

APL64: System Functions for Windows Dialogs

Contents

Summary	1
 MessageBox	1
 OpenFileDialog	3
 OpenFolderDialog	8
 SaveFileDialog	10
Are Additional Windows Dialogs Needed?.....	12

Summary

The Tools.ws64 workspace contains several Win32-based functions to present dialogs for filename to open, filename to save dialog functions, such as WGetOpenFileName. These functions may continue to be used when an APL64-based application is targeting the Windows desktop within the APL64 Developer or the Windows Runtime Executable (WRE) environment. These Win32-based functions have limitations, including available filename characters. Microsoft replaced Win32 dialogs with System.Windows.Forms dialogs in 2003 and replaced those dialogs with Windows Presentation Foundation (WPF) dialogs in 2006.

This document describes the APL64 system functions, based on WPF, that can present some of these dialogs without the Win32 limitations.

These system functions may be used when an APL64-based application system is targeting the Windows desktop within the APL64 Developer or the Windows Runtime Executable (WRE) environment. These system functions are not available in an APL64 cross-platform component (CPC).

When used, these system functions present a modal dialog with respect to the APL64 application using them.

MessageBox

Syntax: result(char[])← MessageBox text caption type image defaultResult

☐ **MessageBox text** is a character singleton, character vector or scalar string which can include newline characters.

☐ **MessageBox caption** is a character singleton, character vector or scalar string which is limited to one line.

☐ MessageBox type is not case sensitive.

☐ MessageBox type
OK
OKCancel
AbortRetryIgnore
YesNoCancel
YesNo
RetryCancel
CancelTryContinue

☐ MessageBox image is not case sensitive.

☐ MessageBox image
None
Error
Question
Exclamation
Information

☐ MessageBox defaultResult is not case sensitive.

☐ MessageBox defaultResponse
OK
Cancel
Abort
Retry
Ignore
Yes
No
TryAgain
Continue

Example:

☐ dr ☐ ← ☐ MessageBox 'Do you agree?' 'Caption' 'YesNo' 'Question' 'Yes'



OpenFileDialog

Syntax using default options: `filename(char[]) ←  OpenFileDialog "`

Syntax specifying option: `filename(char[]) ←  OpenFileDialog optMat`

Syntax specifying options with MultiSelect: `filenames(char[:]) ←  OpenFileDialog optMat`

The result will contain Unicode characters if the user-selected filename(s) include Unicode characters. The recent folder will be the target folder unless otherwise specified in optMat.

optMat is a two-column matrix with one row for each selected [option](#), as described in this link. The option keys are not case sensitive.

Option Key	Option Value
AddExtension	bool
AddToRecent	bool
CheckFileExists	bool
CheckPathExists	bool
DefaultDirectory	text
DefaultExt	text
Filename	text
Filter	text
FilterIndex	integer
InitialDirectory	text
MultiSelect	bool
ReadOnlyChecked	bool
RestoreDirectory	bool
RootDirectory	text
ShowHiddenItems	bool
ShowReadOnly	bool

Title	text
ValidateNames	bool

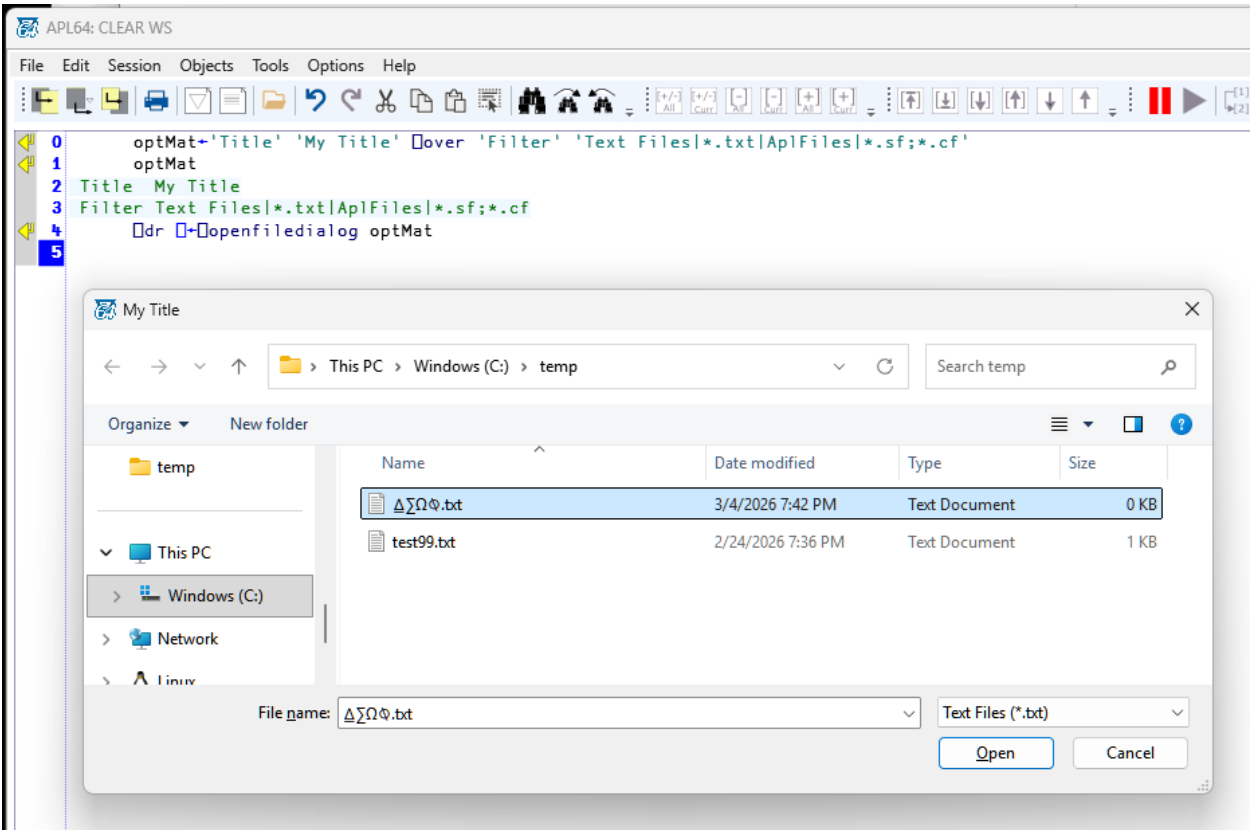
Example #1:

Multiselect not included in options

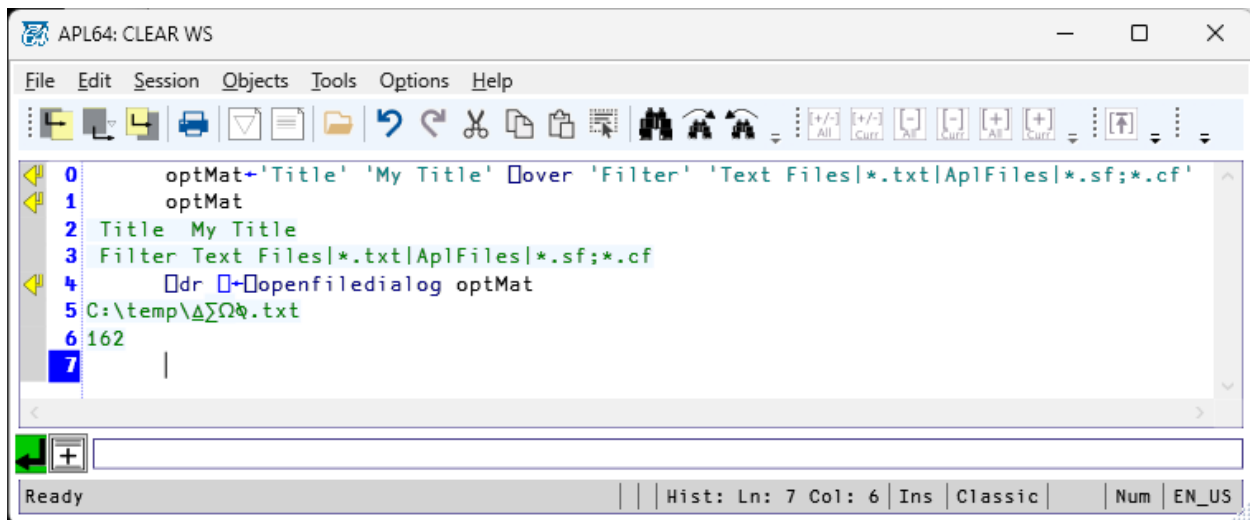
User selected a filename containing Unicode characters.

```
optMat←'Title' 'My Title' ⎕over 'Filter' 'Text Files|*.txt|AplFiles|*.sf;*.cf'  
⎕dr ⎕←⎕openfiledialog optMat
```

Before clicking the ‘Open’ button:



After clicking the ‘Open’ button: Result is a Unicode character vector.



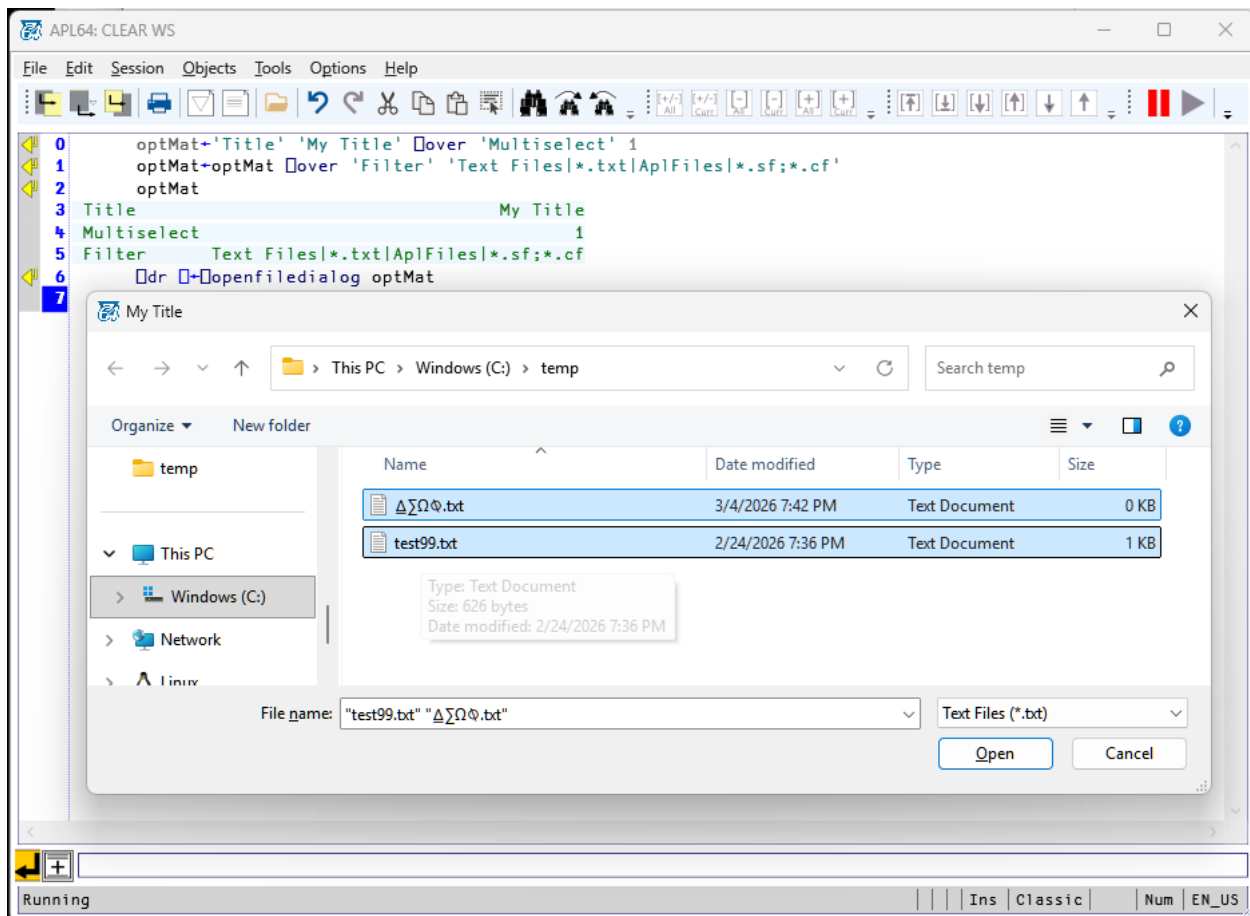
Example #2:

Multiselect option included in options.

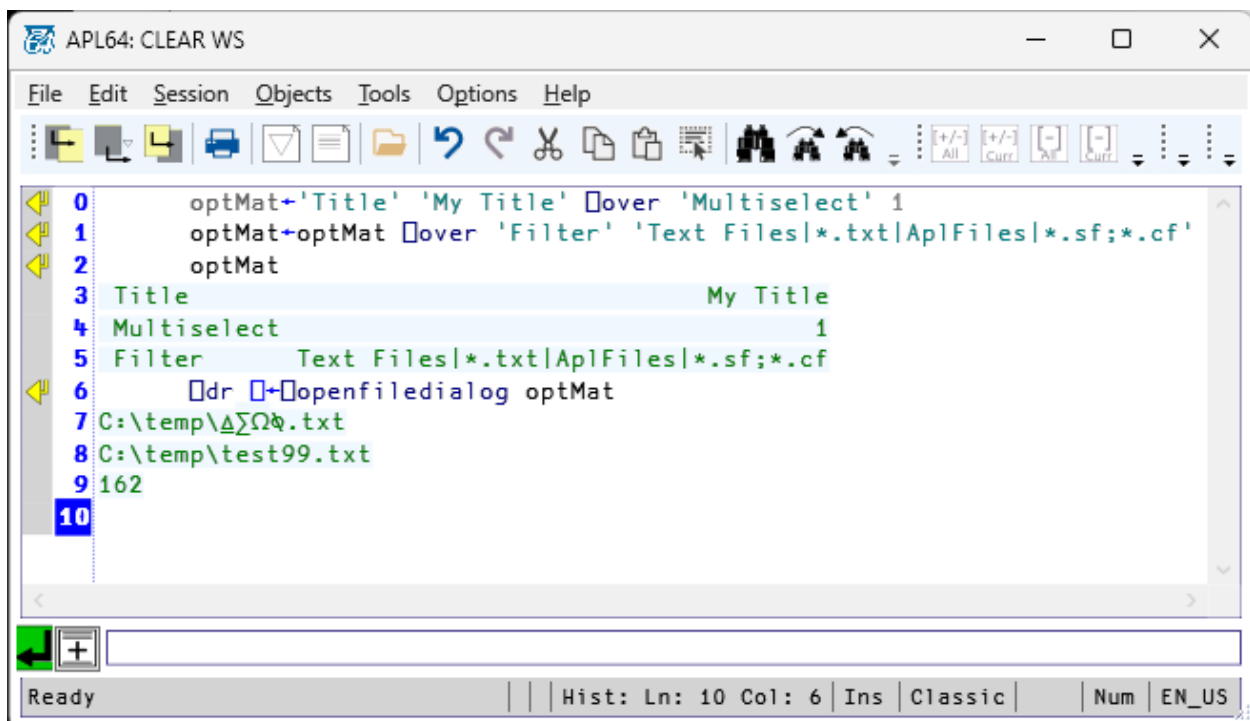
User selected multiple filenames which contained Unicode characters.

```
⎕dr ⎕←⎕openfiledialog('Multiselect' 1 ⎕over 'Title' 'MyTitle')
```

Before clicking the 'Open' button:



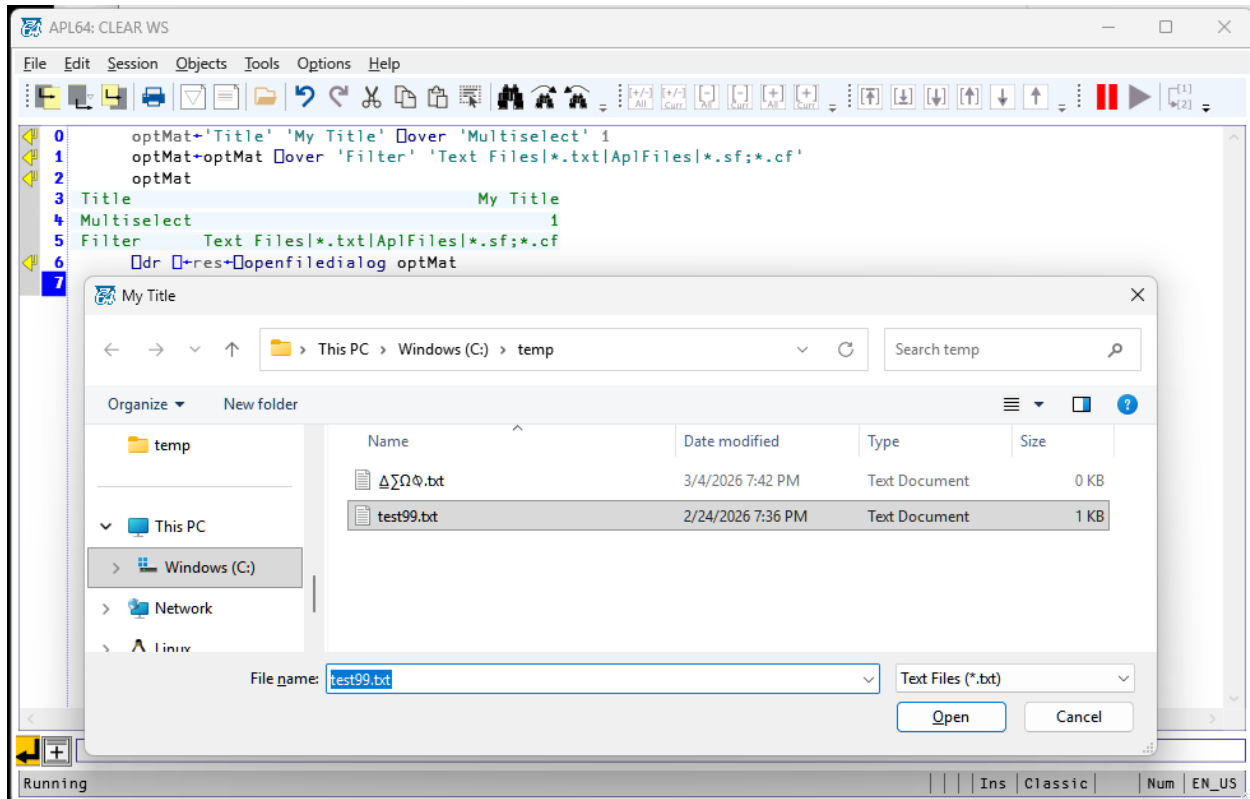
After clicking the 'Open' button:



Example #3: Filename without Unicode characters selected:

```
optMat←'Title' 'My Title' ⎕over 'Multiselect' 1
optMat←optMat ⎕over 'Filter' 'Text Files|*.txt|AplFiles|*.sf;*.cf'
optMat
⎕dr ⎕←res←⎕openfiledialog optMat
```

Before clicking the 'Open' button:



After clicking the 'Open' button:

```

0  optMat←'Title' 'My Title' ⍵over 'Multiselect' 1
1  optMat←optMat ⍵over 'Filter' 'Text Files|*.txt|AplFiles|*.sf;*.cf'
2  optMat
3  Title                               My Title
4  Multiselect                         1
5  Filter      Text Files|*.txt|AplFiles|*.sf;*.cf
6  ⍵dr ⍵res←⍵openfiledialog optMat
7  C:\temp\test99.txt
8  82
9  pres
10 1 18
11 |

```

OpenFolderDialog

Syntax using default options: filename(char[]) ←  OpenFolderDialog "

Syntax specifying options (no MultiSelect): filename(char[]) ←  OpenFolderDialog optMat

Syntax specifying options (MultiSelect): filenames(char[:]) ←  OpenFolderDialog optMat

The result will contain Unicode characters if the user-selected folder name(s) include Unicode characters. The recent folder will be the target folder unless otherwise specified in optMat.

optMat is a two-column matrix with one row for each selected [option](#), as described in this link. The option keys are not case sensitive.

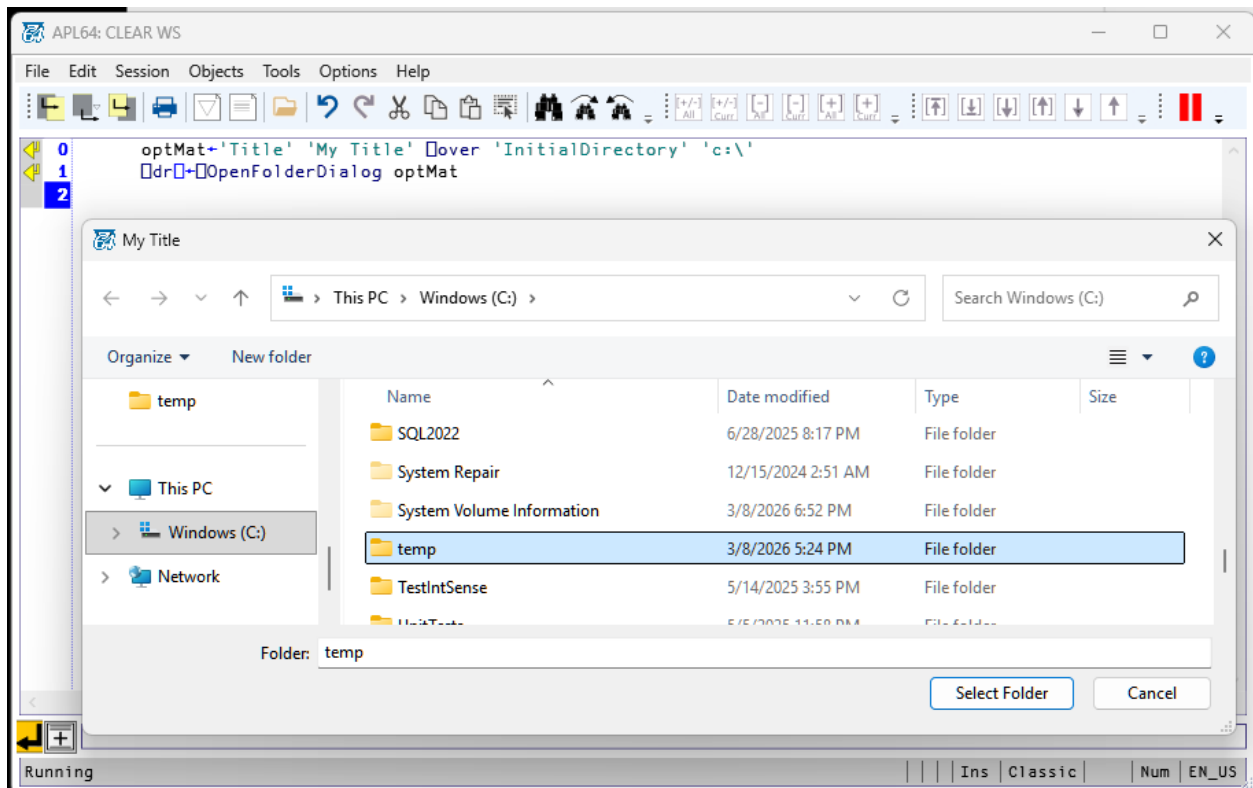
Option Key	Option Value
AddToRecent	bool
DefaultDirectory	text
FolderName	text
InitialDirectory	text
MultiSelect	bool
RootDirectory	text
ShowHiddenItems	bool
Title	text

ValidateNames	bool
---------------	------

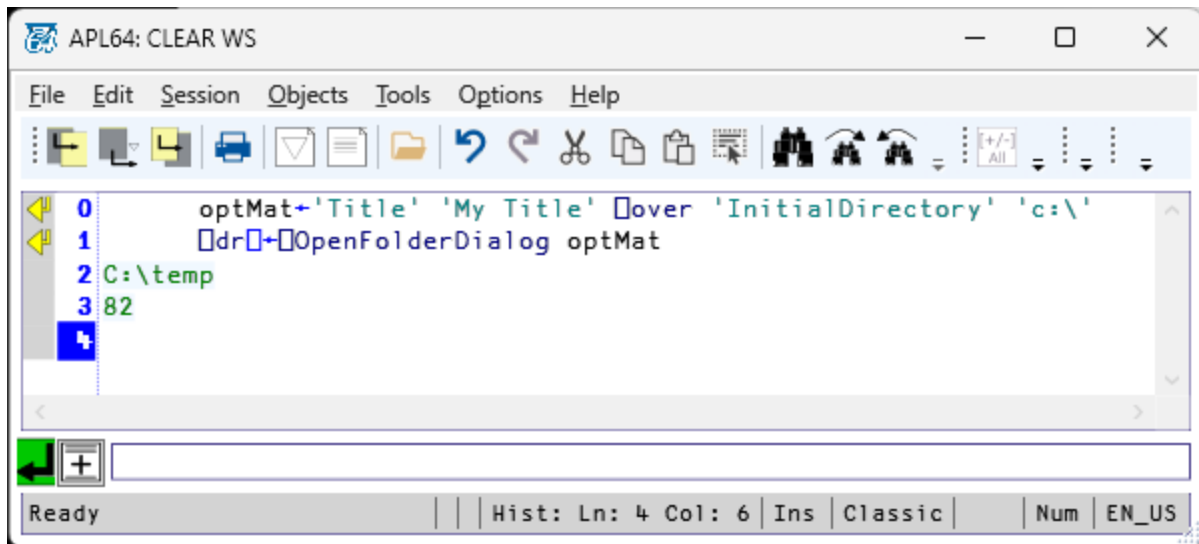
Example:

```
optMat←'Title' 'My Title' □ over 'InitialDirectory' 'c:\'
□dr□←□OpenFolderDialog optMat
```

Before clicking the ‘Select Folder’ button:



After clicking the ‘Select Folder’ button:



SaveFileDialog

Syntax using default options: filename(char[]) ←  SaveFileDialog "

Syntax specifying option: filename(char[]) ←  SaveFileDialog optMat

optMat is a two-column matrix with one row for each selected [option](#), as described in this link. The option keys are not case sensitive.

The recent folder will be the target folder unless otherwise specified in optMat.

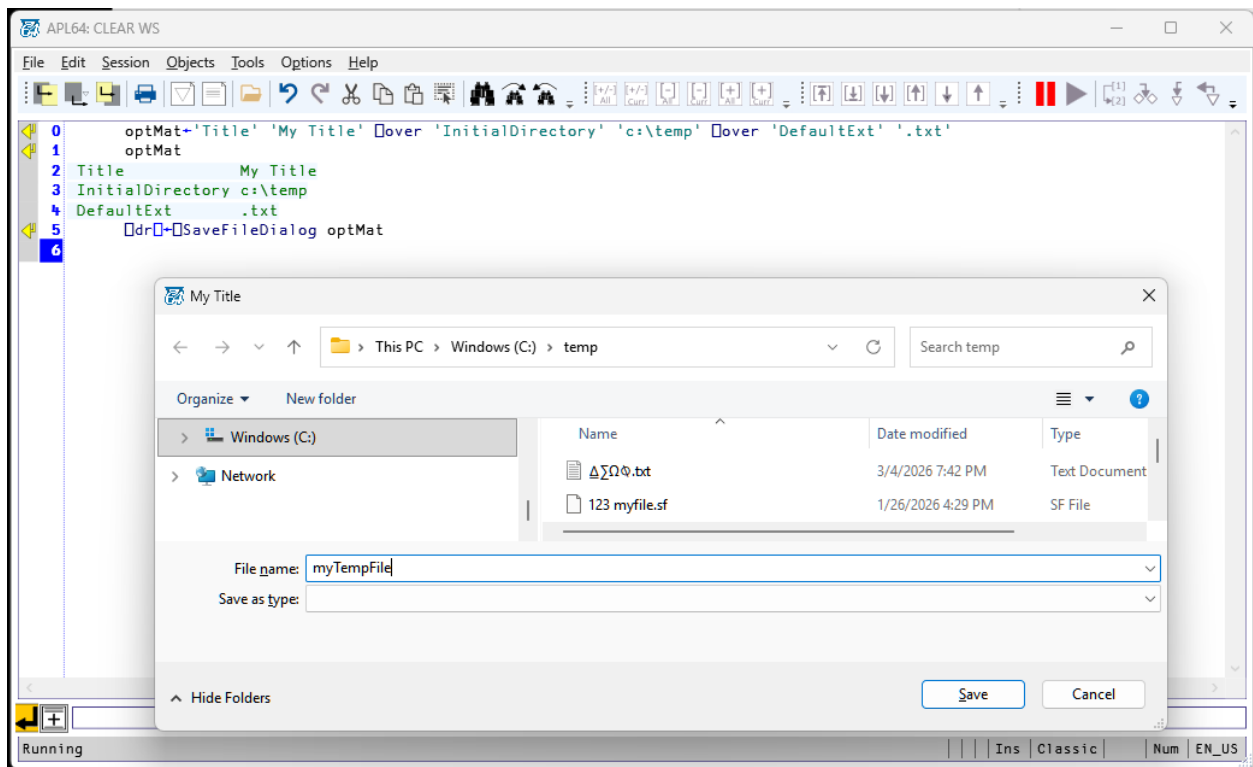
Option Key	Option Value
AddExtension	bool
AddToRecent	bool
CheckFileExists	bool
CheckPathExists	bool
CreatePrompt	bool
DefaultDirectory	text
DefaultExt	text
Filename	text
Filter	text
FilterIndex	integer
InitialDirectory	text
OverwritePrompt	bool
RestoreDirectory	bool
RootDirectory	text
ShowHiddenItems	bool

Title	text
ValidateNames	bool

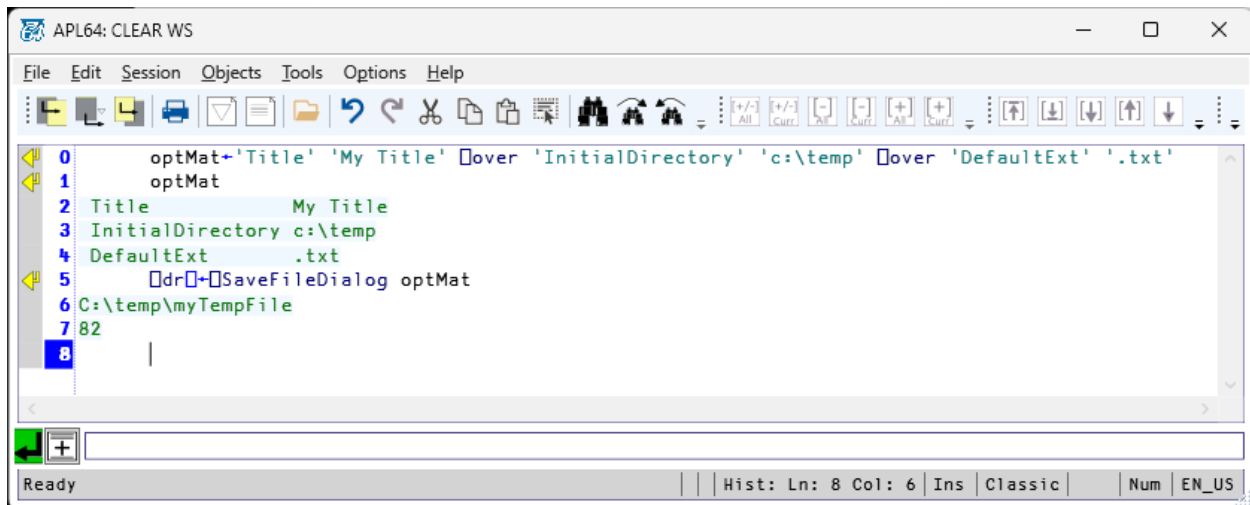
Example:

```
optMat←'Title' 'My Title' ⎕over 'InitialDirectory' 'c:\temp' ⎕over 'DefaultExt' '.txt'
⎕dr⎕←⎕SaveFileDialog optMat
```

Before clicking the ‘Save’ button:



After clicking the ‘Save’ button:



Are Additional Windows Dialogs Needed?

The APL64 development team can implement additional Windows dialogs. Please contact support@apl2000.com for more information.