DoDos Version 3.4

Overview

While running DOS applications from Windows is simple, providing variable run time arguments for DOS applications is not. DoDos allows you to create arguments for DOS applications at run time. DoDos provides powerful file-argument and directory-argument browse-and-select capability comparable to that found in Windows applications. While DoDos is relatively simple to use, it is somewhat hard to explain. Therefore it is suggested that, after you have read through this document, you follow the directions in the DoDos Exercise section to become more familiar with DoDos.

Installing DoDos

To install DoDos, copy both DoDos.EXE and VBRUN100.DLL to your Windows directory, or to a directory in your PATH.

Conventions

Syntax

The following conventions are used in this documentation.

Required	Bold characters indicate required items which must be entered exactly as shown. Required items are not case sensative.
[option]	Items enclosed in [] are optional
{one |two}	At least one of the the items enclosed in {} and separated by | are required.

Punctuation and Special Characters

The DoDos program name must always be separated from the first DoDos command-line element by a space. DosDos command-line elements may be separated with either a blank or comma. Command-line switches need no intervening punctuation. The percent, comma, and space characters are key characters for DoDos, and may not appear in the command line outside of the syntactical context described for command line elements.

DoDos Command Line

Syntax **DODOS** [switches,]{Lprog,A|Vprog}[,A ... ,A]

Where

switches	Option switches
Lprog	Executable file
Vprog	Executable-File parameter
A	Argument

The DoDos command line may be placed in either a program item (Command Line field) or PIF (Program file-name field) created for the target program. For adhoc use, DoDos may also be used from the File Manager File-menu Run command. When executed, DoDos will allow the selection of arguments for the target program based on the Argument-Parameter types chosen, and then run the program with those arguments. DoDos will accept up to 6 Argument Parameters when a literal program name is used, and up to 5 Argument Parameters when an Executable-File parameter is used.

Command-Line Switches

Command-line switches are optional. Switches must apear immediately after DoDos on the command line, and may be in any order.

Current Directory

Syntax **/C=**d:\path

where

d:\pathComplete path for current directory.

DoDos sets the current directory to the path specified.

Terminate on Run

Syntax **/NT**

DoDos will remain active after the target program is started. The default is for DoDos to terminate after starting the target program.

Argument Parameters

DoDos recognizes six types of arguments, **directory path, directory**, **file path**, **file name**, **text**, and **fixed**. The five variable argument-parameter types are summarized below.

Argument Parameter	Type	Allows Selection From	Resolves To
%P	Path	Drives,directories	directory path
%D	Directory	Drives,directories	last directory

%F	File	Drives,directories,files	file path
%N	File name	Drives,directories,files	file name
%T	Text	Keyboard input	input

Directory Path

Syntax [prefix]**%P**[=**{**d:\path\|suffix**}**]

where

prefix Any string of legal characters

d:\path\ Any valid path for the drive indicated. The path must ALWAYS end with a "\".

suffix Any string of legal characters

Examples	/DIR=%P=c:\windows\	-->	/DIR=c:\windows
	%P=c:\dos*.com	-->	c:\dos*.com

The %P argument parameter allows the argument to be resolved to any directory path, beginning with the path specified in the parameter. Any suffix provided is appended to the end of the final directory path selected. Any prefix provided is appended to the front of the final directory path. For simplicity, the examples above are shown to resolve to the same directory path as originally specified in the argument parameter. The original specification is, however, only a starting point, and any directory may be selected as the final resolution of the parameter by browsing and clicking.

Directory

Syntax [prefix]**%D**[=**{**d:\path\|suffix**}**]

where

prefix Any string of legal characters

d:\path\ Any valid path for the drive indicated. The path must ALWAYS end with a "\".

suffix Any string of legal characters

Examples	/DIR=%D=c:\windows\	-->	/DIR=windows
	%D=c:\dos*.com	-->	dos*.com

The %D parameter functions almost identically to the %P parameter. When %D is used, the final resolution of the parameter is the last directory in the selected path, rather than the entire path as provided by %P.

File Path

Syntax [prefix]**%F**[={d:\path\|filename}]

where

prefix Any string of legal characters

d:\path\ Any valid path for the drive indicated. The path must ALWAYS end with a "\".

filename Any valid file name or template.

Examples	/FN=%F=c:\windows*.ini	-->	/FN=c:\windows\win.ini
	%F=c:\dos*.com	-->	c:\dos\command.com

The %F parameter allows the argument to be resolved to any file name path, beginning with the path specified in the parameter. Since this is a file name-type argument, any file name provided is used as the template when selecting the file. If no template is provided, "*.*" is used. The prefix, if supplied, is appended to the front of the final file name path. For simplicity, the examples above are shown to resolve to a file in the same directory path as originally specified in the argument parameter. The original specification is, however, only a starting point, and any file name (matching the file name template) may be selected as the final resolution of the parameter by browsing and clicking.

File Name

Syntax [prefix]**%N**[={d:\path\|filename}]

where

prefix Any string of legal characters

d:\path\ Any valid path for the drive indicated. The path must ALWAYS end with a "\".

filename Any valid file name or template.

Examples	/FN=%N=c:\windows*.ini	-->	/FN=win.ini
	%N=c:\dos*.com	-->	command.com

The %N parameter functions almost identically to the %F parameter. When %N is used, the final resolution of the parameter is the file name only, without the path as provided by %F.

Text

Syntax **%T**[=string]

where
string Any string of legal characters

The %T parameter provides the ability to enter an argument from the keyboard. Any string provided is displayed, but may be overwritten when editing the argument. If string is null or no string is provided, then the initial argument value will be "???".

Fixed

Syntax string

where
string Any non-null string of legal characters.

Any Argument Parameter encountered without an inbedded "%", is assumed to be a **Fixed** argument. The string is displayed as entered, and can not be edited.

Executable-File Parameter

The target file (the program, PIF, or batch file to be executed) may be literally specified or may be selected at DoDos run time by using the %E Executable-File parameter.

Syntax [d:\path\]name.ext for literal name
%E=[d:\path\]file.ext for run time selection

where

d:\path\	Any valid path for the drive indicated.
name	Any valid file name
file	Any valid file name or file name template.
.ext	.COM, .EXE, .PIF, .BAT

Examples %E=c:\windows*.EXE select from .EXE files
c:\unzip.com run C:\UNZIP.COM

The %E parameter allows the argument to be resolved to any file path, beginning with the path specified in the parameter, that matches the template specification. When using %E, any path specified serves as a starting point, allowing any file matching the template to be selected by browsing and clicking.

Argument Selection

Select an argument by clicking the button to its right. If the argument parameter is **directory path, directory**, **file path,** or **file name**, the file-system selection window will appear. The intitial settings for the file-system selection window are deteremined by the path and template specified in the argument parameter. For files, either double-clicking on the file, or clicking on the file and then clicking on OK will indicate your selection. For directories, double-clicking on the directory and then clicking OK will indicate your selection. Clicking Cancel (Esc) will cancel the selection activity.

If the argument parameter is **text**, an input window will open in which you may edit the argument. Clicking the OK button will accept any changes made. Clicking Cancel (Esc) will cancel any changes made.

Current Directory Selection

The current directory to be in effect at the time the target-program is launched may be set from the DoDos command-line and/or from the DoDos Current Directory display. The initial setting for the current directory is either the path specified on the DoDos Command-Line or, if no path is specified, the then current directory as supplied by Windows. In either case, the current directory may be changed at run time by clicking the button to the right of the Current Directory display.

Menu and Button Options

Terminate On RUN (Options Menu)

> When checked (default), DoDos will terminate after launching the target program. When not checked, DoDos will remain active after launching the target program. To indicate when Terminate on Run is <u>not</u> checked, the Run Button caption appears underlined.

No Commas (Options Menu)

> When checked (default), DoDos separates target-program arguments with blank spaces. When not checked, DoDos separates target-program arguments with commas.

Add Argument (Options Menu)

> Additional arguments may be added at run time by selecting the Add Argument menu option. Select any one of the argument types from the Add window, and click OK (Enter). Click Cancel (Esc) to cancel the Add operation. An argument of the type selected is added following the existing arguments.

Paths, Prefixes, and Suffixes are not supported for arguments added in this way.

Run Button

Clicking the Run Button (Enter) launches the target program with a command line as shown in the Command-Line display at the bottom of the DoDos window. Because the capacity of the Command-Line display is limited, long command lines may not be fully displayed. The Run button is enabled only after all arguments are in the Ready condition, indicated by an "*" at the left of each Argument display. **Fixed**-type arguments are always Ready. All other arguments are set to Ready only after having been selected, and changed from their initial state.

Quit Button

Clicking the Quit Button (Esc) terminates DoDos.

About (Help Menu)

About is self-explanatory. There is no on-line help.

A DoDos Exercise

After reading through the documentation above, you may want to try DoDos on a dummy program. Follow the instructions below to set up a dummy program item to try DoDos.

A Simple Exercise

This exercise will use the Run command from the File-Manager Edit menu to run DoDos with a dummy batch file.

1. Open the File Manager
2. Click on the File menu
3. Click on the Run command
4. Enter the Command Line as follows

    DODOS /NT DUMMY.BAT %N SAMPLE %F %P %T

5. Click OK

The DoDos window will appear. Note the following.

1. The 5 arguments generated by the command line
2. The Fixed argument SAMPLE is Ready ("*")
3. The Run button is dimmed because all arguments are not Ready
4. The Run button is underlined due to the /NT option

5. The current Windows directory appears as the current directory.
6. The ??? argument resulting from the %T parameter

Click the button to the right of each argument to select and change it. As each argument is changed, the Ready indicator ("*") appears to its left. When all arguments are Ready, the Run button becomes available. Clicking the Run button will generate an error message because you do not have a file named DUMMY.BAT. DoDos remains active after the error because of the /NT option. Try adding a fifth argument by clicking on the Options menu, and selecting the Add Argument command.

A More Complex Exercise

This exercise will create a program icon from which DoDos will run a dummy batch file.

1. Open the Group window of your choice.
2. Select NEW from the FILE menu.
3. Select PROGRAM ITEM from the NEW PROGRAM OBJECT menu.
4. Enter "Dummy" for the DESCRIPTION.
5. Enter the Command Line as follows.

   DODOS /NT,DUMMY.BAT,/DIR=%P=*.GIF,%F=*.DLL,/FILE=%N,%T,ABC

6. Select OK.

The program item created will appear in the Group window as a DOS icon with the label Dummy. Double-click the Dummy icon, and note the following.

1. The DoDos Window is labeled with the target-program name.
2. The argument parameters specified on the command line appear in the agument display.
3. Since no current directory was specified, the current Windows directory appears in the Current Directory display window.
4. Since not all arguments have a Ready indicator ("*"), the Run button is dimmed (disabled).
5. The RUN button caption is underlined (due to the /NT option), indicating that DoDos will remain active after starting DUMMY.BAT.

Select and change arguments by clicking the button at the right of each. Practice modifying the arguments and the current directory. Notice that once you have changed an argument, it is marked with the Ready indicator. When you have made all of the arguments Ready, the Run Button will be enabled. Notice that as soon as an argument is Ready, the ??? place-holder in the Command-Line display (bottom of window) is replaced with the argument. Clicking the Run button during this excercise will cause an error message because DUMMY.BAT does not exist (unless of course you happen to have a file named DUMMY.BAT in the current directory). DoDos is still active after the error because of the /NT option on the command line. You might want to create a DUMMY.BAT file (be sure to select the current directory containing this file or add the path in the program item created earlier) and use ECHO statements to see the passed arguments.

After you are finished with the exercise, you may remove the dummy program item as follows.

1. Open the window containing the dummy item.
2. Single-click on the item icon to select it.
3. Select DELETE from the FILE menu.
4. Being sure that the dummy item is the one selected, select YES in the DELETE box.