

A

Source Files

This book comes with an accompanying diskette containing code and examples. Neither the author nor the publisher warrants the code or examples in any way. There is no warranty that they are correct, or that they will fulfill their stated purpose. In particular, you include any additions or changes to system classes at your own risk.

On the diskette are three different types of code. First, the `.ex` files in the `examples` directory contain example code that can be highlighted and evaluated. The files contain code matching the examples in the text. These files should *not* be filed in. Instead, you can look at them using a File List or a File Editor, and evaluate the code directly in these tools. Or, if you prefer, you can cut and paste the code from these tools into a Workspace and evaluate it there.

Second, the files in the `classes` directory contain classes that can be filed in. These classes illustrate points that are made in the text. Normally there is an `examples` protocol on the class side that contains methods to execute.

Third, the files in the `sysadds` and `sysmods` directory contain additions and modifications to system classes. Note that there is always a risk in making changes to system classes, so incorporate these changes at your own risk.

The source code and examples are licensed to only one person per copy of this book. That person is free to use the code and examples in their own development environment or applications as they see fit. However, the licensed person may not distribute the code or examples to other people.

Source files mentioned in chapters

This section shows which source files are referenced in which chapters.

16	Processes	<code>psq.st</code> , <code>process.st</code>
18	Cleaning up at Termination	<code>cleanup.st</code>
19	The Dependency Mechanism	<code>depend.st</code>
20	Error Handling	<code>errormsg.st</code>
21	Debugging	<code>debug.st</code> , <code>role.st</code>
22	Common Errors	<code>literal.st</code>

Copyright © 1997 by Alec Sharp

Download more free Smalltalk-Books at:

- The University of Berne: <http://www.iam.unibe.ch/~ducasse/WebPages/FreeBooks.html>

- European Smalltalk Users Group: <http://www.esug.org>

25	Hooks into the System	copydemo.st, framewrk.st, hooks.st
26	Changing Widgets at Runtime	focus.st, listdemo.st, widgets.st
29	Meta-Programming	metademo.st
30	Testing	testing.st
31	Customizing your Environment	launch20.st, launch25.st
33	Managing Source Code	custom.st, imcr20.st, imcr25.st install.st, install2.st, local.st

New classes and subclasses

These files can be found in the directory `classes`. Each file contains a class or classes for which the code is shown and discussed in the book. The files can be directly filed into the image.

cleanup.st	Class to close files, terminate processes, etc., at termination.
copydemo.st	Example of copying, both with and without using <code>postCopy</code> .
custom.st	Class to customize your development environment.
debug.st	Three classes: a class that logs to a file or to the Transcript; a class that intercepts messages being sent to an object; an example for using the inspector to change values.
depend.st	Examples illustrating the dependency mechanism.
errmsg.st	Example showing the use of error message classes and subclasses.
focus.st	Example of determining which widget has requested the focus.
framewrk.st	Extensions to the Extended Framework.
hooks.st	Examples of using various system hooks.
imcr20.st	Classes to automatically create deployment image for VisualWorks 2.0.
imcr25.st	Classes to automatically create deployment image for VisualWorks 2.0.
launch20.st	Subclass of <code>VisualLauncher</code> for VisualWorks 2.0.
launch25.st	Subclass of <code>VisualLauncher</code> for VisualWorks 2.5.
listdemo.st	Example of setting up code to intercept double-clicks and keyboard input.
literal.st	An example of modifying literal strings.
metademo.st	Example showing methods and variables in a class and its metaclass.
process.st	Subclass of <code>Process</code> that includes a name variable.
psq.st	Priority Shared Queue.
role.st	An example showing an object assuming different roles.
testing.st	Test Manager and Test case classes and subclasses.
widgets.st	Examples of modifying widgets and menus at runtime.

Files for installing applications

These files can be found in the directory `manage`. Each file illustrates some aspect of managing application or development environment code, and filing code into the image.

install.st	Installs modifications to system classes, files in application code, and customizes the environment.
install2.st	Shows how to discover the directory from which the file in is being done.
local.st	Local development environment changes.

System class additions

These files can be found in the directory `sysadds`. Each file adds methods to a system class. Use at your own risk.

behavior.st	Additions to Behavior to return all instances of a class and its subclasses, and to change instances into empty strings.
blockcl.st	Additions to BlockClosure to fork processes with names.
c-browse.st	Additions to class side of Browser for looking at all methods in a specified protocol.
c-para.st	Addition to class side of ParagraphEditor to return the keyboard.
char.st	Addition to Character to return character as a string.
chararr.st	Additions to CharacterArray to trim white space, to capitalize the first letter, and to return an array of substrings.
class.st	Additions to Class to copy a class and to create accessors when accepting a class definition.
inspect.st	Additions to Inspector for inspecting owners of and reference paths to an object.
keyproc.st	Addition to KeyboardProcessor to help track which widget has requested the focus.
object.st	Additions to Object which add functionality for halting, displaying objects, printing to the Transcript, and marking methods.
stream.st	Addition to Stream to read a line and strip comments.
sequence.st	Additions to SequenceableCollection to append an object, and to find the first occurrence of an object starting at a specified position.

System class modifications

These files can be found in the directory `sysmods`. Each file modifies existing methods in a system class. Use at your own risk.

behavior.st	Modification to Behavior to provide a different method template.
inspect.st	Modifications to the Inspector menu to allow you to browse owners of and reference paths to the object being inspected.
keyproc.st	Modification to KeyboardProcessor to track which widget has requested the focus. You must add a new instance variable by hand.
standard.st	Modification to StandardSystemController to allow the VisualWorks