

Content

[Introduction](#)

Essentials

[Current Destination Window](#)
[Support](#)

Shortcut Components

[Shortcut Taskbar Icon](#)
[Shortcut Main Window](#)
 [Menu Commands](#)
[Shortcut Popup Mode](#)
[Shortcut Command Line Parameter](#)
[Add/Edit Task](#)
[Schedule](#)
[Script Editor](#)
 [Capturing Tools](#)
[Script Commands](#)
 [Basic Script Syntax](#)
 [Comment](#)
 [Label](#)
 [Predefined Variables](#)

Available Commands

[CALL](#)
[DISPLAY](#)
[EXECUTE](#)
[FOCUS](#)
[GOTO](#)
[GRAB \(DOS only\)](#)
[IF TIMEOUT](#)
[PAUSE](#)
[QUESTION](#)
[SEND](#)
[SET DELAY](#)
[SYSCMD](#)
[WAITDOS \(DOS only\)](#)
[WAITWIN \(Windows App.Only\)](#)

Useful Examples

[Shutting Down Windows Automatically](#)

Troubleshooting

[Troubleshooting](#)

Copyright 1998 FABSoft, Inc. All rights reserved

Introduction



Welcome to Shortcut, the most powerful and easy to use automation tool for your DOS (running in Windows) or Windows applications. Shortcut is a script-based automation tool. The script language is easy to understand and implement. We provide you with many tools to speed up the script development.

The followings are some features that make Shortcut a indispensable automation tool:

- Prompt for answer at runtime
- Send the answer or variable out as part of the keystrokes
- Read the text from a specific location from the DOS screen and save as a variable.
- Wait for the text in a specific location from the DOS screen to trigger the next action.
- Wait for the text from within the windows application for a specified duration to trigger the next action.
- Schedule the automation tasks
- Execute other script or other application from within the script.

Current Destination Window

Current destination window is the window that matches *Window Title* specified in *Task Information*. All SEND or SYSCMD will be directed to this window. However, many applications may contain more than one windows. In order to send keystrokes or system commands to any particular window, we provide you with FOCUS command. FOCUS command waits for the window with the specified title to appear. Once found, FOCUS command will set this window as a current destination window. Therefore, any subsequences SEND or SYSCMD will be sent to this window instead of the initial window specified in *Task Information*

See Also

[Task Information](#)

[FOCUS](#)

[SEND](#)

[SYSCMD](#)

Add/Edit Task

The screenshot shows the 'Task Information' dialog box with the following fields and values:

- Name: Armor Application
- Description: Automate Armor Accounting application.
- Command: E:\advant\adv.bat
- Path: e:\advant\
- Window Title: Add Script
- Script File: C:\ShortCut\Scripts\AmrScr.for
- Application Type: DOS Application
- Auto Close:
- Enable Scheduling:

[Add/Edit Script](#)

This is where you enter all the necessary information for your automation task.

Name

Enter the name of the task. The name must be unique from one another. However, if you enter the name that is already exists, Shortcut appends extra number to the end of the name in order to prevent duplicate task name.

Description

Enter a detailed description of this task. This will appear as a floating hint when you highlight icon of the selected task from the main window.

Command (optional)

This is where you enter a command to run your application prior to executing the script. You can leave this blank if the application you want to automate is already running.

Path (optional)

Default startup path of your application. Leave this blank if Command is not used.

Window Title

This is the title of your application window you want to automate. You must enter a valid title. Shortcut uses the title to determine which window it needs to send keystrokes to. If you are automating a DOS application which is started by Shortcut (see Command above), you must set your

own title for that DOS session.



Use this title locator to quickly enter in the window title for you. Simple drag the hand pointer to any window you want to retrieve the title from.

Script File Name

Enter name of the script file you want to use for the task. If you have not created a script file for the task yet, click Edit/Create... to create a new script file. (See Script Editor)

Edit/Create...

Edit the script specified in **Script File Name**. If Script File Name has not been specified, **Edit/Create** will create an untitled script and open it for editing.

Auto Close

Check this on if you want Shortcut to automatically close the application when the script finishes. **Important Notes:** DOS applications do not handle the closing sequences like the windows applications do. Therefore, you should send appropriate keystrokes to close out your DOS application in order to prevent the lost of data.

Application Type

Select the appropriate application type you are automating. The valid selections are:

- Windows Application
- DOS Application

Be sure to set this option appropriately because Shortcut needs to know the application type you are automating in order to handle the automation appropriately.

Enable Scheduling

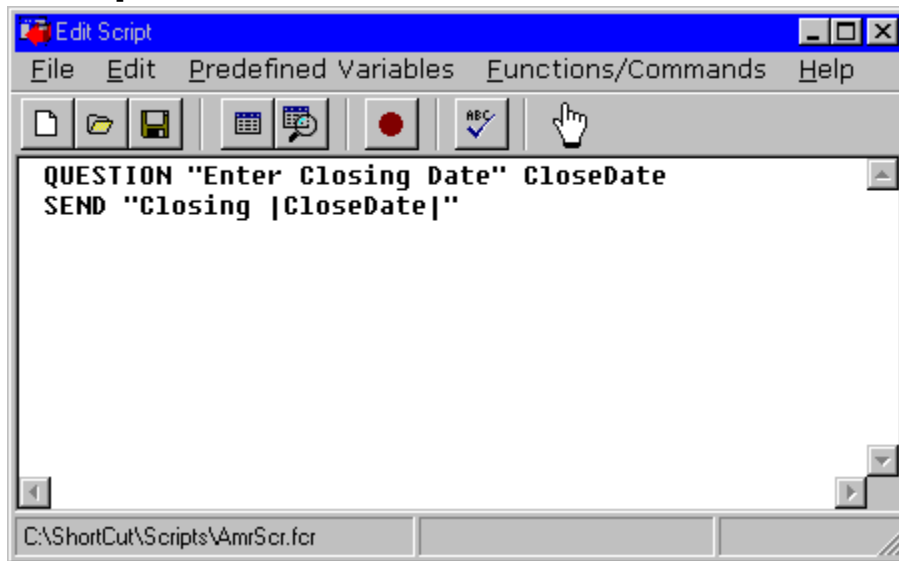
Check this on if you want to enable the scheduling feature. When this is checked on, Schedule... button becomes enable allowing you to configure the schedule date/time. (See Schedule)

See Also

[Script Editor](#)

[Schedule](#)

Script Editor



Script Editor

Script Editor provides you with a convenience way to edit a script for your task automation. Script file is simply a text file that has `.fcr` as an extension. The basic functionalities of this editor is the same as of the notepad. However, we have added features that will help you to create your script quickly and easily. While you are editing a script and you want more detail about a particular command, highlight that command and press F1. This will bring up the help for that command. Before you can start writing your own script, you need to become familiar with the script syntax. (Please refer to Script Commands for more information)

Tips

You can quickly type in {enter} or {tab} (see SEND) keystroke commands by using the popup menu by right clicking the mouse or by pressing the following key combinations:

{Enter}	Ctrl+ENTER
{Tab}	Ctrl+TAB

Script Editor Toolbar



Editor Toolbar



New, Open, Save Script



[Capturing Tools](#)



Record Keystrokes. This will record keystrokes and, when finish, it will automatically insert SEND <keystrokes> command into your script. The recorder only records keystrokes being applied to Windows applications. Keystrokes in DOS application are ignored.

Note: if you are recording a lengthy keystroke task, you might want to break the record session down into different sessions. This way the SEND line will not be too long thus, reducing the chance of keystrokes overrun.



Check Script Syntax. The program will save the script and then perform syntax checking for you. It will notify you of any errors detected.



Use this Window Title Locator when you want to grab the title of the window you want. Drag the pointer to any window you want to retrieve the title from and the editor will insert the window title into the script at the cursor location.

Predefined Variables Menu

Predefined Variables Menu gives you an ability to insert a predefined variables right into the script. Place the cursor at the location you want to insert the predefined variable, then select one of the selections from the menu and the editor will type in the chosen variable for you.

Functions/Commands Menu

Functions/Commands Menu provides you with a quick way to enter in the available script command, thus, reduce the chances of making syntax mistake. Place the cursor at the location you want to insert the command, then select one of the selections.

See Also

[Capturing Tools](#)

[Script Commands](#)

[SEND](#)

[Task Information](#)

Capturing Tools



DOS Capturing Tools

In order to quickly create WAITDOS, WAITWIN or GRAB command, you can use Capturing Tools to assist you with the command creation.



Start application specified in Task information for testing and capturing. (See Task Information)



Capture text from the application. Once the application is started using the option above, you can then grab the text location you want to wait for by using this utility. This utility reacts differently depending on the type of the application you are writing the script for. For DOS application, you will be able to highlight the text you want to wait for and the editor will insert the appropriate WAITWIN or GRAB command for you. For Windows application, you will be able to move the mouse around and capture the text you are interested in. The valid text is indicated by the highlight surrounding it as you move the mouse over that text.

See Also

[Task Information](#)

Basic Script Syntax

Script Syntax

The basic syntax of the script language is:

```
<command> <data1> <data2>           ;<comment>
```

where *<command>* describes the action to perform, *<data1>* and *<data2>* are optional arguments, and *<comment>* is an optional comment. The format of the arguments vary among commands. The various components of each line must be separated by at least one space or a comma. Additional spaces are permitted but ignored.

Variables

Variables are declared by using QUESTION or GRAB command. Use QUESTION command to ask for a value and assign that value to the variable. Use GRAB command to grab a text from a specified point on the DOS screen and assign the grabbed content to a variable. Once the variables are declared, you can then make a reference to a particular variable within the SEND or DISPLAY string by enclosing the variable name with | (see example).

Example:

QUESTION/SEND Example:

```
== Ask for the customer name
QUESTION "Customer Name ?" CustName
== Send the customer name to a window
SEND "|CustName|{ENTER}"
```

GRAB/DISPLAY Example:

```
== Grab a text from specified screen location
GRAB (0, 0, 10) CustName
== Display the name
DISPLAY "|CustName| is found."
```

See Also

[DISPLAY](#)

[GRAB](#)

[QUESTION](#)

[SEND](#)

DISPLAY

```
DISPLAY <"Display Message">
```

This command will Popup a window with the specified message. This is useful for notifying the user for an attention.

Example:

```
DISPLAY "Error Sending Keys"
```

You can make a reference to a variable within the *Display Message* by enclosing it with |.

Example:

```
== Grab a text from specified screen location  
GRAB (0, 0, 10) CustName  
== Display the name  
DISPLAY "|CustName| is found."
```

See Also

[Basic Script Syntax](#)

GRAB (DOS only)

GRAB (<Row>, <Col>, <Width>) <Variable_name>

Grab a text from screen and put the text into a variable. Row and Col specify the location on screen where Width specifies how many character from the specified point. Variable name must not contain any spaces. The valid Row ranges from 0 to 24 and the valid Column ranges from 0 to 79. This command only works with DOS Application.

Example:

```
== Grab a text from the specified point on screen  
GRAB (1, 10, 6) InvoiceNumber
```

Note: use Capturing Tools to help you getting the exact row and col from the DOS screen.

See Also

[Capturing Tools](#)

QUESTION

```
QUESTION <"Question Prompt"> <Variable_Name>
```

Prompt the user for an answer. The answer will be stored in the specified variable. Note that the variable name must not contain any white spaces. Once the answer is stored into the variable, you can then use that variable to reference the answer within SEND command (See SEND).

Example:

```
QUESTION "Customer Name ?" CustName  
SEND "Your customer name is |CustName|{ENTER}"
```

SEND

```
SEND <"Text to send">
```

Send Keystrokes to the current destination window. You must enclose the keystrokes to be sent with "".

Example:

```
SEND "Hello World."
```

You can make a reference to a variable within the *Text to send* by enclosing it with |.

Example:

```
;;= Ask a user for a customer name and store the answer in CustName  
QUESTION "Customer Name ?" CustName  
;;= Send the name out to a window  
SEND "|CustName|{ENTER}"
```

To send combination of keys such as Shift+F, Ctrl+F, Alt+F, precede the key character with the one of followings:

@	Alt
~	Shift
^	Ctrl

The example below illustrates sending Alt+F and then x to exit out of a program.

Example:

```
SEND "@fx"
```

Note: for certain DOS application such as EDIT.COM, you may need to toggle the key character twice in order to get the pull down menu response.

Example:

```
SEND "@ffx"
```

You can also send the special keystrokes to an application. The special keystrokes are as follows:

```
{F1}  
{F2}  
...  
{F12}  
{BKSP}  
{DELETE}  
{DOWN}  
{END}  
{ENTER}  
{ESCAPE}  
{HOME}  
{INSERT}  
{LEFT}  
{PGDN}  
{PGUP}  
{RIGHT}
```

```
{SHIFT}  
{SPACE}  
{TAB}  
{UP}
```

Example:

```
== Start notepad  
EXECUTE "c:\windows\notepad"  
  
== Wait for Notepad window for 10 seconds  
FOCUS "Untitled - Notepad" 10  
  
== send out the keystrokes  
SEND "Hello World.{ENTER}"
```

Tips

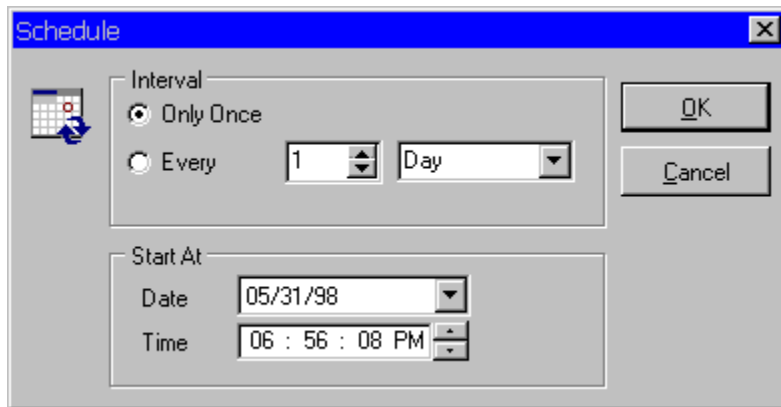
You can quickly type in {enter} or {tab} keystroke commands by using the popup menu by right clicking the mouse or by pressing the following key combinations:

{Enter}	Ctrl+ENTER
{Tab}	Ctrl+TAB

See Also

[Current destination window](#)

Schedule



[Schedule](#)

Schedule provides you with a way to automate the unattended task. Any task can be scheduled to execute at any Date and Time. In addition, you can set it to run once or . (See Shortcut Main Window to see how to activate the scheduled task)

Interval

Select **Only Once** if you want to run this script just once. Otherwise, select **Every** then choose the appropriate interval.

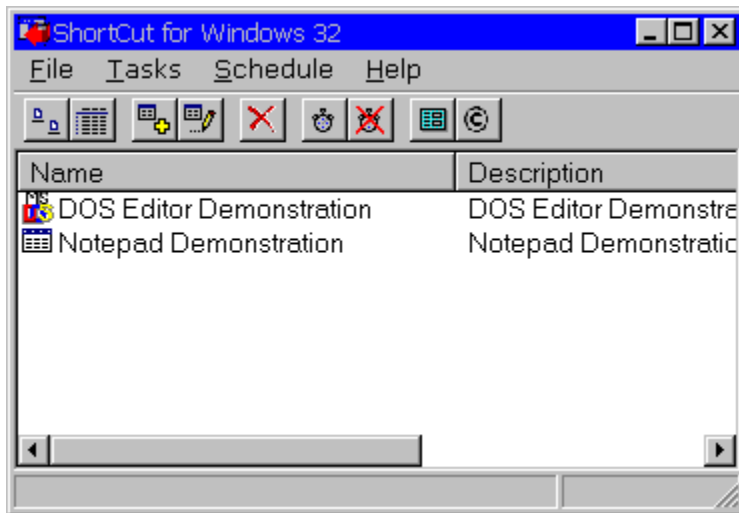
Start At

Specify appropriate Date/Time you want Shortcut to start this scheduling task.

See Also

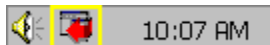
[Shortcut Main Window](#)

Shortcut Main Window



Main Window of Shortcut

Shortcut main window is simple and very easy to navigate. It consists of a menu, toolbar and a task list. When you click on one of the tasks, the detail description of that task will appear as a floating hint. To execute one of the tasks in the list, simply double click on the selection. You can also Run, Add, Edit or Delete the selected task by clicking the right-mouse button and select the appropriate selection from the popup menu. When you close this window, Shortcut minimizes itself into a taskbar icon (see figure below). To execute a task, simply press the left-mouse button and the task list will popup. To add, edit or delete a task, press the right-mouse button and a popup menu will appear (see Shortcut Taskbar Icon for more information).



Shortcut Taskbar Icon

Menu and Equivalent Toolbar Icons

File



Properties...



Close

[Change the properties setting of the shortcut.](#)

When you close this window, Shortcut minimizes itself into a taskbar icon.

Tasks



Run...

Execute the selected task.



Add...

[Add a new task](#)



Edit...

[Edit selected task](#)



Delete...

Delete Task: Delete the selected task. You will be prompted for confirmation.

Schedule



Start

Start Schedule: Begin the schedule run. Any task that has scheduling enabled will be put into suspended mode until the specified time and interval are met.



Stop

Stop Schedule: Cancel the schedule run.

Help



Content

Bring up Shortcut help.



About

About Window: Copyright information and registry information

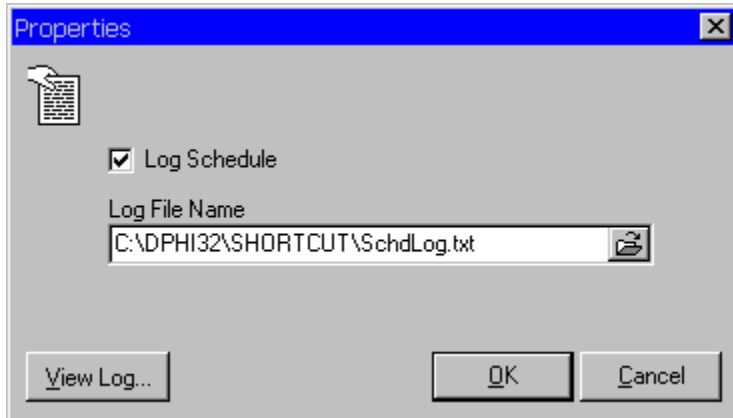


Toggle between Icon/List view

See Also

[Shortcut Taskbar Icon](#)

Properties



[Properties window](#)

Check Log Schedule if you want to keep log of the scheduled tasks. Choose the file name you want to save the log results to. Shortcut appends the new log to this file if the file already exists. Choose **View Log...** to view or edit the log.

See Also

[Schedule](#)

Shortcut Taskbar Icon



Shortcut Taskbar Icon

When you start Shortcut, you will notice the Shortcut icon on your taskbar. Shortcut taskbar icon serves as an access to Shortcut when it is hidden. It also serves as a state that Shortcut is in:



Shortcut is idle.



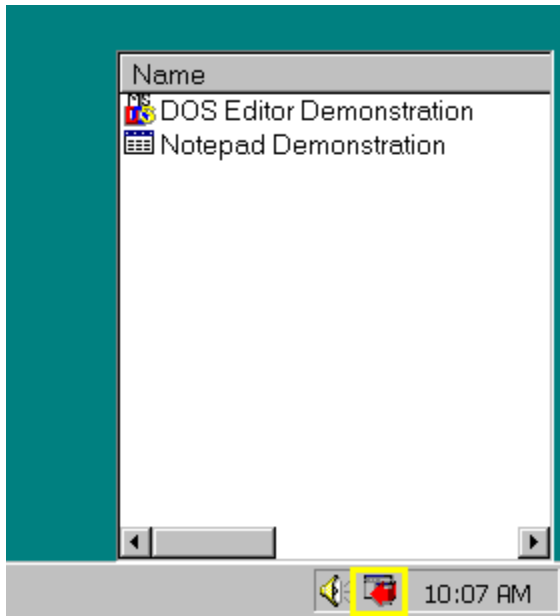
Shortcut is running.



Shortcut is running in schedule mode.

Left Click...

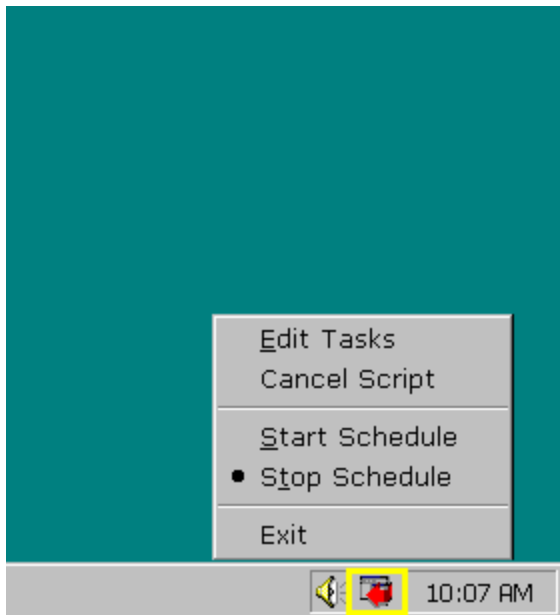
You can run Shortcut in a popup mode. In this mode, Shortcut hides itself from your desktop until you left click on the Shortcut taskbar icon. When the icon is clicked a popup window appears. To make a selection, simply click on a task selection you want to execute.



[Shortcut Popup Window](#)

Right Click...

When you press right mouse button, you will see a popup menu which you can



[Shortcut Taskbar Popup menu](#)

FOCUS

FOCUS <"Window Title"> <Time in Seconds>

Wait for the window to appear at the specified amount of time and then set the focus to it. If the window does not appear within the specified time then TIMEOUT flag will be set. You can use IF TIMEOUT (See IF TIMEOUT command) to evaluate the result of FOCUS command. If FOCUS command finds the matching window, this window will become the current destination window. Therefore, SEND, SYSCMD commands will be directed to this window instead of the initial executed window.

Example:

```
== Change the focus to "Untitled - Notepad"
FOCUS "Untitled - Notepad" 10
== If the window never appears within the specified time then
== Display the error
IF TIMEOUT TimeoutError
SEND "Hello Notepad."
GOTO ScriptEnd
:TimeoutError
DISPLAY "Time Out Error"
:ScriptEnd
```

See Also

[SEND](#)

[SYSCMD](#)

[IF TIMEOUT](#)

SYSCMD

SYSCMD <action>

Send standard system command to the current destination window. The valid commands are (case insensitive):

- CLOSE
- RESTORE
- MINIMIZE
- MAXIMIZE

Example:

```
== Maximize current destination window
SYSCMD maximize
== Minimize current destination window
SYSCMD minimize
== Restore current destination window
SYSCMD restore
== Close current destination window
SYSCMD close
```

See Also

[Current destination window](#)

IF TIMEOUT

IF TIMEOUT <Label to jump to>

Determine if the previous command causes the TIMEOUT condition. This command can be used with WAITDOS, WAITWIN and FOCUS (See FOCUS Command). If TIMEOUT do occurs then the script will jump to the specified label and resumes the execution from that point on.

Example:

```
== Wait for the word Name to appear at 1,1
== for 10 seconds
WAITDOS (1, 1, "Name") 10
IF TIMEOUT TimeoutError
SEND "Ok{enter}"
GOTO ScriptComplete

:TimeoutError
DISPLAY "Cannot Find text."

:ScriptComplete
```

See Also

[FOCUS](#)

[WAITDOS](#)

[WAITWIN](#)

WAITDOS (DOS Only)

```
WAITDOS (Row, Col, "Text") <Timeout in seconds>
```

Wait for a DOS text to appear at the specified Row and Column for the specified amount of time. The valid Row ranges from 1 to 25. The valid Column ranges from 1 to 80. If the text does not appear within the time, TIMEOUT flag will be set. You can then use IF TIMEOUT to evaluate the result of WAITDOS command. This command only works with DOS Application.

Example:

```
== Wait for the word Name: to appear for 10 seconds ==  
WAITDOS (1, 1, "Name") 10  
IF TIMEOUT TimeoutError
```

The example above waits for "Name:" to appear at Row 0 and Column 0 for 10 Seconds. If the text does not appear within 10 seconds TIMEOUT condition will be set. As a result, IF TIMEOUT will be executed (See IF TIMEOUT).

Note: use DOS capturing tool to help you getting the exact row and col from the DOS screen.

See Also

[IF TIMEOUT](#)

WAITWIN (Windows App. Only)

WAITWIN ("Window Title, "Text") <Timeout in seconds>

Wait for a text to appear within the specified window for the specified amount of time. If the text does not appear within the time, TIMEOUT flag will be set. You can then use IF TIMEOUT to evaluate the result of WAITWIN command. This command only works with windows Application.

Example:

```
== Wait for the word Name: to appear from within the == window "Windoze"
for 10 seconds
WAITWIN ("Windoze", "Name") 10
IF TIMEOUT TimeoutError
DISPLAY "Found Name"
GOTO EndIt

== Display the error
:TimeoutError
DISPLAY "Timeout waiting for text!"

:EndIt
```

The example above waits for "Name:" to appear at Row 0 and Column 0 for 10 Seconds. If the text does not appear within 10 seconds TIMEOUT condition will be set. As a result, IF TIMEOUT will be executed (See IF TIMEOUT).

Note: use DOS capturing tool to help you getting the exact row and col from the DOS screen.

See Also

[IF TIMEOUT](#)

Support

WEBSITE

<http://www.fabsoft.com>

FAX SUPPORT

Fax Support (973) 334-7020

MAIL-IN SUPPORT

FABSoft, Inc.

85 Washington Ct.

Towaco, N.J. 07082

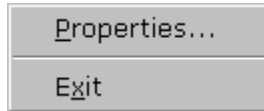
TELEPHONE SUPPORT

Telephone Number (973) 335-0313

Our supports open from 9:00AM-5:00PM Eastern time.

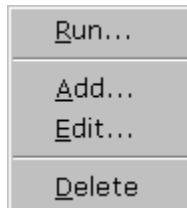
Menu Commands

File Menu



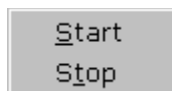
Properties... Change the settings (see Change the settings)
Exit Exit Shortcut.

Task Menu



Run... Execute the selected task.
Add... Add a new task
Edit... Edit the selected task.
Delete... Delete the selected task.

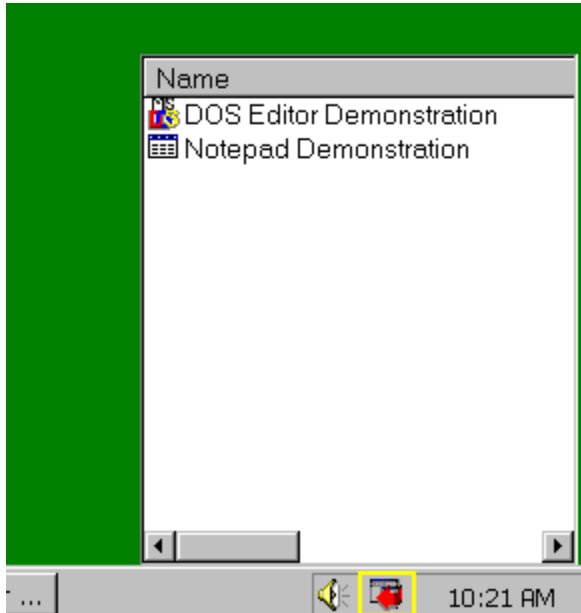
Schedule Menu



Start Start schedule mode.
Stop Stop schedule mode.

Shortcut Popup Mode

You can run Shortcut in a popup mode. In this mode, Shortcut hides itself from your desktop until you left click on the Shortcut taskbar Icon. When the icon is clicked a popup window appears. To make a selection, simply click on a task selection you want to execute.



[Shortcut Popup window in action](#)

Note: The width of the popup window is determine by the width of the *Name* Column. (see Shortcut Main Window)

See Also

[Shortcut Main Window](#)

Shortcut Command Line Parameter

Running Shortcut in Schedule Mode

Include *Schedule* as a parameter and Shortcut will start in Schedule mode.

Example

```
SHORTCUT schedule
```

Comment

`;<Comments>`

Comment is useful for commenting the script and make it more readable. Precede any comment line with ";". Note that anything after ";" is ignore by the script.

Example:

```
=== Wait for "Name" to appear for 10 seconds ===  
WAITDOS (1, 1, "Name") 10  
IF TIMEOUT TimeoutError  
SEND "Ok{enter}"  
GOTO ScriptComplete  
:TimeoutError  
DISPLAY "Cannot Find text."  
:ScriptComplete
```


Label

:<Label_Name>

Line Label is useful for conditional branching of the script. Precede the Label name with ":". The label name must not contain any white spaces.

Example:

```
WAITDOS (1, 1, "Date") 10
IF TIMEOUT TimeoutError
SEND "Ok{enter}"
GOTO ScriptComplete
:TimeoutError
DISPLAY "Cannot Find text."
:ScriptComplete
```

Predefined Variables

The followings are some predefined variables you can use from within the script:

BOM	Beginning of Month in the format of mm/dd/yy
EOM	End of Month in the format of mm/dd/yy
TIME	Current Time in the format of hh:mm:ss AM/PM

Note that Variables are always enclosed with ||

Example:

```
== Send beginning of month
SEND "The beginning of month is |BOM|"
== Send end of month
SEND "The beginning of month is |EOM|"
== Send current time
SEND "The beginning of month is |TIME|"
```

CALL

```
CALL <"Script Name">
```

This command will call other script and resume upon finish.

Example:

```
== wait for "Name" to appear at 1,1  
WAITDOS (1, 1, "Name") 10  
IF TIMEOUT TimeoutError  
CALL "c:\shortcut\scripts\amrscr.fcr"  
DISPLAY "Back from called script"  
GOTO ScriptEnd
```

```
:TimeoutError  
DISPLAY "Timeout Waiting for text!"  
:ScriptEnd
```

EXECUTE

```
EXECUTE <"Application Name">
```

This command will execute other application and resume the next script line without waiting for the application to finish.

Example:

```
EXECUTE "c:\windows\notepad.exe"  
FOCUS "Untitled - Notepad" 10  
SEND "Hello Notepad"
```

GOTO

GOTO <Label Name>

Skip to the specified Label Name.

Example:

```
GOTO TimeoutError
```

PAUSE

PAUSE <Number of Seconds>

Pause the script for Number of Seconds.

Example:

```
== Wait for 10 seconds  
PAUSE 10
```

SET DELAY

```
SET DELAY <milliseconds>
```

SET DELAY is used to set the internal setting of the script. It is used to control the amount of delay before SEND command actually sends each set of the keystrokes to an application. This is to overcome the problem with some applications that absorb the keystroke typeahead. SET Commands are normally placed in the very beginning of the script file.

Example:

```
== Set delay between SEND commands to half a second  
SET DELAY 500
```

Shutting Down Windows Automatically

Description

You can set this task to activate at the specific time.

Task information

Command: c:\windows\taskman.exe

Path: c:\

Window Title: Tasks

Application Type: Windows Application

Auto Close: Unchecked

Script

```
-- Send Alt-F and u to signify the shutdown of windows
SEND "@Fu"
-- Wait for the shutdown confirmation to appear then
-- Select the default selection
FOCUS "Shut Down Windows" 10
SEND "{enter}"
```


Troubleshooting

Problem: Keystrokes send out too fast.

Solution: Use SET DELAY to control the amount of milliseconds to wait between each SEND command. Then, break the keystrokes down into several SEND commands.

Problem: Shortcut only does not execute all the commands specified within the script.

Solution: You may be running the evaluation version. Evaluation version limits number of script lines to 30.

Problem: Keystrokes Recorder does not record any keystrokes activities from within a DOS session.

Solution: Windows 95 operating system does not notify any keystrokes activities from within any DOS session. Therefore, Keystrokes Recorder has no way of receiving any keystroke messages from Windows 95. However, Keystrokes Recorder works under Windows NT operation system.

