

## **Fist Script Reference Manual**

Sachihiko Kadotani  
Copyright and All Rights Reserved  
Jun 26.1999

### **Contents**

- [ **Applicable Cases** ]
- [ **Sample Fist Scripts** ]
- [ **Start Fist** ]
- [ **Terminate Fist** ]
- [ **Fist Command Format and Configuration Lines**]
- [ **Fist Command Reference** ]
- [ **Rules of Keystrokes** ]
- [ **Detailed Descriptions** ]
- [ **How to make Fist Job Plans** ]

### **[Applicable cases ]**

Applicable to all cases as follows

Initializing a network configuration automatically such as TCP/IP configuration dialogues.

Handling GUI applications seamlessly.

FIST responds to a GUI installer of new software version for your distributed clients remotely.

Demonstrating your new GUI software attractively.

Operating Power Point automatically on the international conference.

Providing smooth description of your product on your mobile PC in your client's office.

Sending/Receiving mail automatically

Schedule playing of a musical instrument.

For any other unmanned PC operation

FIST has many batch commands and automatic filename generator

Improving efficiency of your software production line

Automatic System shutdown and Power off available

Needless programming ability

FIST is so small size and portable

## [ Sample Fist Scripts ]

Creating HTML document and starting browser. (Ncmd1.txt)

```
"" ,0,"PASSWORD"
"" ,0,"Ncmd1.log"
"" ,0,"-"
"" ,1000,"-"
"" ,1,"!EXEc:¥Program Files¥Hidemaru¥HIDEMARU.EXE"
"" ,1,"<html>{ENTER}"
"" ,1,"<body bgcolor={ }#ffffcc{ }>{ENTER}"
"" ,1,"<Font size=5>{ENTER}"
"" ,1,"<pre>{ENTER 2}"
"" ,1," ----- Test -----{ENTER 2}"
"" ,1,"Item{TAB 2}$60.00{ENTER}"
"" ,1,"</pre>{ENTER}"
"" ,1,"</body>{ENTER}"
"" ,1,"</html>{ENTER}"
"" ,1,"%F"
"" ,1,"A"
"" ,1,"htest3.html{ENTER}"
"" ,1,"%F"
"" ,1,"X"
"" ,1,"!EXEe:¥Program¥netscape.exe"
"" ,2,"{TAB}file:///C | /WINDOWS¥desktop¥ntest¥htest3.html{ENTER}"
"" ,2,"!END"
```

Starting LotusNotes and send mail and close it automatically. ( another Ncmd1.txt)

```
"" , 0,"PASSWORD"
"" , 0,"-"
"" , 0,"-"
"" ,500,"C:¥notes¥notes.exe"
"" , 10 , "{RIGHT}"
"" , 1 , "{ENTER}"
"LOTUSPASSWORD",1 , "kadotanisachihiko"
"" , 1 ,"{ENTER}"
"" , 5 ,"% (C)"
"" , 1 , "E"
"TO", 1 , "aoki"
"" , 1 ,"{ENTER}"
"" , 1 ,"{DOWN 3}"
"SUBJECT",1 , "Report"
"" , 1 ,"{DOWN}"
"" , 1 , "Hello what are you doing there?{ENTER 2}"
"" , 2 , "% (A)"
"" , 2 , "E"
"" , 2 , "% (F)"
"" , 1 , "{DOWN 14}"
"" , 1 , "{ENTER}"
"" , 1 , "!END"
```

## [ Start Fist ]

After Fist starts by your double-click on FIST icon, Fist reads "Ncmd1.txt"(default name) as a Fist script file. So you make Fist script file and save it as Ncmd1.txt beforehand.

Once you start Fist , you can start another Fist task from the script file like" !RUNFist.exe fcmd.txt".

Also you can make any name as a Fist script file on its argument.

And when Fist gets to the end of the Fist script (!END), if you have set a following Fist job on the third line in the Fist script file, Fist will execute the next Fist script job you have made.

## [ Terminate Fist ]

Fist terminates by "!END" or "!STP". "!STP" means to terminate current Fist process and cancel following Fist job if you assign next Fist job on the third line.

## [ Fist Command Format and Configuration Lines ]

**Fist command format consists of a comment, waiting time and command.**

Both a comment and a command need double quote as characters.

No need for waiting time. These three parts are separated by commas.

From the first line to the fourth line of the Fist script are indispensable lines for configuration statements. From the fifth line to the end line (!END) are acceptable any Fist commands and keystrokes. After the end line, Fist ignores following statements. You may place your useful parts of Fist scripts on the next to "!END".

**Description about Rules of Configuration Lines ( From 1st line to the 4th line )**

**The first line** ( for the license key)

Fist distinguishes trial product and licensed one by this line.

If you are a licensee , place your license number here. If not, place "GUEST" instead.

Ex) "" ,0,"GUEST"

**The second line** ( for log file name or starting monitoring tool )

(1) output log record with process name of current focus of its application.

(2) monitoring executing script line on the task bar.

(3) monitoring x-y coordinates of the current mouse pointer on the screen.

Example:

"LOG",0,"-" ← NO make log file

"LOG",0,"Ncmd1.log" ← NO make log file

"LOG",1,"Ncmd1.log" ← make log file as Ncmd1.log

- “LOG”,2,”Ncmd1.log” ← make log file include process name for **!GFC** and **!CFC**
- “LOG”,3,”Ncmd1.log” ← monitoring your comment of current script on the task bar
- “LOG”,4,”Ncmd1.log” ← monitoring line number of current script on the task bar
- “LOG”,50,”Ncmd1.log” ← monitoring x-y coordinates of the current mouse pointer  
This function is useful for **!MVW** or **!MMV** ...,etc.

**The third line** ( for continuous Fist script file)

You can start following Fist job after the last script line (!END), if you assign the script file name here. This line is a connector for continuous Fist job.

Ex.

“Next Job”,1,”Ncmd2.txt”

You may not want to assign following Fist job, you can ignore this line as follows.

“Next Job”,0,”Ncmd2.txt” or

“Next Job”,0,”-”

**The fourth line** ( for execution speed of Fist script and for initial start an application )

This value should be milliseconds. 1,000 milliseconds = 1 second.

The smaller this value , the faster of Fist script execution speed.  
Even though you place 1,000 milliseconds here, if there are heavy tasks on your script, Fist needs more wait time to synchronize. So you can give this wait time on each script line after this statement.

Also you can place path for an application here in order to have the application start here.

Fist script has two commands in order to start an application, **!RUN** and **!EXE**.  
However needless to use these commands in this line because this third line is a special line.

Ex.

“,1000,”c:¥windows¥Regedit.exe”

You may not want to start an application here , you can place hyphen instead.

Ex.

“,1000,”-c:¥windows¥Regedit.exe” or

“,1000,”-”

You may write arguments for the application after one blank .

Also you may change execution speed by **!CTM** dynamically.

### **Difference between !RUN and !EXE.**

You can use these commands after fourth line in order to start an application.

!RUN command is **able to accept arguments** of the starting application.  
However !EXE command is unable to any arguments.

!RUN command is submitted **asynchronously**.  
If the application has not started yet, Fist goes next command.

On the other hand, starting application by !EXE command, Fist waits until the application gets ready to accept data.

These things have the expected consequence that the wait time value of next to !EXE command means more waiting time after the application has been able to accept data.

Remember,

!EXE command is a synchronous command. !RUN is asynchronous.

**From the fifth line to the end of the scripts (= "!END")**

**Please refer to [ Fist Command Reference ]**

**The end line of the Fist script**

**"",1,"!END"**

This means end line of the script file.

If you have assigned following Fist job on the third line, you may trim this wait time value for starting next script job file, because next Fist task is started after this line.

## [ Fist Command Reference ]

### Sending keystrokes

“abcdefg”

Please Refer to [ **Rules of Keystrokes** ]

### Starting an application

**!RUN** starting an application asynchronously

“!RUN:¥Windows¥calc.exe”

**!EXE** starting an application synchronously (No arguments acceptable)

“!EXEe:¥Program¥netscape.exe”

### Windows control

**!GFC** Get Focus

“!GFCThe application title”

You can get the title name easily to set **2** on the log file configuration line.  
You can copy and paste the name from the Fist log file.

**!SH1** Save Handler 1 ( you can use max five registers **!SH1,,,4** and **!SH5.** )

“!SH1”

You can get window handler on current focus.  
This handler will be used for **!SWT,!ICZ,!OPI,!ZOO,!MVW** and **!MWW** like this “!MWW(SH1)300,300”.

**!SWT** Set Window Title

“!SWT(SH1)G1”

It is useful command in order to get an application title for **!GFC**.  
Because you can give simple name to the application for easy to handle.

**!ICZ** Iconize an applicaton window to task bar

“!ICZ(SH1)”

**!OPI** Open Icon on task bar for an application widow

“!OPI(SH1)”

**!ZOO** Maximize (Zooming) an application window

**“!ZOO(SH1)”**

**!MVW** Move Window to the point you want to.

**“!MVW(SH1)230,180”**

You may need to know the value of the point ,you can place value 50 on the log file configuration line . Fist has monitoring tool run to get x-y coordinates of the current mouse pointer.  
The value will be monitored on the task bar.

**!MWW** Move( change ) window with width and height

**“!MWW(SH1)230,180”**

You may need to know the value of the point ,you can place value 50 on the log file configuration line . Fist has monitoring tool run to get x-y coordinates of the current mouse pointer.  
The value will be monitored on the task bar.

#### **Mouse control**

You may need to know the value of the current mouse pointer , you can place value 50 on the log file configuration line .  
Fist has monitoring tool run to get x-y coordinates of the current mouse pointer. The value will be monitored on the task bar.

**!MMV** Move Mouse pointer

**“!MMV160,296”**

**!MLC** Left Click

**“!MLC160,296”**

**!MDC** Double Click

**“!MDC160,296”**

**!MRC** Right Click

**“!MRC160,296”**

**!MLK** Lock mouse pointer on Zero

**“!MLK”**

After Fist starts ,a mouse pointer is locked on zero screen coordinates.  
However, once you have used these mouse control command,  
The lock is released until you lock it again by !MLK.

This command has no operand.

**!MLD** Left Down and Hold

**"!MLD160,296"**

This command is usually used for beginning to drag

**!MLU** Left Up

**"!MLU160,296"**

for releasing left button

### Conditional Operations

**!CFC** Compare Focus

**"!CFCThe application title"**

If current window's title name is not this name,  
Fist displays a dialogue with YES/NO buttons.  
An operator selects YES , Fist continues next script.  
If selects NO, Fist terminates.

You can get the title name easily to set **2** on the log file configuration line.  
You can copy and paste the name from the Fist log file.  
This command is usually used with !GFC .

**!FCK** File Check

**"!FCKc:¥temp¥txt.txt"**

If Fist fails to find out the file name on the path,  
Fist displays a dialogue with YES/NO buttons.  
An operator selects YES , Fist continues next script.  
If selects NO, Fist terminates.

**!SCF** Set Check Flag

**"!SCFc:¥temp¥txt.txt"**

Fist checks a file and saves its result,  
after this, you can test the condition by **!RIF**.  
This command is usually used with !RIF in pairs.

**!RIF** Run If Flag is On

**"!RIFc:¥program¥Fist.exe recov.txt"**

If the result of **!RIF** is TRUE ,Fist has an application run.

**!GF1** Get File Information 1

“**!GF1**c:¥temp¥txt.txt”

**Refer to Condition Matrix about !CFI and !CFJ**

**!GF2** Get File Information 2

“**!GF2**c:¥temp¥txt.txt”

**Refer to Condition Matrix about !CFI and !CFJ**

**!CFI** Compare File Information

“**!CFI**c:¥program¥Fist.exe recov.txt”

**Refer to Condition Matrix about !CFI**

**!CFJ** Compare File Information

“**!CFJ**c:¥program¥Fist.exe recov.txt”

**Refer to Condition Matrix about !CFJ**

<b>Condition Matrix about !CFI</b>				
Target file's condition	Result of <b>GF1</b>	Result of <b>GF2</b>	condition	Action of <b>CFI</b>
	<b>NON EXIST</b>	<b>NON EXIST</b>	<b>unchanged</b>	<b>EXECUTE</b>
	<b>EXIST</b>	<b>EXIST</b>	<b>UpdateTime Unchanged</b>	
			<b>UpdateTime changed</b>	<b>NO EXECUTE</b>
	<b>NON EXIST</b>	<b>EXIST</b>	<b>changed</b>	
	<b>EXIST</b>	<b>NON EXIST</b>	<b>changed</b>	

Condition Matrix about !CFJ				
Target file's condition	Result of GF1	Result of GF2	condition	Action of CFJ
	NON EXIST	NON EXIST	unchanged	NO EXECUTE
	EXIST	EXIST	UpdateTime Unchanged	
			UpdateTime changed	EXECUTE
	NON EXIST	EXIST	changed	
	EXIST	NON EXIST	changed	

**!CFX** Display Message dialogue after Compare File Information

"!CFXIF continue select YES"

!CFX does the same judgement of !CFI, If the result is EXECUTE, dialogue with YES/NO button displays with your message.

**!WGF** Wait until Get Focus successfully

"!WGFThe application title"

You can get the title name easily to set **2** on the log file configuration line. You can copy and paste the name from the Fist log file.

**!WLF** Wait until Lost Focus

"!WLFThe application title"

You can get the title name easily to set **2** on the log file configuration line. You can copy and paste the name from the Fist log file.

## Message dialogue

**!MBX** display dialog box

**“!MBX** If you select YES ,continues”

Fist displays message dialogue with YES/NO button.  
If an operator selects YES, Fist continues next script.  
If select NO, Fist terminates.

**!MSG** set message on Fist window

**“!MSG** Let me introduce myself....”

Fist, itself is needless a foreground window.  
Fist's foreground window is usually iconized by default.  
When you need to display messages to an operator, you  
can set your messages and display it on the screen.  
After you set your message by **!MSG** ,you need to do  
**!STD** or **!MAX** in order to display it on the screen.  
Also you can iconize the message by **!MIN**.

**!STD** Standardize Fist window

**“!STD”**

**Refer to !MSG**

**!MIN** Minimize Fist window

**“!MIN”**

**Refer to !MSG**

**!MAX** Maximize Fist window

**“!MAX”**

**Refer to !MSG**

## Batch commands

**!KIL** Delete File

```
"!KILc:¥temp¥work¥*.bak"
```

You can use a wild card such as "\*.\*".

However in case of including directories in target files, !KIL can not eliminate the directories. Because !KIL is a command for deleting "File" not for "Directory". If you need to delete those directories, you can use **!RMD** command beforehand.

Recommendation : If you want to delete large amount of files including Many directories, you can have EXPLORER do it.

For example

```
"!RUNc:¥explorer.exe"  
"%T"  
"FF"  
"*.*"  
{TAB 3}  
"TargetDirectoryName"  
{ENTER}  
"%EA"  
"%FD"  
{ENTER}
```

**!RMD** Remove Directory

```
"!RMDc:¥temp¥temp"
```

When you remove a directory, you have to clear out subordinate all files and directories in the target directory beforehand.

**Refer to !KIL.**

**!MKD** Make Directory

```
"!MKDc:¥temp¥temp"
```

Create new directory.

Also you can set **!N\*G** to the operand like this

```
"!MKDc:¥temp¥!N*G".
```

**Refer to !N\*G.**

**!FCY** Copy file

**"!FCYfromtemp.txt,totemp.txt"**

Also you can set **!N\*G** to the second operand like this  
**"!FCYtext.txt,c:¥temp¥!N\*G"**.  
You can not place it on the first operand.

**Refer to !N\*G.**

**!NAM** changing file name

**"!NAMoldname.txt,newname.txt"**

Also you can set **!N\*G** to the second operand like this  
**"!NAMtext.txt,c:¥temp¥!N\*G"**.  
You can not place it on the first operand.

**Refer to !N\*G.**

**!N\*\*** Name [Create Unique name]

**"!N\*\*.txt"**

Fist creates name and memorizes it in order to be retrieved by **!N\*G** .

**"!N\*\*.html"** ← To memory as **N18004018.html**  
**"!N\*G"** ← retrieves the name and ketstrokes as **N18004018.html**  
**"!N\*\*"** ← To memory as **N18004023**  
**"!N\*G"** ← retrieves the name and ketstrokes as **N18004023**

Also you can apply for **!MKD,!FCY** and **NAM** as follows:

**"!N\*\*"**  
**"!MKDc:¥temp¥!N\*G"**  
**"!N\*\*.txt"**  
**"!FCYtext.txt,c:¥temp¥!N\*G"**  
**"!NAMtext.txt,c:¥temp¥!N\*G"**

or you may use in an application's saving dialogue box.

You can place this command as Fist's log file name.  
**"",1,"!N\*\*.log"** on the second line.

**Refer to !N\*G.**

**!N\*G** Recall Just Before Create Name

**“!N\*G”**

Fist keystrokes stored name just before creating unique name.

**“!N\*\*.html”** ← To memory as **N18004018.html**

**“!N\*G”** ← retrieves the name and ketstrokes as **N18004018.html**

**“!N\*\*”** ← To memory as **N18004023**

**“!N\*G”** ← retrieves the name and ketstrokes as **N18004023**

Also you can apply for !MKD,!FCY and NAM as follows:

**“!N\*\*”**

**“!MKDc:¥temp¥!N\*G”**

**“!N\*\*.txt”**

**“!FCYtext.txt,c:¥temp¥!N\*G”**

**“!NAMtext.txt,c:¥temp¥!N\*G”**

or you may use in an application’s saving dialogue box.

**Refer to !N\*\*.**

**!CHD** Change Current Directory

**“!CHDc:¥temp”**

**!CHV** Change Current Drive

**“!CHVF”**

### **Force System Shutdown**

**!DWN1** ShutDown

**“!DWN1-14:40:45”**

System will shutdown after 14:40:45.

This command should be used on a cyclic Fist script file.

**!DWN2** Reboot

**“!DWN2-14:40:45”**

System will reboot after 14:40:45.

This command should be used on a cyclic Fist script file.

**!DWN8** System Power Off

"!DWN8-14:40:45"

System will shutdown and be off after 14:40:45.

This command should be used on a cyclic Fist script file.

## Script Control

**!STA** Start Job after time expired

"!STA0:40:45"

Fist will have start script file as sta.txt script after 0:40:45.

This command should be used on a cyclic Fist script file.

**!SCL** Fist Force Terminate after time expired

"!SCL0:40:45"

Fist will stop after 0:40:45.

This command should be used on a cyclic Fist script file.

**!CTM** Change execution Timer speed

"!CTM3000"

You can control Fist's execution speed dynamically.

This sample indicates for changing execution speed to 3 seconds.

**!STP** Stop Fist and Cancel Following Fist job

"!STP"

Fist terminates and cancels following Fist job.

**!END** Fist terminates and start following Fist job

Fist terminates.

And following Fist job will be started if you have assigned on the third line as a continuous statement.

- hyphen means Non-operational line on the first column of a command line

"-!RUNc:¥windows¥explorer.exe"

## For synchronous operation

**!IDA** Is directory alive

“!IDAc:¥temp¥temp”

If the directory exists, Fist terminates.

**!WFD** Wait until Find Directory

“!WFDc:¥temp¥temp”

Fist waits until If the directory is made.

**!WLD** Wait until Lost Directory

“!WLDc:¥temp¥temp”

Fist waits until If the directory is eliminated.

**!WFI** Wait until File Information changes

“!WFIc:¥temp¥temp.txt”

Fist waits until If the file information is changed.  
The condition of judgement is the same as !CFI table.

**Refer to !CFI.**

**!WFJ** Wait until File Information changes

“!WFJc:¥temp¥temp.txt”

Fist waits until If the file information is changed.  
The condition of judgement is the same as !CFJ table.

**Refer to !CFJ.**

**!WGF** Wait until Get Focus successfully

“!WGFThe application title”

You can get the title name easily to set **2** on the log file configuration line.  
You can copy and paste the name from the Fist log file.

**!WLF** Wait until Lost Focus

“!WLFThe application title”

You can get the title name easily to set **2** on the log file configuration line.  
You can copy and paste the name from the Fist log file.

## [ Rules of Keystrokes ]

All keystrokes need double quote as characters.

Also special function characters need "{" to distinguish normal characters.

{+} {^} {%} {~} {() }

If you use { or } as a character you can express {} or {}.

If you include double quote within your keystrokes, you make single quote {} instead of double quote.

Such as Enter key or Tab key are as action keys. Also these key are used with {}.

BackSpace	<b>{BS}</b>	Ctrl + Break	<b>{BREAK}</b>
CapsLock	<b>{CAPSLOCK}</b>	Del or Delete	<b>{DEL}</b>
	<b>{DOWN}</b>	End	<b>{END}</b>
Enter	<b>{ENTER}</b>	Esc	<b>{ESC}</b>
Help	<b>{HELP}</b>	Home	<b>{HOME}</b>
Ins or Insert	<b>{INS}</b>		<b>{LEFT}</b>
NumLock	<b>{NUMLOCK}</b>	PageDown	<b>{PGDN}</b>
PageUp	<b>{PGUP}</b>	PrintScreen	<b>{PRTSC}</b>
	<b>{RIGHT}</b>	ScrollLock	<b>{SCROLLLOCK}</b>
Tab	<b>{TAB}</b>		<b>{UP}</b>
F1	<b>{F1}</b>	F2	<b>{F2}</b>
F3	<b>{F3}</b>	F4	<b>{F4}</b>
F5	<b>{F5}</b>	F6	<b>{F6}</b>
F7	<b>{F7}</b>	F8	<b>{F8}</b>
F9	<b>{F9}</b>	F10	<b>{F10}</b>
F11	<b>{F11}</b>	F12	<b>{F12}</b>
F13	<b>{F13}</b>	F14	<b>{F14}</b>
F15	<b>{F15}</b>	F16	<b>{F16}</b>
Shift	+		
Ctrl	^		
Alt	%		

### Combination with Shift, Ctrl and Alt and other keystrokes.

In case of shift in and AB is "+(AB)"

In case of shift in and E and shift out C is "+EC"

### How to Repeating same character

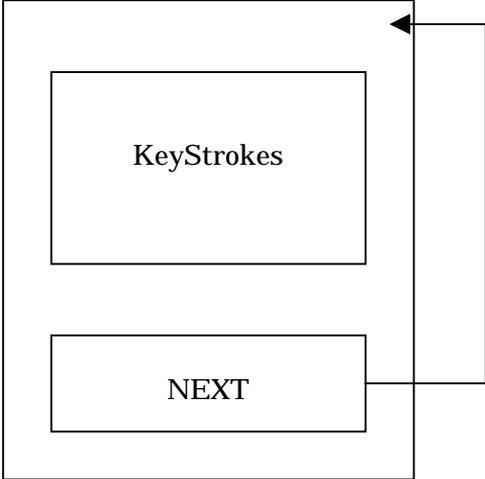
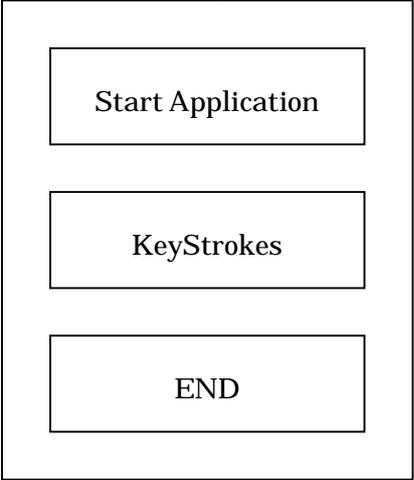
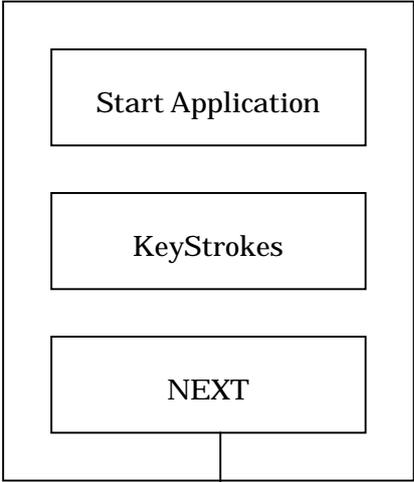
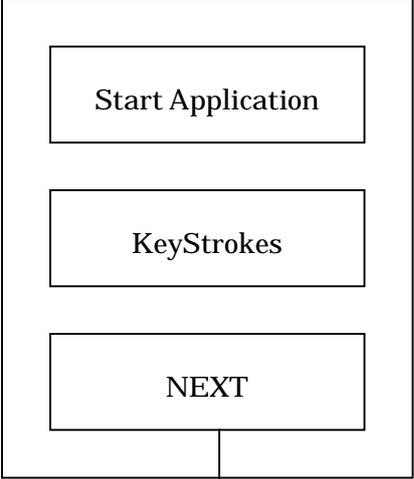
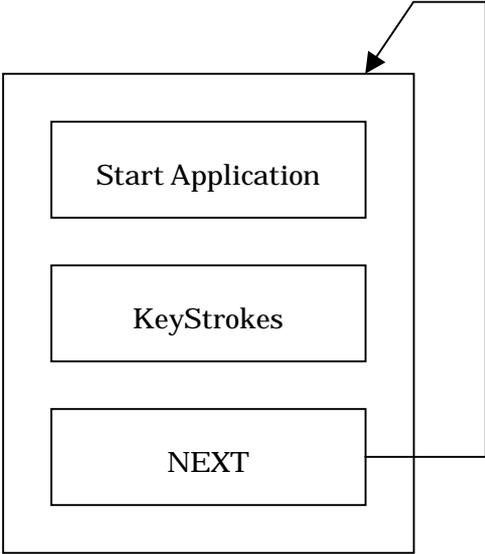
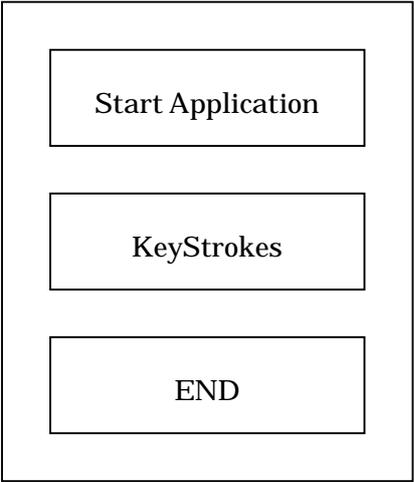
{g 10} will strokes as gggggggggg.

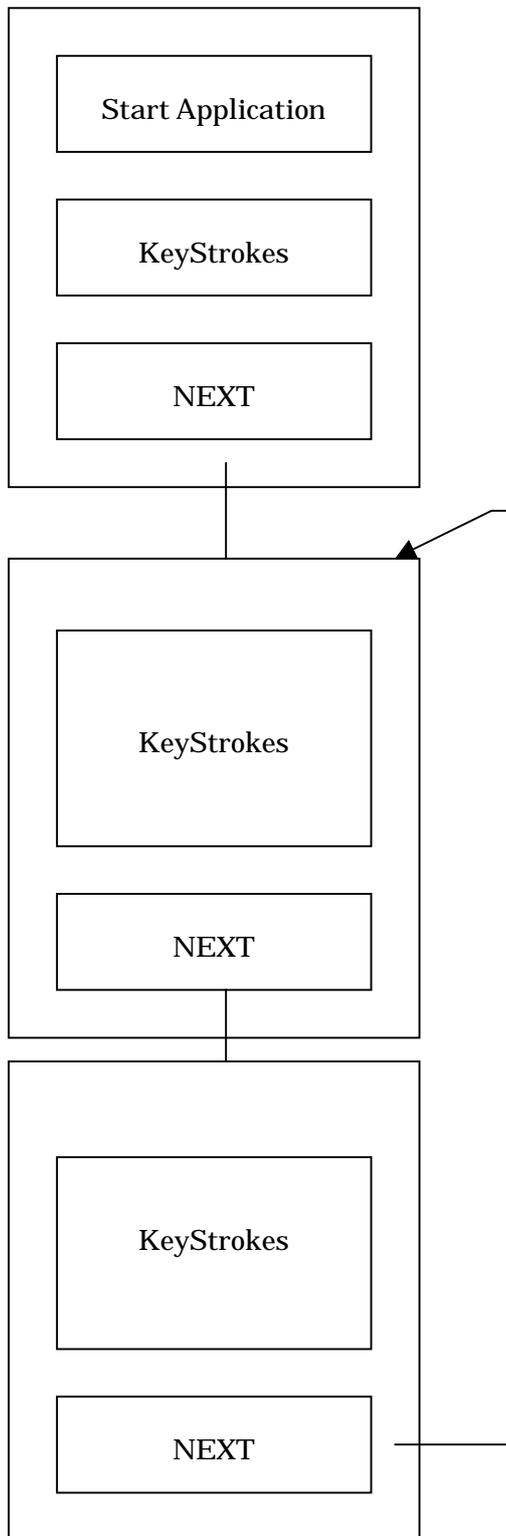
Three times of {TAB} is {TAB 3}.

{RIGHT 15} means fifteen times of {RIGHT}.

You have to give one space in front of repeating value.  
Sorry, Copy key is not acceptable on the Fist script.

Cyclic Fist Script File Patterns





Copy Rights (c) P-6387-1,P-6388-1 and P-6417-1 (registration in Japan)  
Jun 1999

(author) Sachihiko Kadotani

[a800@ask.ne.jp](mailto:a800@ask.ne.jp)

<http://www.ask.ne.jp/~a800/index3.html>

#101 Terrace-Oji,5-6-23 Oji, Kita-ku,  
Tokyo 1140002,  
Japan