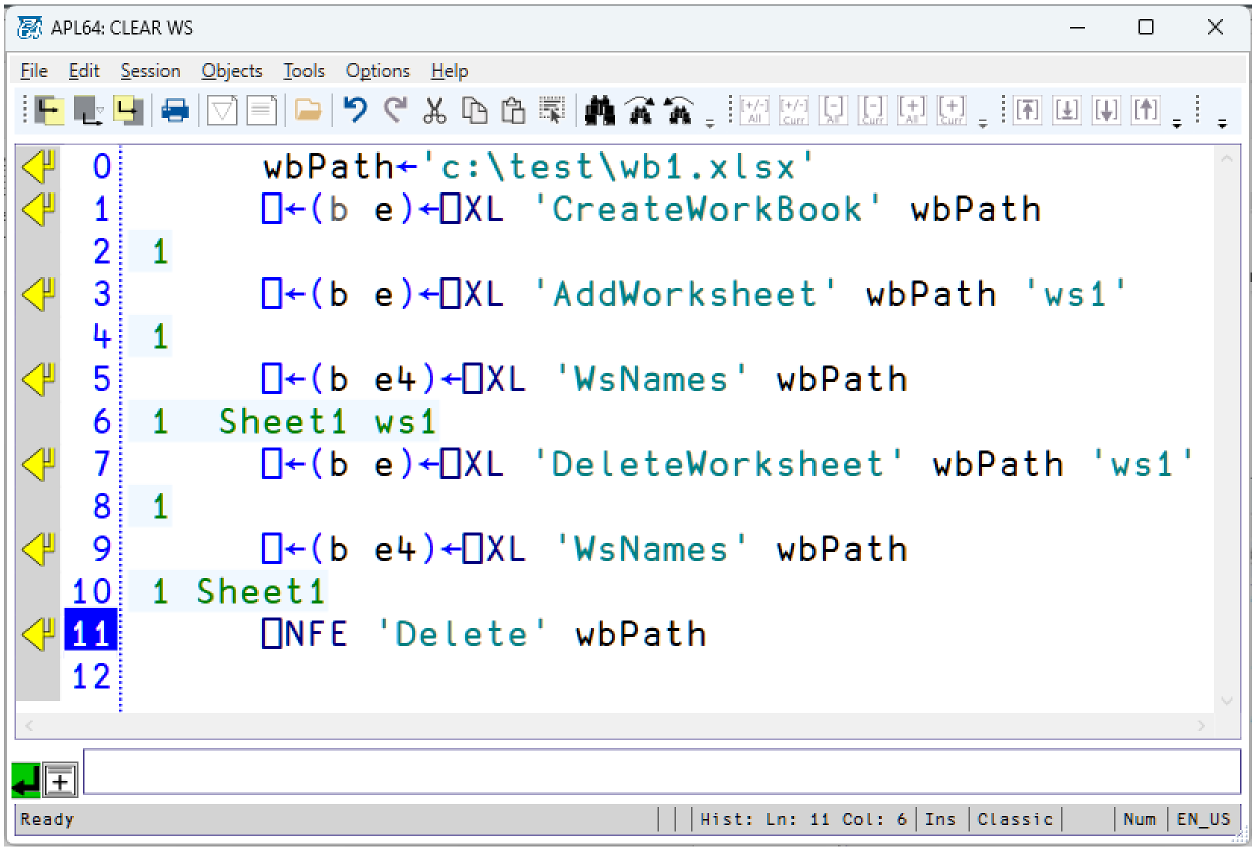
aRes: ErrMsg (APL64 character vector)/Fail, ''/OK

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
```



```
      wbPath←'c:\test\wb1.xlsx'
      ⎕←(b e)←⎕XL 'CreateWorkBook' wbPath
 1
      ⎕←(b e)←⎕XL 'AddWorksheet' wbPath 'ws1'
 1
      ⎕←(b e4)←⎕XL 'WsNames' wbPath
 1  Sheet1 ws1
      ⎕←(b e)←⎕XL 'DeleteWorksheet' wbPath 'ws1'
 1
      ⎕←(b e4)←⎕XL 'WsNames' wbPath
 1 Sheet1
      ⎕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.

inclRows: APL64 integer vector of the rows of the worksheet to fill with the APL64 array data. Order and ⎕IO are significant.

inclCols: APL64 integer vector of the column of the worksheet to fill with the APL64 array data. Order and ⎕IO are significant.

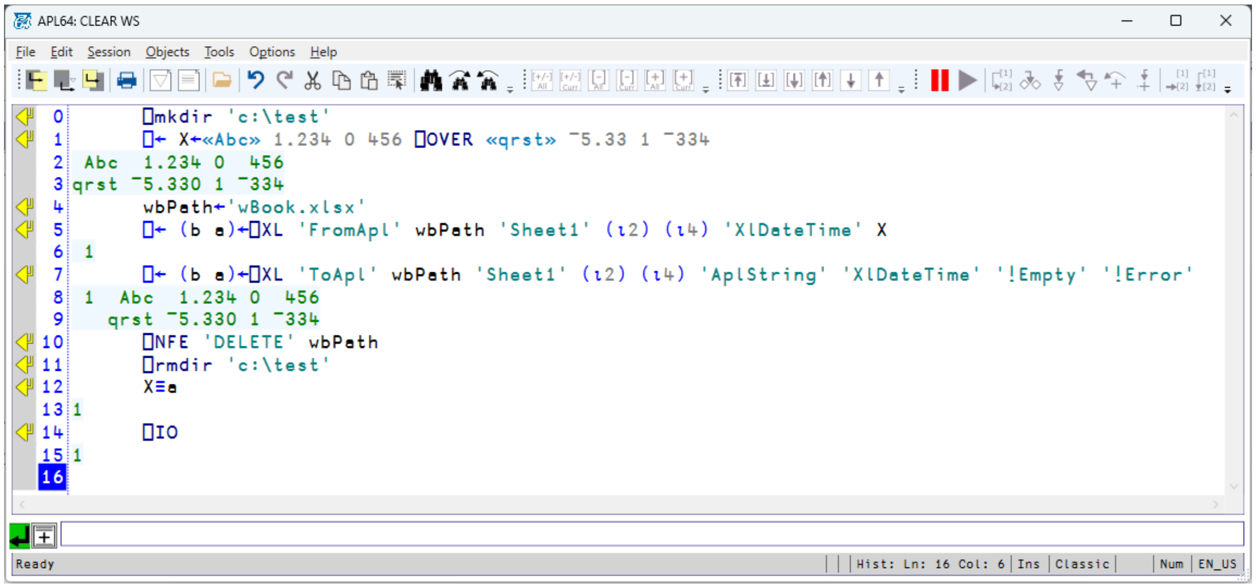
dateTimeConv: Two options are available:

- '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)

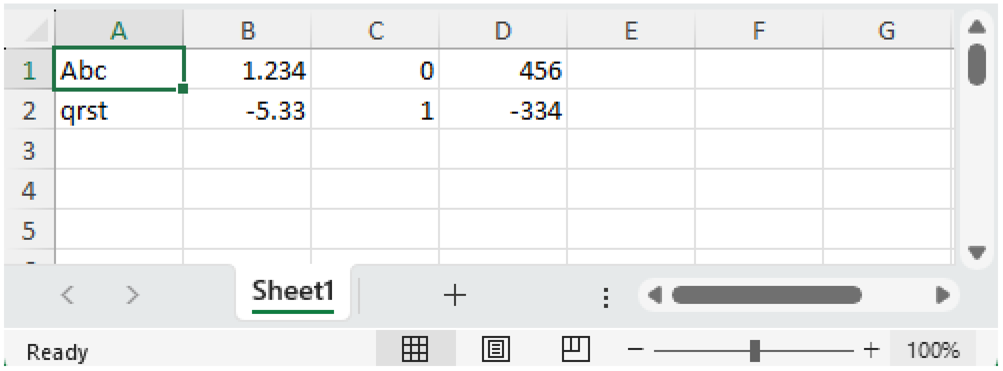
sourceAplValues: The source APL64 elements are accessed in ravel order.

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' (ι2) (ι4) 'XlDateTime' X
⎕← (b a)←⎕XL 'ToApl' wbPath 'Sheet1' (ι2) (ι4) 'AplString' 'XlDateTime' '!Empty' '!Error'
⎕NFE 'DELETE' wbPath
⎕rmdir 'c:\test'
X≡a
⎕IO
```



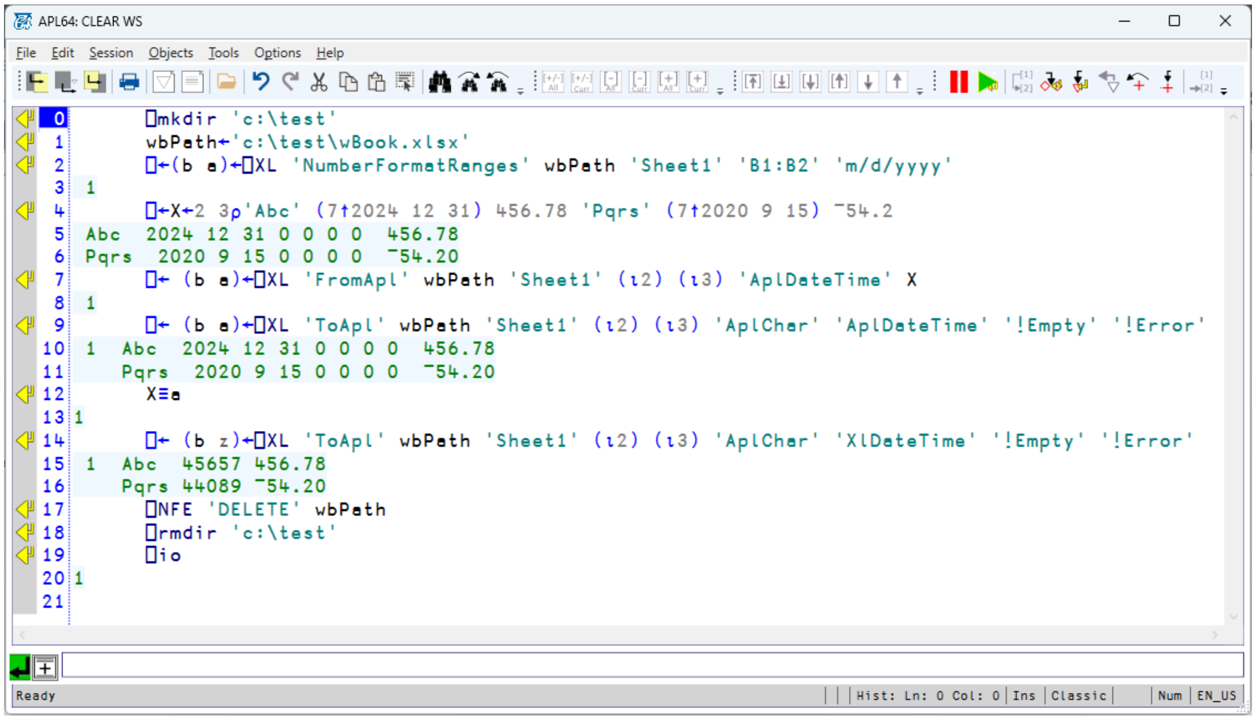
```
      ⎕mkdir 'c:\test'
      ⎕← X←«Abc» 1.234 0 456 ⎕OVER «qrst» ¯5.33 1 ¯334
 Abc   1.234 0   456
qrst ¯5.330 1 ¯334
      wbPath←'wBook.xlsx'
      ⎕← (b a)←⎕XL 'FromApl' wbPath 'Sheet1' (ι2) (ι4) 'XlDateTime' X
 1
      ⎕← (b a)←⎕XL 'ToApl' wbPath 'Sheet1' (ι2) (ι4) 'AplString' 'XlDateTime' '!Empty' '!Error'
 1  Abc   1.234 0   456
   qrst ¯5.330 1 ¯334
      ⎕NFE 'DELETE' wbPath
      ⎕rmdir 'c:\test'
      X≡a
1
      ⎕IO
1
```



	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 3⍴'Abc' (7↑2024 12 31) 456.78 'Pqrs' (7↑2020 9 15) ¯54.2
⎕← (b a)←⎕XL 'FromApl' wbPath 'Sheet1' (⍳2) (⍳3) 'AplDateTime' X
⎕← (b a)←⎕XL 'ToApl' wbPath 'Sheet1' (⍳2) (⍳3) 'AplChar' 'AplDateTime' '!Empty' '!Error'
X≡a
⎕← (b z)←⎕XL 'ToApl' wbPath 'Sheet1' (⍳2) (⍳3) 'AplChar' 'XlDateTime' '!Empty' '!Error'
⎕NFE 'DELETE' wbPath
⎕rmdir 'c:\test'
⎕io
```

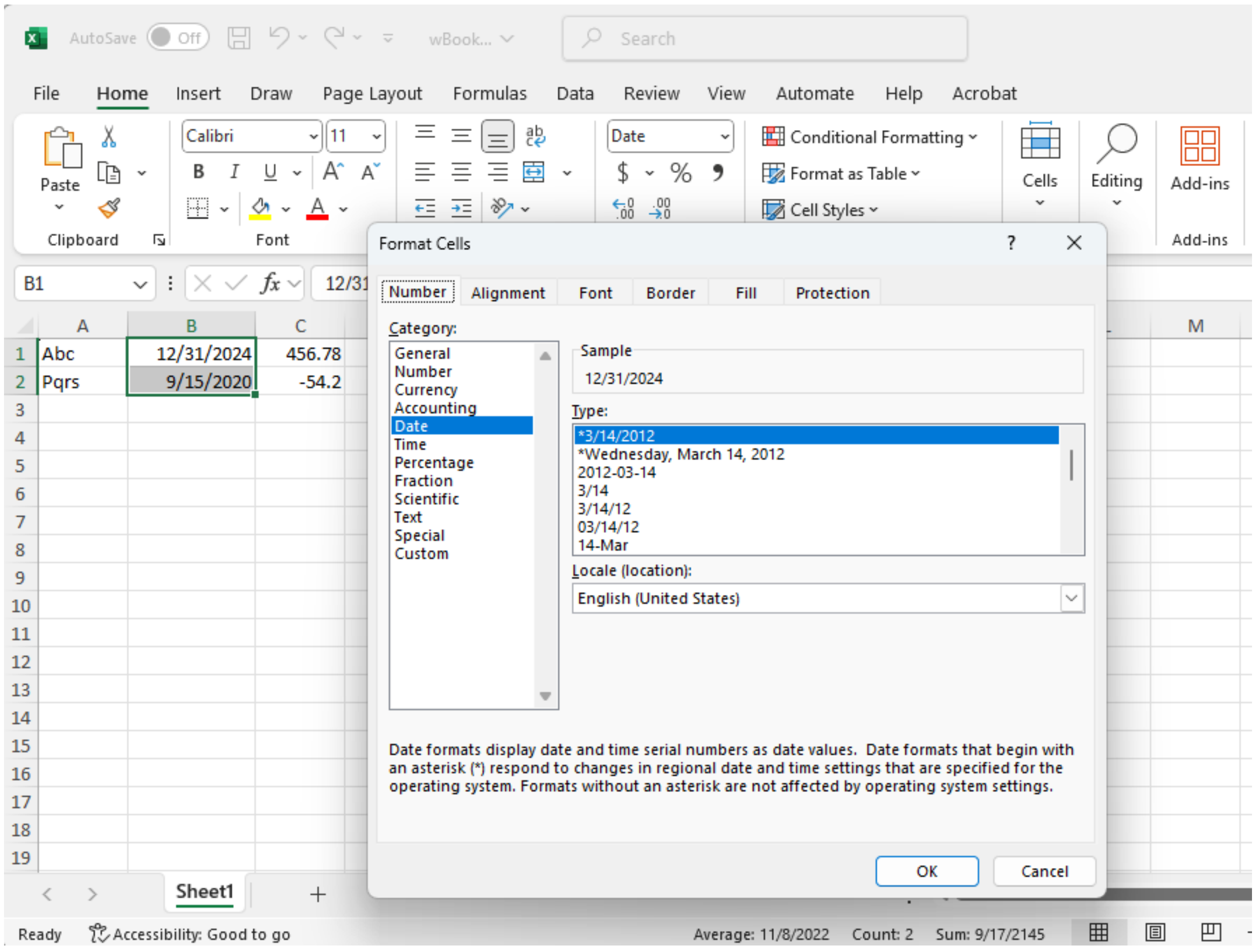


APL64: CLEAR WS
File Edit Session Objects Tools Options Help
Ready
Hist: Ln: 0 Col: 0 Ins Classic Num EN_US

```
      ⎕mkdir 'c:\test'
      wbPath←'c:\test\wBook.xlsx'
      ⎕←(b a)←⎕XL 'NumberFormatRanges' wbPath 'Sheet1' 'B1:B2' 'm/d/yyyy'
 1
      ⎕←X←2 3⍴'Abc' (7↑2024 12 31) 456.78 'Pqrs' (7↑2020 9 15) ¯54.2
 Abc  2024 12 31 0 0 0 0  456.78
 Pqrs  2020 9 15 0 0 0 0  ¯54.20
      ⎕← (b a)←⎕XL 'FromApl' wbPath 'Sheet1' (⍳2) (⍳3) 'AplDateTime' X
 1
      ⎕← (b a)←⎕XL 'ToApl' wbPath 'Sheet1' (⍳2) (⍳3) 'AplChar' 'AplDateTime' '!Empty' '!Error'
 1  Abc  2024 12 31 0 0 0 0  456.78
    Pqrs  2020 9 15 0 0 0 0  ¯54.20
      X≡a
1
      ⎕← (b z)←⎕XL 'ToApl' wbPath 'Sheet1' (⍳2) (⍳3) 'AplChar' 'XlDateTime' '!Empty' '!Error'
 1  Abc  45657 456.78
    Pqrs 44089 ¯54.20
      ⎕NFE 'DELETE' wbPath
      ⎕rmdir 'c:\test'
      ⎕io
1
```

⎕XL 'NumberFormatRanges' formats the range 'B1:B2' as Excel dates. The APL variable X contains dates in ⎕TS format. ⎕XL 'FromApl' writes the APL variable values to the specified cells. Because the range

'B1:B2' is number-formatted as Excel dates, ⎕XL 'ToApl' assigns the XL values from the specified cells to the APL variable a, using the AplDateTime format for those cell values. ⎕XL 'ToApl' with the XlDateTime format returns the date values in Excel as Excel date numbers.



B1 12/31

	A	B	C
1	Abc	12/31/2024	456.78
2	Pqrs	9/15/2020	-54.2

Average: 11/8/2022 Count: 2 Sum: 9/17/2145

Category: General, Number, Currency, Accounting, **Date**, Time, Percentage, Fraction, Scientific, Text, Special, Custom

Sample: 12/31/2024

Type: ***3/14/2012**, *Wednesday, March 14, 2012, 2012-03-14, 3/14, 3/14/12, 03/14/12, 14-Mar

Locale (location): English (United States)

Date formats display date and time serial numbers as date values. Date formats that begin with an asterisk (*) respond to changes in regional date and time settings that are specified for the operating system. Formats without an asterisk are not affected by operating system settings.

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 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 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:

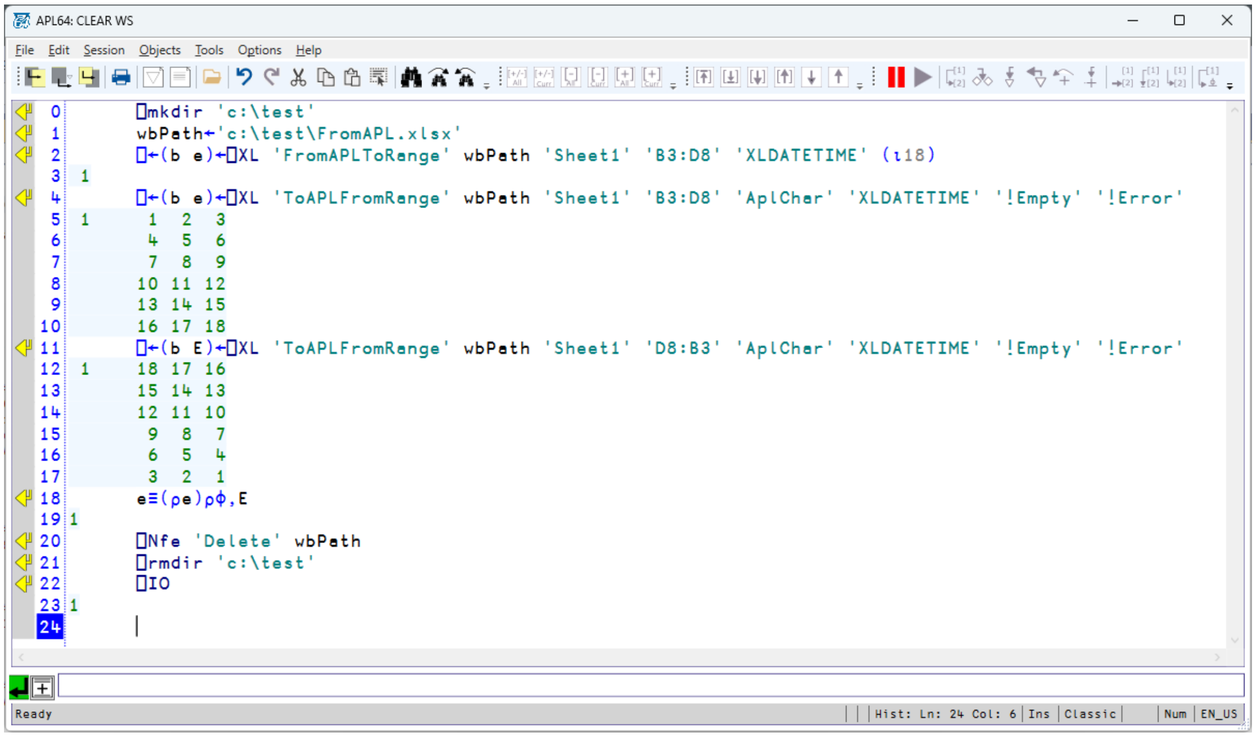
- 'AplDateTime': 7-element APL ⎕TS date time will be converted to XL DateTime
- 'XlDateTime' : Floating point value relative to midnight 12/30/1899 (effectively no conversion)

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

FromAplToRange

```
⎕mkdir 'c:\test'
wbPath←'c:\test\FromAPL.xlsx'
⎕←(b e)←⎕XL 'FromAPLToRange' wbPath 'Sheet1' 'B3:D8' 'XLDATETIME' (⍳18)
⎕←(b e)←⎕XL 'ToAPLFromRange' wbPath 'Sheet1' 'B3:D8' 'AplChar' 'XLDATETIME' '!Empty' '!Error'
⎕←(b E)←⎕XL 'ToAPLFromRange' wbPath 'Sheet1' 'D8:B3' 'AplChar' 'XLDATETIME' '!Empty' '!Error'
e≡(⍴e)⍴⌽,E
⎕Nfe 'Delete' wbPath
⎕rmdir 'c:\test'
⎕IO
```

The order of cells specifying the target range is significant.



APL64: CLEAR WS

```
      ⎕mkdir 'c:\test'
      wbPath←'c:\test\FromAPL.xlsx'
      ⎕←(b e)←⎕XL 'FromAPLToRange' wbPath 'Sheet1' 'B3:D8' 'XLDATETIME' (⍳18)
 1
      ⎕←(b e)←⎕XL 'ToAPLFromRange' wbPath 'Sheet1' 'B3:D8' 'AplChar' 'XLDATETIME' '!Empty' '!Error'
 1     1  2  3
       4  5  6
       7  8  9
      10 11 12
      13 14 15
      16 17 18
      ⎕←(b E)←⎕XL 'ToAPLFromRange' wbPath 'Sheet1' 'D8:B3' 'AplChar' 'XLDATETIME' '!Empty' '!Error'
 1    18 17 16
      15 14 13
      12 11 10
       9  8  7
       6  5  4
       3  2  1
      e≡(⍴e)⍴⌽,E
1
      ⎕Nfe 'Delete' wbPath
      ⎕rmdir 'c:\test'
      ⎕IO
1
      
```

Ready | Hist: Ln: 24 Col: 6 | Ins | Classic | Num | EN_US

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.

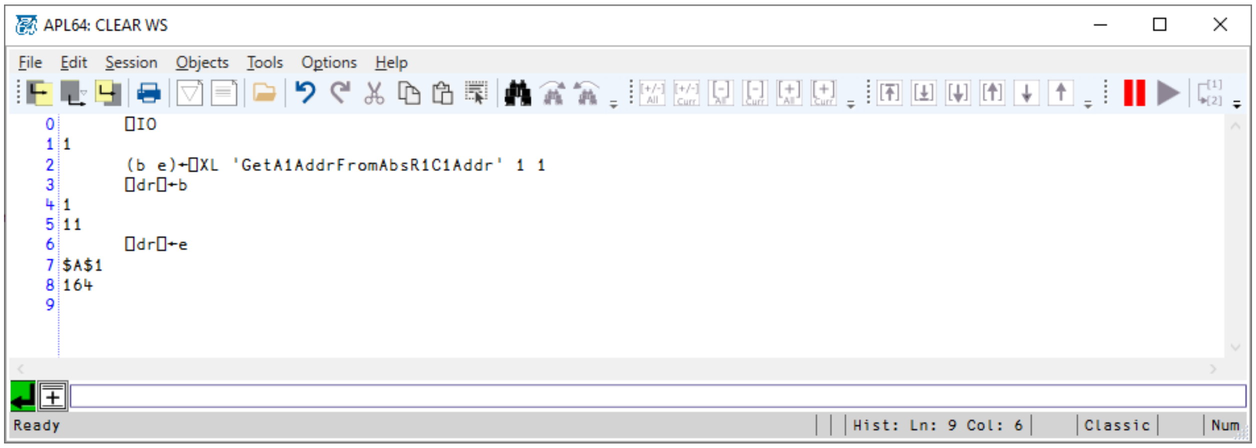
bRes: 0/Fail, 1/OK

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
```



```
      ⎕IO
1
      (b e)←⎕XL 'GetA1AddrFromAbsR1C1Addr' 1 1
      ⎕dr⎕←b
1
11
      ⎕dr⎕←e
$A$1
164
```

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.

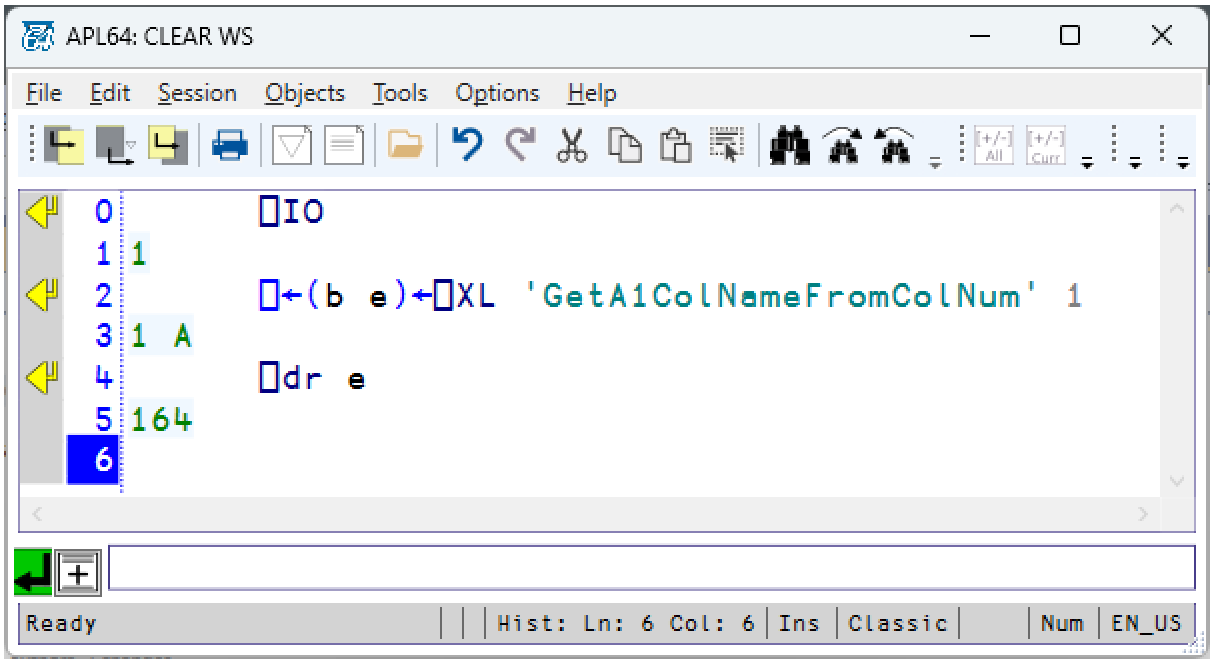
bRes: 0/Fail, 1/OK

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

aRes: OK: APL64 string containing Excel column name

GetA1ColNameFromColNum Example #1

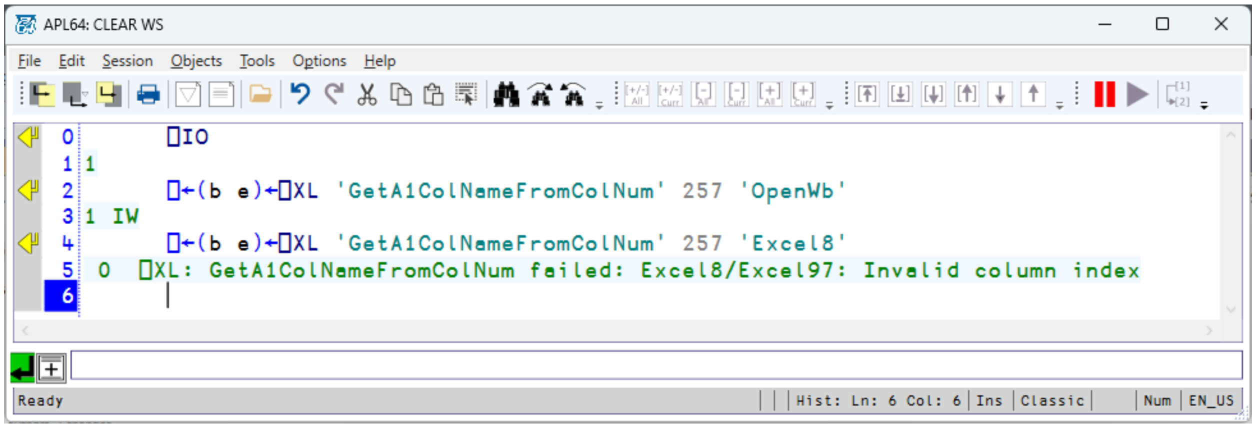
```
⎕IO
⎕←(b e)←⎕XL 'GetA1ColNameFromColNum' 1
⎕dr e
```



```
      ⎕IO
1
      ⎕←(b e)←⎕XL 'GetA1ColNameFromColNum' 1
1 A
      ⎕dr e
164
```

GetA1ColNameFromColNum Example #2

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


APL64: CLEAR WS
File Edit Session Objects Tools Options Help

```
      ⎕IO
1
      ⎕←(b e)←⎕XL 'GetA1ColNameFromColNum' 257 'OpenWb'
1 IW
      ⎕←(b e)←⎕XL 'GetA1ColNameFromColNum' 257 'Excel8'
 0  ⎕XL: GetA1ColNameFromColNum failed: Excel8/Excel97: Invalid column index
```

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

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.

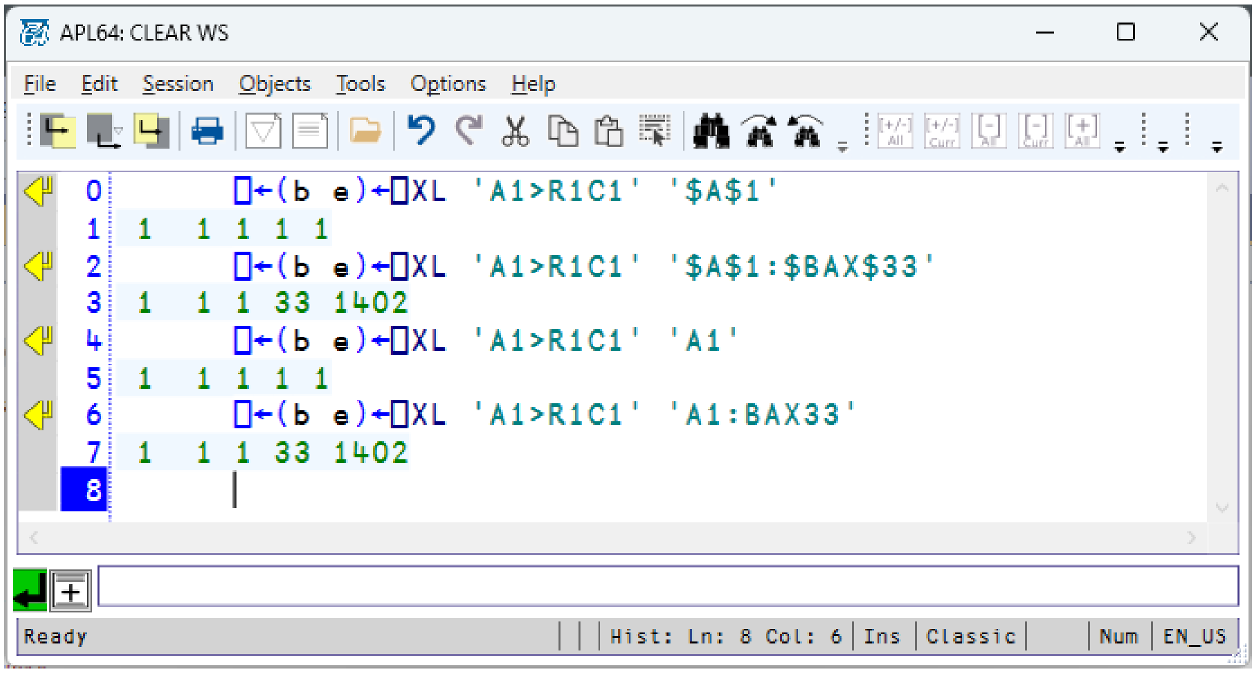
bRes: 0/Fail, 1/OK

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'
```


APL64: CLEAR WS
File Edit Session Objects Tools Options Help

```
      ⎕←(b e)←⎕XL 'A1>R1C1' '$A$1'
 1    1 1 1 1
      ⎕←(b e)←⎕XL 'A1>R1C1' '$A$1:$BAX$33'
 1    1 1 33 1402
      ⎕←(b e)←⎕XL 'A1>R1C1' 'A1'
 1    1 1 1 1
      ⎕←(b e)←⎕XL 'A1>R1C1' 'A1:BAX33'
 1    1 1 33 1402
```

Ready | Hist: Ln: 8 Col: 6 | Ins | Classic | Num | EN_US

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.

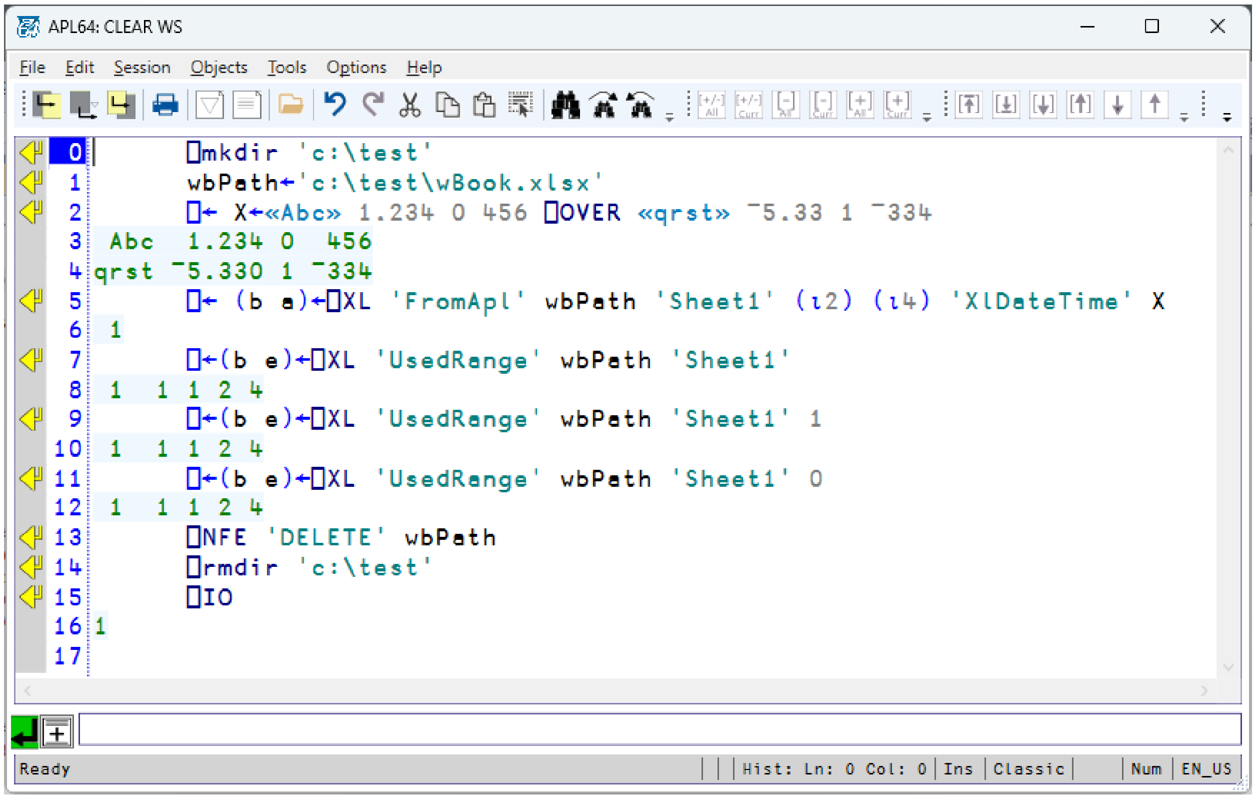
bRes: 0/Fail, 1/OK

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' (⍳2) (⍳4) 'XlDateTime' 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
```


APL64: CLEAR WS
File Edit Session Objects Tools Options Help
Ready
Hist: Ln: 0 Col: 0 Ins Classic
Num EN_US

```
      ⎕mkdir 'c:\test'
      wbPath←'c:\test\wBook.xlsx'
      ⎕← X←«Abc» 1.234 0 456 ⎕OVER «qrst» ¯5.33 1 ¯334
 Abc   1.234 0   456
qrst ¯5.330 1 ¯334
      ⎕← (b a)←⎕XL 'FromApl' wbPath 'Sheet1' (⍳2) (⍳4) 'XlDateTime' X
 1
      ⎕←(b e)←⎕XL 'UsedRange' wbPath 'Sheet1'
 1   1 1 2 4
      ⎕←(b e)←⎕XL 'UsedRange' wbPath 'Sheet1' 1
 1   1 1 2 4
      ⎕←(b e)←⎕XL 'UsedRange' wbPath 'Sheet1' 0
 1   1 1 2 4
      ⎕NFE 'DELETE' wbPath
      ⎕rmdir 'c:\test'
      ⎕IO
1
```

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

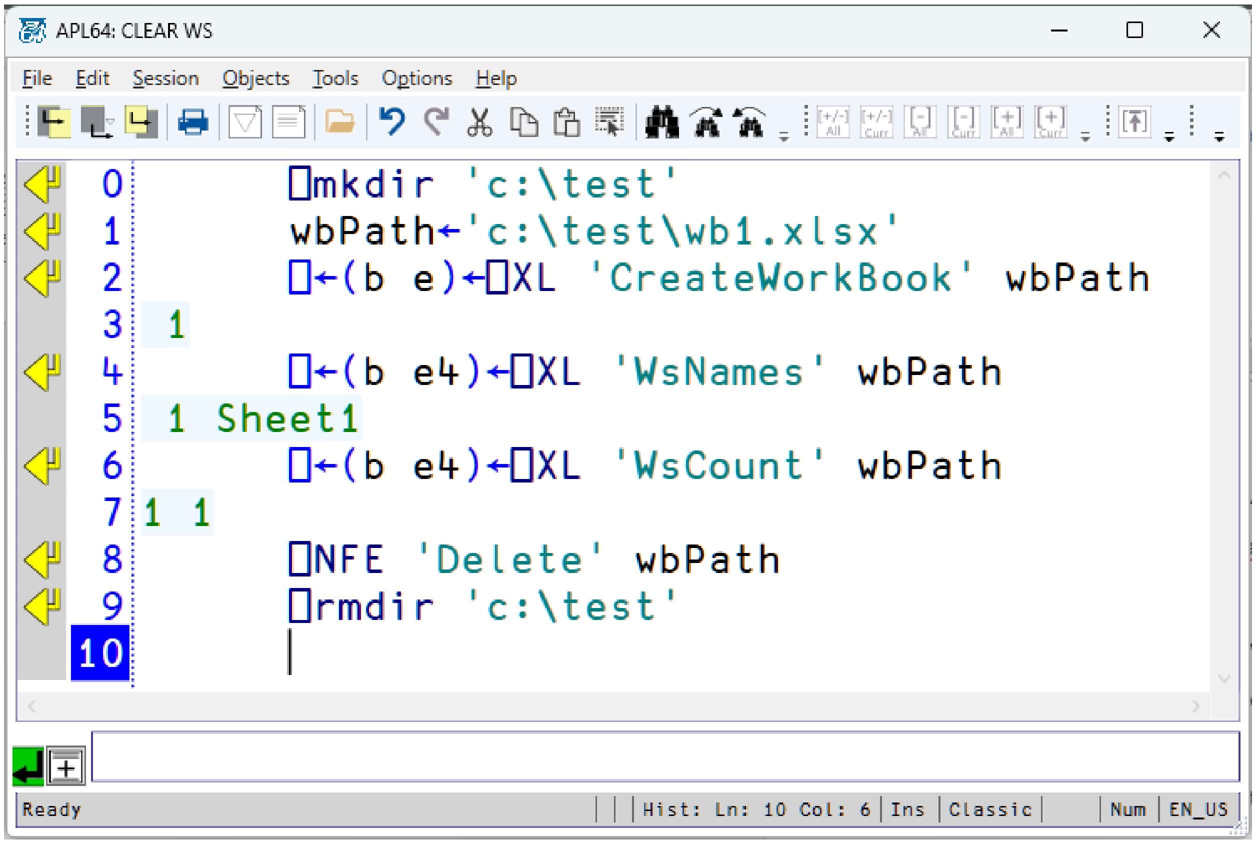
bRes: 0/Fail, 1/OK

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'
```


APL64: CLEAR WS
File Edit Session Objects Tools Options Help

```
      ⎕mkdir 'c:\test'
      wbPath←'c:\test\wb1.xlsx'
      ⎕←(b e)←⎕XL 'CreateWorkBook' wbPath
 1
      ⎕←(b e4)←⎕XL 'WsNames' wbPath
 1 Sheet1
      ⎕←(b e4)←⎕XL 'WsCount' wbPath
1  1
      ⎕NFE 'Delete' wbPath
      ⎕rmdir 'c:\test'
```

Ready | Hist: Ln: 10 Col: 6 | Ins | Classic | Num | EN_US

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

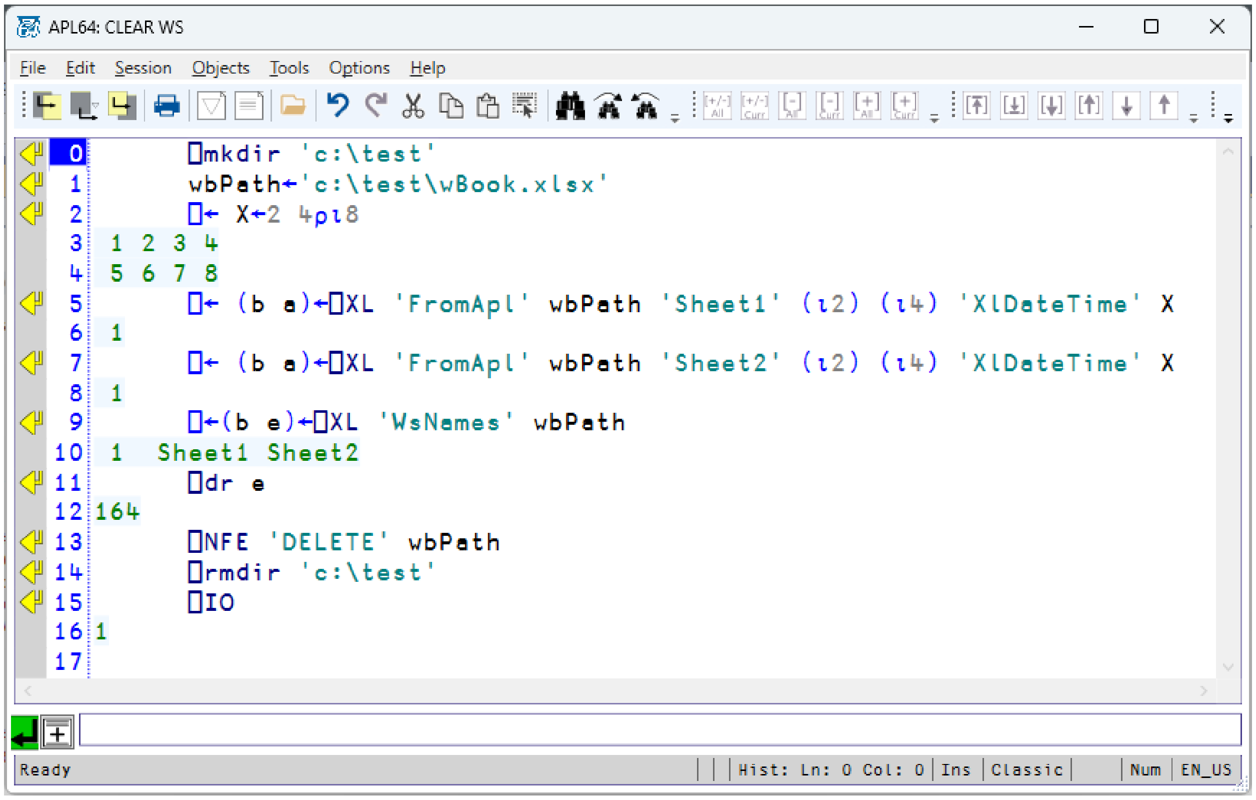
bRes: 0/Fail, 1/OK

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 4ρι8
⎕← (b a)←⎕XL 'FromApl' wbPath 'Sheet1' (ι2) (ι4) 'XlDateTime' X
⎕← (b a)←⎕XL 'FromApl' wbPath 'Sheet2' (ι2) (ι4) 'XlDateTime' X
⎕←(b e)←⎕XL 'WsNames' wbPath
⎕dr e
⎕NFE 'DELETE' wbPath
⎕rmdir 'c:\test'
⎕IO
```



```
      ⎕mkdir 'c:\test'
      wbPath←'c:\test\wBook.xlsx'
      ⎕← X←2 4ρι8
 1 2 3 4
 5 6 7 8
      ⎕← (b a)←⎕XL 'FromApl' wbPath 'Sheet1' (ι2) (ι4) 'XlDateTime' X
 1
      ⎕← (b a)←⎕XL 'FromApl' wbPath 'Sheet2' (ι2) (ι4) 'XlDateTime' X
 1
      ⎕←(b e)←⎕XL 'WsNames' wbPath
 1  Sheet1 Sheet2
      ⎕dr e
164
      ⎕NFE 'DELETE' wbPath
      ⎕rmdir 'c:\test'
      ⎕IO
1
```

Help

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

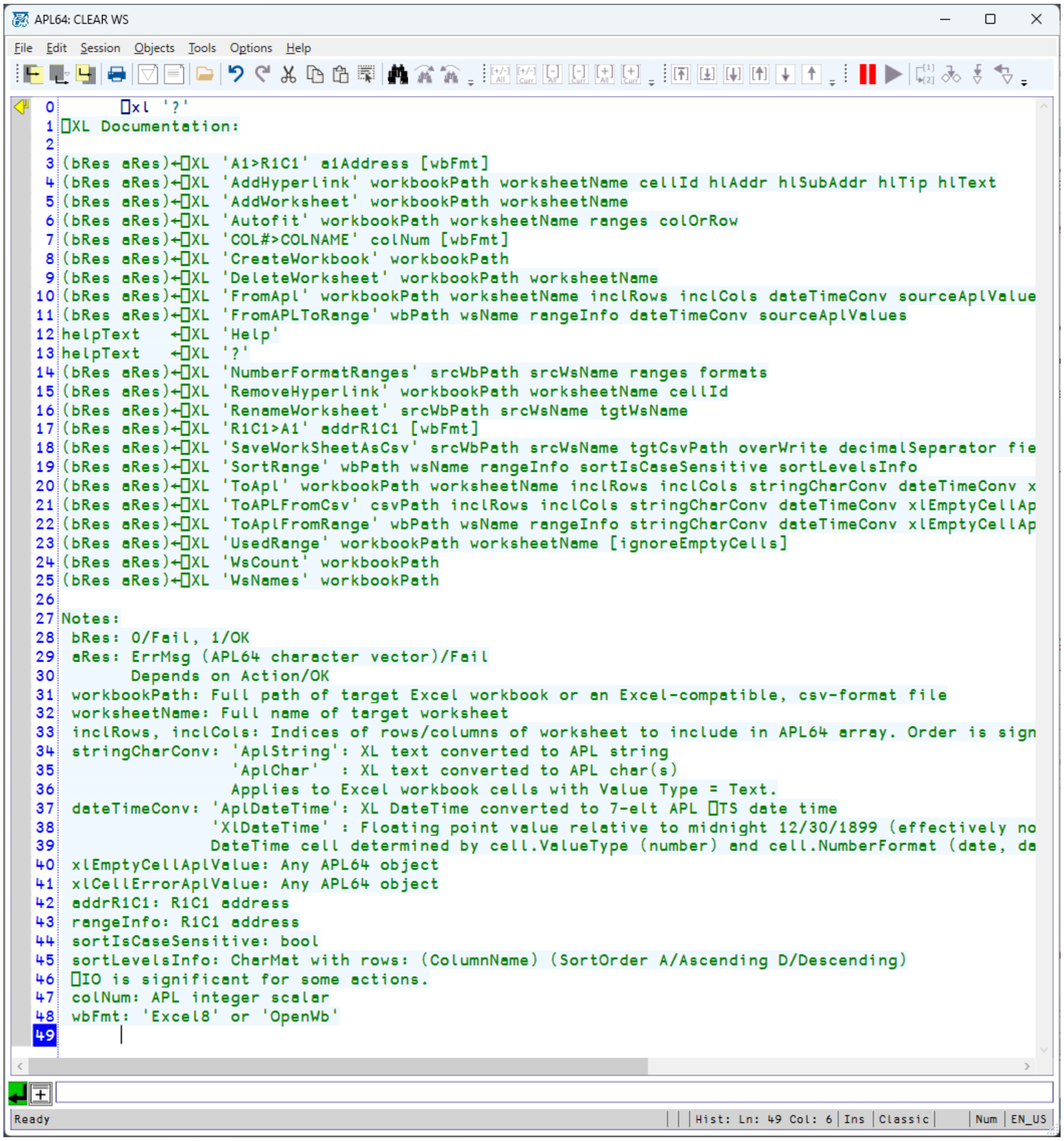
(bRes aRes) ← ⎕XL 'Help' or (bRes aRes) ← ⎕XL '?'

bRes: 0/Fail, 1/OK

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

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

Help Example



```
      ⎕xl '?'
⎕XL Documentation:

(bRes aRes)←⎕XL 'A1>R1C1' a1Address [wbFmt]
(bRes aRes)←⎕XL 'AddHyperlink' workbookPath worksheetName cellId hlAddr hlSubAddr hlTip hlText
(bRes aRes)←⎕XL 'AddWorksheet' workbookPath worksheetName
(bRes aRes)←⎕XL 'Autofit' workbookPath worksheetName ranges colOrRow
(bRes aRes)←⎕XL 'COL#>COLNAME' colNum [wbFmt]
(bRes aRes)←⎕XL 'CreateWorkbook' workbookPath
(bRes aRes)←⎕XL 'DeleteWorksheet' workbookPath worksheetName
(bRes aRes)←⎕XL 'FromApl' workbookPath worksheetName inclRows inclCols dateTimeConv sourceAplValue
(bRes aRes)←⎕XL 'FromAPLToRange' wbPath wsName rangeInfo dateTimeConv sourceAplValues
helpText    ←⎕XL 'Help'
helpText    ←⎕XL '?'
(bRes aRes)←⎕XL 'NumberFormatRanges' srcWbPath srcWsName ranges formats
(bRes aRes)←⎕XL 'RemoveHyperlink' workbookPath worksheetName cellId
(bRes aRes)←⎕XL 'RenameWorksheet' srcWbPath srcWsName tgtWsName
(bRes aRes)←⎕XL 'R1C1>A1' addrR1C1 [wbFmt]
(bRes aRes)←⎕XL 'SaveWorkSheetAsCsv' srcWbPath srcWsName tgtCsvPath overWrite decimalSeparator fie
(bRes aRes)←⎕XL 'SortRange' wbPath wsName rangeInfo sortIsCaseSensitive sortLevelsInfo
(bRes aRes)←⎕XL 'ToApl' workbookPath worksheetName inclRows inclCols stringCharConv dateTimeConv x
(bRes aRes)←⎕XL 'ToAPLFromCsv' csvPath inclRows inclCols stringCharConv dateTimeConv xlEmptyCellAp
(bRes aRes)←⎕XL 'ToAplFromRange' wbPath wsName rangeInfo stringCharConv dateTimeConv xlEmptyCellAp
(bRes aRes)←⎕XL 'UsedRange' workbookPath worksheetName [ignoreEmptyCells]
(bRes aRes)←⎕XL 'WsCount' workbookPath
(bRes aRes)←⎕XL 'WsNames' workbookPath

Notes:
 bRes: 0/Fail, 1/OK
 aRes: ErrMsg (APL64 character vector)/Fail
       Depends on Action/OK
 workbookPath: Full path of target Excel workbook or an Excel-compatible, csv-format file
 worksheetName: Full name of target worksheet
 inclRows, inclCols: Indices of rows/columns of worksheet to include in APL64 array. Order is sign
 stringCharConv: 'AplString': XL text converted to APL string
                 'AplChar'  : XL text converted to APL char(s)
                 Applies to Excel workbook cells with Value Type = Text.
 dateTimeConv: 'AplDateTime': XL DateTime converted to 7-elt APL ⎕TS date time
               'XlDateTime' : Floating point value relative to midnight 12/30/1899 (effectively no
               DateTime cell determined by cell.ValueType (number) and cell.NumberFormat (date, da
 xlEmptyCellAplValue: Any APL64 object
 xlCellErrorAplValue: Any APL64 object
 addrR1C1: R1C1 address
 rangeInfo: R1C1 address
 sortIsCaseSensitive: bool
 sortLevelsInfo: CharMat with rows: (ColumnName) (SortOrder A/Ascending D/Descending)
 ⎕IO is significant for some actions.
 colNum: APL integer scalar
 wbFmt: 'Excel8' or 'OpenWb'
```

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

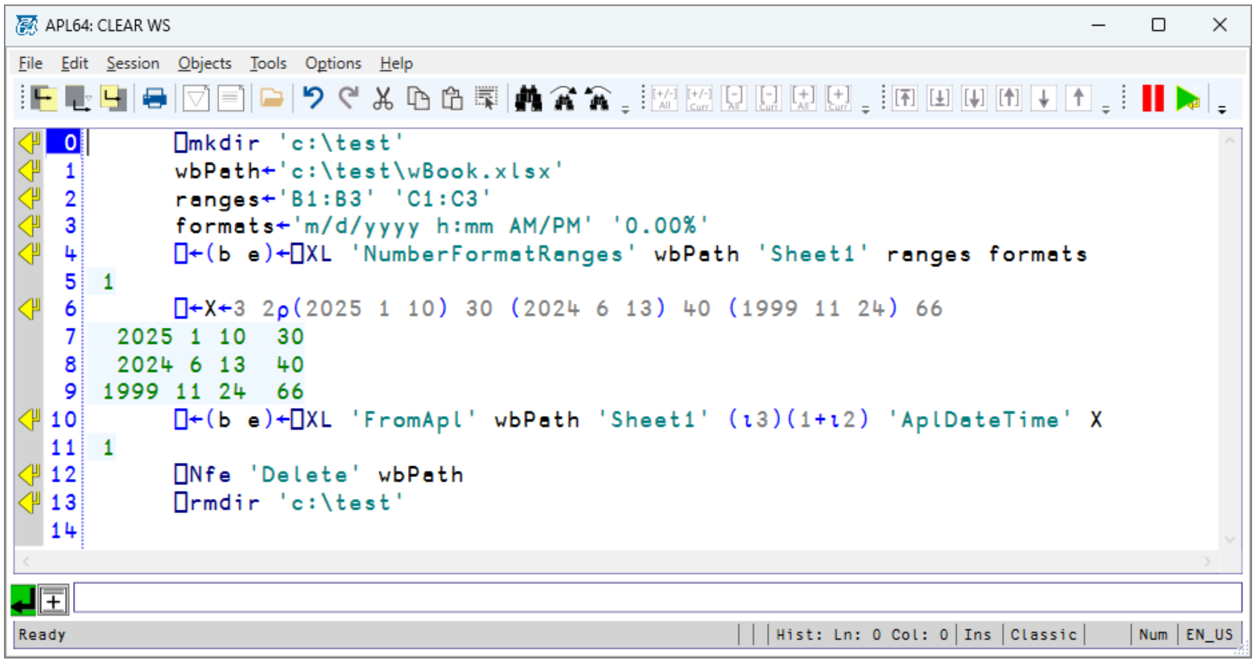
Each format is applied to the entire corresponding range

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

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

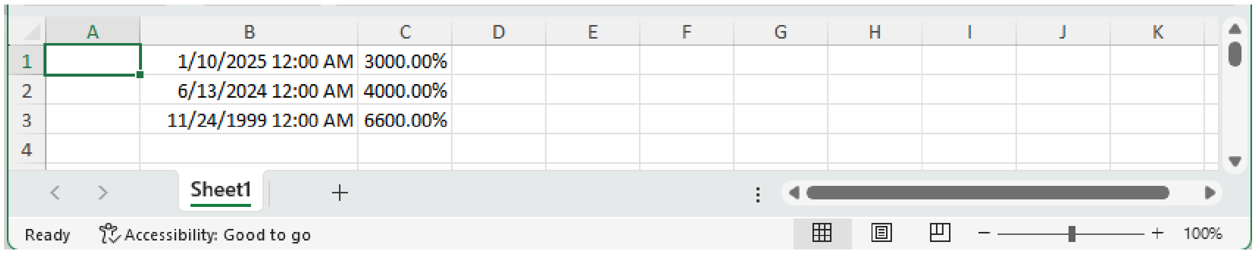
NumberFormatRanges Example

```
⎕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 2ρ(2025 1 10) 30 (2024 6 13) 40 (1999 11 24) 66
⎕←(b e)←⎕XL 'FromApl' wbPath 'Sheet1' (ι3)(1+ι2) 'AplDateTime' X
⎕Nfe 'Delete' wbPath
⎕rmdir 'c:\test'
```



```
      ⎕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
1
      ⎕←X←3 2⍴(2025 1 10) 30 (2024 6 13) 40 (1999 11 24) 66
  2025 1 10   30
  2024 6 13   40
 1999 11 24   66
      ⎕←(b e)←⎕XL 'FromApl' wbPath 'Sheet1' (⍳3)(1+⍳2) 'AplDateTime' X
1
      ⎕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 RemoveHyperling 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.

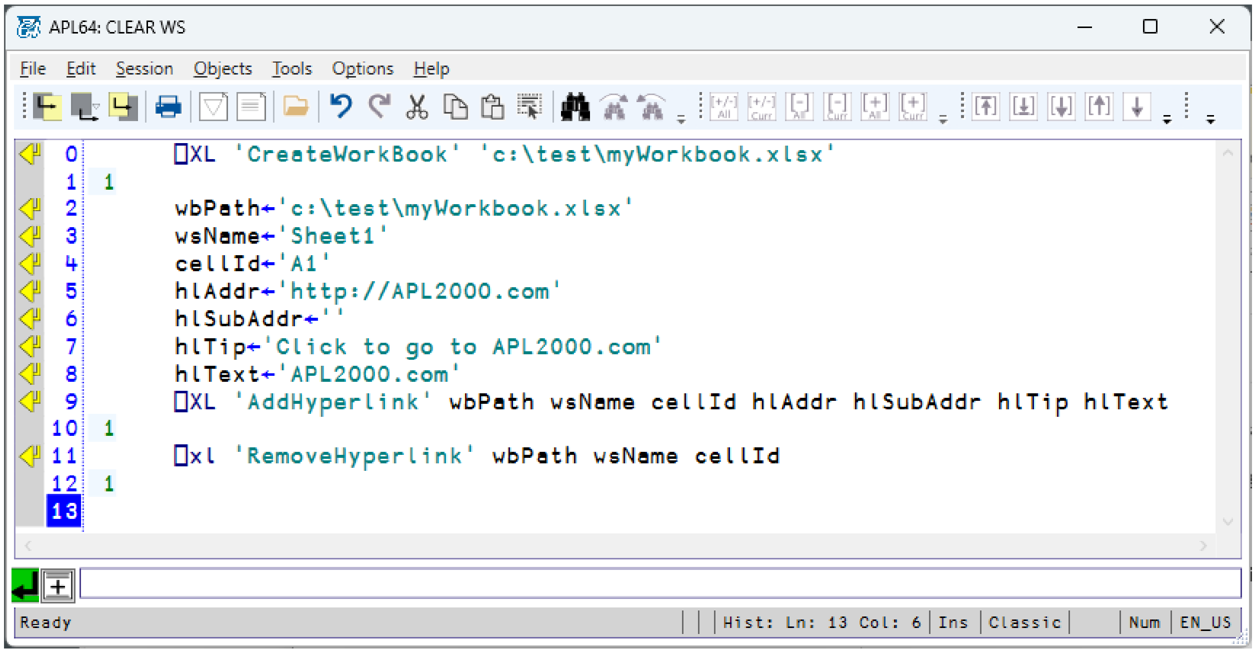
wsName: Name of the worksheet in the target workbook. The worksheet must exist

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
```



```
      ⎕XL 'CreateWorkBook' 'c:\test\myWorkbook.xlsx'
1
      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
1
      ⎕xl 'RemoveHyperlink' wbPath wsName cellId
1
```

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.

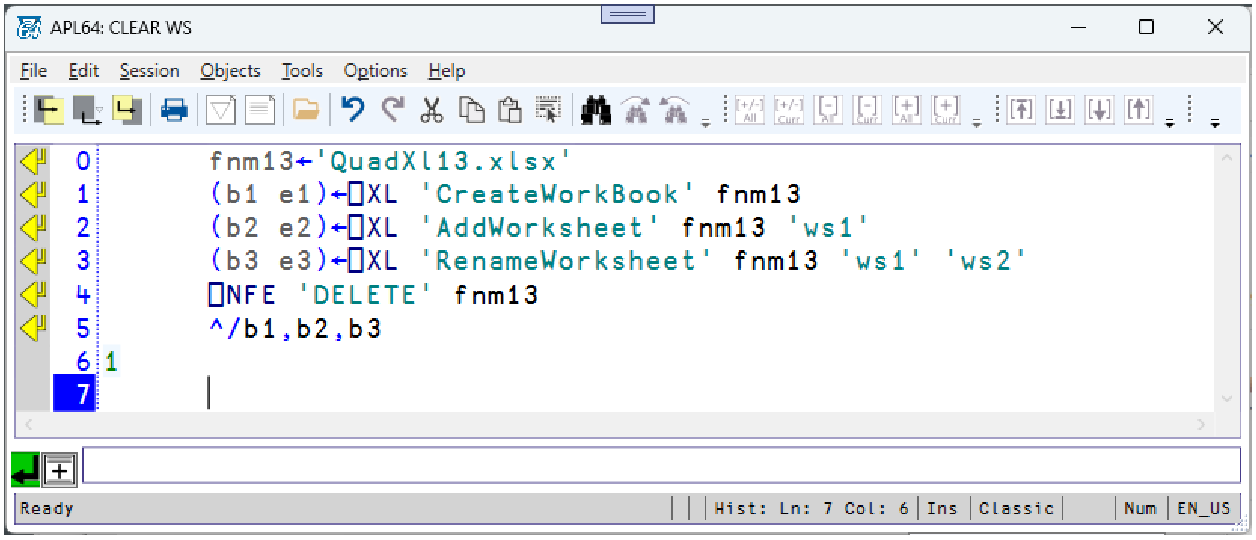
srcWsName: APL64 text object containing the name of the specified existing worksheet in the source workbook.

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

RenameWorksheet Example

```
fnm13←'QuadXl13.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
```



```
      fnm13←'QuadXl13.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
1
```

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

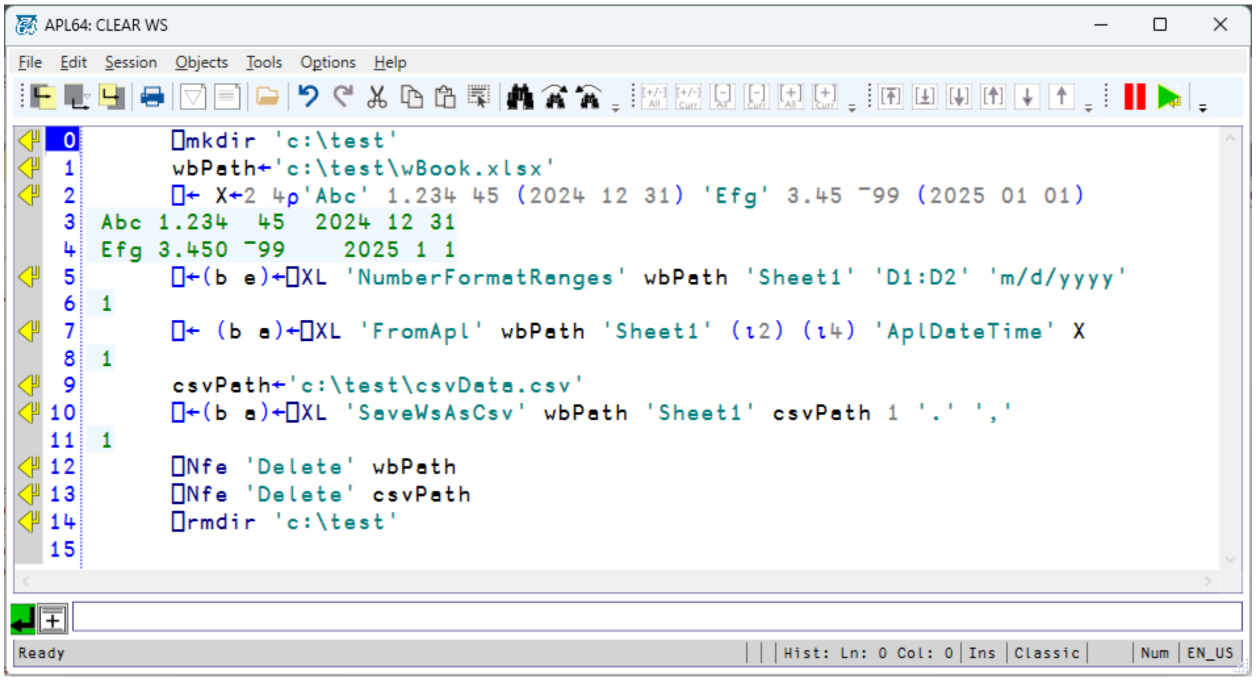
The target csv-format file will be created if it does not exist.

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

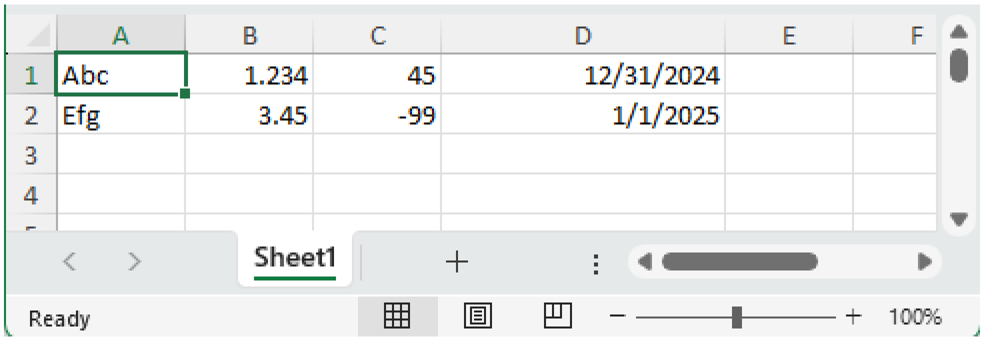
SaveWorksheetAsCsv with EN-US Decimal and Field Separators Example

```
⎕mkdir 'c:\test'
wbPath←'c:\test\wBook.xlsx'
⎕← X←2 4⍴'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' (⍳2) (⍳4) 'AplDateTime' X
csvPath←'c:\test\csvData.csv'
⎕←(b a)←⎕XL 'SaveWsAsCsv' wbPath 'Sheet1' csvPath 1 '.' ','
⎕Nfe 'Delete' wbPath
⎕Nfe 'Delete' csvPath
⎕rmdir 'c:\test'
```



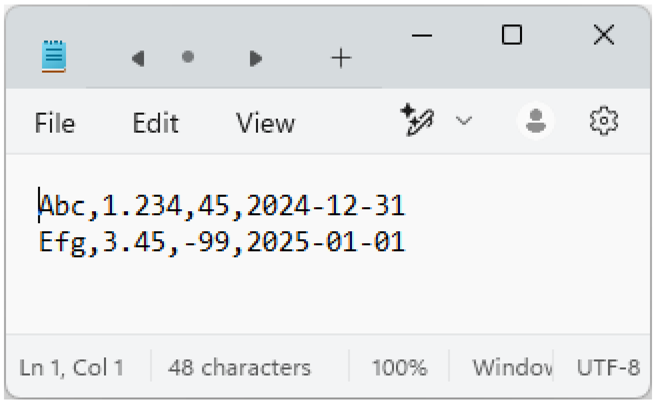
```
      ⎕mkdir 'c:\test'
      wbPath←'c:\test\wBook.xlsx'
      ⎕← X←2 4⍴'Abc' 1.234 45 (2024 12 31) 'Efg' 3.45 ¯99 (2025 01 01)
Abc 1.234  45  2024 12 31
Efg 3.450 ¯99     2025 1 1
      ⎕←(b e)←⎕XL 'NumberFormatRanges' wbPath 'Sheet1' 'D1:D2' 'm/d/yyyy'
1
      ⎕← (b a)←⎕XL 'FromApl' wbPath 'Sheet1' (⍳2) (⍳4) 'AplDateTime' X
1
      csvPath←'c:\test\csvData.csv'
      ⎕←(b a)←⎕XL 'SaveWsAsCsv' wbPath 'Sheet1' csvPath 1 '.' ','
1
      ⎕Nfe 'Delete' wbPath
      ⎕Nfe 'Delete' csvPath
      ⎕rmdir 'c:\test'
```

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						

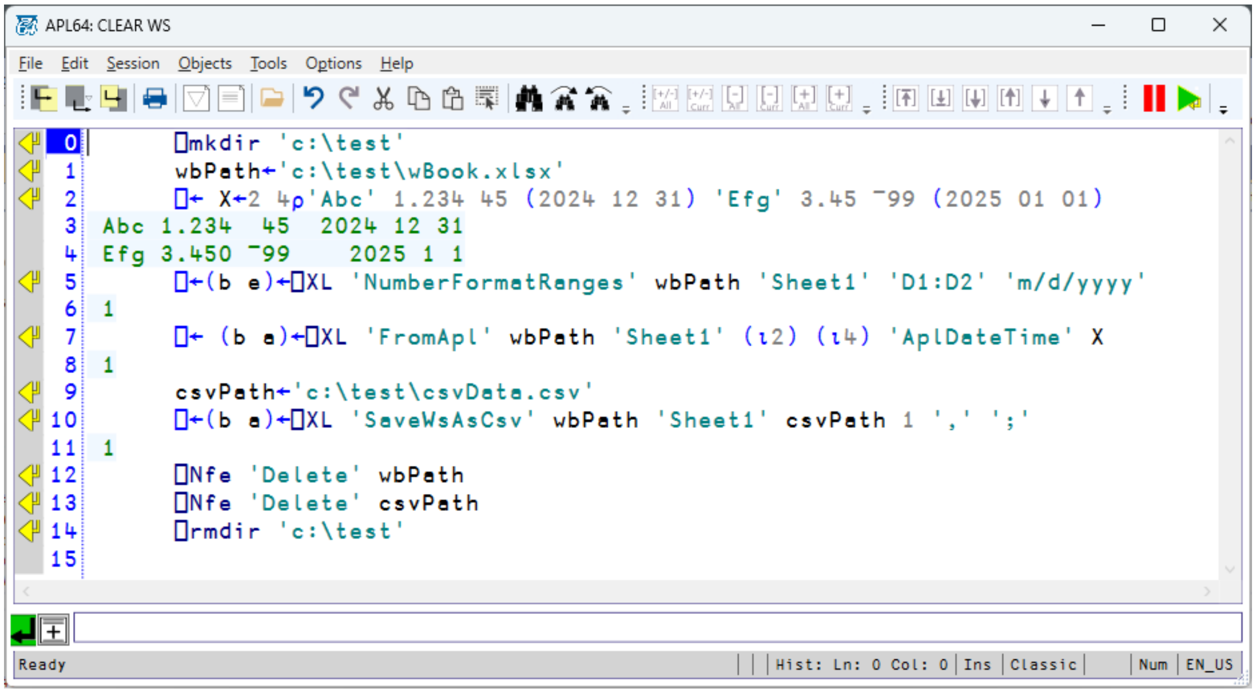
Resulting csv-format file:



```
Abc,1.234,45,2024-12-31
Efg,3.45,-99,2025-01-01
```

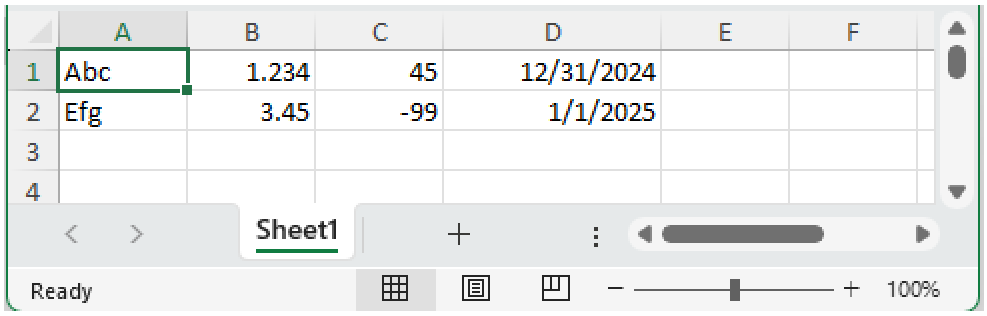
SaveWorksheetAsCsv with non-EN-US Decimal and Field Separators Example

```
⎕mkdir 'c:\test'
wbPath←'c:\test\wBook.xlsx'
⎕← X←2 4⍴'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' (⍳2) (⍳4) 'AplDateTime' X
csvPath←'c:\test\csvData.csv'
⎕←(b a)←⎕XL 'SaveWsAsCsv' wbPath 'Sheet1' csvPath 1 ',' ';'
⎕Nfe 'Delete' wbPath
⎕Nfe 'Delete' csvPath
⎕rmdir 'c:\test'
```



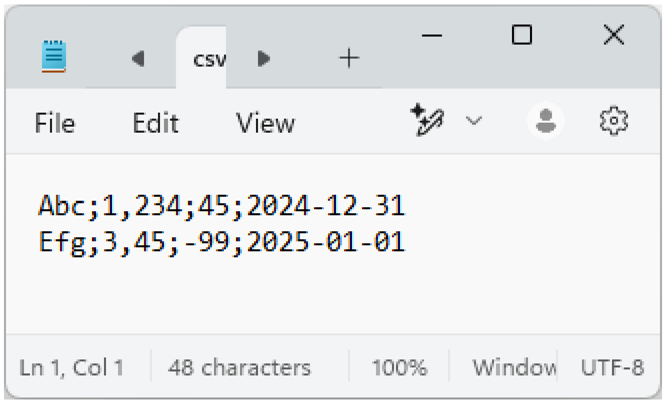
```
      ⎕mkdir 'c:\test'
      wbPath←'c:\test\wBook.xlsx'
      ⎕← X←2 4⍴'Abc' 1.234 45 (2024 12 31) 'Efg' 3.45 ¯99 (2025 01 01)
 Abc 1.234  45   2024 12 31
 Efg 3.450 ¯99     2025 1 1
      ⎕←(b a)←⎕XL 'NumberFormatRanges' wbPath 'Sheet1' 'D1:D2' 'm/d/yyyy'
 1
      ⎕← (b a)←⎕XL 'FromApl' wbPath 'Sheet1' (⍳2) (⍳4) 'AplDateTime' X
 1
      csvPath←'c:\test\csvData.csv'
      ⎕←(b a)←⎕XL 'SaveWsAsCsv' wbPath 'Sheet1' csvPath 1 ',' ';'
 1
      ⎕Nfe 'Delete' wbPath
      ⎕Nfe 'Delete' csvPath
      ⎕rmdir 'c:\test'
```

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						

Resulting csv-format file:



```
Abc;1,234;45;2024-12-31
Efg;3,45;-99;2025-01-01
```

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

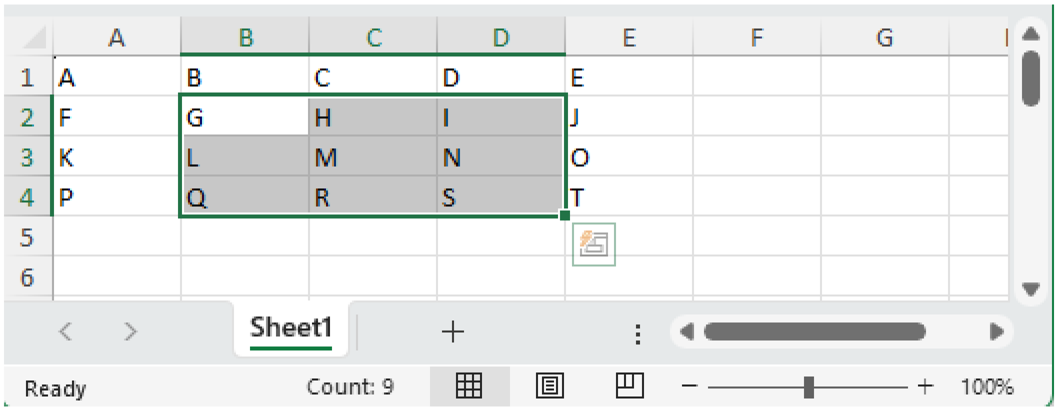
aRes: OK: ''

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:

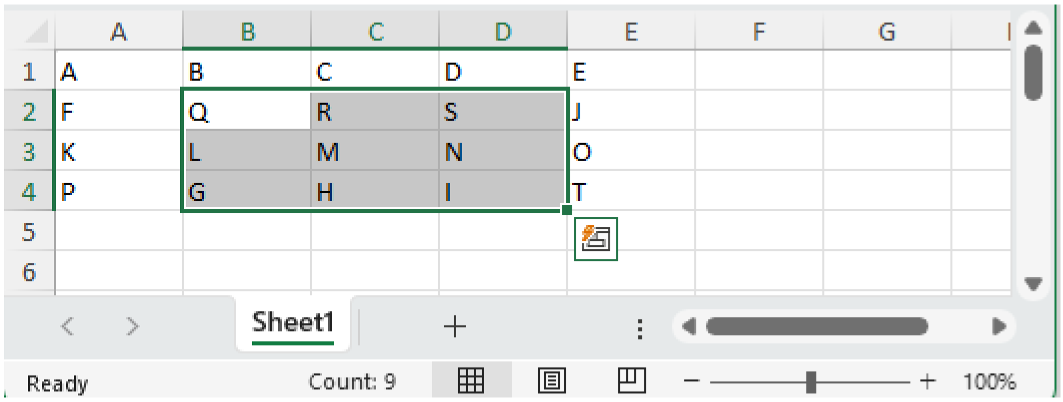
```
⎕mkdir 'c:\test'
wbPath←'c:\test\wBook.xlsx'
⎕←X←4 5⍴⎕a
⎕← (b a)←⎕XL 'FromApl' wbPath 'Sheet1' (⍳4) (⍳5) 'AplDateTime' X
rangeInfo←"B2:D4"
sortIsCaseSensitive←0
sortLevelsInfo←1 2⍴'B' 'D'
⎕←(b e)←⎕XL 'SortRange' wbPath 'Sheet1' rangeInfo sortIsCaseSensitive sortLevelsInfo
⎕Nfe 'Delete' wbPath
⎕rmdir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

Worksheet with selected range prior to sorting:

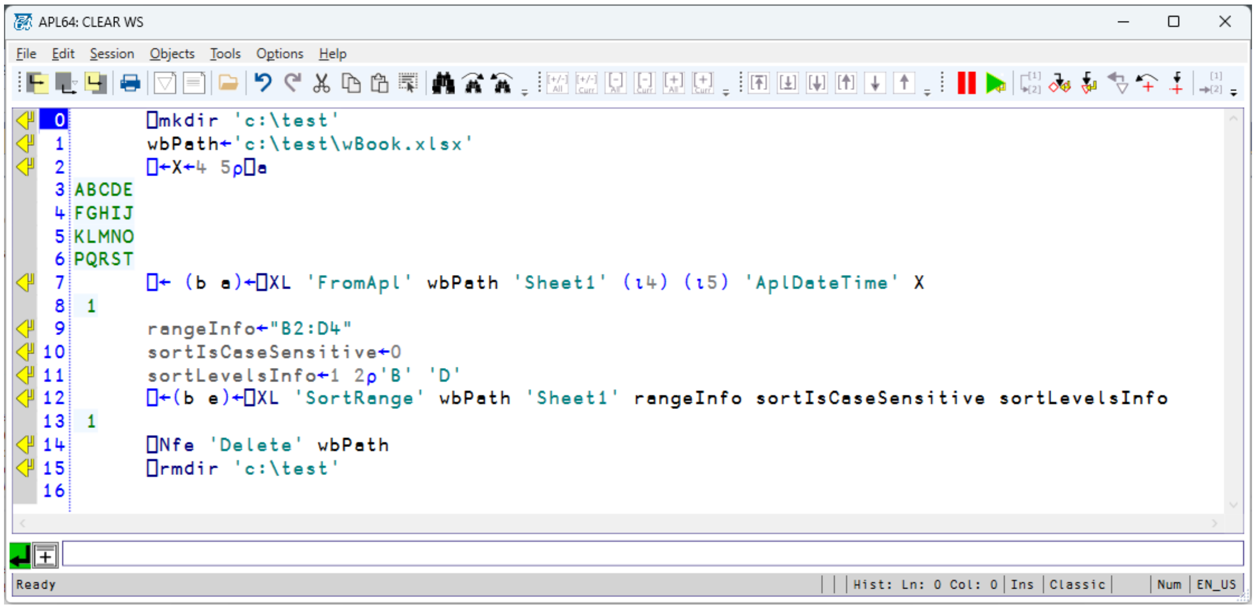


	A	B	C	D	E	F	G
1	A	B	C	D	E		
2	F	G	H	I	J		
3	K	L	M	N	O		
4	P	Q	R	S	T		
5							
6							

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							



```
      ⎕mkdir 'c:\test'
      wbPath←'c:\test\wBook.xlsx'
      ⎕←X←4 5⍴⎕a
ABCDE
FGHIJ
KLMNO
PQRST
      ⎕← (b a)←⎕XL 'FromApl' wbPath 'Sheet1' (⍳4) (⍳5) 'AplDateTime' X
 1
      rangeInfo←"B2:D4"
      sortIsCaseSensitive←0
      sortLevelsInfo←1 2⍴'B' 'D'
      ⎕←(b e)←⎕XL 'SortRange' wbPath 'Sheet1' rangeInfo sortIsCaseSensitive sortLevelsInfo
 1
      ⎕Nfe 'Delete' wbPath
      ⎕rmdir 'c:\test'
```

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.

inclRows: APL64 integer vector of the rows of the worksheet to fill with the APL64 array data.

inclCols: APL64 integer vector of the column of the worksheet to fill with the APL64 array data.

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

Row/Col Order and ⎕IO is significant.

bRes: 0/Fail, 1/OK

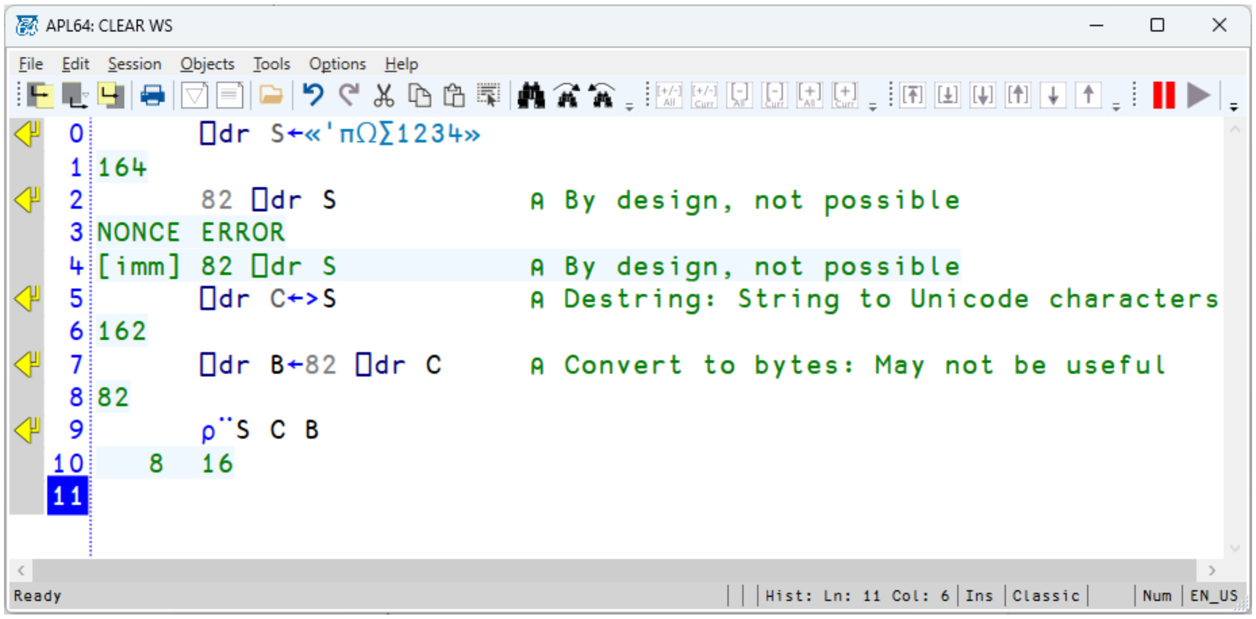
aRes: Fail: ErrMsg (APL64 character vector)/Fail

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 destring 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 'πΩ∑1234' 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:



```
      ⎕dr S←«'πΩ∑1234»
164
      82 ⎕dr S            ⍝ By design, not possible
NONCE ERROR
[imm] 82 ⎕dr S            ⍝ By design, not possible
      ⎕dr C←>S            ⍝ Destring: String to Unicode characters
162
      ⎕dr B←82 ⎕dr C      ⍝ Convert to bytes: May not be useful
82
      ⍴¨S C B
   8  16
```

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: Excel A1-style range, e.g. 'Z1' or 'A5:C8'

inclCols: APL64 integer vector of the column of the worksheet to fill with the APL64 array data.

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

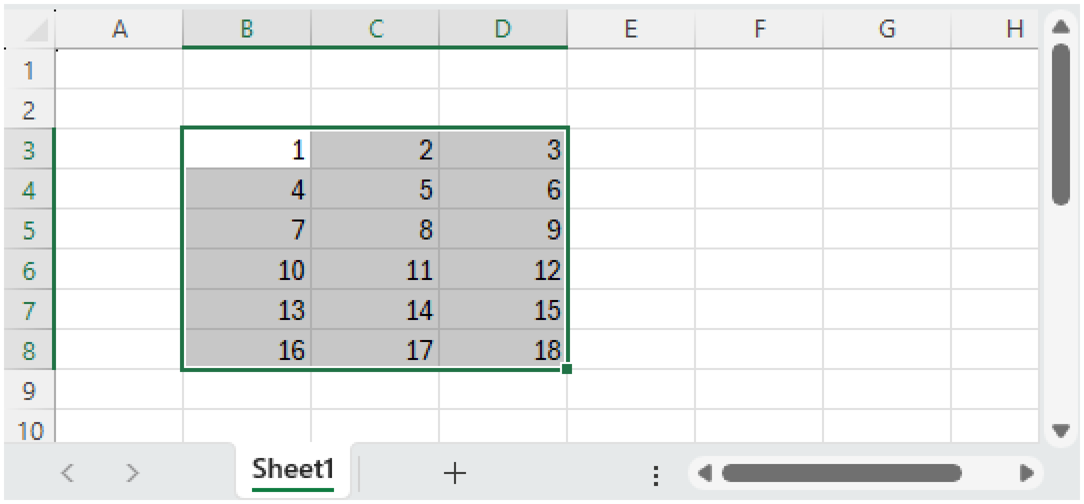
aRes: OK: APL64 data array containing information obtained from specified cells of the target Excel worksheet.

ToAPLFromRange Example

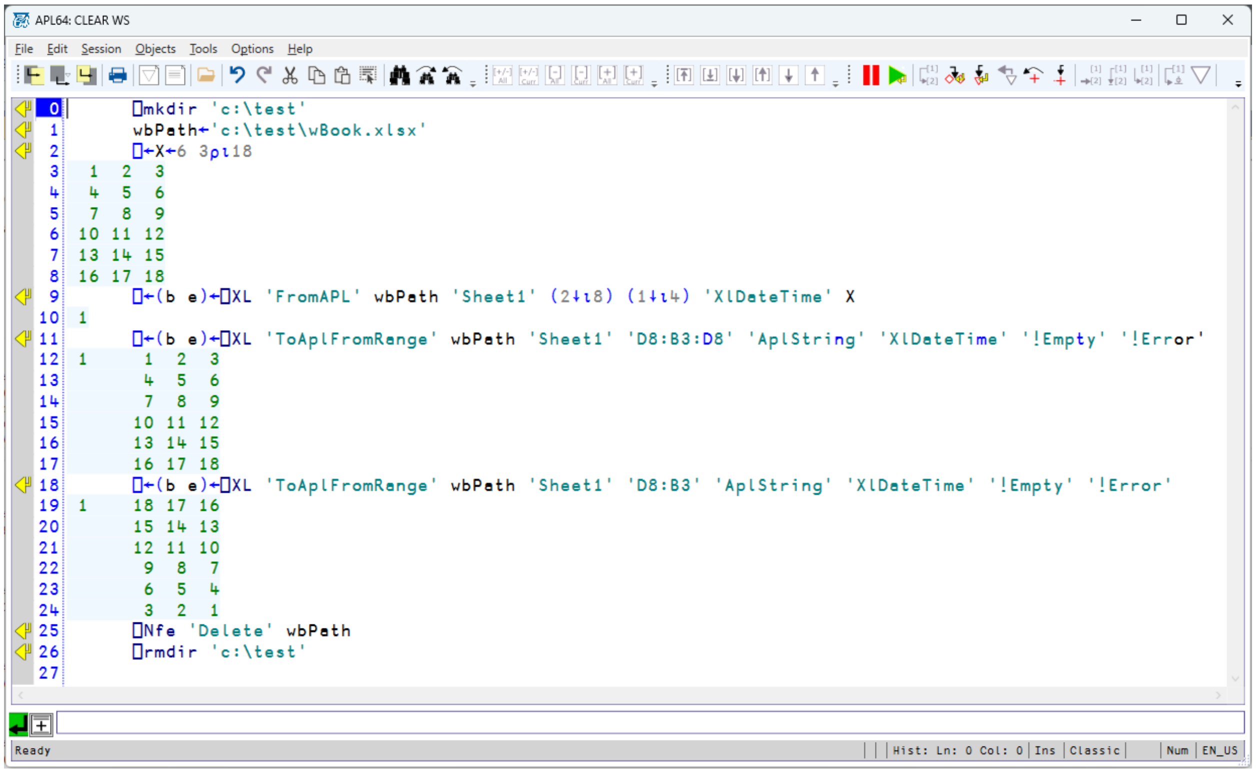
Create a sample worksheet using the FromApl action, and obtain selected values from the worksheet using the ToAPLFromRange action:

```
⎕mkdir 'c:\test'
wbPath←'c:\test\wBook.xlsx'
⎕←X←6 3ρι18
⎕←(b e)←⎕XL 'FromAPL' wbPath 'Sheet1' (2↓ι8) (1↓ι4) 'XlDateTime' X
⎕←(b e)←⎕XL 'ToAplFromRange' wbPath 'Sheet1' 'D8:B3:D8' 'AplString' 'XlDateTime' '!Empty' '!Error'
⎕←(b e)←⎕XL 'ToAplFromRange' wbPath 'Sheet1' 'D8:B3' 'AplString' 'XlDateTime' '!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								



```
      ⎕mkdir 'c:\test'
      wbPath←'c:\test\wBook.xlsx'
      ⎕←X←6 3⍴⍳18
 1  2  3
 4  5  6
 7  8  9
10 11 12
13 14 15
16 17 18
      ⎕←(b e)←⎕XL 'FromAPL' wbPath 'Sheet1' (2↓⍳8) (1↓⍳4) 'XlDateTime' X
1
      ⎕←(b e)←⎕XL 'ToAplFromRange' wbPath 'Sheet1' 'D8:B3:D8' 'AplString' 'XlDateTime' '!Empty' '!Error'
1       1  2  3
        4  5  6
        7  8  9
       10 11 12
       13 14 15
       16 17 18
      ⎕←(b e)←⎕XL 'ToAplFromRange' wbPath 'Sheet1' 'D8:B3' 'AplString' 'XlDateTime' '!Empty' '!Error'
1      18 17 16
       15 14 13
       12 11 10
        9  8  7
        6  5  4
        3  2  1
      ⎕Nfe 'Delete' wbPath
      ⎕rmdir 'c:\test'
```

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.

inclRows: APL64 integer vector of the rows of the csv-format file to fill with the APL64 array data.

inclCols: APL64 integer vector of the number of fields of each row of the csv-format file to fill with the APL64 array data.

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

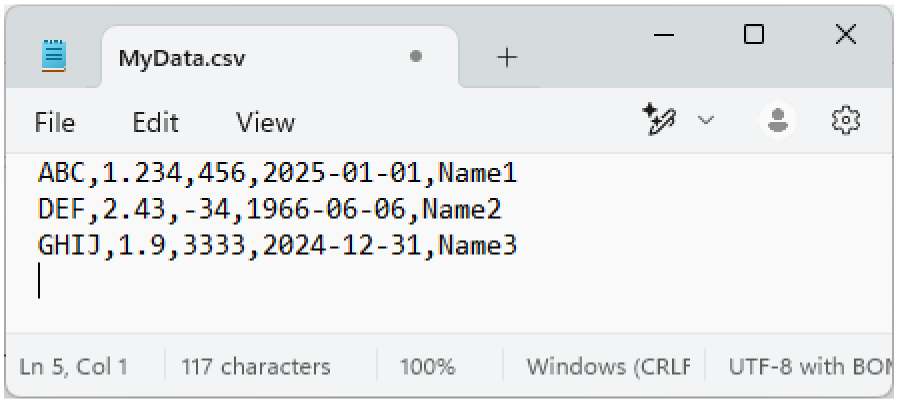
aRes: Fail: ErrMsg (APL64 character vector)/Fail

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

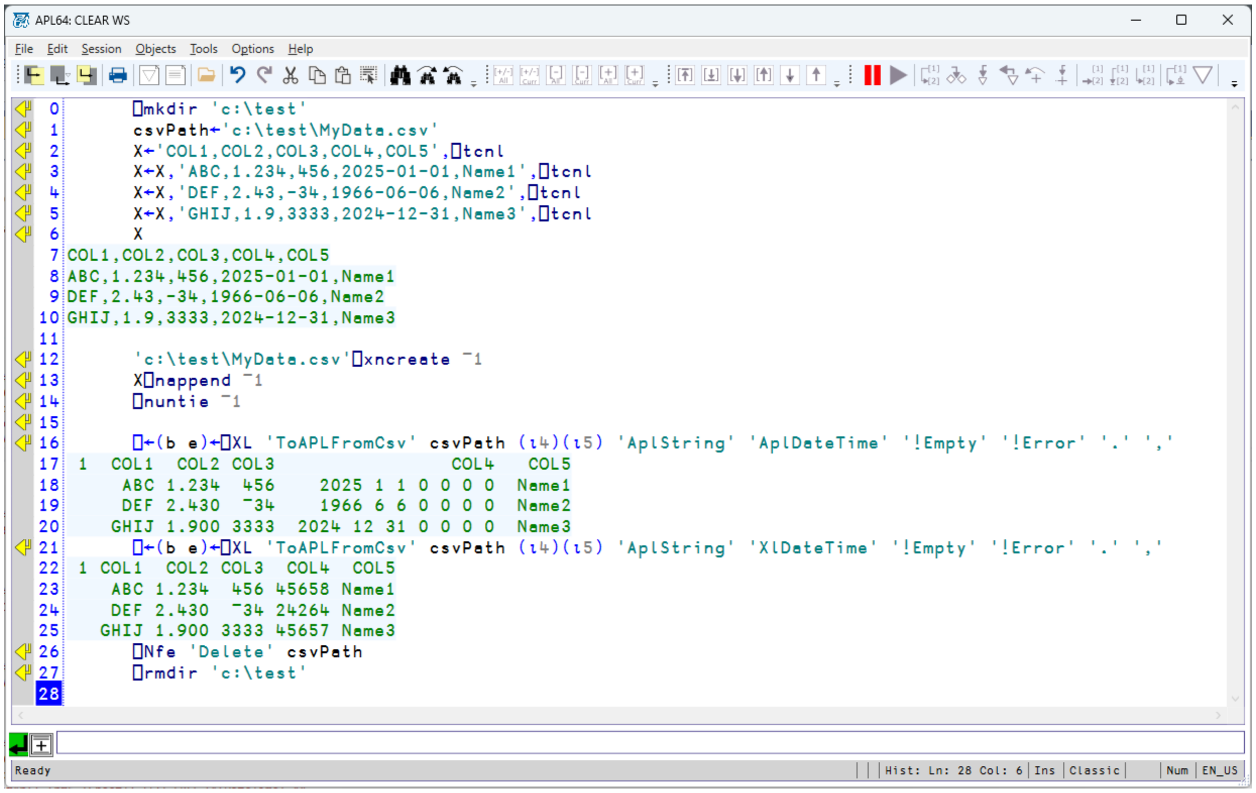
ToAplFromCsv Example Csv with EN-US Delimiters and Date Information

```
⎕mkdir 'c:\test'
      csvPath←'c:\test\MyData.csv'
      X←'COL1,COL2,COL3,COL4,COL5',⎕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
      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'⎕xncreate ¯1
X⎕nappend ¯1
⎕nuntie ¯1
⎕←(b e)←⎕XL 'ToAPLFromCsv' csvPath (⍳4)(⍳5) 'AplString' 'AplDateTime' '!Empty' '!Error' '.' ','
⎕←(b e)←⎕XL 'ToAPLFromCsv' csvPath (⍳4)(⍳5) 'AplString' 'XlDateTime' '!Empty' '!Error' '.' ','
⎕Nfe 'Delete' csvPath
⎕rmdir 'c:\test'
```

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



```
ABC,1.234,456,2025-01-01,Name1
DEF,2.43,-34,1966-06-06,Name2
GHIJ,1.9,3333,2024-12-31,Name3
```



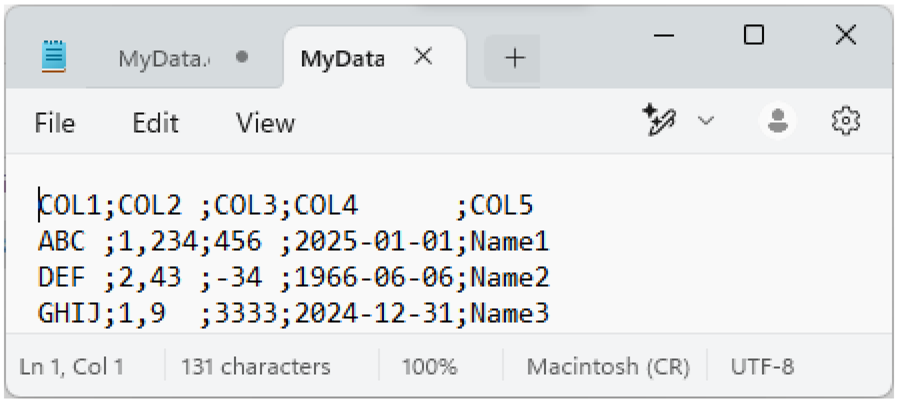
```
      ⎕mkdir 'c:\test'
      csvPath←'c:\test\MyData.csv'
      X←'COL1,COL2,COL3,COL4,COL5',⎕tcnl
      X←X,'ABC,1.234,456,2025-01-01,Name1',⎕tcnl
      X←X,'DEF,2.43,-34,1966-06-06,Name2',⎕tcnl
      X←X,'GHIJ,1.9,3333,2024-12-31,Name3',⎕tcnl
      X
COL1,COL2,COL3,COL4,COL5
ABC,1.234,456,2025-01-01,Name1
DEF,2.43,-34,1966-06-06,Name2
GHIJ,1.9,3333,2024-12-31,Name3

      'c:\test\MyData.csv'⎕xncreate ¯1
      X⎕nappend ¯1
      ⎕nuntie ¯1

      ⎕←(b e)←⎕XL 'ToAPLFromCsv' csvPath (⍳4)(⍳5) 'AplString' 'AplDateTime' '!Empty' '!Error' '.' ','
 1   COL1  COL2 COL3                      COL4    COL5
      ABC 1.234  456     2025 1 1 0 0 0 0   Name1
      DEF 2.430  ¯34     1966 6 6 0 0 0 0   Name2
     GHIJ 1.900 3333  2024 12 31 0 0 0 0   Name3
      ⎕←(b e)←⎕XL 'ToAPLFromCsv' csvPath (⍳4)(⍳5) 'AplString' 'XlDateTime' '!Empty' '!Error' '.' ','
 1 COL1  COL2 COL3  COL4  COL5
     ABC 1.234  456 45658 Name1
     DEF 2.430  ¯34 24264 Name2
    GHIJ 1.900 3333 45657 Name3
      ⎕Nfe 'Delete' csvPath
      ⎕rmdir 'c:\test'
```

ToAplFromCsv Example Csv with non-EN-US Delimiters and Date Information

Comma for decimal separator and semi-colon for field separator

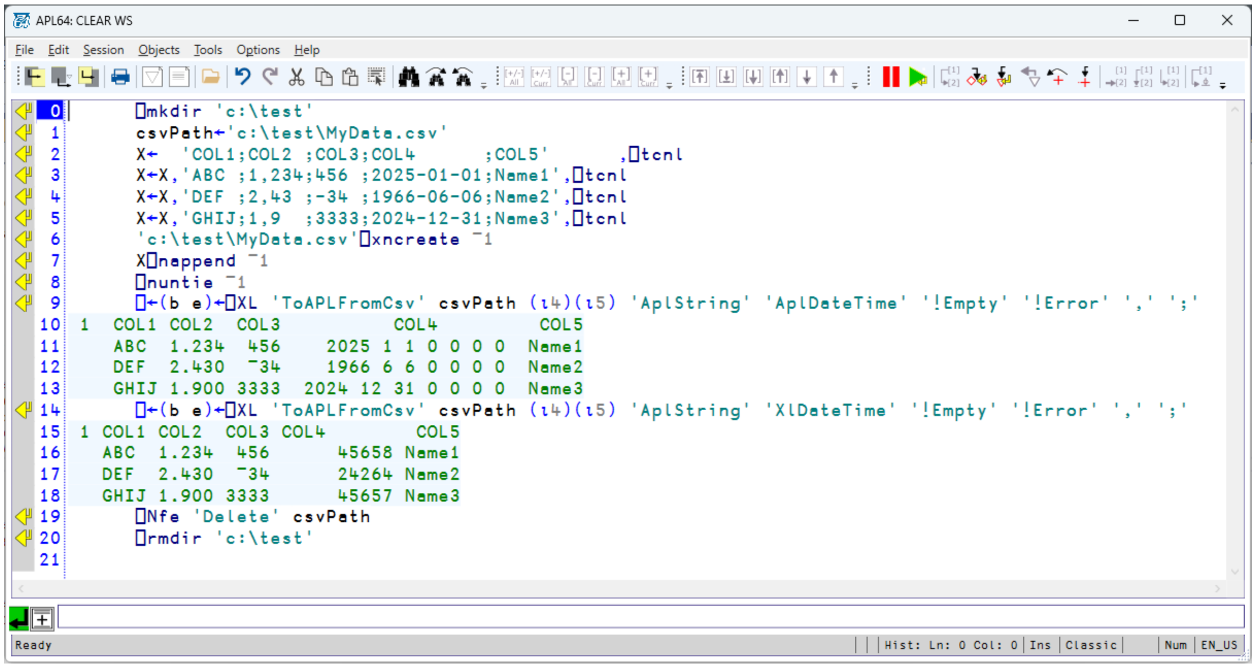
```
⎕mkdir 'c:\test'
csvPath←'c:\test\MyData.csv'
X← 'COL1;COL2 ;COL3;COL4     ;COL5'     ,⎕tcnl
X←X,'ABC ;1,234;456 ;2025-01-01;Name1',⎕tcnl
X←X,'DEF ;2,43 ;-34 ;1966-06-06;Name2',⎕tcnl
X←X,'GHIJ;1,9  ;3333;2024-12-31;Name3',⎕tcnl
'c:\test\MyData.csv'⎕xncreate ¯1
X⎕nappend ¯1
⎕nuntie ¯1
⎕←(b e)←⎕XL 'ToAPLFromCsv' csvPath (⍳4)(⍳5) 'AplString' 'AplDateTime' '!Empty' '!Error' ',' ';'
⎕←(b e)←⎕XL 'ToAPLFromCsv' csvPath (⍳4)(⍳5) 'AplString' 'XlDateTime' '!Empty' '!Error' ',' ';'
⎕Nfe 'Delete' csvPath
⎕rmdir 'c:\test'
```

Resulting Csv-format file:



```
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
```

Ln 1, Col 1 | 131 characters | 100% | Macintosh (CR) | UTF-8



```
      ⎕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 (⍳4)(⍳5) 'AplString' 'AplDateTime' '!Empty' '!Error' ',' ';'
 1  COL1 COL2  COL3           COL4          COL5
    ABC  1.234  456    2025 1  1 0 0 0 0  Name1
    DEF  2.430  ¯34    1966 6  6 0 0 0 0  Name2
    GHIJ 1.900 3333   2024 12 31 0 0 0 0  Name3
      ⎕←(b e)←⎕XL 'ToAPLFromCsv' csvPath (⍳4)(⍳5) 'AplString' 'XlDateTime' '!Empty' '!Error' ',' ';'
 1 COL1 COL2  COL3 COL4         COL5
   ABC  1.234  456        45658 Name1
   DEF  2.430  ¯34        24264 Name2
   GHIJ 1.900 3333        45657 Name3
      ⎕Nfe 'Delete' csvPath
      ⎕rmdir 'c:\test'
```

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