

Using the ☐ASKx System Functions

Table of Contents

Overview	1
☐ ASK	2
☐ ASKEX	2
☐ ASKL.....	2
☐ ASKN.....	2
☐ ASK Examples	3
☐ ASKEX Examples	4
☐ ASKL Example	7
☐ ASKN Examples	8
Multi-line Prompt for ☐ ASK, ☐ ASKN or ☐ ASKL	10
Sample APL64 Command Line Application	11

Overview

An APL-based command line application uses the APL64 developer version rather than a runtime version of APL64. The ☐ASKx system functions facilitate the development of command line applications which use the APL64 developer version with a GUI format employing independent field-level user queries.

Syntax:

APL64 CharVector	← [TextCaption] TextPrompt ☐ ASK	TextDefaultValue
APL64 Value	← [TextCaption] TextPrompt ☐ ASKEX	TextDefaultValue
APL64 IntegerValue	← [TextCaption] OptionsList ☐ ASKL	NumericDefaultValue
APL64 NumericValue	← [TextCaption] TextPrompt ☐ ASKN	NumericDefaultValue(s)

When an ☐ASKx dialog is presented, the default value is highlighted and contains the keyboard focus.

The ☐ASKx dialogs are modal. The user can close these dialogs clicking the close button in the dialog title bar, Enter/Return or Esc key.

The ☐ASKx dialogs are implemented without dependence on ☐WI Win32 technology.

The ☐ASKx dialogs are available only the APL64 developer version.

The caption argument is optional. If present, the caption value must be a character scalar, character vector or a string scalar.

A prompt argument is required. The prompt argument may be a character scalar, character vector, character matrix, or a string scalar.

☐ASK

☐ASK can be used to present a dialog with user defined text caption, text prompt and text default value. The result of ☐ASK is an APL64 character vector containing the user-edited default value. The default value must be a possibly empty, character scalar, character vector or a string scalar.

☐ASKEX

☐ASKEX can be used to present a dialog with user defined text caption, text prompt and text default value. The result of ☐ASKEX is a two-element vector:

Element #	Element Description
1	User-entered text
2	Interpreter evaluation of user-entered text

The default value must be a possibly empty, character scalar, character vector or a string scalar. When the user closes the ☐ASKEX dialog, the APL64 interpreter evaluates the user-provided text input to obtain the second element of the ☐ASKEX result. The user-provided text input for ☐ASKEX cannot be an empty text value. If the user-provided text value cannot be evaluated by the APL64 interpreter, an exception is thrown and the ☐ASKEX dialog is displayed again until the user-provided text input can be evaluated and produce a result.

☐ASKL

☐ASKL can be used to present a dialog with user defined text caption, options list and integer default selected item# (index origin zero). The result of ☐ASKL is an APL64 integer value containing the index (index origin zero) in the options list of the user-selected item. If the default value is outside of the range of available options, no option will be pre-selected when the ☐ASKL dialog is presented. The user must selection an available option to exit the ☐ASKL dialog. The default value must be an integer scalar.

The OptionsList argument of ☐ASKL must be A:

- Character scalar, vector or matrix or
- String scalar, string vector or string 2-column matrix or
- A vector with each element a character scalar, character vector or string scalar
- A 1-column matrix where each row is a character scalar, character vector or string scalar

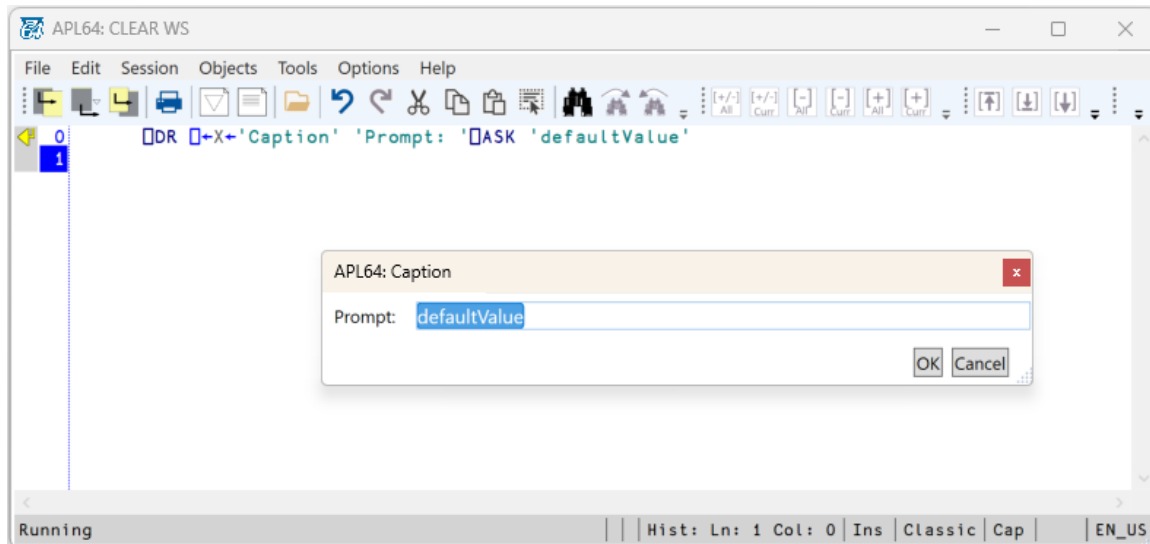
☐ASKN

☐ASKN can be used to present a dialog with user defined text caption, text prompt and a numeric scalar or vector (rank 1) default value. The result of ☐ASKN is an APL64 vector (rank 1) value containing the user-edited numeric value(s). The user must enter a valid numeric value or values to exit the ☐ASKN dialog. The default value or valuers must be numeric scalar or numeric vector or zilde.

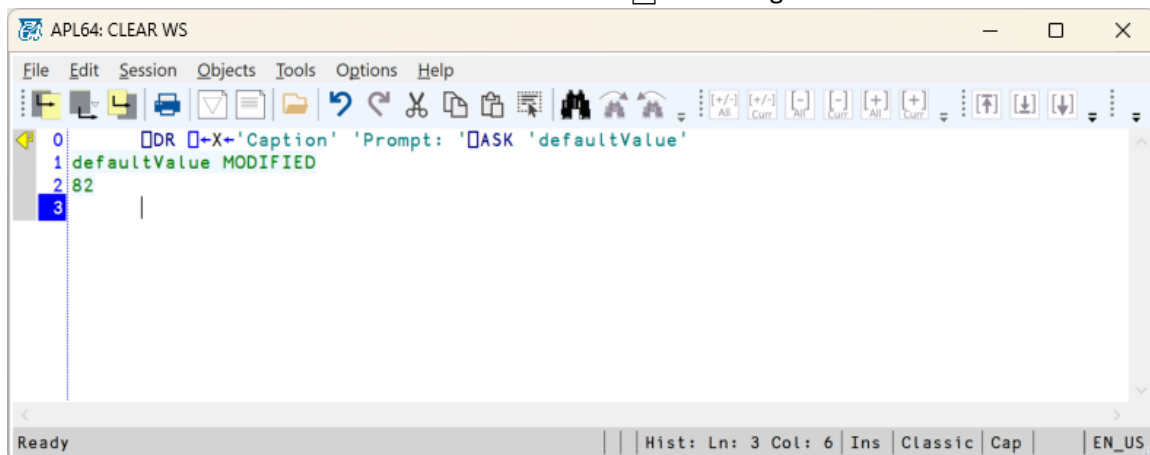
ASK Examples

```
 DR  ←X←'Caption' 'Prompt: 'ASK 'defaultValue'
```

Before the user closes the dialog:



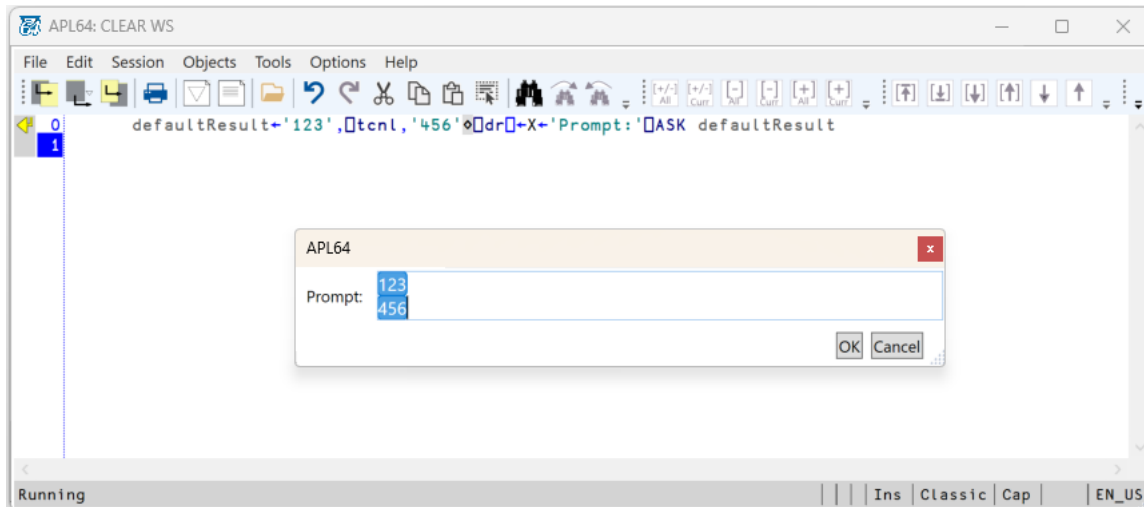
After the user edits the default value and closes the ASK dialog:



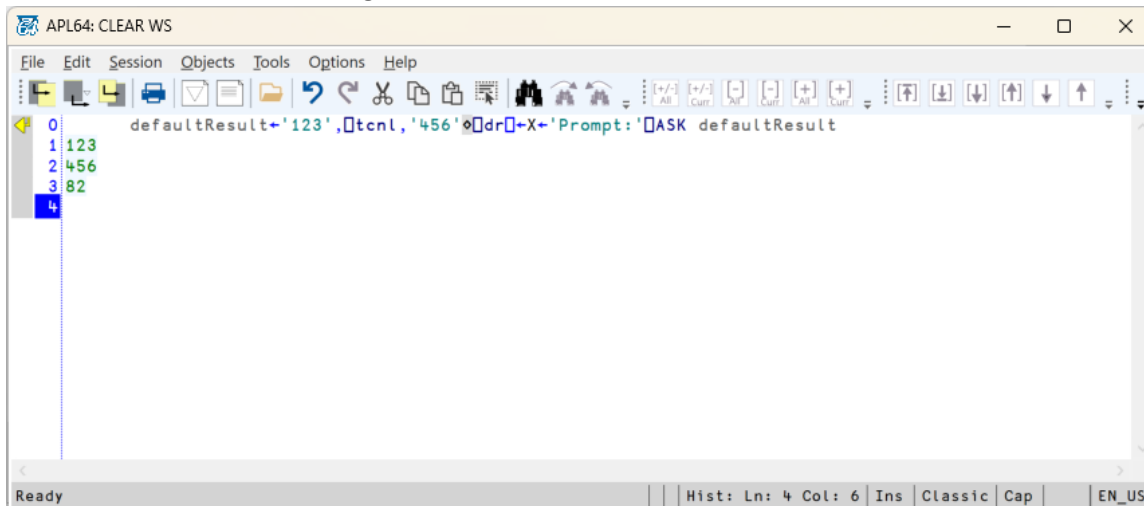
```
defaultResult←'123',tcnl,'456'◊dr←X←'Prompt:'ASK defaultResult
```

The default result (right argument of ASK) may contain tcnl characters, so that a multi-line default value is presented in the dialog.

Before the user closes the dialog:



After the user closes the dialog:

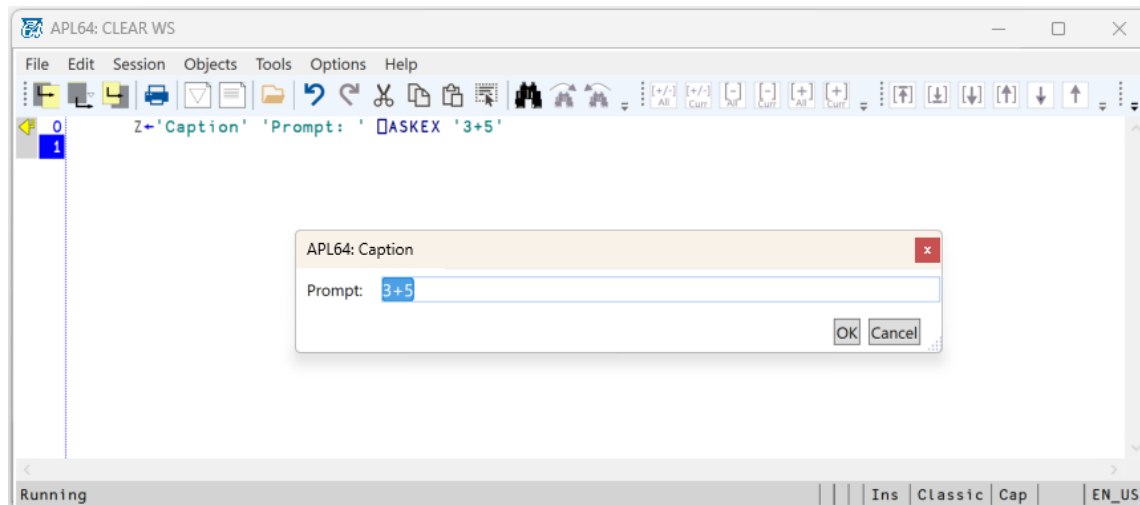


☐ ASKEX Examples

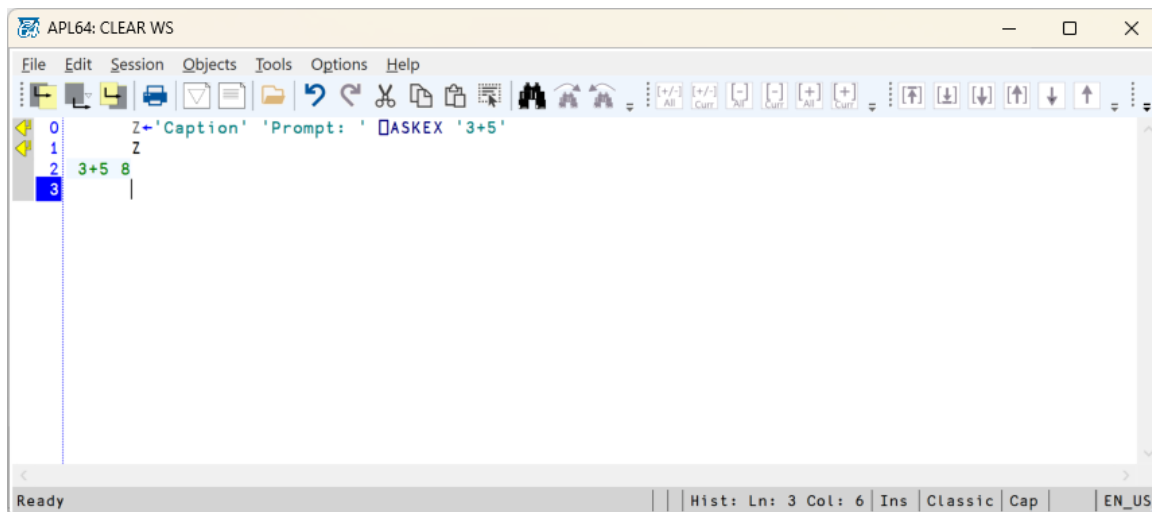
The user can enter APL64 executable expressions in the ☐ ASKEX dialog with a caption and prompt. If the user enters an invalid expression which cannot be executed, the prompt window is re-displayed until a valid user response, if provided.

☐ Z←'Caption' 'Prompt: ' ☐ ASKEX '3+5'

The ☐ ASKEX dialog is presented:



After the ☐ ASKEX dialog is closed:



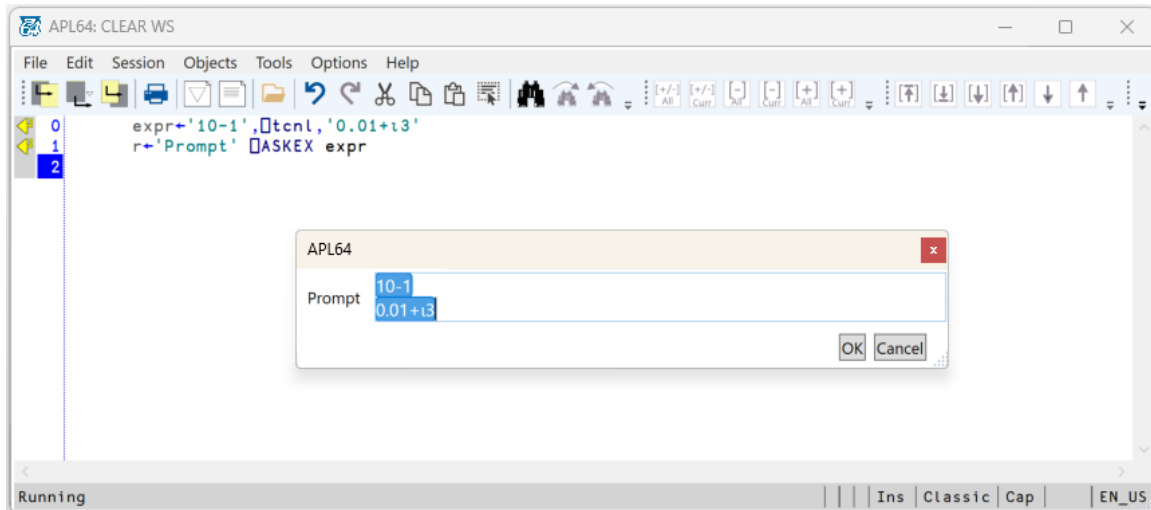
```

expr←'10-1',☐tcnl,'0.01+ι3'
r←'Prompt' ☐ASKEX expr
1⌈r
2⌈r
x←⌊ expr
x

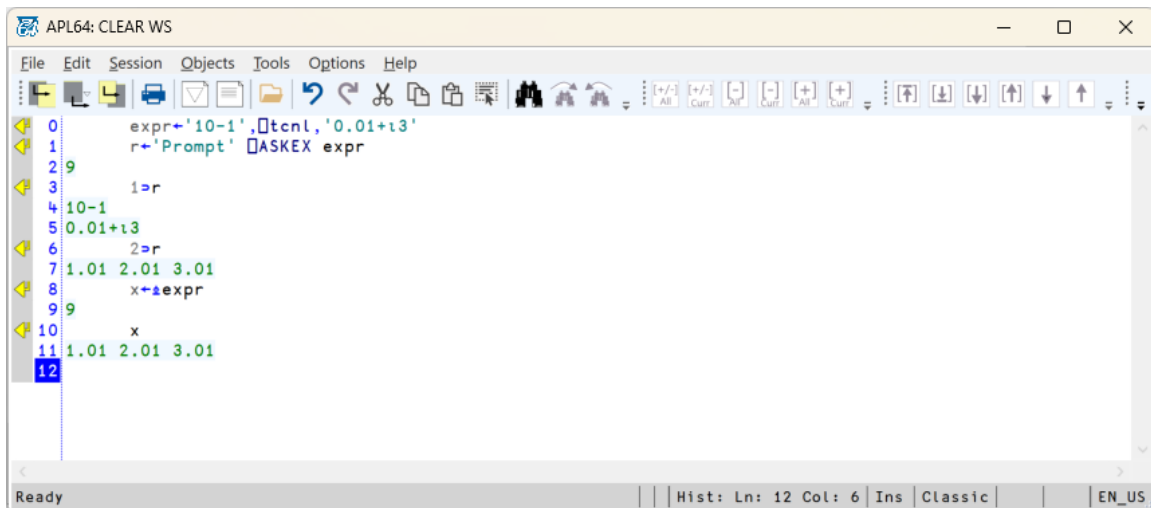
```

When the default response text of ☐ ASKEX contains a ☐tcnl character, a multi-line default response is presented in the dialog. The result of ☐ ASKEX is the same as if the response was executed using the APL execute (☐⌊) function.

Before the user closes the dialog:



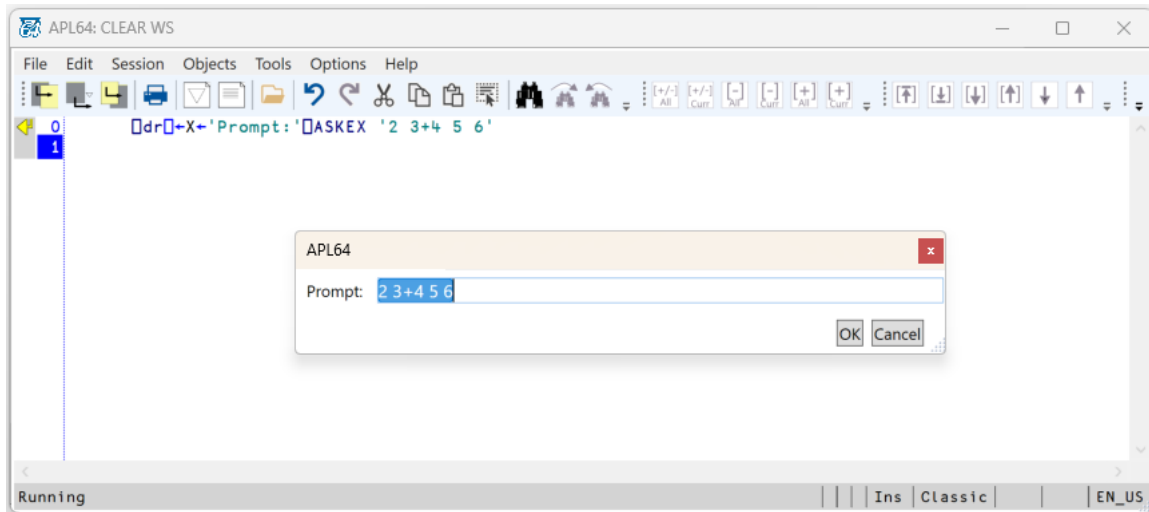
After the user closes the dialog:



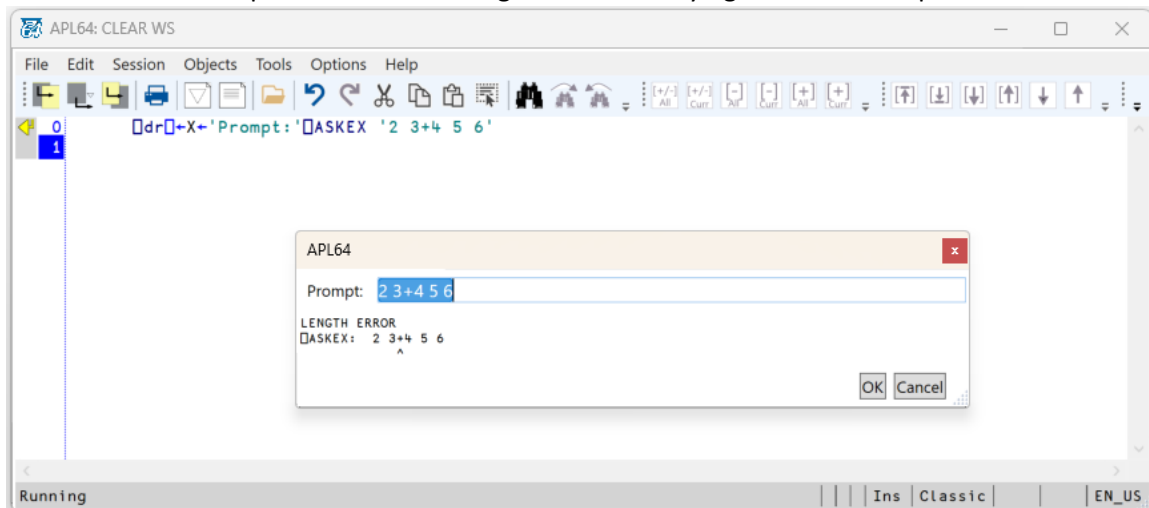
☐dr☐←X←'Prompt:'☐ASKEK '2 3+4 5 6'

In case the default response cannot be evaluated by the interpreter, an exception message is presented in the dialog. The dialog requires the user to enter a response which can be successfully evaluated by the interpreter:

Before the dialog is closed:



After the user attempts to close the dialog without modifying the default response:



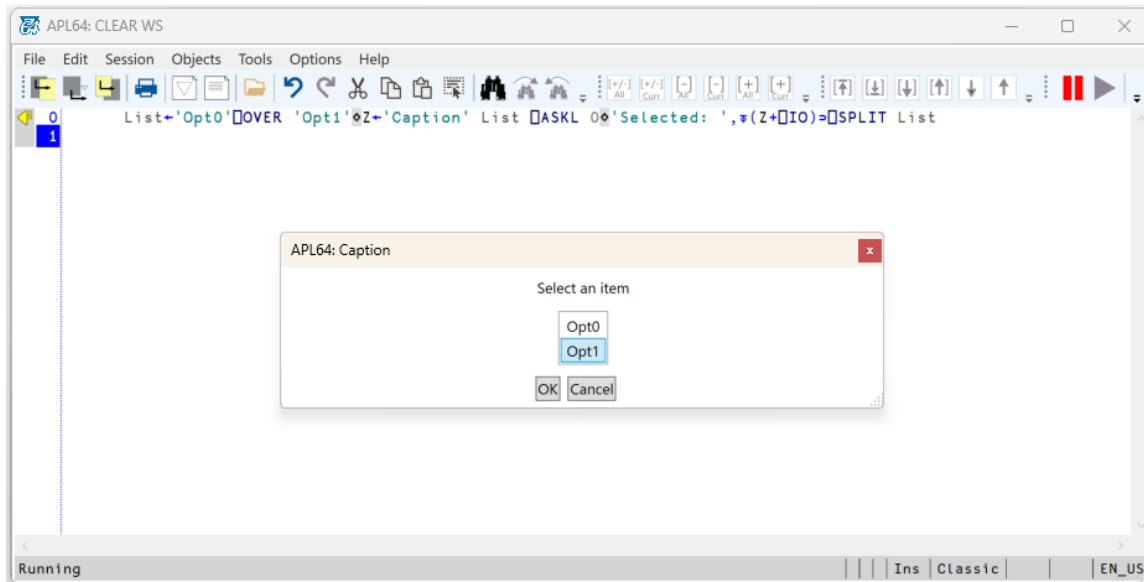
ASKL Example

Show selected list item in history:

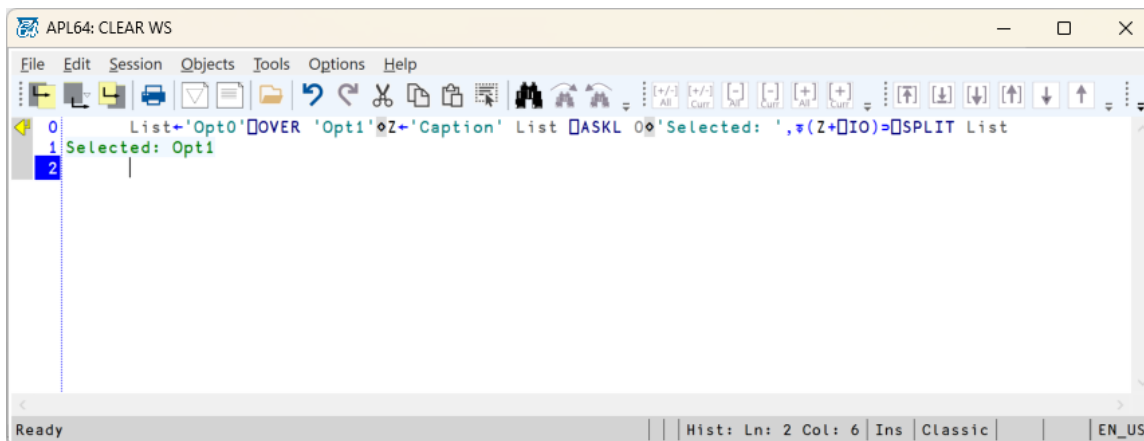
```
List←'Opt0'OVER 'Opt1'◊Z←'Caption' List ASKL 0◊'Selected: ',⌈(Z+IO)⌉SPLIT List
```

Caption is a character vector, list is a character matrix and the default list index is 0 (index origin 0):

Before user clicks [Enter/Return]:



After user clicks [Enter/Return]:

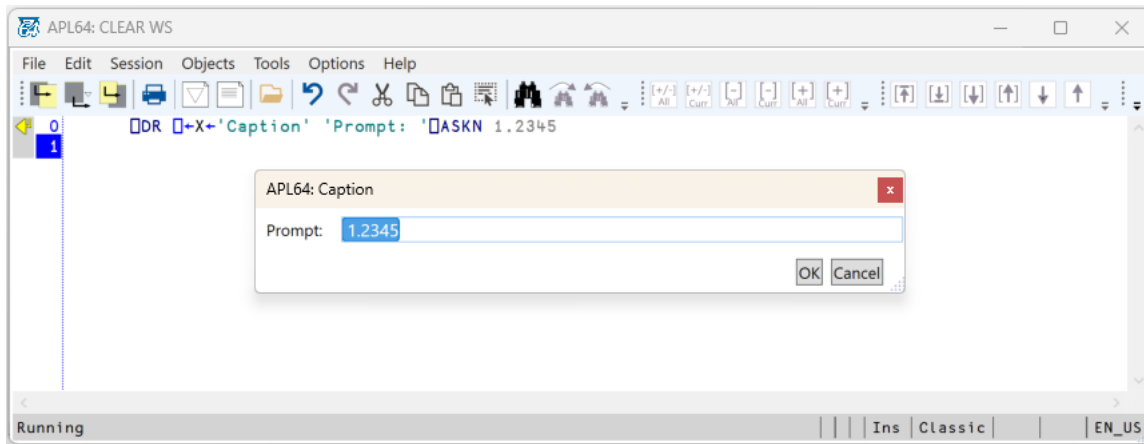


ASKN Examples

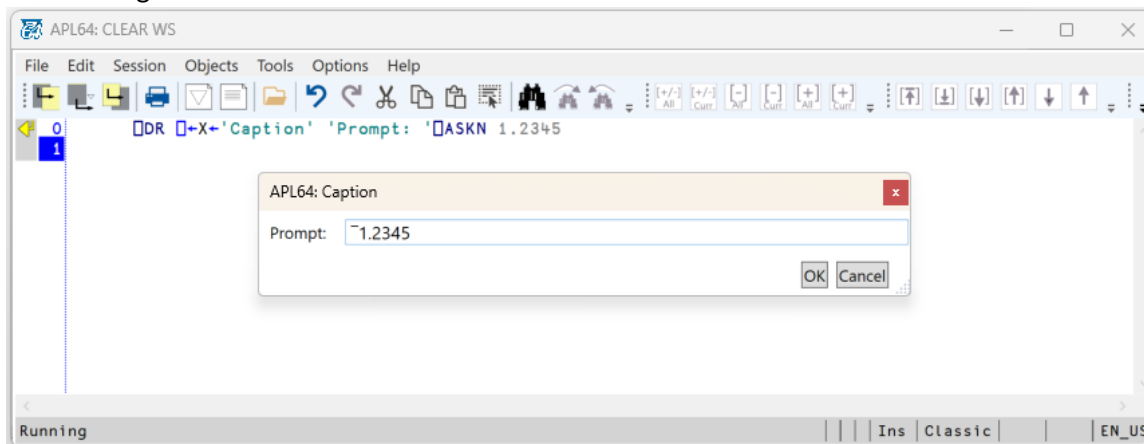
ASKN with numeric scalar of type double

In this example the user modifies the default value and clicked the Enter key

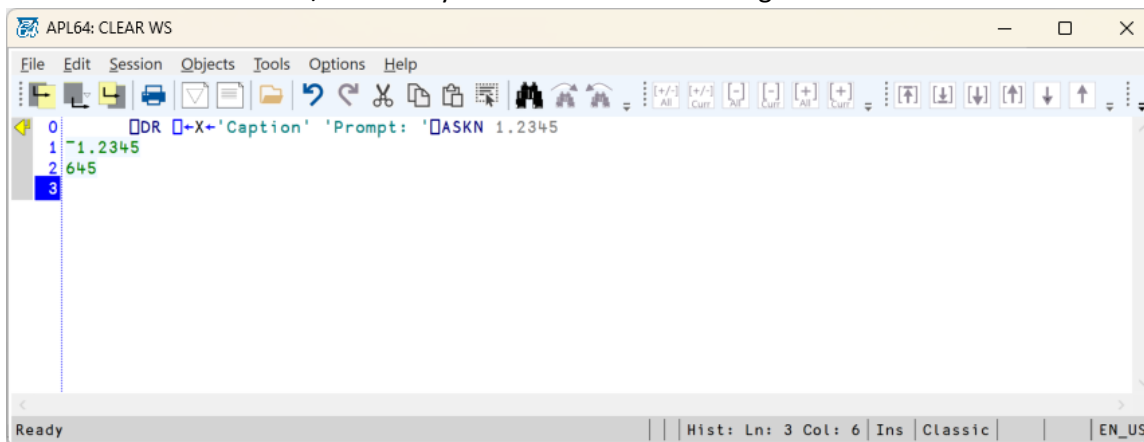
DR ←X←'Caption' 'Prompt: ' ASKN 1.2345



User editing of the default value:



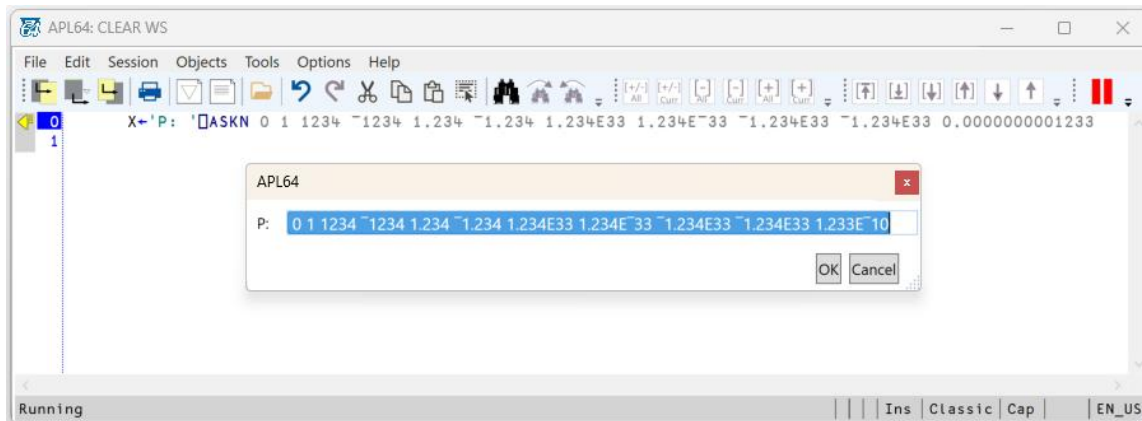
After user clicks the Enter/Return key or OK button or the dialog close button:



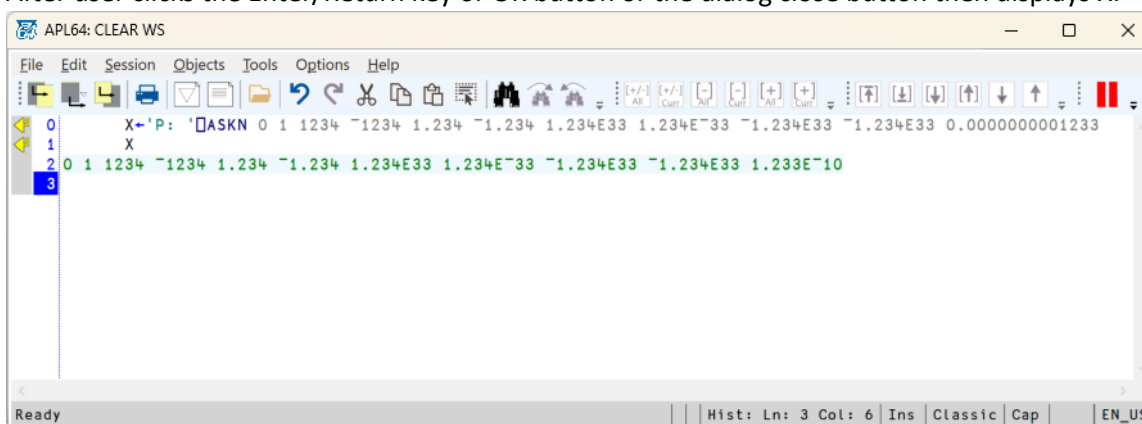
☐ ASKN with numeric vector of mixed data types

In this example the user clicked the Enter key without editing the default values

X←'P:' ☐ ASKN 0 1 1234 ~1234 1.234 ~1.234 1.234E33 1.234E~33 ~1.234E33 ~1.234E33 0.000000001233



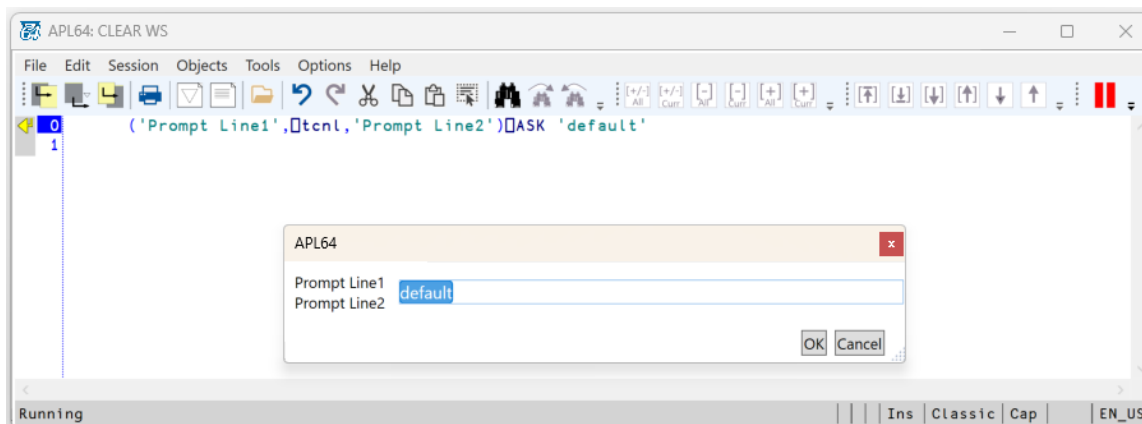
After user clicks the Enter/Return key or OK button or the dialog close button then displays X:



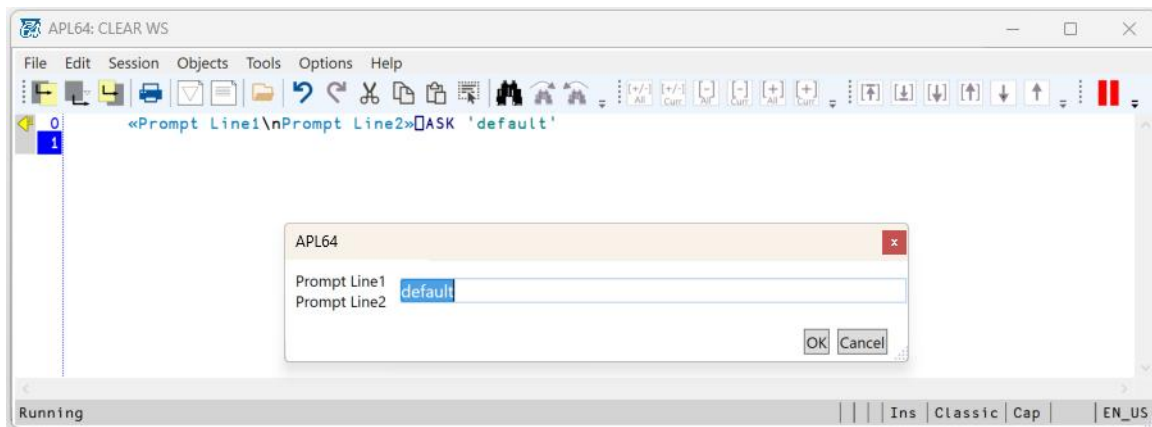
Multi-line Prompt for ☐ASK, ☐ASKN or ☐ASKL

A 'new line' delimited text value may be used as the prompt argument to ☐ASK, ☐ASKN or ☐ASKL:

```
('Prompt Line1',☐tcln,'Prompt Line2')☐ASK 'default'
```



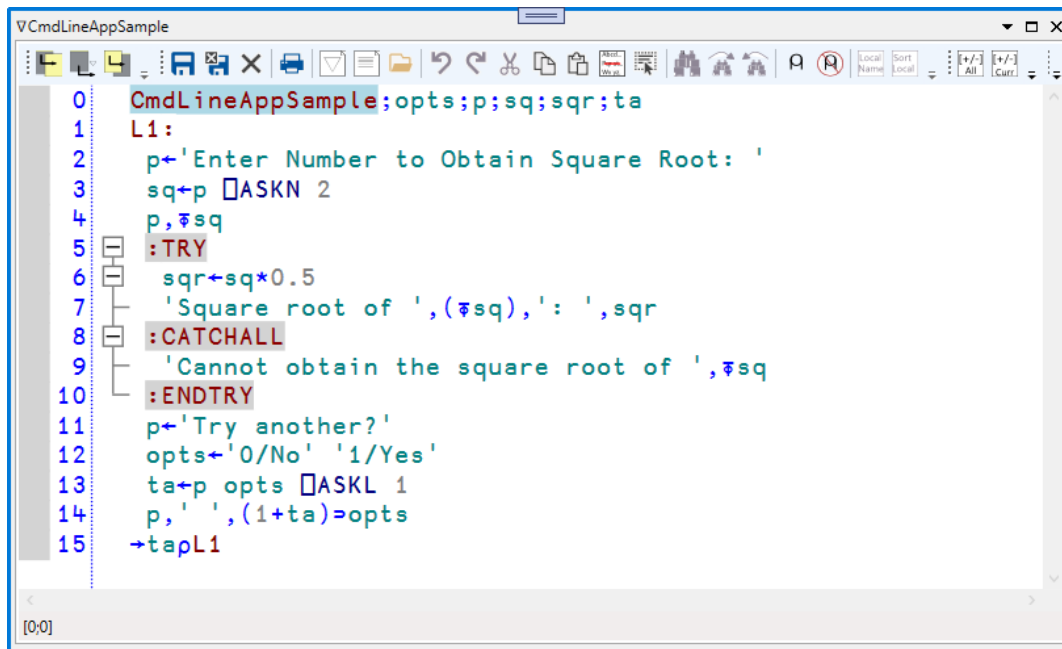
```
«Prompt Line1\nPrompt Line2»☐ASK 'default'
```



Sample APL64 Command Line Application

The APL programmer-defined CmdLineAppSample function illustrates how the ☐ASKx system functions can be used instead of Quad and Quote-Quad in an desktop application running in the APL64 developer version.

```
CmdLineAppSample;opts;p;sq;sqr;ta
L1:
p←'Enter Number to Obtain Square Root: '
sq←p ☐ASKN 2
p, ⍎ sq
:TRY
sqr←sq*0.5
'Square root of ',(⍎ sq),': ',sqr
:CATCHALL
'Cannot obtain the square root of ',⍎ sq
:ENDTRY
p←'Try another?'
opts←'0/No' '1/Yes'
ta←p opts ☐ASKL 1
p,',(1+ta)▷opts
→tapL1
```

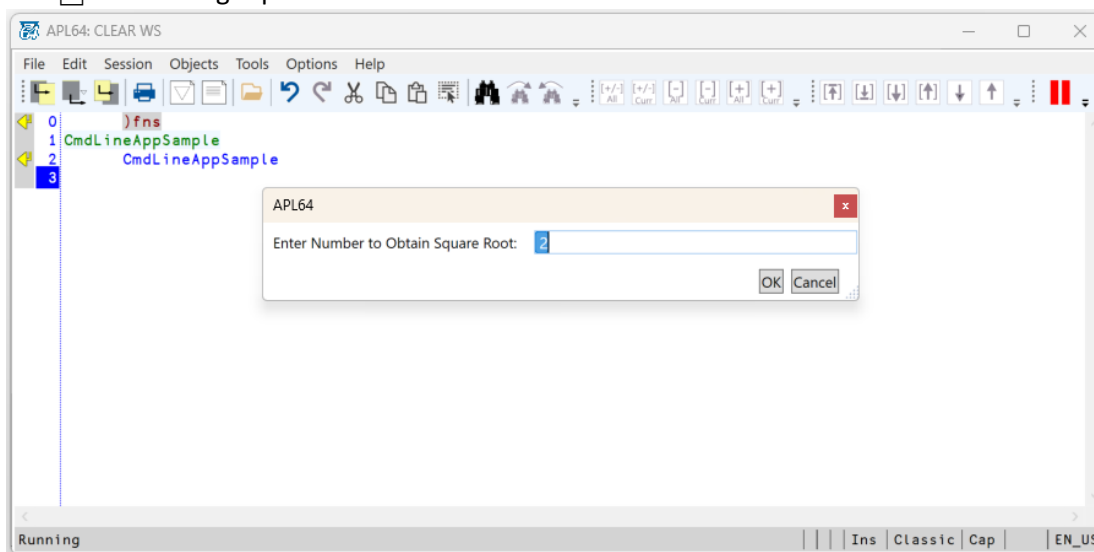


```
0 CmdLineAppSample;opts;p;sq;sqr;ta
1 L1:
2 p←'Enter Number to Obtain Square Root: '
3 sq←p ⎕ASKN 2
4 p,⌘sq
5 :TRY
6   sqr←sq*0.5
7   'Square root of ',(⌘sq),': ',sqr
8 :CATCHALL
9   'Cannot obtain the square root of ',⌘sq
10 :ENDTRY
11 p←'Try another?'
12 opts←'0/No' '1/Yes'
13 ta←p opts ⎕ASKL 1
14 p,' ',(1+ta)⌘opts
15 →tapL1
```

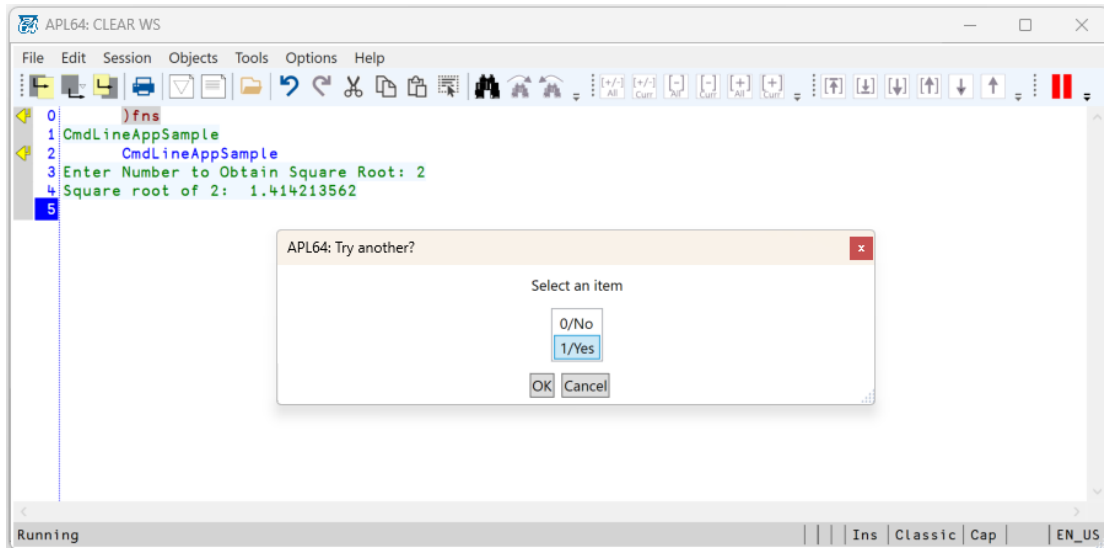
Sample usage of the CmdLineAppSample function

- Calculated the square root of 2

The ☐ASKN dialog is presented:

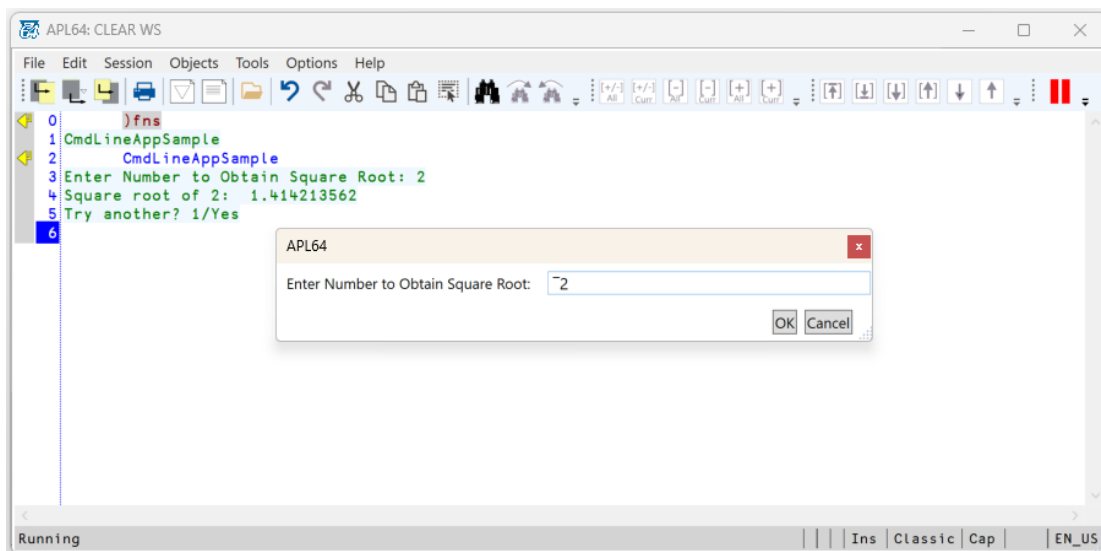


After user clicks the Enter/Return key or OK button:

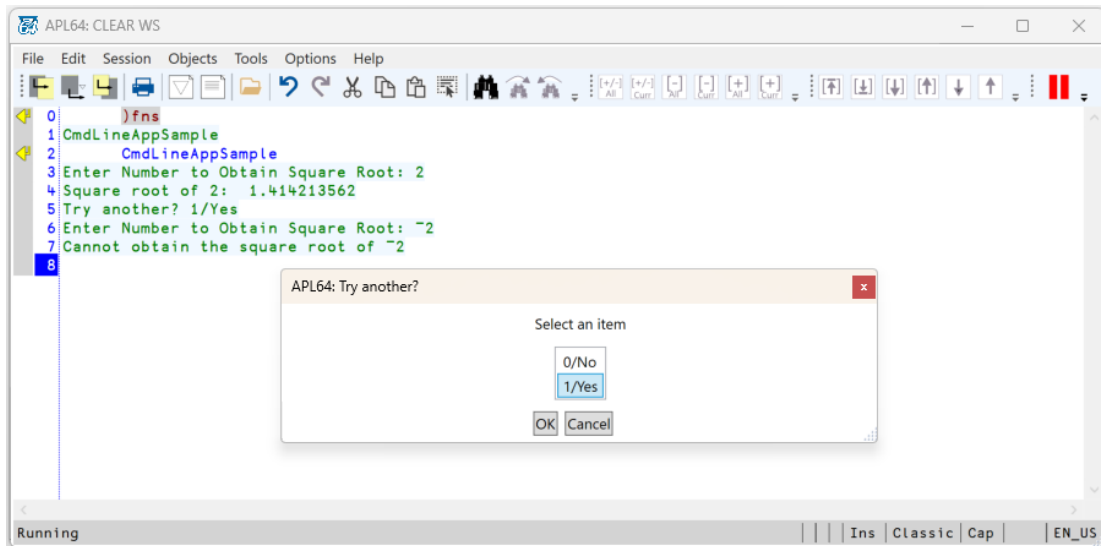


- Attempt to calculate the square root of -2

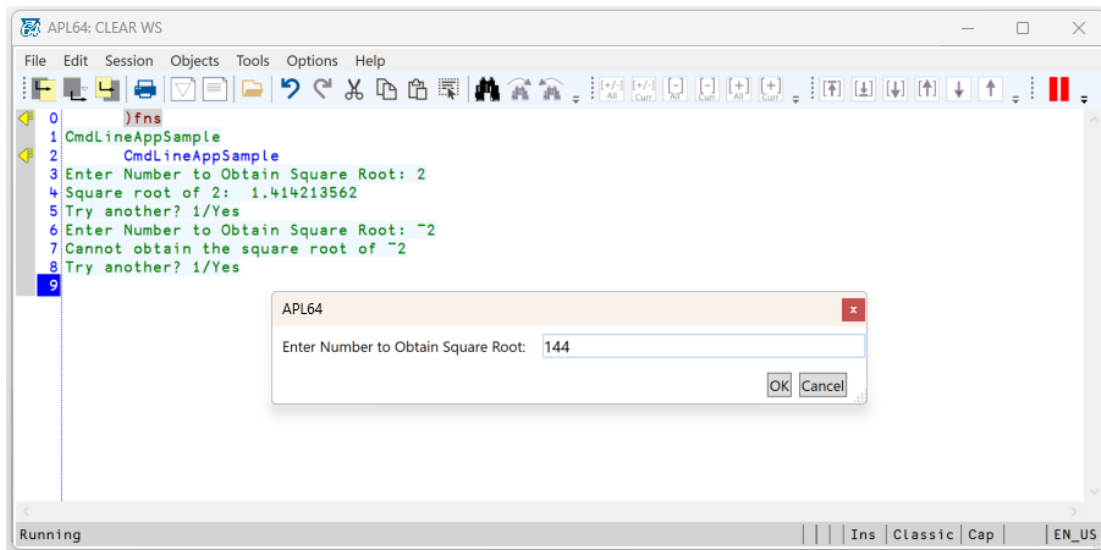
After user clicks the Enter/Return key or OK button then enters $\sqrt{-2}$:



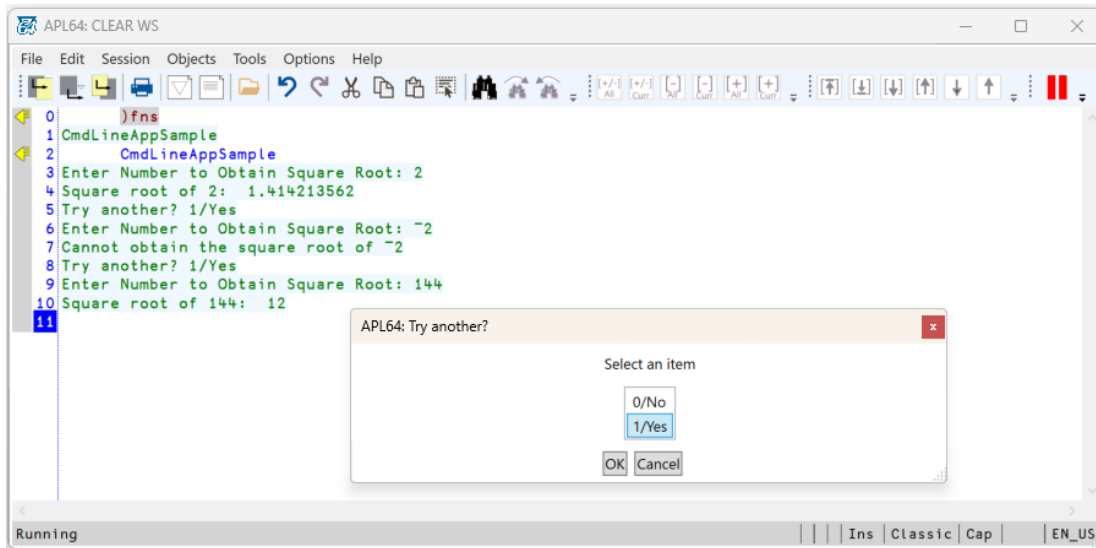
After user clicks the Enter/Return key or OK button:



- After user clicks the Enter/Return key or OK button then enters 144:



After user clicks the Enter/Return key or OK button:



After user selects No then clicks the Enter/Return key or OK button:

