



# **FORTRAN/TK,**

an easy-to-learn widget toolkit for the Open Watcom  
FORTRAN 77 compiler

*<http://qdllos.sourceforge.net/fortrantk>*

# Overview

1. Features
2. Possible uses
3. How to use it
  - 3.1. Arguments, Return values
  - 3.2. Typical structure of a FORTRAN/TK program
  - 3.3. The API (Short overview)
  - 3.4. Compiling FORTRAN/TK
4. Current status and remaining problems
5. The projects future

# 1. Features

- Widget toolkit: combining the ease of the Tk-toolkit from Tcl/Tk with the power of FORTRAN
- Additional FORTRAN/TK extensions combined with the FSUBLIB make FORTRAN a fairly complete application development tool!
- Cross platform: OS/2 32-bit and Windows 32-bit
  - OS/2 developers can increase their audience
  - First real solution for cross platform GUI development in OW FORTRAN 77
- Small ongoing Open Source project
- uses Rexx/Tk by Roger O'Connor and Mark Hessling
- Rexx interpreter unnecessary

## 2. Possible uses

- Easy to use, short learning curve
  - Write programs from scratch
  - Extending FORTRANs functional range
  - Addicting more people to FORTRAN
  - E.g. base for my projects (QDLOS Editor)
- 
- Backwards compatibility of FORTRAN 77
  - Port old programs straight / create user interface easy and clean
  - No »pipes« or difficult techniques necessary



Makes OS/2 more attractive for programmers

### 3. How to use it

#### 3.1. Arguments, Return values

- Uses null-terminated strings: `'...'`
- Actual length of *character*-variables as arguments is variable
- *ntlen*-function: get length of string without trailing *null*
- Variable number of arguments in Tk-Functions
- Last 2 chars of the last argument must be *null*-chars - use the predefined *dn*-variable
- All arguments must be *character* - conversion with *int2str* / *real2str*-functions

#### **Example:**

```
character*256 mainwin/ '.'c/  
call TkFocus('-force'c, mainwin//dn)
```

### 3. How to use it

#### 3.1. Arguments, Return values

- String function return value:  
256 byte long *character* (*null*-terminated)
- If you just want the value, use a substring: `var(:ntlen(var))`
- Value can be an argument of another function call

**Example:**

```
call TkPack(TkButton( ' .b1 ' //dn ) //dn )
```

### 3. How to use it

#### 3.1. Arguments, Return values

- Return value bigger than 256 bytes
- Extended functions (e.g. *TkGetEx*, *TkCgetEx*) - return *RXSTRING* structure
- Manual allocation possible

#### **Example:**

```
        record /RXSTRING/ rval
        character*(*) buffer
*
*   ...
        rval = TkGetEx(' .e1' //dn)
        allocate(buffer*rval.strlength,
&                location=rval.strptr)
* use buffer
        deallocate(buffer)
```

## 3.2. Typical structure of a FORTRAN/TK program

```
* Include file for auxillary pragmas:
c$include fortrantk.fap

      program STRUCTURE
* Include file for function definitions/structures:
      include 'fortrantk.fi'

* Initialize FORTRAN/TK or stop
      if(TkInit('.'c))
& stop 'FORTRAN/TK couldn't get initialized!'

      call CreateWindow()      ! call function to create widgets

      loop                    ! Main event loop
      cmpval = TkWait()        ! get event
      if (cmp('Quit'c)) then ! Standard event: Quit
        exit
        ! possibly other events (use else if / else)
        ! ...
      end if
      end loop

      call TkDestroy('.'//dn) ! Because of error under Win2k
      if(TkUnload()) write(*,*) 'FORTRAN/TK couldn't get unloaded!'
      end

* ... functions (e.g. CreateWindow) ...
```



### 3.3. The API (Short overview)

- ♦ FORTRAN/TK basic functions:  
TkInit, TkUnload, CDRoot
- ♦ Tk-Functions - all Rexx/Tk functions and extensions:  
TkWait, TkButton, TkBind, TkCanvas, TkEntry, TkMenu,  
TkMessageBox, TkPack, TkGrid, TkGet, ...
- ♦ 13 extended functions:  
TkCgetEx, TkGetEx, TkVarEx, ...  
Adding more functions is relatively easy!
- ♦ FORTRAN/TK Extensions - over 20 useful functions:  
GetCurDir, CreateFile, GetFileAttrib, SetClipboard,  
GetFirstFile, GetNextFile, ...
- It makes FORTRAN an all-purpose language!

## 3.4. Compiling FORTRAN/TK

- ♦ Initiate the Open Watcom environmental variables:  
*setvars.cmd* - or - use your *config.sys*
- ♦ Add the FORTRAN/TK path to the FINCLUDE variable:  
e.g.: *set FINCLUDE=%FINCLUDE%;[path]*
- ♦ Compile FORTRAN/TK:  
*mfortrantk\_os2.cmd*  
→ *fortrantk.lib* will be created
- ♦ Compile your program:  
*wf1386 [app.for] -"f [fortrantk.lib]" -FE=[app.exe] -BM -DEF=\_\_OS2\_\_ -L=OS2V2*  
→ Use *-L=OS2V2\_PM* for an executable without console

## 4. Current status and remaining problems

- ♦ Quite usable yet
- All Rexx/Tk functions and extensions implemented
- FORTRAN/TK extensions enable complete application development
- ♦ Lack of documented sourcecode and documentation
- ♦ Still a lot of bugs - some will be never fixed!?
- ♦ More than one time the same function as argument: must append something (e.g. `//A`)

### **Example:**

```
call TkPack(TkButton( '.b1' //dn ) ,  
&          TkButton( '.b2' //dn ) // 'A' ,  
&          '-expand' c , 'yes' //dn )
```

## 4. Current status and remaining problems

Windows:

- ✦ REXX/Tk or Tcl/Tk:

Calling a *TkMessageBox* before *TkWait* »freezes« the windows

- ✦ Tcl/Tk 8.0.5: needs the *lib*-directory to work

OS/2:

- ✦ Tcl/Tk 8.0.5 and later (under some OS/2 systems):

Program crashes when clicking on a control icon, e.g. to close it

- ✦ Tcl/Tk really buggy:

e.g.: popup-menus don't work properly; not all *events/keybindings* are interpreted; sometimes graphics errors

- The Windows version of FORTRAN/TK still works better.

- REXX/Tk is a dead project: e.g. lacks the *place*-command of Tk

- Tcl/Tk for OS/2 is nearly a dead project, rare updates (v.8.3.5 is latest)

## 5. The projects future

- ♦ Some problems of FORTRAN/TK could/will be solved:  
complete the documentation; replace my way of reading arguments, ...
- ♦ Maybe I'll try to improve Rexx/Tk
- ♦ Some problems will probably never be solved!
- Anyway, FORTRAN/TK could be the base for a lot of cool projects.
- Help to improve FORTRAN/TK - it's Open Source!

`http://qdllos.sourceforge.net/fortrantk  
None-Brain@web.de`