

# DASH BOARD SCRIPTER'S GUIDE

Version 1.5, March 1, 1999

Dash Board version at publication time: 1.5

The ***Dash Board Scripter's Guide*** is intended to help you get the most out of Dash Board's scripting capabilities. In the sections that follow, you'll read about topics that will help you create Dash Board scripts. This book covers how Dash

# INTRO TO SCRIPTING WITH DASH BOARD

## The Script Frame

A Dash Board **script** is simply a bit of NewtonScript source code. When a script is chosen from the Newton Menu or Letter Launcher, Dash Board looks up the soup entry, reads the source code, and compiles it into a function object.

Assuming there are no errors in compilation, Dash Board then creates a **script frame**. A script frame is a frame containing various slots, as described below. One of those slots contains the function object that resulted when the script code was compiled.

Dash Board then sends a message to the frame, to execute that function object.

Let's look at the script frame (items between the "<>" are English representations of the slot's contents):

### a Dash Board script frame

```
scriptFrame :=  
{  
  <_preference to Dash Board itelf>,  
  <_parent:>
```

```
:Notify(3, "A Notify Alert", "I am inheriting the Notify  
method from the root view.");
```

The next two slots are also significant. The `|scriptFunc:FiveSpeed|`

`info` slot contains a string that should contain a brief informative description

slot contains a symbol that should be unique, so it should contain a variant of your unique developer's signature. For example, Five Speed

`'|SomeScript:FiveSpeed|`. The  
 ":",") and your unique

If you do not have a unique developer's signature, and you think you might want with Apple Computer, Inc. for a registered developer signature.

We encourage you to do this as soon as possible, as Apple has been very lackluster col supporting the Newton after its discontinuation. We think that there

suitable outrage if that actually occurs, and insist that they continue maintaining the registry, it would seem prudent to register for aTjque developer signature as Also, be kind to the actual folks who process your registration-  
 -they don't have anything to do with Apple's poor treatment of Newton users and developers.)

## USING INHERITED DASH BOARD METHODS

Dash Board 1.5 provides five methods which you can use in your scripts:

**DashBoardDoDialog()** - prompts the user for input and returns a value

**DashBoardGetScriptjresult()** - executes another script, and returns the resulting value

**DashBoardDoSpecialItem()** - executes a Dash Board Special Item, and returns **NIL**

**DashBoardInsertText()** - inserts text at the caret, in a standard text entry view such as the Notes app, or in an editable TXview, such as those used by the Works app

**DashBoardFakePNButton()** – This command is for users of howerNames, the Names enhancement from SilverWARE (<http://www.silverware.com>). It simulates a tap on the special howerNames button bar icon.

A25 TD shown in the preceding chapter, your scripts can access these built in Dash Board methods through proto inheritance. You should call these method25 TD simply by sending a message to the script frame, so all that is required is a trailing colon

## The DashboardDoDialog( Method

This method displays a modal dialog, with a prompt, an input line for the user to enter an answer, a cancel button, and an enter button. The dialog can be customized depending on the arguments your script supplies. The arguments to this method are described below.

*aScriptFrame*:DashboardDoDialog(*theValueType*, *theMessage*, *theTitle*,

*aScriptFrame* - the script frame. You can just use

*theAnswer*, *theCancelBtnTxt*, *theOkBtnText*

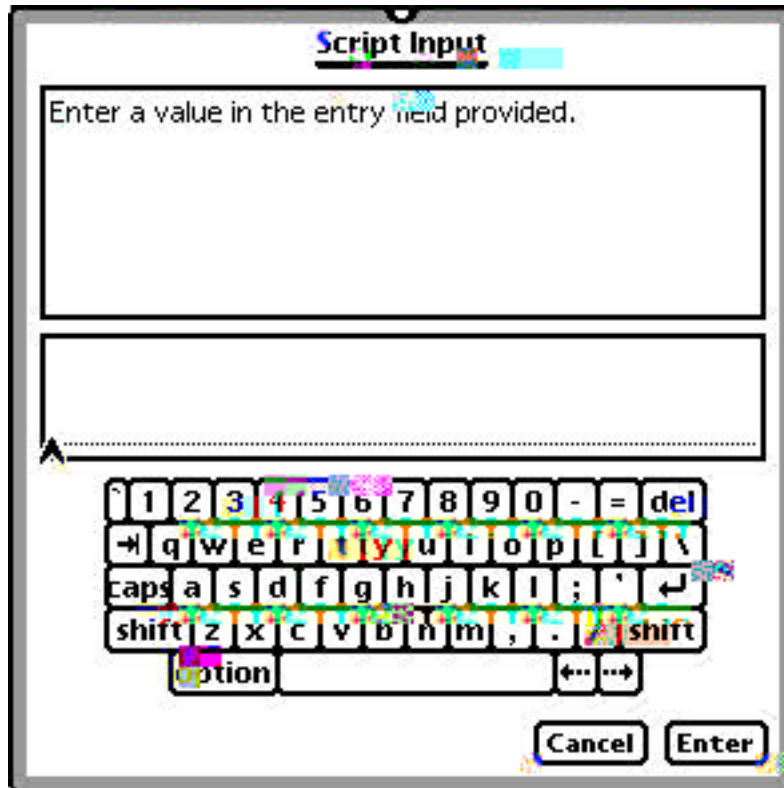
If the user taps the enter button, `DashboardDoDialog()` returns the value entered by the user, coerced to the type specified by the *theValueType* argument. Otherwise, if the user taps the cancel button, this method returns `NIL`.

The following code produces the dialog shown below:

**Figure 4. - a generic use of the DashboardDoDialog() method**

```
:DashboardDoDialog(NIL, NIL, NIL, NIL, NIL, NIL);
```

---





Taking advantage of the customization options of this method, we can create a more interesting dialog:

**Figure 5. - using DashboardDoDialog() with parameters specified**

```
:DashboardDoDialog('string, "Isn't Dash Board cool?",  
"Stupid iuestion Dialog", "Yes, very", "I jefuse To  
Answer", "Enter My Answer");
```

The DashboardDoDialog() method should be useful when you want to write a script that requires some user input; for example, a jeminder script might prompt the user for the number of minutes later that they want to be reminded, and then set a system alarm for that number of minutes in the future.

## Method

then use that data for whatever it needs to do.

*scriptName*

## Method

*aScriptFrame*:DashboardDoSpecialItem(*specialItemSymbol*)

'cmdGuestModeOff - HWR Guest Mode Off  
'cmdGuestModeOn - HWR Guest Mode On  
'cmdKbd1 - Keyboard  
'cmdKbd2 - Keyboard (Numeric)  
'cmdKbd3 - Keyboard (Phone)  
'cmdKbd4 - Keyboard (Date)  
'cmdLLHelp - Show LL Shortcuts  
'cmdMIAAG - MI At A Glance  
  
'cmdMIEvent - MI New Event  
'cmdMIMtg - MI New Meeting  
'cmdMITodo - MI New Todo  
'cmdMute - Volume Mute  
'cmdNewCall - New Call  
'cmdNewChecklist - New Checklist  
'cmdNewEvt - New Event  
'cmdNewMtg - New Meeting  
'cmdNewNote - New Note  
'cmdNewOutline -

## The DashboardInsertText( Method

This method inserts a text string at the caret. It works in any standard text entry view, such as the Notes app, and also works with views based on the protoTXView, used in the Works app.

*aScriptFrame*:DashboardInsertText(*theString*

*aScriptFrame* - the script frame. You can just use :DashboardInsertText without explicitly specifying this.

*theString* - a text string. This is the text that will be inserted at the caret.

The return value of this method is undefined.

### example of the DashboardInsertText( method:

```
local myText :=  
"John Doe  
123 Main Street  
New York, NY 12345";
```

## **The DashboardFakePNButton() Method**

This method simulates a tap on the special button bar icon that ships with

Dash Board can import scripts written for GestureLaunch 3, from Innovative

Scripters should be aware of.

## ABOUT EXPORTING SCRIPTS

Dash Board can now export scripts, as well as importing them. There are two types of Export.

### **Notes (Text) Export**

To export a script or scripts to the notes program, go to the Scripts panel of the Dash Board Prefs app. Select the script(s) you want to export, and tap the Export button. Choose "Export to Notes" from the popup menu.



We're always happy to hear suggestions for what we can do to improve your

Also, please check out the Dash Board Scripting page