

# User's Guide

Spread™

## Notices

Information in the documentation is subject to change without notice and does not represent a commitment on the part of FarPoint Technologies, Inc. The software described in this document is furnished under a license or nondisclosure agreement. The software may be used or copied only in accordance with the terms of the agreement. It is against the law to copy this software on any medium except as is specifically allowed in the license or nondisclosure agreement. No part of the documentation may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or information storage and retrieval systems, for any purpose without the express written permission of FarPoint Technologies, Inc.

© 1998, 1999, 2000 FarPoint Technologies, Inc. All rights reserved.

Unless otherwise noted, all names of companies, products, street addresses, and persons contained herein are part of a completely fictitious scenario or scenarios and are designed solely to document the use of a FarPoint Technologies, Inc., product.

Spread and List Pro are trademarks of FarPoint Technologies, Inc.

ActiveX, AppStudio, Excel, FoxPro, Microsoft, Visual Basic, Visual C++, Windows, Windows NT, and Word for Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Borland is a registered trademark and Resource Workshop is a trademark of Borland International, Inc. TrueType is a registered trademark of Apple Computer, Inc. Adobe, Acrobat, and FrameMaker are registered trademarks of Adobe Systems Incorporated. WebWorks Publisher is a registered trademark of Quadralay Corporation. Other brand and product names are trademarks or registered trademarks of their respective holders.

This guide was produced using Adobe FrameMaker® 6.0. The online help file was produced using WebWorks Publisher® 2000. The PDF version of this guide was produced using Adobe® Acrobat® 5.0.

3.k

## Distribution Restrictions

If you are using the trial or evaluation version of this product, then you may not distribute any of the files provided with the trial or evaluation version. The control DLL's, OCX's, and/or VBX's distributed with the retail product may be distributed by the user royalty free. Distribution by the user of any designers or design-time assistants (EXE's or DLL's), executables, source code, or license files (.LIC) distributed by FarPoint Technologies as part of this product is prohibited. You shall not develop applications that provide an application programmable interface to the Software or the Software as modified. Redistribution by your users of FarPoint Technologies' DLL's, OCX's, and/or VBX's or your modified or wrapped version of our OCX's without the appropriate redistribution license from FarPoint is prohibited.

# Summary Table of Contents

## Part I Introduction to Spread

*Introduces the Spread control.*

### **Chapter 1 Introducing Spread . . . . . 3**

*Provides an overview of the Spread control.*

### **Chapter 2 The fpSpread Control . . . . . 21**

*Provides descriptions of the fpSpread control features, including navigation and lists of events, methods, functions, messages, structures, and styles.*

### **Chapter 3 The fpSpreadPreview Control . . . . . 39**

*Provides descriptions of the fpSpreadPreview control features and lists of properties, events, methods, functions, and messages.*

### **Chapter 4 Using the Spread Designer . . . . . 49**

*Provides an overview of the Spread Designer.*

## Part II How-to Guides

*Describe tasks you can accomplish using the fpSpread and fpSpreadPreview controls and provide detailed, step-by-step instructions.*

### **Chapter 5 Working with Data . . . . . 75**

*Describes how to add, copy, move, swap, delete, and retrieve data and use formulas.*

### **Chapter 6 Working with Spreadsheets . . . . . 115**

*Describes how to set up and customize the fpSpread control, including saving spreadsheets to files, printing spreadsheets, using the control as a list box, and using virtual mode.*

<b>Chapter 7 Printing Spreadsheets</b> . . . . .	<b>139</b>
<i>Describes how to print the fpSpread control using a variety of features.</i>	
<b>Chapter 8 Previewing a Printed Spreadsheet</b> . . . . .	<b>153</b>
<i>Describes how to customize and use the fpSpreadPreview control to preview your printed spreadsheet.</i>	
<b>Chapter 9 Customizing Columns and Rows</b> . . . . .	<b>161</b>
<i>Describes how to work with columns and rows, including inserting and deleting them, sizing them, displaying them, and customizing headers.</i>	
<b>Chapter 10 Working with Cells</b> . . . . .	<b>189</b>
<i>Describes how to work with the active cell, select cells, work with blocks of cells, allow cells to overflow, and customize borders, colors, and fonts.</i>	
<b>Chapter 11 Setting the Cell Type.</b> . . . . .	<b>209</b>
<i>Describes how to set the cells to specific cell types and customize the cell with the available features for that cell type.</i>	
<b>Chapter 12 Binding to Databases</b> . . . . .	<b>269</b>
<i>Describes how to bind the fpSpread control to a database and customize data binding, including setting cell types and using virtual mode.</i>	
<b>Chapter 13 Customizing the Spreadsheet Display</b> . . . . .	<b>283</b>
<i>Describes how to change the spreadsheet display, including customizing grid lines, setting measurement units, and designing border appearance.</i>	
<b>Chapter 14 Customizing User Interaction</b> . . . . .	<b>291</b>
<i>Describes how to customize user interaction such as dragging and dropping operations, scroll bars, pointers, and navigation.</i>	

**Appendix A fpSpread Control Property Summary Table . 313**

*Lists all properties available for the fpSpread ActiveX and VBX controls, including a description of each.*

**Appendix B fpSpreadPreview Control Property Summary Table . . . . . 335**

*Lists all properties available for the fpSpreadPreview ActiveX and VBX controls, including a description of each.*

**Appendix C fpSpread and fpSpreadPreview DLL Function Summary . . . . . 339**

*Lists the functions available for the fpSpread and fpSpreadPreview DLL controls.*

**Appendix D fpSpread Control ActiveX/VBX to DLL Cross Reference . . . . . 353**

*Lists the fpSpread control's ActiveX and VBX control properties and functions and shows their corresponding DLL functions.*

**Appendix E fpSpread Control DLL to ActiveX/VBX Cross Reference . . . . . 369**

*Lists the fpSpread control's DLL functions and shows their corresponding ActiveX and VBX control properties and functions.*

**Appendix F Spread Designer Cross Reference . . . . . 385**

*Provides lists detailing the Spread Designer interface and its corresponding fpSpread control properties.*

**Appendix G Formula Operators and Functions . . . . . 429**

*Lists the available operators and functions for use in formulas.*

**Appendix H Using DLL Controls. . . . . 467**

*Describes use of the fpSpread and fpSpreadPreview DLL controls.*

## Summary Table of Contents

### **Appendix I Excel-Formatted File Import/Export . . . . .471**

*Provides detailed information about how the fpSpread control handles importing and exporting Excel-formatted files.*

### **Appendix J HTML File Export . . . . .505**

*Provides detailed information about how the fpSpread control handles exporting HTML files.*

# Table of Contents

<b>Preface</b> .....	<b>xv</b>
System Requirements .....	xv
Installation and Redistribution Information .....	xv
About the Documentation .....	xvi
Documentation Conventions.....	xvi
Using Online Help.....	xvii
FarPoint Technologies Technical Support .....	xvii
Contacting FarPoint Technologies, Inc. ....	xix
<b>Part I Introduction to Spread</b> .....	<b>1</b>
<b>Chapter 1 Introducing Spread</b> .....	<b>3</b>
Spread Features .....	3
New Spread Features.....	5
Information for Users of Previous Versions .....	8
<b>Chapter 2 The fpSpread Control</b> .....	<b>21</b>
Features .....	21
Concepts .....	30
Components of the fpSpread ActiveX and VBX Controls .....	34
Components of the fpSpread DLL Controls .....	36
<b>Chapter 3 The fpSpreadPreview Control</b> .....	<b>39</b>
Features .....	39
Components of the fpSpreadPreview ActiveX and VBX Controls .....	43
Components of the fpSpreadPreview DLL Controls .....	44
fpSpreadPreview Property Pages .....	45

<b>Chapter 4 Using the Spread Designer</b> . . . . .	<b>49</b>
Starting the Spread Designer . . . . .	49
Selecting Spreadsheet Elements . . . . .	51
Design Mode Versus Run-Time Mode . . . . .	52
Using Learn Mode . . . . .	53
Using Cell Type Mode . . . . .	55
Opening Multiple Spreadsheets . . . . .	57
Spread Designer View Versus Active View . . . . .	58
Printing the Active View . . . . .	58
Previewing the Printed Spreadsheet in Spread Designer . . . . .	65
Saving and Applying Changes to the Spreadsheet . . . . .	67
Customizing Balloon Help . . . . .	71
<b>Part II How-to Guides</b> . . . . .	<b>73</b>
<b>Chapter 5 Working with Data</b> . . . . .	<b>75</b>
Adding Data to a Spreadsheet . . . . .	75
Loading an Existing Spreadsheet . . . . .	81
Copying Data . . . . .	84
Moving Data . . . . .	91
Swapping Data . . . . .	97
Deleting Data . . . . .	98
Sorting Data . . . . .	100
Entering Formulas . . . . .	102
Returning Data from a Spreadsheet . . . . .	110
<b>Chapter 6 Working with Spreadsheets</b> . . . . .	<b>115</b>
Saving Spreadsheet Data . . . . .	115
Specifying Which Elements Users Can Select . . . . .	126
Resetting a Spreadsheet to its Default Settings . . . . .	127
Monitoring Data Changes in the Spreadsheet . . . . .	128

Specifying the Operation Mode .....	128
Using Virtual Mode .....	130
<b>Chapter 7 Printing Spreadsheets .....</b>	<b>139</b>
Features .....	139
Working in the Spread Designer .....	140
Sending a Spreadsheet to a Printer .....	141
Calculating the Number of Pages to Print .....	142
Specifying Print Options .....	142
Printing the Portion of the Spreadsheet with Data .....	146
Specifying a Print Range .....	147
Specifying Print Page Breaks .....	149
Adding Header and Footer Text to Printed Pages .....	151
<b>Chapter 8 Previewing a Printed Spreadsheet .....</b>	<b>153</b>
Previewing Your Spreadsheet .....	153
Customizing the Page View .....	154
Zooming In and Out .....	155
Displaying Scroll Bars .....	156
Changing the Appearance of the Preview Area .....	157
<b>Chapter 9 Customizing Columns and Rows .....</b>	<b>161</b>
Inserting Columns and Rows .....	161
Deleting Columns and Rows .....	164
Sizing Columns and Rows .....	166
Preventing Resizing of Columns and Rows with the Mouse .....	170
Setting the Maximum Number of Columns and Rows .....	172
Displaying a Specified Number of Columns and Rows .....	173
Designating the Leftmost Column to Display .....	174
Designating the Topmost Row to Display .....	174
Hiding Columns and Rows .....	175
Freezing Columns and Rows .....	177

## Table of Contents

Specifying Where the Last Column and Row Stop Scrolling . . . . .	177
Customizing Column Headers and Row Headers . . . . .	179
Setting Odd and Even Row Colors. . . . .	186
<b>Chapter 10 Working with Cells . . . . .</b>	<b>189</b>
Working with the Active Cell. . . . .	189
Working with a Range of Cells . . . . .	190
Selecting Multiple Blocks of Cells. . . . .	191
Locking Cells . . . . .	192
Resizing Cells. . . . .	196
Allowing Cells to Overflow . . . . .	198
Using Edit Mode. . . . .	199
Customizing Cell Borders . . . . .	202
Setting Cell Background and Text Colors . . . . .	205
Setting the Font. . . . .	207
Specifying the Background and Text Color of Selected Cells. . . . .	207
<b>Chapter 11 Setting the Cell Type. . . . .</b>	<b>209</b>
Creating Button Cells . . . . .	209
Creating Check Box Cells . . . . .	219
Creating Combo Box Cells. . . . .	225
Creating Date Cells. . . . .	233
Creating Edit Cells . . . . .	238
Creating Float Cells . . . . .	242
Creating Integer Cells. . . . .	249
Creating Owner-Drawn Cells. . . . .	252
Creating PIC Cells . . . . .	253
Creating Picture Cells. . . . .	255
Creating Static Text Cells. . . . .	258
Creating Time Cells . . . . .	263

<b>Chapter 12 Binding to Databases . . . . .</b>	<b>269</b>
Connecting to a Database Using Default Settings . . . . .	269
Displaying Selected Fields in Spreadsheet Columns . . . . .	273
Customizing Headers for Bound Spreadsheets . . . . .	275
Sizing Columns in Bound Spreadsheets . . . . .	276
Assigning Cell Types for Bound Spreadsheets . . . . .	278
Specifying Spreadsheet and Database Interaction . . . . .	280
Using Virtual Mode with Bound Spreadsheets . . . . .	282
<b>Chapter 13 Customizing the Spreadsheet Display . . . . .</b>	<b>283</b>
Customizing Grid Lines . . . . .	283
Setting Spreadsheet Measurement Units . . . . .	285
Preventing the Spreadsheet from Flickering . . . . .	286
Setting the Gray Area Color . . . . .	287
Working with the Control and Spreadsheet Borders . . . . .	288
Hiding Selections When the Spreadsheet Does Not Have the Focus . .	290
<b>Chapter 14 Customizing User Interaction . . . . .</b>	<b>291</b>
Allowing Users to Drag and Drop Cells . . . . .	291
Displaying Scroll Bars . . . . .	293
Specifying the Pointer . . . . .	296
Turning Off the Spreadsheet Beep . . . . .	297
Customizing Navigation . . . . .	298
Using Arrow Keys to Move Between Cells . . . . .	301
Turning Off the Clipboard Shortcut Keys . . . . .	301
Changing or Turning Off the F2, F3, and F4 Action Keys . . . . .	302
Specifying When Buttons Appear . . . . .	305
Providing Text Tips . . . . .	306
<b>Appendix A fpSpread Control Property Summary Table .</b>	<b>313</b>

<b>Appendix B fpSpreadPreview Control Property Summary Table</b> .....	<b>335</b>
<b>Appendix C fpSpread and fpSpreadPreview DLL Function Summary</b> .....	<b>339</b>
fpSpread Control DLL Function Summary .....	339
fpSpreadPreview Control DLL Function Summary .....	350
<b>Appendix D fpSpread Control ActiveX/VBX to DLL Cross Reference</b> .....	<b>353</b>
ActiveX/VBX Properties to DLL Functions .....	353
ActiveX/VBX Functions to DLL Functions .....	364
<b>Appendix E fpSpread Control DLL to ActiveX/VBX Cross Reference</b> .....	<b>369</b>
<b>Appendix F Spread Designer Cross Reference</b> .....	<b>385</b>
Menus and Menu Items .....	385
Toolbars and Buttons .....	405
Properties Cross-Referenced to Buttons and Menus .....	410
<b>Appendix G Formula Operators and Functions</b> .....	<b>429</b>
Formula Operators .....	429
Formula Functions .....	429
<b>Appendix H Using DLL Controls</b> .....	<b>467</b>
Working with Libraries .....	467
Creating the fpSpread Control or fpSpread Preview Control .....	468
Using Get and Set Functions .....	469
Working with String Data .....	469

**Appendix I Excel-Formatted File Import/Export . . . . . 471**  
    Excel-Formatted File Import. . . . . 471  
    Excel-Formatted File Export. . . . . 484  
    Log File Messages. . . . . 497

**Appendix J HTML File Export . . . . . 505**  
    File Export Information. . . . . 505  
    Log File Information. . . . . 516

**Index . . . . . 521**



# Preface

The following sections describe the Spread™ product and the material provided with it.

## System Requirements

To run Spread, you need a 386 system or higher; 8MB of memory; a VGA card with support for 256 colors or better; CD-ROM (provided) or 3 1/2 inch, 1.44 MB floppy disk for install; and 20 MB of hard disk space (full install).

Depending on which control you use, you also need to use the following operating system and development environment.

<b>Control</b>	<b>Required Operating System and Development Environment</b>
16-bit DLL	Microsoft Windows 3.1, Windows 95, Windows NT, or Windows 98 Microsoft Visual C++ 1.5x
32-bit DLL	Microsoft Windows 95, Windows NT, Windows 98, Windows 2000, or Windows Millennium Edition Microsoft Visual C++ 4.1 or higher
VBX	Microsoft Windows 3.1, Windows 95, Windows NT, or Windows 98 Microsoft Visual Basic 3.0 or 4.0 16-bit or Microsoft Visual C++ 1.5x
DAO ActiveX	Microsoft Windows 95 with OLE Automation Update, Windows NT SP3 or higher, Windows 98, Windows 2000, or Windows Millennium Edition Microsoft Visual Basic 4.0 32-bit or higher or Microsoft Visual C++ 4.1 or higher
ADO ActiveX	Microsoft Windows 95 with OLE Automation Update, Windows NT SP3 or higher, Windows 98, Windows 2000, or Windows Millennium Edition Microsoft Visual Basic 6.0 or higher or Microsoft Visual C++ 6.0 or higher

## Installation and Redistribution Information

Installation instructions and information are provided in the SPREADME.HLP file that accompanies this product. Before you run the installation program, you might want to make a backup copy of each of the Spread distribution disks, if you received Spread on 3.5-inch disks.

Information about the files necessary for redistributing the Spread control is also provided in the SPREADME.HLP file that accompanies this product.

## About the Documentation

Throughout the Spread documentation, it is assumed that you are familiar with your development environment. The intended audience is users of the Spread control, including interface designers and developers.

The documentation provided for Spread is intended to describe the features of the fpSpread control and how to use them. Reference material describing the available properties, events, methods, functions, messages, structures, and styles is available in the Spread online help file. For information about accessing online help, see “Using Online Help” on page xvii.

## Documentation Conventions

Spread documentation uses the following conventions:

<b>Example</b>	<b>Description</b>
Illustrations	Illustrations of the fpSpread control and other screen objects were created in the Windows 95™ operating environment and the Visual Basic™ version 5.0 development environment unless otherwise noted.
	This symbol indicates the beginning of instructions for performing a task.
	This symbol indicates the beginning of an example in the documentation.
	<b>Note</b> Pixel values in examples are calculated for a 640×480 screen resolution.
	The line continuation character (↪) indicates that code continued from one line to the next in the documentation should be typed all on one line in the code window.
Spread Designer	Text indicates the condition for the subsequent instructions or explanation. For example, the words “Spread Designer” in the left margin indicates that the subsequent information pertains to using the Spread Designer. Some possible conditions include version of the controls, development environment, and run-time versus design-time interaction.
<b>CellBorderType</b>	Names of properties, events, and functions appear in bold with initial letters capitalized.
A: \SETUP	Words you need to type appear in this font, as do code examples.
MYFILE.TXT	Words in all capital letters indicate file and path names.
<i>value, color</i>	Italicized items in programming syntax are placeholders for information you supply.

Example	Description
<i>short</i> , <b>BOOL</b>	In programming syntax, bold, italicized arguments indicate data types, pointers, and user-defined types.
[ = <i>value</i> & ]	In programming syntax, items inside square brackets are optional.
fpSpread control	When speaking generically about the spreadsheet control, refers to both the DAO vaSpread control and the ADO fpSpread control

## Using Online Help

This product comes with extensive online help. The online help provides an online version of this *User's Guide*. It also provides online reference guides that describe the properties, events, functions, messages, structures, and styles provided for the controls.

While using the ActiveX or VBX versions of the controls, you can access the online help by pressing F1 while the focus is on the control or a property within the Visual Basic Properties window. While using the ActiveX versions of the fpSpreadPreview control, you can also choose the Help button on the property page to access online help. An icon for the online help also appears within the Spread Windows program group or folder.

## FarPoint Technologies Technical Support

If you have a question about Spread, consult the following sources:

- The online help and other documentation files installed with the product.  
For instructions for accessing the online help, see “Using Online Help” on page xvii.
- The code samples available on the product CD (in the \SPREAD30\SAMPLES directory).

If you cannot find the answer using these sources, please contact our Technical Support department using the methods provided for your support package. For information about technical support packages, see the following section. For contact information, see “Contacting FarPoint Technical Support” on page xix.

## Technical Support Packages

FarPoint Technologies offers a variety of technical support packages that provide access to our responsive technical support team. For more information on these support packages, visit our web site ([www.fpoin.com](http://www.fpoin.com)), or contact our Sales department at (800) 645-5913 or [fpsales@fpoin.com](mailto:fpsales@fpoin.com).

## Preface

### Trial Support

Trial support is available for a period of 30 days to anyone who downloads a product trial version. With trial support, you can consult the online help file and code samples provided with the trial version or search our knowledge base and read white papers available through our web site. You can also contact our technical support team via e-mail or by using the support web site to have a request answered by our support team.

### Standard Support

Standard support is available at no additional cost to all registered users of our products. You receive 30 days of free phone support beginning with your first phone call to our support team. You also have unlimited use of our support web site, where you can submit, view, and update any support issues. Product updates are also available for you to download at any time.

### Per-Incident Support

Incident support is available to registered users whose 30-day, free phone support has expired for a particular product and who wish to speak directly with a support technician. Incident support must be purchased from the FarPoint Sales department before calling our support team. Contact our Sales department for price information. (Quantity discount packages are available.)

---

**Note** There will not be a charge if the incident reported is due to a verifiable bug in FarPoint's software.

---

### Priority Support

Priority support contracts can be purchased by any registered user. Priority support contracts are valid for one year from the date of purchase, and are purchased for a specific product. With Priority support you receive the following services:

- You are entitled to unlimited technical support phone calls for the specified product during your contract period.
- When you submit a support request for the specified product using the support web site, your message is marked as priority and automatically moved ahead of all other non-priority requests.
- You receive automatic e-mail notification when a new maintenance release is posted for the specified product.
- You are one of the first to be notified when a product beta becomes available for the specified product.

## Premium Support

Premium support can be purchased by any registered user. Premium support contracts are valid for one year from the date of purchase, and are purchased for particular products. With Premium support you receive the following services:

- All the services provided for Priority Support.
- You are automatically notified and receive any new product versions or maintenance releases for the specified product.

## Contacting FarPoint Technical Support

If you need to contact FarPoint Technical support concerning a question or a problem with Spread, please use one of the following methods depending on your technical support package:

<b>Support Web site</b>	<a href="http://www.fpoint.com">http://www.fpoint.com</a>
<b>E-mail</b>	<a href="mailto:fpsupport@fpoint.com">fpsupport@fpoint.com</a>
<b>Fax</b>	(919) 460-7606
<b>FTP site</b>	<a href="ftp://ftp.fpoint.com">ftp://ftp.fpoint.com</a>
<b>Phone</b>	(919) 460-1887

FarPoint Technologies' Technical Support department is available between the hours of 9:00 A.M. and 5:30 P.M. eastern time, Monday through Friday

See "Technical Support Packages" on page xvii for more information about packages.

## Contacting FarPoint Technologies, Inc.

If you would like to find out more about FarPoint Technologies and our products, contact our Sales department using one of the following methods:

<b>World Wide Web site</b>	<a href="http://www.fpoint.com">http://www.fpoint.com</a>
<b>E-mail</b>	<a href="mailto:fpsales@fpoint.com">fpsales@fpoint.com</a>
<b>Phone</b>	(800) 645-5913

## Preface

**Fax** (919) 460-7606  
**Mail** FarPoint Technologies, Inc.  
175 Southport Drive  
Suite 900  
Morrisville, NC 27560

# Part I Introduction to Spread

Part I introduces Spread by providing an overview of the fpSpread and fpSpreadPreview controls, a more in-depth discussion of each control's features, and an overview of the Spread Designer provided for setting properties.

If you have used FarPoint Technologies' Spread control in the past, you might want to go on to the instructional information provided in Part II, the how-to guides, after reviewing the information in "Information for Users of Previous Versions" on page 8.

If you are a new user of this product, you should review most of the contents of Part I, particularly Chapter 2, "The fpSpread Control," and Chapter 3, "The fpSpreadPreview Control," which provide an overview of the Spread controls.



# Chapter 1 Introducing Spread

Spread offers a versatile and easy-to-use spreadsheet control for your interface. The Spread package offers DLL, VBX, and ActiveX control types so you can easily work in almost any development environment.

Spread provides a spreadsheet control that can be configured to work as a fully functional spreadsheet, but can also work as a table or a data-entry control connected to a database. Spread also provides a print preview control that you can use to view your spreadsheet before printing.

This chapter introduces the fpSpread and fpSpreadPreview controls and reviews changes from previous versions. You might want to look over this chapter, and, particularly if you have used previous versions of Spread, be sure to review “Information for Users of Previous Versions” on page 8.

## Spread Features

The Spread controls offer unique capabilities that are described throughout the book and the online help. The following sections describe some of Spread’s most often-used features.

- **Create formulas that use our Calc Engine**

Spread comes with a calculation engine, or Calc Engine, that can implement over 70 mathematical, logical, and statistical functions. For a complete list of operators and predefined functions, see Appendix G, “Formula Operators and Functions.”

For more information about using the Calc Engine, see “Entering Formulas” on page 102.

- **Create custom spreadsheets quickly using Spread Designer**

The Spread Designer lets you create customized spreadsheets that fit your application’s needs. You can open multiple existing spreadsheets within the Spread Designer and apply characteristics (except data-binding characteristics) from any of them to the current spreadsheet you are designing. You can generate documented Visual Basic code for properties set using the Spread Designer. The Spread Designer shortens your development time and offers quick access to the multiple facets of our powerful fpSpread control.

For more information, see Chapter 4, “Using the Spread Designer.”

- **Share files between controls**

Spreadsheet file formats are consistent across all fpSpread control types. For example, a spreadsheet file created for a DLL control can be loaded and used in an ActiveX or VBX control.

- **Bind to databases**

Using Visual Basic, you can bind fpSpread ActiveX and VBX controls to databases. Spread contains two ActiveX controls. The FPSPR30.OCX control supports the ActiveX Data Object for use in Visual Basic 6.0 and Visual C++ 6.0. The SPR32X30.OCX control supports the Data Access Object for use in earlier versions of Visual Basic and Visual C++.

For more information, see Chapter 12, “Binding to Databases.”

- **Optimize performance with virtual mode**

If you have a large amount of data, you can use virtual mode to increase the responsiveness of the fpSpread control. You can customize the virtual buffer where rows are stored, and you can display special scroll bars that let you move up and down within the virtual buffer.

For more information, see “Using Virtual Mode” on page 130.

- **Assign cells one of twelve cell types**

The cell types you can include in the fpSpread control are: button, check box, combo box, date, edit, float, integer, owner-drawn, PIC (or mask), picture, text, and time.

For more information, see Chapter 11, “Setting the Cell Type.”

- **Have your spreadsheet operate as a single-, multiple-, or extended-selection list box**

The spreadsheet can operate similarly to a list box. When the user clicks a row, that row is selected. You can choose whether the user can select one and only one row (single-selection) or whether the user can select additional rows either by clicking them singly (multiple-selection) or by using the Ctrl or Shift key to select a range (extended-selection).

For more information, see “Specifying the Operation Mode” on page 128.

## New Spread Features

Take advantage of these new features in Spread 3.0:

- **Import files and export files in Microsoft Excel™ format**

You can import a file created in Excel 97 (BIFF8 format), and you can export a spreadsheet created with the fpSpread control to an .XLS file.

For more information, see “Loading an Existing Spreadsheet” on page 81 and “Exporting a Spreadsheet to an Excel-Formatted File” on page 117. For information on how spreadsheet data and property settings are translated into the Excel format and vice versa, see Appendix I, “Excel-Formatted File Import/Export.”

- **Export spreadsheets as HTML files**

You can export a spreadsheet created with the fpSpread control to an HTML (version 3.2 or later) file.

For more information, see “Exporting a Spreadsheet to an HTML File” on page 119. For information on how spreadsheet data and property settings are translated into the HTML format, see Appendix J, “HTML File Export.”

- **Use the new fpSpreadPreview control to preview the spreadsheet**

When you preview your spreadsheet, you can allow the user to zoom in and out to view specific areas, change the number of pages on the screen, and change the border, gray area, and shadow appearance. You can also add scroll bars to the preview.

For more information, see Chapter 8, “Previewing a Printed Spreadsheet.”

- **Support OLE Drag and Drop**

Spread supports manual drag-and-drop operations. For more information on OLE Drag and Drop, refer to the online help.

- **Load data from or save data to an array**

For more information, see “Providing the Data” on page 76 and “Saving Data to an Array” on page 126.

- **Enhance a combo box cell using the FarPoint List Pro fpCombo control**

You can integrate the drop-down list portion of the fpCombo control and the fpSpread combo box cell.

For more information, see “Creating a Combo Box Cell Using the FarPoint fpCombo Control” on page 230.

- **Provide text tips**

The fpSpread control provides text tips that can be displayed for any header or cell in the control.

For more information, see “Providing Text Tips” on page 306.

- **Change or turn off the F2, F3, or F4 action keys**

The F2 (clear, for all cell types), F3 (display current date or time, for date and time cells), and F4 (display current calendar, for date cells) keys perform specific actions in Spread. You can turn these keys off or you can change the key sequence to perform the action.

For more information, see “Changing or Turning Off the F2, F3, and F4 Action Keys” on page 302.

- **Set more than one border color and style for the same cell**

You can display a border on the top, bottom, right side, and left side of a cell. You can specify more than one style and color for the same cell. For example, the bottom border could be red and solid, and the top border could be green and dotted.

For more information, see “Customizing Cell Borders” on page 202.

- **Respond to events from combo boxes, such as dropping down or closing the combo box list**

Events now fire when the combo box drops down, closes up, and when the user makes another selection.

For more information, see the event reference in the online help.

- **Provide different colors for alternating rows**

You can specify one color for odd rows and a different color for even rows.

For more information, see “Setting Odd and Even Row Colors” on page 186.

- **Take advantage of improved printing features**

You can specify the page number that prints on the first page and the order in which pages print. You can calculate the number of pages that will print based on the current print settings. You can also determine the next column or row at which a page break occurs.

For more information, see “Specifying Print Options” on page 142, “Calculating the Number of Pages to Print” on page 142, and “Specifying Print Page Breaks” on page 149.

- **Define a custom mouse pointer for drag-and-drop operations**

You can define a cursor for use when the mouse is over an area that allows drag-and-drop operations and a cursor for use when a drag-and-drop operation is being performed.

For more information, see “Specifying the Pointer” on page 296.

- **Make the scroll bar track mouse movement through the spreadsheet**

You can specify that the columns and rows move in synchronization with the horizontal and vertical scroll box movement.

For more information, see “Synchronizing the Scroll Box with Columns and Rows” on page 295.

- **Align the cell contents vertically in all cell types except button cells**

For all cell types, except the button cell, you can align the contents vertically at either the top, bottom, or center of the cell.

For more information, see the information for the specific cell type in Chapter 11, “Setting the Cell Type.”

- **Specify a different background and text color for selected cells**

Instead of inverting the default color, you can specify the background and text color for cells that the user selects.

For more information, see “Specifying the Background and Text Color of Selected Cells” on page 207.

- **Choose whether or not to include headers when copying data to the Clipboard**

You can specify whether or not to include row headers when data is copied from or pasted to the Clipboard.

For more information, see “Copying Data Using the Clipboard” on page 88.

- **Sort with up to 256 sort keys**

You can now use up to 256 sort keys when sorting spreadsheet data.

For more information, see “Sorting Data” on page 100.

- **Automatically create a Windows 3.1 or Windows 95 appearance for the control**

The fpSpread control can display no border, a flat border, a three-dimensional appearance, or a three-dimensional appearance with an outline border.

For more information, see “Changing the Three-Dimensional Appearance” on page 288.

- **For date cells, display an enhanced calendar**

The pop-up calendar for date cells now displays a Windows 95 appearance. You can also now specify short and long day and month names for the pop-up calendar.

For more information, see “Customizing Day and Month Names for the Pop-up Calendar” on page 237.

## Information for Users of Previous Versions

If you have used previous versions of Spread, or the DLL Spreadsheet control provided in FarPoint’s ToolBox product, you should read the following sections to familiarize yourself with the updates available in Spread version 3.0. For additional information and instructions for upgrading your existing projects, consult the ReadMe help file that accompanies this product (SPREADME.HLP).

Several properties and functions have been changed, and some properties are no longer supported.

## New File Name Extension for Binary and Tab-Delimited Files

All Spread version 3.0 binary files are now saved with the .SS3 extension. Spread version 3.0 can open and read .SS2 files that were created using previous versions of Spread.

Tab-delimited files are now saved with a .TB3 extension. Spread version 3.0 can open and read .TB2 files that were created using previous versions of Spread.

## DLL Users

### Changes to Functions

The **SSGetBool** and **SSSetBool** functions have two new booleans.

<b>Boolean</b>	<b>Description</b>
SSB_HSCROLLBARTRACK	Columns move as you drag the scroll box on the horizontal scroll bar
SSB_VSCROLLBARTRACK	Rows move as you drag the scroll box on the vertical scroll bar

The **SSGetCursor** and **SSSetCursor** functions *wType* parameter supports the following new values.

<b>Value</b>	<b>Description</b>
SS_CURSORTYPE_DRAGDROPAREA	Displays specified pointer on the edge of the selection to drag and drop when the pointer passes over it
SS_CURSORTYPE_DRAGDROP	Displays specified pointer while drag-and-drop operation is being performed

## New Functions

The following table lists the new functions that have been added to the fpSpread control for Spread version 3.0.

<b>Function</b>	<b>Description</b>
<b>SSEnumCustomFunction</b>	Queries the list of custom functions
<b>SSEnumCustomFunctionLen</b>	Returns the size of the string containing the custom function
<b>SSExportRangeToHTML</b>	Exports a range of cells to an .HTM file
<b>SSExportToExcel</b>	Exports a spreadsheet to an .XLS file
<b>SSExportToHTML</b>	Exports a cell, column, row, or spreadsheet to an .HTM file
<b>SSGetActionKey</b>	Retrieves the shortcut key or navigation key assigned to a specific action
<b>SSGetAppearance</b>	Retrieves the predefined border style for the control
<b>SSGetArray</b>	Retrieves data from the spreadsheet and places it in an array
<b>SSGetBorderEx</b>	Specifies more than one border color and style in a cell
<b>SSGetClipboardOptions</b>	Retrieves whether headers are included when data is copied to or pasted from the Clipboard
<b>SSGetCustomFunction</b>	Retrieves information on custom functions
<b>SSGetExcelSheetList</b>	Retrieves list of Excel worksheet names
<b>SSGetNextPageBreakCol</b>	Retrieves the column number at which a page break occurs while printing
<b>SSGetNextPageBreakRow</b>	Retrieves the row number at which a page break occurs while printing
<b>SSGetOddEvenRowColor</b>	Retrieves the background and text colors for odd and even rows
<b>SSGetPrintOptions</b>	Retrieves the current print settings
<b>SSGetPrintPageCount</b>	Retrieves the number of pages based on the current print settings
<b>SSGetSelColor</b>	Retrieves the background and text color of the selected cells

<b>Function</b>	<b>Description</b>
<b>SSGetTextTip</b>	Retrieves whether text tips are displayed, when text tips are displayed, how text tips appear, how text tips are positioned, and how long the control waits before displaying the text tip
<b>SSGetTwoDigitYearMax</b>	Retrieves the year value used to calculate the century for two-digit year values
<b>SSImportExcelSheet</b>	Imports an Excel 97 worksheet
<b>SSIsColHidden</b>	Retrieves whether a specific column is hidden
<b>SSIsExcelFile</b>	Determines whether an Excel file is in BIFF8 format
<b>SSIsRowHidden</b>	Retrieves whether a specific row is hidden
<b>SSReCalcCell</b>	Recalculates the formula in an individual cell
<b>SSRemoveCustomFunction</b>	Removes a custom function
<b>SSSetActionKey</b>	Specifies the shortcut key or navigation key assigned to a specific action
<b>SSSetAppearance</b>	Specifies a predefined border style for the control
<b>SSSetArray</b>	Loads data into a spreadsheet from an array
<b>SSSetCalText</b>	Specifies the text for the pop-up calendar in a date cell
<b>SSSetClipboardOptions</b>	Specifies whether headers are included when data is copied to or pasted from the Clipboard
<b>SSSetOddEvenRowColor</b>	Specifies the background and text colors for odd and even rows
<b>SSSetPrintOptions</b>	Presets the SS_PRINTFORMAT structure
<b>SSSetSelColor</b>	Specifies the background and text color of selected cells
<b>SSSetTextTip</b>	Specifies whether text tips are displayed, when text tips are displayed, how text tips appear, how text tips are positioned, and how long the control waits before displaying the text tip
<b>SSSetTwoDigitYearMax</b>	Specifies the year value used to calculate the century for two-digit year values
<b>SSSetTypeComboBoxEx</b>	Sets the style, list items, drop-down list display, edit field maximum characters, and optional control to display for the drop-down list for combo box cells
<b>SSSortEx</b>	Sort spreadsheet data using up to 256 sort keys

The following table lists the new functions that have been added for the fpSpreadPreview control for Spread version 3.0.

<b>Function</b>	<b>Description</b>
<b>SpvGetAllowUserZoom</b>	Retrieves whether the user can use the mouse to zoom in or out while viewing pages
<b>SpvGetGrayAreaColor</b>	Retrieves the color of the preview area of the control
<b>SpvGetGrayAreaMarginH</b>	Retrieves the horizontal space between the displayed page and the edge of the control
<b>SpvGetGrayAreaMarginType</b>	Retrieves whether the control uses the actual gray area margin settings or scales the settings to optimize the page view
<b>SpvGetGrayAreaMarginV</b>	Retrieves the vertical space between the displayed page and the edge of the control
<b>SpvGethWndSpread</b>	Retrieves the window handle of the spreadsheet you want to preview
<b>SpvGetPageBorderColor</b>	Retrieves the border color of the displayed pages
<b>SpvGetPageBorderWidth</b>	Retrieves the border width of the displayed pages
<b>SpvGetPageCurrent</b>	Retrieves the page to display in the control
<b>SpvGetPageGutterH</b>	Retrieves the horizontal space between displayed pages
<b>SpvGetPageGutterV</b>	Retrieves the vertical space between displayed pages
<b>SpvGetPageMultiCntH</b>	Retrieves the number of pages displayed in the horizontal direction
<b>SpvGetPageMultiCntV</b>	Retrieves the number of pages displayed in the vertical direction
<b>SpvGetPagePercentageActual</b>	Retrieves the size of the displayed pages as a percentage of the actual size
<b>SpvGetPageShadowColor</b>	Retrieves the color of the shadow behind the displayed pages
<b>SpvGetPageShdaowWidth</b>	Retrieves the width of the shadow behind the displayed pages
<b>SpvGetPagesPerScreen</b>	Retrieves the number of pages displayed at one time
<b>SpvGetPageViewPercentage</b>	Retrieves the percent size of the displayed pages
<b>SpvGetPageViewType</b>	Retrieves how pages are displayed
<b>SpvGetScrollBarH</b>	Retrieves whether and how the horizontal scroll bars are displayed
<b>SpvGetScrollBarV</b>	Retrieves whether and how the vertical scroll bars are displayed

<b>Function</b>	<b>Description</b>
<b>SpvGetScrollIncH</b>	Retrieves the horizontal distance the control scrolls when the user scrolls through the displayed pages
<b>SpvGetScrollIncV</b>	Retrieves the vertical distance the control scrolls when the user scrolls through the displayed pages
<b>SpvGetZoomState</b>	Retrieves the way the control zooms between different views of the displayed pages
<b>SpvSetAllowUserZoom</b>	Specifies whether the user can use the mouse to zoom in or out while viewing pages
<b>SpvSetGrayAreaColor</b>	Specifies the color of the preview area of the control
<b>SpvSetGrayAreaMarginH</b>	Specifies the horizontal space between the displayed page and the edge of the control
<b>SpvSetGrayAreaMarginType</b>	Specifies whether the control uses the actual gray area margin settings or scales the settings to optimize the page view
<b>SpvSetGrayAreaMarginV</b>	Specifies the vertical space between the displayed page and the edge of the control
<b>SpvSetHWndSpread</b>	Specifies the window handle of the spreadsheet you want to preview
<b>SpvSetPageBorderColor</b>	Specifies the border color of the displayed pages
<b>SpvSetPageBorderWidth</b>	Specifies the border width of the displayed pages
<b>SpvSetPageCurrent</b>	Specifies the page to display in the control
<b>SpvSetPageGutterH</b>	Specifies the horizontal space between displayed pages
<b>SpvSetPageGutterV</b>	Specifies the vertical space between displayed pages
<b>SpvSetPageMultiCntH</b>	Specifies the number of pages displayed in the horizontal direction
<b>SpvSetPageMultiCntV</b>	Specifies the number of pages displayed in the vertical direction
<b>SpvSetPageShadowColor</b>	Specifies the color of the shadow behind the displayed pages
<b>SpvSetPageShadowWidth</b>	Specifies the width of the shadow behind the displayed pages
<b>SpvSetPageViewPercentage</b>	Specifies the percent size of the displayed pages
<b>SpvSetPageViewType</b>	Specifies how pages are displayed
<b>SpvSetScrollBarH</b>	Specifies whether and how the horizontal scroll bars are displayed
<b>SpvSetScrollBarV</b>	Specifies whether and how the vertical scroll bars are displayed

Function	Description
<b>SpvSetScrollIncH</b>	Specifies the horizontal distance the control scrolls when the user scrolls through the displayed pages
<b>SpvSetScrollIncV</b>	Specifies the vertical distance the control scrolls when the user scrolls through the displayed pages
<b>SpvSetZoomState</b>	Specifies the way the control zooms between different views of the displayed pages

## Changes to Structures

The following items have been added to the **SS\_PRINTFORMAT** structure:

- *fSmartPrint* field
- **SS\_PRINTTYPE\_OVERTHENDOWN** and **SS\_PRINTTYPE\_DOWNTHENOVER** defines for the *dPrintType* field

The **LPSS\_PRINTINFO** *lpPrintInfo* parameter has been added to the **SS\_PRINTFUNC** callback function.

## New Structures

The following new structures have been added for the fpSpread control for Spread version 3.0.

Structure	Description
<b>SS_COMBOCLOSEUP</b>	Provides information about the item selected when the combo box list closes
<b>SS_SORTKEY</b>	Provides information about the sort order
<b>SS_TEXTTIPFETCH</b>	Provides information about the text tip
<b>SS_PRINTINFO</b>	Provides information for the print preview control

## New Messages

The following new messages have been added for the fpSpread control for Spread version 3.0.

Message	Description
<b>SSM_EDITCHANGE</b>	Sent when the user is in edit mode and the text changes
<b>SSM_COMBOCLOSEUP</b>	Sent when the list portion of the combo box closes

Message	Description
<b>SSM_COMBODROPDOWN</b>	Sent when the list portion of the combo box drops down
<b>SSM_TEXTTIPFETCH</b>	Sent when the control displays a text tip

The following new messages have been added for the fpSpreadPreview control for Spread version 3.0.

Message	Description
<b>SPVN_PAGECHANGE</b>	Sent when the page changes
<b>SPVN_ZOOM</b>	Sent when the user zooms in or out on a page

### New Alignment Styles for Cell Types

In Spread version 3.0, the following new styles have been added for the fpSpread control for all cell types except button and owner-drawn.

The following vertical alignment styles are available:

Define Value	Numeric Value	Description
<b>SSS_ALIGN_BOTTOM</b>	0x000040000L	Bottom aligns data in the cell
<b>SSS_ALIGN_TOP</b>	0x000010000L	Top aligns data in the cell
<b>SSS_ALIGN_VCENTER</b>	0x000020000L	Vertically centers the data in the cell

The following horizontal alignment styles are available:

Define Value	Numeric Value	Description
<b>SSS_ALIGN_CENTER</b>	0x000400000L	Horizontally centers the data in the cell
<b>SSS_ALIGN_LEFT</b>	0x000100000L	Left aligns data in the cell
<b>SSS_ALIGN_RIGHT</b>	0x000200000L	Right aligns data in the cell

## ActiveX and VBX Users

### Obsolete Properties

The following properties are no longer supported. Please review your existing projects and update them according to the options described in the following table.

Property	Instead use . . .
<b>AllowResize</b>	<b>UserResizeCol</b> and <b>UserResizeRow</b> properties
<b>AllowSelBlock</b>	<b>SelectBlockOptions</b> property

Property	Instead use . . .
<b>CalcDependencies</b>	No replacement. Spread refreshes the dependencies tables automatically as needed.
<b>DataConnect</b>	No replacement. The Q+E Database Library is no longer supported.
<b>DataSelect</b>	No replacement. The Q+E Database Library is no longer supported.
<b>GrayAreaForeColor</b>	No replacement
<b>GridLines</b>	<b>GridShowHoriz</b> and <b>GridShowVert</b> properties
<b>LoadTabFile</b>	<b>LoadTabFile</b> function
<b>TypeEditAlign</b>	<b>TypeHAlign</b> property
<b>TypeEditCase</b>	<b>TypeEditCharCase</b> property
<b>TypeEditLen</b>	<b>TypeMaxEditLen</b> property
<b>TypeTextAlignHoriz</b>	<b>TypeHAlign</b> property

## Changes to Properties

The **TypeEditLen** property has been renamed and is now the **TypeMaxEditLen** property. The **TypeMaxEditLen** property is also available for the combo box cell.

The **CursorType** property has two new settings:

Setting	Description
SS_CURSOR_TYPE_DRAGDROPAREA	Displays pointer on the edge of the selection to drag and drop when the pointer passes over it
SS_CURSOR_TYPE_DRAGDROP	Displays pointer while drag-and-drop operation is being performed

## New Properties

The following table lists the new properties that have been added to the fpSpread control for Spread version 3.0.

Property	Description
<b>Appearance</b>	Sets or returns predefined border styles
<b>ClipboardOptions</b>	Sets or returns whether headers are included when data is copied to or pasted from the Clipboard
<b>OLEDropMode</b> (ActiveX only)	Sets or returns whether the control can act as an OLE drop object

<b>Property</b>	<b>Description</b>
<b>PrintFirstPageNumber</b> (ActiveX only)	Sets or returns the number to use to begin numbering the pages in the spreadsheet
<b>PrintNextPageBreakCol</b> (ActiveX only)	Returns the column number of the next column in the spreadsheet at which a page break occurs
<b>PrintNextPageBreakRow</b> (ActiveX only)	Returns the row number of the next row in the spreadsheet at which a page break occurs
<b>PrintPageCount</b> (ActiveX only)	Returns the number of pages required for printing the spreadsheet with the current print settings
<b>PrintPageOrder</b> (ActiveX only)	Sets or returns the order in which pages are printed
<b>PrintSmartPrint</b> (ActiveX only)	Sets the spreadsheet to print using Smart Printing
<b>ScrollBarTrack</b>	Sets or returns whether the spreadsheet scrolls as you drag the scroll box
<b>SelBackColor</b>	Sets or returns the background color of the selected cells
<b>SelForeColor</b>	Sets or returns the foreground color of the selected cells
<b>TextTip</b>	Sets or returns whether the text tips are displayed at all times, only when the control has the focus, or not at all, and when the tips are displayed and how they are positioned
<b>TextTipDelay</b>	Sets or returns the number of milliseconds the control waits before displaying the text tip for the cell under the pointer
<b>TwoDigitYearMax</b> (ActiveX only)	Sets or returns the year value used to calculate the century when year values are given as two-digit values
<b>TypeComboBoxWnd</b> (ActiveX only)	Specifies the window handle of the List Pro fpCombo control
<b>TypeComboBoxMaxDrop</b>	Sets or returns the number of rows to display in the drop-down list of a combo box cell
<b>TypeComboBoxWidth</b>	Sets or returns the width of the drop-down list portion of a combo box cell
<b>TypeMaxEditLen</b>	Sets or returns the maximum number of characters of text that the user can enter either in an edit cell or in the edit field of a combo box cell
<b>TypeVAlign</b>	Sets or returns the vertical alignment in a cell

The following table lists the new properties that have been added for the fpSpreadPreview control for Spread version 3.0.

<b>Property</b>	<b>Description</b>
<b>AllowUserZoom</b>	Sets or returns whether the user can use the mouse to zoom in or out while viewing pages
<b>GrayAreaColor</b>	Sets or returns the color of the preview area of the control
<b>GrayAreaMarginH</b>	Sets or returns the horizontal space between the displayed page and the edge of the control
<b>GrayAreaMarginType</b>	Sets or returns whether the control uses the actual gray area margin settings or scales the settings to optimize the page view
<b>GrayAreaMarginV</b>	Sets or returns the vertical space between the displayed page and the edge of the control
<b>hWndSpread</b>	Sets or returns the window handle of the spreadsheet you want to preview
<b>PageBorderColor</b>	Sets or returns the border color of the displayed pages
<b>PageBorderWidth</b>	Sets or returns the border width of the displayed pages
<b>PageCurrent</b>	Sets or returns the page to display in the control
<b>PageGutterH</b>	Sets or returns the horizontal space between displayed pages
<b>PageGutterV</b>	Sets or returns the vertical space between displayed pages
<b>PageMultiCntH</b>	Sets or returns the number of pages displayed in the horizontal direction
<b>PageMultiCntV</b>	Sets or returns the number of pages displayed in the vertical direction
<b>PagePercentageActual</b>	Returns the size of the displayed pages as a percentage of the actual size
<b>PageShadowColor</b>	Sets or returns the color of the shadow behind the displayed pages
<b>PageShadowWidth</b>	Sets or returns the width of the shadow behind the displayed pages
<b>PagesPerScreen</b>	Returns the number of pages displayed at one time
<b>PageViewPercentage</b>	Sets or returns the percent size of the displayed pages
<b>PageViewType</b>	Sets or returns how pages are displayed
<b>ScrollBarH</b>	Sets or returns whether and how the horizontal scroll bars are displayed
<b>ScrollBarV</b>	Sets or returns whether and how the vertical scroll bars are displayed
<b>ScrollIncH</b>	Sets or returns the horizontal distance the control scrolls when the user scrolls through the displayed pages

## Chapter 1

<b>Property</b>	<b>Description</b>
<b>ScrollIncV</b>	Sets or returns the vertical distance the control scrolls when the user scrolls through the displayed pages
<b>ZoomState</b>	Sets or returns the way the control zooms between different views of the displayed pages

## New Events

The following table lists the new events that have been added to the fpSpread control for Spread version 3.0.

<b>Event</b>	<b>Description</b>
<b>ComboCloseUp</b>	Occurs when the list portion of a combo box cell closes
<b>ComboDropDown</b>	Occurs when the list portion of a combo box cell drops down
<b>ComboSelChange</b>	Occurs when the user changes the selection in a combo box cell
<b>EditChange</b>	Occurs when a cell is in edit mode and the text changes in a cell
<b>OLECompleteDrag</b> (ActiveX only)	Occurs at the source control after an OLE drag-drop operation has been completed or canceled
<b>OLEDragDrop</b> (ActiveX only)	Occurs when data is dropped onto the control
<b>OLEDragOver</b> (ActiveX only)	Occurs when the mouse is moved over the target control during an OLE drag-drop
<b>OLEGiveFeedback</b> (ActiveX only)	Provides visual feedback through the pointer, by the source control for a drag-drop operation
<b>OLESetData</b> (ActiveX only)	Occurs at the drag-drop source control when the drop target requests data that was not provided to the fpDataObject
<b>OLEStartDrag</b> (ActiveX only)	Occurs when an OLE drag-drop operation is initiated either manually or automatically
<b>TextTipFetch</b>	Occurs when the control displays a text tip

The following table lists the new events that have been added for the fpSpreadPreview control for Spread version 3.0.

<b>Event</b>	<b>Description</b>
<b>PageChange</b>	Occurs when the user scrolls or advances to another page
<b>Zoom</b>	Occurs when the control changes the view of the displayed pages by zooming in or out

## New Functions and Methods

The following table lists the new functions and methods that have been added to the fpSpread control for Spread version 3.0

<b>Function</b>	<b>Description</b>
<b>EnumCustomFunction</b>	Returns the name of the next custom function
<b>ExportRangeToHTML</b> (ActiveX only)	Exports a range of cells to an .HTM file
<b>ExportToExcel</b> (ActiveX only)	Exports a spreadsheet to an .XLS file
<b>ExportToHTML</b> (ActiveX only)	Exports a spreadsheet to an .HTM file
<b>GetActionKey</b>	Retrieves the shortcut key or navigation key assigned to a specific action
<b>GetArray</b>	Retrieves data from the spreadsheet and places it in an array
<b>GetCustomFunction</b>	Retrieves information about the specified custom function
<b>GetExcelSheetList</b> (ActiveX only)	Retrieves a handle to the Excel workbook and a list of the sheet names in the specified Excel-formatted file
<b>GetFloat</b>	Retrieves the value in a float cell as a floating-point value
<b>GetInteger</b>	Retrieves the value in an integer cell as a numeric value
<b>GetNextPageBreakCol</b> (VBX only)	Returns the column number of the next column in the spreadsheet at which a page break occurs
<b>GetNextPageBreakRow</b> (VBX only)	Returns the row number of the next row in the spreadsheet at which a page break occurs
<b>GetOddEvenRowColor</b>	Retrieves the background and text colors for odd and even rows
<b>GetPrintOptions</b> (VBX only)	Retrieves whether to use Smart Print, the order in which pages are printed, and the number to use to begin numbering the pages in the spreadsheet
<b>GetPrintPageCount</b> (VBX only)	Retrieves the number of pages required for printing the spreadsheet with the current print settings
<b>GetTextTipAppearance</b>	Retrieves the appearance of the text tip
<b>GetTwoDigitYearMax</b> (VBX only)	Retrieves the year value used to calculate the century when year values are given as two-digit values
<b>ImportExcelSheet</b> (ActiveX only)	Loads an Excel 97 worksheet
<b>IsExcelFile</b> (ActiveX only)	Determines whether an Excel file is in BIFF8 format
<b>OLEDrag</b> (ActiveX only)	Initiates an OLE drag-drop operation with the control as the source

## Chapter 1

<b>Function</b>	<b>Description</b>
<b>ReCalcCell</b>	Recalculates the formula in an individual cell
<b>RemoveCustomFunction</b>	Removes the specified custom function
<b>SetActionKey</b>	Specifies the shortcut key or navigation key assigned to a specific action
<b>SetArray</b>	Loads data into the spreadsheet from an array
<b>SetCalText</b>	Specifies the text for the pop-up calendar in a date cell
<b>SetFloat</b>	Specifies the floating-point value for a float cell
<b>SetInteger</b>	Specifies the integer value for an integer cell
<b>SetOddEvenRowColor</b>	Specifies the background and text colors for odd and even rows
<b>SetPrintOptions</b> (VBX only)	Specifies whether to use Smart Print, the order in which pages are printed, and the number to use to begin numbering the pages in the spreadsheet
<b>SetTextTipAppearance</b>	Specifies the appearance of a text tip
<b>SetTwoDigitYearMax</b> (VBX only)	Specifies the year value used to calculate the century when year values are given as two-digit values

### ActiveX Controls

Spread version 3.0 does not include a 16-bit fpSpread ActiveX control. If you currently use the Spread version 2.5 16-bit vaSpread ActiveX control and want to upgrade to Spread version 3.0, you must upgrade your projects to use the 32-bit vaSpread ActiveX control. For more information on upgrading your projects, see the Read Me help file (SPREADME.HLP) that accompanies this product.

The Spread version 3.0 ActiveX controls are safe for scripting. If you are using the Spread ActiveX control in a container that supports IObjectSafety such as Internet Explorer 4, the user cannot export or save files.

# Chapter 2 The fpSpread Control

FarPoint Technologies' fpSpread control provides a flexible, powerful tool for creating spreadsheets and tables.

## Features

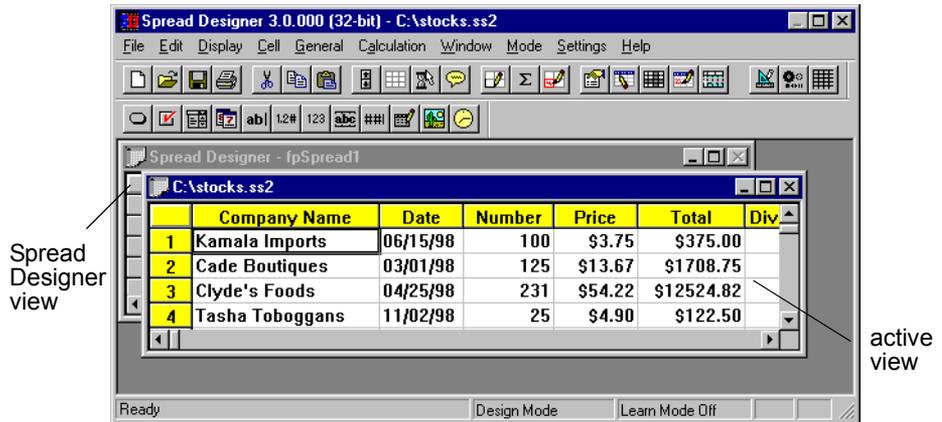
The fpSpread control provides several unique features that are described in the following sections. For more information about using these features, consult the referenced sections in Part II, "How-to Guides."

## Spread Designer

The Spread Designer lets ActiveX and VBX users design spreadsheets quickly by offering a design-time environment that displays changes to the spreadsheet as they are made.

Within the Spread Designer, you can view a design-time version of your spreadsheet in design mode. You can also switch to a run-time version of the spreadsheet, which shows your spreadsheet as it will appear when it is in your application. For example, when working with hidden columns, you can view the columns and design them in design mode. When you switch to run-time mode, the columns are hidden.

The Spread Designer lets you open multiple spreadsheet views. The Spread Designer view is the default view provided by the designer. The Spread Designer view displays the spreadsheet in its current design-time state. You can open views of other existing spreadsheets created using FarPoint's fpSpread control, as shown in the following figure.



## Chapter 2

In Spread Designer, you can also open spreadsheets that were created in Excel 97.

Once open, you can modify the Spread Designer view or one of the open views, and then choose which view, if any, to apply to the fpSpread control.

Using learn mode, you can generate documented Visual Basic code for actions you perform while working in the Spread Designer. You can view this code in the Spread Designer, and you can paste this code in your application.

Using cell type mode, you can see at a glance the type of cell in the spreadsheet.

For instructions for using the Spread Designer, see Chapter 4, “Using the Spread Designer.”

## Cells

By default, cells in the spreadsheet are edit cells. You can choose to designate cells to be one of 12 available types as described in the following table.

<b>Cell Type</b>	<b>Description</b>
Button	Displays a button in a cell (The button cell can contain text, graphics, or both.)
Check Box	Displays a check box in a cell
Combo Box	Displays a combo box in a cell and provides a list of items for display in the combo box
Date	Displays or allows editing of formatted dates in a cell
Edit	Displays or allows editing of text in a cell
Float	Displays or allows editing of a formatted floating-point value in a cell (This cell type is usually used to display currency values.)
Integer	Displays or allows editing of an integer value in a cell (If you are using Visual Basic, this cell type handles long integers.)
Owner-Drawn	Displays a user-defined cell type
PIC	Displays or allows editing of a formatted PIC (mask) in a cell
Picture	Displays a picture in a cell
Static Text	Displays noneditable, nonscrolling text in a cell
Time	Displays or allows editing of a formatted time value in a cell

Each cell type presents special characteristics that you can customize for your needs. For example, you can specify maximum and minimum values, a separator character, and a format for date cells.

The following figure illustrates date cells that have the format MM/DD/YY and that have a slash character as a separator.

Date
06/15/94
03/01/94
04/25/94
11/02/94

date cell

For more information about selecting a cell type for a cell, see Chapter 11, “Setting the Cell Type.”

Individual cells can display specialized borders and background and foreground colors. You can align text in a cell, and in addition, cells can display text as overflowing into adjacent cells, if you prefer. For more information about customizing cell appearance and cell text, see “Customizing Cell Borders” on page 202, “Allowing Cells to Overflow” on page 198, and the descriptions for customizing cell types in Chapter 11, “Setting the Cell Type.”

You can prevent users from interacting with a cell by locking the cell. For more information, see “Locking Cells” on page 192.

## Columns and Rows

Many of the custom features you can apply to an individual cell you can also specify for a column, a row, or a range of columns or rows. The spreadsheet also provides headers in the form of column headers and row headers that number the columns and rows or provide additional information about their content. In addition, you can freeze columns and rows, size them according to the data they contain, and hide them.

Headers can be customized as you need. Numbering can be displayed as numbers or letters, or you can display text in the header for a specific row or column. You can specify the background and foreground colors for the headers, and specify whether shadows are displayed and what colors are used in the shadows. If you prefer, you can hide the column headers or row headers.

## Chapter 2

The following figure illustrates a spreadsheet with customized column headers and row headers.

	M	T	W	T
7:00		Meet the Teacher Day at Vicky's school	Met. Appt. Tasha at 7:45	
8:00	Production meeting	Teacher - Parent conference at 8:15		

For more information about customizing headers, see “Customizing Column Headers and Row Headers” on page 179.

Frozen columns do not scroll as the user scrolls through the spreadsheet. By default, the spreadsheet’s column header and row header are frozen. You can specify additional adjacent columns and rows that do not scroll. For information about freezing columns and rows, see “Freezing Columns and Rows” on page 177.

You can specify the width of a column and the height of a row, or you can let the dimensions vary depending on the data contained in the column or row. In addition, you can allow users to resize columns and rows in your application. For information about working with column width and row height, see “Sizing Columns and Rows” on page 166.

## Printing

Spread lets you customize your printed spreadsheet with a variety of options.

Your printed pages can include headers and footers. You can decide whether you want to print the spreadsheet border, the column header and row header, the spreadsheet grid, colors, and shadows in headers. You can specify the page number that prints on the first page.

Spread lets you customize which portion of the spreadsheet is printed and the order in which pages print. You can print just the portion of the spreadsheet that contains data or specify a page range. You can place page breaks anywhere you need them.

Pages can print in either portrait or landscape orientation. In addition, you can specify left, right, top, and bottom margins for the printed pages. You can also use the Smart Print option, which automatically adjusts the printing to accommodate the spreadsheet data.

Spread Designer users can choose to print the active view as they work with open spreadsheets, and to apply the print options set for the active view to the options set for the spreadsheet.

For more information about printing spreadsheets, see Chapter 7, “Printing Spreadsheets.” For more information about printing the active view from the Spread Designer, see “Printing the Active View” on page 58.

## Loading, Saving, or Exporting Data

You can add formatted or unformatted data to a spreadsheet in one of the following ways:

- Bind the data to a database
- Load an existing spreadsheet that is either a binary file, a tab-delimited file, or a Microsoft Excel 97 worksheet
- Load an existing spreadsheet from a buffer (DLL control users only)
- Load data using Spread properties and functions

For more information about data binding, see Chapter 12, “Binding to Databases.” For more information about the other methods for adding data, see “Adding Data to a Spreadsheet” on page 75.

You can save spreadsheet data using one of the following file formats:

- Binary file format
- Tab-delimited format

You can export spreadsheet data using one of the following file formats:

- Excel 97 format (BIFF8) (ActiveX or 32-bit DLL controls only)
- HTML format (ActiveX or 32-bit DLL controls only)

---

**Note** The Spread version 3.0 ActiveX controls are safe for scripting. If you are using the Spread ActiveX control in a container that supports IObjectSafety, the user cannot export or save files.

---

For more information on saving and exporting data, see “Saving Spreadsheet Data” on page 115.

### Formulas

With Spread you can provide formulas in cells, taking advantage of Spread's Calc Engine. In addition, Spread offers relative and absolute cell referencing. You can use the 74 operations provided by the Calc Engine, or you can create your own functions to use in formulas. If you prefer, you can allow users to enter formulas in cells.

For more information about formulas, see "Entering Formulas" on page 102.

### Virtual Mode

Virtual mode lets the fpSpread control read in only the amount of data it needs to display the requested rows. Therefore, virtual mode increases responsiveness and conserves system resources.

When the fpSpread control is in virtual mode and the user scrolls through spreadsheet rows, rows are read and stored in a buffer until they are displayed. Spread lets you customize virtual mode to fit your needs: you can specify how many rows are retrieved into the buffer and how many displayed rows are retained in the buffer. In addition, you can customize the scroll bars for use with virtual mode.

For more information about virtual mode, see "Using Virtual Mode" on page 130.

### Data Binding

Using the ActiveX or VBX controls, you can bind the fpSpread control using Data Access Object (DAO), Remote Data Object (RDO), or ActiveX Data Object (ADO) to display database records. Many spreadsheet characteristics can be set automatically when the database is read by the fpSpread control, including column header text, column size, and cell type. Alternatively, you can tailor these and other characteristics to meet your needs, including optimizing performance using virtual mode.

The following figure illustrates a fpSpread control bound to a database. The fpSpread control has automatically configured itself to display the database records.

	Author	Title	Year	ISBN
1	Anne Ball	Alex: A parrot's tale	1995	0-0280875-2
2	Lydia Kernville	Curly tails and white paws	1994	0-0480095-2
3	Julia Shell	Clyde's adventures	1994	0-0275195-2
4	Russell Lumky	Dog training for children	1995	0-0283095-4
5	Robby Dixie-Smith	Weimaraners: Your friends	1992	0-3256095-2
6	Scott Peterson	Caddy: A boy and his dog	1992	0-0453495-4
7	Chris Persons	Iguanas, the perfect pets	1993	0-3256325-2
8	Edith Anne Leach	The care and feeding of puppies	1996	0-3258695-2

For more information about data binding, see Chapter 12, “Binding to Databases.”

## Navigation

Spread provides multiple ways to navigate within the spreadsheet. Besides allowing you to set how the Tab and Enter keys are interpreted by your application, Spread supports the following navigation keys:

Key	Action
up arrow	Moves active cell up one row
down arrow	Moves active cell down one row
right arrow	Moves active cell right one column
left arrow	Moves active cell left one column
Shift+arrow key	Extends selection in direction of arrow key
Page Up	Moves active cell one page up
Page Down	Moves active cell one page down
Ctrl+Page Up	Moves active cell one page left
Ctrl+Page Down	Moves active cell one page right
Home	Moves active cell to first cell in row
End	Moves active cell to last cell in row that contains data
Ctrl+Home	Moves active cell to first row, first column
Ctrl+End	Moves active cell to last row and column that contain data
Tab	If the <b>ProcessTab</b> property is set to True or the <b>SSB_PROCESSTAB</b> option for the <b>SSSetBool</b> function is set to True, moves active cell to next cell to the right (or at end of row moves to beginning of next row)

## Chapter 2

<b>Key</b>	<b>Action</b>
Shift+Tab	If the <b>ProcessTab</b> property is set to True or the SSB_PROCESSTAB option for the <b>SSSetBool</b> function is set to True, moves active cell to next cell to the left (or at beginning of row moves up to end of row above)
Shift+space	Selects current row
Ctrl+space	Selects current column
Shift+Ctrl+space	Selects entire spreadsheet
Ctrl+X or Shift+Del	Cuts current selection or active cell's data to Clipboard
Ctrl+V or Shift+Ins	Pastes Clipboard contents into active cell
Ctrl+C or Ctrl+Ins	Copies current selection or active cell's data to Clipboard
Enter	Depends on setting of the <b>EditEnterAction</b> property or <b>SSSetEditEnterAction</b> function
Esc	If spreadsheet is in edit mode, previous cell value replaces new value and edit mode is turned off
F2	If edit mode is on, cell value is cleared
F3	If edit mode is on in a date or time cell, current date or time is placed in cell
F4	If edit mode is on in a date cell, spreadsheet displays a pop-up calendar to let you choose a date

For information about specifying how the control handles the Tab and Enter keys, see “Using the Tab Key to Move Between Cells” on page 299 and “Using the Enter Key to Move Between Cells” on page 298.

If you prefer, you can change or turn off the F2, F3, or F4 keys. For instructions for changing the action key settings, see “Changing or Turning Off the F2, F3, and F4 Action Keys” on page 302.

## Display

You can customize the spreadsheet display to create colorful spreadsheets, tables, and more. If you prefer, your spreadsheet need not display an outer border.

You can also choose whether to display horizontal or vertical grid lines or both. If grid lines are displayed, you can specify their color and style.

A spreadsheet with customized grid lines is shown in the following figure.

1st Year	2nd Year	3rd Year	4th Year
English I	English II	Brit. Lit.	Amer. Lit.
History	Economics	Polit. Sci.	Amer. Hist.
Spanish I	Spanish II	Spanish III	Spanish IV
Free	Chemistry	Free	Physics
Biology	Free	Biology II	Calculus
Algebra	Geometry	Algebra II	Chorus
Chorus	Art	Drama	Dance
Phys. Ed.	Phys. Ed.	Dance	Free

For instructions and more information about customizing grid lines, see “Customizing Grid Lines” on page 283.

You can ensure that the spreadsheet only displays complete columns or rows rather than partial ones. In addition, you can specify how the spreadsheet aligns with the outer borders of the control and how many columns or rows are displayed at one time. For more information, see “Displaying a Specified Number of Columns and Rows” on page 173.

You can customize many additional display features, including the color of the area behind the spreadsheet in the fpSpread control, as described in “Setting the Gray Area Color” on page 287.

For additional information and instructions, see Chapter 13, “Customizing the Spreadsheet Display.”

## User Interaction

By default, users can select cells, columns, rows, or the entire spreadsheet. You can restrict which elements in the spreadsheet users can select, as described in “Specifying Which Elements Users Can Select” on page 126.

Cells that allow user input can provide customized edit modes, such as automatically replacing existing text as the user types. For more information, see “Replacing Text as You Edit” on page 201.

You can let users drag and drop cells or blocks of cells, as described in “Allowing Users to Drag and Drop Cells” on page 291.

Scroll bars can be displayed all the time or only when needed. You can make the columns and rows move in synchronization with the horizontal and vertical scroll box movement. You may want to display a special vertical scroll bar that displays scroll arrows rather than a scroll box, particularly if you are using virtual mode. For more information about scroll

bars, see “Displaying Scroll Bars” on page 293 and “Displaying the Special Vertical Scroll Bar” on page 136.

You can customize many additional features, including how selected items are displayed and when buttons appear in cells that can display buttons. For more information, see “Hiding Selections When the Spreadsheet Does Not Have the Focus” on page 290 and “Specifying When Buttons Appear” on page 305.

If you prefer, the spreadsheet can act similarly to a list box, letting users select single rows or multiple rows, depending on your settings. Alternatively, you can set the spreadsheet to act as a read-only control, or work in rows. For more information, see “Specifying the Operation Mode” on page 128.

You can display text tips for any header and cell in the control. For more information, see “Providing Text Tips” on page 306.

## Concepts

The following sections describe concepts that you need to understand when using the fpSpread control. You should review and understand these concepts before working with the control as described in Part II, “How-to Guides.”

### Formatted and Unformatted Data

Data in the spreadsheet can have two “parts” to it, the data itself and the additional characteristics assigned to the data. For example, a cell with monetary information contains the numeric value for the money, such as 1324.56, but might also contain formatting that identifies the contents as money, such as a dollar sign, a separator character, and a decimal symbol (\$1,324.56).

In addition to the data in the spreadsheet, the cell in which the data resides can have characteristics assigned to it. For example, the cell containing the data could be a float cell type, with a red background, and it might contain a formula.

When you work with data in the spreadsheet, adding it or moving it, for example, you need to keep in mind the different characteristics associated with the data. For example, when you move data in the spreadsheet, the values in the cells are moved. However, depending on how you move the cells, the cell type information for the cells might not move.

Throughout the instructional material in Part II, “How-to Guides,” where applicable, the introduction to the section describes the different ways you can work with data and what formatting and cell information will be lost or maintained.

You also need to keep in mind what information you want returned when returning data from the spreadsheet. Depending on how you return the information, you might or might not return the formatting information, such as a dollar sign in a monetary value.

## Selecting Spreadsheet Elements

Many spreadsheet properties, functions, and methods perform actions on a cell, a range of cells, a column, a row, or the entire spreadsheet. Before using these properties, functions, and methods, you must often select or specify the element for which you want to specify characteristics.

DLL users specify the cell, column, row, or spreadsheet as parameters of functions. ActiveX and VBX users set the **Col** and **Row** properties to identify the coordinates of a cell, the column number, the row number, or the entire spreadsheet, as explained in the following sections.

### Relationship of Col and Row Properties or Parameters

The following table shows how the **Col** and **Row** properties for the fpSpread ActiveX and VBX controls or the *Col* and *Row* parameters in DLL functions are used in relation to one another to specify the cell, column (or row headers), row (or column headers), or spreadsheet.

Col Property Value	Row Property Value	Col Parameter Value	Row Parameter Value	Action is performed on . . .
-1	-1	SS_ALLCOLS	SS_ALLROWS	The entire spreadsheet
-1	0	SS_ALLCOLS	SS_HEADER	The column headers
-1	nonzero	SS_ALLCOLS	nonzero	The specified row
0	-1	SS_HEADER	SS_ALLROWS	The row headers
0	0	SS_HEADER	SS_HEADER	The gray cell in the upper-left corner of the spreadsheet
0	nonzero	SS_HEADER	nonzero	The header cell at the specified row within the row headers
nonzero	-1	nonzero	SS_ALLROWS	The specified column
nonzero	0	nonzero	SS_HEADER	The header cell at the specified column within the column headers
nonzero	nonzero	nonzero	nonzero	The specified cell

Many spreadsheet properties, functions, and methods rely on the **Col** and **Row** properties (ActiveX or VBX controls) or the *Col* and *Row* parameters (DLL controls) to specify the cell, column, row, or the entire spreadsheet on which to perform the desired operation.

If an operation is to be performed on an entire column or row, for example, when setting the **CellType** property, specify to apply the action to the entire column or row instead of performing the operation on each cell within the column or row. This optimizes speed and memory.

### Working with Selected Cells

Because of the spreadsheet's predefined precedence, as described in "Cell, Row, Column, and Spreadsheet Precedence," cells can have information stored for them at four levels:

- For the cell itself
- For the cell's row (when you have set the **Col** property or *Col* parameter to -1)
- For the cell's column (when you have set the **Row** property or *Row* parameter to -1), and
- For the entire spreadsheet

When you perform an action on a cell, such as moving the cell, only the information stored for the cell itself is acted on. For example, if you copy a cell, the background color defined for the cell is also copied; however, the background color defined for the cell's column, row, or the entire spreadsheet is not copied.

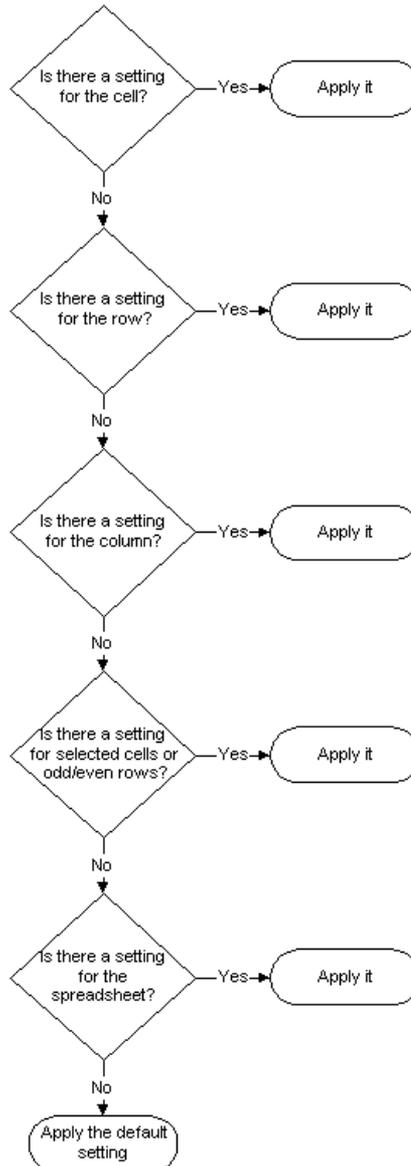
Logically, when you set a characteristic for a column, a row, or the entire spreadsheet, that setting is not stored for the cell, and therefore is not included in actions performed on the cell.

### Cell, Row, Column, and Spreadsheet Precedence

You can designate which portion of the spreadsheet you want to set characteristics for, such as a particular column or cell. Therefore, for an individual cell, there might be settings for the background colors for the spreadsheet, the row the cell is in, the column the cell is in, and the cell itself.

The spreadsheet determines which color to display by using a predefined precedence when applying property, function, or method settings. Individual cells have the highest precedence. For example, if you set the background color for cell B2 to red and the background color of row 2 is blue, the background color of column 2 is white, and the background color of the spreadsheet is gray, the cell would display the color set for it (red), not for the row, column, or spreadsheet.

The spreadsheet uses the following method to determine the setting to apply to each element in the spreadsheet:



Because of the precedence used by the spreadsheet, many properties, functions, and methods that can be applied separately to different portions of the spreadsheet provide a

“default” setting. When you choose the default setting, the element, such as a cell, uses the setting for the next level setting available: the row, then the column, then the selected cell or odd/even row, or finally, the spreadsheet.

The following properties, functions, and methods allow you to set characteristics for different elements in the spreadsheet, and use the predefined precedence to determine which setting to apply to which element:

BackColor	FontBold	ForeColor
CellBorderColor	FontItalic	Lock
CellBorderStyle	FontName	All “Type . . .” properties
CellBorderType	FontSize	UserResizeCol
CellType	FontStrikethru	UserResizeRow
Font	FontUnderline	

## Components of the fpSpread ActiveX and VBX Controls

If you are using the fpSpread ActiveX or VBX control, properties and events let you customize the control for your interfaces, and methods provide ActiveX and VBX users the opportunity to set multiple characteristics at one time.

The following sections provide complete lists of the events, and functions or methods available for the fpSpread control.

### Properties

The fpSpread control provides over 200 properties for customizing the control. For a summary list of the properties provided for the fpSpread control, refer to Appendix A, “fpSpread Control Property Summary Table.”

### Events

The following events are provided for the fpSpread control. For detailed information about each event, refer to the event reference in the online *ActiveX/VBX Reference Guide*.

Advance	ComboSelChange	DragOver
BlockSelected	CustomFunction	DrawItem
ButtonClicked	DataAddNew	EditChange
Change	DataColConfig	EditError
Click	DataFill	EditMode
ColWidthChange	DbClick	EnterRow
ComboCloseUp	DragDrop	GotFocus
ComboDropDown	DragDropBlock	KeyDown

KeyPress	OLEDragDrop (ActiveX)	RightClick
KeyUp	OLEDragOver (ActiveX)	RowHeightChange
LeaveCell	OLEGiveFeedback (ActiveX)	SelChange
LeaveRow	OLESetData (ActiveX)	TextTipFetch
LostFocus	OLEStartDrag (ActiveX)	TopLeftChange
MouseDown	PrintAbort	UserFormulaEntered
MouseMove	PrintMsgBox	Validate
MouseUp	QueryAdvance	VirtualClearData
OLECompleteDrag (ActiveX)	QueryData	

## Functions and Methods

The following functions and methods are provided for the fpSpread control. ActiveX users use methods with the same names. For detailed information about each function or method, refer to the functions and method reference in the online *ActiveX/VBX Reference Guide*.

AddCustomFunction	GetCustomFunction	LoadFromFile
AddCustomFunctionExt	GetDataFillData	LoadTabFile
CFGetCellParam	GetExcelSheetList (ActiveX)	OLEDrag (ActiveX)
CFGetDoubleParam	GetFirstValidCell	QueryCustomName
CFGetDoubleParamExt	GetFloat	ReCalcCell
CFGetLongParam	GetFormulaSync	RemoveCustomFunction
CFGetParamInfo	GetInteger	RowHeightToTwips
CFGetRangeParam	GetItemData	SaveTabFile
CFGetStringParam	GetIteration	SaveToFile
CFSetResult	GetLastValidCell	SetActionKey
ColNumberToLetter	GetMultiSelItem	SetArray
ColWidthToTwips	GetNextPageBreakCol (VBX)	SetCalText
DerefHlstrLen	GetNextPageBreakRow (VBX)	SetCellDirtyFlag
EnumCustomFunction	GetOddEvenRowColor	SetColItemData
ExportRangeToHTML (ActiveX)	GetPrintOptions (VBX)	SetCustomName
ExportToExcel (ActiveX)	GetPrintPageCount (VBX)	SetDataFillData
ExportToHTML (ActiveX)	GetRefStyle	SetFloat
GetActionKey	GetRowItemData	SetFormulaSync
GetArray	GetText	SetInteger
GetBottomRightCell	GetTextTipAppearance	SetItemData
GetCellDirtyFlag	GetTwoDigitYearMax (VBX)	SetIteration
GetCellFromScreenCoord	ImportExcelSheet (ActiveX)	SetOddEvenRowColor
GetCellPos	IsCellSelected	SetPrintOptions (VBX)
GetClientArea	IsExcelFile (ActiveX)	SetRefStyle
GetColItemData	IsFormulaValid	SetRowItemData
GetCustomName	IsVisible	SetText

SetTextTipAppearance  
SetTwoDigitYearMax (VBX)

TwipsToColWidth

TwipsToRowHeight

## Components of the fpSpread DLL Controls

Functions let you set characteristics of the fpSpread control. Messages convey information from the fpSpread control to your application. Structures provide a mechanism for conveying multiple characteristics' settings. Styles let you customize the fpSpread control.

The following sections provide complete lists of the messages, structures, and styles available for the fpSpread control.

### Functions

The fpSpread control provides over 200 functions for customizing the control. For a summary list of the functions provided for the fpSpread control, refer to Appendix C, "fpSpread and fpSpreadPreview DLL Function Summary."

### Messages

The following messages are sent from the fpSpread control to the application. For detailed information about each message, refer to the message reference in the online *DLL Reference Guide*.

SSM_CELLLOCKED	SSM_EDITMODEON	SSN_BLOCKDESELECTED
SSM_CLICK	SSM_ENTERROW	SSN_BLOCKSELECTED
SSM_COLRESTRICTED	SSM_KEYDOWN	SSN_BUTTONDOWN
SSM_COLROWRESTRICTED	SSM_KEYPRESS	SSN_BUTTONUP
SSM_COLWIDTHCHANGE	SSM_KEYUP	SSN_EXITNEXT
SSM_COMBOCLOSEUP	SSM_LEAVECELL	SSN_EXITPREV
SSM_COMBODROPDOWN	SSM_LEAVEROW	SSN_INVALIDDATA
SSM_COMBOSELCHANGE	SSM_RBUTTON	SSN_KILLFOCUS
SSM_DATACHANGE	SSM_ROWHEIGHTCHANGE	SSN_LEFTCOLCHANGE
SSM_DBLCLK	SSM_ROWRESTRICTED	SSN_SETFOCUS
SSM_DRAGDROP	SSM_TEXTTIPFETCH	SSN_TOPLEFTCHANGE
SSM_DRAWITEM	SSM_VCLEARDATA	SSN_TOPROWCHANGE
SSM_EDITCHANGE	SSM_VQUERYDATA	SSN_USERFORMULA
SSM_EDITMODEOFF	SSN_BLOCKCHANGED	

## Structures

The following structures are provided for the fpSpread control. For detailed information about each structure, refer to the structures reference in the online *DLL Reference Guide*.

SS_CELLCOORD	SS_ENTERROW	SS_SORT
SS_CELLREF	SS_FLOATFORMAT	SS_SORTKEY
SS_CELLTYPE	SS_LEAVECELL	SS_TEXTTIPFETCH
SS_COMBOCLOSEUP	SS_LEAVEROW	SS_TIME
SS_COORDRANGE	SS_PRINTFORMAT	SS_TIMEFORMAT
SS_DATE	SS_PRINTINFO	SS_VALUE
SS_DATEFORMAT	SS_RANGEREFF	SS_VCLEARDATA
SS_DRAGDROP	SS_RBUTTON	SS_VQUERYDATA
SS_DRAWITEMSTRUCT	SS_SELBLOCK	SUPERBTNCOLOR
SS_EDITMODEOFF		

## Styles

The following styles are provided for the fpSpread control. For detailed information about each style, refer to styles reference in the online *DLL Reference Guide*.

SSS_ALLOWCELLOVERFLOW	SSS_LOCKALL	SSS_PROTECT
SSS_ALLOWUSERFORMULAS	SSS_NOGRIDLINES	SSS_RESTRICTCOLS
SSS_HIDECOLHEADERS	SSS_NORESIZE	SSS_RESTRICTROWS
SSS_HIDEROWHEADERS	SSS_NOSELECTBLOCK	SSS_SCROLLBAREXTMODE



# Chapter 3 The fpSpreadPreview Control

FarPoint Technologies' fpSpreadPreview control lets you provide users with a way to view spreadsheets before they print them.

## Features

The fpSpreadPreview control provides several unique features that are described in the following sections. For more information about using these features, consult the referenced sections in Part II, "How-to Guides."

A sample project that illustrates the fpSpreadPreview control is provided with Spread 3.0 in the \Spread30\Samples directory.

## Print Preview

The fpSpreadPreview control lets you allow users to see how their fpSpread control's spreadsheet will print; in other words, it lets you provide a print preview feature for the fpSpread control.

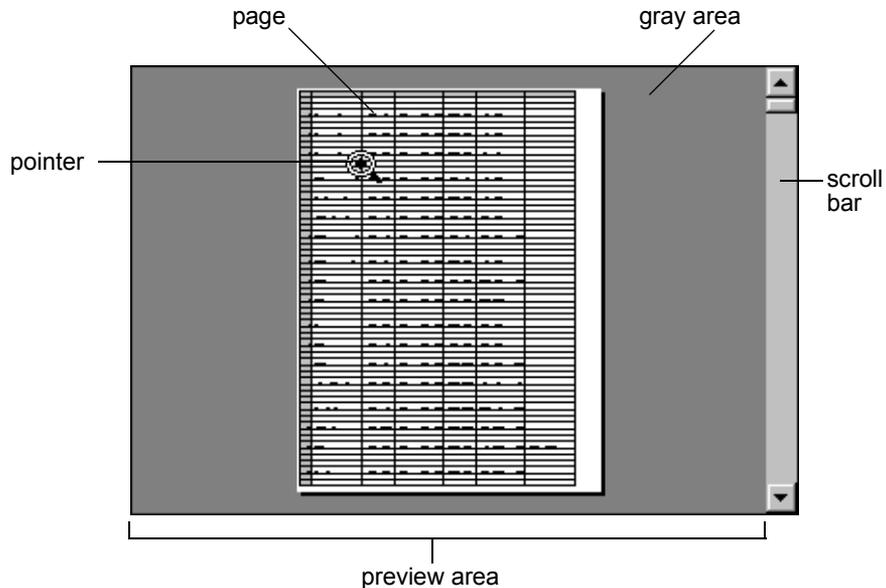
The pages displayed in the fpSpreadPreview control reflect the various print settings available, including the SmartPrint option, headers and footers, and page break settings. Therefore, you might want to prompt your users to set the various printing options they want before they preview the printed spreadsheet.

---

### Notes

- If your printer does not support the fonts that are used in the spreadsheet and shown on the preview page, Windows will substitute another font. Consequently, the printed spreadsheet will not be the same as what you see on the preview page. To avoid this problem, use TrueType® fonts.
  - If you are using virtual mode, the fpSpreadPreview control displays only the number of rows that are in the fpSpread control's virtual buffer. For more information about how to specify the size of the virtual buffer, see "Customizing Virtual Mode" on page 134.
- 

The following figure shows an fpSpreadPreview control with default settings displaying the pages for a sample spreadsheet in an fpSpread control (which is not illustrated in the figure).



For an overview of using the fpSpread control with the fpSpreadPreview control and step-by-step instructions, see “Previewing Your Spreadsheet” on page 153.

## Page Views

As illustrated in the preceding figure, the fpSpreadPreview control displays a page or multiple pages of the spreadsheet as they will appear when printed. You can change how many pages are displayed in the preview area by specifying how many pages are displayed vertically and horizontally.

For instructions for customizing the page view, see “Customizing the Page View” on page 154.

## Zooming

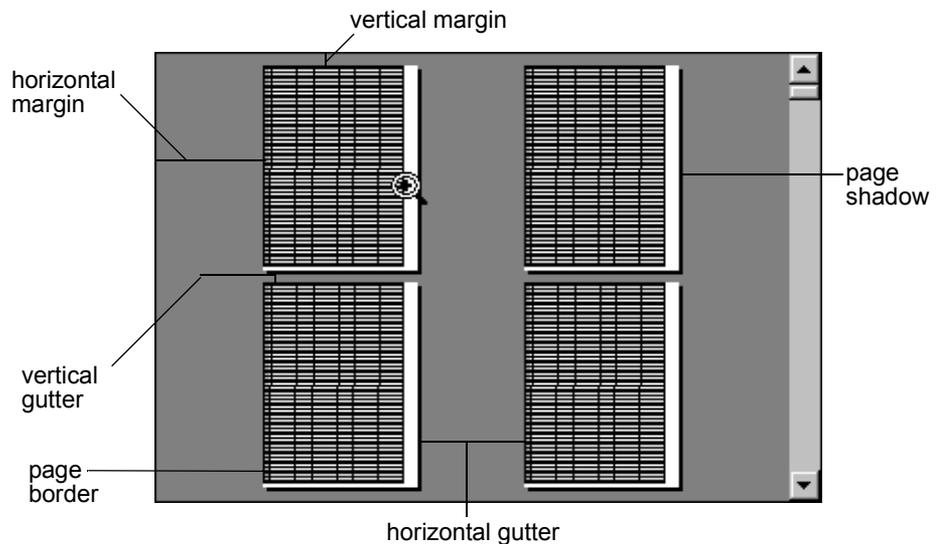
By default, users can zoom in or out to view pages by just pointing to the page to zoom and clicking the mouse. You can provide additional zooming capabilities, including allowing users to zoom to a specific percentage, to view the whole page or a dimension of the page (such as the entire page width), and to zoom using a button or another control. If you prefer, you can prevent users from zooming using the mouse, and just let them zoom using the controls and options you provide.

For instructions for setting up and providing zoom capabilities, see “Zooming In and Out” on page 155.

## Appearance of the Preview Area

The fpSpreadPreview control displays pages in a *preview area*, which by default has a gray background, and is also referred to as the *gray area*. You can specify the color of the background of the preview area, the space between displayed pages and the edge of the control (*margins*), and the space between displayed pages themselves (*gutters*).

The fpSpreadPreview control in the following figure displays four pages in the preview area, and uses the default settings for margins and gutters.



As illustrated in the preceding figure, each page displays a color border along the top and left sides of the page. You can customize the color and width of the border. In addition, pages are displayed with a shadow behind them (along the right and bottom sides of the page), creating a three-dimensional appearance. You can customize the color and the width of the shadow, as well.

For instructions to customize these features, see the following topics:

- “Customizing the Gray Area” on page 158
- “Changing the Page Gutters” on page 159

## Chapter 3

- “Customizing the Page Border” on page 158
- “Customizing the Page Shadow” on page 157.

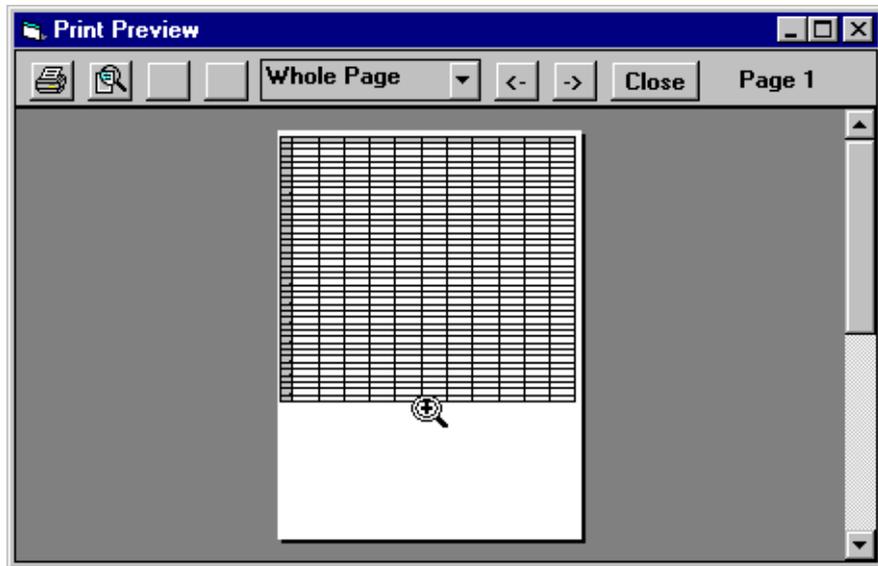
### Scroll Bars

As you can see in the preceding illustrations, the fpSpreadPreview control displays scroll bars by default if the page displayed or the total number of pages fall outside the preview area in the control.

You can specify, for both horizontal and vertical scroll bars independently, that the scroll bars are not displayed or that they are displayed at all times. You can also specify for either scroll bar the increment used for scrolling when the user moves the scroll box. For instructions, see “Displaying Scroll Bars” on page 156.

### Summary

The following figure illustrates an example of how you might configure and provide the fpSpreadPreview control in an application. In the figure, you can see that the application provides multiple ways to let the user view the pages in the preview area.



If you would like to see the code that created the project for this figure, look in the `\Spread30\Samples` directory at the provided PRINTPRV sample projects.

## Components of the fpSpreadPreview ActiveX and VBX Controls

If you are using the fpSpreadPreview ActiveX or VBX control, properties and events let you customize the control for your interface.

The following sections provide complete lists of the properties and events available for the fpSpreadPreview control.

### Properties

The following properties are provided for the fpSpreadPreview control. For detailed information about each property, refer to the property reference in the online *ActiveX/VBX Reference Guide*. For a summary table with information about each property, see Appendix B, “fpSpreadPreview Control Property Summary Table.”

AllowUserZoom	PageCurrent	PagesPerScreen
GrayAreaColor	PageGutterH	PageViewPercentage
GrayAreaMarginH	PageGutterV	PageViewType
GrayAreaMarginType	PageMultiCntH	ScrollBarH
GrayAreaMarginV	PageMultiCntV	ScrollBarV
hWndSpread	PagePercentageActual	ScrollIncH
OLEDropMode (ActiveX)	PageShadowColor	ScrollIncV
PageBorderColor	PageShadowWidth	ZoomState
PageBorderWidth		

### Events

The following events are provided for the fpSpreadPreview control. For detailed information about these events, refer to the event reference in the online *ActiveX/VBX Reference Guide*.

OLEDragDrop (ActiveX)	OLESetData (ActiveX)	Zoom
OLEDragOver (ActiveX)	OLEStartDrag (ActiveX)	
OLEGiveFeedback (ActiveX)	PageChange	

### Method

The **OLEDrag** method is provided for the fpSpreadPreview ActiveX control. For detailed information about the method, refer to the OLE drag-and-drop information in the online *ActiveX/VBX Reference Guide*.

## Components of the fpSpreadPreview DLL Controls

Functions let you set characteristics of the fpSpreadPreview control. Messages convey information from the fpSpreadPreview control to your application.

The following sections provide complete lists of the functions and messages available for the fpSpreadPreview control.

### Functions

The following functions are provided for the fpSpreadPreview control. For detailed information about each function, refer to the function reference in the online *DLL Reference Guide*.

SpvGetAllowUserZoom	SpvGetPagesPerScreen	SpvSetPageBorderWidth
SpvGetGrayAreaColor	SpvGetPageViewPercentage	SpvSetPageCurrent
SpvGetGrayAreaMarginH	SpvGetPageViewType	SpvSetPageGutterH
SpvGetGrayAreaMarginType	SpvGetScrollBarH	SpvSetPageGutterV
SpvGetGrayAreaMarginV	SpvGetScrollBarV	SpvSetPageMultiCntH
SpvGetHWndSpread	SpvGetScrollIncH	SpvSetPageMultiCntV
SpvGetPageBorderColor	SpvGetScrollIncV	SpvSetPageShadowColor
SpvGetPageBorderWidth	SpvGetZoomState	SpvSetPageShadowWidth
SpvGetPageCurrent	SpvSetAllowUserZoom	SpvSetPageViewPercentage
SpvGetPageGutterH	SpvSetGrayAreaColor	SpvSetPageViewType
SpvGetPageGutterV	SpvSetGrayAreaMarginH	SpvSetScrollBarH
SpvGetPageMultiCntH	SpvSetGrayAreaMarginType	SpvSetScrollBarV
SpvGetPageMultiCntV	SpvSetGrayAreaMarginV	SpvSetScrollIncH
SpvGetPagePercentageActual	SpvSetHWndSpread	SpvSetScrollIncV
SpvGetPageShadowColor	SpvSetPageBorderColor	SpvSetZoomState
SpvGetPageShadowWidth		

### Messages

The following messages are sent from the fpSpreadPreview control to the application. For detailed information about each message, refer to the message reference in the online *DLL Reference Guide*.

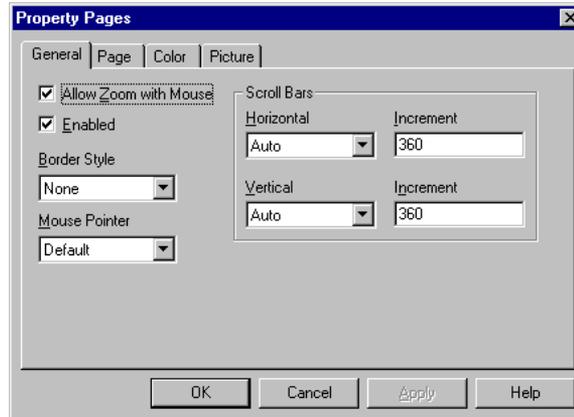
SPVN\_PAGECHANGE  
SPVN\_ZOOM

## fpSpreadPreview Property Pages

The fpSpreadPreview control provides four property pages for customizing your interface.

### General Property Page

The General property page lets you adjust the scroll bar settings, the scroll bar increments, and the border style, customize the pointer, and allow the user to zoom using the mouse.



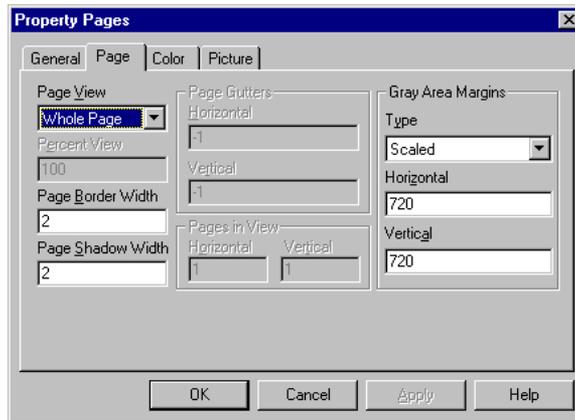
The following items are available on the General property page:

Item on Property Page	Description	Corresponding Property
Allow Zoom with Mouse check box	Determines whether the user can use the mouse to zoom in or out while viewing pages	AllowUserZoom
Enabled check box	Determines whether the control will respond to events	Enabled
Border Style drop-down list box	Specifies the style of the border of the control	BorderStyle
Mouse Pointer drop-down list box	Determines the pointer displayed while over the control	MousePointer
Scroll Bars— Horizontal drop-down list box	Determines whether and how the horizontal scroll bars are displayed	ScrollBarH

Item on Property Page	Description	Corresponding Property
Scroll Bars— Increment box	Specifies the horizontal distance the control scrolls when the user scrolls through the displayed pages	ScrollIncH
Scroll Bars— Vertical drop-down list box	Determines whether and how the vertical scroll bars are displayed	ScrollBarV
Scroll Bars— Increment box	Specifies the vertical distance the control scrolls when the user scrolls through the displayed pages	ScrollIncV

## Page Property Page

The Page property page lets you customize the settings for the page.



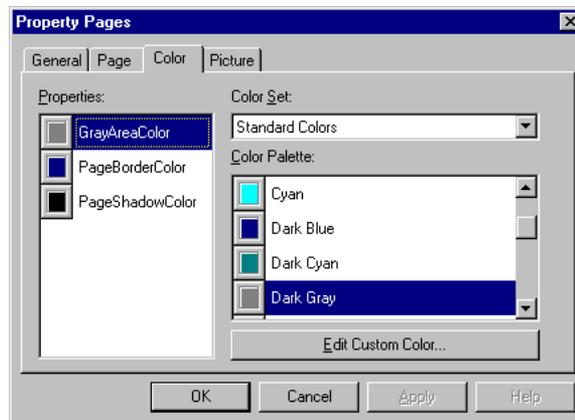
The following items are available on the Page property page:

Item on Property Page	Description	Corresponding Property
Page View drop-down	Determines how pages are displayed	PageViewType
Percent View box	Specifies the percent size of the displayed pages	PageViewPercentage
Page Border Width box	Specifies the border width of the displayed pages	PageBorderWidth
Page Shadow Width box	Specifies the width of the shadow behind the displayed pages	PageShadowWidth

Item on Property Page	Description	Corresponding Property
Page Gutters— Horizontal box	Specifies the horizontal space in twips between displayed pages	PageGutterH
Page Gutters— Vertical box	Specifies the vertical space in twips between displayed pages	PageGutterV
Pages in View— Horizontal box	Specifies the number of pages displayed in the horizontal direction	PageMultiCntH
Pages in View— Vertical box	Specifies the number of pages displayed in the vertical direction	PageMultiCntV
Gray Area Margins— Type drop-down list box	Determines whether the control uses the actual gray area margin settings or scales the settings to optimize the page view	GrayAreaMarginType
Gray Area Margins— Horizontal box	Specifies the horizontal space between the displayed page and the edge of the control	GrayAreaMarginH
Gray Area Margins— Vertical box	Specifies the vertical space between the displayed page and the edge of the control	GrayAreaMarginV

## Color Property Page

The Color property page lets you set the gray area color, the border color around the page, and the shadow color around the page.

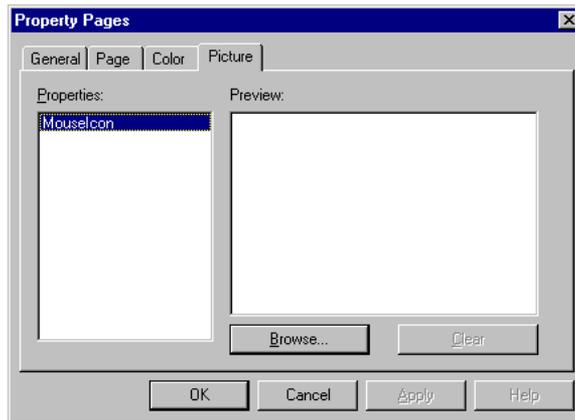


The following items are available on the Color property page:

Item on Property Page	Description	Corresponding Property
Properties list box	Lists the available color properties you can set	<b>GrayAreaColor</b> <b>PageBorderColor</b> <b>PageShadowColor</b>
Color Set drop-down list box	Lets you select Standard Colors or Windows System Colors	
Color Palette list box	Lets you select one of the available colors as the value for the specified color properties	

## Picture Property Page

The Picture property page lets you specify the picture to display for the mouse pointer.



The following items are available on the Picture property page:

Item on Property Page	Description	Corresponding Property
Properties list box	Specify the icon or bitmap for the pointer	MouseIcon
Preview area	Displays selected graphics file	N/A
Browse button	Displays a file selection dialog box from which you can select a graphics file	N/A

# Chapter 4 Using the Spread Designer

You can use the Spread Designer at design time to design the look of the fpSpread control. You can set both design-time and run-time properties. By setting properties at design time instead of run time in Spread Designer, you can view the run-time appearance of the spreadsheet.

The Spread Designer creates a “snapshot” of the fpSpread control. Once all changes are made, you apply those changes to the selected fpSpread control on your form or dialog.

---

**Note** You can run the Spread Designer as a stand-alone program to create spreadsheet, HTML, or Excel files. However, you cannot set data-binding properties when running the Spread Designer as a stand-alone program.

---

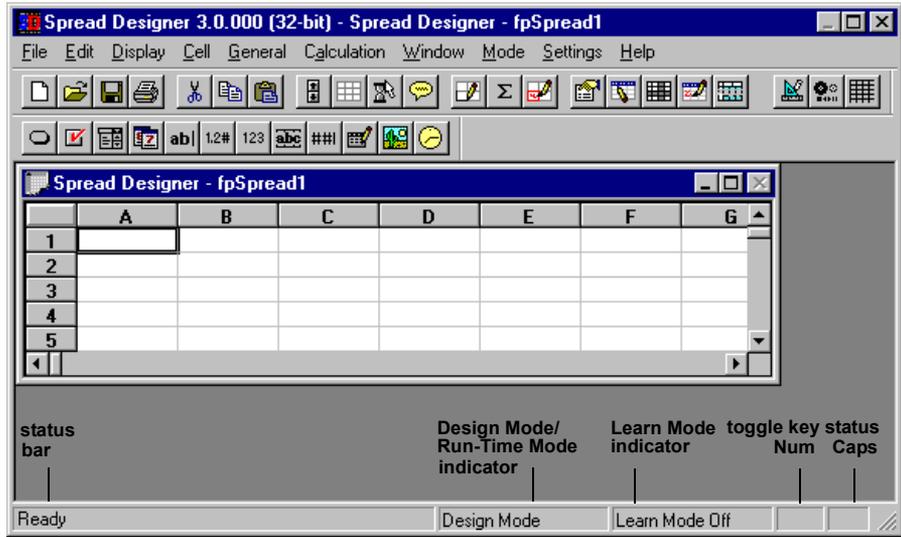
## Starting the Spread Designer

Before you use the Spread Designer, you should be familiar with the Spread features described in Chapter 2, “The fpSpread Control.”

► **To start the Spread Designer**

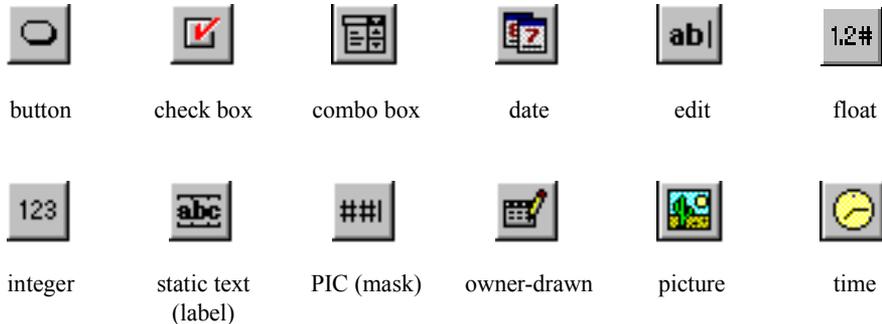
1. Place an fpSpread control on the form or dialog.
  - a. If you are using the ActiveX control,
    - i. Display the pop-up menu by clicking the fpSpread control with the right mouse button.
    - ii. Choose Spread Designer.
  - b. If you are using the VBX control, double-click the **InterfaceDesigner** property in the Properties window or dialog box.
2. If you want to run the Spread Designer as a stand-alone program, type:  
`[drive]:\[directory]\SD32D30.EXE`  
if you are using a 32-bit development environment, or type:  
`[drive]:\[directory]\SD16D30.EXE`  
if you are using a 16-bit development environment.

When you start the Spread Designer, the Spread Designer appears, as illustrated in the following figure.



You can quickly access most of the drop-down menu choices by using the toolbar buttons. The following table describes the toolbar buttons:

					
scroll bars	grid lines	pointers	text tip	border lines	formulas
					
cell colors	spreadsheet environment	headers	columns and rows	spreadsheet colors	virtual mode
					
run-time mode (off)	run-time mode (on)	learn mode (off)	learn mode (on)	cell type mode (off)	cell type mode (on)



For a complete listing and descriptions of the buttons and menu items available, see Appendix F, “Spread Designer Cross Reference.”

## Selecting Spreadsheet Elements

Many spreadsheet properties, functions, and methods perform actions on a cell, a range of cells, a column, a row, or the entire spreadsheet. Before using these properties, functions, and methods, you must often select or specify the element you for which you want to specify characteristics.

The following table shows how to select spreadsheet elements in Spread Designer.

<b>To select . . .</b>	<b>Do this . . .</b>
The entire spreadsheet	Click the top-left header cell (the cell in the upper-left corner of the spreadsheet)
An entire row	Click the row header for that row
An entire column	Click the column header for that column
The specified cell	Click the cell
Block of cells (in run-time mode)	<p>Allow block selections:            From the General menu choose Spreadsheet Environment. Choose the BlockMode tab. Select the Blocks check box under Settings.</p> <p>To make the selection:            Move the pointer over the cell to begin the block, press the left mouse button, then drag the pointer to select the block. Release the left mouse button when you have selected the block you want.</p>

**To select . . .**

Noncontiguous blocks of cells (in run-time mode)

**Do this . . .**

Allow noncontiguous block selections:  
From the General menu choose Spreadsheet Environment. Choose the General tab. Select the Multiple Block Selections check box under Settings.

To make selections:  
Move the pointer over the cell to begin the block, press the left mouse button, then drag the pointer to select the block. Release the left mouse button when you have selected the block you want. To add noncontiguous blocks, press the Ctrl key while selecting additional blocks.

For more information about run-time mode, see the following section.

## Design Mode Versus Run-Time Mode

In the Spread Designer you can be in either design mode or run-time mode. By default, when you start the Spread Designer you are operating in design mode.

In design mode, the “design” spreadsheet displays in a format where certain properties are *temporarily* reset to values that optimize the spreadsheet design process. For example, if you hide a row, it will still display in the Spread Designer and you can still work with that row.

The following table shows the properties that are temporarily reset while in design mode.

<b>Property</b>	<b>Setting while in design mode</b>
AllowDragDrop	True
AllowMultiBlocks	True
AllowUserFormulas	True
AutoClipboard	True
DisplayColHeaders	True
DisplayRowHeaders	True
Lock	True
MoveActiveOnFocus	True
OperationMode	SS_OP_MODE_NORMAL
Protect	False
RetainSelBlock	True
ScrollBars	SS_SCROLLBAR_BOTH

Property	Setting while in design mode
SelectBlockOptions	SS_SELBLOCKOPT_BLOCKS   SS_SELBLOCKOPT_COLS   SS_SELBLOCKOPT_ROWS   SS_SELBLOCKOPT_ALL
UserResize	SS_USER_RESIZE_COL   SS_USER_RESIZE_ROW
UserResizeCol	SS_USER_RESIZE_DEFAULT
UserResizeRow	SS_USER_RESIZE_DEFAULT
VisibleCols	0
VisibleRows	0

In run-time mode, the “design” spreadsheet displays in a format where all properties you have set using the Spread Designer, both design-time and run-time, are applied as they are currently set. For example, if you hide a row, it will not display in the Spread Designer and you cannot work with that row.

You can switch between design and run-time modes. When you switch to run-time mode, the spreadsheet reflects the original property settings. Design mode does not permanently change the property values you have set.

#### ► To switch between design mode and run-time mode

Do one of the following:

- From the Mode menu, choose Design Mode or Run-Time Mode.
- Click the run-time button () or the design-time button () on the toolbar.

This toggle button switches the Spread Designer between design mode and run-time mode.

## Using Learn Mode

In other programs you might have used a macro recorder to record a series of actions and create the macro code to let you repeat those actions. The Spread Designer offers a similar feature that records the settings and actions you have made in the Spread Designer and creates the code for those settings and actions.

Spread Designer’s *learn mode* generates Visual Basic code and writes that code to the Clipboard. You can view the code on the Clipboard and you can paste the code into your project. You might want to use learn mode to generate code for an unfamiliar property or to test a new application.

---

**Note** Learn mode generates code only for the settings made through the Spread Designer dialogs. Learn mode does not record actions such as resizing columns using the mouse or typing text in a cell.

---

**Tip** The information that is recorded during learn mode is stored in an internal buffer while learn mode is active. When you exit learn mode, the buffer immediately places the information on the system Clipboard. You should move the information on the Clipboard to your new application or another file as soon as possible to avoid accidentally overwriting it.

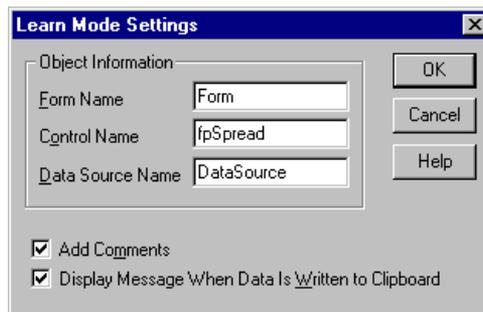
---

Before you use learn mode, you need to provide the Spread Designer with the name of the form, the name of the control, and the name of the data source, if one is available. This information provides the names for the objects in the generated code. In addition, you can specify whether or not you want comments in the generated code and whether you want to be notified when the code is sent to the Clipboard.

► **To set up and use learn mode**

1. Start the Spread Designer.
2. From the Settings menu, choose Learn Mode.

The Learn Mode Settings dialog box appears.



3. Under Object Information,
  - a. Type the name of the form containing the fpSpread control in the Form Name box.
  - b. Type the name of the fpSpread control in the Control Name box.
  - c. If the fpSpread control is data bound, type the name of the data control in the Data Source Name box.

4. If you do not want comments in the generated code, clear the Add Comments check box.
5. If you do not want a message displayed to confirm when the code is written to the Clipboard, clear the Display Message When Data Is Written to Clipboard check box.
6. Choose the OK button.
7. To start learn mode, do one of the following:
  - From the Mode menu, choose Learn Mode On.
  - Click the learn mode button (  ) on the toolbar.  
This toggle button switches learn mode on and off.
8. Perform actions using the Spread Designer.
9. Turn learn mode off by doing one of the following:
  - From the Mode menu, choose Learn Mode Off
  - Click the learn mode button (  ) on the toolbar.

If you chose to display a confirmation message in Step 5, it appears.



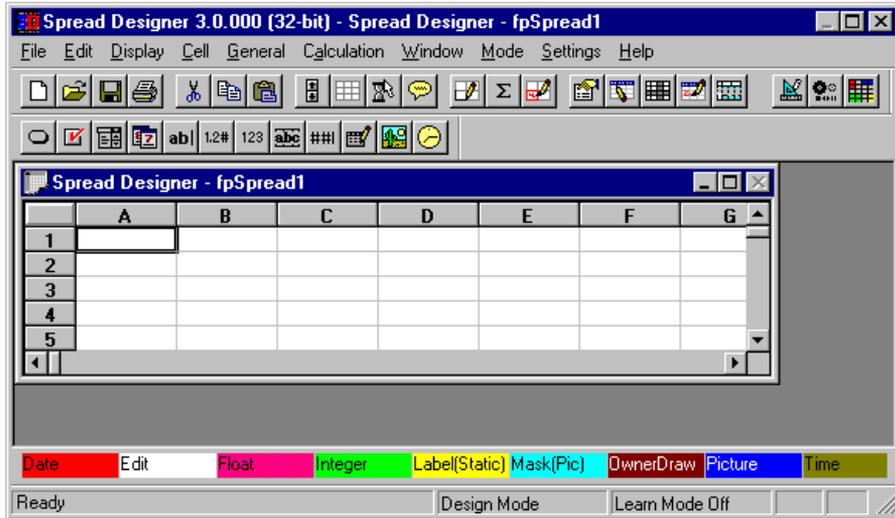
10. Choose the OK button.
11. To view the generated code, from the Window menu, choose Clipboard Viewer.
12. To paste the code from the Clipboard in your application, do the following:
  - a. Click an insertion point in your application where you want to paste the code.
  - b. Press Ctrl+V to paste the code.

## Using Cell Type Mode

The Spread Designer provides *cell type mode* to visually indicate which cells are assigned which cell type. In cell type mode, the date, edit, float, integer, owner-drawn, PIC (mask), picture, static text (label), and time cell types are each indicated by a different background

color. (No colors are displayed for the button, check box, and combo box cell types because their type is self-evident.)

In cell type mode, the cell type bar appears above the status bar, and previously set date, edit, float, integer, owner-drawn, PIC (mask), picture, static text (label), and time cell types are displayed with the appropriate background colors, as illustrated in the following figure.



The spreadsheet does not display the cell background colors in cell type mode; the colors are temporarily overridden.

---

**Note** When the Spread Designer is in cell type mode, you cannot save a spreadsheet, apply the changes back to the original, selected control, or set cell colors. If you attempt to perform any of these actions, the spreadsheet prompts you to exit cell type mode.

---

► **To use cell type mode**

1. Start the Spread Designer.
2. To start cell type mode, do one of the following:
  - From the Mode menu, choose Cell Type Mode On.
  - Click the cell type mode button (  ) on the feature toolbar.  
This toggle button switches cell type mode on and off.

3. When you are ready to exit cell type mode, do one of the following:
  - From the Mode menu, choose Cell Type Mode Off.
  - Click the cell type mode button () on the feature toolbar.

## Opening Multiple Spreadsheets

You can open more than one spreadsheet at a time in the Spread Designer. You can open files that have .SS2 (binary), .SS3 (binary), .TB2 (tab-delimited), .TB3 (tab-delimited), and .XLS (Excel 97) filename extensions. If the file has any other filename extension, it will open as a tab-delimited file.

---

**Tip** If you have saved files using the fpSpread control or other applications to files with different extensions, rename the files using the appropriate default extension to open the files correctly using the Spread Designer.

---

### Notes

- Keep in mind the following if you are using data binding:
    - If you start the Spread Designer from the command line, the Spread Designer does not recognize any spreadsheet as being data bound.
    - The Spread Designer only recognizes the Spread Designer view as being data bound (see “Spread Designer View Versus Active View” on page 58).
  - Spread version 3.0 can open and read .SS2 files that were created using previous versions of Spread, but all Spread version 3.0 binary files are now saved with the .SS3 extension.
- 

### ► To open multiple spreadsheets

1. If you want to open an Excel file and you want to create a log file that records the manner in which Spread handles the import of Excel features,
  - a. From the Settings menu, choose General and select the Create a Log File when Importing/Exporting Files.
  - b. Type the path and filename in the Create a Log File when Importing/Exporting Files box.
2. From the File menu, choose Open.
3. In the Open dialog box, specify the path and filename and choose the Open button. The spreadsheet appears in the Spread Designer window.

## Spread Designer View Versus Active View

When you select a fpSpread control and start the Spread Designer, the spreadsheet view displayed is the *Spread Designer view*. The title bar of the Spread Designer view reads “Spread Designer - *control name*”. The Spread Designer creates a “snapshot” of the selected fpSpread control.

If you have more than one spreadsheet open in the Spread Designer, the spreadsheet that has the focus is called the *active view*. Therefore, the Spread Designer view can be the active view, but the active view does not have to be the Spread Designer view.

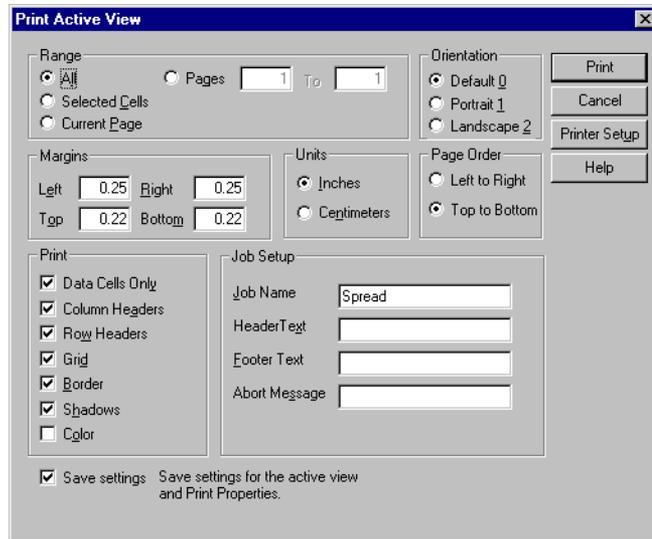
You can apply the changes from either the Spread Designer view or the active view to the selected fpSpread control (on the form). For more information on applying changes made in the Spread Designer, see “Saving and Applying Changes to the Spreadsheet” on page 67 and “Customizing How Changes Are Applied” on page 67.

To set data-binding properties, the Spread Designer view must be the active view and the fpSpread control must be linked to a data source control on the form. You cannot set data-binding properties for a spreadsheet that you open in the Spread Designer. For more information on using the Spread Designer to set data binding properties, see Chapter 12, “Binding to Databases.”

## Printing the Active View

In the Spread Designer, you can print the active view or you can set print properties to apply back to the spreadsheet. If you want to set print properties to apply to the spreadsheet, see Chapter 7, “Printing Spreadsheets,” for more information and instructions.

If you want to print the active view, from the File menu, choose Print, or press the Print button () in the toolbar. The Print Active View dialog box appears, as shown in the following figure.



You can customize how the active view is printed using the options provided in the dialog box.

By default, the active view will be printed one page at a time, from top to bottom, then left to right. The active view is printed on the current default printer in your Windows environment. If you want to print the active view on another printer, choose the Printer Setup button and select another printer configuration. You can print the active view on any Windows-supported printer.

By default, the first time you open the Print Active View dialog box, it loads the current settings of the corresponding options in the Print Properties dialog box. The Print Properties dialog, accessed from the General menu, is designed to allow you to set print properties to apply to the spreadsheet. Subsequent times when you open the Print Active View dialog box, it loads either the current settings from the Print Properties dialog box or the saved settings from changes you have made and saved.

You can specify that the settings you choose for the active view are saved, instead of being overridden by the settings from the Print Properties dialog box or from the previous saved settings. To do so, select the Save Settings check box. When the Save Settings check box is selected, the settings in the Print Active View dialog are saved; you can then use the same settings to print the view again. These settings are also saved to the Print Properties dialog accessed from the General menu. When you clear the Save Settings check box, the Print Active View dialog does not save the current settings.

For instructions on setting print properties using the Print Properties dialog box and printing the spreadsheet, see Chapter 7, “Printing Spreadsheets.”

### ► **To print the active view**

In the Spread Designer in the active view,

1. From the File menu, choose Print, or press the Print button () in the toolbar to display the Print Active View dialog box.

---

**Note** If this is the first time you have opened this dialog box, the settings displayed are the settings from the Print Properties dialog box.

---

2. In the Print Active View dialog box,
  - a. Prepare the active view for printing by specifying print options and the portion of the view to print using the instructions in the following topics:
    - “Specifying Print Options” on page 60
    - “Printing the Portion of the View with Data” on page 62
    - “Specifying a Print Range” on page 62
    - “Specifying Print Page Breaks” on page 63
    - “Adding Page Headers and Footers to Printed Pages” on page 64
  - b. If you want to save these settings to use the next time you print this view, select the Save Settings check box.

This also saves print settings to the Print Properties dialog accessed from the General menu. In the Print Properties dialog, you can apply the print settings back to the control.

- c. If you want to change to a different printer other than the default printer or change other printer setup options, choose the Printer Setup button and configure your printer in the Print Setup dialog box.

When you are finished, choose the OK button.
- d. Choose the Print button.

## **Specifying Print Options**

Before printing the active view, you might want to specify print options such as margins and page orientation.

---

**Note** These settings are not applied to the spreadsheet, but are used to print the active view. These settings do not necessarily affect the print settings for the spreadsheet. For instructions on setting print properties and printing the spreadsheet, see Chapter 7, “Printing Spreadsheets.”

---

► **To specify print options**

In the Spread Designer in the active view,

1. From the File menu, choose Print, or press the Print button () in the toolbar to display the Print Active View dialog box.
2. In the Print Active View dialog box,
  - a. Under Orientation, select an option button to specify the page orientation used to print the view.
  - b. Under Page Order, select an option button to specify the order in which pages are printed.
  - c. Specify the margins for the printed view.
    - i. Under Units, select an option button to specify the unit type used for calculating the margins.
    - ii. Under Margins, provide the size of the margins in the Left, Right, Top, and Bottom boxes.
  - d. Under Print, specify which features are printed by setting or clearing the following items.
    - i. If you do not want to print the column header row, clear the Column Headers check box.
    - ii. If you do not want to print the row header column, clear the Row Headers check box.
    - iii. If you do not want to print the grid lines, clear the Grid check box.
    - iv. If you do not want to print the border, clear the Border check box.
    - v. If you do not want to print the shadow effect within the headers, clear the Shadows check box.  
  
Note that if you clear the Column Headers and Row Headers check boxes, the Shadows setting does not have an effect.
    - vi. If you want to print the view’s colors as they appear on the screen, select the Color check box.

- e. Under Job Setup, specify job settings for the print job by setting the following items.
  - i. Specify the print job name to display in the Print Manager when printing the view by typing the name in the Job Name box.
  - ii. Specify the text to display in an abort dialog box during printing by typing the message in the Abort Message box.

### Printing the Portion of the View with Data

You can specify that only the view's columns and rows that contain data are printed.

You can also set a specific block of cells or range of pages to print. For more information about setting a print range, see "Specifying a Print Range" on page 62.

---

#### Notes

- When printing a view that is using virtual mode, set the **PrintUseDataMax** property to False.
  - These settings are not applied to the spreadsheet, but are used to print the active view. These settings do not necessarily affect the print settings for the spreadsheet. For instructions on setting print properties and printing the spreadsheet, see Chapter 7, "Printing Spreadsheets."
- 

#### ► To print the portion of the view with data

In the Spread Designer in the active view,

1. From the File menu, choose Print, or press the Print button () in the toolbar to display the Print Active View dialog box.
2. In the Print Active View dialog box under Print, select the Data Cells Only check box.

### Specifying a Print Range

You can specify either a block of cells or a range of pages as a print range.

You can also specify that only the active view's columns and rows that contain data are printed. For more information, see "Printing the Portion of the View with Data" on page 62.

---

**Note** These settings are not applied to the spreadsheet, but are used to print the active view. These settings do not necessarily affect the print settings for the spreadsheet. For instructions on setting print properties and printing the spreadsheet, see Chapter 7, “Printing Spreadsheets.”

---

▶ **To specify a block of cells as a print range**

In the Spread Designer in the active view,

1. Select the block of cells you want to print.
2. From the File menu, choose Print, or press the Print button () in the toolbar to display the Print Active View dialog box.
3. In the Print Active View dialog box,
  - a. Under Print, clear the Data Cells Only check box.
  - b. Under Range, select the Selected Cells option button.

▶ **To specify a range of pages as a print range**

In the Spread Designer in the active view,

1. From the File menu, choose Print, or press the Print button () in the toolbar to display the Print Active View dialog box.
2. In the Print Active View dialog box,
  - a. Under Print, clear the Data Cells Only check box.
  - b. Under Range,
    - i. Select the Pages option button.
    - ii. Enter the range of pages in the Pages boxes.

## Specifying Print Page Breaks

You can set a print page break before a specified column or row. Page breaks are not displayed on the screen, but force page breaks when you print the view. A column page break occurs to the left of the specified column; a row page break occurs above the specified row.

---

**Note** These settings are not applied to the spreadsheet, unless you apply the active view to the spreadsheet. For more information about applying views to the spreadsheet, see “Customizing How Changes Are Applied” on page 67.

---

### ► To specify a print page break before a column or row

In the Spread Designer in the active view,

1. If you want to specify a print page break before a column,
  - a. Select the column by clicking the column header or selecting a cell in the column.
  - b. From the General menu, choose Columns and Rows or press the Columns and Rows button () in the tool bar.
  - c. In the Columns / Rows dialog box on the Columns tab, select the Column Page Break check box.
  - d. Choose the OK or Apply button.
2. If you want to specify a print page break before a row,
  - a. Select the row by clicking the row header or selecting a cell in the row.
  - b. From the General menu, choose Columns and Rows or press the Columns and Rows button () in the tool bar.
  - c. In the Columns / Rows dialog box on the Rows tab, select the Row Page Break check box.
  - d. Choose the OK or Apply button.

## Adding Page Headers and Footers to Printed Pages

You can print page headers or footers for your spreadsheet. You can use control characters to customize the headers and footers, including alignment and font.

---

**Note** These settings are not applied to the spreadsheet, but are used to print the active view. These settings do not necessarily affect the print settings for the spreadsheet. For instructions on setting print properties and printing the spreadsheet, see Chapter 7, “Printing Spreadsheets.”

---

### ► To add header or footer text to printed pages

In the Spread Designer in the active view,

1. From the File menu, choose Print, or press the Print button () in the toolbar to display the Print Active View dialog box.
2. In the Print Active View dialog box under Job Setup, type the header and footer text in the Header Text and Footer Text boxes.

You can use control characters to specify header and footer format. For more information, refer to the **PrintHeader** and **PrintFooter** properties in the online *ActiveX/VBX Reference Guide*.

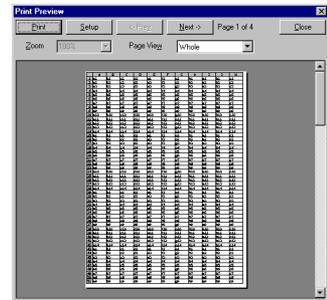
## Previewing the Printed Spreadsheet in Spread Designer

You can preview how the active view will print in Spread Designer. You can also print the active view from the Print Preview dialog.

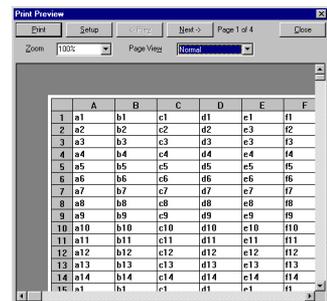
You can view the spreadsheet in four ways.

Page View	Description
Whole	Displays one whole page

### Representation

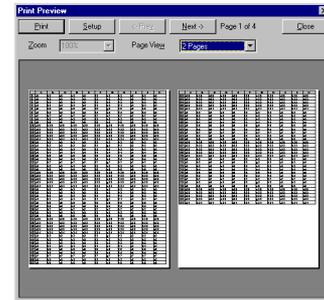


Normal	Displays a page the size of the fpSpread control. You can display the spreadsheet from 25%–300% of its original size.
--------	---

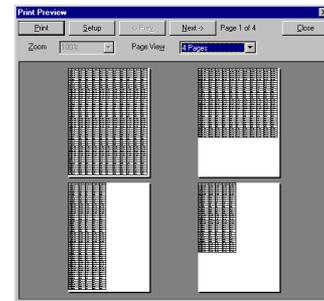


Page View	Description	Representation
-----------	-------------	----------------

2 Pages	Displays 2 pages at a time if the spreadsheet contains 2 or more pages
---------	--



4 Pages	Displays 4 pages at a time if the spreadsheet contains 4 or more pages
---------	--



### ► To preview the printed spreadsheet

In the Spread Designer in the active view,

1. From the File menu, choose Print Preview.
2. Select the view type from the Page View drop-down list box.
3. If you chose Normal in step 2 and you want to zoom in or out on the page, select a percentage from the Zoom drop-down list box.
4. Choose the Setup button to select a printer.
5. Choose the Print button to print the active view.

The spreadsheet will print using the settings from the Print Active View dialog. From the File menu, choose Print, or press the Print button () in the toolbar to display the Print Active View dialog box. For more information about changing these settings, see “Printing the Active View” on page 58.

## Saving and Applying Changes to the Spreadsheet

When you are satisfied with your spreadsheet design you can

- Save your changes to a spreadsheet file that you can open in the Spread Designer and use as a template.
- Save your changes to a spreadsheet file that you can load.
- Apply your changes back to the fpSpread control and continue working in the Spread Designer.

The way the available views are applied to the control is determined by settings you choose in the Spread Designer.

- Apply your changes and exit the Spread Designer.

---

**Caution** If you do not save or apply your changes, all your changes will be lost.

---

### Notes

- If you have more than one fpSpread control on your form or dialog, the changes are only applied back to the original, selected fpSpread control.
  - The changes you apply back to the original, selected fpSpread control are not permanent until you save the Form file.
  - If you exit the Spread Designer without choosing Save or Apply first, you might or might not be prompted to apply your changes, depending on how you have set up your Spread Designer environment. For more information, see “Customizing How Changes Are Applied” on page 67.
- 

## Customizing How Changes Are Applied

When applying changes, you can specify whether to apply the Spread Designer view or the active view to the selected fpSpread control. If you choose to apply the active view, you can do so with or without verification.

If you exit the Spread Designer without choosing Save or Apply first, you can specify whether a dialog box appears that prompts you to apply your changes. When you exit the Spread Designer without doing a Save or Apply and the dialog box appears prompting you to apply your changes, you can specify whether format, data, or both are applied.

---

**Caution** If you do not display a dialog box and you exit the Spread Designer without saving or applying your changes, all changes will be lost.

---

► **To customize how changes are applied**

1. If you want to specify the view applied to the selected control,
  - a. From the Settings menu, choose General.
  - b. Select an option button under View Applied to Control.
2. If you want to display a dialog box when exiting the Spread Designer without saving or applying changes,
  - a. From the Settings menu, choose General.
  - b. Select the Verify When Closing Window without Saving/Applying check box.
3. If you want to specify what is applied at exit verification,
  - a. From the Settings menu, choose General.
  - b. Select an option button under Type Applied by Exit Verification.

## Saving Changes to a File

You can save spreadsheet changes in one of the following formats:

- Format and data as a Spread file (.SS3)
- Data only in a tab-delimited file (.TB3)
- Format and data with a user-defined extension

---

### Notes

- The Spread version 3.0 ActiveX controls are safe for scripting. If you are using the Spread ActiveX control in a container that supports IObjectSafety such as Internet Explorer 4, you cannot export or save files.
  - All Spread version 3.0 binary files are now saved with the .SS3 extension. Spread version 3.0 can open and read .SS2 files that were created using previous versions of Spread, but can only save as a .SS3 file.
  - If you use a different filename extension, the file will still be saved as the file type chosen in the Save As Type drop-down list box in the Save As dialog box.
-

► **To save changes to a spreadsheet file**

Use one of the following methods to save changes:

- If you want to save format and data, from the File menu, choose Save As, and select .SS3 as the filename extension.
- If you want to save data only in a tab-delimited file, from the File menu, choose Save As, and select .TB3 as the filename extension.
- If you want to save format and data and use a different extension, type the filename in the Filename box.

---

**Note** If you use a different filename extension, the file will still be exported as the file type chosen in the Save As Type drop-down list box in the Save As dialog box.

---

## Exporting the Spreadsheet as an Excel or HTML File in Spread Designer

You can export a spreadsheet as an Excel spreadsheet (.XLS) or as a HTML file (.HTM).

Spreadsheet data and property settings are translated into Excel format as explained in Appendix I, “Excel-Formatted File Import/Export.” Spreadsheet data and property settings are translated into HTML as explained in Appendix J, “HTML File Export.”

---

### Notes

- The Spread version 3.0 ActiveX controls are safe for scripting. If you are using the Spread ActiveX control in a container that supports IObjectSafety such as Internet Explorer 4, you cannot export or save files.
  - Turn off virtual mode before exporting a spreadsheet to an HTML or Excel-formatted file.
- 

► **To export the spreadsheet as an Excel or HTML**

1. If you want to export format and data as an HTML file,
  - a. If you want to create a log file that records the manner in which Spread handles the export of HTML features,
    - i. From the Settings menu, choose General and select the Create a Log File when Importing/Exporting Files.
    - ii. Type the path and filename in the Create a Log File when Importing/Exporting Files box.

- b. From the File menu, choose Save As, and select .HTM as the filename extension.

If you want to export format and data and use a different extension, type the filename in the Filename box.

---

**Note** If you use a different filename extension, the file will still be exported as the file type chosen in the Save As Type drop-down list box in the Save As dialog box.

---

2. If you want to export format and data as an Excel file,
  - a. If you want to create a log file that records the manner in which Spread handles the export of Excel features,
    - i. From the Settings menu, choose General and select the Create a Log File when Importing/Exporting Files.
    - ii. Type the path and filename in the Create a Log File when Importing/Exporting Files box.
  - b. From the File menu, choose Save As, and select .XLS as the filename extension.

If you want to export format and data and use a different extension, type the filename in the Filename box.

---

**Note** If you use a different filename extension, the file will still be exported as the file type chosen in the Save As Type drop-down list box in the Save As dialog box.

---

- c. Choose OK.
- d. In the Save Excel File dialog box, type the name of the worksheet in the Enter the Sheet Name box.

If you do not type a name, Spread will name it Sheet1.

### Applying Changes to the Control

When you exit the Spread Designer, you can apply the format changes, the data changes, or both back to the selected fpSpread control.

► **To apply changes to the selected control**

1. If you want to apply the changes before you exit the Spread Designer,
  - a. To save format and data, from the File menu, choose Apply, All (Both Data and Format).
  - b. To save format only, from the File menu, choose Apply, Format.
  - c. To save data only, from the File menu, choose Apply, Data.
2. If you want to apply changes when you exit the Spread Designer, from the File menu, choose Apply & Exit or choose Exit Spread Designer.

---

**Caution** If you choose not to be prompted to apply your changes and you exit the Spread Designer without saving or applying your changes, all changes will be lost. For more information, see “Customizing How Changes Are Applied” on page 67.

---

## Customizing Balloon Help

By default, balloon help is displayed for the toolbar and cell type bar buttons. You can turn off the display of balloon help, if you prefer. If you choose to display the balloon help, you can customize the shape, tail type, and shadow characteristics of the balloon.

► **To customize balloon help**

1. From the Settings menu, choose General.
2. To hide the balloon help, clear the Display Balloons check box under Balloon Settings.
3. To change the balloon shape, select an option button under Balloon Settings, Shape.
4. To change the balloon tail, select an option button under Balloon Settings, Tail Type.
5. To change the shadow behind the balloon, select an option button under Balloon Settings, Shadow.



# Part II How-to Guides

The how-to guides provide step-by-step instructions for completing tasks using the Spread controls.

Instructions for completing tasks in Spread are given for using the following methods:

- The Spread Designer (marked “Spread Designer”)
- Your control type (marked “ActiveX”, “VBX”, “ActiveX, VBX”, or “DLL”)

Before reading these chapters, you should be familiar with the fpSpread and fpSpreadPreview control features described in Chapter 2, “The fpSpread Control,” Chapter 3, “The fpSpreadPreview Control,” and with the Spread Designer described in Chapter 4, “Using the Spread Designer.”



# Chapter 5 Working with Data

This chapter explains how to work with spreadsheet data, as described in the following topics:

<b>Topic</b>	<b>On Page</b>
Adding Data to a Spreadsheet	75
Loading an Existing Spreadsheet	81
Copying Data	84
Moving Data	91
Swapping Data	97
Deleting Data	98
Sorting Data	100
Entering Formulas	102
Returning Data from a Spreadsheet	110

## Adding Data to a Spreadsheet

You can add data to the spreadsheet, either to individual cells or to blocks of cells.

You can also load existing files created using the Spread Designer or other applications directly into the spreadsheet. For instructions, see “Loading an Existing Spreadsheet” on page 81.

You can add both formatted and unformatted data in the spreadsheet, depending on your needs. Formatted data usually includes information that denotes the context of the data. Unformatted data does not include additional information and might require a specific format to convey meaning. For example, formatted monetary data might include currency and separator characters to indicate monetary value, as in \$1,025.34. Unformatted data would not include the currency and separator characters, only the numeric value, as in 1025.34. For a more thorough discussion of formatted and unformatted data, see “Formatted and Unformatted Data” on page 30.

## Determining How to Add Your Data

You need to determine whether the data you are adding to the spreadsheet contains formatting and the layout of the data (for example, if it is in a tab-delimited array) to decide the best way to add the data to the spreadsheet.

For example, if you have numeric values in a set of data that you can copy, and the data is arranged with tab delimiters and carriage returns between “rows,” you can provide that data in code, adding to the control using the **ClipValue** property or the **SSClipValueIn** function.

Use the following table to determine the best way to add your data to the spreadsheet.

Type of Data	Spreadsheet Area	ActiveX/VBX Properties	ActiveX/VBX Methods or Functions	DLL Functions
Formatted	A cell, a column, a row, or the entire spreadsheet	<b>Text</b>	<b>SetArray, SetText</b>	<b>SSSetArray, SSSetData</b>
	A block of cells	<b>Clip, Text</b>	<b>SetArray</b>	<b>SSClipIn, SSSetArray, SSSetDataRange</b>
Unformatted	A cell, a column, a row, or the entire spreadsheet	<b>Value</b>	<b>SetFloat, SetInteger</b>	<b>SSSetFloat, SSSetInteger, SSSetValue</b>
	A block of cells	<b>ClipValue, Value</b>		<b>SSClipValueIn, SSSetFloatRange, SSSetIntegerRange, SSSetValueRange</b>

The **Clip** and **ClipValue** properties and the **SSClipIn** and **SSClipValueIn** functions work with ranges of cells, with tab characters separating columns and carriage return characters separating rows.

## Providing the Data

If you are providing formatted data, you should supply a formatted data string equivalent to the text representation being used in the spreadsheet for date, edit, float, integer, PIC, static text, time, and combo box cell data. For button cells, use 0 for the up state and 1 for the down state for formatted data. For check box cells, use 0 for the unchecked state, 1 for the checked state, and, in the case of a three-state check box, 2 for the indeterminate state, for a formatted data string.

If you are providing unformatted data, use the following notation for the listed cell types:

Cell Type	Notation
Date cells	“MMDDYYYY”
Time cells	“HHMMSS”, where “HH” is in 24-hour format
Float cells	Remove any currency and separator characters

Cell Type	Notation
Combo box cells	Use the index (base zero) of the item in the list
PIC cells	Remove all nonmask characters
Edit, integer, static text, button, check box, and owner-drawn cells	Use the same representation for both formatted and unformatted string data
Picture cells	No text representation

The **Clip**, **ClipValue**, **Text**, and **Value** properties, the **SetText** function or method, and the **SSClipIn**, **SSClipValueIn**, **SSSetData**, **SSSetDataRange**, **SSSetValue**, and **SSSetValueRange** functions provide data as string values. Alternatively, you can use the **SetArray** function to set values of the type you need.

You can provide data for float and integer cells as a floating-point or integer value. If you are using the DLL control and are adding data to one of the following cell types, refer to the listed functions or information in the online *DLL Reference Guide*:

Cell Type	Functions
Float	<b>SSSetFloat</b> , <b>SSSetFloatRange</b>
Integer	<b>SSSetInteger</b> , <b>SSSetIntegerRange</b>

If you are using the ActiveX or VBX control and are adding data to one of the following cell types, refer to the listed functions or information in the online *ActiveX/VBX Reference Guide*:

Cell Type	Functions or Methods
Float	<b>SetFloat</b>
Integer	<b>SetInteger</b>

## Adding Formatted Data

You can add formatted data to a specified spreadsheet cell, a column, a row, a block of cells, or the entire spreadsheet.

Supply data in the format in which it is to be displayed. For more information, refer to “Providing the Data” on page 76, the **Text** property in the online *ActiveX/VBX Reference Guide*, or the **SSSetData** function in the online *DLL Reference Guide*.

► **To add formatted data to a cell, a column, a row, or the entire spreadsheet**

ActiveX Either use the **Text** property or call the **SetArray** or **SetText** method. Note that calling the **SetText** method is faster than using the **Text** property.

- If you are using the **Text** property, at run time,
  1. Set the **Col** and **Row** properties as appropriate to specify the spreadsheet area to which you want to add data.

For more information about setting the **Col** and **Row** properties, see “Relationship of Col and Row Properties or Parameters” on page 31.
  2. Set the **Text** property to the data you want to add.
- Call the **SetText** method.
- If you want to add data to the entire spreadsheet from an array, call the **SetArray** method.

VBX Either use the **Text** property or call the **SpreadSetArray** or **SpreadSetText** function. Note that calling the **SpreadSetText** function is faster than using the **Text** property.

- If you are using the **Text** property, at run time,
  1. Set the **Col** and **Row** properties as appropriate to specify the spreadsheet area to which you want to add data.

For more information about setting the **Col** and **Row** properties, see “Relationship of Col and Row Properties or Parameters” on page 31.
  2. Set the **Text** property to the data you want to add.
- Call the **SpreadSetText** function.
- If you want to add data to the entire spreadsheet from an array, call the **SpreadSetArray** method.

DLL Call the **SSSetArray** or **SSSetData** function.

► **To add formatted data to a block of cells**

- ActiveX At run time, either
- Use properties
    1. Specify the block of cells to which you want to add data by setting the **Col**, **Row**, **Col2**, and **Row2** properties.  
You do not need to set the **BlockMode** property.
    2. Set the **Clip** property to the string of data you want to add.  
Separate data for each column with a tab character; separate each row with a carriage return character.
  - Call the **SetArray** method.
- VBX At run time, either
- Use properties.
    1. Specify the block of cells to which you want to add data by setting the **Col**, **Row**, **Col2**, and **Row2** properties.  
You do not need to set the **BlockMode** property.
    2. Set the **Clip** property to the string of data you want to add.  
Separate data for each column with a tab character; separate each row with a carriage return character.
  - Call the **SpreadSetArray** function.
- DLL Use one of the following methods:
- If you need to load tab-delimited, formatted data, call the **SSSetArray** or **SSClipIn** function.
  - If you need to load formatted data that is not tab-delimited, call the **SSSetDataRange** function.

## Adding Unformatted Data

You can add unformatted data to a specified spreadsheet cell, a column, a row, a block of cells, or the entire spreadsheet.

For more information about providing unformatted data, refer to “Providing the Data” on page 76, the **Value** property in the online *ActiveX/VBX Reference Guide*, or the **SSSetValue** function in the online *DLL Reference Guide*.

### ► To add unformatted data to a cell, a column, a row, or the entire spreadsheet

ActiveX, VBX

At run time,

- If you are adding float or integer data, use the **SetFloat** or **SetInteger** function or method.
- If you are adding other data,
  1. Set the **Col** and **Row** properties as appropriate to specify the spreadsheet area to which you want to add data.

For more information about setting the **Col** and **Row** properties, see “Relationship of Col and Row Properties or Parameters” on page 31.

2. Set the **Value** property to the data you want to add.

DLL

- If you are adding float or integer data, call the **SSSetFloat** or **SSSetInteger** function.
- If you are adding other data, call the **SSSetValue** function.

### ► To add unformatted data to a block of cells

ActiveX, VBX

At run time,

1. Set the **Col**, **Row**, **Col2**, and **Row2** properties to specify the block of cells to which you want to add data.

You do not need to set the **BlockMode** property.

2. Set the **ClipValue** property to the data you want to add.

Separate data for each column with a tab character; separate each row with a carriage return character.

DLL

1. If you need to load tab-delimited unformatted data, call the **SSClipValueIn** function.
2. If you need to load the same, unformatted data to each cell in the block,
  - If you are adding float or integer data, call the **SSSetFloatRange** or **SSSetIntegerRange** function.
  - If you are adding other data, call the **SSSetValueRange** function.

## Loading an Existing Spreadsheet

You can load an existing spreadsheet that is in either a binary file or a tab-delimited file. If you are using the fpSpread ActiveX or 32-bit DLL control, you can load a spreadsheet that was created in Microsoft® Excel™. DLL users can also load a spreadsheet from a buffer.

Binary files can contain all spreadsheet data and formatting or just spreadsheet data. Binary files are in a proprietary format that only the fpSpread control or the Spread Designer can read.

---

**Note** Spread version 3.0 can open and read binary (.SS2) files that were created using previous versions of Spread, but all Spread version 3.0 binary files are now saved with the .SS3 extension.

---

Tab-delimited files contain spreadsheet data separated by tabs and carriage returns. Tab-delimited files can be opened, modified, and saved using any standard text editor.

Excel-formatted files are files you have created in Excel, or another spreadsheet application, and exported to the Excel file format (usually .XLS or .WBK). Spread can load only spreadsheets created as Excel 97 spreadsheets (BIFF8 format). Spread does not support all the features offered in Excel 97 files. For information about the manner in which Spread deals with importing Excel-formatted files, including nonsupported features, see Appendix I, “Excel-Formatted File Import/Export.” Note that you cannot import a password-protected Excel-formatted file.

You can save spreadsheets from Spread to binary files or tab-delimited files. If you are using the fpSpread ActiveX or 32-bit DLL control, you can export spreadsheets to an Excel- or an HTML-formatted file. For instructions, see “Saving Spreadsheet Data” on page 115. By default, spreadsheet files created using the fpSpread control or the Spread Designer have the extension .SS3 (binary) or .TB3 (tab-delimited) or, if you are using a 32-bit control, .XLS (Excel format) or .HTM (HTML format). However, you can use any extension when you save or export from the fpSpread control or the Spread Designer.

For information and instructions about opening an existing spreadsheet using the Spread Designer, see “Opening Multiple Spreadsheets” on page 57.

► **To load an existing spreadsheet from a binary-formatted file**

- |                 |   |
|-----------------|---|
| Spread Designer | See “Opening Multiple Spreadsheets” on page 57. |
| ActiveX         | Call the <b>LoadFromFile</b> method.            |

## Chapter 5

VBX Call the **SpreadLoadFromFile** function.

DLL Call the **SSLoadFromFile** function.

▶ **To load an existing spreadsheet from a tab-delimited file**

ActiveX Call the **LoadTabFile** method.

VBX Call the **SpreadLoadTabFile** function.

DLL Call the **SSLoadTabFile** function.

▶ **To load an existing spreadsheet from an Excel-formatted file**

Spread Designer (32-bit only) See “Opening Multiple Spreadsheets” on page 57.

ActiveX (Visual Basic) 1. Call the **IsExcelFile** method to determine if the file is an Excel 97 file (in BIFF8 format).

If the file is an Excel 97 file, continue to step 2. If the file is not an Excel 97 file, Spread cannot import the file.

2. Call the **GetExcelSheetList** method, and set the parameters as follows:

- a. Set the value of the *FileName* parameter to the path and file name of the Excel-formatted spreadsheet.
- b. Declare the array to contain the list of sheet names. Declare the *VarArray* parameter as a string with the same number of elements as the total number of sheets in the workbook (if you know the number), as the following code illustrates:

```
Dim List(x) as String
```

where *x* is the total number of sheets in the workbook.

---

**Note** If you don’t know the total number of sheets, guess the total number, or use the instructions in the notes following these instructions to return the sheet names for every sheet in the workbook.

---

- c. Set the value of the *LogFileName* parameter to the path and file name of the log file that records the manner in which Spread handles the import of Excel features, including cell types and other settings. If you do not want to generate a log file, set the *LogFileName* parameter to 0.

- d. Set the value of the *Replace* parameter to True to read the entire Excel file.

Review the notes following these instructions for more information about the **GetExcelSheetList** function.

3. Call the **ImportExcelSheet** method, setting the parameters as follows:
  - a. Set the *WorkbookHandle* parameter to the value returned by the *WorkbookHandle* parameter from the **GetExcelSheetList** method.

The workbook handle is a global and unique identifier that represents an Excel file that has been opened during the import conversion process. For more information about the workbook handle, see the **GetExcelSheetList** method in the *Reference Guide*.

- b. Set the *Sheet* parameter to one of the following values:
  - The Worksheet number
 

The Worksheet number is zero-based and is an index in the array (*VarArray*) returned by the **GetExcelSheetList** method in step 2.
  - The Worksheet name
 

The Worksheet name is a name from the array (*VarArray*) returned by the **GetExcelSheetList** function in step 2.

---

## Notes

- The preceding instructions return the name of sheets in the workbook. If you know the number of sheets in the workbook, you can declare an array to contain all the sheets using the preceding instructions. If you do not know the number of sheets in the workbook, you can use the following instructions to return the sheet names for all the sheets in the workbook:
  1. Declare the *VarArray* parameter as a string with no Dimension (specifying it as dynamic), and then ReDim one element for the array, as the following code illustrates:
 

```
Dim List() as String
Redim List(1)
```
  2. Set the value of the *Replace* parameter to True to read the entire Excel file.
  3. After you have called the **GetExcelSheetList** method, check the value returned by the *ListCount* parameter. If the value is greater than one,
    - a. ReDim the *VarArray* parameter again, allocating the number of elements as the number of sheets for which you want information.
    - b. Call the **GetExcelSheetList** method, setting the *Replace* parameter to False.

- Set the *Replace* parameter to True in any case in which you want to force a Refresh of the information read from the Excel-formatted file. For example, you would want to force a Refresh any time you are importing a different Excel-formatted file to overwrite the earlier data read when importing the previous Excel-formatted file or if the Excel file has changed.

DLL  
(32-bit only)

1. Call the **SSIsExcelFile** function to determine if the file is in the Excel BIFF8 format.
2. Call the **SSGetExcelSheetList** function, and set the parameters as follows:
  - a. Set the value of the *lpzFileName* parameter to the path and filename of the Excel spreadsheet.
  - b. Set the value of the *lpzLogFile* parameter to the path and filename of the log file that records the manner in which Spread handles the import of Excel features, including cell types and other settings. If you do not want to generate a log file, set the *lpzLogFileName* parameter to NULL.
3. Call the **SSImportExcelSheet** function, and set the function parameters as follows.
  - a. If you want to use a Worksheet number,
    - i. Set the *sSheetNum* parameter to the Worksheet number.  
The Worksheet number is zero-based and is an index in the array (*lpghList*) returned by the **SSGetExcelSheetList** function in step 2.
    - ii. Set the *lpzSheetName* to NULL.
  - b. If you want to use a Worksheet name,
    - i. Set the *lpzSheetName* parameter to the Worksheet name.  
The Worksheet name is a name from the array (*lpghList*) returned by the **SSGetExcelSheetList** function in step 2.
    - ii. Set the *sSheetName* parameter to -1.

► **To load a spreadsheet from a buffer**

DLL Call the **SSLoadFromBuffer** function.

## Copying Data

You can copy data to and from cells using the familiar Clipboard method. Also, you can drag and drop data or use code to copy a selected block of cells. Choose the method that copies the data and formatting characteristics that you need to have copied, including formatting characters and cell formatting.

The following table summarizes the ways you can copy data and the formatting characteristics that are copied with the data.

Copy method		Formatting copied
<b>ActiveX or VBX control</b>	<b>DLL control</b>	
<b>Action</b> property setting 19 (Copy Range)	<b>SSCopyRange</b> function	Formatting characters (such as monetary symbols)  Cell settings (such as cell background color)
Drag and drop	Drag and drop	Formatting characters (such as monetary symbols)  Cell settings (such as cell background color)
Clipboard copy	Clipboard copy	Data is copied exactly as it appears in the cell, including formatting characters, if they are displayed.  Cell settings are not copied.

As listed in the preceding table, when you copy data from a cell or a range of cells, the settings for the cell are copied with the data, unless you are copying to the Clipboard. If you provided settings for the column or the row containing the cell, or the spreadsheet, but not the cell itself, those settings are not copied. For example, if you have set the source cell's background color to red, the background color is copied and the target cell will have a red background. However, if you have set the background color of the column containing the source cell to red, that setting is not copied. For more information, see "Formatted and Unformatted Data" on page 30 and "Selecting Spreadsheet Elements" on page 31.

When you copy data from one cell to another, the data from the cell you copied replaces the data in the cell you are pasting into. For example, if cell A1 contains the value 4 and you copy the value and paste it into cell B3, the contents of cell B3 are replaced by the value 4. Also, when you copy data from one block of cells to another, the data from the copied block replaces the values in the block you paste into.

## Chapter 5

If the copy operation copies a block of cells and then pastes it at an overlapping location, the values of all the cells of the block you are pasting to are replaced with the values of the cells in the copied block, as shown in the following figure.

A1	B1	C1
A2	B2	C2
A3	B3	C3

Copy cells A1 through B2

A1	B1	C1
A2	A1	B1
A3	A2	B2

Paste to cells B2 through C3  
(destination cell B2)

### Copying Data Using Code

#### ► To copy data using code

ActiveX, VBX

At run time,

1. Set the **Col**, **Row**, **Col2**, and **Row2** properties to specify the block of cells you want to copy.
2. Specify the upper-left cell of the destination block by setting the **DestCol** and **DestRow** properties.
3. Set the **Action** property to 19 (Copy Range).

DLL

Call the **SSCopyRange** function.

### Copying Data Using Drag-and-Drop

#### ► To copy data using drag and drop

Spread  
Designer

In the Spread Designer,

1. If you are working in design mode, go on to step 2.  
If you are working in run-time mode,
  - a. From the General menu, choose Spreadsheet Environment or press the Spreadsheet Environment button () in the toolbar.

- b. On the General tab in the Sheet Environment Settings dialog box, select the Cell Dragdrop check box.
  - c. Choose the OK or Apply button.
2. Select the cells or cells you want to copy using the mouse.
  3. Press and hold the Ctrl key.
  4. Position the pointer over the edge of the selection so that the arrow pointer appears.
  5. Press the left mouse button and drag the cell or cells.
  6. Release the left mouse button and the Ctrl key when the pointer is over the drop location.

---

**Note** When the Operation Mode is set to Row Mode (SheetMode tab in the Sheet Environment Settings dialog box), if you select the Cell Dragdrop check box and you drag and drop a cell or block of cells, the entire row or rows containing the selected cell or cells are dragged and dropped.

---

## ActiveX, VBX

1. Set the **AllowDragDrop** property to True.
2. In your application, select the cell or cells you want to copy using the mouse.
3. Press and hold the Ctrl key.
4. Position the mouse pointer over the edge of the selection so that the arrow pointer appears.
5. Press the left mouse button and drag the cell or cells.
6. Release the left mouse button when the mouse pointer is over the drop location.

---

**Note** When the **OperationMode** property is set to 2 (Row Mode), if you set the **AllowDragDrop** property to True and you drag and drop a cell or block of cells, the entire row or rows containing the selected cell or cells are dragged and dropped.

---

## DLL

1. Call the **SSSetBool** function, and set the value of the SSB\_ALLOWDRAGDROP boolean to TRUE.
2. In your application, select the cell or cells you want to copy using the mouse.
3. Press and hold the Ctrl key.
4. Position the mouse pointer over the edge of the selection so that the arrow pointer appears.

5. Press the left mouse button and drag the cell or cells.
6. Release the left mouse button when the mouse pointer is over the drop location.

---

**Note** When the *wMode* parameter of the **SSSetOperationMode** function is set to **SS\_OPMODE\_ROWMODE** and the **SSB\_ALLOWDRAGDROP** boolean of the **SSSetBool** function is set to **TRUE**, and you drag and drop a cell or block of cells, the entire row or rows containing the selected cell or cells are dragged and dropped.

---

## Copying Data Using the Clipboard

You can copy unformatted data with tab delimiters from the currently selected spreadsheet cell or block of cells to the Clipboard. Use the standard Clipboard keys to copy data to (Ctrl+C) and paste data from (Ctrl+V) the Clipboard, or use code to accomplish the same operations.

When pasting data from the Clipboard, if a block of cells is selected, the block is replaced with the Clipboard contents. If the contents of the Clipboard are bigger than the selected block, the contents of the Clipboard are truncated at the edge of the block.

If no cells are selected, the contents of the Clipboard are pasted into the spreadsheet starting at the active cell. The contents of the cells to the right of and below the active cell are replaced by the contents of the Clipboard.

By default, column and row headers are included when data is copied and pasted and the pasted headers overwrite the selected headers. You can specify that column and row headers are not included when data is copied and column and row headers are not overwritten when data is pasted.

### ► To copy data using the Clipboard

ActiveX, VBX

1. If you do not want to include the column or row headers when you copy data, and you do not want data pasted into the selected column or row headers, set the **ClipboardOptions** property to 0 (**SS\_CLIP\_NOHEADERS**).
2. If you want to include the row headers when you copy data, but you do not want data pasted into the selected row headers, set the **ClipboardOptions** property to 1 (**SS\_CLIP\_COPYROWHEADERS**).
3. If you do not want to include the row headers when you copy data, but you do want data to be pasted into the selected row headers, set the **ClipboardOptions** property to 2 (**SS\_CLIP\_PASTEROWHEADERS**).

4. If you want to include the column headers when you copy data, but you do not want data pasted into the selected column headers, set the **ClipboardOptions** property to 4 (SS\_CLIP\_COPYCOLHEADERS).
5. If you do not want to include the column headers when you copy data, but you do want data to be pasted into the selected column headers, set the **ClipboardOptions** property to 8 (SS\_CLIP\_PASTECOLHEADERS).
6. If you want to include column and row headers when you copy data and you want data to be pasted into the selected column and row headers, set the **ClipboardOptions** property to 15 (SS\_CLIP\_COPYPASTEALLHEADERS).
7. If you want a combination of copy and pasting column and row headers, you can Or the 1 (SS\_CLIP\_COPYROWHEADERS), 2 (SS\_CLIP\_PASTEROWHEADERS), 4 (SS\_CLIP\_COPYCOLHEADERS), and 8 (SS\_CLIP\_PASTECOLHEADERS) settings of the **ClipboardOptions** property as appropriate.

At design time, add the values for the settings and enter that numeric value to provide the following results:

<b>ClipboardOptions property setting</b>	<b>Copy row header</b>	<b>Paste row header</b>	<b>Copy column header</b>	<b>Paste column header</b>
3	Yes	Yes	No	No
5	Yes	No	Yes	No
6	No	Yes	Yes	No
7	Yes	Yes	Yes	No
9	Yes	No	No	Yes
10	No	Yes	No	Yes
11	Yes	Yes	No	Yes
12	No	No	Yes	Yes
13	Yes	No	Yes	Yes
14	No	Yes	Yes	Yes

---

**Note** These property settings do not have a constant (SS\_CLIP\_XXXX) associated with them. Just use the numeric Or value.

---

8. At run time,
  - a. Specify the cell or block of cells from which you want to copy data by setting the **Col**, **Row**, **Col2**, and **Row2** properties.
  - b. Set the **Action** property to 2 (Select Block).
  - c. Set the **Action** property to 22 (Clipboard Copy).

- d. Set the **Col**, **Col2**, **Row**, and **Row2** properties as appropriate to specify the cell or block of cells into which you want to paste data.
  - e. Set the **Action** property to either 0 (Activate Cell) or 2 (Select Block).
  - f. Set the **Action** property to 24 (Clipboard Paste).
- DLL
1. If you do not want to include the row headers when you copy data, and you do not want data pasted into the selected row headers, call the **SSSetClipboardOptions** function and set the value of the *wOptions* parameter of the function to **SS\_CLIP\_NOHEADERS**.
  2. If you want to include the row headers when you copy data, but you do not want data pasted into the selected row headers, call the **SSSetClipboardOptions** function and set the value of the *wOptions* parameter of the function to **SS\_CLIP\_COPYROWHEADERS**.
  3. If you do not want to include the row headers when you copy data, but you do want data to be pasted into the selected row headers, call the **SSSetClipboardOptions** function and set the value of the *wOptions* parameter of the function to **SS\_CLIP\_PASTEROWHEADERS**.
  4. If you want to include the column headers when you copy data, but you do not want data pasted into the selected column headers, call the **SSSetClipboardOptions** function and set the value of the *wOptions* parameter of the function to **SS\_CLIP\_COPYCOLHEADERS**.
  5. If you do not want to include the column headers when you copy data, but you do want data to be pasted into the selected column headers, call the **SSSetClipboardOptions** function and set the value of the *wOptions* parameter of the function to **SS\_CLIP\_PASTECOLHEADERS**.
  6. If you want to include column and row headers when you copy data and you want data to be pasted into the selected column and row headers, call the **SSSetClipboardOptions** function and set the value of the *wOptions* parameter of the function to **SS\_CLIP\_COPYPASTEALLHEADERS**.
  7. If you want a combination of copy and pasting column and row headers, call the **SSSetClipboardOptions** function and Or the appropriate **SS\_CLIP\_COPYROWHEADERS**, **SS\_CLIP\_PASTEROWHEADERS**, **SS\_CLIP\_COPYCOLHEADERS**, and **SS\_CLIP\_PASTECOLHEADERS** values of the *wOptions* parameter of the function.
  8. Call the **SSClipboardCopy** function.
  9. Call the **SSClipboardPaste** function.

## Moving Data

You can move data to and from cells using the familiar Clipboard method. Also, you can drag and drop data or use code to move a selected block of cells. You need to choose the method to move data that moves the data and formatting characteristics that you need to have moved, including formatting characters and cell formatting.

The following table summarizes the ways you can move data and the formatting characteristics that are moved with the data.

Move method		Formatting moved
<b>ActiveX or VBX control</b>	<b>DLL control</b>	
Action property setting 20 (Move Range)	SSMoveRange function	Formatting characters (such as monetary symbols)  Cell settings (such as cell background color)
Drag and drop	Drag and drop	Formatting characters (such as monetary symbols)  Cell settings (such as cell background color)
Clipboard cut	Clipboard cut	Data is moved exactly as it appears in the cell, including formatting characters, if they are displayed.  Cell settings are not moved.

As listed in the preceding table, when you move data from a cell or a range of cells, the settings for the cell are moved with the data, unless you are cutting to the Clipboard. If you provided settings for the column or the row containing the cell, or the spreadsheet, but not the cell itself, those settings are not moved. For example, if you have set the source cell's background color to red, the background color is moved and the target cell will have a red background. However, if you have set the background color of the column containing the source cell to red, that setting is not moved. For more information, see "Formatted and Unformatted Data" on page 30 and "Selecting Spreadsheet Elements" on page 31.

When you move data from one cell to another, the data from the first cell replaces the data in the cell you are moving data to. For example, if cell A1 contains the value 4 and you move the value and paste it into cell B3, the contents of cell B3 are replaced by the value 4.

When you move a block of data, the destination block is automatically sized based on the size of the block being moved. If data exists in the destination block, the existing data is replaced by the data in the block being moved.

## Chapter 5

If the move operation moves a block of cells to an overlapping location, the values of all the cells of the block you are moving to are replaced with the values of the cells in the moved block, as shown in the following figure.

A1	B1	C1
A2	B2	C2
A3	B3	C3

Move cells A1 through B2

		C1
	A1	B1
A3	A2	B2

Replace cells B2 through C3  
(destination cell B2)

When you move blocks of data, you can specify whether formulas are adjusted. For more information, see “Automatically Recalculating and Updating Formulas” on page 107.

### Moving Data Using Code

#### ► To move data using code

ActiveX, VBX

At run time,

1. Set the **Col**, **Row**, **Col2**, and **Row2** properties to specify the block of cells you want to move.
2. Specify the upper-left cell of the destination block by setting the **DestCol** and **DestRow** properties.
3. Set the **Action** property to 20 (Move Range).

DLL

Call the **SSMoveRange** function.

## Moving Data Using Drag and Drop

### ► To move data using drag and drop

Spread  
Designer

In the Spread Designer,

1. If you are working in design mode, go on to step 2.  
If you are working in run-time mode,
  - a. From the General menu, choose Spreadsheet Environment or press the Spreadsheet Environment button () in the toolbar.
  - b. On the General tab in the Sheet Environment Settings dialog box, select the Cell Dragdrop check box.
  - c. Choose the OK or Apply button.
2. Select the cells or cells you want to move using the mouse.
3. Position the pointer over the edge of the selection so that the arrow pointer appears.
4. Press the left mouse button and drag the cell or cells.
5. Release the left mouse button when the pointer is over the drop location.

---

**Note** When the Operation Mode is set to Row Mode (SheetMode tab in the Sheet Environment Settings dialog box), if you select the Cell Dragdrop check box and you drag and drop a cell or block of cells, the entire row or rows containing the selected cell or cells are dragged and dropped.

---

ActiveX, VBX

1. Set the **AllowDragDrop** property to True.
2. In your application, select the cell or cells you want to move using the mouse.
3. Position the mouse pointer over the edge of the selection so that the arrow pointer appears.
4. Press the left mouse button and drag the cell or cells.
5. Release the left mouse button when the mouse pointer is over the drop location.

---

**Note** When the **OperationMode** property is set to 2 (Row Mode), if you set the **AllowDragDrop** property to True and you drag and drop a cell or block of cells, the entire row or rows containing the selected cell or cells are dragged and dropped.

---

- DLL
1. Call the **SSSetBool** function, and set the value of the `SSB_ALLOWDRAGDROP` boolean to `TRUE`.
  2. In your application, select the cell or cells you want to move using the mouse.
  3. Position the mouse pointer over the edge of the selection so that the arrow pointer appears.
  4. Press the left mouse button and drag the cell or cells.
  5. Release the left mouse button when the mouse pointer is over the drop location.

---

**Note** When the *wMode* parameter of the **SSSetOperationMode** function is set to `SS_OPMODE_ROWMODE` and the `SSB_ALLOWDRAGDROP` boolean of the **SSSetBool** function is set to `TRUE`, and you drag and drop a cell or block of cells, the entire row or rows containing the selected cell or cells are dragged and dropped.

---

## Moving Data Using the Clipboard

You can cut unformatted data from a selected spreadsheet cell or block of cells to the Clipboard, and paste unformatted data from the Clipboard into a selected cell or block of cells. Use the standard Clipboard keys to move data to (`Ctrl+X`) and paste data from (`Ctrl+V`) the Clipboard.

When pasting data from the Clipboard, if a block of cells is selected, the block is replaced with the Clipboard contents. If the contents of the Clipboard are bigger than the selected block, the contents of the Clipboard are truncated at the edge of the block.

If no cells are selected, the contents of the Clipboard are pasted into the spreadsheet starting at the active cell. The contents of the cells to the right of and below the active cell are replaced by the contents of the Clipboard.

By default, column and row headers are included when data is cut and pasted and the pasted headers overwrite the selected headers. You can specify that column and row headers are not included when data is cut and column and row headers are not overwritten when data is pasted.

### ► To move data using the Clipboard

- ActiveX, VBX
1. If you do not want to include the row headers when you move data, and you do not want data pasted into the selected row headers, set the **ClipboardOptions** property to 0 (`SS_CLIP_NOHEADERS`).

2. If you want to include the row headers when you move data, but you do not want data pasted into the selected row headers, set the **ClipboardOptions** property to 1 (SS\_CLIP\_COPYROWHEADERS).
3. If you do not want to include the row headers when you move data, but you do want data to be pasted into the selected row headers, set the **ClipboardOptions** property to 2 (SS\_CLIP\_PASTEROWHEADERS).
4. If you want to include the column headers when you move data, but you do not want data pasted into the selected column headers, set the **ClipboardOptions** property to 4 (SS\_CLIP\_COPYCOLHEADERS).
5. If you do not want to include the column headers when you move data, but you do want data to be pasted into the selected column headers, set the **ClipboardOptions** property to 8 (SS\_CLIP\_PASTECOLHEADERS).
6. If you want a combination of moving column and row headers, you can Or the 1 (SS\_CLIP\_COPYROWHEADERS), 2 (SS\_CLIP\_PASTEROWHEADERS), 4 (SS\_CLIP\_COPYCOLHEADERS), and 8 (SS\_CLIP\_PASTECOLHEADERS) settings of the **ClipboardOptions** property as appropriate.

At design time, add the values for the settings and enter that numeric value to provide the following results:

<b>ClipboardOptions property setting</b>	<b>Cut row header</b>	<b>Paste row header</b>	<b>Cut column header</b>	<b>Paste column header</b>
3	Yes	Yes	No	No
5	Yes	No	Yes	No
6	No	Yes	Yes	No
7	Yes	Yes	Yes	No
9	Yes	No	No	Yes
10	No	Yes	No	Yes
11	Yes	Yes	No	Yes
12	No	No	Yes	Yes
13	Yes	No	Yes	Yes
14	No	Yes	Yes	Yes

---

**Note** These property settings do not have a constant (SS\_CLIP\_XXXXX) associated with them. Just use the numeric Or value.

---

7. At run time,
  - a. Set the **Col**, **Row**, **Col2**, and **Row2** properties as appropriate to specify the cell or block of cells from which you want to cut data.
  - b. Set the **Action** property to 2 (Select Block).
  - c. Set the **Action** property to 23 (Clipboard Cut).
  - d. Set the **Col**, **Col2**, **Row**, and **Row2** properties as appropriate to specify the cell or block of cells into which you want to paste data.
  - e. Set the **Action** property to either 0 (Activate Cell) or 2 (Select Block).
  - f. Set the **Action** property to 24 (Clipboard Paste).
  
- DLL
  1. If you do not want to include the row headers when you move data, and you do not want data pasted into the selected row headers, call the **SSSetClipboardOptions** function and set the value of the *wOptions* parameter of the function to `SS_CLIP_NOHEADERS`.
  2. If you want to include the row headers when you move data, but you do not want data pasted into the selected row headers, call the **SSSetClipboardOptions** function and set the value of the *wOptions* parameter of the function to `SS_CLIP_COPYROWHEADERS`.
  3. If you do not want to include the row headers when you move data, but you do want data to be pasted into the selected row headers, call the **SSSetClipboardOptions** function and set the value of the *wOptions* parameter of the function to `SS_CLIP_PASTEROWHEADERS`.
  4. If you want to include the column headers when you move data, but you do not want data pasted into the selected column headers, call the **SSSetClipboardOptions** function and set the value of the *wOptions* parameter of the function to `SS_CLIP_COPYCOLHEADERS`.
  5. If you do not want to include the column headers when you move data, but you do want data to be pasted into the selected column headers, call the **SSSetClipboardOptions** function and set the value of the *wOptions* parameter of the function to `SS_CLIP_PASTECOLHEADERS`.
  6. If you want to include column and row headers when you move data and you want data to be pasted into the selected column and row headers, call the **SSSetClipboardOptions** function and set the value of the *wOptions* parameter of the function to `SS_CLIP_COPYPASTEALLHEADERS`.

7. If you want a combination of cutting and pasting column and row headers, call the **SSSetClipboardOptions** function and Or the appropriate **SS\_CLIP\_COPYROWHEADERS**, **SS\_CLIP\_PASTEROWHEADERS**, **SS\_CLIP\_COPYCOLHEADERS**, and **SS\_CLIP\_PASTECOLHEADERS** values of the *wOptions* parameter of the function.
8. Call the **SSClipboardCut** function to cut the data.
9. Call the **SSClipboardPaste** function.

## Swapping Data

You can swap the contents of two cells or two blocks of cells.

When you swap data from a cell or a range of cells to another cell or range of cells, the settings for the cell are swapped along with the data. If you provided settings for the column or the row containing the cell, or the spreadsheet, but not the cell itself, those settings are not swapped. For example, if you have set the source cell's background color to red, the background color is swapped and the target cell will have a red background. However, if you have set the background color of the column containing the source cell to red, that setting is not swapped. For more information, see "Formatted and Unformatted Data" on page 30 and "Selecting Spreadsheet Elements" on page 31.

When you swap data from one cell to another, the data in one cell becomes the data in the other cell, and vice versa. For example, if cell A1 contains the value 4 and cell B3 contains the value 6 and you swap the values of the cells, the value of cell A1 becomes 6 and the value of cell B3 becomes 4.

If you attempt to swap a block that is larger than the available block at the destination, the swap operation is not performed. For example, if you attempt to swap a block of four cells and specify the destination as a cell at the edge of the spreadsheet, the swap does not take place.

## Chapter 5

If the swap operation swaps overlapping blocks of cells, individual cells are swapped starting at the overlapping corner. As shown in the following figures, this method maintains the spreadsheet's data.

A1	B1	C1
A2	B2	C2
A3	B3	C3

Swap selected cells A1 through B2 with cells B2 through C3

C3	C2	C1
B3	A1	B1
A3	A2	B2

Cells are swapped starting with the overlapping cell, thus maintaining the data

When you swap blocks of data, you can specify whether formulas are adjusted. For more information, see “Automatically Recalculating and Updating Formulas” on page 107.

### ► To swap formatted data between two blocks of cells

ActiveX, VBX

At run time,

1. Set the **Col**, **Row**, **Col2**, and **Row2** properties to specify the first block of cells you want to swap.
2. Specify the upper-left cell of the second block of cells you want to swap by setting the **DestCol** and **DestRow** properties.
3. Set the **Action** property to 21 (Swap Range).

DLL Call the **SSSwapRange** function.

## Deleting Data

You can delete both data and cell characteristics from a selected spreadsheet cell or block of cells, or delete only the data, leaving the cell characteristics, such as cell type, intact. For more information about cell characteristics, see “Formatted and Unformatted Data” on page 30 and “Selecting Spreadsheet Elements” on page 31.

Other actions you perform can delete spreadsheet data and cell, column, row, and spreadsheet characteristics. For more information, see “Resetting a Spreadsheet to its Default Settings” on page 127 and “Deleting Columns and Rows” on page 164.

► **To delete the data and cell characteristics from a cell or block of cells**

ActiveX, VBX

At run time,

1. Set the **Col**, **Row**, **Col2**, and **Row2** properties as appropriate to specify the cell or block of cells from which you want to delete the data and cell characteristics.
2. If you are selecting a block of cells, set the **BlockMode** property to True.
3. Set the **Action** property to 3 (Clear).
4. If you selected a block of cells, set the **BlockMode** property back to False.

DLL

1. If you want to delete the data and cell characteristics from a cell, call the **SSClear** function.
2. If you want to delete the data and cell characteristics from a block of cells, call the **SSClearRange** function.

► **To delete only the data in a cell or block of cells**

ActiveX, VBX

At run time,

1. Set the **Col**, **Row**, **Col2**, and **Row2** properties as appropriate to specify the cell or block of cells from which you want to delete data.
2. If you are selecting a block of cells, set the **BlockMode** property to True.
3. Set the **Action** property to 12 (Clear Text).
4. If you selected a block of cells, set the **BlockMode** property back to False.

DLL

1. If you want to delete the data from a cell, call the **SSClearData** function.
2. If you want to delete the data from a block of cells, call the **SSClearDataRange** function.

## Sorting Data

You can sort spreadsheet data using up to 256 sort keys.

To sort data, first specify the data you want to sort, then specify which column or row you want to sort by, the priority of that column or row in the sort (sort by that column or row first, second, third, and so on), and whether you want to sort in ascending or descending order.

When specifying the data you want to sort, refer to “Relationship of Col and Row Properties or Parameters” on page 31 for more information. For DLL controls, you can set the *Col* or *Row* parameter to 0 to perform a sort on the column or row header. For ActiveX and VBX controls, you can set the **Col** or **Row** properties to 0 to perform a sort on the column or row header.

---

### Notes

- You cannot sort a spreadsheet that is using virtual mode.
- Button, check box, combo box, and picture cells are sorted by the value of the **Text** property.
- For DLL users, if you sort by the column header and perform a sort by row, the row item data will move. If you sort by the row header and perform a sort by column, the column item data will move. Perform a sort on the column or row header by setting the **SSSort** function's *Col* (or *Row*) parameter to 0 and the *Col2* (or *Row2*) parameter to -1.

For ActiveX or VBX users, if you sort by the column header and perform a sort by row, the row item data will move. If you sort by the row header and perform a sort by column, the column item data will move. Perform a sort on the column or row header by setting the **Col** (or **Row**) property to 0 and the **Col2** (or **Row2**) property to -1.

- The spreadsheet does not update cell references in formulas when you sort blocks of cells in the spreadsheet.
- 

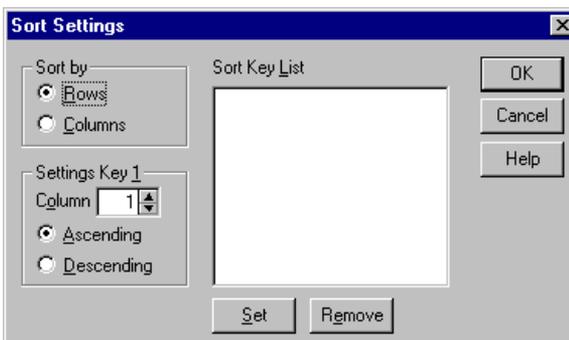
### ► To sort spreadsheet data

Spread  
Designer

In the Spread Designer,

1. Select the block of cells, a row, or column with the mouse you want to sort.
2. From the General menu, choose Spreadsheet Sort.

The Sort Settings dialog box appears:



3. In the Sort Settings dialog box,
  - a. Specify whether to sort by columns or by rows by selecting the appropriate option button under Sort by.
 

Sorting by rows sorts the rows by the values in the designated rows. Sorting by columns sorts the columns by the values in the designated columns.
  - b. For each sort key desired,
    - i. Specify the column number to sort by through setting the Column spin box under Settings Key 1.
    - ii. Specify whether to sort in ascending or descending order by selecting the appropriate option button under Settings Key 1 box.
    - iii. Choose the Set button.
 

The sort key appears in the Sort Key List.
  - c. Choose the OK button.

ActiveX, VBX

At run time,

1. Set the **Col**, **Row**, **Col2**, and **Row2** properties to specify the block of cells you want to sort.
2. Specify whether to sort by columns or by rows with the **SortBy** property.
 

Sorting by rows sorts the rows by the values in the designated rows. Sorting by columns sorts the columns by the values in the designated columns.

3. For each sort key,
    - a. Specify the column or row number to sort by and the sort priority for that column or row (the *sort key*) using the **SortKey** property. Set the *Index* parameter to specify the sort key, ranging from 1 to 256.
    - b. Specify whether to sort in ascending or descending order with the **SortKeyOrder** property. Set the *Index* parameter to specify the sort key, ranging from 1 to 256.
  4. Set the **Action** property to 25 (Sort).
- DLL
1. If you want to use three or fewer sort keys,
    - a. Call the **SSSort** function.
    - b. Use the **SS\_SORT** structure to specify sort keys and sort order.
  2. If you want to use more than three sort keys,
    - a. Call the **SSSortEx** function.
    - b. Use the **SS\_SORTKEY** structure to specify the sort keys and sort order.

## Entering Formulas

You can define formulas (mathematical expressions) for float and integer cells in a spreadsheet. The fpSpread control has over 70 mathematical, logical, and statistical functions, as described in Appendix G, “Formula Operators and Functions.”

The following sections describe adding formulas to the spreadsheet, and ways you can customize how you and users create and use formulas, including setting the style for cell references and creating custom functions.

### Adding Formulas to a Spreadsheet

You can add a formula to any cell in a spreadsheet. You can also let users enter formulas directly into the spreadsheet. For more information on letting users add formulas, see “Allowing Users to Enter Formulas” on page 107.

You can control the way in which formulas are updated and calculated. For more information, see “Automatically Recalculating and Updating Formulas” on page 107 and “Using Circular References in Formulas” on page 109. The way in which the control refers to the data in the cells is determined by the cell reference style you choose for the spreadsheet. For instructions and information about cell reference styles, see “Specifying Cell References” on page 103.

For a list of the operators and functions you can use in formulas, see Appendix G, “Formula Operators and Functions.” If the spreadsheet does not provide the logical or mathematical function you need, you can define your own. For instructions, see “Creating and Using Custom Functions” on page 105.

► **To add a formula to a spreadsheet**

Spread  
Designer

In the Spread Designer,

1. Select the cell or block of cells to which you want to add a formula.
2. From the Cell menu, choose Formula or press the Formula button () in the toolbar.
3. On the Formula tab in the Cell Settings dialog box, type the formula in the Formula box.

A list of valid operators available as mathematical expressions is displayed under Formula Operations. If you want to use functions supplied by Spread, choose the function from the drop-down list box and choose the Insert button.

4. Choose the OK or Apply button.

ActiveX, VBX

At run time,

1. Specify the cell or cells to which you want to add a formula with the **Col**, **Col2**, **Row**, and **Row2** properties as appropriate.
2. If you are selecting a block of cells, set the **BlockMode** property to True.
3. Define the formula with the **Formula** property.
4. If you selected a block of cells, set the **BlockMode** property back to False.

DLL

1. If you want to add a formula to a cell, call the **SSSetFormula** function.
2. If you want to add a formula to a range of cells, call the **SSSetFormulaRange** function.

## Specifying Cell References

The fpSpread control can use *absolute or relative* cell references. In absolute cell references, for example, “B2”, the column and row specified are the column and row used for calculations. In relative cell references, the column and row used for calculations change based on the location of the formula. For example, if you copy a formula using relative cell references from column B to column C, cell references such as “B2” change to “C2” to reflect the new formula location.

The fpSpread control offers three different cell notations, or *cell reference styles*, for representing cell addresses. The different cell reference styles default to creating either

absolute or relative cell references, but offer some options for either type of cell reference. Be sure to use the cell reference style that best suits your needs, and your users' expectations, if you are allowing users to enter formulas.

The following list describes and gives examples of each cell reference style.

- **Default cell reference style**

The default reference style uses letters and numbers for column and row coordinates, where the letters and numbers represent absolute coordinates. You can use a symbol, the number sign (#), to represent the current column or row as relative coordinates. For example, the reference A1 refers to cell A1; the reference B# refers to the cell in the current row in column B.

- **“A1” cell reference style**

The “A1” reference style uses letters and numbers for column and row coordinates, where the letters and numbers represent relative coordinates, similar to Microsoft Excel cell references. You can use a symbol, the dollar sign (\$), to indicate absolute coordinates. For example, if the cell containing the formula is C1, the reference D4 refers to the cell one column over and three rows down. If you wanted to create an absolute reference to cell D4, you can use the notation `DS$4`.

- **“R1C1” cell reference style**

The “R1C1” reference style uses “R” and number for row, “C” and number for column coordinates, where the notation represents absolute coordinates. You can use square brackets ([ ]) to represent relative coordinates, and the letters “C” or “R” without a number represent the current row or column. For example, R1C1 refers to the cell A1. If you are in cell C1, the reference `R[1]C[1]` refers to the cell one column to the right and one row down, which is D2. The reference R1C refers to the cell in the first row and the same column as the cell containing the formula.

---

### Notes

- Your formula cannot contain both absolute and relative row or column references. For example, the following formula, which uses the default reference style and adds the values in the first column up to the current row, is invalid.

```
fpSpread1.Formula = "sum(A1:A#) "
```

Likewise, the following formula to add the values in the first row up to the current column is invalid.

```
fpSpread1.Formula = "sum(A1:#1) "
```

- Returning the value of the **Formula** property provides a string containing the written expression of the formula, for example, SUM(A1:B1). However, if you have changed the cell reference style to a style that cannot represent the formula, the control provides the formula with question marks as placeholders for cell references that cannot be represented.

For example, if you have defined relative cell references used in a formula in cell B1 as RC[-1]+R[-1]C, the formula is interpreted as add the value in the cell to the left (A1) to the value in the cell above (“B0”). The control treats the value in the cell “B0” as an empty cell. If you change the cell reference style to the A1 style, the formula becomes A1+B?, because the A1 style cannot represent cell “B0”. However, the control still evaluates the formula as it would using the R1C1 reference style.

---

► **To specify the cell reference style**

Spread Designer

In the Spread Designer,

1. From the Calculation menu, choose Reference Style.
2. On the Reference Style tab in the Calculation dialog box, select the appropriate option button under Reference Style.
3. Choose the OK or Apply button.

ActiveX

Call the **SetRefStyle** method.

VBX

Call the **SpreadSetRefStyle** function.

DLL

Call the **SSSetRefStyle** function.

## Creating and Using Custom Functions

You can define custom functions for the spreadsheet. You can also remove a custom function.

► **To create and use a custom function**

ActiveX

At run time,

1. If you want to define a custom function with a specified number of parameters, call the **AddCustomFunction** method.
2. If you want to define a custom function with a variable number of parameters, call the **AddCustomFunctionExt** method.

3. Define the response to the **CustomFunction** event.

Use the **CFGetParamInfo** function to return information about the custom function's parameters. Use the **CFGetCellParam**, **CFGetDoubleParam**, **CFGetDoubleParamExt**, **CFGetLongParam**, **CFGetParamInfo**, **CFGetRangeParam**, and **CFGetStringParam** functions to return the values for the parameters. Use the **CFSetResult** function to return the value of the function.

4. Use the custom name in formulas as you would other operators.

VBX At run time,

1. If you want to define a custom function with a specified number of parameters, call the **SpreadAddCustomFunction** function.
2. If you want to define a custom function with a variable number of parameters, call the **SpreadAddCustomFunctionExt** function.
3. Define the response to the **CustomFunction** event.

Use the **SpreadCFGetParamInfo** function to return information about the custom function's parameters. Use the **SpreadCFGetCellParam**, **SpreadCFGetDoubleParam**, **SpreadCFGetDoubleParamExt**, **SpreadCFGetLongParam**, **SpreadCFGetParamInfo**, **SpreadCFGetRangeParam**, and **SpreadCFGetStringParam** functions to return the values for the parameters. Use the **SpreadCFSetResult** function to return the value of the function.

4. Use the custom name in formulas as you would other operators.

- DLL
1. If you want to define a custom function with a specified number of parameters, call the **SSAddCustomFunction** function.
  2. If you want to define a custom function with a variable number of parameters, call the **SSAddCustomFunctionExt** function.
  3. Use the custom name in formulas as you would other operators.

► **To remove a custom function**

ActiveX Call the **RemoveCustomFunction** method.

VBX Call the **SpreadRemoveCustomFunction** function.

DLL Call the **SSRemoveCustomFunction** function.

## Allowing Users to Enter Formulas

By default, users cannot enter formulas in cells in the spreadsheet. If you prefer, you can let users enter formulas in float or integer cells.

### ► To allow user-entered formulas

Spread  
Designer

In the Spread Designer,

1. From the General menu, choose Spreadsheet Environment or press the Spreadsheet Environment button () in the toolbar.
2. On the General tab in the Sheet Environment Settings dialog box under Settings, select the Custom Formulas check box.
3. Choose the OK or Apply button.

ActiveX, VBX

Set the **AllowUserFormulas** property to True.

DLL

Call the **SSSetBool** function, and set the parameters as follows:

1. Set the value of the *nIndex* parameter to `SSB_ALLOWUSERFORMULAS`.
2. Set the value of the *bNewVal* parameter to `TRUE`.

## Automatically Recalculating and Updating Formulas

By default, the spreadsheet recalculates formulas in the spreadsheet when the contents of dependent cells change. You can turn this recalculation off. You can also recalculate an individual cell.

Also by default, the spreadsheet updates formulas when you insert or delete columns or rows or when you move or swap blocks of cells. You can turn off these automatic formula updates. However, generally, you probably want the spreadsheet to update formulas when you insert or delete columns or rows or when you move or swap blocks of cells.

Keep in mind how turning off automatic formula updating might impact the spreadsheet if the user moves data, adds rows or columns, or performs other actions that affect the location of data.

When automatic formula updating is on, the spreadsheet updates absolute and relative cell references, as follows:

- When the spreadsheet is updating formulas, it updates absolute cell references when the cell referenced by the formula is part of the block that has changed.

For example, if you have a formula in cell C3 that references cell A1, which uses an absolute reference, and then add a row to the top of the spreadsheet, you now want the

formula to reference cell A2, because cell A1 is empty. If the spreadsheet did not update the formula, your formula would be referencing different data.

- When the spreadsheet is updating formulas, it updates relative cell references when the cell referenced by the formula is not part of the block that has changed.

For example, if you have a formula in cell C3 that references cell A3 as a relative reference, it references cell A3 as the cell that is two cells above it. If you add a row between row 2 and row 3, cell C3 is now C4, and the relative address references cell B3, the cell two cells above it. Therefore, to use the same data in the formula, the spreadsheet updates the cell reference to the cell three cells above it, A3.

► **To turn off automatic formula calculation and updates**

Spread  
Designer

In the Spread Designer,

1. From the General menu, choose Spreadsheet Environment or press the Spreadsheet Environment button () in the toolbar.
2. On the General tab in the Sheet Environment Settings dialog box under Settings,
  - If you want to turn off automatic formula calculation, clear the Calculate Formulas check box.
  - If you want to turn off automatic formula updates, clear the Formula Sync check box.
3. Choose the OK or Apply button.

ActiveX

1. If you want to turn off automatic formula calculation, set the **AutoCalc** property to False.
2. If you want to turn off automatic formula updates, set the **FormulaSync** property to False.

VBX

1. If you want to turn off automatic formula calculation, set the **AutoCalc** property to False.
2. If you want to turn off automatic formula updates, call the **SpreadSetFormulaSync** function, and set the value of the *Sync* parameter to False.

DLL

1. If you want to turn off automatic formula calculation, call the **SSSetBool** function, and set the parameters as follows:
  - a. Set the value of the *nIndex* parameter to `SSB_AUTOCALC`.
  - b. Set the value of the *bNewVal* parameter to `FALSE`.

2. If you want to turn off automatic formula updates, call the **SSSetBool** function, and set the parameters as follows:
  - a. Set the value of the *nIndex* parameter to `SSB_FORMULASYNC`.
  - b. Set the value of the *bNewVal* parameter to `FALSE`.

► **To recalculate an individual cell**

- |         |  |
|---------|--|
| ActiveX | <ol style="list-style-type: none"> <li>1. Set the <b>AutoCalc</b> property to <code>False</code>.</li> <li>2. Call the <b>ReCalcCell</b> method.</li> </ol>  |
| VBX     | <ol style="list-style-type: none"> <li>1. Set the <b>AutoCalc</b> property to <code>False</code>.</li> <li>2. Call the <b>SpreadReCalcCell</b> function.</li> </ol>  |
| DLL     | <ol style="list-style-type: none"> <li>1. Call the <b>SSSetBool</b> function, and set the parameters as follows:           <ol style="list-style-type: none"> <li>a. Set the value of the <i>nIndex</i> parameter to <code>SSB_AUTO CALC</code>.</li> <li>b. Set the value of the <i>bNewVal</i> parameter to <code>FALSE</code>.</li> </ol> </li> <li>2. Call the <b>SSReCalcCell</b> function and set the <i>lCol</i> and <i>lRow</i> parameters to specify the cell.</li> </ol> |

## Using Circular References in Formulas

You can specify how the spreadsheet calculates *circular references* used in formulas. A circular reference occurs when a formula uses the value in a cell to compute another value, which in turn is used by the first cell to recalculate its value. A simple example is if cell A1's value is equal to the formula `B1+C1`, and the value of cell C1 is equal to `A1*3`.

You can control the number of times the spreadsheet calculates a circular reference.

► **To customize circular references**

- |                 |  |
|-----------------|--|
| Spread Designer | <p>In the Spread Designer,</p> <ol style="list-style-type: none"> <li>1. From the Calculation menu, choose Iteration.</li> <li>2. On the Iteration tab in the Calculation dialog box,           <ol style="list-style-type: none"> <li>a. Specify that you want to allow iterations by selecting the Allow Iterations check box.</li> <li>b. Specify the maximum number of iterations for the spreadsheet to perform in the Maximum Iterations text box.</li> </ol> </li> <li>3. Choose the OK or Apply button.</li> </ol> |
|-----------------|--|

ActiveX	Call the <b>SetIteration</b> method.
VBX	Call the <b>SpreadSetIteration</b> function.
DLL	Call the <b>SSSetIteration</b> function.

## Returning Data from a Spreadsheet

You can return either formatted or unformatted data from a specified spreadsheet cell or block of cells. Returning formatted data returns the formatting characters or information with the numeric value, such as monetary symbols, as in the value \$1,234.56. Returning the same value as an unformatted value returns the numeric value 1234.56, with no dollar sign and no separator character. For more information about formatted versus unformatted data, see “Formatted and Unformatted Data” on page 30.

You need to choose the method to return data that returns the data and formatting characteristics that you need. The following table lists the ActiveX/VBX properties, methods, and functions and DLL functions used to return formatted and unformatted data from the spreadsheet.

Type of Data Returned	Spreadsheet Area	ActiveX/VBX Properties	ActiveX/VBX Methods or Functions	DLL Functions
Formatted	A cell, a column, a row, or the entire spreadsheet	<b>Text</b>	<b>GetText</b>	<b>SSGetData</b>
	A block of cells	<b>Clip</b>		<b>SSClipOut</b>
Unformatted	A cell, a column, a row, or the entire spreadsheet	<b>Value</b>	<b>GetFloat,</b> <b>GetInteger</b>	<b>SSGetFloat,</b> <b>SSGetInteger,</b> <b>SSGetValue</b>
	A block of cells	<b>ClipValue</b>		<b>SSClipValueOut</b>

The properties, methods, and functions listed in the table return data as string values, except for the **GetFloat** and **GetInteger** functions or methods, and the **SSGetFloat** and **SSGetInteger** functions. You can return data for float and integer cells as a floating-point or integer value. If you are using the DLL control and are returning data from one of the following cell types, refer to the listed functions’ information in the online *DLL Reference Guide*:

Cell Type	Functions
Float	<b>SSGetFloat</b>
Integer	<b>SSGetInteger</b>

If you are using the ActiveX or VBX control and are returning data from one of the following cell types, refer to the listed functions' information in the online *ActiveX/VBX Reference Guide*:

Cell Type	Functions
Float	<b>GetFloat</b>
Integer	<b>GetInteger</b>

## Returning Formatted Data

You can return formatted data from a specified spreadsheet cell, a column, a row, a block of cells, or the entire spreadsheet.

### ► To return formatted data from a cell, a column, a row, or the entire spreadsheet

**ActiveX** Either use the **Text** property or call the **GetText** method. Note that using the **GetText** method is faster than using the **Text** property.

1. If you are using the **Text** property, at run time,
  - a. Set the **Col** and **Row** properties as appropriate to specify the spreadsheet area from which you want to return data.
  - b. Return the data using the **Text** property.
2. If you are using the method, call the **GetText** method.

**VBX** Either use the **Text** property or call the **SpreadGetText** function. Note that using the **SpreadGetText** function is faster than using the **Text** property.

1. If you are using the **Text** property, at run time,
  - a. Set the **Col** and **Row** properties as appropriate to specify the spreadsheet area from which you want to return data.
  - b. Return the data using the **Text** property.
2. If you are using the function, call the **SpreadGetText** function.

**DLL** Call the **SSGetData** function.

---

**Note** If you set the **SSGetData** function's *Col* and *Row* parameters to **SS\_ALLCOLS** and **SS\_ALLROWS**, the **SSGetData** function will return the default text for the entire spreadsheet only if you added the text using the **SSSetData** function and the **SS\_ALLCOLS** and **SS\_ALLROWS** parameter values for the *Col* and *Row* parameters.

---

## Chapter 5

### ► To return formatted data from a block of cells

ActiveX, VBX

At run time,

1. Set the **Col**, **Row**, **Col2**, and **Row2** properties to specify the block of cells from which you want to return data.

You do not need to set the **BlockMode** property.

2. Return the data using the **Clip** property.

Data for each column is separated with a tab character; each row is separated with a carriage return character.

DLL Call the **SSClipOut** function.

## Returning Unformatted Data

You can return unformatted data from a specified spreadsheet cell, a column, a row, a block of cells, or the entire spreadsheet.

### ► To return unformatted data from a cell, a column, a row, or the entire spreadsheet

ActiveX, VBX

At run time,

- If you are returning float or integer data, use the **GetFloat** or **GetInteger** function or method.
- If you are returning other data,
  1. Set the **Col** and **Row** properties as appropriate to specify the spreadsheet area from which you want to return data.
  2. Return the data using the **Value** property.

DLL

- If you are returning float or integer data, call the **SSGetFloat** or **SSGetInteger** function.
- If you are returning other data, call the **SSGetValue** function.

► **To return unformatted data from a block of cells**

ActiveX, VBX

At run time,

1. Set the **Col**, **Row**, **Col2**, and **Row2** properties to specify the block of cells from which you want to return data.

You do not need to set the **BlockMode** property.

2. Return the data using the **ClipValue** property.

Data for each column is separated with a tab character; each row is separated with a carriage return character.

DLL Call the **SSClipValueOut** function.



# Chapter 6 Working with Spreadsheets

This chapter explains how to save, print, and reset a spreadsheet, and set other spreadsheet characteristics, as described in the following topics:

Topic	On Page
Saving Spreadsheet Data	115
Specifying Which Elements Users Can Select	126
Resetting a Spreadsheet to its Default Settings	127
Monitoring Data Changes in the Spreadsheet	128
Specifying the Operation Mode	128
Using Virtual Mode	130

## Saving Spreadsheet Data

You can save spreadsheet data to a file, using one of the following file formats:

- Binary file format
- Tab-delimited format

You can export spreadsheet data to a file, using one of the following formats:

- Excel 97 format (BIFF8) (ActiveX or 32-bit DLL controls only)
- HTML format (version 3.2 or later) (ActiveX or 32-bit DLL controls only)

DLL users can save a spreadsheet to a buffer, saving only the data or the data and formatting.

You can also save data to an array.

---

**Note** The Spread version 3.0 ActiveX controls are safe for scripting. If you are using the Spread ActiveX control in a container that supports IObjectSafety such as Internet Explorer 4, the user cannot export or save files.

---

## Saving a Spreadsheet to a Binary or Tab-Delimited File or to a Buffer

Binary files can contain all spreadsheet data and formatting or just spreadsheet data. Binary files are in a proprietary format that only the fpSpread control or the Spread Designer can read. You can save the spreadsheet's data and formatting or save only the spreadsheet's

data to a binary file. Saving only the spreadsheet data creates a much smaller file; however, no formatting information is saved.

Tab-delimited files contain spreadsheet data separated by tabs and carriage returns. Tab-delimited files can be opened, modified, and saved using any standard text editor.

If you use the fpSpread DLL control, you can save a spreadsheet to a buffer, saving only the data or the data and formatting.

By default, spreadsheet files created using the fpSpread control or the Spread Designer have the extension .SS3 (binary) or .TB3 (tab-delimited) or, if you are using a 32-bit control, .XLS (Excel format) or .HTM (HTML format). However, you can use any extension when you save or export from the fpSpread control or the Spread Designer.

---

**Note** Spread version 3.0 can open and read binary (.SS2) files that were created using previous versions of Spread, but all Spread version 3.0 binary files are now saved with the .SS3 extension.

---

If you are using the Spread Designer, see “Saving and Applying Changes to the Spreadsheet” on page 67 for instructions for saving spreadsheets to a file.

For instructions on opening spreadsheet files, see “Loading an Existing Spreadsheet” on page 81.

### ► **To save spreadsheet data or formatting and data to a binary file**

ActiveX Call the **SaveToFile** method, and set the parameters as follows:

1. Set the value of the *FileName* parameter to the path and file name of the spreadsheet file you want to save.
2. Set the value of the *DataOnly* parameter to one of the following:
  - False to specify that both data and formatting be saved
  - True to specify that only data be saved

VBX Call the **SpreadSaveToFile** function, and set the parameters as follows:

1. Set the value of the *FileName* parameter to the path and file name of the spreadsheet file you want to save.
2. Set the value of the *DataOnly* parameter to one of the following:
  - False to specify that both data and formatting be saved
  - True to specify that only data be saved

- DLL Call the **SSSaveToFile** function, and set the parameters as follows:
1. Set the value of the *lpFileName* parameter to the path and file name of the spreadsheet file you want to save.
  2. Set the value of the *fDataOnly* parameter to one of the following:
    - FALSE to specify that both data and formatting be saved
    - TRUE to specify that only data be saved

▶ **To save a spreadsheet to a tab-delimited file**

ActiveX Call the **SaveTabFile** method.

VBX Call the **SpreadSaveTabFile** function.

DLL Call the **SSSaveTabFile** function.

▶ **To save a spreadsheet to a buffer**

- DLL Call the **SSSaveToBuffer** function, and set the parameters as follows:
1. If you want to save both spreadsheet data and formatting, set the value of the *fDataOnly* parameter to FALSE.
  2. If you want to save spreadsheet data only, set the value of the *fDataOnly* parameter to TRUE.

---

**Note** The calling application is responsible for freeing the return buffer.

---

## Exporting a Spreadsheet to an Excel-Formatted File

---

### Notes

- You can only export a spreadsheet to an Excel-formatted file if you are using the ActiveX or 32-bit DLL control, or if you are using the 32-bit Spread Designer.
  - If your spreadsheet contains numeric data and you are planning to export the spreadsheet to Excel, for best results place any numeric data in integer or float cells in Spread.
- 

You can export a spreadsheet to the Microsoft Excel 97 format (BIFF8). Spreadsheet data and property settings are translated into Excel format as explained in Appendix I, “Excel-Formatted File Import/Export.”

By default, the file created by exporting to an Excel-formatted file has the extension .XLS; however, you can use any extension when you export from the fpSpread control or the Spread Designer.

---

**Tip** You might want to save your spreadsheet as an .SS3 file in addition to exporting it to Excel so that you can open the .SS3 file for future modifications.

---

If you are using the Spread Designer, see “Saving and Applying Changes to the Spreadsheet” on page 67 for instructions for exporting spreadsheets to an Excel-formatted file.

For instructions on importing Excel-formatted files, see “Loading an Existing Spreadsheet” on page 81.

► **To export spreadsheet data to an Excel-formatted file**

ActiveX Call the **ExportToExcel** method, and set the parameters as follows:

1. Set the value of the *FileName* parameter to the path and file name of the spreadsheet file you want to export.

If you do not specify an extension, Spread will add .XLS.

If you do not specify a path, Spread will export the file to the current drive and directory.

2. Set the value of the *SheetName* parameter to specify the name of the sheet you are creating in the file.

If you do not specify a name in the parameter, the sheet is automatically named Sheet1.

3. Set the value of the *LogFileName* parameter to the path and file name of the log file that records the manner in which Spread handles the export of Excel features, including cell types and other settings.

If you do not want to generate a log file, set the *LogFileName* parameter to "".

If you do not specify a path, Spread will save the file to the current drive and directory.

DLL  
(32-bit only)

Call the **SSExportToExcel** function, and set the parameters as follows:

1. Set the value of the *lpszFileName* parameter to the path and file name of the spreadsheet file you want to export.  
  
If you do not specify an extension, Spread will add .XLS.  
  
If you do not specify a path, Spread will export the file to the current drive and directory.
2. Set the value of the *lpszSheetName* parameter to specify the name of the sheet you are creating in the file.
3. Set the value of the *lpszLogFileName* parameter to the path and file name of the log file that records the manner in which Spread handled the import or export of features, including cell types and other settings. If you do not want to generate a log file, set the *LogFileName* parameter to NULL.

If you do not specify a path, Spread will save the file to the current drive and directory.

## Exporting a Spreadsheet to an HTML File

---

### Notes

- You can only export a spreadsheet to an HTML file format if you are using the ActiveX or 32-bit DLL control, or if you are using the 32-bit Spread Designer.
  - Turn off virtual mode before exporting a spreadsheet to an HTML file.
  - You can use the URL function in a cell formula to create a hypertext link in the exported HTML file. The link can display either a complete URL address (for example, <http://www.fpoint.com>) or a descriptive name (for example, FarPoint).
- 

You can export a cell, a block of cells, a column, a row, or the entire spreadsheet to an HTML file. You can also export discontinuous cells, blocks of cells, columns, and rows by performing one of the following actions:

- Hide the cells you do not want to export  
  
The **ExportToHTML** and **ExportRangeToHTML** methods do not export hidden rows and columns.
- Append the cells, block of cells, columns, or rows to the first cell, block of cells, column, or row you export.

The **ExportRangeToHTML** method has an *AppendFlag* parameter. The first time you call the method, set the *AppendFlag* parameter to False. For each succeeding method call, set the *AppendFlag* parameter to True until all the cells, blocks of cells, columns, or rows you want to add have been added.

For example, assume your spreadsheet has 15 rows and you want to export only the subtotals of your spreadsheet (rows 5, 10, and 15) to an HTML file. You could use the following code to hide all rows except the subtotal rows and then export rows 5, 10, and 15.

```
Sub Form_Load
    Dim i As Integer
    Dim rc As Integer
    fpSpread1.MaxRows = 15
    For i = 1 to 15
        fpSpread1.Row = i
        If i = 5 Or i = 10 Or i = 15 Then
            fpSpread1.RowHidden = False
        Else
            fpSpread1.RowHidden = True
        End If
    Next i
    rc = fpSpread1.ExportToHTML("c:\subtotal.htm", 0,
    ↪"c:\mylog.txt")
End Sub
```

You could also use the following code to append rows 10 and 15 to row 5.

```
SpreadExportRangeToHTML(1, 5, -1, 5, "c:\subtotal.htm", 0,
    ↪"c:\mylog.txt")
SpreadExportRangeToHTML(1, 10, -1, 10, "c:\subtotal.htm", 1,
    ↪"c:\mylog.txt")
SpreadExportRangeToHTML(1, 15, -1, 15, "c:\subtotal.htm", 1,
    ↪"c:\mylog.txt")
```

When you export discontinuous cells, blocks of cells, columns, or rows,

- Data is loaded into the HTML table in the order in which it is appended.
- Appended data is placed below the previously exported data.
- If the cell is blank or the block of cells, row, or column contains blank cells, the blank cell or cells will appear in the HTML file.
- If you display the column or row headers, the headers will display for each range you export.

For example, assume you have the following spreadsheet data and you want to export blocks A1:B3 and D4:E5.

	A	B	C	D	E
1	1001	2001	3001	4001	5001
2	1002	2002	3002	4002	5002
3	1003	2003	3003	4003	5003
4	1004	2004	3004	4004	
5	1005	2005	3005	4005	5005
6					
7					

The code to export the blocks is:

```
SpreadExportRangeToHTML(1, 1, 2, 3, "c:\block.htm", 0,
↳"c:\mylog.txt")
SpreadExportRangeToHTML(4, 4, 5, 5, "c:\block.htm", 1,
↳"c:\mylog.txt")
```

The resulting HTML file would appear like this.

	A	B
1	1001	2001
2	1002	2002
3	1003	2003
	D	E
4	4004	
5	4005	5005

Notice that the blank cell E4 is exported and column and row headers display for each range that is exported. Row headers also re-number so that the row numbering sequence is correct.

---

**Tip** After the first export, set the **DisplayColHeaders** property to False to have subsequent exports just export the data, not the column headers. When finished, set the **DisplayColHeaders** property to True.

---

Spreadsheet data and property settings are translated into HTML as explained in Appendix J, “HTML File Export.”

By default, the file created by exporting to an HTML file has the extension .HTM; however, you can use any extension when you export from the fpSpread control or the Spread Designer.

---

**Tip** You might want to save your spreadsheet as an .SS3 file in addition to exporting it to HTML so that you can open the .SS3 file for future modifications.

---

If you are using the Spread Designer, see “Saving and Applying Changes to the Spreadsheet” on page 67 for instructions for saving spreadsheets to a file.

► **To export spreadsheet data to an HTML file**

ActiveX

1. If you want to exclude part of the spreadsheet, hide the appropriate columns or rows with the **ColHidden** or **RowHidden** properties.
2. If you do not want to display column or row headers, set the **DisplayColHeaders** or **DisplayRowHeaders** property to False.
3. Complete the following steps, depending on whether you want to export a cell, rows, columns, the entire spreadsheet, or blocks of cells in the spreadsheet.
  - If you want to export the entire spreadsheet or export several spreadsheets into one HTML file, call the **ExportToHTML** method, and set the parameters as follows:
    - a. Set the *FileName* parameter to a valid path and file name.  
If you do not specify an extension, Spread will add .HTM.  
If you do not specify a path, Spread will export the file to the current drive and directory.
    - b. If one of the following statements is true, set the *AppendFlag* parameter to 0 (False).
      - This is the only spreadsheet you want to export.
      - This is the first time you are calling the **ExportToHTML** method for exporting multiple spreadsheets into one HTML file.

---

**Note** The first time you call the **ExportToHTML** method, you must set the *AppendFlag* parameter to 0.

---

- c. If this is the second or later time you are calling the **ExportToHTML** method, set the *AppendFlag* parameter to 1 (True).

- d. Set the value of the *LogFile* parameter to the path and file name of the log file that records the manner in which Spread handles the export of HTML features, including cell types and other settings.

If you do not want to generate a log file, set the *LogFile* parameter to "".

If you do not specify a path, Spread will save the file to the current drive and directory.

- e. Repeat steps 3a–3d until you have exported all the spreadsheets.
- If you want to export a cell, a column, a row, or blocks of cells, call the **ExportRangeToHTML** method, and set the parameters as follows:
  - a. Set the value of the *Col*, *Row*, *Col2*, and *Row2* parameters to specify the cell, row, column, or block of cells.  
 “Relationship of Col and Row Properties or Parameters” on page 31 explains valid values for the *Col*, *Row*, *Col2*, and *Row2* parameters.
  - b. Set the *FileName* parameter to a valid path and file name.  
 If you do not specify an extension, Spread will add .HTM.  
 If you do not specify a path, Spread will export the file to the current drive and directory.
  - c. If one of the following statements is true, set the *AppendFlag* parameter to 0 (False).
    - This is the only cell, column, row, or block of cells you want to export.
    - This is the first time you are calling the **ExportRangeToHTML** method for a set of discontinuous data.

---

**Note** The first time you call the **ExportRangeToHTML** method, you must set the *AppendFlag* parameter to 0.

---

- d. If this is the second or later time you are calling the **ExportRangeToHTML** method for a set discontinuous data, set the *AppendFlag* parameter to 1 (True).

- e. Set the value of the *LogFile* parameter to the path and file name of the log file that records the manner in which Spread handles the export of HTML features, including cell types and other settings.

If you do not want to generate a log file, set the *LogFile* parameter to "".

If you do not specify a path, Spread will save the file to the current drive and directory.

- f. Repeat steps 3a–3e until you have exported all the data you want.

DLL  
(32-bit only)

1. If you want to exclude part of the spreadsheet, hide the appropriate columns or rows call the **SSShowCol** or **SSShowRow** functions and set the *fShow* parameter to FALSE.
2. If you do not want to display column or row headers, call the **SSSetBool** function and set the *SSB\_SHOWCOLHEADERS* and *SSB\_SHOWROWHEADERS* boolean to FALSE.
3. If you want to export the entire spreadsheet or export several spreadsheets into one HTML file, call the **SSExportToHTML** function, and set the parameters as follows:

- a. Set the *col* and *row* parameters to -1.
- b. Set the *lpszHtmlFile* parameter to a valid path and file name.

If you do not specify an extension, Spread will add .HTM.

If you do not specify a path, Spread will export the file to the current drive and directory.

- c. If one of the following statements is true, set the *bAppendFlag* parameter to 0 (False).
  - This is the only spreadsheet you want to export.
  - This is the first function call for exporting multiple spreadsheets into one HTML file.

---

**Note** The first time you call the **SSExportToHTML** function, you must set the *bAppendFlag* parameter to 0.

---

- d. If this is the second or later function call, set the *bAppendFlag* parameter to 1 (True).

- e. Set the value of the *lpszLogFile* parameter to the path and file name of the log file that records the manner in which Spread handles the export of HTML features, including cell types and other settings.

If you do not want to generate a log file, set the *lpszLogFile* parameter to NULL.

If you do not specify a path, Spread will export the file to the current drive and directory.

- f. Repeat steps 3a–3e until all spreadsheets are appended.

4. If you want to export a cell, column, rows, or blocks of cells, call the **SSExportRangeToHTML** function, and set the parameters as follows:

- a. Set the value of the *col*, *row*, *col2*, and *row2* parameters to specify the cell, column, row, or block of cells.

“Relationship of Col and Row Properties or Parameters” on page 31 defines valid values for the *col* and *row* parameters.

- b. Set the *lpszHtmlFile* parameter to a valid path and file name.

If you do not specify an extension, Spread will add .HTM.

If you do not specify a path, Spread will export the file to the current drive and directory.

- c. If one of the following statements is true, set the *bAppendFlag* parameter to 0 (False).

- This is the only cell, column, row, or block of cells you want to export.
- This is the first function call for a set of discontinuous data.

---

**Note** The first time you call the **SSExportToHTML** function, you must set the *bAppendFlag* parameter to 0.

---

- d. If this is the second or later function call for a set discontinuous data, set the *bAppendFlag* parameter to 1 (True).
- e. Set the value of the *lpszLogFile* parameter to the path and file name of the log file that records the manner in which Spread handles the export of HTML features, including cell types and other settings.

If you do not want to generate a log file, set the *lpszLogFile* parameter to NULL.

If you do not specify a path, Spread will export the file to the current drive and directory.

- f. Repeat steps 4a–4e until all data is appended.

## Saving Data to an Array

You can save spreadsheet data to an array.

If you are using the VBX control, you must declare the array as a variant. If you are using the ActiveX control, you can declare the array as a string, integer, long, byte, single, double, or variant. If you are using the DLL control, the array type can be short, long, float, double, or int.

► **To save data to an array**

ActiveX Call the **GetArray** method and set the parameters as follows:

1. Specify the top, leftmost cell that contains data you want to save with the *ColLeft* and *RowTop* parameters.
2. Specify the array name with the *VarArray* parameter.

VBX Call the **SpreadGetArray** function and set the parameters as follows:

1. Specify the top, leftmost cell that contains data you want to save with the *ColLeft* and *RowTop* parameters.
2. Specify the array name with the *hAD()* parameter.

DLL Call the **SSGetArray** function and set the parameters as follows:

1. Specify the name of the array with the *lpArray* parameter.
2. Specify the top, leftmost cell that contains data you want to save with the *ColLeft* and *RowTop* parameters.
3. Define the array size with the *ArrayColCnt* and *ArrayRowCnt* parameters.
4. Declare the data type of the array with the *wDataType* parameter.

## Specifying Which Elements Users Can Select

You can let the user select only columns, rows, blocks, all cells, or any combination of these.

In addition to this setting, the operation mode of the spreadsheet affects which elements users can select. For example, in single-selection mode, users can only select one row in the spreadsheet at a time, similar to the behavior of a single-selection list box. For more information about operation mode, see “Specifying the Operation Mode” on page 128.

► **To specify elements users can select**

Spread  
Designer

In the Spread Designer,

1. From the General menu, choose Spreadsheet Environment or press the Spreadsheet Environment button () in the toolbar.
2. On the BlockMode tab in the Sheet Environment Settings dialog box, select the appropriate check boxes under User Can Select.
3. Choose the OK or Apply button.

ActiveX, VBX

Set the **SelectBlockOptions** property.

DLL

Call the **SSSetSelBlockOptions** function.

## Resetting a Spreadsheet to its Default Settings

You can reset a spreadsheet to its default settings. Resetting a spreadsheet returns the spreadsheet to its initial state (prior to any design-time or run-time changes). Resetting a spreadsheet clears data, resets colors, and returns all cells to the default cell type (edit).

---

**Note** If you are using the ActiveX or VBX control, the font is reset to 8 pt MS San Serif. If you are using the DLL control, the font is not reset.

---

You might want to reset the spreadsheet when you let users choose the New choice from a File menu.

► **To reset a spreadsheet to its default settings**

Spread  
Designer

In the Spread Designer,

1. Select the entire spreadsheet by clicking the top left cell.
2. From the Edit menu choose Clear, Clear All.

ActiveX, VBX

At run time, set the **Action** property to 28 (Reset).

DLL

Call the **SSReset** function.

## Monitoring Data Changes in the Spreadsheet

---

**Note** This topic applies to ActiveX and VBX users only.

---

You can specify that any time data changes in a cell, a column, a row, or the entire spreadsheet, the change is noted. You might want to do this for bound spreadsheets to check for and write changed data to the bound database.

► **To monitor data changes**

- ActiveX
1. Using the **SetCellDirtyFlag** method,
    - a. Set the value of the *Col* parameter to the column number of the cell to monitor, or set the value to -1 to specify an entire row.
    - b. Set the value of the *Row* parameter to the row number of the cell or cells to monitor.
    - c. Set the value of the *Dirty* parameter to True to flag changed cells.
  2. Use the **GetCellDirtyFlag** method to check whether a specific cell's data has been changed.
- VBX
1. Using the **SpreadSetCellDirtyFlag** function,
    - a. Set the value of the *Col* parameter to the column number of the cell to monitor, or set the value to -1 to specify an entire row.
    - b. Set the value of the *Row* parameter to the row number of the cell or cells to monitor.
    - c. Set the value of the *Dirty* parameter to True to flag changed cells.
  2. Use the **SpreadGetCellDirtyFlag** function to check whether a specific cell's data has been changed.

## Specifying the Operation Mode

Operation modes control how the spreadsheet reacts to user interaction and the types of user interaction available. The operation mode you apply also limits user interaction such as moving the focus and selecting cells.

As described in the following list, you can set the spreadsheet to operate like a single-selection, multiple-selection, or extended-selection list box. Choose from the following operation modes:

- **Normal**  
The default spreadsheet mode. Clicking a row selects the row. Clicking a column selects the column. Clicking a cell moves the focus to that cell. Double-clicking a cell turns on edit mode.
- **Read only**  
The cells, rows, and columns cannot receive the focus.
- **Row**  
Clicking a row moves the focus to that row. Double-clicking a cell moves the focus to that cell.
- **Single-selection**  
The spreadsheet operates similarly to a single-selection list box. Clicking a row selects the row. One and only one row is selected at all times. The user cannot move the focus to an individual cell.
- **Multiple-selection**  
The spreadsheet operates similarly to a multiple-selection list box. Clicking a row selects the row. The user can click additional rows to select them. Clicking a selected row deselects it. The user cannot move the focus to an individual cell.
- **Extended-selection**  
The spreadsheet operates similarly to a extended-selection list box. Clicking a row selects the row. The user can click while pressing Ctrl to select additional rows. The user can click and press Shift to select a range of rows. Pressing Ctrl and clicking a selected row deselects it. The user cannot move the focus to an individual cell.

---

**Note** Do not set the spreadsheet to use multiple- or extended-selection operation mode if the spreadsheet is using virtual mode.

---

### ► To specify an operation mode

Spread Designer	<p>In the Spread Designer,</p> <ol style="list-style-type: none"><li>1. From the General menu, choose Spreadsheet Environment or press the Spreadsheet Environment button () in the tool bar.</li><li>2. On the SheetMode tab in the Sheet Environment Settings dialog box, select the appropriate option button under Spreadsheet Operation.</li><li>3. Choose the OK or Apply button.</li></ol>
ActiveX, VBX	Specify the operation mode using the <b>OperationMode</b> property.
DLL	Call the <b>SSSetOperationMode</b> function and set the <i>wMode</i> parameter.

## Using Virtual Mode

Use virtual mode to increase fpSpread control responsiveness when working with large amounts of data. You can use virtual mode for a spreadsheet with a large number of rows or for a spreadsheet bound to a large database.

If you do not use virtual mode, the spreadsheet can take a long time to display data. The diminished responsiveness occurs because the spreadsheet must read through all the rows before displaying the spreadsheet. In virtual mode, the spreadsheet reads ahead only when needed to display additional rows and retrieves records from a buffer area for faster response.

You can specify characteristics of virtual mode, such as how many rows are read into the buffer area at a time. In addition, you can display a special vertical scroll bar to use with virtual mode.

---

## Notes

Keep in mind the following when using virtual mode:

- If you are using a bound spreadsheet that is using virtual mode, the Data control bound to the spreadsheet might not behave as you expect. For more information, see “Using Virtual Mode with Bound Spreadsheets” on page 282.
  - If the spreadsheet does not know the total number of rows, the row headers might not display the correct numbers for the rows. For best results, hide the row headers. For instructions, see “Hiding the Column and Row Headers” on page 183.
  - For best results, do not use multiple- or extended-selection operation mode in a spreadsheet that is using virtual mode.
  - Do not freeze rows in a spreadsheet that is using virtual mode.
  - You cannot sort a spreadsheet that is using virtual mode.
  - When printing a spreadsheet that is using virtual mode,
    - For DLL users, in the **SS\_PRINTFORMAT** structure, set the *fUseDataMax* field to FALSE.
    - For ActiveX, VBX users, set the **PrintUseDataMax** property to False.
  - For ActiveX, VBX users, do not use the **Action** property setting 32 (Smart Print) to print a spreadsheet that is using virtual mode. Use the **Action** property setting 13 (Print) instead.
- 

## Turning on Virtual Mode

You can customize virtual mode using the instructions in the following sections. Use these instructions to turn virtual mode on.

► **To turn on virtual mode**

Spread  
Designer

In the Spread Designer,

1. From the General menu, choose Virtual Mode or press the Virtual Mode button () in the tool bar.
2. In the Virtual Mode dialog box, select the Virtual Mode check box under Settings.
3. Choose the OK or Apply button.

ActiveX, VBX

Set the **VirtualMode** property to True.

- DLL Call the **SSSetBool** function, and set the parameters as follows:
1. Set the value of the *nIndex* parameter to `SSB_VIRTUALMODE`.
  2. Set the value of the *bNewVal* parameter to `TRUE`.

### Specifying the Maximum Number of Rows

When using virtual mode, the maximum number of rows in the spreadsheet, which is set using the **MaxRows** property or the **SSSetMaxRows** function, is no longer valid. You can specify the maximum number of rows for virtual mode using the **VirtualMaxRows** property or **SSVSetMax** function.

If you add rows to a spreadsheet that is using virtual mode after all the rows have been read, the **VirtualMaxRows** property or the **SSVSetMax** function will still be set to the original number of rows and the `fpSpread` control will not be able to read all the rows in the spreadsheet. You must set the value of the **VirtualMaxRows** property or the *Max* parameter of the **SSVSetMax** function to `-1` to have it reset the number of rows, or you must provide the new number of rows.

If you are using a bound `fpSpread` control, once the database has been read and a value assigned to the **VirtualMaxRows** property or the **SSVSetMax** function, that value will not change until either you assign a new value to the **VirtualMaxRows** property or to the **SSVSetMax** function or a new database is bound to the `fpSpread` control and read. However, if you bind a database after first binding and reading a smaller database, the **VirtualMaxRows** property or the **SSVSetMax** function will be set to the number of records in the smaller database and the `fpSpread` control will not be able to read all the records in the larger database.

---

**Note** In the Spread Designer, when you turn virtual mode on, by default, the **VirtualMaxRows** and **MaxRows** properties are set to 1 million rows. If you do not specify a value for the **VirtualMaxRows** property and you turn off virtual mode, the value of the **MaxRows** property is still set to 1 million rows.

---

If you exit the Spread Designer without changing the **VirtualMaxRows** or **MaxRows** property, the control will verify 1 million rows. The control appears to be hung when in reality it is checking the 1 million rows. To avoid this problem, when you exit Spread Designer, a message box appears asking whether or not you would like to reset the **MaxRows** property to its default value of 500 rows.

► **To specify the maximum number of rows when using virtual mode**

Spread  
Designer

In the Spread Designer,

1. From the General menu, choose Virtual Mode or press the Virtual Mode button () in the tool bar.
2. Type the number of rows in the Virtual Max Rows box under Settings as follows:
  - If you know the exact number of rows in the spreadsheet or bound database, specify that number.
  - If you do not know the exact number of rows in the spreadsheet or bound database, but would like the spreadsheet to determine it once it has read all the rows, specify -1.
  - If you do not know the exact number of rows in the spreadsheet or bound database, but would like to estimate it, overestimate the number rather than underestimating it.

If you underestimate the number, the spreadsheet never reads the records past the value you assign.
3. Choose the OK or Apply button.

ActiveX, VBX

Set the **VirtualMaxRows** property as follows:

- If you know the exact number of rows in the spreadsheet or bound database, specify that number.
- If you do not know the exact number of rows in the spreadsheet or bound database, but would like the spreadsheet to determine it once it has read all the rows, specify -1.
- If you do not know the exact number of rows in the spreadsheet or bound database, but would like to estimate it, overestimate the number rather than underestimating it.

If you underestimate the number, the spreadsheet never reads the records past the value you assign.

DLL

Call the **SSVSetMax** function, and set the value of the *Max* parameter as follows:

- If you know the exact number of rows in the spreadsheet, specify that number.
- If you do not know the exact number of rows in the spreadsheet, but would like the spreadsheet to determine it once it has read all the rows, specify -1.
- If you do not know the exact number of rows in the spreadsheet, but would like to estimate it, overestimate the number rather than underestimating it.

If you underestimate the number, the spreadsheet never reads the records past the value you assign the *Max* parameter.

## Customizing Virtual Mode

You can specify how many rows are read into the virtual buffer at a time and whether some of the previously displayed rows are kept in the buffer when additional rows are read.

In addition, you can customize the vertical scroll bar to work more smoothly with virtual mode. For example, you can specify that the scroll box reflects only the number of records in the virtual buffer, rather than the entire number of records.

Alternatively, you can provide the number of rows in the spreadsheet. Providing the number of rows helps the vertical scroll bar more accurately reflect the current position in the spreadsheet and permits the row headers to accurately number the rows. For more information about specifying the number of rows in the spreadsheet, see “Specifying the Maximum Number of Rows” on page 132.

If you prefer, the control can display a special vertical scroll bar as described in “Displaying the Special Vertical Scroll Bar” on page 136.

### ► To customize virtual mode

Spread  
Designer

In the Spread Designer,

1. From the General menu, choose Virtual Mode or press the Virtual Mode button () in the tool bar.
2. Type the number of rows read into the virtual buffer at a time in the Rows Per Read box under Settings.

If the spreadsheet can display more rows at a time than the number of rows specified, the value in the Rows Per Read box changes to the number of rows the spreadsheet can display.

3. Type the total number of previously retrieved rows that are retained in a buffer in the Memory Max Rows box under Settings.
4. Under Virtual Scrollbar, select the Thumb Reflects Row in Buffer check box to have the vertical scroll box reflect only the number of rows in the virtual buffer rather than the total number of rows.
5. If you do not select the Thumb Reflects Row in Buffer check box or if the control displays the row headers with numbers, type a number in the Virtual Max Rows box to provide the spreadsheet with the number of rows in the spreadsheet or bound database.
6. Choose the OK or Apply button.
7. If your spreadsheet is not bound to a database, you must indicate the number of rows loaded in the **QueryData** event. After each data request, set the value of the *RowsLoaded* parameter to the number of rows that have been loaded.

- ActiveX, VBX
1. Specify the number of rows read into the virtual buffer at a time using the **VirtualRows** property.  
If the spreadsheet can display more rows at a time than the number of rows specified, the value of the **VirtualRows** property is reset to the number of rows the spreadsheet can display.
  2. Set the **VirtualOverlap** property to specify the total number of previously retrieved rows that are retained in a buffer.
  3. Set the **VirtualScrollBuffer** property to True to have the vertical scroll box reflect only the number of rows in the virtual buffer rather than the total number of rows.
  4. If you set the **VirtualScrollBuffer** property to False or if the control displays the row headers with numbers, use the **VirtualMaxRows** property to provide the spreadsheet with the number of rows in the spreadsheet or bound database.
  5. If your spreadsheet is not bound to a database, you must indicate the number of rows loaded in the **QueryData** event. After each data request, set the value of the *RowsLoaded* parameter to the number of rows that have been loaded.
- DLL
1. Call the **SSVSetBufferSize** function, and set the parameters as follows:
    - a. Set the value of the *dBufferSize* parameter to specify the number of rows read into the virtual buffer at a time.  
If the spreadsheet can display more rows at a time than the number of rows specified, the value of the *dBufferSize* parameter is reset to the number of rows the spreadsheet can display.
    - b. Set the value of the *dOverlap* parameter to specify the total number of previously retrieved rows that are retained in a buffer.
  2. Call the **SSVSetStyle** function, and set the value of the *lStyle* parameter as follows:
    - Set the value to **SSV\_SCROLLBARSHOWBUFFER** to have the vertical scroll bar reflect only the number of rows in the virtual buffer rather than the total number of rows.
    - Set the value to **SSV\_NOROWNUMBERS** to hide the text in the row header column.
    - Set the value to **SSV\_SCROLLBARSHOWBUFFER** combined with **SSV\_NOROWNUMBERS** using the Or operator.
  3. If the scroll bar is reflecting the total number of rows or if the control displays the row headers with numbers, call the **SSVSetMax** function, and set the value of the *Max* parameter to the number of rows in the spreadsheet.

- Each time data is loaded into the buffer, in the **SS\_VQUERYDATA** structure, set the value of the *RowsLoaded* field to the number of rows that have been loaded.

When the spreadsheet requests data, it sends the **SSM\_VQUERYDATA** message. The **SS\_VQUERYDATA** structure provides information to this message about the data loaded into the buffer.

## Displaying the Special Vertical Scroll Bar

When using virtual mode, the vertical scroll bar does not always accurately represent the current location in the data. Also, the scroll bar can be slow to respond to changes in location.

Instead of displaying the default vertical scroll bar, you can display a special vertical scroll bar. The special scroll bar does not contain a scroll box. Instead, it displays scroll arrows that let you move to the first or last page, up or down a row, and up or down a page. The fpSpread control can display all or some of the scroll arrows, as you specify.

### ► To display the special vertical scroll bar

Spread  
Designer

In the Spread Designer,

- From the Display menu, choose Scroll Bars or press the Scroll Bars button () in the tool bar.
- On the Scroll Bars tab in the Display dialog box, select either the Both or Vertical option button under Display.
- Choose the OK button.
- From the General menu, choose Virtual Mode or press the Virtual Mode button () in the tool bar.
- In the Virtual Mode dialog box, select the Use Special Scroll Bar check box under Virtual Scroll Bar.
- Specify which scroll arrows appear on the special scroll bar by selecting all, none, or some of the available check boxes under Use Special Scroll Bar.
- Choose the OK button.

ActiveX, VBX

- Set the **ScrollBars** property to 2 (Vertical) or 3 (Both).
- Set the **VScrollSpecial** property to True.
- At run time, use the **VScrollSpecialType** property to specify which scroll arrows are displayed.

- DLL
1. Call the **SSSetBool** function, and set the parameters as follows:
    - a. Set the value of the *nIndex* parameter to `SSB_VERTSCROLLBAR`.
    - b. Set the value of the *bNewVal* parameter to `TRUE`.
  2. Call the **SSVScrollSetSpecial** function, and set the parameters as follows:
    - a. Set the value of the *fUseSpecialVScroll* parameter to `TRUE`.
    - b. Set the value of the *wOptions* parameter to specify which scroll arrows to display.



# Chapter 7 Printing Spreadsheets

You can print your spreadsheet and use a variety of options to customize printing. By default, spreadsheets are printed one page at a time, from top to bottom, then left to right. Spreadsheets are printed on the current default printer in your Windows environment. You can print spreadsheets on any Windows-supported printer.

This chapter explains how to print, as described in the following topics:

<b>Topic</b>	<b>On Page</b>
Features	139
Working in the Spread Designer	140
Sending a Spreadsheet to a Printer	141
Calculating the Number of Pages to Print	142
Specifying Print Options	142
Printing the Portion of the Spreadsheet with Data	146
Specifying a Print Range	147
Specifying Print Page Breaks	149
Adding Header and Footer Text to Printed Pages	151

## Features

You can simply print your spreadsheet, as described in “Sending a Spreadsheet to a Printer” on page 141, or you can specify print options.

You can set print page margins, set the number to use to begin numbering the pages in the spreadsheet, select the page orientation, and specify whether to print the spreadsheet border, column and row headers, grid lines, colors, and shadow effect. You can specify the order in which pages are printed—down then across, or across then down. You can also specify header and footer text to appear on printed spreadsheet pages. For instructions, see “Specifying Print Options” on page 142 and “Adding Header and Footer Text to Printed Pages” on page 151.

You can print a portion of a spreadsheet by specifying either a range of pages or a block of cells to print. Alternatively, you can print only the spreadsheet columns and rows that contain data. You can also specify page breaks before columns or rows.

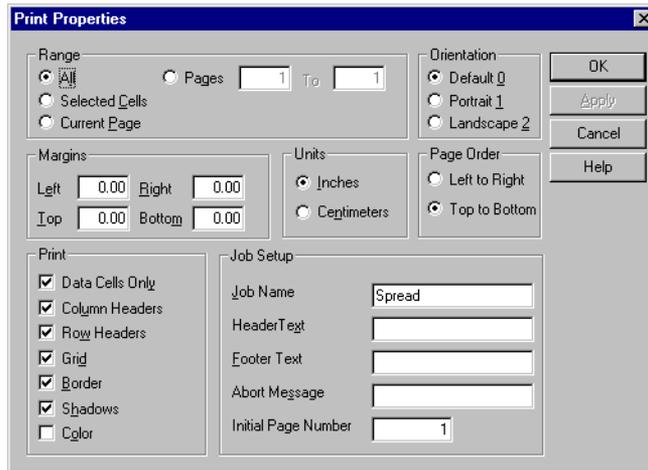
For instructions, see the following sections:

- “Specifying a Print Range” on page 147
- “Printing the Portion of the Spreadsheet with Data” on page 146
- “Specifying Print Page Breaks” on page 149

## Working in the Spread Designer

If you are using the Spread Designer, you can print either the active view in the designer, or you can set print properties to apply back to the spreadsheet to be used when the spreadsheet is printed. You cannot print the spreadsheet itself using the Spread Designer, only the spreadsheet view when it is the active view.

In the Spread Designer, if you want to set print properties to apply to the spreadsheet, from the General menu, choose Print Properties. The Print Properties dialog box appears, as shown in the following figure.



By default, the Print Properties dialog box loads the current print settings of the spreadsheet.

If you want to, you can save the settings you have chosen for printing the active view. For information about setting properties for and printing the active view in the designer, see “Printing the Active View” on page 58.

## Sending a Spreadsheet to a Printer

### ► To send a spreadsheet to a printer

- ActiveX
1. Prepare the spreadsheet for printing by specifying print options and the portion of the spreadsheet to print.
  2. At run time,
    - a. If you want to use the current print property settings, set the **Action** property to 13 (Print).
    - b. If you want to use Smart Printing provided by the spreadsheet, perform the following actions:
      - i. Set the **PrintSmartPrint** property to True.
      - ii. Set the **Action** property to 32 (Smart Print).

The Smart Print option determines the best orientation for printing the spreadsheet. It also scales the spreadsheet as appropriate to fit it on the page.

---

**Note** Do not use the **Action** property setting 32 (Smart Print) to print a spreadsheet that is using virtual mode. Use setting 13 (Print) instead.

---

- VBX
1. Prepare the spreadsheet for printing by specifying print options and the portion of the spreadsheet to print.
  2. At run time,
    - a. If you want to use the current print property settings, set the **Action** property to 13 (Print).
    - b. If you want to use Smart Printing provided by the spreadsheet, perform one of the following actions:
      - i. Call the **SpreadSetPrintOptions** function and set the *SmartPrint* parameter to True.
      - ii. Set the **Action** property to 32 (Smart Print)

The Smart Print option determines the best orientation for printing the spreadsheet. It also scales the spreadsheet as appropriate to fit it on the page.

---

**Note** Do not use the **Action** property setting 32 (Smart Print) to print a spreadsheet that is using virtual mode. Use setting 13 (Print) instead.

---

## Chapter 7

- DLL
1. Prepare the spreadsheet for printing by specifying print options and the portion of the spreadsheet to print.
  2. Call the **SSPrint** function to print the spreadsheet.

If you want to use the Smart Printing provided by the spreadsheet, set the *fSmartPrint* field of the **SS\_PRINTFORMAT** structure to 1 (True).

The Smart Print option determines the best orientation for printing the spreadsheet. It also scales the spreadsheet as appropriate to fit it on the page.

## Calculating the Number of Pages to Print

Based on the current print settings, you can determine how many pages will print.

► **To calculate the number of pages to print**

ActiveX At run time, use the **PrintPageCount** property to return the number of pages.

VBX At run time, call the **SpreadGetPrintPageCount** function.

DLL Call the **SSGetPrintPageCount** function.

## Specifying Print Options

Before printing a spreadsheet, you might want to specify print options such as margins, page orientation, and starting page number.

► **To specify print options**

Spread Designer In the Spread Designer,

1. From the General menu, choose Print Properties to display the Print Properties dialog box.
2. In the Print Properties dialog box,
  - a. If you want to use the settings in the Print Active View dialog box, choose the Import button to import the settings from the corresponding items in the Print Active View dialog box.

For more information about printing the active view and the Print Active View dialog box, see “Printing the Active View” on page 58.

- b. Under Orientation, select an option button to specify the page orientation used to print the spreadsheet.

- c. Under Page Order, select an option button to specify the order in which pages are printed.
- d. Specify the margins for the printed spreadsheet.
  - i. Under Units, select an option button to specify the unit type used for calculating the margins.
  - ii. Under Margins, provide the size of the margins in the Left, Right, Top, and Bottom boxes.
- e. Under Print, specify which spreadsheet features are printed by setting or clearing the following items:
  - i. If you do not want to print the column header, clear the Column Headers check box.
  - ii. If you do not want to print the row header, clear the Row Headers check box.
  - iii. If you do not want to print the spreadsheet grid lines, clear the Grid check box.
  - iv. If you do not want to print the spreadsheet border, clear the Border check box.
  - v. If you do not want to print the shadow effect within the headers, clear the Shadows check box.  
  
Note that if you clear the Column Headers and Row Headers check boxes, the Shadows setting does not have an effect.
  - vi. If you want to print the spreadsheet colors as they appear on the screen, select the Color check box.
- f. Under Job Setup, specify job settings for the print job by setting the following items.
  - i. Specify the print job name to display in the Print Manager when printing the spreadsheet by typing the name in the Job Name box.
  - ii. Specify the number to use to begin numbering pages in the spreadsheet by typing the number in the Initial Page Number box.
  - iii. Specify the text to display in an abort dialog box during printing by typing the message in the Abort Message box.
- g. Choose the OK or Apply button.

## Chapter 7

ActiveX At run time, customize any of the following print options by setting the corresponding property or properties:

1. Specify the page orientation used to print the spreadsheet with the **PrintOrientation** property.
2. Specify the size of the margins in twips using the **PrintMarginLeft**, **PrintMarginRight**, **PrintMarginTop**, and **PrintMarginBottom** properties.
3. Specify whether to print the spreadsheet border with the **PrintBorder** property.
4. Specify whether to print the spreadsheet column header and row header with the **PrintColHeaders** and **PrintRowHeaders** properties.
5. Specify whether to print the shadow effect within the spreadsheet column header and row header with the **PrintShadows** property.

Note that if you set the **PrintColHeaders** and **PrintRowHeaders** properties to False in step 4, the **PrintShadows** property setting will have no effect.

6. Specify whether to print the spreadsheet grid lines with the **PrintGrid** property.
7. Specify whether to print the colors in the spreadsheet as they appear on the screen with the **PrintColor** property.
8. Specify the print job name to display in the Print Manager when printing the spreadsheet with the **PrintJobName** property.
9. Specify the number to use to begin numbering pages in the spreadsheet using the **PrintFirstPageNumber** property.
10. Specify whether pages print down then across or across then down with the **PrintPageOrder** property.
11. Specify the text to display in an abort dialog box during printing with the **PrintAbortMsg** property.

VBX At run time, customize any of the following print options by setting the corresponding property or properties:

1. Specify the page orientation used to print the spreadsheet with the **PrintOrientation** property.
2. Specify the size of the margins in twips using the **PrintMarginLeft**, **PrintMarginRight**, **PrintMarginTop**, and **PrintMarginBottom** properties.
3. Specify whether to print the spreadsheet border with the **PrintBorder** property.
4. Specify whether to print the spreadsheet column header and row header with the **PrintColHeaders** and **PrintRowHeaders** properties.

5. Specify whether to print the shadow effect within the spreadsheet column header and row header with the **PrintShadows** property.  
 Note that if you set the **PrintColHeaders** and **PrintRowHeaders** properties to False in step 4, the **PrintShadows** property setting will have no effect.
6. Specify whether to print the spreadsheet grid lines with the **PrintGrid** property.
7. Specify whether to print the colors in the spreadsheet as they appear on the screen with the **PrintColor** property.
8. Specify the print job name to display in the Print Manager when printing the spreadsheet with the **PrintJobName** property.
9. Call the **SpreadSetPrintOptions** function and set the following parameters:
  - Set the *FirstPageNumber* parameter to the number to use to begin numbering pages in the spreadsheet.
  - Set the *PageOrder* parameter to specify whether pages print down then across or across then down.
10. Specify the text to display in an abort dialog box during printing with the **PrintAbortMsg** property.

- DLL
1. Using the **SS\_PRINTFORMAT** structure, set the fields to specify
    - Page orientation
    - Left, right, top, and bottom margins
    - Whether pages print down then across or across then down
    - Whether to print the spreadsheet border, column header and row header, shadows in the column and row headers, grid lines, and colors

---

**Note** You can use the **SSSetPrintOptions** function to preset the **SS\_PRINTFORMAT** structure. You should preset the **SS\_PRINTFORMAT** structure if you plan on using the `fpSpreadPreview` control.

---

2. Call the **SSPrint** function, and set the parameters to specify
  - Print job name
  - Abort message box text

## Printing the Portion of the Spreadsheet with Data

You can specify that only the spreadsheet columns and rows that contain data are printed.

You can also set a specific block of cells or range of pages to print. For more information about setting a print range, see “Specifying a Print Range” on page 147.

---

### Note for DLL Users

When printing a spreadsheet that is using virtual mode, in the **SS\_PRINTFORMAT** structure, set the *fUseDataMax* field to **FALSE**.

---



---

### Note for ActiveX, VBX Users

When printing a spreadsheet that is using virtual mode, set the **PrintUseDataMax** property to **False**.

---

#### ► To print the portion of the spreadsheet with data

- |                 |   |
|-----------------|---|
| Spread Designer | In the Spread Designer, <ol style="list-style-type: none"> <li>1. From the General menu, choose Print Properties to display the Print Properties dialog box.</li> <li>2. In the Print Properties dialog box           <ol style="list-style-type: none"> <li>a. Under Print, select the Data Cells Only check box.</li> <li>b. Choose the OK or Apply button.</li> </ol> </li> </ol>  |
| ActiveX, VBX    | At run time, set the <b>PrintUseDataMax</b> property to <b>True</b> .   |
| DLL             | <ol style="list-style-type: none"> <li>1. Using the <b>SS_PRINTFORMAT</b> structure, set the value of the <i>fUseDataMax</i> field to <b>TRUE</b>.</li> </ol> <hr/> <p><b>Note</b> You can use the <b>SSSetPrintOptions</b> function to preset the <b>SS_PRINTFORMAT</b> structure. You should preset the <b>SS_PRINTFORMAT</b> structure if you plan on using the <b>fpSpreadPreview</b> control.</p> <hr/> <ol style="list-style-type: none"> <li>2. Call the <b>SSPrint</b> function.</li> </ol> |

## Specifying a Print Range

You can specify either a block of cells or a range of pages as a print range.

You can also specify that only the spreadsheet columns and rows that contain data are printed. For more information, see “Printing the Portion of the Spreadsheet with Data” on page 146

### ► To specify a block of cells as a print range

Spread  
Designer

In the Spread Designer,

1. Select the block of cells you want to print.
2. From the General menu, choose Print Properties to display the Print Properties dialog box.
3. In the Print Properties dialog box,
  - a. Under Print, clear the Data Cells Only check box.
  - b. Under Range, select the Selected Cells option button.
  - c. Choose the OK or Apply button.

ActiveX, VBX

At run time,

1. Set the **PrintUseDataMax** property to False.
2. Set the **Col**, **Col2**, **Row**, and **Row2** properties to specify the block of cells you want to print.
3. Set the **PrintType** property to 1 (Cell Range).

DLL

1. Using the **SS\_PRINTFORMAT** structure,
  - a. Set the value of the *fUseDataMax* field to FALSE.
  - b. Set the values of the *x1CellRange*, *y1CellRange*, *x2CellRange*, and *y2CellRange* fields to specify the block of cells you want to print.
  - c. Set the value of the *dPrintType* parameter to **SS\_PRINTTYPE\_CELLRANGE**.

---

**Note** You can use the **SSSetPrintOptions** function to preset the **SS\_PRINTFORMAT** structure. You should preset the **SS\_PRINTFORMAT** structure if you plan on using the `fpSpreadPreview` control.

---

2. Call the **SSPrint** function.

### ► To specify a range of pages as a print range

- |                 |  |
|-----------------|--|
| Spread Designer | <p>In the Spread Designer,</p> <ol style="list-style-type: none"><li>1. From the General menu, choose Print Properties to display the Print Properties dialog box.</li><li>2. In the Print Properties dialog box,<ol style="list-style-type: none"><li>a. Under Print, clear the Data Cells Only check box.</li><li>b. Under Range,<ol style="list-style-type: none"><li>i. Select the Pages option button.</li><li>ii. Enter the range of pages in the Pages boxes.</li></ol></li><li>c. Choose the OK or Apply button.</li></ol></li></ol>   |
| ActiveX, VBX    | <p>At run time,</p> <ol style="list-style-type: none"><li>1. Set the <b>PrintUseDataMax</b> property to False.</li><li>2. Specify the range of pages you want to print with the <b>PrintPageStart</b> and <b>PrintPageEnd</b> properties.</li><li>3. Set the <b>PrintType</b> property to 3 (Page Range).</li></ol>  |
| DLL             | <ol style="list-style-type: none"><li>1. Using the <b>SS_PRINTFORMAT</b> structure,<ol style="list-style-type: none"><li>a. Set the value of the <i>fUseDataMax</i> field to FALSE.</li><li>b. Set the values of the <i>nPageStart</i> and <i>nPageEnd</i> fields to specify the range of pages you want to print.</li><li>c. Set the value of the <i>dPrintType</i> parameter to <b>SS_PRINTTYPE_PAGERANGE</b>.</li></ol></li></ol> <hr/> <p><b>Note</b> You can use the <b>SSSetPrintOptions</b> function to preset the <b>SS_PRINTFORMAT</b> structure. You should preset the <b>SS_PRINTFORMAT</b> structure if you plan on using the fpSpreadPreview control.</p> <hr/> |
2. Call the **SSPrint** function.

## Specifying Print Page Breaks

You can set a print page break before a specified column or row. Page breaks are not displayed on the screen, but force page breaks when you print the spreadsheet. A column page break occurs to the left of the specified column; a row page break occurs above the specified row.

You can also retrieve the number of the next column or row in the spreadsheet where a page break occurs.

### ► To specify a print page break before a column or row

Spread  
Designer

In the Spread Designer,

1. If you want to specify a print page break before a column,
  - a. Select the column by clicking the column header or selecting a cell in the column.
  - b. From the General menu, choose Columns and Rows or press the Columns and Rows button () in the tool bar.
  - c. In the Columns / Rows dialog box on the Columns tab, select the Column Page Break check box.
  - d. Choose the OK or Apply button.
2. If you want to specify a print page break before a row,
  - a. Select the row by clicking the row header or selecting a cell in the row.
  - b. From the General menu, choose Columns and Rows or press the Columns and Rows button () in the tool bar.
  - c. In the Columns / Rows dialog box on the Rows tab, select the Row Page Break check box.
  - d. Choose the OK or Apply button.

ActiveX, VBX

1. If you want to specify a print page break before a column, at run time,
  - a. Specify the column with the **Col** property.
  - b. Set the **ColPageBreak** property to True.
2. If you want to specify a print page break before a row, at run time,
  - a. Specify the row with the **Row** property.
  - b. Set the **RowPageBreak** property to True.

## Chapter 7

- DLL
1. If you want to specify a print page break before a column, call the **SSSetColPageBreak** function.
  2. If you want to specify a print page break before a row, call the **SSSetRowPageBreak** function.

► **To retrieve the number of the next column or row where a page break occurs**

ActiveX

At run time,

1. If you want to return the number of the next column or row where a page break occurs, use the **PrintNextPageBreakCol** or **PrintNextPageBreakRow** property.  
The column or row number returned is the first column or row of the next page.
2. If you want to return all columns or rows where a page breaks occurs, loop through the spreadsheet using the **PrintNextPageBreakCol** or **PrintNextPageBreakRow** property.

VBX

At run time,

1. If you want to return the number of the next column or row where a page break occurs, call the **SpreadGetNextPageBreakCol** or **SpreadGetNextPageBreakRow** function.
2. If you want to return all columns or rows where a page break occurs, using a loop,
  - a. Call the **SpreadGetNextPageBreakCol** or **SpreadGetNextPageBreakRow** function and set the *PrevCol* or *PrevRow* parameter to the first column of row to be queried.
  - b. Each time through the loop, set the *PrevCol* or *PrevRow* parameter to the return value from step 2a.  
When  $-1$  is returned, the loop is finished. The value returned is the first column or row of the next page.

DLL

1. If you want to return the number of the next column or row where a page break occurs, call the **SSGetNextPageBreakCol** or **SSGetNextPageBreakRow** function.
2. If you want to return all columns or rows where a page break occurs, using a loop,
  - a. Call the **SSGetNextPageBreakCol** or **SSGetNextPageBreakRow** function and set the *lPrevCol* or *lPrevRow* parameter to the first column of row to be queried.
  - b. Each time through the loop, set the *PrevCol* or *PrevRow* parameter to the return value from step 2a.  
When  $-1$  is returned, the loop is finished. The value returned is the first column or row of the next page.

## Adding Header and Footer Text to Printed Pages

You can specify header and footer text to appear on printed spreadsheet pages.

### ► To add header and footer text to printed spreadsheet pages

Spread  
Designer

In the Spread Designer,

1. From the General menu, choose Print Properties to display the Print Properties dialog box.
2. In the Print Properties dialog box,
  - a. Under Job Setup, type the header and footer text in the Header Text and Footer Text boxes.

You can use control characters to specify header and footer format. For more information, refer to the **PrintHeader** and **PrintFooter** properties in the online *ActiveX/VBX Reference Guide*.

- b. Choose the OK or Apply button.

ActiveX, VBX

At run time, provide the header and footer text using the **PrintHeader** and **PrintFooter** properties.

You can use control characters to specify header and footer format and content. For more information, refer to the **PrintHeader** and **PrintFooter** properties in the online *ActiveX/VBX Reference Guide*.

- DLL
1. Create a callback function to handle the printing of the headers and footers.
  2. Call the **SSPrint** function.



# Chapter 8 Previewing a Printed Spreadsheet

You can use the fpSpreadPreview control to preview your spreadsheet's appearance before you print it. Before using the instructions in this chapter, you should be familiar with the features and concepts described in Chapter 3, "The fpSpreadPreview Control."

This chapter explains how to use the fpSpreadPreview control, as described in the following topics:

Topic	On Page
Previewing Your Spreadsheet	153
Customizing the Page View	154
Zooming In and Out	155
Displaying Scroll Bars	156
Changing the Appearance of the Preview Area	157

## Previewing Your Spreadsheet

To preview your spreadsheet, you link the fpSpread control to the fpSpreadPreview control.

The pages displayed in the fpSpreadPreview control reflect the print options you set for the fpSpread control. These include the SmartPrint option, headers and footers, and page break settings.

---

**Note** If you are using virtual mode, the fpSpreadPreview control displays only the number of rows that are in the virtual buffer.

---

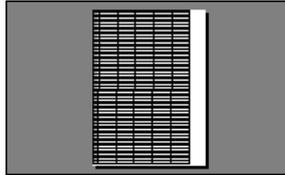
► **To preview your spreadsheet**

ActiveX, VBX At run time, set the **hWndSpread** property to the window handle of the fpSpread control.

DLL Call the **SpvSethWndSpread** function.

## Customizing the Page View

By default, one whole page is displayed in the control.

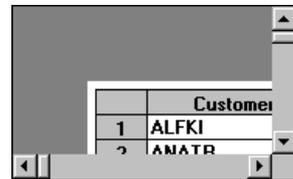


You can change how the page displays the spreadsheet.

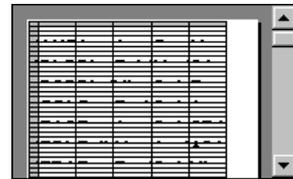
### Page displays . . .

The size of the fpSpread control

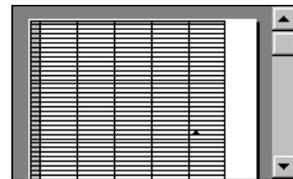
### Representation



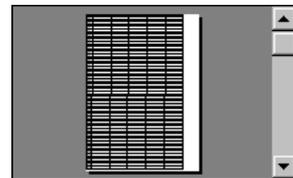
At the percent size specified by the **PageViewPercentage** property (In this example, 20%.)



At the size where the entire width fits in the client area



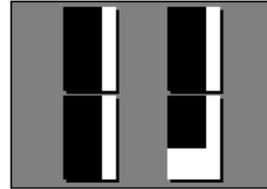
At the size where the entire height fits in the client area



**Page displays . . .**

As one of multiple pages as specified by the **PageMultiCntH** and **PageMultiCntV** properties (In this example, 2 horizontal pages and 2 vertical pages.)

**Representation**



► **To customize the page view**

ActiveX, VBX

1. Specify the way you want the page to display using the **PageViewType** property.
2. If you chose setting 2 (Percentage) in step 1, specify the percentage using the **PageViewPercentage** property.
3. If you chose setting 5 (Multiple Pages) in step 1, specify the number of horizontal pages per view using the **PageMultiCntH** property and the number of vertical pages per view using the **PageMultiCntV** property.

DLL

1. Call the **SpvSetPageViewType** function.
2. If you set the *value* parameter to 2 (SPV\_VIEWTYPE\_PERCENTAGE) in step 1, call the **SpvSetPageViewPercentage** function.
3. If you set the *value* parameter to 5 (SPV\_MULTIPLE\_PAGES) in step 1, call the **SpvSetPageMultiCntH** and the **SpvSetPageMultiCntV** functions.

## Zooming In and Out

By default, the preview page is zoomed out. When the user needs to have a closer view of a particular area of the spreadsheet, you can allow the user to zoom into the area. You can turn off the zoom feature.

You can change the way the control zooms between different views of the displayed pages.

---

**Notes**

- When the **PageViewType** property is set to 2 (Percentage) and the **PageViewPercentage** property is set to some value, the **ZoomState** property is automatically set to 0 (Undefined).
  - You can use the **ZoomState** property to perform the zoom action.
-

### ► To zoom in and out

- ActiveX, VBX
1. To turn off the zoom feature, set the **AllowUserZoom** property to False.
  2. To change the way the control zooms,
    - If you want to zoom in on the page, set the **ZoomState** property to 1 (In).
    - If you want to zoom out on the page, set the **ZoomState** property to 2 (Out).
    - If you want to switch between zooming in and zooming out, set the **ZoomState** property to 3 (Switch).
- DLL
1. To turn off the zoom feature, call the **SpvSetAllowUserZoom** function and set the *value* parameter to 0 (False).
  2. To change the way the control zooms,
    - If you want to zoom in on the page, call the **SpvSetZoomState** property and set the *value* parameter to 1 (SPV\_ZOOMSTATE\_IN).
    - If you want to zoom out on the page, call the **SpvSetZoomState** function and set the *value* parameter to 2 (SPV\_ZOOMSTATE\_OUT).
    - If you want to switch between zooming in and zooming out, call the **SpvSetZoomState** function and set the *value* parameter to 3 (SPV\_ZOOMSTATE\_SWITCH).

## Displaying Scroll Bars

By default, the horizontal and vertical scroll bars are shown only if needed. You can customize either the horizontal or vertical scroll bar, or both, to be displayed at all times or never displayed.

You can also control the amount by which the page scrolls when you click the scroll arrow or move the scroll box on the horizontal or vertical scroll bars.

### ► To display scroll bars

- ActiveX, VBX
1. If you want to hide the scroll bars, set the **ScrollBarH** and **ScrollBarV** properties to 2 (Hide).
  2. If you want to display the scroll bars at all times, set the **ScrollBarH** and **ScrollBarV** properties to 0 (Show).
  3. If you want to specify how far the page scrolls when you click the scroll arrows or move the scroll box on the horizontal and vertical scroll bars, specify the number of units (in twips) the page scrolls with the **ScrollIncH** and **ScrollIncV** properties.

- DLL
1. If you want to hide the scroll bars, call the **SpvSetScrollBarH** and **SpvSetScrollBarV** functions and set the *value* parameter to 2 (SPV\_SCROLLBAR\_HIDE).
  2. If you want to display the scroll bars at all times, call the **SpvSetScrollBarH** and **SpvSetScrollBarV** functions and set the *value* parameter to 0 (SPV\_SCROLLBAR\_SHOW).
  3. If you want to specify how far the page scrolls when you click the scroll arrows or move the scroll box on the horizontal and vertical scroll bars, call the **SpvSetScrollIncH** and **SpvSetScrollIncV** functions.

## Changing the Appearance of the Preview Area

You can customize the following parts of the preview area.

- The color and width of the page shadow
- The color and width of the page border
- The color of the gray area (the area behind the pages within the control border)
- The horizontal and vertical space between the page and the control border
- The horizontal and vertical space between displayed pages (the page gutters)

## Customizing the Page Shadow

In the fpSpreadPreview control, a shadow is displayed behind each page which creates a three-dimensional effect. You can change the width and color of the shadow, or you can hide the shadow.

For a description and example showing the page shadow, see “Appearance of the Preview Area” on page 41.

### ► To customize the page shadow

- ActiveX, VBX
1. Specify the width of the page shadow with the **PageShadowWidth** property.  
If you want to hide the page shadow, set the **PageShadowWidth** property to 0.
  2. Specify the color of the page shadow with the **PageShadowColor** property.
- DLL
1. Call the **SpvSetPageShadowWidth** function to specify the width of the page shadow.  
If you want to hide the page shadow, call the **SpvSetPageShadowWidth** function and set the *value* parameter to 0.
  2. Call the **SpvSetPageShadowColor** function to specify the color of the page shadow.

## Customizing the Page Border

In the fpSpreadPreview control, a border is displayed around each page. You can change the width and color of the border, or you can hide the border.

For a description and example showing the page border, see “Appearance of the Preview Area” on page 41.

### ► To customize the page border

ActiveX, VBX

1. Specify the width of the page border with the **PageBorderWidth** property.  
If you want to hide the page border, set the **PageBorderWidth** property to 0.
2. Specify the color of the page border with the **PageBorderColor** property.

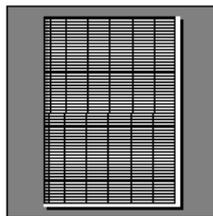
DLL

1. Call the **SpvSetPageBorderWidth** function to specify the width of the page border.  
If you want to hide the page border, call the **SpvSetPageBorderWidth** function and set the *value* parameter to 0.
2. Call the **SpvSetPageBorderColor** function to specify the color of the page border.

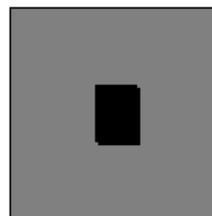
## Customizing the Gray Area

The area behind the control and within the control’s borders is the preview area, or gray area. The space between a displayed page and the edge of the control are the margins. You can change the horizontal margin, the vertical margin, or both.

By default, the size of the gray area depends on the page view type. The horizontal and vertical margins maintain their proportions. You can specify that the gray area display the exact horizontal and vertical gray area margins that you define. In the following example, the page view type is whole page and the horizontal and vertical margins are 720 twips.



Gray area based  
on page view  
type



Gray area based  
on actual margin  
values

You can also specify the color of the gray area.

For a description and illustration showing the gray area and margins, see “Appearance of the Preview Area” on page 41.

► **To customize the gray area**

- |              |   |
|--------------|---|
| ActiveX, VBX | <ol style="list-style-type: none"> <li>1. If you want to specify that the gray area display the exact horizontal and vertical gray area margins, set the <b>GrayAreaMarginType</b> property to 1 (Actual).</li> <li>2. Specify the color of the gray area using the <b>GrayAreaColor</b> property.</li> <li>3. Specify the horizontal and vertical margins in twips using the <b>GrayAreaMarginH</b> and <b>GrayAreaMarginV</b> properties.</li> </ol>  |
| DLL          | <ol style="list-style-type: none"> <li>1. If you want to specify that the gray area display the exact horizontal and vertical gray area margins, call the <b>SpvSetGrayAreaMarginType</b> function and set the <i>value</i> parameter to 1 (SPV_MARGINTYPE_ACTUAL).</li> <li>2. Call the <b>SpvSetGrayAreaColor</b> function to specify the color of the gray area.</li> <li>3. Call the <b>SpvSetGrayAreaMarginH</b> and <b>SpvSetGrayAreaMarginV</b> functions to specify the horizontal and vertical margins.</li> </ol> |

## Changing the Page Gutters

If you display multiple pages (set the **PageViewType** property to 5 (Multiple Pages)), the space between the pages are the gutters. You can change the horizontal gutter, the vertical gutter, or both.

For a description and example showing the horizontal and vertical gutters, see “Appearance of the Preview Area” on page 41.

► **To change the page gutters**

- |              |   |
|--------------|---|
| ActiveX, VBX | Set the <b>PageGutterH</b> and <b>PageGutterV</b> properties.             |
| DLL          | Call the <b>SpvSetPageGutterH</b> and <b>SpvSetPageGutterV</b> functions. |



# Chapter 9 Customizing Columns and Rows

This chapter explains how to work with spreadsheet columns and rows, as described in the following topics:

<b>Topic</b>	<b>On Page</b>
Inserting Columns and Rows	161
Deleting Columns and Rows	164
Sizing Columns and Rows	166
Preventing Resizing of Columns and Rows with the Mouse	170
Setting the Maximum Number of Columns and Rows	172
Displaying a Specified Number of Columns and Rows	173
Designating the Leftmost Column to Display	174
Designating the Topmost Row to Display	174
Hiding Columns and Rows	175
Freezing Columns and Rows	177
Specifying Where the Last Column and Row Stop Scrolling	177
Customizing Column Headers and Row Headers	179
Setting Odd and Even Row Colors	186

## Inserting Columns and Rows

When you insert a column in a spreadsheet, the total number of columns does not increase. Current data in the selected column and in columns to the right shifts right and the last column in the spreadsheet, including data, is deleted. To ensure that the data in the far right column is not deleted, increase the maximum number of columns in the spreadsheet before inserting columns. For more information on changing the maximum number of columns, see “Setting the Maximum Number of Columns and Rows” on page 172.

Similarly, when you insert a row in a spreadsheet, the total number of rows does not increase. Current data in the selected row and in rows below shifts down and the last row in the spreadsheet, including data, is deleted. To ensure that the data in the bottom row is not deleted, increase the maximum number of rows in the spreadsheet before inserting rows. For more information on changing the maximum number of rows, see “Setting the Maximum Number of Columns and Rows” on page 172.

You can insert more than one column or row at a time. The number of columns or rows you select in block mode determines how many columns or rows are added.

When you insert columns or rows, the spreadsheet automatically updates formulas in the spreadsheet. For more information and instructions on turning off automatic updating, see “Automatically Recalculating and Updating Formulas” on page 107.

---

**Note** When you insert columns or rows in the Spread Designer, the total number of columns and rows does increase by the number of columns and rows that you insert.

---

### ► **To insert columns and rows**

Spread  
Designer

In the Spread Designer,

1. If you want to insert a column,
  - a. Select the column before which the new column will be inserted by using one of the following methods:
    - Click the column header.
    - Select a cell in the column.
  - b. From the Edit menu, choose Insert, Column.
2. If you want to add more than one column,
  - a. Select the number of columns by using one of the following methods:
    - Click the column header and drag through the selected number of columns to be added.
    - Select a block of cells across the selected number of columns to be added.
  - b. From the Edit menu, choose Insert, Column.
3. If you want to insert a row,
  - a. Select the row above which the new row will be inserted by using one of the following methods:
    - Click the row header.
    - Select a cell in the row.
  - b. From the Edit menu, choose Insert, Row.

4. If you want to add more than one row,
  - a. Select the number of rows by using one of the following methods:
    - Click the row header and drag through the selected number of rows to be added.
    - Select a block of cells across the selected number of rows to be added.
  - b. From the Edit menu, choose Insert, Row.

## ActiveX, VBX

At run time,

1. If you want to insert a column,
  - a. Set the **Col** property to specify the column before which the new column will be inserted.
  - b. Set the **Action** property to 6 (Insert Col).
2. If you want to add more than one column,
  - a. Set the **Col** property to specify the column before which the new columns will be inserted.
  - b. Set the **Col2** property to indicate the number of the last column to be inserted.  
For example, if you want to insert 3 columns in front of column 2, set the **Col** property to 2 and the **Col2** property to 4.
  - c. Set the **BlockMode** property to True.
  - d. Set the **Action** property to 6 (Insert Col).
  - e. Set the **BlockMode** property back to False.
3. If you want to insert a row,
  - a. Set the **Row** property to specify the row above which the new row will be inserted.
  - b. Set the **Action** property to 7 (Insert Row).
4. If you want to add more than one row,
  - a. Set the **Row** property to specify the row above which the new rows will be inserted.
  - b. Set the **Row2** property to indicate the number of the last row to be inserted.  
For example, if you want to insert 3 rows in front of row 2, set the **Row** property to 2 and the **Row2** property to 4.
  - c. Set the **BlockMode** property to True.

- d. Set the **Action** property to 7 (Insert Row).
  - e. Set the **BlockMode** property back to False.
- DLL
1. If you want to insert a column, call the **SSInsCol** function.
  2. If you want to insert a range of columns, call the **SSInsColRange** function.
  3. If you want to insert a row, call the **SSInsRow** function.
  4. If you want to insert a range of rows, call the **SSInsRowRange** function.

## Deleting Columns and Rows

When you delete a column from a spreadsheet, the total number of columns does not decrease. Current data in the selected column is deleted and data in the columns to the right of the deleted column shifts left. A new blank column appears as the far right column of the spreadsheet. If you want to decrease the number of columns in the spreadsheet, change the maximum number of columns immediately after deleting a column. For more information on changing the maximum number of columns, see “Setting the Maximum Number of Columns and Rows” on page 172.

Similarly, when you delete a row from a spreadsheet, the total number of rows does not decrease. Current data in the selected row is deleted and data in rows below shifts up. A new blank row appears as the bottom row of the spreadsheet. If you want to decrease the number of rows in the spreadsheet, change the maximum number of rows immediately after deleting a row. For more information on changing the maximum number of rows, see “Setting the Maximum Number of Columns and Rows” on page 172.

You can delete more than one column or row at a time. The number of columns or rows you select in block mode determines how many columns or rows are deleted.

When you delete columns or rows, the spreadsheet automatically updates formulas in the spreadsheet. For more information and instructions on turning off automatic updating, see “Automatically Recalculating and Updating Formulas” on page 107.

---

**Note** When you delete columns or rows in the Spread Designer, the total number of columns and rows does decrease by the number of columns and rows that you delete.

---

► **To delete columns and rows**

Spread  
Designer

In the Spread Designer,

1. If you want to delete a column,
  - a. Select the column to be deleted using one of the following methods:
    - Click the column header.
    - Select a cell in the column.
  - b. From the Edit menu, choose Delete, Column.
2. If you want to delete more than one column,
  - a. Select the columns using one of the following methods:
    - Click the first column header and drag to select the columns to be deleted.
    - Select a block of cells across the columns to be deleted.
  - b. From the Edit menu, choose Delete, Column.
3. If you want to delete a row,
  - a. Select the row to be deleted by using one of the following methods:
    - Click the row header.
    - Select a cell in the row.
  - b. From the Edit menu, choose Delete, Row.
4. If you want to delete more than one row,
  - a. Select the rows by using one of the following methods:
    - Click the first row header and drag to select the rows to be deleted.
    - Select a block of cells across the rows to be deleted.
  - b. From the Edit menu, choose Delete, Row.

ActiveX, VBX

At run time,

1. If you want to delete a column,
  - a. Set the **Col** property to specify the column to be deleted.
  - b. Set the **Action** property to 4 (Delete Col).

2. If you want to delete more than one column,
    - a. Set the **Col** property to specify the number of the first column to be deleted.
    - b. Set the **Col2** property to specify the number of the last column to be deleted.

For example, if you want to delete 3 columns starting with column 2, set the **Col** property to 2 and the **Col2** property to 4.
    - c. Set the **BlockMode** property to True.
    - d. Set the **Action** property to 4 (Delete Col).
    - e. Set the **BlockMode** property back to False.
  3. If you want to delete a row,
    - a. Specify the row to be deleted with the **Row** property.
    - b. Set the **Action** property to 5 (Delete Row).
  4. If you want to delete more than one row,
    - a. Set the **Row** property to specify the number of the first row to be deleted.
    - b. Set the **Row2** property to specify the number of the last row to be deleted.

For example, if you want to delete 3 rows starting with row 2, set the **Row** property to 2 and the **Row2** property to 4.
    - c. Set the **BlockMode** property to True.
    - d. Set the **Action** property to 5 (Delete Row).
    - e. Set the **BlockMode** property back to False.
- DLL
1. If you want to delete a column, call the **SSDelCol** function.
  2. If you want to delete a range of columns, call the **SSDelColRange** function.
  3. If you want to delete a row, call the **SSDelRow** function.
  4. If you want to delete a range of rows, call the **SSDelRowRange** function.

## Sizing Columns and Rows

You can change the size of columns and rows using code by setting the column width and row height explicitly or by setting the column width or row height to fit the widest or tallest data. When designing your spreadsheet in the Spread Designer, you can change the column width and row height by dragging using your mouse.

Users can change the size of columns and rows using the mouse in your application; however, you can prevent users from resizing columns and rows with the mouse, if you prefer. For instructions, see “Preventing Resizing of Columns and Rows with the Mouse” on page 170.

## Setting the Width of Columns

You can set the width of a spreadsheet column, including the row headers, to a specific value.

► **To set the width of a column**

Spread  
Designer

In the Spread Designer,

1. Set the specific value of a column by completing the following steps:
  - a. Click on the column you want to resize.
  - b. From the General menu, choose Columns and Rows or press the Columns and Rows button () in the tool bar.
  - c. On the Columns tab in the Columns / Rows dialog box,
    - i. Type the width of the column in the Col Width box.
    - ii. Choose the OK or Apply button.
2. Use the mouse to change the width by completing the following steps:
  - a. Position the pointer over the column header boundary of the column you want to resize.
  - b. When the pointer changes to the resize pointer, press the left mouse button and drag the boundary to resize the column.

ActiveX, VBX

At run time, specify the width with the **ColWidth** property.

Specify the column number in the *Index* parameter of this property. Use the following index values to designate the column width for the following parts of the control:

To set the column height for . . .	Use the index value . . .
Row header	0
All columns	-1
Column specified by the <b>Col</b> property	-2

DLL Call the **SSSetColWidth** or **SSSetColWidthInPixels** function.

To set the width of all columns, set the value of the *Col* parameter to **SS\_ALLCOLS**. To set the width of the row header, set the value of the *Col* parameter to **SS\_HEADER**.

## Setting the Height of Rows

You can set the height of a spreadsheet row, including the column headers, to a specific value.

### ► To set the height of a row

Spread  
Designer

In the Spread Designer,

1. Set the specific value of a row by completing the following steps:
  - a. Click on the row you want to resize.
  - b. From the General menu, choose Columns and Rows or press the Columns and Rows button () in the tool bar.
  - c. On the Rows tab in the Columns / Rows dialog box,
    - i. Type the height of the row in the Row Height box.
    - ii. Choose the OK or Apply button.
2. Use the mouse to set the height by completing the following steps:
  - a. Position the pointer over the row header boundary of the row you want to resize.
  - b. When the pointer changes to the resize pointer, press the left mouse button and drag the boundary to resize the row.

ActiveX, VBX

At run time, specify the row height with the **RowHeight** property.

Specify the row in the *Index* parameter of this property. Index values are zero-based. Use the following index values to designate the row height for the following parts of the control:

To set the row height for . . .	Use the index value . . .
Column header	0
All rows	-1
Row specified by the <b>Row</b> property	-2

DLL Call the **SSSetRowHeight** or **SSSetRowHeightInPixels** function.

To set the height of all rows, set the value of the *Row* parameter to *SS\_ALLROWS*. To set the height of the column header, set the value of the *Row* parameter to *SS\_HEADER*.

## Resizing Columns and Rows to Fit Text

You can resize a column to fit the widest text within that column and you can resize a row to fit the tallest text within that row.

### ► To resize columns and rows to fit text

ActiveX, VBX

At run time,

1. If you want to resize a column to fit the widest text,
  - a. Set the **MaxTextColWidth** property to a variable.  
Specify the column in the *Index* parameter of this property. To specify the row header, enter 0 as the index.
  - b. Set the **ColWidth** property to the variable defined in Step 1.  
To resize the width of all columns to the maximum width of the specified column, set the value of the *Index* parameter to  $-1$  and the value of the **Col** property to  $-1$ . To resize the width of the row header to the maximum width of the specified column, set the value of the *Index* parameter to 0.
2. If you want to resize a row to fit the tallest text,
  - a. Set the **MaxTextRowHeight** property to a variable.  
Specify the row in the *Index* parameter of this property. To specify the column header, enter 0 as the index.
  - b. Set the **RowHeight** property to the variable defined in Step 1.  
To resize the height of all rows to the maximum height of the specified row, set the value of the *Index* parameter to  $-1$  and the value of the **Row** property to  $-1$ . To resize the height of the column header to the maximum height of the specified row, set the value of the *Index* parameter to 0.

- DLL
1. If you want to resize a column to fit text,
    - a. Call the **SSGetMaxTextColWidth** function.
    - b. Call the **SSSetColWidth** or **SSSetColWidthInPixels** function.

To resize the width of all columns to the maximum width of the specified column, set the value of the *Col* parameter to *SS\_ALLCOLS*. To resize the width of the

row header to the maximum width of the specified column, set the value of the *Col* parameter to `SS_HEADER`.

2. If you want to resize a row to fit text,
  - a. Call the `SSGetMaxTextRowHeight` function.
  - b. Call the `SSSetRowHeight` or `SSSetRowHeightInPixels` function.

To resize the height of all rows to the maximum height of the specified row, set the value of the *Row* parameter to `SS_ALLROWS`. To resize the height of the column header to the maximum height of the specified row, set the value of the *Row* parameter to `SS_HEADER`.

## Preventing Resizing of Columns and Rows with the Mouse

By default, users can resize both columns and rows with the mouse at run time. You can prevent resizing of columns, rows, or both. You can also prevent resizing of individual columns or rows.

### ► To prevent resizing of columns and rows

Spread  
Designer

In the Spread Designer,

1. If you want to prevent resizing of all columns,
  - a. From the General menu, choose Headers or press the Headers button () in the tool bar.
  - b. On the Columns tab in the Headers dialog box,
    - i. Clear the User Resize check box under Settings.
    - ii. Choose the OK or Apply button.
2. If you want to prevent resizing of an individual column or multiple columns,
  - a. Select the column or columns you want to prevent from being resized.
  - b. From the General menu, choose Columns and Rows or press the Columns and Rows button () in the tool bar.
  - c. On the Columns tab in the Columns / Rows dialog box,
    - i. Select the Off option button under Resizable.
    - ii. Choose the OK or Apply button.

3. If you want to prevent resizing of all rows,
  - a. From the General menu, choose Headers or press the Headers button () in the tool bar.
  - b. On the Row tab in the Headers dialog box,
    - i. Clear the User Resize check box under Settings.
    - ii. Choose the OK or Apply button.
4. If you want to prevent resizing of an individual row or multiple rows,
  - a. Select the row or rows you want to prevent from being resized.
  - b. From the General menu, choose Columns and Rows or press the Columns and Rows button () in the tool bar.
  - c. On the Rows tab in the Columns / Rows dialog box,
    - i. Select the Off option button under Resizable.
    - ii. Choose the OK or Apply button.

ActiveX, VBX

1. If you want to prevent resizing of columns and rows, set the **UserResize** property to 0.
2. If you want to prevent resizing of columns only, set the **UserResize** property to 2 (Rows).
3. If you want to prevent resizing of rows only, set the **UserResize** property to 1 (Cols).
4. If you want to prevent resizing of an individual column, at run time,
  - a. Specify the column with the **Col** property.
  - b. Set the **UserResizeCol** property to 2 (Off).
5. If you want to prevent resizing of an individual row, at run time,
  - a. Specify the row with the **Row** property.
  - b. Set the **UserResizeRow** property to 2 (Off).

DLL

1. If you want to prevent resizing of both columns and rows, call the **SSSetUserResize** function, and set the value of the *wUserResize* parameter to 0.
2. If you want to prevent resizing of all columns, call the **SSSetUserResize** function, and set the value of the *wUserResize* parameter to either 0 (neither columns nor rows can be resized) or **SS\_USERRESIZE\_ROW** (only rows can be resized).
3. If you want to prevent resizing of a specific column, call the **SSSetUserResizeCol** function.

4. If you want to prevent resizing of all rows, call the **SSSetUserResize** function, and set the value of the *wUserResize* parameter to either 0 (neither columns nor rows can be resized) or **SS\_USERRESIZE\_COL** (only columns can be resized).
5. If you want to prevent resizing of a specific row, call the **SSSetUserResizeRow** function.

## Setting the Maximum Number of Columns and Rows

You can specify the maximum number of columns and rows in a spreadsheet. When you scroll beyond the maximum number of columns or rows, a gray area appears indicating no additional cells exist. You can set the color of the gray area, as described in “Setting the Gray Area Color” on page 287.

You might want to increase or decrease the maximum number of columns and rows when inserting or deleting columns and rows. For more information on inserting and deleting columns and rows, see “Inserting Columns and Rows” on page 161 and “Deleting Columns and Rows” on page 164.

### ► To set the maximum number of columns and rows

Spread  
Designer

In the Spread Designer,

1. From the General menu, choose Columns and Rows or press the Columns and Rows button () in the tool bar.
2. In the Columns / Rows dialog box,
  - a. Do one or both of the following:
    - Set the maximum number of columns on the Columns tab by typing the maximum number of columns in the Max Cols box under Settings.
    - Set the maximum number of rows on the Rows tab by typing the maximum number of rows in the Max Rows box under Settings.
  - b. Choose the OK or Apply button.

ActiveX, VBX

1. Specify the maximum number of columns by setting the **MaxCols** property.
2. Specify the maximum number of rows by setting the **MaxRows** property.

DLL

1. Set the maximum number of columns by calling the **SSSetMaxCols** function.
2. Set the maximum number of rows by calling the **SSSetMaxRows** function.

## Displaying a Specified Number of Columns and Rows

By default, the number of columns and rows that display in a spreadsheet is equal to the maximum number of usable columns and rows. You can define a specific number of columns and rows that are fully displayed in a spreadsheet.

### ► To display a specified number of columns and rows

Spread  
Designer

In the Spread Designer,

1. From the General menu, choose Columns and Rows or press the Columns and Rows button () in the tool bar.
2. In the Columns / Rows dialog box,
  - a. Do either or both of the following:
    - Specify the number of columns to display on the Columns tab by clicking the arrows or typing a value in the Visible Cols spin box under Settings.
    - Specify the number of rows to display on the Rows tab by clicking the arrows or typing a value in the Visible Rows spin box under Settings.
  - b. Choose the OK or Apply button.
3. If you want to prevent partial display of columns and rows,
  - a. From the General menu, choose Spreadsheet Environment or press the Spreadsheet Environment button () in the tool bar.
  - b. On the General tab in the Sheet Environment Settings dialog box,
    - i. Select the Autosize On check box under Settings.
    - ii. Choose the OK or Apply button

ActiveX, VBX

1. Specify the number of columns to display with the **VisibleCols** property.
2. Specify the number of rows to display with the **VisibleRows** property.
3. If you want to prevent partial display of columns and rows, set the **AutoSize** property to True.

DLL

1. Call the **SSSetAutoSizeVisible** function.
2. Call the **SSSetBool** function, and set the parameters as follows:
  - a. Set the value of the *nIndex* parameter to `SSB_AUTOSIZE`.
  - b. Set the value of the *bNewVal* parameter to `TRUE`.

## Designating the Leftmost Column to Display

You can set a specific column to be the far left column in the viewing area.

### ► To designate the leftmost column to display

Spread  
Designer

In the Spread Designer,

1. From the General menu, choose Columns and Rows or press the Columns and Rows button () in the tool bar.
2. On the Columns tab in the Columns / Rows dialog box,
  - a. Click the arrows or type a value in the Left Col spin box under Settings to specify the leftmost column.
  - b. Choose the OK or Apply button.

---

### Notes

- Spread Designer displays the change only in run-time mode.
  - This setting is not preserved when you save your changes to a spreadsheet file or apply your changes to the selected control.
- 

ActiveX, VBX

At run time, set the **LeftCol** property.

DLL

Call the **SSShowCell** function, and set the parameters as follows:

1. Set the value of the *Col* parameter to the number of the leftmost column you want to display.
2. Set the value of the *Row* parameter to `SS_HEADER`.
3. Set the value of the *Position* parameter to `SS_SHOW_TOP_LEFT`.

## Designating the Topmost Row to Display

You can set a specific row as the top scrollable row in the viewing area.

### ► To designate the topmost row to display

Spread  
Designer

In the Spread Designer,

1. From the General menu, choose Columns and Rows or press the Columns and Rows button () in the tool bar.

2. On the Rows tab in the Columns / Rows dialog box,
  - a. Click the arrows or type a value in the Top Row spin box under Settings to specify the topmost row to display.
  - b. Choose the OK or Apply button.

---

### Notes

- Spread Designer displays the change only in run-time mode.
  - This setting is not preserved when you save your changes to a spreadsheet file or apply your changes to the selected control.
- 

ActiveX, VBX      At run time, set the **TopRow** property.

DLL      Call the **SSShowCell** function, and set the parameters as follows:

1. Set the value of the *Col* parameter to SS\_HEADER.
2. Set the value of the *Row* parameter to the number of the topmost row you want to display.
3. Set the value of the *Position* parameter to SS\_SHOW\_TOP\_LEFT.

## Hiding Columns and Rows

You can hide a column or row in a spreadsheet. You can also hide the column headers and row headers, as described in “Hiding the Column and Row Headers” on page 183.

When you hide a column or row, the value of the column width or the row height is “remembered” by the fpSpread control. If you redisplay the column or row, it is displayed at the width or height it was before it was hidden. For example, assume the width of column 1 is 15 twips and you hide the column. Then you redisplay column 1. The width of column 1 is 15 twips.

### ► To hide a column or row

Spread  
Designer

In the Spread Designer,

1. If you want to hide a column,
  - a. Move the focus to the column you want to hide by selecting the column header or a cell within the column.
  - b. From the General menu, choose Columns and Rows or press the Columns and Rows button () in the tool bar.

- c. On the Columns tab in the Columns / Rows dialog box,
  - i. Select the Columns Hidden check box.
  - ii. Choose the OK or Apply button.
2. If you want to hide a row,
  - a. Move the focus to the row you want to hide by selecting the row header or a cell within the row.
  - b. From the General menu, choose Columns and Rows or press the Columns and Rows button () in the tool bar.
  - c. On the Row tab in the Columns / Rows dialog box,
    - i. Select the Row Hidden check box.
    - ii. Choose the OK or Apply button.

---

**Note** In design mode, the spreadsheet displays in a format that assists you in designing the spreadsheet. Even if you hide columns and rows, in design mode the spreadsheet always displays the columns and rows. If you display the spreadsheet in run-time mode and you have hidden the columns or rows, the spreadsheet does not display the columns or rows.

---

### ActiveX, VBX

At run time,

1. If you want to hide a column,
  - a. Specify the column to hide with the **Col** property.
  - b. Set the **ColHidden** property to True.
2. If you want to hide a row,
  - a. Specify the row to hide with the **Row** property.
  - b. Set the **RowHidden** property to True.

### DLL

1. If you want to hide a column, call the **SSShowCol** function, and set the *fShow* parameter to FALSE.
2. If you want to hide a row, call the **SSShowRow** function, and set the *fShow* parameter to FALSE.

## Freezing Columns and Rows

You can freeze (make nonscrollable) any number of columns and rows. The frozen columns are always the far left columns. The frozen rows are always the top rows.

---

**Note** Do not freeze rows in spreadsheets that are using virtual mode.

---

### ► To freeze columns and rows

Spread  
Designer

In the Spread Designer,

1. From the General menu, choose Columns and Rows or press the Columns and Rows button (  ) in the tool bar.
2. In the Columns / Rows dialog box,
  - a. Do one or both of the following:
    - Specify the number of columns you want to freeze on the Columns tab by clicking the arrows or typing a value in the Frozen Cols spin box under Settings.
    - Specify the number of rows you want to freeze on the Rows tab by clicking the arrows or typing a value in the Frozen Rows spin box under Settings.
  - b. Choose the OK or Apply button.

---

**Note** In design mode, the spreadsheet displays in a format that assists you in designing the spreadsheet. Even if you freeze columns and rows, in design mode the frozen columns and rows are scrollable. If you display the spreadsheet in run-time mode and you have frozen the columns or rows, the columns or rows are nonscrollable.

---

- ActiveX, VBX
1. Specify the number of columns to freeze with the **ColsFrozen** property.
  2. Specify the number of rows to freeze with the **RowsFrozen** property.
- DLL
- Call the **SSSetFreeze** function.

## Specifying Where the Last Column and Row Stop Scrolling

By default, the last column and row stop scrolling at the lower-right corner of the spreadsheet. You can specify that the last column and row stop scrolling at the upper-left corner of the spreadsheet.

When the last column and row stop scrolling at the upper-left corner, the gray area of the control is displayed. You can change the color of the gray area, as described in “Setting the Gray Area Color” on page 287.

### ► To specify where the last column and row stop scrolling

- Spread Designer In the Spread Designer,
1. From the Display menu, choose Scroll Bars or press the Scroll Bars button (  ) in the tool bar.
  2. In the Display dialog box,
    - a. Specify where you want the last column and row to stop scrolling by choosing one of the following options on the Scroll Bars tab:
      - If you want the last column and row to stop scrolling at the lower-right corner, select the Align at Last Row and Column check box.
      - If you want the last column and row to scroll to the upper-left corner, clear the Align at Last Row and Column check box.
    - b. Choose the OK or Apply button.
- ActiveX, VBX Set the **ScrollBarMaxAlign** property as follows:
1. If you want the last column and row to stop scrolling at the lower-right corner, set the property to True.
  2. If you want the last column and row to scroll to the upper-left corner, set the property to False.
- DLL Call the **SSSetBool** function, and set the parameters as follows:
1. If you want the last column and row to stop scrolling at the lower-right corner,
    - a. Set the value of the *nIndex* parameter to SSB\_SCROLLBARMAXALIGN.
    - b. Set the value of the *bNewVal* parameter to TRUE.
  2. If you want the last column and row to scroll to the upper-left corner,
    - a. Set the value of the *nIndex* parameter to SSB\_SCROLLBARMAXALIGN.
    - b. Set the value of the *bNewVal* parameter to FALSE.

## Customizing Column Headers and Row Headers

You can define the information displayed in the column headers and row headers. You can also choose not to display these headers.

	A	B	C
1			
2			
3			
4			

By default, header cells are displayed with a three-dimensional appearance, which you can change. You can also specify colors for the shading displayed around the bottom and right of the headers, as well as the background color and the text color.

### Specifying Header Content

The column headers and the row headers can contain sequential letters or sequential numbers. Also, if you prefer, the headers can be blank or display custom text. If the headers do display sequential numbers or letters, you can specify where the sequence starts.

### Specifying Header Content Type

By default, the column headers display letters and the row headers display numbers. You can customize the column headers to display numbers or the row headers to display letters. You can also have the headers appear blank.

If you want the headers to display custom text, see “Specifying Header Text” on page 180.

#### ► To specify the header content type

Spread  
Designer

In the Spread Designer,

1. From the General menu, choose Headers or press the Headers button () in the tool bar.

## Chapter 9

2. In the Headers dialog box,
  - a. Do one or both of the following:
    - On the Columns tab, select the column headers' content from the Header Styles drop-down list box under Headers.
    - On the Rows tab, select the row headers' content from the Header Styles drop-down list box under Headers.
  - b. Choose the OK or Apply button.

ActiveX, VBX Specify the header content as follows:

1. Specify the column headers' content by setting the **ColHeaderDisplay** property.
2. Specify the row headers' content by setting the **RowHeaderDisplay** property.

- DLL
1. Call the **SSSetColHeaderDisplay** function to specify the column headers' content.
  2. Call the **SSSetRowHeaderDisplay** function to specify the row headers' content.

### Specifying Header Text

You can display custom text in the column headers and the row headers (instead of displaying sequential numbers or letters).

#### ► To specify header text

Spread  
Designer

In the Spread Designer for each header for which you want to specify custom text,

1. If you want to provide custom text for the column headers,
  - a. Move the focus to the column to which you want to provide custom text by selecting the column header or a cell within the column.
  - b. From the General menu, choose Headers or press the Headers button () in the tool bar.
  - c. In the Headers dialog box,
    - i. On the Columns tab type the text in the text box under User Defined Headers.
    - ii. Choose the OK or Apply button.

2. If you want to provide custom text for the row headers,
  - a. Move the focus to the row to which you want to provide custom text by selecting the row header or a cell within the row.
  - b. From the General menu, choose Headers or press the Headers button () in the tool bar.
  - c. In the Headers dialog box,
    - i. On the Rows tab type the text in the text box under User Defined Headers.
    - ii. Choose the OK or Apply button.

ActiveX, VBX      At run time,

1. To provide custom text for the column headers,
  - a. If you want to provide custom text for a column header,
    - i. Set the **Row** property to 0.
    - ii. Specify the column with the **Col** property.
    - iii. Specify the text with the **Text** property.
  - b. If you want to provide custom text for a range of column headers,
    - i. Set the **Row** property to 0.
    - ii. Set the **BlockMode** property to True.
    - iii. Select a column range with the **Col** and **Col2** properties.
    - iv. Specify the text with the **Text** property.
    - v. Set the **BlockMode** property back to False.
2. To provide custom text for the row headers,
  - a. If you want to provide custom text for a row header,
    - i. Set the **Col** property to 0.
    - ii. Specify the row with the **Row** property.
    - iii. Specify the text with the **Text** property.
  - b. If you want to provide custom text for a range of row headers,
    - i. Set the **Col** property to 0.
    - ii. Set the **BlockMode** property to True.
    - iii. Select a column range with the **Row** and **Row2** properties.

- iv. Specify the text with the **Text** property.
  - v. Set the **BlockMode** property back to False.
- DLL
1. If you want to set the header text for a specific column or row, call the **SSSetData** function.
  2. If you want to set the header text for a range of columns or rows, call the **SSSetDataRange** function.

### Changing Column and Row Numbering

If the column headers and the row headers display letters or numbers, you can define the starting letter or number. The starting letter or number you specify is used only for display purposes and has no effect on the actual row and column coordinates.

---

**Note** This value is expressed as an integer. For example, if the column headers display letters and you set the starting column letter to 10, the first column header contains a 'J'.

---

#### ► To change column and row numbering

Spread  
Designer

In the Spread Designer,

1. From the General menu, choose Headers or press the Headers button () in the tool bar.
2. In the Headers dialog box,
  - a. Do one or both of the following:
    - Specify the starting column letter or number on the Columns tab by clicking the arrows or typing a value in the Start Col Num spin box under Headers.
    - Specify the starting row letter or number on the Rows tab by clicking the arrows or typing a value in the Start Row Num spin box under Headers.
  - b. Choose the OK or Apply button.

ActiveX, VBX

Specify the starting letter or number as follows:

1. For column headers, set the **StartingColNumber** property.
2. For row headers, set the **StartingRowNumber** property.

DLL Call the **SSSetStartingNumbers** function.

## Hiding the Column and Row Headers

By default, the fpSpread control displays the column headers and the row headers. You can hide the column headers, the row headers, or both.

If your spreadsheet is in virtual mode, the row headers might not display the correct numbers for the rows. You might want to hide the row headers, or set other characteristics to hide header text or display the correct numbers. For more information, see “Customizing Virtual Mode” on page 134.

### ► To hide the column headers and row headers

Spread  
Designer

In the Spread Designer,

1. From the General menu, choose Headers or press the Headers button () in the tool bar.
2. In the Headers dialog box,
  - a. Do one or both of the following:
    - If you want to hide the column headers, on the Columns tab clear the Display Headers check box under Settings.
    - If you want to hide the row headers, on the Rows tab clear the Display Headers check box under Settings.
  - b. Choose the OK or Apply button.

---

**Note** In design mode, the spreadsheet displays in a format that assists you in designing the spreadsheet. Even if you hide headers, in design mode the spreadsheet always displays the headers. If you display the spreadsheet in run-time mode and you have hidden the headers, the spreadsheet does not display the headers.

---

ActiveX, VBX

1. If you want to hide the column headers, set the **DisplayColHeaders** property to False.
2. If you want to hide the row headers, set the **DisplayRowHeaders** property to False.

DLL

Call the **SSSetBool** function and set the parameters as follows:

1. If you want to hide the column headers,
  - a. Set the value of the *nIndex* parameter to SSB\_SHOWCOLHEADERS.
  - b. Set the value of the *bNewVal* parameter to FALSE.

2. If you want to hide the row headers,
  - a. Set the value of the *nIndex* parameter to `SSB_SHOWROWHEADERS`.
  - b. Set the value of the *bNewVal* parameter to `FALSE`.

## Setting the Headers' Three-Dimensional Appearance

By default, header cells are displayed with a raised three-dimensional appearance. You can change the appearance to be flat or to display a lowered three-dimensional appearance.

► **To customize the headers' three-dimensional appearance**

ActiveX, VBX

At run time,

1. Set the **Col**, **Col2**, **Row**, and **Row2** properties to designate the header cells to change.
2. If you are selecting a block of cells, set the **BlockMode** property to `True`.
3. Provide a three-dimensional appearance by doing one of the following:
  - If you want the header cell to display a flat appearance, set the **TypeTextShadow** property to `False`.
  - If you want the cell to display a lowered three-dimensional appearance, set the **TypeTextShadow** property to `False` and set the **TypeTextShadowIn** property to `True`.

---

**Note** If you set both the **TypeTextShadow** and **TypeTextShadowIn** properties to `True`, the cell displays a raised shadow effect as specified by the **TypeTextShadow** property.

---

4. If you selected a block of cells, set the **BlockMode** property back to `False`.

DLL

1. Call the **SSSetTypeStaticText** function and set the *TextStyle* parameter to `0` to display a flat appearance or to `SS_TEXT_SHADOWIN` to display a lowered three-dimensional appearance.
2. Call the **SSSetCellType** or **SSSetCellTypeRange** function and set the *CellType* parameter to `SS_TYPE_STATICTEXT`.

## Setting the Header Shadow Color

By default, header cells display shading along the bottom and right edge of the cells. The shading is dark gray, but you can change the shadow color. Setting the shadow color for header cells sets the color for both column headers and row headers.

---

**Note** Headers are static text cells. If you change the shadow color and you have static text cells in your spreadsheet that display a three-dimensional shadow, their shadow color will also change.

---

### ► To customize the headers' shadow color

Spread  
Designer

In the Spread Designer,

1. From the General menu, choose Spreadsheet Colors or press the Spreadsheet Colors button (  ) in the tool bar.
2. On the Colors tab in the Default dialog box,
  - a. Select the **ShadowDark** property from the Property drop-down list box.
  - b. Choose the Color button to display the Color dialog box.
  - c. In the Color dialog box,
    - i. Select a basic color or customize your own color.
    - ii. Choose the OK button.
  - d. Choose the OK or Apply button.

ActiveX, VBX Set the **ShadowDark** property.

DLL Call the **SSSetShadowColor** function.

## Setting the Header Background and Text Colors

You can set the colors of the background and text in the headers. When you set the background and text colors for the headers, you set the colors for both the column headers and the row headers.

---

**Note** Headers are static text cells. If you change the background or text color and you have static text cells in your spreadsheet that display a three-dimensional appearance, their background or text color will also change.

---

---

**Tip** If you want to change the background or text color of an individual header, in code at run time, change the cell type of that header to button and use the **TypeButtonColor** and **TypeButtonTextColor** properties to specify the background and text colors.

---

► **To set the background and text colors for headers**

Spread  
Designer

In the Spread Designer,

1. From the General menu, choose Spreadsheet Colors or press the Spreadsheet Colors button () in the tool bar.
2. On the Colors tab in the Default dialog box,
  - a. Do one of the following:
    - Specify the background color by selecting the **ShadowColor** property from the Property drop-down list box.
    - Specify the text color by selecting the **ShadowText** property from the Property drop-down list box.
  - b. Choose the Color button to display the Color dialog box.
  - c. In the Color dialog box,
    - i. Select a basic color or customize your own color.
    - ii. Choose the OK button.
  - d. Choose the OK or Apply button.

ActiveX, VBX

1. Specify the background color with the **ShadowColor** property.
2. Specify the text color with the **ShadowText** property.

DLL

Call the **SSSetShadowColor** function.

## Setting Odd and Even Row Colors

You can set the background and text color for both odd and even rows. For example, you could set the background color of odd rows to light gray or you could set the text color of even rows to red.

► **To set the odd and even row colors**

Spread  
Designer

In the Spread Designer,

1. From the General menu, choose Spreadsheet Colors or press the Spreadsheet Colors button (  ) in the tool bar.
2. On the Colors tab in the Default dialog box,
  - a. Specify the background color of the even rows by selecting the **EvenRowBackColor** from the Property drop-down list box.
  - b. Choose the Color button to display the Color dialog box.
  - c. In the Color dialog box,
    - i. Select a basic color or customize your own color.
    - ii. Choose the OK button.
  - d. Choose the Apply button.
  - e. Specify the foreground color of the even rows by selecting the **EvenRowForeColor** from the Property drop-down list box.
  - f. Choose the Color button to display the Color dialog box.
  - g. In the Color dialog box,
    - i. Select a basic color or customize your own color.
    - ii. Choose the OK button.
  - h. Choose the Apply button.
  - i. Specify the background color of the odd rows by selecting the **OddRowBackColor** from the Property drop-down list box.
  - j. Choose the Color button to display the Color dialog box.
  - k. In the Color dialog box,
    - i. Select a basic color or customize your own color.
    - ii. Choose the OK button.
  - l. Choose the Apply button.
  - m. Specify the foreground color of the odd rows by selecting the **OddRowForeColor** from the Property drop-down list box.
  - n. Choose the Color button to display the Color dialog box.

## Chapter 9

- o. In the Color dialog box,
  - i. Select a basic color or customize your own color.
  - ii. Choose the OK button.
- p. Choose the OK or Apply button.

- ActiveX Call the **SetOddEvenRowColor** method.
- VBX Call the **SetSpreadOddEvenRowColor** function.
- DLL Call the **SSSetOddEvenRowColor** function.

# Chapter 10 Working with Cells

This chapter describes how to work with cells in the spreadsheet, as described in the following topics:

Topic	On Page
Working with the Active Cell	189
Working with a Range of Cells	190
Selecting Multiple Blocks of Cells	191
Locking Cells	192
Resizing Cells	196
Allowing Cells to Overflow	198
Using Edit Mode	199
Customizing Cell Borders	202
Setting Cell Background and Text Colors	205
Setting the Font	207
Specifying the Background and Text Color of Selected Cells	207

You can set cells to be one of twelve cell types. For instructions and information about cell types, see Chapter 11, “Setting the Cell Type.” You might want to review this chapter before setting cell types, though, to find out more about selecting and working with cells.

## Working with the Active Cell

The active cell is the cell that currently receives any user interaction. You can specify the active cell and you can return the coordinates of the current active cell.

Usually, the active cell displays a focus rectangle. You can hide the focus rectangle, if you prefer, as described in “Hiding the Focus Rectangle” on page 300.

### ► To specify the active cell

Spread  
Designer

In the Spread Designer, do either of the following:

- Click the left mouse button when the pointer is over the cell that you want to be the active cell.
- Move the active cell by pressing the left, right, up, or down arrow keys.

## Chapter 10

- ActiveX, VBX     At run time,
1. Specify the cell coordinates with the **Col** and **Row** properties.
  2. Set the **Action** property to 0 (Activate Cell).
- DLL     Call the **SSSetActiveCell** function.
- ▶     **To return the coordinates of the active cell**
- ActiveX, VBX     At run time, return the values of the **ActiveCol** and **ActiveRow** properties, respectively.
- DLL     Call the **SSGetActiveCell** function.

## Working with a Range of Cells

---

**Note** This section applies to ActiveX and VBX users only. DLL users can work with blocks of cells by using parameters provided in appropriate functions.

---

Use the **Col**, **Col2**, **Row**, and **Row2** properties to define a block of cells in the spreadsheet. Then use the **BlockMode** property to specify that you want to work with a block of cells.

You can also work with individual cells, columns, rows, or the entire spreadsheet by using the appropriate settings of the **Col** and **Row** properties without setting the **BlockMode** property. For more information, see “Selecting Spreadsheet Elements” on page 31.

The instructions throughout Part II, “How-to Guides,” provide information about which spreadsheet elements you can select and how to select them. Keep in mind that any task for which you can set the **Col** and **Row** properties will let you work with columns, rows, or the entire spreadsheet as well.

▶     **To work with a block of cells**

- ActiveX, VBX     At run time,
1. Set the **Col** and **Col2** properties.
  2. Set the **Row** and **Row2** properties.
  3. Set the **BlockMode** property to True.
  4. Set the properties you need to accomplish your task.
  5. Set the **BlockMode** property back to False.

## Selecting Multiple Blocks of Cells

Sometimes you or users might want to select several blocks of cells that are not adjacent to one another and therefore cannot be selected as one group. You can select multiple, discontinuous blocks of cells, and you can let users select multiple discontinuous blocks as well.

If you allow users to select multiple, discontinuous blocks of cells, to select the blocks, they first select a block of cells. Then, they press the Ctrl key and select additional blocks using the mouse.

For more information on how to perform an action on multiple blocks of cells, see the example for **Action** property setting 18 in the online *ActiveX/VBX Reference Guide* or the **SSGetMultiSelBlocks** function example in the online *DLL Reference Guide*.

### ► To select multiple blocks of cells in the Spread Designer

Spread  
Designer

In the Spread Designer,

1. From the General menu, choose Spreadsheet Environment or press the Spreadsheet Environment button (  ) in the toolbar.
2. On the General tab in the Sheet Environment Settings dialog box, select the Multiple Block Selections check box under Settings.
3. Choose the OK button.
4. In the spreadsheet view or the active view, select a block of cells using the mouse.
5. Press Ctrl and select additional blocks using the mouse.

### ► To select multiple blocks of cells using code

ActiveX, VBX

At run time,

1. Set the **AllowMultiBlocks** property to True.
2. Select a cell or a cell range with the **Col**, **Col2**, **Row**, and **Row2** properties.
3. Set the **Action** property to 17 (Add MultiSel Blocks).
4. Repeat steps 2 and 3 until all blocks are selected.

DLL

1. Call the **SSSetBool** function, and set the parameters as follows:
  - a. Set the value of the *nIndex* parameter to SSB\_ALLOWMULTIBLOCKS.
  - b. Set the value of the *bNewVal* parameter to TRUE.

2. Call the **SSAddMultiSelBlocks** function.
3. Repeat step 2 until all blocks are selected.

## Locking Cells

You can lock cells to prevent editing. To do so, you first mark cells as locked, then tell the spreadsheet to lock the marked cells. You can remove the locked status from marked cells to edit them, and you can change cells so that they are no longer marked as locked.

To indicate that cells are marked as locked, you can change their background and text colors. The settings for the background and text colors for cells marked as locked override the settings for the individual cell, the column, the row, or the spreadsheet.

## Marking Cells As Locked

You can mark an individual cell or a range of cells as locked. Cells that are marked as locked are not actually locked until you lock the spreadsheet. If the spreadsheet is locked, cells marked as locked are locked and the user cannot edit them.

You can change cells that are marked so that they are no longer marked as locked. For instructions, see “Clearing Cells That Are Marked as Locked” on page 194.

► **To mark cells as locked**

Spread  
Designer

In the Spread Designer,

1. Select a cell or block of cells.
2. From the Cell menu, choose Lock.
3. Repeat for additional cells you want to mark as locked.

---

**Note** In design mode, the spreadsheet displays in a format that assists you in designing the spreadsheet. Even if you lock cells, in design mode you can still edit the locked cells. If you display the spreadsheet in run-time mode and you have locked cells, you cannot edit those cells.

---

ActiveX, VBX

At run time,

1. Set the **Col**, **Col2**, **Row**, and **Row2** properties as appropriate.

For more information about setting the **Col** and **Row** properties, see “Relationship of Col and Row Properties or Parameters” on page 31.

2. If you are selecting a block of cells, set the **BlockMode** property to True.

3. Set the **Lock** property to True.
  4. If you selected a block of cells, set the **BlockMode** property back to False.
  5. Repeat for additional cells you want to mark as locked.
- DLL
1. Call the **SSSetLock** or **SSSetLockRange** function, and set the parameters as follows:
    - a. Set the values of the *Col*, *Row*, *Col2*, and *Row2* parameters as appropriate.
    - b. Set the value of the *Lock* parameter to TRUE.
  2. Repeat for additional cells you want to mark as locked.

## Locking the Spreadsheet

When you have finished marking cells as locked and you are ready to prevent users from editing the marked cells, lock the spreadsheet. The cells that you have marked as locked then become locked cells.

You can unlock the spreadsheet to allow editing of cells that are marked as locked. For instructions, see “Unlocking the Spreadsheet” on page 194.

### ► To lock the spreadsheet

Spread  
Designer

In the Spread Designer,

1. From the General menu, choose Spreadsheet Environment or press the Spreadsheet Environment button () in the toolbar.
2. On the General tab in the Sheet Environment Settings dialog box,
  - a. Select the Protect Locked Cells check box under Settings.
  - b. Choose the OK or Apply button.

---

**Note** If you are working in design mode, you can still edit the locked cells, and colors you set for locked cells are not displayed. When you change to run-time mode, you cannot edit the locked cells, and the locked cells display any colors you have set for locked cells.

---

ActiveX, VBX At run time, set the **Protect** property to True.

- DLL Call the **SSSetBool** function, and set the parameters as follows:
1. Set the value of the *nIndex* parameter to SSB\_PROTECT.
  2. Set the value of the *bNewVal* parameter to TRUE.

## Clearing Cells That Are Marked as Locked

You can change cells that are marked as long so that they are no longer locked.

► **To remove marks from cells that are marked as locked**

Spread  
Designer

In the Spread Designer,

1. Select the cells or block of cells that are marked as locked.
2. From the Cell menu, choose Unlock.

ActiveX, VBX

At run time,

1. Set the **Col**, **Col2**, **Row**, and **Row2** properties as appropriate.

For more information about setting the **Col** and **Row** properties, see “Relationship of Col and Row Properties or Parameters” on page 31.

2. If you are selecting a block of cells, set the **BlockMode** property to True.
3. Set the **Lock** property to False.
4. If you selected a block of cells, set the **BlockMode** property back to False.

DLL

Call the **SSSetLock** or **SSSetLockRange** function, and set the parameters as follows:

1. Set the values of the *Col*, *Row*, *Col2*, and *Row2* parameters as appropriate.
2. Set the value of the *Lock* parameter to FALSE.

## Unlocking the Spreadsheet

You can unlock the spreadsheet to allow cells that are marked as locked to be edited.

► **To unlock the spreadsheet**

Spread  
Designer

In the Spread Designer,

1. From the General menu, choose Spreadsheet Environment or press the Spreadsheet Environment button () in the toolbar.
2. On the General tab in the Sheet Environment Settings dialog box,
  - a. Clear the Protect Locked Cells check box under Settings.
  - b. Choose the OK or Apply button.

ActiveX, VBX

Set the **Protect** property to False.

DLL Call the **SSSetBool** function, and set the parameters as follows:

1. Set the value of the *nIndex* parameter to `SSB_PROTECT`.
2. Set the value of the *bNewVal* parameter to `FALSE`.

## Setting Background and Text Colors for Locked Cells

By default, the background and text colors of cells that are marked as locked and of locked cells are the same as the background and text colors of other cells. You can change the background and text colors of cells that are marked as locked and of locked cells.

The background and text colors set for cells that are marked as locked and for locked cells override the colors set for the cell, the column, the row, or the spreadsheet.

---

**Note** You cannot change the background color of the drop-down portion of a combo box cell. The background color of the drop-down portion of a combo box cell is always white.

---

### ► To set the background and text colors of locked cells

Spread  
Designer

In the Spread Designer,

1. From the General menu, choose Spreadsheet Colors or press the Spreadsheet Colors button () in the tool bar.
2. On the Colors tab in the Default dialog box,
  - a. Do one of the following:
    - Specify the background color by selecting the **LockBackColor** property from the Property drop-down list box.
    - Specify the text color by selecting the **LockForeColor** property from the Property drop-down list box.
  - b. Choose the Color button to display the Color dialog box.
  - c. In the Color dialog box,
    - i. Select a basic color or customize your own color.
    - ii. Choose the OK button.
  - d. Choose the OK or Apply button.

## Chapter 10

- ActiveX, VBX    At run time,
1. Specify the background color by setting the **LockBackColor** property.
  2. Specify the text color by setting the **LockForeColor** property.
- DLL    Call the **SSSetLockColor** function.

## Resizing Cells

You cannot scroll through text in a cell. However, you can resize a cell to fit the widest or tallest data in that cell. When you change the width and height of a cell, the column width and row height also change.

You can base the column width and row height on either of the following:

- The width and height of the data in a particular cell
- The widest data in a column and the tallest data in a row

---

**Note** If you base the column width and row height on the width and height of a particular cell and you later enter data that is wider or taller in another cell in the same row or column, the data will be truncated.

---

### Resizing a Cell to Fit the Widest Data

If you resize a cell to fit the widest data, the column width changes accordingly.

► **To resize a cell based on the widest data in that cell**

- ActiveX, VBX    At run time,
1. Define the cell you want to resize with the **Col** and **Row** properties.
  2. Return the minimum width necessary to display all the text within that cell using the **MaxTextCellWidth** property and assign it to a variable.
  3. Set the **ColWidth** property to the variable from step 2.
- DLL    1. Call the **SSGetMaxTextCellSize** function.
2. Call the **SSSetColWidth** or **SSSetColWidthInPixels** function, and set the parameters as follows:
    - a. Set the value of the *Col* parameter to the number of the column you want to resize.
    - b. Set the value of the *dWidth* (*dWidth* for **SSSetColWidthInPixels**) parameter to the text width returned in step 1.

► **To resize a cell based on the widest data in that column**

ActiveX, VBX

At run time,

1. Return the minimum width necessary to display the widest data in a specific column with the **MaxTextColWidth** property (specify the column using the *Index* parameter) and assign it to a variable.
2. Set the **ColWidth** property for that column equal to the variable from step 1.

DLL

1. Call the **SSGetMaxTextColWidth** function.
2. Call the **SSSetColWidth** or **SSSetColWidthInPixels** function, and set the parameters as follows:
  - a. Set the value of the *Col* parameter to the number of the column you want to resize.
  - b. Set the value of the *dfWidth* (*dWidth* for **SSSetColWidthInPixels**) parameter to the text width returned in step 1.

## Resizing a Cell to Fit the Tallest Data

If you resize a cell to fit the tallest data, the row height changes accordingly.

► **To resize a cell based on the tallest data in that cell**

ActiveX, VBX

At run time,

1. Define the cell you want to resize with the **Col** and **Row** properties.
2. Return the minimum height necessary to display all the text within that cell with the **MaxTextCellHeight** property and assign it to a variable.
3. Set the **RowHeight** property to the variable from step 2.

DLL

1. Call the **SSGetMaxTextCellSize** function.
2. Call the **SSSetRowHeight** or **SSSetRowHeightInPixels** function, and set the parameters as follows:
  - a. Set the value of the *Row* parameter to the number of the row you want to resize.
  - b. Set the value of the *dfHeight* (*dHeight* for **SSSetRowHeightInPixels**) parameter to the maximum text height returned in step 1.

► **To resize a cell based on the tallest data in that row**

ActiveX, VBX

At run time,

1. Return the minimum height necessary to display the tallest data in a specific row with the **MaxTextRowHeight** property (specify the row using the *Index* parameter) and assign it to a variable.
2. Set the **RowHeight** property for that row equal to the variable from step 1.

DLL

1. Call the **SSGetMaxTextRowHeight** function.
2. Call the **SSSetRowHeight** or **SSSetRowHeightInPixels** function, and set the parameters as follows:
  - a. Set the value of the *Row* parameter to the number of the row you want to resize.
  - b. Set the value of the *dHeight* (*dHeight* for **SSSetRowHeightInPixels**) parameter to the maximum text height returned in step 1.

## Allowing Cells to Overflow

You can specify that the contents of a cell can overflow into an adjacent cell if that cell is empty. If you allow cell overflow,

- Left-aligned text in a cell overflows to the adjacent right cell.
- Right-aligned text in a cell overflows to the adjacent left cell.
- When text is centered in a cell, text overflows to both the left and right adjacent cells.

In the following figure, the text in cell C1 is left-aligned and overflows. The first character of the cell text is aligned with the left border of C1. The text in cell C3 is right-aligned and overflows. The last character of the cell text is aligned with the right border of C3. The text in cell C5 is centered and overflows. The middle character of the cell text is centered in the cell that contains the original text.

	A	B	C	D	E	
1			Cell text is allowed to overflow.			▲
2						
3	Cell text is allowed to overflow.					
4						
5		Cell text is allowed to overflow.				
6						▼
◀						▶

If you want the text to overflow into a specific block of cells, you must look at how the text is aligned in the cell (set with the **TypeHAlign** property) and the cell in which the original text resides. For example, assume you want the example text shown in the previous figure to be displayed across columns A, B, and C. If the text is left-aligned, the cell text should originate in cell A1. If the text is right-aligned, the cell text should originate in cell C3. If the text is centered, the cell text should originate in cell B5.

	A	B	C	D	E
1	Cell text is allowed to overflow.				
2					
3		Cell text is allowed to overflow.			
4					
5	Cell text is allowed to overflow.				
6					

► **To allow the contents of a cell to overflow into an adjacent empty cell**

Spread  
Designer

In the Spread Designer,

1. From the General menu, choose Spreadsheet Environment or press the Spreadsheet Environment button (  ) in the toolbar.
2. On the General tab in the Sheet Environment Settings dialog box,
  - a. Select the Cell Overflow check box under Settings.
  - b. Choose the OK or Apply button.

ActiveX, VBX

Set the **AllowCellOverflow** property to True.

DLL

Call the **SSSetBool** function, and set the parameters as follows:

1. Set the value of the *nIndex* parameter to `SSB_ALLOWCELLOVERFLOW`.
2. Set the value of the *bNewVal* parameter to `TRUE`.

## Using Edit Mode

When the user enters data in a cell in the spreadsheet or double clicks a cell, the spreadsheet automatically turns edit mode on. When the user moves from the cell, the spreadsheet automatically turns edit mode off. If edit mode is on, the active cell is the cell being edited.

---

**Notes**

- Picture cells, static text cells, and locked cells cannot be edited.
  - For ActiveX, VBX users, the **EditMode** event with the *Mode* parameter set to 1 (True), is sent to the application each time the spreadsheet turns edit mode on. The **EditMode** event, with the *Mode* parameter set to 0 (False), is sent to the application each time the spreadsheet turns edit mode off. An application might want to know when the user leaves edit mode so that validation can be performed on the data.
  - For DLL users, the fpSpread control sends its owner the **SSM\_EDITMODEON** message when the spreadsheet turns edit mode on, and it sends the **SSM\_EDITMODEOFF** message when the spreadsheet turns edit mode off. An application might want to know when the user leaves edit mode so that validation can be performed on the data.
- 

You can tell the spreadsheet to turn edit mode on or off using code. If you prefer, you can turn edit mode on all the time. In addition, you can specify what happens to existing text in the cell when the spreadsheet turns edit mode on.

## Setting the Edit Mode

You can tell the spreadsheet to turn edit mode on or off using code.

### ► To turn edit mode on and off using code

ActiveX, VBX

At run time, set the **EditMode** property as follows:

- If you want to turn edit mode off, set the property to False.
- If you want to turn edit mode on, set the property to True.

DLL

Call the **SSSetEditMode** function, and set the parameters as follows:

- If you want to turn edit mode off, set the value of the *fEditModeOn* parameter to FALSE.
- If you want to turn edit mode on, set the value of the *fEditModeOn* parameter to TRUE.

## Keeping Edit Mode Permanently On

You can specify that edit mode remains on when the user switches between cells. You might want to do this to emulate a table of edit controls.

When edit mode remains on, no selections are possible.

---

**Notes**

- For DLL users, when the **SSSetBool** function's `SSB_EDITMODEPERMANENT` option is `TRUE`, it overrides the **SSSetEditMode** function, and the focus rectangle is hidden (see “Hiding the Focus Rectangle” on page 300).
  - For ActiveX, VBX users, when the **EditModePermanent** property is set to `True`, it overrides the **EditMode** property and the focus rectangle is hidden (see “Hiding the Focus Rectangle” on page 300).
- 

► **To specify that edit mode stays on**

Spread  
Designer

In the Spread Designer,

1. From the General menu, choose Spreadsheet Environment or press the Spreadsheet Environment button () in the toolbar.
2. On the General tab in the Sheet Environment Settings dialog box,
  - a. Select the Make Edit Mode Permanent check box under Settings.
  - b. Choose the OK or Apply button.

ActiveX, VBX

Set the **EditModePermanent** property to `True`.

DLL

Call the **SSSetBool** function, and set the parameters as follows:

1. Set the value of the *nIndex* parameter to `SSB_EDITMODEPERMANENT`.
2. Set the value of the *bNewVal* parameter to `TRUE`.

## Replacing Text as You Edit

By default, when the spreadsheet turns edit mode on, the cursor appears after the text in the active cell. If you prefer, you can have the spreadsheet select the text in the active cell. You might want to do this if you think users will want to replace the text in the cell, rather than append text to it.

### ► To replace text during edit mode

- Spread Designer In the Spread Designer,
1. From the General menu, choose Spreadsheet Environment or press the Spreadsheet Environment button () in the toolbar.
  2. On the General tab in the Sheet Environment Settings dialog box,
    - a. Select the Replace Existing Text check box. under Settings
    - b. Choose the OK or Apply button.
- ActiveX, VBX Set the **EditModeReplace** property to True.
- DLL Call the **SSSetBool** function, and set the parameters as follows:
1. Set the value of the *nIndex* parameter to `SSB_EDITMODEREPLACE`.
  2. Set the value of the *bNewVal* parameter to `TRUE`.

## Customizing Cell Borders

You can specify whether a cell or group of cells has a border. A group of cells can be a row, a column, a range of cells, or an entire spreadsheet. If you have a border, it can be displayed on the left, right, top, or bottom, or around all four sides of a cell or cell range.

A border can be displayed as a solid, dashed, dotted, dash-dot, or dash-dot-dot line in regular and fine line weights. By default, no border is displayed.

You can customize the border color of a cell, a column, a row, a block of cells, or the entire spreadsheet. If you want to specify the border color for a button cell, see “Setting the Border Color for Button Cells” on page 215.

You can specify more than one style and color for the same cell, column, row, or block of cells. The cell borders are drawn from left to right and top to bottom in the spreadsheet. If two adjacent borders have a different style or color, the last one drawn has precedence and is the one that is displayed.

Cell borders reflect the precedence used by the spreadsheet to determine the characteristics for spreadsheet elements. That is, cell settings override row, column, and spreadsheet settings, in that order. For more information, see “Cell, Row, Column, and Spreadsheet Precedence” on page 32.

---

**Notes**

- If two adjacent cells have different settings, the cell that is to the right or to the bottom has precedence.

Keep in mind that the spreadsheet is drawn from left to right and from top to bottom on the screen. Each subsequent cell's border properties take precedence over the cell drawn before it.

- In the following example, a dotted, black border is set for each cell in the first five rows of column 2. Then a solid, blue border is set for the block B2:C4. Notice that the top and left sides of cell B2 and the left side of cells B3 and B4 display a solid, blue border. For these borders, the solid, blue border has precedence because it was set last. Also notice that cell B5 displays a dotted, black border on the top. Although the solid, blue border for the adjacent cell (the bottom border of cell B4) was set after the dotted, black border for the cell, cell B5 is drawn after cell B4.

	A	B	C	
1				
2				
3				
4				
5				
6				
7				
8				

- If you want to return the border style or border color for a cell, you must first specify the side for which you want to return the style or color by setting the **CellBorderStyle** property. Then use the **CellBorderStyle** or **CellBorderColor** property to return the style or color.

---

Borders are different from grid lines in that they create a border around a cell or group of cells rather than distinguishing rows and columns. For more information about grid lines, see “Customizing Grid Lines” on page 283.

---

**Note** The top and left borders of the cell displayed in the top-left corner of the spreadsheet are not displayed. Keep this in mind if you choose not to display a border for the Spread control or headers for the spreadsheet, as the result might be visually confusing.

---

### ► To customize cell borders

Spread  
Designer

In the Spread Designer,

1. Select the cell or cells for which you want to customize the borders.
2. From the Cell menu choose Border Lines or press the Border Lines button () in the tool bar.
3. On the Border Lines tab in the Cell Settings dialog box,
  - a. Specify whether you want to customize a border for all cells in a block or for individual cells by selecting an option button under Border Around.
  - b. Select which side of the cell or block border you want to customize from the Border Side drop-down list box.
  - c. Choose the Color button to display the Color dialog box.
  - d. In the Color dialog box,
    - i. Select a basic color or customize your own color.
    - ii. Choose the Apply button.
  - e. Specify the style of the border by selecting an option button under Style.  
For example, if you want to display a fine border line, select the Fine check box.
  - f. Choose the OK or Apply button.

ActiveX, VBX

At run time,

1. Set the **Col**, **Col2**, **Row**, and **Row2** properties as appropriate.  
For more information about setting the **Col** and **Row** properties, see “Relationship of Col and Row Properties or Parameters” on page 31.
2. If you are selecting a block of cells, set the **BlockMode** property to True.
3. Set the **CellBorderStyle** property as follows:
  - If you do not want to display a border around the cell or cell range, set the property to 0 (None).
  - If you want to display a border around the cell or each cell in the range, set the property to one or more of the available settings, using the Or operator to combine values 1 (Left), 2 (Right), 4 (Top), or 8 (Bottom).
  - If you want to display a border around a range of cells, set the property to setting 16 (Outline).

4. Specify the border style with the **CellBorderStyle** property.  
If you want the cell or cell range to display the border style of the next element with precedence (the row, the column, or the spreadsheet), set the property to 0 (Default).
5. Specify the border color of the selected cells with the **CellBorderColor** property.
6. Set the **Action** property to 16 (Set Border).
7. If you selected a block of cells, set the **BlockMode** property back to False.

DLL Call the **SSSetBorder** or **SSSetBorderRange** function and set the *Col*, *Row*, *Col2*, *Row2*, *wBorderStyle*, *wBorderStyle*, and *Color* parameters.

## Setting Cell Background and Text Colors

You can change the background and text colors of a cell, a column, a row, a block of cells, or the entire spreadsheet.

---

**Note** You cannot change the background color of the drop-down list portion of a combo box cell. The background color of the drop-down list portion of a combo box cell is always white.

---

By default, the background color of a cell covers the right and bottom sides of the cell's grid lines. You can specify that control displays the complete grid lines instead, or that the color covers all the grid lines. For more information, see "Customizing Grid Lines" on page 283.

If you want to specify the background and text colors of selected cells, see "Specifying the Background and Text Color of Selected Cells" on page 207. If you want to specify the background and text colors for button cells, see "Setting Background and Text Colors of Button Cells" on page 218. If you want to specify the background and text colors for locked cells, see "Setting Background and Text Colors for Locked Cells" on page 195. If you want to specify the background color for a static text cell that displays a three-dimensional appearance, see "Creating and Customizing a Static Text Cell" on page 258.

### ► To set the background and text colors of cells

Spread  
Designer

In the Spread Designer,

1. If you want to change the background and text colors
  - Of a single cell, select the cell.
  - Of a block of cells, select a block of cells.
  - Of a row or column, select the row header or column header.
2. From the Cell menu, choose Cell Colors or press the Cell Colors button () in the tool bar.
3. On the Colors tab in the Cell Settings dialog box,
  - a. Do either of the following:
    - Specify the background color by selecting the **BackColor** property from the Property drop-down list box.
    - Specify the text color by selecting the **ForeColor** property from the Property drop-down list box.
  - b. Choose the Color button to display the Color dialog box.
  - c. In the Color dialog box,
    - i. Select a basic color or customize your own color.
    - ii. Choose the OK button.
  - d. Choose the OK or Apply button.

ActiveX, VBX

At run time,

1. Set the **Col**, **Col2**, **Row**, and **Row2** properties as appropriate.

For more information about setting the **Col** and **Row** properties, see “Relationship of Col and Row Properties or Parameters” on page 31.
2. If you are selecting a block of cells, set the **BlockMode** property to True.
3. Specify the background color with the **BackColor** property.
4. Specify the text color with the **ForeColor** property.
5. If you selected a block of cells, set the **BlockMode** property back to False.

DLL

Call the **SSSetColor** or **SSSetColorRange** function and set the *Col*, *Row*, *Col2*, *Row2*, and *Background* or *Foreground* parameters.

## Setting the Font

You can set the font characteristics of a cell, a column, a row, a block of cells, or the entire spreadsheet.

---

**Note** If you are using the fpSpread ActiveX control or the 32-bit DLL control with a language that uses the double-byte character set (DBCS), you must change the **Font** or **FontName** property or call the **SSSetFont** or **SSSetFontRange** function and change the font to a font that supports DBCS.

---

### ► To set the font

Spread  
Designer

In the Spread Designer,

1. Select the cell, column, row, block of cells, or the entire spreadsheet.
2. From the Cell menu, choose Font.
3. In the Font dialog box, specify the font name, style, size, and effects.
4. Choose the OK or Apply button.

ActiveX, VBX

At run time,

1. Set the **Col**, **Col2**, **Row**, and **Row2** properties as appropriate.  
For more information about setting the **Col** and **Row** properties, see “Relationship of Col and Row Properties or Parameters” on page 31.
2. If you are selecting a block of cells, set the **BlockMode** property to True.
3. Set the appropriate font properties (**Font** or **FontName**, **FontSize**, **FontBold**, **FontItalic**, **FontStrikethru**, and **FontUnderline**).
4. If you selected a block of cells, set the **BlockMode** property back to False.

DLL Call the **SSSetFont** or **SSSetFontRange** function.

## Specifying the Background and Text Color of Selected Cells

By default, the spreadsheet inverts the background and text color in selected cells. You can specify the background color of the selected cells. You can also specify the color of the text in the selected cells.

► **To specify the background and text color of selected cells**

- Spread Designer
- In the Spread Designer,
1. From the General menu, choose Spreadsheet Colors or press the Spreadsheet Colors button () in the tool bar.
  2. On the Colors tab in the Default dialog box,
    - a. Specify the background color of the selected cells by selecting the **SelBackColor** property from the Property drop-down list box.
    - b. Choose the Color button to display the Color dialog box.
    - c. In the Color dialog box,
      - i. Select a basic color or customize your own color.
      - ii. Choose the OK button.
    - d. Choose the Apply button.
    - e. Specify the foreground color of the selected cells by selecting the **SelForeColor** property from the Property drop-down list box.
    - f. Choose the Color button to display the Color dialog box.
    - g. In the Color dialog box,
      - i. Select a basic color or customize your own color.
      - ii. Choose the OK button.
    - h. Choose the OK or Apply button.
- ActiveX/VBX
1. Specify the background color of the selected cells with the **SelBackColor** property.
  2. Specify the text color of the selected cells with the **SelForeColor** property.
- DLL
- Call the **SSSetSelColor** function.

# Chapter 11 Setting the Cell Type

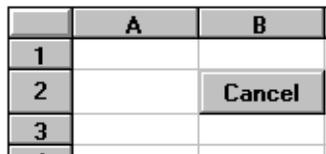
Cell types define the type of information that appears in a cell, and how that information is displayed. Examples of cell types include button, combo box, date, and integer. You can specify the cell type for cells, columns, rows, blocks of cells, or the entire spreadsheet.

This chapter describes how to set the cell type, as described in the following topics.

Topic	On Page
Creating Button Cells	209
Creating Check Box Cells	219
Creating Combo Box Cells	225
Creating Date Cells	233
Creating Edit Cells	238
Creating Float Cells	242
Creating Integer Cells	249
Creating Owner-Drawn Cells	252
Creating PIC Cells	253
Creating Picture Cells	255
Creating Static Text Cells	258
Creating Time Cells	263

## Creating Button Cells

Button cells look and act like rectangular buttons in your spreadsheet, as shown in the following figure.

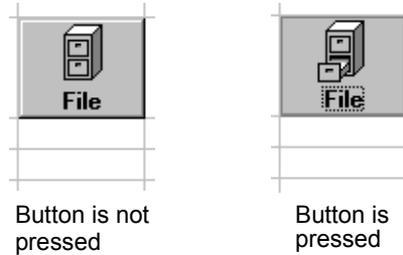


	A	B
1		
2		Cancel
3		

By default, button cells act like push buttons, which you can press by pressing your left mouse button, and which do not stay pressed when you release your mouse button. If you prefer, button cells can act like toggle, or two-state, buttons, which stay pressed when you

click them using your left mouse button. Buttons are false when they are not pressed, and true when they are pressed.

Button cells can display text, pictures, or both. If they display pictures, you can choose that a different picture is displayed when the button is pressed. You can specify the alignment of text and pictures in button cells.



You can customize the colors in button cells, including the color of the border, text, and background. In addition, button cells can display a three-dimensional appearance, and you can customize the colors of the highlight and shadow in the appearance.

	A	B
1		
2		Cancel
3		

The following sections describe how to create a button cell, set its type, and customize it.

## Creating a Button Cell and Setting Its Type

### ► To create a button cell and set its type

Spread Designer

In the Spread Designer,

1. Select a cell or block of cells.
2. Choose the cell type using one of the following methods:
  - Display the pop-up menu by right-clicking the cell or block of cells and then choose Button.
  - From the Cell menu, choose Cell Type, Button.
  - Choose the button cell button (  ) on the cell type toolbar.

The Cell Type dialog box appears.

3. In the Cell Type dialog box,
  - a. Select the type of button from the Style drop-down list box.
  - b. Choose the OK or Apply button.

ActiveX, VBX

At run time,

1. Set the **Col**, **Col2**, **Row**, and **Row2** properties as appropriate.  
For more information about setting the **Col** and **Row** properties, see “Relationship of Col and Row Properties or Parameters” on page 31.
2. If you are selecting a block of cells, set the **BlockMode** property to True.
3. Set the **CellType** property to 7 (Button).
4. Set the **TypeButtonType** property to specify the button type.
5. If you selected a block of cells, set the **BlockMode** property back to False.

DLL

1. Call the **SSSetTypeButton** function and specify the type of button using the *ButtonType* parameter.
2. Call the **SSSetCellType** or **SSSetCellTypeRange** function and set the *CellType* parameter to `SS_TYPE_BUTTON`.

## Displaying Text in a Button Cell

You can provide text to display in the button cell, and you can specify the alignment for the text.

### ► To display text in a button cell

Spread  
Designer

In the Spread Designer,

1. Select a cell or block of cells.
2. Choose the cell type using one of the following methods:
  - Display the pop-up menu by right-clicking the cell or block of cells and then choose Button.
  - From the Cell menu, choose Cell Type, Button.
  - Choose the button cell button () on the cell type toolbar.

The Cell Type dialog box appears.

3. In the Cell Type dialog box,
  - a. Type the button text in the Text box.
  - b. Select the text alignment from the Text Pos drop-down list box.

If you are also displaying a picture in the button cell, the text alignment setting moves the text to whichever side of the picture you specify. For example, if you set the position of the text to be on the left, the button cell displays the text on the left and the picture on the right.
  - c. Choose the OK or Apply button.

ActiveX, VBX      At run time,

1. Set the **Col**, **Col2**, **Row**, and **Row2** properties as appropriate.

For more information about setting the **Col** and **Row** properties, see “Relationship of Col and Row Properties or Parameters” on page 31.
2. If you are selecting a block of cells, set the **BlockMode** property to True.
3. Set the **CellType** property to 7 (Button).
4. Set the **TypeButtonText** property to specify the text to display in the button cell.
5. Set the **TypeButtonAlign** property to specify how to align the text in the button cell.

If you are also displaying a picture in the button cell, the text alignment setting moves the text to whichever side of the picture you specify. For example, if you set the position of the text to be on the left, the button cell displays the text on the left and the picture on the right.
6. If you selected a block of cells, set the **BlockMode** property back to False.

- DLL
1. Call the **SSSetTypeButton** function and specify the text for the button using the *Text* parameter and the position of the text using the *Style* parameter.

If you are also displaying a picture in the button cell, the text alignment setting moves the text to whichever side of the picture you specify. For example, if you set the position of the text to be on the left, the button cell displays the text on the left and the picture on the right.
  2. Call the **SSSetCellType** or **SSSetCellTypeRange** function and set the *CellType* parameter to `SS_TYPE_BUTTON`.

## Displaying Pictures in a Button Cell

Your button cell can display a picture. If you prefer, it can also display a different picture when it is pressed.

### ► To display a picture in a button cell

Spread  
Designer

In the Spread Designer,

1. Select a cell or block of cells.
2. Choose the cell type using one of the following methods:
  - Display the pop-up menu by right-clicking the cell or block of cells and then choose Button.
  - From the Cell menu, choose Cell Type, Button.
  - Choose the button cell button () on the cell type toolbar.

The Cell Type dialog box appears.

3. In the Cell Type dialog box under Picture,
  - a. Choose whether you are setting the picture for the normal (“up” or False state) position of the button or the picture for the pressed (“down” or True state) position from the drop-down list box.
  - b. Choose the Set button to display an Open dialog box.
  - c. In the Open dialog box, designate the path and file name of the picture you want to display in the button, then choose the Open button.
  - d. Specify the alignment for the picture in relation to button text by selecting an item from the Text Pos drop-down list box.

The text alignment setting moves the text to whichever side of the picture you specify. For example, if you set the position of the text to be on the left, the button cell displays the text on the left and the picture on the right. If the button cell does not display text, the Text Pos selection has no effect, and the control centers the picture in the cell.

- e. Choose the OK or Apply button.

ActiveX, VBX

At run time,

1. Set the **Col**, **Col2**, **Row**, and **Row2** properties as appropriate.

For more information about setting the **Col** and **Row** properties, see “Relationship of Col and Row Properties or Parameters” on page 31.

2. If you are selecting a block of cells, set the **BlockMode** property to True.
3. Set the **CellType** property to 7 (Button).
4. Set the **TypeButtonPicture** property to specify the picture to display when the button is not pressed (“up” or False).
5. Set the **TypeButtonPictureDown** property to specify the picture to display when the button is pressed (“down” or True).
6. Set the **TypeButtonAlign** property to specify the alignment for the picture in relation to button text.

The text alignment setting moves the text to whichever side of the picture you specify. For example, if you set the position of the text to be on the left, the button cell displays the text on the left and the picture on the right. If the button cell does not display text, the setting of the **TypeButtonAlign** property has no effect, and the control centers the picture in the cell.

7. If you selected a block of cells, set the **BlockMode** property back to False.

- DLL
1. Call the **SSSetTypeButton** function and use the following parameters to specify the picture to display:
    - a. Set the *Picture* parameter to provide the name of the picture to display when the button is not pressed (“up” or False).
    - b. Set the *nPictureType* parameter to tell the control the type of picture to display.
    - c. Set the *PictureDown* parameter to provide the name of the picture to display when the button is pressed (“down” or True).
    - d. Set the *nPictureDownType* parameter to tell the control the type of picture to display.
    - e. Set the *Style* parameter to specify the alignment for the picture in relation to button text.

The text alignment setting moves the text to whichever side of the picture you specify. For example, if you set the position of the text to be on the left, the button cell displays the text on the left and the picture on the right. If the button cell does not display text, the *Style* parameter alignment setting has no effect, and the control centers the picture in the cell.

2. Call the **SSSetCellType** or **SSSetCellTypeRange** function and set the *CellType* parameter to `SS_TYPE_BUTTON`.

## Setting the Border Color for Button Cells

You can change the colors of a button cell border and the dark and light highlight areas that create a three-dimensional appearance around the button cell.

By default, the border for button cells is black. If you prefer another color, you can change the color of the border.

---

**Note** The button border appearance varies according to your operating system. In a pre-Windows 95® operating system, the button has a simple three-dimensional shading and a border. In Windows 95, Windows NT® 4.0 or later, or Windows 98, the button has a complex three-dimensional shading and no border. Therefore, if your operating system is Windows 95 or Windows NT 4.0 or later, the button border color is ignored.

---

### ► To set the border color for button cells

Spread  
Designer

In the Spread Designer,

1. Select the button cell.
2. Display the Cell Type dialog using one of the following methods,
  - Display the pop-up menu by right-clicking the cell and then choose Button.
  - From the Cell menu, choose Cell Type and choose button.
  - Choose the button cell button on the cell type toolbar.

The Cell Type dialog box appears.

3. In the Cell Type dialog box, on the Settings tab under Color,
  - a. Choose BorderColor from the drop-down list box
  - b. Choose the Set button.
  - c. In the Color dialog,
    - i. Select a basic color or customize your color.
    - ii. Choose the OK button.
  - d. Choose the OK or Apply button.

## Chapter 11

- ActiveX, VBX     At run time,
1. Set the **Col**, **Col2**, **Row**, and **Row2** properties as appropriate.  
For more information about setting the **Col** and **Row** properties, see “Relationship of Col and Row Properties or Parameters” on page 31.
  2. If you are selecting a block of cells, set the **BlockMode** property to True.
  3. Set the **CellType** property to 7 (Button).
  4. If you want to specify the border color, set the **TypeButtonBorderColor** property.
  5. If you selected a block of cells, set the **BlockMode** property back to False.
- DLL
1. Use the **SUPERBTNCOLOR** structure to set the border color.
  2. Call the **SSSetTypeButton** function.
  3. Call the **SSSetCellType** or **SSSetCellTypeRange** function and set the *CellType* parameter to **SS\_TYPE\_BUTTON**.

### Customizing the Three-Dimensional Appearance of Button Cells

By default, button cells display a three-dimensional appearance created by highlight and shadow colors displayed around the outside of the cell. You can customize the colors used as well as the width of the highlight and shadow. The default three-dimensional colors depend on your Windows system button highlight and shadow colors.

When you press the button, the three-dimensional appearance changes to show the button is pressed. If you are using the DLL control, you can specify that the three-dimensional appearance not change when the button is pressed.

#### ► To customize the three-dimensional appearance of button cells

Spread  
Designer

In the Spread Designer,

1. Select the button cell.
2. Display the Cell Type dialog using one of the following methods,
  - Display the pop-up menu by right-clicking the cell and then choose Button.
  - From the Cell menu, choose Cell Type and choose button.
  - Choose the button button on the cell type toolbar.

The Cell Type dialog box appears.

3. To specify the border color, on the Settings tab under Color in the Cell Type dialog box,
  - a. Choose BorderColor from the drop-down list box
  - b. Choose the Set button.
  - c. Select a basic color or customize your color.
  - d. Choose the OK button.
4. To specify the shadow color, on the Settings tab under Color in the Cell Type dialog box,
  - a. Choose DarkColor from the drop-down list box
  - b. Choose the Set button.
  - c. Select a basic color or customize your color.
  - d. Choose the OK button.
5. To specify the highlight color, on the Settings tab under Color in the Cell Type dialog box,
  - a. Choose LightColor from the drop-down list box
  - b. Choose the Set button.
  - c. Select a basic color or customize your color.
  - d. Choose the OK button.
6. Choose the OK or Apply button.

ActiveX, VBX

At run time,

1. Set the **Col**, **Col2**, **Row**, and **Row2** properties as appropriate.  
For more information about setting the **Col** and **Row** properties, see “Relationship of Col and Row Properties or Parameters” on page 31.
2. If you are selecting a block of cells, set the **BlockMode** property to True.
3. Set the **CellType** property to 7 (Button).
4. If you want to specify the border color, set the **TypeButtonBorderColor** property.
5. If you want to specify the shadow color, set the **TypeButtonDarkColor** property.
6. If you want to specify the highlight color, set the **TypeButtonLightColor** property.
7. If you want to change the width of the highlight and shadow area, set the **TypeButtonShadowSize** property.

- DLL
1. Use the **SUPERBTNCOLOR** structure to set the border, shadow, and highlight colors.
  2. Call the **SSSetTypeButton** function (and set the *Style* and *nShadowSize* parameters).  
If you do not want the three-dimensional appearance to change when the button is pressed, set the *Style* parameter to `SBS_NOINVERTFRAME`.
  3. Call the **SSSetCellType** or **SSSetCellTypeRange** function and set the *CellType* parameter to `SS_TYPE_BUTTON`.

## Setting Background and Text Colors of Button Cells

By default, the background color of a button cell is the Windows system button face color and the text color is black. You can customize the background and text colors.

### ► To set the background and text colors of button cells

Spread  
Designer

In the Spread Designer,

1. Select the button cell.
2. Display the Cell Type dialog using one of the following methods,
  - Display the pop-up menu by right-clicking the cell and then choose Button.
  - From the Cell menu, choose Cell Type and choose button.
  - Choose the button button on the cell type toolbar.The Cell Type dialog box appears.
3. To specify the background color, on the Settings tab under Color in the Cell Type dialog box,
  - a. Choose ButtonColor from the drop-down list box
  - b. Choose the Set button.
  - c. Select a basic color or customize your color.
  - d. Choose the OK button.

4. To specify the text color, on the Settings tab under Color in the Cell Type dialog box,
  - a. Choose TextColor from the drop-down list box
  - b. Choose the Set button.
  - c. Select a basic color or customize your color.
  - d. Choose the OK button.
5. Choose the OK or Apply button.

ActiveX, VBX

At run time,

1. Set the **Col**, **Col2**, **Row**, and **Row2** properties as appropriate.

For more information about setting the **Col** and **Row** properties, see “Relationship of Col and Row Properties or Parameters” on page 31.

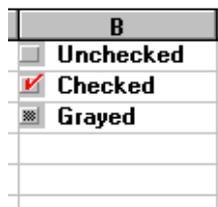
2. If you are selecting a block of cells, set the **BlockMode** property to True.
3. Set the **CellType** property to 7 (Button).
4. Specify the background color with the **TypeButtonColor** property.
5. Specify the text color with the **TypeButtonText** property.

DLL

1. Use the **SUPERBTNCOLOR** structure to specify the background and text colors.
2. Call the **SSSetTypeButton** function.
3. Call the **SSSetCellType** or **SSSetCellTypeRange** function and set the *CellType* parameter to **SS\_TYPE\_BUTTON**.

## Creating Check Box Cells

Check box cells provide a check box that can show two or three states, as shown in the following figure.



Check box cells can display text and a check box picture, which you can customize.

## Creating a Check Box Cell and Setting Its Type

By default, check box cells provide two-state check boxes, which can be either True or False. You can provide a three-state check box instead, which can be either True, False, or Grayed. The grayed state is an indeterminate state. An example of when the check box might be grayed is when it represents whether text is bold, and in the current selection, some text is bold and some is not.

You can horizontally and vertically align the check box and text within the cell.

### ► To create a check box cell and set its type

Spread  
Designer

In the Spread Designer,

1. Select a cell or block of cells.
2. Choose the cell type using one of the following methods:
  - Display the pop-up menu by right-clicking the cell or block of cells and then choose **Checkbox**.
  - From the **Cell** menu, choose **Cell Type, Checkbox**.
  - Choose the check box button () on the cell type toolbar.

The Cell Type dialog box appears.

3. In the Cell Type dialog box, on the **Settings** tab,
  - a. If you want the text to be aligned to the left of the check box, select the **Text Left of Picture** check box.
  - b. If you want to align the check box and text in the cell, select the appropriate option buttons under **Horizontal Alignment** and **Vertical Alignment**.
  - c. If you want to provide a three-state check box, under **Type**, select the **Three State** option button.
  - d. Choose the **OK** or **Apply** button.

ActiveX, VBX

At run time,

1. Set the **Col**, **Col2**, **Row**, and **Row2** properties as appropriate.
2. If you are selecting a block of cells, set the **BlockMode** property to **True**.
3. Set the **CellType** property to 10 (**Check Box**).

4. Specify the horizontal and vertical alignment of the check box and text with the **TypeHAlign** and **TypeVAlign** properties.  
If you want to center the check box and text horizontally and vertically in the cell, perform one of the following actions:
  - Set the **TypeCheckCenter** property to True.
  - Set the **TypeHAlign** and **TypeVAlign** properties to 2 (Center) and 2 (VCenter), respectively.
5. If you want to provide a three-state check box, set the **TypeCheckType** property to 1 (Three-state).
6. If you selected a block of cells, set the **BlockMode** property back to False.

- DLL
1. Call the **SSSetTypeCheckBox** function and set the parameters as follows:
    - a. If you want to center the check box in the cell, set the *lStyle* parameter to BS\_CENTER.
    - b. If you want to provide a three-state check box, set the *lStyle* parameter to BS\_AUTO3STATE.
    - c. Use the *lStyle* parameter to specify the text alignment.
  2. Call the **SSSetCellType** or **SSSetCellTypeRange** function and set the *CellType* parameter to SS\_TYPE\_CHECKBOX.

## Specifying the Text for a Check Box Cell

You can provide text for the check box, labeling what the value of the check box represents. You can display the text to either side of the displayed picture.

### ► To specify the text for a check box cell

Spread  
Designer

In the Spread Designer,

1. Select a cell or block of cells.
2. Choose the cell type using one of the following methods:
  - Display the pop-up menu by right-clicking the cell or block of cells and then choose Checkbox.
  - From the Cell menu, choose Cell Type, Checkbox.
  - Choose the check box button () on the cell type toolbar.

The Cell Type dialog box appears.

3. In the Cell Type dialog box, on the Settings tab,
  - a. Type the check box text in the Text box.
  - b. To display the text to the left of the check box or pictures, select the Text Left of Picture check box.
  - c. Choose the OK or Apply button.

ActiveX, VBX

At run time,

1. Set the **Col**, **Col2**, **Row**, and **Row2** properties as appropriate.
2. If you are selecting a block of cells, set the **BlockMode** property to True.
3. Set the **CellType** property to 10 (Check Box).
4. Set the **TypeCheckText** property to specify the text to display in the cell.
5. If you want to display the text to the left of the check box or the picture in the cell, set the **TypeCheckTextAlign** property to 0 (Left).
6. If you selected a block of cells, set the **BlockMode** property back to False.

DLL

1. Call the **SSSetTypeCheckBox** function and set the parameters as follows:
  - a. Set the *lpText* parameter to specify the text to display in the cell.
  - b. If you want to display the text to the left of the check box or picture, set the *lStyle* parameter to BS\_LEFTTEXT.
2. Call the **SSSetCellType** or **SSSetCellTypeRange** function and set the *CellType* parameter to SS\_TYPE\_CHECKBOX.

## Displaying Pictures in a Check Box Cell

The fpSpread control provides six default pictures for check box cells, as shown in the following table.

Picture	Check box state it represents
	False (Up position)
	False pressed (Up position with the pointer over it when user is pressing mouse button)
	True (Down position)
	True pressed (Down position with the pointer over it when user is pressing mouse button)

**Picture**      **Check box state it represents**

Grayed (Check box reflects data that has an indeterminate state)



Grayed pressed (Check box reflects data that has an indeterminate state, user is pressing mouse button while pointer is over the button)

If you prefer, you can use your own custom pictures for any or all of the pictures that represent the check box states.

---

**Tip**    When using custom pictures to represent states, try to create a consistent look and feel for the check box cell. Also, be aware of common interface conventions used to represent states, such as using lighter shades of text or graphics to represent the grayed state.

---

► **To display pictures in a check box cell**

Spread  
Designer

In the Spread Designer,

1. Select a cell or block of cells.
2. Choose the cell type using one of the following methods:
  - Display the pop-up menu by right-clicking the cell or block of cells and then choose **Checkbox**.
  - From the **Cell** menu, choose **Cell Type, Checkbox**.
  - Choose the check box button () on the cell type toolbar.

The Cell Type dialog box appears.
3. In the Cell Type dialog box, on the **Settings** tab,
  - a. Under **Picture**, to specify pictures for different states of the check box,
    - i. Select one of the following from the drop-down list box:
      - **Unchecked**—displays when check box is cleared (False state)
      - **Unchecked Down**—displays when check box is cleared (False state) and pressed
      - **Checked**—displays when check box is selected (True state)
      - **Checked Down**—displays when check box is selected (True state) and pressed
      - **Grayed**—displays when check box is indeterminate
      - **Grayed Down**—displays when check box is indeterminate and pressed

## Chapter 11

- ii. Choose the Set button to display the Open dialog box.
- iii. Select an available graphics file.
- iv. Choose the Open button.
- b. If you want the text to be displayed to the left of the picture, select the Text Left of Picture check box.
- c. Choose the OK or Apply button.

ActiveX, VBX

At run time,

1. Set the **Col**, **Col2**, **Row**, and **Row2** properties as appropriate.
2. If you are selecting a block of cells, set the **BlockMode** property to True.
3. Set the **CellType** property to 10 (Check Box).
4. Set the **TypeCheckPicture** property using the following index values to specify which picture you are setting for each state:

<b>Index Value</b>	<b>Sets picture displayed for this state</b>
0 - Unchecked	False
1 - Checked	True
2 - Unchecked pressed	False pressed
3 - Checked pressed	True pressed
4 - Grayed	Grayed
5 - Grayed pressed	Grayed pressed

5. Specify where the text is displayed in relation to the picture by setting the **TypeCheckTextAlign** property.
6. If you selected a block of cells, set the **BlockMode** property back to False.

DLL

1. Call the **SSSetTypeCheckBox** function and set the parameters as described in the following table to specify the pictures for the check box states.

<b>If you want to specify the picture for this state . . .</b>	<b>Set these parameters . . .</b>
False	<i>lpPictUp</i> <i>wPictUpType</i>
False pressed	<i>lpPictFocusUp</i> <i>wPictFocusUpType</i>
True	<i>lpPictDown</i> <i>wPictDownType</i>

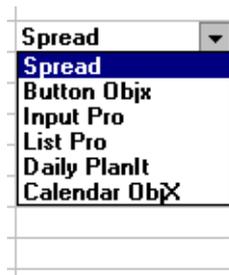
**If you want to specify the picture for this state . . .    Set these parameters . . .**

True pressed	<i>lpPictFocusDown</i> <i>wPictFocusDownType</i>
Grayed	<i>lpPictGray</i> <i>wPictGray</i>
Grayed pressed	<i>lpPictFocusGray</i> <i>wPictFocusGrayType</i>

2. Call the **SSSetCellType** or **SSSetCellTypeRange** function and set the *CellType* parameter to **SS\_TYPE\_CHECKBOX**.

## Creating Combo Box Cells

You can provide combo box cells to let the user choose from a list of available items, as shown in the following figure.



## Creating and Setting up a Combo Box Cell

---

**Note** You can create a more powerful combo box cell by placing a List Pro fpCombo control on the fpSpread control. For more information, see “Creating a Combo Box Cell Using the FarPoint fpCombo Control” on page 230.

---

Combo box cells can function as drop-down combo boxes or drop-down list boxes. By default, they function as drop-down list boxes, in which the user cannot edit the text in the cell. If you prefer, you can change the cell to act as a drop-down combo box, in which users can edit the text in the cell. The cell then acts as the edit field for the combo box.

By default, the drop-down list displays three rows. You can change the number of rows that display in the drop down list. Also by default, the text in the edit field is limited to 60 characters and is left aligned. You can change the maximum number of characters allowed, and you can horizontally and vertically align the text in the edit field.

### ► To create and set up a combo box cell

Spread  
Designer

In the Spread Designer,

1. Select a cell or block of cells.
2. Choose the cell type using one of the following methods:
  - Display the pop-up menu by right-clicking the cell or block of cells and then choose Combobox.
  - From the Cell menu, choose Cell Type, Combobox.
  - Choose the combo box button () on the cell type toolbar.

The Cell Type dialog box appears.

3. In the Cell Type dialog box, on the Settings tab,
  - a. If you want the combo box to act as a drop-down combo box, select the Editable check box.
  - b. If you want to horizontally or vertically align the text in the edit field, select the appropriate option button under Horizontal Alignment and Vertical Alignment.
  - c. Provide list items for the combo box by completing the following steps:
    - i. Type the combo box list items in the Text box.
    - ii. Choose the Add button under Combo Box List.
  - d. Delete a combo box list item by completing the following steps:
    - i. Click the item or choose the Prev or Next buttons to highlight the item under Combo Box List.
    - ii. Choose the Delete button.
  - e. Edit a combo box list item by completing the following steps:
    - i. Under Combo Box List, click the item to select it.
    - ii. Type the changes in the Text box.
    - iii. Choose the Update button under Combo Box List.
  - f. Choose the OK or Apply button.

ActiveX, VBX

At run time,

1. Set the **Col**, **Col2**, **Row**, and **Row2** properties as appropriate.
2. If you are selecting a block of cells, set the **BlockMode** property to True.

3. Set the **CellType** property to 8 (Combo Box).
  4. If you want the combo box to act as a drop-down combo box, set the **TypeComboBoxEditable** property to True.
  5. Specify the horizontal and vertical alignment of the text in the edit field with the **TypeHAlign** and **TypeVAlign** properties.
  6. Provide list items for the combo box by providing a string of tab-delimited items using the **TypeComboBoxList** property.
  7. Specify the maximum number of characters allowed in the cell using the **TypeMaxEditLen** property.
  8. Specify the number of rows in the drop-down list with the **TypeComboBoxMaxDrop** property.
  9. If you selected a block of cells, set the **BlockMode** property back to False.
- DLL
1. If you want to use the default combo box, call the **SSSetTypeComboBox** function and set the parameters as follows:
    - a. If you want the combo box to act as a drop-down combo box, set the *lStyle* parameter to SS\_CB\_DROPDOWN.
    - b. Provide list items for the combo box by providing a string of tab-delimited items using the *lpItems* parameter.
    - c. Use the *lStyle* parameter to specify the text alignment.
  2. If you want to customize the combo box, call the **SSSetTypeComboBoxEx** function and set the parameters as follows:
    - a. If you want the combo box to act as a drop-down combo box, set the *lStyle* parameter to SS\_CB\_DROPDOWN.
    - b. Provide list items for the combo box by providing a string of tab-delimited items using the *lpItems* parameter.
    - c. If you want to change the number of rows in the drop-down list, set the *dMaxRows* parameter.
    - d. If you want to specify the maximum number of characters that are displayed in the edit field, set the *dMaxEditLen* parameter.
    - e. Use the *lStyle* parameter to specify the text alignment.

- f. If you want to specify the width of the combo box, set the *dComboWidth* parameter.

For more information about how to set this parameter, see “Specifying the Width and Alignment of the Combo Box Drop-Down List” on page 228.

- 3. Call the **SSSetCellType** or **SSSetCellTypeRange** function and set the *CellType* parameter to `SS_TYPE_COMBOBOX`.

## Specifying the Width and Alignment of the Combo Box Drop-Down List

By default, the width of the combo box drop-down list is the current width of the cell as shown in the following figure.

	Owner	Breed	Date	D
1				
2		Rottweil		
3		Bichon		
4		Cavalier		
5		Labrador		
6		Airedale		
7				

You can

- Set the width to display the longest line of text in the drop-down list. The drop-down list box will be left aligned or right-aligned to the selection box depending on the cell width and the length of the longest line of text in the drop-down list. If the cell width is less than the longest line of text, the drop-down list is left-justified. If the cell width is greater than the longest of text, the drop-down list is right-justified.

	Owner	Breed	Date	D
1				
2		Rottweiler		
3		Bichon Friese		
4		Cavalier King Charles Spaniel		
5		Labrador Retriever		
6		Airedale Terrier		
7				

- Set the width to display the longest line of text in the drop-down list and right align the drop-down list box to the selection box

	Owner	Breed	Date	D
1				
6				
7				

- Specify the width as a negative number. The drop-down list box will be right aligned to the selection box.

	Owner	Breed	Date	D
1				
6				
7				

- Specify the width as a positive number. The drop-down list box will be left aligned or right-aligned to the selection box depending on the cell width and the length of the longest line of text in the drop-down list. If the cell width is less than the longest line of text, the drop-down list is left-justified. If the cell width is greater than the longest of text, the drop-down list is right-justified.

	Owner	Breed	Date	D
1				
2				
3				
4				
5				
6				
7				

► **To specify the width and alignment of the combo box drop-down list box**

Spread  
Designer

In the Spread Designer,

1. Select a cell or block of cells.
2. Choose the cell type using one of the following methods:
  - Display the pop-up menu by right-clicking the cell or block of cells and then choose Combobox.
  - From the Cell menu, choose Cell Type, Combobox.
  - Choose the combo box button () on the cell type toolbar.

The Cell Type dialog box appears.

3. In the Cell Type dialog box, on the Settings tab,
  - a. If you want the width of the drop-down list to be equal to the current width of the cell, choose Cell Width from the List Width drop-down combo box.
  - b. If you want the width of the drop-down list to display the longest line of text and the text to be right-justified, choose Right Justify from the List Width drop-down combo box.
  - c. If you want to set a specific width, type a value in the edit field of the List Width drop-down combo box.
 

If you enter a negative number, the drop-down list will be right-justified. If you enter a positive number, the drop-down list box will be left aligned or right-aligned to the selection box depending on the cell width and the length of the longest line of text in the drop-down list. If the cell width is less than the longest line of text, the drop-down list is left-justified. If the cell width is greater than the longest of text, the drop-down list is right-justified.
  - d. Choose the OK or Apply button.

ActiveX, VBX At run time, set the **TypeComboBoxWidth** property.

- DLL
1. Call the **SSSetTypeComboBoxEx** function and set the *dComboWidth* parameter.
  2. Call the **SSSetCellType** or **SSSetCellTypeRange** function and set the *CellType* parameter to `SS_TYPE_COMBOBOX`.

## Creating a Combo Box Cell Using the FarPoint fpCombo Control

You can enhance a Spread combo box cell by using the drop-down list portion of the fpCombo control, FarPoint Technologies' customizable, powerful combo box control provided in the List Pro™ product.

In the following illustration, column 2, row 1 of the fpSpread control is a combo box cell that contains the List Pro combo box control. The list of the fpCombo control is bound to the BIBLIO.MDB database and the Author record source.

	A	B	C
1			
2		ISBN	Au_ID
3		0-0131985-2-1	13
4		0-0238669-4-2	113
5		0-0280042-4-8	11
6		0-0280042-4-8	120
		0-0280095-2-5	171
		0-0702063-1-7	26

---

**Tip** The combo box data you enter using the Spread **TypeComboBoxList** or **TypeComboBoxString** properties will supersede the data contained in the fpCombo control.

---

The following fpCombo control properties are ignored when the fpCombo control is used in a combo box cell in a spreadsheet.

ColumnBound	EditHeight	ListWidth	SelLength
ComboGap	GrayAreaColor	MaxDrop	SelStart
DataField	ListDown	MaxEditLen	SelText
DataSource	ListLeftOffset	MouseOverArea	Style

The following fpSpread control properties will not return the correct values when the fpCombo control is used in a combo box cell in a spreadsheet:

TypeComboBoxCount	TypeComboBoxList
TypeComboBoxEditable	TypeComboBoxString

Some of the samples provided with the Spread control illustrate how to use the fpSpread control and the fpCombo control together. When you install Spread using the default settings, the samples are installed in the \SPREAD30\SAMPLES directory.

► **To create a combo box cell using the fpCombo control**

ActiveX, VBX

1. Set the **Col**, **Row**, **Col2**, and **Row2** properties to specify the cell or block of cells.
2. If you are selecting a block of cells, set the **BlockMode** property to True.
3. Set the **CellType** property to 8 (Combo Box).
4. If you selected a block of cells, set the **BlockMode** property back to False.

## Chapter 11

5. At run time, set the **TypeComboBoxWnd** property to the window handle of the fpCombo control.
6. For the fpCombo control, either hide the control or set the **Visible** property to False.

- DLL
1. Call the **SSSetTypeComboBox** function.
  2. Call the **SSSetCellType** or **SSSetCellTypeRange** function and set the *CellType* parameter to `SS_TYPE_COMBOBOX`.
  3. Call the **SSSetTypeComboBoxEx** function and set the *hWndDropDown* parameter to the window handle of the `fpCombo` control.
  4. Hide the `fpCombo` control.

## Creating Date Cells

Date cells display date values. You can specify one of several date formats, as shown in the following figure.

12/28/98
12/NOV/98
28/11/98
98/11/28

You can set maximum and minimum values for the date values, and you can customize the available date formats.

## Creating and Setting up a Date Cell

If you prefer, you can limit the date values allowed in the date cell. By default, the maximum date value allowed is 12/31/2100 and the minimum date value allowed is 01/01/1990. However, you can set the maximum and minimum date values to any dates between 01/01/1000 and 12/31/9999.

With dates containing two-digit century values, you can determine how the century is calculated. You set a four-digit calculation date and Spread calculates the century by using years between the calculation date and the calculation date minus 99 years. For example, if you specify a calculation date of 2099, the dates range between the year 2000 and 2099. In this case, the century for all date calculations is 20. As another example, if you specify a calculation date of 2029, the dates range between 1930 and 2029. The century is either 19 or 20, depending on the last two year digits. In this case if the last two year digits are 30–99, the century is 19; if the last two year digits are 00–29, the century is 20.

By default, the text is left-aligned and displays at the top of the cell. You can change the horizontal and vertical alignment of text in a date cell.

### ► To create and set up a date cell

Spread  
Designer

In the Spread Designer,

1. Select a cell or block of cells.
2. Choose the cell type using one of the following methods:
  - Display the pop-up menu by right-clicking the cell or block of cells and then choose Date.
  - From the Cell menu, choose Cell Type, Date.
  - Choose the date button () on the cell type toolbar.

The Cell Type dialog box appears.

3. To specify how the century is calculated,
  - a. From the General menu, choose Spreadsheet Environment or press the Spreadsheet Environment button () in the toolbar.
  - b. On the General tab in the Sheet Environment Settings dialog box, click the arrows or type a number in the Pivot Year spin box under Settings.  
  
Spread calculates the minimum year by subtracting 99 years from the calculation date.
4. In the Cell Type dialog box, on the Settings tab,
  - a. Type the maximum date for the cell in the Max Date box.
  - b. Type the minimum date for the cell in the Min Date box.
  - c. If you want to horizontally or vertically align the text, select the appropriate option button under Horizontal Alignment and Vertical Alignment.
  - d. Choose the OK or Apply button.

ActiveX, VBX

At run time,

1. Set the **Col**, **Col2**, **Row**, and **Row2** properties as appropriate.
2. If you are selecting a block of cells, set the **BlockMode** property to True.
3. Set the **CellType** property to 0 (Date).
4. Set the **TypeDateMax** and **TypeDateMin** properties to specify the maximum and minimum date values for the cell.

When providing the maximum and minimum date values, use the format “MMDDYYYY”, where “MM” stands for two-digit month, “DD” stands for two-digit day, and “YYYY” stands for four-digit year.

5. Specify how the century is calculated by setting the **TwoDigitYearMax** property (ActiveX control) or calling the **SetTwoDigitYearMax** function (VBX control) to the calculation date.  
Spread calculates the minimum year by subtracting 99 years from the calculation date.
6. Specify the horizontal and vertical alignment of the text by setting the **TypeHAlign** and **TypeVAlign** properties.
7. If you selected a block of cells, set the **BlockMode** property back to False.

- DLL
1. Call the **SSSetTypeDate** function and set the parameters as follows:
    - a. Set the *lpMax* parameter to point to the SS\_DATE structure, which contains the maximum date for the cell.
    - b. Set the *lpMin* parameter to point to the SS\_DATE structure, which contains the minimum date for the cell.
    - c. Use the *lStyle* parameter to specify the text alignment.
  2. Specify how the century is calculated by calling the **SSSetTwoDigitYearMax** function and setting the *nTwoDigitYearMax* parameter to the calculation date.
  3. Call the **SSSetCellType** or **SSSetCellTypeRange** function and set the *CellType* parameter to SS\_TYPE\_DATE.

## Formatting a Date Cell

Date cells come with predefined date format options from which you can choose, as illustrated in the following table.

<b>Date Format</b>	<b>Date Representation</b>
28/NOV/98	DDMONYY
28/11/98	DDMMYY
11/28/98	MMDDYY
98/11/28	YYMMDD

You can also further customize the dates by customizing the separator character and by specifying whether the century value is displayed.

---

**Note** If your date cell format displays only two-digit years, the control makes an assumption about the four-digit year value that corresponds to the displayed two-digit year value every time the century value is needed. If you choose not to provide four-digit year values and instead let the fpSpread control assume a century value, be aware that the control might extrapolate the century value using the method specified by the **TwoDigitYearMax** property or assume the value is Null. You must check the resulting century value to ensure it is the value you intend.

---

In addition, you can align the text within the cell. If you want, the cell can also provide a spin button to let users change dates quickly.

► **To format a date cell**

Spread  
Designer

In the Spread Designer,

1. Select a cell or block of cells.
2. Choose the cell type using one of the following methods:
  - Display the pop-up menu by right-clicking the cell or block of cells and then choose Date.
  - From the Cell menu, choose Cell Type, Date.
  - Choose the date button () on the cell type toolbar.

The Cell Type dialog box appears.
3. In the Cell Type dialog box, on the Settings tab,
  - a. Select the date format from the Format drop-down list box.
  - b. To display the year in century format, select the Display Century check box.
  - c. Type the character that separates the day, month, and year in the Separator box.
  - d. If you want to horizontally or vertically align the text, select the appropriate option button under Horizontal Alignment and Vertical Alignment.
  - e. To display a spin button, select the Display Spin Button check box.
  - f. Choose the OK or Apply button.

ActiveX, VBX

At run time,

1. Set the **Col**, **Col2**, **Row**, and **Row2** properties as appropriate.
2. If you are selecting a block of cells, set the **BlockMode** property to True.
3. Set the **CellType** property to 0 (Date).

4. Specify the date format by setting the **TypeDateFormat** property.
5. Display the year in century format (four-digits), by setting the **TypeDateCentury** property to True.
6. Specify the character that separates the day, month, and year by setting the **TypeDateSeparator** property.
7. If you want to display a spin button, set the **TypeSpin** property to True.
8. If you selected a block of cells, set the **BlockMode** property back to False.

- DLL
1. Call the **SSSetTypeDate** function and set the parameters as follows:
    - a. Set the *Style* parameter to specify the date and century format to display.
    - b. Set the *Style* parameter to specify the text alignment.
    - c. Set the *Style* parameter to DS\_SPIN to display a spin button in the cell.

You can also use the *lpFormat* parameter to point to the **SS\_DATEFORMAT** structure. Set the fields in the **SS\_DATEFORMAT** structure as follows:

To specify this formatting . . .	Set this field . . .
Displays century	<i>bCentury</i>
Separator character	<i>cSeparator</i>
Displays spin button	<i>bSpin</i>
Date format	<i>nFormat</i>

2. Call the **SSSetCellType** or **SSSetCellTypeRange** function and set the *CellType* parameter to SS\_TYPE\_DATE.

## Customizing Day and Month Names for the Pop-up Calendar

If you double click a date cell when it is in edit mode, a pop-up calendar appears. The date you choose displays in the date cell.

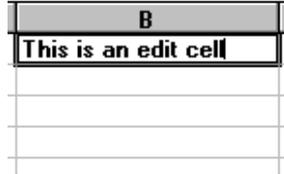
You can specify short day names, long day names, short month names, and long month names. You can also specify the text that displays for the Ok and Cancel buttons.

### ► To customize day and month names for the pop-up calendar

- ActiveX, VBX    At run time, call the **SetCalText** function or method and set the *ShortDays*, *LongDays*, *ShortMonths*, *LongMonths*, *OkText*, and *CancelText* parameters as appropriate.
- DLL    Call the **SSSetCalText** function and set the *ShortDays*, *LongDays*, *ShortMonths*, *LongMonths*, *OkText*, and *CancelText* parameters as appropriate.

## Creating Edit Cells

Edit cells accept and display text typed by the user, as shown in the following figure.



If you prefer, you can restrict entry into the edit field, such as which characters users are allowed to type. In addition, you can specify characteristics for the displayed text.

### Creating and Customizing an Edit Cell

You can specify that letters typed, pasted, or provided in code or from a database to the cell, are displayed in all upper case, all lower case, or as they were provided. The text in the cell can be horizontally and vertically aligned in the cell.

If you are using the ActiveX or VBX control, you can choose to have the edit field appear as a password field, displaying the '\*' character instead of the actual typed character, as shown in the following figure.

	A	B
1		
2	Password	*****
3		

---

#### Notes

- Providing the appearance of a password field does not provide the function of a password. If you want the edit cell to act as a password, you must add code to your project to do so.
  - Changing the setting for the character case does not change the case of existing text in the specified edit cell; the setting only affects the case of text provided after the setting is changed.
-

► **To create an edit cell**

Spread  
Designer

In the Spread Designer,

1. Select a cell or block of cells.
2. Choose the cell type using one of the following methods:
  - Display the pop-up menu by right-clicking the cell or block of cells and then choose Edit.
  - From the Cell menu, choose Cell Type, Edit.
  - Choose the edit button () on the cell type toolbar.

The Cell Type dialog box appears.

3. In the Cell Type dialog box, on the Settings tab,
  - a. Specify the case of the provided characters by selecting an option button under Text Case.
  - b. Specify text alignment by selecting an option button under Horizontal Alignment or Vertical Alignment.
  - c. Set up the cell to appear as a password entry field by selecting the Password check box.
  - d. Choose the OK or Apply button.

ActiveX, VBX

At run time,

1. Set the **Col**, **Col2**, **Row**, and **Row2** properties as appropriate.
2. If you are selecting a block of cells, set the **BlockMode** property to True.
3. Set the **CellType** property to 1 (Edit).
4. Specify the case of the characters provided to the cell by setting the **TypeEditCharCase** property.
5. Specify the text alignment using the **TypeHAlign** and **TypeVAlign** properties.
6. Set up the cell to appear as a password entry field by setting the **TypeEditPassword** property to True.
7. If you selected a block of cells, set the **BlockMode** property back to False.

- DLL
1. Call the **SSSetTypeEdit** function and set the parameters as follows:
    - a. Use the *ChrCase* parameter to specify the case of the provided characters.
    - b. Use the *Style* parameter to specify the text alignment.
  2. Call the **SSSetCellType** or **SSSetCellTypeRange** function and set the *CellType* parameter to `SS_TYPE_EDIT`.

### Limiting User Entry in an Edit Cell

You can limit the characters the cell allows the user to type by specifying a valid character set from one of the available sets, such as alphabetic characters or numbers. In addition, you can limit the number of characters the user types in the cell.

By default, the cell does not allow users to type in multiple lines of text. However, if you prefer, you can allow the user to enter and the cell to display multiple lines of text.

---

**Note** Changing the setting of the characters allowed in the cell or the number of characters allowed in the cell does not affect existing text in the cell; the setting only affects text provided after the setting is changed.

---

#### ► To limit user entry in an edit cell

Spread  
Designer

In the Spread Designer,

1. Select a cell or block of cells.
2. Choose the cell type using one of the following methods:
  - Display the pop-up menu by right-clicking the cell or block of cells and then choose Edit.
  - From the Cell menu, choose Cell Type, Edit.
  - Choose the edit button () on the cell type toolbar.The Cell Type dialog box appears.
3. In the Cell Type dialog box, on the Settings tab,
  - a. Select a character set that can be used to type data into the cell from the Character Set drop-down list box.
  - b. Click the arrows or type a value in the Length spin box to specify the maximum number of characters allowed in the cell.

- c. Specify that multiple lines of data can be typed in the cell by selecting the Multiple Lines check box.
- d. Choose the OK or Apply button.

## ActiveX, VBX

At run time,

1. Set the **Col**, **Col2**, **Row**, and **Row2** properties as appropriate.
2. If you are selecting a block of cells, set the **BlockMode** property to True.
3. Set the **CellType** property to 1 (Edit).
4. Provide a character set that specifies what the user can type into the cell by setting the **TypeEditCharSet** property.
5. Specify the maximum number of characters allowed in the cell using the **TypeMaxEditLen** property.
6. Allow users to type text in multiple lines by setting the **TypeEditMultiLine** property to True.
7. If you selected a block of cells, set the **BlockMode** property back to False.

## DLL

1. Call the **SSSetTypeEdit** function and set the parameters as follows:
  - a. Set the *ChrSet* parameter to specify a character set that can be used to type data.
  - b. Set the *Len* parameter to specify the maximum number of characters allowed in the cell.
  - c. Set the *Style* parameter to ES\_MULTILINE to let users type multiple lines of text in the cell.
  - d. If you set the *Style* parameter to ES\_MULTILINE in step 1c and you want to allow the user to press the Enter key to add a new line, set the *Style* parameter to ES\_WANTRETURN.
  - e. Set the *Style* parameter to ES\_AUTOHSCROLL to allow horizontal scrolling in the cell.
2. Call the **SSSetCellType** or **SSSetCellTypeRange** function and set the *CellType* parameter to SS\_TYPE\_EDIT.

## Creating Float Cells

Float cells display integer and decimal values and offer a variety of formatting options, as shown in the following figure.

	A	B
1	Decimal Character	1000.00
2	Currency Character	\$1000.00
3	Decimal Places	1000.1234
4	Min/Max Values	9999999.99
5	Money	£1000.00
6	Thousands Sepearator	10,000.00
7		

You can customize the float cell's formatting, as well as characteristics such as maximum and minimum values and data alignment.

## Creating and Customizing a Float Cell

You can specify the maximum and minimum values allowed in the cell. In addition, you can limit the number of digits allowed in the integer and decimal values.

You can also vertically align the data in the cell to the top or bottom, or center it within the cell.

### ► To create and customize a float cell

Spread  
Designer

In the Spread Designer,

1. Select a cell or block of cells.
2. Choose the cell type using one of the following methods:
  - Display the pop-up menu by right-clicking the cell or block of cells and then choose Float.
  - From the Cell menu, choose Cell Type, Float.
  - Choose the float button (  ) on the cell type toolbar.

The Cell Type dialog box appears.

3. In the Cell Type dialog box,
  - a. Type the maximum value for the cell in the Max Value box.
  - b. Type the minimum value for the cell in the Min Value box.

- c. Specify the number of digits that can be typed to the right of the decimal place by clicking the arrows or typing the value in the Decimal Digits spin box.

---

**Note** The total number of digits in the Decimal Digits and Integer Digits spin boxes cannot be more than 14.

---

- d. Specify the number of digits that can be typed to the left of the decimal place by clicking the arrows or typing the value in the Integer Digits spin box.

---

**Note** The total number of digits in the Decimal Digits and Integer Digits spin boxes cannot be more than 14.

---

- e. Specify how the data is horizontally and vertically aligned in the cell by selecting an option button under Horizontal Alignment and Vertical Alignment.
- f. Choose the OK or Apply button.

ActiveX, VBX      At run time,

1. Set the **Col**, **Col2**, **Row**, and **Row2** properties as appropriate.
2. If you are selecting a block of cells, set the **BlockMode** property to True.
3. Set the **CellType** property to 2 (Float).
4. Specify the maximum value for the cell using the **TypeFloatMax** property.
5. Specify the minimum value for the cell using the **TypeFloatMin** property.
6. Specify the number of decimal digits to display by setting the **TypeFloatDecimalPlaces** property.

---

**Note** The total number of digits in the decimal value cannot be more than 14.

---

7. Specify the horizontal and vertical alignment of the data in the cell by setting the **TypeHAlign** and **TypeVAlign** property.
8. If you selected a block of cells, set the **BlockMode** property back to False.

- DLL      1. If you want to set the format characters used to display currency, the thousands separator, or the decimal character, call the **SSSetTypeFloatExt** function.

If you want to use the default characters used to display currency (\$), the thousands separator (,), and the decimal character (.), call the **SSSetTypeFloat** function.

Set the parameters for the functions as follows:

- a. Set the *Max* parameter to specify the maximum value for the cell.
- b. Set the *Min* parameter to specify the minimum value for the cell.
- c. Set the *Right* parameter to specify the number of digits allowed to the right of the decimal symbol.

---

**Note** The total number of digits to the left and the right of the decimal symbol cannot total more than 14.

---

- d. Set the *Left* parameter to specify the number of digits allowed to the left of the decimal symbol.

---

**Note** The total number of digits to the left and the right of the decimal symbol cannot total more than 14.

---

- e. Set the *Style* parameter to specify how the data is horizontally and vertically aligned in the cell.
2. Call the **SSSetCellType** or **SSSetCellTypeRange** function and set the *CellType* parameter to `SS_TYPE_FLOAT`.

## Formatting a Float Cell

Float cells are commonly used to represent currency values, which often display currency symbols. Numeric data also often displays separators between thousands and a decimal symbol, as shown in the following figure.

B
\$399.99
\$1,245.99
\$56.88
\$41,033.00
\$490.34

By default, the float cell uses the Windows international or regional settings for the currency symbol, the separator character, and the decimal symbol. If you prefer, you can specify custom symbols and a custom separator character.

If you prefer, you can specify the currency symbol, separator character, or decimal symbol globally for all float cells in the spreadsheet, as described in “Providing Default Formatting

Characters for Float Cells” on page 247. However, you can use the following instructions to override any formatting characters you have specified globally.

► **To format a float cell**

Spread  
Designer

In the Spread Designer,

1. Select a cell or block of cells.
2. Choose the cell type using one of the following methods:
  - Display the pop-up menu by right-clicking the cell or block of cells and then choose Float.
  - From the Cell menu, choose Cell Type, Float.
  - Choose the float button (  ) on the cell type toolbar.

The Cell Type dialog box appears.

3. In the Cell Type dialog box, on the Settings tab,
  - a. Display a currency symbol by selecting the Money check box and typing the character to display in the Money box.

---

**Note** You can provide a default currency symbol for all float cells using the **FloatDefCurrencyChar** property as described in “Providing Default Formatting Characters for Float Cells” on page 247. Use this setting to override the setting of the **FloatDefCurrencyChar** property.

---

- b. Specify the decimal symbol to display by typing the character to display in the Decimal box.

---

**Note** You can provide a default decimal symbol for all float cells using the **FloatDefDecimalChar** property as described in “Providing Default Formatting Characters for Float Cells” on page 247. Use this setting to override the setting of the **FloatDefDecimalChar** property.

---

- c. Display a separator character by selecting the Separator check box and typing the character to display in the Separator box.

---

**Note** You can provide a default separator character for all float cells using the **FloatDefSepChar** property as described in “Providing Default Formatting Characters for Float Cells” on page 247. Use this setting to override the setting of the **FloatDefSepChar** property.

---

- d. Choose the OK or Apply button.

ActiveX, VBX      At run time,

1. Set the **Col**, **Col2**, **Row**, and **Row2** properties as appropriate.
2. If you are selecting a block of cells, set the **BlockMode** property to True.
3. Set the **CellType** property to 2 (Float).
4. If you want the float cell to display a currency symbol,
  - a. Set the **TypeFloatMoney** property to True.
  - b. Specify the currency symbol by setting the **TypeFloatCurrencyChar** property.

---

**Note** You can provide a default currency symbol for all float cells using the **FloatDefCurrencyChar** property as described in “Providing Default Formatting Characters for Float Cells” on page 247. Use the **TypeFloatCurrencyChar** property to override the setting of the **FloatDefCurrencyChar** property.

---

5. If you want to use a different decimal symbol, provide the symbol using the **TypeFloatDecimalChar** property.

---

**Note** You can provide a default decimal symbol for all float cells using the **FloatDefDecimalChar** property as described in “Providing Default Formatting Characters for Float Cells” on page 247. Use the **TypeFloatDecimalChar** property to override the setting of the **FloatDefDecimalChar** property.

---

6. If you want the float cell to display a separator character,
  - a. Set the **TypeFloatSeparator** property to True.
  - b. Set the **TypeFloatSepChar** property to specify the separator character to display.

---

**Note** You can provide a default separator character for all float cells using the **FloatDefSepChar** property as described in “Providing Default Formatting Characters for Float Cells” on page 247. Use the **TypeFloatSepChar** property to override the setting of the **FloatDefSepChar** property.

---

7. If you selected a block of cells, set the **BlockMode** property back to False.

- DLL 1. If you want to set the format characters used to display currency, the thousands separator, or the decimal character, call the **SSSetTypeFloatExt** function.
- If you want to use the default characters used to display currency (\$), the thousands separator (,), and the decimal character (.), call the **SSSetTypeFloat** function.

Set the parameters for the functions as follows:

- a. If you want the cell to display a monetary symbol, set the *Style* parameter to FS\_MONEY.
- b. If you want the cell to display separator characters, set the *Style* parameter to FS\_SEPARATOR.
- c. If you want to use characters other than the default characters, using the **SSSetTypeFloatExt** function set the *lpFormat* parameter to point to the SS\_FLOATFORMAT structure.

Set the fields for the SS\_FLOATFORMAT structure as follows:

<b>To set this character . . .</b>	<b>Use this field . . .</b>
Currency symbol	<i>cCurrencySign</i>
Decimal symbol	<i>cDecimalSign</i>
Separator character	<i>cSeparator</i>

---

**Note** Use the **SSSetDefFloatFormat** function to specify formatting characteristics for all the float cells in the spreadsheet, as described in “Providing Default Formatting Characters for Float Cells” on page 247. If you want to override the settings from the **SSSetDefFloatFormat** function, use the **SSSetTypeFloatExt** function.

---

2. Call the **SSSetCellType** or **SSSetCellTypeRange** function and set the *CellType* parameter to SS\_TYPE\_FLOAT.

## Providing Default Formatting Characters for Float Cells

By default, the float cell uses the Windows international or regional settings for the currency symbol, the separator character, and the decimal symbol. If you prefer, you can specify custom symbols and a custom separator character globally for all float cells in the spreadsheet. However, you can use the instructions in “Formatting a Float Cell” on page 244 to override any formatting characters you have specified globally.

► **To provide default formatting characters for float cells**

ActiveX, VBX

At run time,

1. If you want the float cell to display the default currency symbol,
  - a. Set the **TypeFloatMoney** property to True.
  - b. Specify the currency symbol by setting the **FloatDefCurrencyChar** property.

---

**Note** You can provide a custom currency symbol for any float cell using the **TypeFloatCurrencyChar** property as described in “Formatting a Float Cell” on page 244. The **TypeFloatCurrencyChar** property overrides the setting of the **FloatDefCurrencyChar** property.

---

2. If you want to use the default decimal symbol, provide the symbol using the **FloatDefDecimalChar** property.

---

**Note** You can provide a custom decimal symbol for any float cell using the **TypeFloatDecimalChar** property as described in “Formatting a Float Cell” on page 244. The **TypeFloatDecimalChar** property overrides the setting of the **FloatDefDecimalChar** property.

---

3. If you want the float cell to display the default separator character,
  - a. Set the **TypeFloatSeparator** property to True.
  - b. Set the **FloatDefSepChar** property to specify the separator character to display.

---

**Note** You can provide a custom separator character for any float cells using the **TypeFloatSepChar** property as described in “Formatting a Float Cell” on page 244. The **TypeFloatSepChar** property overrides the setting of the **FloatDefSepChar** property.

---

DLL Using the **SSSetDefFloatFormat** function set the *lpFloatFormat* parameter to point to the **SS\_FLOATFORMAT** structure.

Set the fields for the **SS\_FLOATFORMAT** structure as follows:

<b>To set this character . . .</b>	<b>Use this field . . .</b>
Currency symbol	<i>cCurrencySign</i>
Decimal symbol	<i>cDecimalSign</i>
Separator character	<i>cSeparator</i>

---

**Note** You can provide a custom currency, decimal, and separator character for any float cell using the **SSSetTypeFloatExt** function as described in “Formatting a Float Cell” on page 244.

---

## Creating Integer Cells

Integer cells display integer values and offer a variety of formatting options, as shown in the following figure.

<b>B</b>
<b>432555</b>
<b>233</b>
<b>9863</b> 

You can customize the integer cell’s formatting, as well as characteristics such as maximum and minimum values and vertical data alignment. In addition, you can display a spin button in the cell and customize its behavior.

## Creating and Customizing an Integer Cell

You can specify the maximum and minimum values allowed in the cell. You can also vertically align the data in the cell to the top or bottom, or center it within the cell.

### ► To create and customize an integer cell

Spread  
Designer

In the Spread Designer,

1. Select a cell or block of cells.
2. Choose the cell type using one of the following methods:
  - Display the pop-up menu by right-clicking the cell or block of cells and then choose Integer.
  - From the Cell menu, choose Cell Type, Integer.
  - Choose the integer button (  ) on the cell type toolbar.

The Cell Type dialog box appears.

## Chapter 11

3. In the Cell Type dialog box, on the Settings tab,
  - a. Type the maximum value of the cell in the Max Value box.
  - b. Type the minimum value of the cell in the Min Value box.
  - c. Specify how the data is horizontally and vertically aligned in the cell by selecting an option button under Horizontal Alignment and Vertical Alignment.
  - d. Choose the OK or Apply button.

### ActiveX, VBX

At run time,

1. Set the **Col**, **Col2**, **Row**, and **Row2** properties as appropriate.
2. If you are selecting a block of cells, set the **BlockMode** property to True.
3. Set the **CellType** property to 3 (Integer).
4. Specify the maximum value of the cell using the **TypeIntegerMax** property.
5. Specify the minimum value of the cell using the **TypeIntegerMin** property.
6. Specify how the data is horizontally and vertically aligned in the cell by setting the **TypeHAlign** and **TypeVAlign** properties.
7. If you selected a block of cells, set the **BlockMode** property back to False.

### DLL

1. If you want to specify the text alignment or display a spin button, call the **SSSetTypeIntegerExt** function.

If you want to use the default text alignment and do not want to display a spin button, you can call the **SSSetTypeInteger** function.

Set the parameters for the functions as follows:

- a. Set the *Max* parameter (*IMax* for **SSSetTypeIntegerExt**) to specify the maximum value of the cell.
  - b. Set the *Min* parameter (*IMin* for **SSSetTypeIntegerExt**) to specify the minimum value of the cell.
  - c. Set the *IStyle* parameter for the **SSSetTypeIntegerExt** function to specify how the data is aligned in the cell.
2. Call the **SSSetCellType** or **SSSetCellTypeRange** function and set the *CellType* parameter to `SS_TYPE_INTEGER`.

## Displaying a Spin Button in an Integer Cell

Integer cells can display spin buttons to let users change the integer value quickly. You can specify how much the integer value increments or decrements when the user clicks the spin button, and whether the integer value wraps when it reaches the maximum or minimum value.

### ► To display a spin button in an integer cell

Spread  
Designer

In the Spread Designer,

1. Select a cell or block of cells.
2. Choose the cell type using one of the following methods:
  - Display the pop-up menu by right-clicking the cell or block of cells and then choose Integer.
  - From the Cell menu, choose Cell Type, Integer.
  - Choose the integer button (  ) on the cell type toolbar.

The Cell Type dialog box appears.

3. In the Cell Type dialog box, on the Settings tab,
  - a. Display a spin button in the cell by selecting the Display Spin Button check box under Spin Settings.
  - b. Specify the amount by which the value in the cell is incremented or decremented the user clicks the spin button by clicking the arrows or typing the value in the Increment spin box under Spin Settings.
  - c. Specify that the value in the cell wraps when the cell reaches the maximum or minimum value by selecting the Spin Wrap check box under Spin Settings.
  - d. Choose the OK or Apply button.

ActiveX, VBX

At run time,

1. Set the **Col**, **Col2**, **Row**, and **Row2** properties as appropriate.
2. If you are selecting a block of cells, set the **BlockMode** property to True.
3. Set the **CellType** property to 3 (Integer).
4. Display a spin button in the cell by setting the **TypeSpin** property to True.
5. Specify the amount by which the value in the cell is incremented or decremented when the user clicks the spin button by setting the **TypeIntegerSpinInc** property.

6. Specify that the value in the cell wraps when the cell reaches the maximum or minimum value by setting the **TypeIntegerSpinWrap** property to True.
  7. If you selected a block of cells, set the **BlockMode** property back to False.
- DLL
1. Call the **SSSetTypeIntegerExt** function and set the parameters as follows:
    - a. Set the *lStyle* parameter to IS\_SPIN to display a spin button in the cell.
    - b. Set the *lSpinInc* parameter to specify the amount by which the value in the cell is incremented or decremented the user clicks the spin button.
    - c. Set the *fSpinWrap* parameter to True to specify that the value in the cell wraps when the cell reaches the maximum or minimum value.
  2. Call the **SSSetCellType** or **SSSetCellTypeRange** function and set the *CellType* parameter to SS\_TYPE\_INTEGER.

## Creating Owner-Drawn Cells

### ► To create an owner-drawn cell

Spread  
Designer

In the Spread Designer,

1. Select a cell or block of cells.
2. Choose the cell type using one of the following methods:
  - Display the pop-up menu by right-clicking the cell or block of cells and then choose Owner Draw.
  - From the Cell menu, choose Cell Type, Owner Draw.
  - Choose the owner-drawn button () on the cell type toolbar.The Cell Type dialog box appears.
3. In the Cell Type dialog box, on the Settings tab,
  - a. Click the arrows or type a value in the Owner Draw ID spin box to specify an identification number for an owner-drawn cell.
  - b. Choose the OK or Apply button.

ActiveX, VBX

At run time,

1. Set the **Col**, **Col2**, **Row**, and **Row2** properties as appropriate.
2. If you are selecting a block of cells, set the **BlockMode** property to True.
3. Set the **CellType** property to 11 (Owner Drawn).

4. Specify the style of the owner-drawn cell with the **TypeOwnerDrawStyle** property. This style is called by the **DrawItem** event, which occurs when the cell needs to be drawn.
5. If you selected a block of cells, set the **BlockMode** property back to False.

- DLL
1. Call the **SSSetTypeOwnerDraw** function.
  2. Call the **SSSetCellType** or **SSSetCellTypeRange** function and set the *CellType* parameter to **SS\_TYPE\_OWNERDRAW**.

## Creating PIC Cells

PIC cells act as masks, limiting user entry for each item in the mask to characters sets that you specify. A sample PIC cell is shown in the following figure.

Phone
919-555-1234
212-555-5608
704-555-8005
206-555-1127
404-555-8812

You specify the mask using the available mask characters. If you prefer, you can left or right align the data, or center it in the cell.

► **To create a PIC cell**

Spread  
Designer

In the Spread Designer,

1. Select a cell or block of cells.
2. Choose the cell type using one of the following methods:
  - Display the pop-up menu by right-clicking the cell or block of cells and then choose Mask (Pic).
  - From the Cell menu, choose Cell Type, Mask (Pic).
  - Choose the PIC (Mask) button (###) on the cell type toolbar.

The Cell Type dialog box appears.

3. In the Cell Type dialog box,
  - a. Type the mask characters to use to create the mask in the Mask Definition box.

Use the following characters to specify what type of data the user can type into the mask:

Mask Character	Characters Allowed
X	Any ASCII character
9	0–9, decimal
A	A–Z, a–z plus any alphabetic character from larger character sets such as ANSI or Unicode
N	A–Z, a–z, 0–9, decimal
U	A–Z
L	a–z
H	(Hexadecimal) 0–9, a–f, A–F

- b. Specify text alignment by selecting an option button under Horizontal Alignment and Vertical Alignment.
  - c. Choose the OK or Apply button.

ActiveX, VBX      At run time,

1. Set the **Col**, **Col2**, **Row**, and **Row2** properties as appropriate.
2. If you are selecting a block of cells, set the **BlockMode** property to True.
3. Set the **CellType** property to 4 (PIC).
4. Create the mask by providing the mask characters using the **TypePicMask** property.

Use the following characters to specify what type of data the user can type into the mask:

Mask Character	Characters Allowed
X	Any ASCII character
9	0–9, decimal
A	A–Z, a–z plus any alphabetic character from larger character sets such as ANSI or Unicode
N	A–Z, a–z, 0–9, decimal
U	A–Z

Mask Character	Characters Allowed
L	a-z
H	(Hexadecimal) 0-9, a-f, A-F

5. Specify the text to display in an empty mask using the **TypePicDefaultText** property.
6. Specify the horizontal and vertical text alignment in the cell by setting the **TypeHAlign** and **TypeVAlign** properties.
7. If you selected a block of cells, set the **BlockMode** property back to False.

DLL 1. Call the **SSSetTypePic** function and set the parameters as follows:

- a. Set the *Mask* parameter to create the mask.

Use the following characters to specify what type of data the user can type into the mask:

Mask Character	Characters Allowed
X	Any ASCII character
9	0-9, decimal
A	A-Z, a-z plus any alphabetic character from larger character sets such as ANSI or Unicode
N	A-Z, a-z, 0-9, decimal
U	A-Z
L	a-z
H	(Hexadecimal) 0-9, a-f, A-F

- b. Set the *Style* parameter to specify the text alignment.

2. Call the **SSSetCellType** or **SSSetCellTypeRange** function and set the *CellType* parameter to `SS_TYPE_PIC`.

## Creating Picture Cells

Picture cells display pictures that you specify, as shown in the following figure.

	Organization	Photo
1	American Cancer Society	

The displayed picture can be either a bitmap or an icon. You can specify how the picture is displayed in the cell: centering it or stretching it to fit the cell. If you stretch the picture to fit the cell, the picture might become distorted if its proportions are changed. To prevent this, you can specify that pictures the cell stretches to fit maintain their original proportions.

### ► **To create a picture cell**

Spread  
Designer

In the Spread Designer,

1. Select a cell or block of cells.
2. Choose the cell type using one of the following methods:
  - Display the pop-up menu by right-clicking the cell or block of cells and then choose Picture.
  - From the Cell menu, choose Cell Type, Picture.
  - Choose the picture button () on the cell type toolbar.

The Cell Type dialog box appears.

3. In the Cell Type dialog, on the Settings tab,
  - a. Specify the picture to display by completing the following steps:
    - i. Choose the Picture button to display the Open dialog box.
    - ii. In the Open dialog box, designate the path and the file name for the picture to display, then choose the Open button.
  - b. Stretch the picture to fit the size of the cell by selecting the Stretch check box under Appearance.
  - c. If the picture is stretched, maintain the picture's proportions by selecting the Maintain Scale check box under Appearance.
  - d. Specify picture alignment by selecting an option button under Horizontal Alignment and Vertical Alignment.
  - e. Choose the OK or Apply button.

ActiveX, VBX

At run time,

1. Set the **Col**, **Col2**, **Row**, and **Row2** properties as appropriate.
2. If you are selecting a block of cells, set the **BlockMode** property to True.
3. Set the **CellType** property to 9 (Picture).
4. Specify the picture to display by setting the **TypePictPicture** property.

5. Specify how the picture is displayed in the cell by doing one or more of the following items:
    - If you want to center the picture in the cell, set the **TypePictCenter** property to True.
    - If you want to stretch the picture to fit the size of the cell, set the **TypePictStretch** property to True.
    - If you have stretched the picture to fit the size of the cell, but you want to maintain the picture's proportions, set the **TypePictMaintainScale** property to True.
  6. Specify the horizontal and vertical picture alignment in the cell by setting the **TypeHAlign** and **TypeVAlign** properties.
  7. If you selected a block of cells, set the **BlockMode** property back to False.
- DLL
1. If you want to specify a picture using a handle and palette, call the **SSSetTypePictureHandle** function and set the parameters as follows:
    - a. Set the *hPict* parameter to specify the handle to the bitmap.
    - b. Set the *hPal* parameter to specify the handle to the palette to use with the bitmap.
    - c. Set the *Style* parameter to VPS\_CENTER if you want the picture centered in the cell.
    - d. Set the *Style* parameter to VPS\_STRETCH if you want the picture to fit the size of the cell.
    - e. Set the *Style* parameter to VPS\_STRETCH OR VPS\_MAINTAINSCALE to maintain the picture's proportions when it is stretched.
    - f. Use the *Style* parameter to specify the picture alignment.
  2. If you do not want to specify a picture using a handle and palette, call the **SSSetTypePicture** function and set the parameters as follows:
    - a. Set the *PictName* parameter to specify the name of the bitmap to display.
    - b. Set the *Style* parameter to VPS\_CENTER if you want the picture centered in the cell.
    - c. Set the *Style* parameter to VPS\_STRETCH if you want the picture to fit the size of the cell.

- d. Set the *Style* parameter to `VPS_STRETCH` OR `VPS_MAINTAINSCALE` to maintain the picture's proportions when it is stretched.
  - e. Use the *Style* parameter to specify the picture alignment.
3. Call the `SSSetCellType` or `SSSetCellTypeRange` function and set the *CellType* parameter to `SS_TYPE_PICTURE`.

## Creating Static Text Cells

Static text cells act as labels. Users cannot edit the cell, only see its contents.

You can customize how static text cells display text, including setting the text's horizontal and vertical alignment and specifying whether text can wrap in the cell. In addition, if you want to provide the appearance of providing a mnemonic character, you can underline one of the letters in the text.

---

### Notes

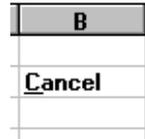
- Underlining a character in the cell does not provide the mnemonic function, only the appearance of providing a mnemonic. If you want to provide the mnemonic, you must add code to your project to do so.
  - You cannot scroll through text in a static text cell. You can resize the cell to fit the text. For more information, see “Resizing Cells” on page 196.
- 

Static text cells and header cells display the same background and text colors. If you prefer, static text cells can also display the same three-dimensional appearance as header cells. You can specify the background color for the cells, as well as the color of the text, and the colors used for the three-dimensional appearance.

## Creating and Customizing a Static Text Cell

By default, static text cells display text as left aligned and aligned at the top of the cell on a single line. You can change both the horizontal and vertical alignment of the text. If you want, the text in the cell can also appear on multiple lines, wrapping to succeeding lines when the text is too wide for the current line.

You can underline a letter in the text in the cell to appear as a mnemonic, as shown in the following figure.



Be aware that underlining a letter does not provide the mnemonic; you must add code to your project to provide the mnemonic if you want to do so.

You can set the background and text colors for all static text cells. Note that when you set these colors, you are also setting the colors for the header cells.

► **To create a static text cell**

Spread  
Designer

In the Spread Designer,

1. Select a cell or block of cells.
2. Choose the cell type using one of the following methods:
  - Display the pop-up menu by right-clicking the cell or block of cells and then choose Label(Static).
  - From the Cell menu, choose Cell Type, Label(Static).
  - Choose the static text (label) button () on the cell type toolbar.

The Cell Type dialog box appears.

3. In the Cell Type dialog box, on the Settings tab,
  - a. Type the text to display in the cell in the Text box.
  - b. Specify the horizontal and vertical text alignment by selecting an option button under Horizontal Alignment and Vertical Alignment.
  - c. If you want to underline a character in the text, select the Display Mnemonic check box, and in the Text box, type an ampersand (&) in front of the character you want to underline.
  - d. Specify that text wraps in the cell if the text string is wider than the cell width by selecting the Word Wrap check box.
  - e. Choose the OK or Apply button.

## Chapter 11

4. To set the color of the text in the cell or the background color of the cell,
  - a. From the General menu, choose Spreadsheet colors.
  - b. On the Colors tab in the Default dialog box,
    - i. From the Property drop-down list box,
      - Choose the **ShadowText** property to set the color of the text.
      - Choose the **ShadowColor** property to set the background color of the cell.
    - ii. Choose the Color button.
    - iii. In the Color dialog box, choose the color for the text or background, and then choose the OK button.
    - iv. Choose the OK or Apply button.

### ActiveX, VBX

At run time,

1. Set the **Col**, **Col2**, **Row**, and **Row2** properties as appropriate.
2. If you are selecting a block of cells, set the **BlockMode** property to True.
3. Set the **CellType** property to 5 (Static Text).
4. Provide the text to display in the cell using the **Text** or **Value** property.
5. Specify the horizontal and vertical text alignment by setting the **TypeHAlign** and **TypeVAlign** properties.
6. If you want to underline a character in the text, set the **TypeTextPrefix** property to True and when you provide the text for the cell, type an ampersand (&) in front of the character you want to underline.
7. Specify that text wraps in the cell if the text string is wider than the cell width by setting the **TypeTextWordWrap** property to True.
8. Specify the color of the text in the cell by setting the **ShadowText** property.
9. Specify the background color of the cell by setting the **ShadowColor** property.
10. If you selected a block of cells, set the **BlockMode** property back to False.

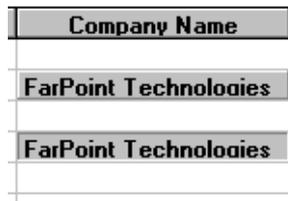
### DLL

1. Call the **SSSetTextStaticText** function and set the parameters as follows:
  - a. Set the *TextStyle* parameter to SS\_TEXT\_CENTER to center the text horizontally in the cell.
  - b. Set the *TextStyle* parameter to SS\_TEXT\_VCENTER to center the text vertically in the cell.

- c. Set the *TextStyle* parameter to `SS_TEXT_PREFIX` to underline a character in the text when you type an ampersand (&) in front of the character you want to underline.
  - d. Set the *TextStyle* parameter to `SS_TEXT_WORDWRAP` to have text wrap in the cell if the text string is wider than the cell width.
2. Call the **SSSetCellType** or **SSSetCellTypeRange** function and set the *CellType* parameter to `SS_TYPE_STATICTEXT`.
  3. Call the **SSSetValue** function to provide text for the cell.
  4. Call the **SSSetShadowColor** function and set the parameters as follows:
    - a. Set the *ShadowText* parameter to specify the color of the text in the cell.
    - b. Set the *ShadowColor* parameter to specify the background color of the cell.

## Displaying Three-Dimensional Appearance in a Static Text Cell

Static text cells can display a three-dimensional appearance that is created by a highlight and shadow area around the edge of the cell. The three-dimensional appearance can make the cell appear raised or lowered, as illustrated in the following figure.



You can specify the colors used for the three-dimensional appearance.

---

**Note** Because header cells are static text cells, when you set these colors, you are also setting the colors for the header cells.

---

► **To display three-dimensional appearance in a static text cell**

Spread  
Designer

In the Spread Designer,

1. Select a cell or block of cells.
2. Choose the cell type using one of the following methods:
  - Display the pop-up menu by right-clicking the cell or block of cells and then choose Label(Static).
  - From the Cell menu, choose Cell Type, Label (Static).
  - Choose the static text (label) button () on the cell type toolbar.

The Cell Type dialog box appears.

3. In the Cell Type dialog box,
  - a. Under Shadow, select a check box specifying either a raised or lowered three-dimensional appearance.
  - b. Choose the OK button.
4. To set the color of the shadow in the three-dimensional appearance,
  - a. From the General menu, choose Spreadsheet Colors.
  - b. On the Colors tab in the Default dialog box,
    - i. From the Property drop-down list box, choose the **ShadowDark** property.
    - ii. Choose the Color button.
    - iii. In the Color dialog box, choose the color for the shadow, and then choose the OK button.
    - iv. Choose the OK or Apply button.

ActiveX, VBX

At run time,

1. Set the **Col**, **Col2**, **Row**, and **Row2** properties as appropriate.
2. If you are selecting a block of cells, set the **BlockMode** property to True.
3. Set the **CellType** property to 5 (Static Text).
4. Provide a three-dimensional appearance by doing one of the following:
  - If you want the cell to display a raised three-dimensional appearance, set the **TypeTextShadow** property to True.
  - If you want the cell to display a lowered three-dimensional appearance, set the **TypeTextShadowIn** property to True.

---

**Note** If you set both the **TypeTextShadow** and **TypeTextShadowIn** properties to True, the cell displays a raised shadow effect as specified by the **TypeTextShadow** property.

---

5. To set the color of the shadow in the three-dimensional appearance, set the **ShadowDark** property.
  6. If you selected a block of cells, set the **BlockMode** property back to False.
- DLL
1. Call the **SSSetTextStaticText** function and set the *TextStyle* parameter to **SS\_TEXT\_SHADOW** to display a raised three-dimensional appearance or to **SS\_TEXT\_SHADOWIN** to display a lowered three-dimensional appearance.
  2. Call the **SSSetCellType** or **SSSetCellTypeRange** function and set the *CellType* parameter to **SS\_TYPE\_STATICTEXT**.
  3. Call the **SSSetShadowColor** function and set the parameters as follows:
    - a. Set the *ShadowDark* parameter to specify the color of the shadow portion of the three-dimensional appearance.
    - b. Set the *ShadowLight* parameter to specify the color of the highlight portion of the three-dimensional appearance.

## Creating Time Cells

Time cells display time values. You can specify one of several time formats, as shown in the following figure.

Time
12:12
12:12:12
23:23
23:23:23
12:12 

You can set maximum and minimum values for the time values, and you can customize the available time formats. In addition, you can provide a spin button to let users quickly change time values.

## Creating and Customizing a Time Cell

By default, the maximum time value for the cell is 23:59:59 (11:59:59 p.m.) and the minimum time value is 00:00:00 (12:00:00 midnight). You can specify other values as the maximum and minimum time values.

You can align the time value to the left or right, or center it in the cell.

If you prefer, you can provide a spin button to let users change the time value. If you provide a spin button, the portion of the time value where the cursor is located changes when the user presses the spin button.

### ► To create and customize a time cell

Spread  
Designer

In the Spread Designer,

1. Select a cell or block of cells.
2. Choose the cell type using one of the following methods:
  - Display the pop-up menu by right-clicking the cell or block of cells and then choose Time.
  - From the Cell menu, choose Cell Type, Time.
  - Choose the time button () on the cell type toolbar.

The Cell Type dialog box appears.

3. In the Cell Type dialog box, on the Settings tab,
  - a. Type the maximum time value for the cell in the Max Time box.
  - b. Type the minimum time value for the cell in the Min Time box.
  - c. To display a spin button in the cell, select the Display Spin Button check box.
  - d. Specify the horizontal and vertical text alignment by selecting an option button under Horizontal Alignment and Vertical Alignment.
  - e. Choose the OK or Apply button.

ActiveX, VBX

At run time,

1. Set the **Col**, **Col2**, **Row**, and **Row2** properties as appropriate.
2. If you are selecting a block of cells, set the **BlockMode** property to True.
3. Set the **CellType** property to 6 (Time).
4. Set the maximum time value for the cell using the **TypeTimeMax** property.

5. Set the minimum time value for the cell using the **TypeTimeMin** property.
  6. If you want to display a spin button in the cell, set the **TypeSpin** property to True.
  7. Specify how the text is horizontally and vertically aligned in the cell by setting the **TypeHAlign** and **TypeVAlign** properties.
  8. If you selected a block of cells, set the **BlockMode** property back to False.
- DLL
1. Call the **SSSetTypeTime** function and set the parameters as follows:
    - a. Set the *IMax* parameter to specify the maximum time value for the cell.
    - b. Set the *IMin* parameter to specify the minimum time value for the cell.
    - c. Set the *Style* parameter to TS\_SPIN to display a spin button in the cell.
    - d. Set the *Style* parameter to specify the text alignment.
  2. Call the **SSSetCellType** or **SSSetCellTypeRange** function and set the *CellType* parameter to SS\_TYPE\_TIME.

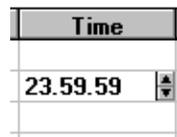
## Formatting a Time Cell

By default, the time cell displays time values in either 12- or 24-hour format, as specified by the Windows international or regional settings. You can specify that the cell displays time values in either format, independent of the Windows settings.

The time cell also uses the Windows international or regional settings to determine what separator character to use between the hour, minute, and if displayed, seconds, portions of the time value. If you prefer, you can specify a different character to use as the separator.

By default, the time cell does not display the seconds portion of the time value, but it can do so if you prefer.

The following figure illustrates a time cell that displays time in the 24-hour time format with seconds and uses a custom separator character.



### ► To format a time cell

Spread  
Designer

In the Spread Designer,

1. Select a cell or block of cells.
2. Choose the cell type using one of the following methods:
  - Display the pop-up menu by right-clicking the cell or block of cells and then choose Time.
  - From the Cell menu, choose Cell Type, Time.
  - Choose the time button () on the cell type toolbar.

The Cell Type dialog box appears.

3. In the Cell Type dialog box, on the Settings tab,
  - a. Select the 12- or 24-hour time format from the Format drop-down list box.
  - b. Display seconds by selecting the Display Seconds check box.
  - c. Type the character used to separate the hours, minutes, and seconds in the Separator box.
  - d. Choose the OK or Apply button.

ActiveX, VBX

At run time,

1. Set the **Col**, **Col2**, **Row**, and **Row2** properties as appropriate.
2. If you are selecting a block of cells, set the **BlockMode** property to True.
3. Set the **CellType** property to 6 (Time).
4. Specify whether the cell displays time in the 12- or 24-hour time format by setting the **TypeTime24Hour** property.
5. If you want the cell to display seconds in the time value, set the **TypeTimeSeconds** property to True.
6. Specify the character used to separate hours, minutes, and seconds by setting the **TypeTimeSeparator** property.
7. If you selected a block of cells, set the **BlockMode** property back to False.

DLL

1. Call the **SSSetTypeTime** function and set the parameters as follows:
  - a. Set the *Style* parameter to TS\_12HOUR or TS\_24HOUR to specify whether the cell displays time in the 12- or 24-hour time format.
  - b. Set the *Style* parameter to TS\_SECONDS to display seconds in the time value.

- c. Set the *Style* parameter to `TS_SPIN` to display a spin button in the cell.
- d. Set the *Style* parameter to specify the text alignment.

You can also use the *lpFormat* parameter to point to the `SS_TIMEFORMAT` structure. Set the fields for the `SS_TIMEFORMAT` structure as follows:

<b>To specify this format characteristic . . .</b>	<b>Use this field . . .</b>
Separator character	<code>cSeparator</code>
Displays spin button	<code>bSpin</code>
Displays seconds	<code>bSeconds</code>
Time format	<code>b24Hour</code>

2. Call the `SSSetCellType` or `SSSetCellTypeRange` function and set the *CellType* parameter to `SS_TYPE_TIME`.



# Chapter 12 Binding to Databases

Using Visual Basic®, you can bind the Spread ActiveX and VBX controls to databases. When you bind the fpSpread control to a database, it automatically displays the data from the database in columns, where each field is assigned to a column and each row represents a record.

You can customize how the data from the database is displayed in the spreadsheet, as well as how the fpSpread control interacts with a data control. If you are working with a large database, you can use virtual mode to save time and system resources.

The following topics provide more information about setting up and customizing data binding:

<b>Topic</b>	<b>On Page</b>
Connecting to a Database Using Default Settings	269
Displaying Selected Fields in Spreadsheet Columns	273
Customizing Headers for Bound Spreadsheets	275
Sizing Columns in Bound Spreadsheets	276
Assigning Cell Types for Bound Spreadsheets	278
Specifying Spreadsheet and Database Interaction	280
Using Virtual Mode with Bound Spreadsheets	282

## Connecting to a Database Using Default Settings

Spread 3.0 offers two ActiveX controls: the SPR32X30.OCX control, which lets you data bind the control to a database using the Remote Data Object (RDO) or the Data Access Object (DAO), and the FPSPR30.OCX control, which lets you data bind the control to a database using ActiveX Data Objects (ADO). It also offers the VBX control, SPRVBX30.VBX, which lets you data bind the control to a database using the Data Access Object. The following sections describe the steps you take to bind to a database using the different controls.

No matter which control you use, when you bind a spreadsheet to a database using the fpSpread control's default settings, the control automatically loads the database into the spreadsheet and configures the spreadsheet based on the data in the database. That is, the fields in the database table are assigned sequentially to spreadsheet columns, the field names are displayed as the column headers, the columns are sized based on the data in the

bound fields, and the cells in each column are assigned cell types based on the data in the bound fields.

You can override any of this default behavior in the fpSpread control, as described in

- “Displaying Selected Fields in Spreadsheet Columns” on page 273
- “Customizing Headers for Bound Spreadsheets” on page 275
- “Sizing Columns in Bound Spreadsheets” on page 276
- “Assigning Cell Types for Bound Spreadsheets” on page 278

### Binding the vaSpread Control using DAO or RDO

If you are using Visual Basic 3.0, 4.0, or 5.0 or other development environments that support either RDO or DAO, use either the vaSpread VBX control or the vaSpread DAO ActiveX control (SPR32X30.OCX). You can upgrade existing Spread 2.5 projects to use these Spread 3.0 controls as data-bound controls in your applications.

Complete the following instructions to bind your vaSpread VBX or DAO ActiveX control to a database using the default settings.

---

**Note** If you are binding to a RemoteData control, you cannot insert rows into the RemoteData control. For more information about the RemoteData control, refer to the Visual Basic documentation.

---

---

**Note** If the bound cells are edit cells, be aware that the default setting of the **TypeMaxEditLen** property is 60 characters. You must reset the **TypeMaxEditLen** property to 255 characters in response to the **DataColConfig** event to accommodate a text field in the bound database.

---

► **To bind a vaSpread VBX or DAO ActiveX control to a database using default vaSpread control settings**

- ActiveX, VBX
1. Create a data control on the form.
  2. Set the **DatabaseName** property for the data control.
  3. Set the **RecordSource** property for the data control.
  4. Add either the vaSpread VBX control or the vaSpread DAO ActiveX control (SPR32X30.OCX) to your project.
  5. Create a vaSpread control on the form.
  6. Set the **DataSource** property to the name of the data control.

## Binding the fpSpread Control using ADO

Spread 3.0 provides an ADO version of its ActiveX control (FPSPR30.OCX), which you can use to take advantage of ActiveX Data Objects (ADO) offered in Visual Basic 6.0 and later and other development environments that support ADO.

---

**Note** You cannot upgrade existing projects that use Remote Data Objects (RDO) or Data Access Objects (DAO) to use ActiveX Data Objects (ADO); however, you can convert projects as described in the upgrade information in the Read Me help file (SPREADME.HLP) that accompanies Spread 3.0.

---

ActiveX Data Objects (ADO) offer several advantages over RDO or DAO:

- Using ADO, you can create or change data binding settings at run time.
- ADO supports data binding through SQL statements, as well as other commonly used database formats, such as FoxPro® and Access.
- Controls that support data binding using ADO provide additional properties for customizing data binding, including the **DataMember** property, which lets you refer to a particular rowset name when the DataSource references multiple rowsets.

The following instructions describe how to set up data binding using the fpSpread ADO ActiveX control in Visual Basic 6.0 using the Jet 3.51 OLE DB provider. For more information about ADO, using FarPoint products and ADO, and instructions for setting up other types of data binding using ADO, consult the FarPoint Technologies web site at <http://www.fpoint.com>.

---

**Note** If the bound cells are edit cells, be aware that the default setting of the **TypeMaxEditLen** property is 60 characters. You must reset the **TypeMaxEditLen** property to 255 characters in response to the **DataColConfig** event to accommodate a text field in the bound database.

---

### ► To bind an fpSpread ADO ActiveX control to a database using default fpSpread control settings

- ActiveX
1. Add the Microsoft ADO Data Control 6.0 (OLEDB) to your project.
  2. Create a Microsoft ADO Data Control 6.0 (OLEDB) on the form.
  3. Right-click on the Data control, and choose ADODC Properties from the pop-up menu.

4. In the Property Pages, on the General property page, under Source of Connection,
  - a. Choose the Use Connection String option button.
  - b. Choose the Build button.
5. In the Data Link Properties dialog,
  - a. On the Provider tab,
    - i. Select Microsoft Jet 3.51 OLE DB Provider from the list of available providers.
    - ii. Choose the Next button.
  - b. On the Connection tab,
    - i. Under item 1 on the tab, provide the database name to connect to either by typing it in the field or by pressing the ellipses button and selecting the database using the Select Access Database dialog.
    - ii. Under item 2 on the tab, if necessary, enter the user information required to log on to the database.
    - iii. Choose the Test Connection button.
    - iv. If the connection succeeded, select OK in the message box. If the connection did not succeed, repeat steps 5.b.i through 5.b.iii.
  - c. Choose OK to continue and close the Data Link Properties dialog.
6. In the Property Pages, your connection string has been created in the Use Connection String text box.

You can use your connection string at run time to access this database connection.
7. In the Property Pages, choose the RecordSource page.
  - a. On the RecordSource page, under RecordSource, from the CommandType drop-down list box choose 2 - adCmdTable.
  - b. From the Table or Stored Procedure Name drop-down list box, choose the database table to which you want to connect.
  - c. Choose the OK button.
8. Add the FarPoint Spreadsheet ADO Control 3.0 (OLEDB) to your project.

9. Create an fpSpread control on the form.
10. Set the **DataSource** property to the name of the Microsoft ADO Data Control 6.0 (OLEDB).

## Displaying Selected Fields in Spreadsheet Columns

If you bind the spreadsheet using the default settings, the spreadsheet automatically displays all the fields in the database sequentially in the spreadsheet. If you prefer, the spreadsheet can display selected fields in designated columns.

You cannot use both the default settings and custom field and column specifications at the same time. Once you override the default settings for the spreadsheet, the spreadsheet only displays the fields in the columns you specify. For example, if you specify that the third field should be displayed in the first column, the spreadsheet does not display the other fields in the database unless you specify the columns in which they are to be displayed.

Use the following instructions to display only certain fields or to display fields in a different column order.

### ► To display a database field in a specific column

Spread  
Designer

---

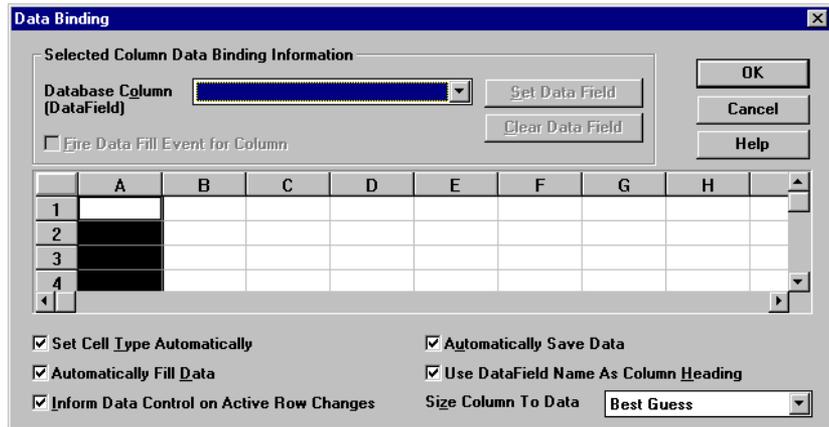
#### Notes

- To set data-binding properties, the Spread Designer view must be the active view. You cannot set data-binding properties for a spreadsheet that you open in the Spread Designer (that is, by choosing Open from the File menu).
  - If you bind columns to data fields using the Spread Designer and then change the record source for the data control, you must refresh the data control (for example, run your project) before resetting data fields with the Spread Designer. Unless the data control is refreshed, the new data fields will not be reflected in the Spread Designer.
- 

1. Follow the steps listed for “Connecting to a Database Using Default Settings” on page 269 to bind the spreadsheet to a database.
2. Open the Spread Designer.

3. From the General menu, choose Data Binding.

The Data Binding dialog box appears:



4. In the Data Binding dialog box,
  - a. Click a column to select it.
  - b. Under Selected Column Data Binding Information, select a database field from the Database Column drop-down list box.
  - c. Choose the Set Data Field button.
  - d. Repeat steps a, b, and c for each column for which you want to specify a field.
  - e. Choose the OK button.

ActiveX, VBX

1. Follow the steps listed for “Connecting to a Database Using Default Settings” on page 269 to bind the spreadsheet to a database.
2. At run time, set the **Col** property to designate the first column to display a specific field.
3. At run time, set the **DataField** property to the field name in the database table.
4. Repeat steps 2 and 3 for each column for which you want to specify a field.

To see the spreadsheet display the bound field, use Visual Basic’s **Refresh** method.

## Customizing Headers for Bound Spreadsheets

If you use the default data-binding settings, the column headers display the database field names. However, you can change the column headers to display your own text.

Note that once you have created a customized header for one column, the other columns will no longer display the database field names. Instead, they will display letters or numbers, as determined by the **ColHeaderDisplay** property.

### ► To display customized column headers

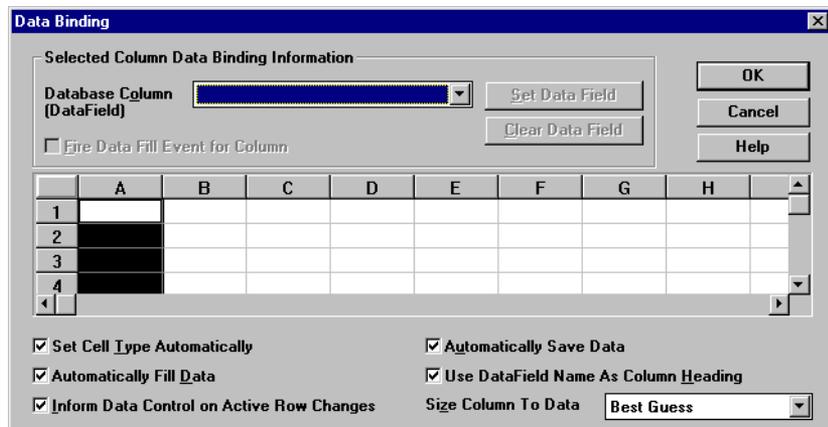
Spread  
Designer

#### Notes

- To set data-binding properties, the Spread Designer view must be the active view. You cannot set data-binding properties for a spreadsheet that you open in the Spread Designer (that is, by choosing Open from the File menu).
- If you bind columns to data fields using the Spread Designer and then change the record source for the data control, you must refresh the data control (for example, run your project) before resetting data fields with the Spread Designer. Unless the data control is refreshed, the new data fields will not be reflected in the Spread Designer.

1. Follow the steps listed for “Connecting to a Database Using Default Settings” on page 269 to bind the spreadsheet to a database.
2. Open the Spread Designer.
3. From the General menu, choose Data Binding.

The Data Binding dialog box appears:



4. In the Data Binding dialog box,
  - a. Clear the Use DataField Name As Column Heading check box.
  - b. Choose the OK button.
5. At run time,
  - a. Specify the column header in which to display the custom text by setting the **Col** and **Row** properties.
  - b. Supply the text for the column header using the **Text** property.

- ActiveX, VBX
1. Follow the steps listed for “Connecting to a Database Using Default Settings” on page 269 to bind the spreadsheet to a database.
  2. Set the **DAutoHeadings** property to False.
  3. At run time, specify the column header in which to display the custom text by setting the **Col** and **Row** properties.
  4. At run time, supply the text for the column header using the **Text** property.

## Sizing Columns in Bound Spreadsheets

If you use the default data-binding settings, the spreadsheet automatically sizes columns based on the length of the largest string in the corresponding bound database field. If you prefer, you can customize how the spreadsheet determines the column width when it sizes the columns automatically or you can manually size the columns.

► **To size columns in bound spreadsheets**

Spread  
Designer

---

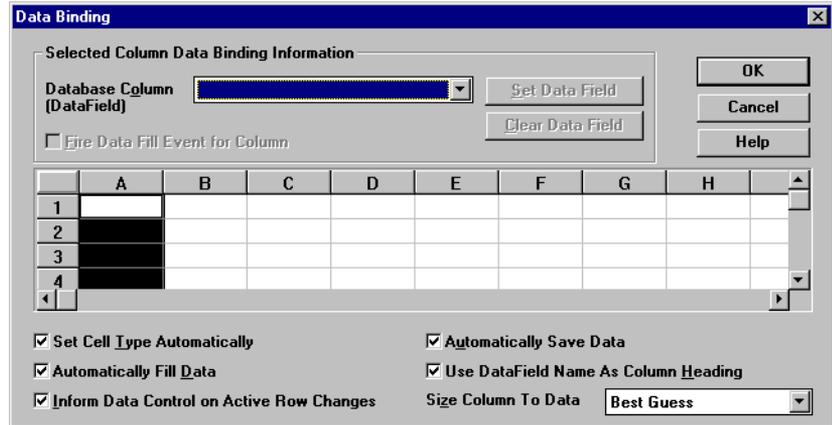
**Notes**

- To set data-binding properties, the Spread Designer view must be the active view. You cannot set data-binding properties for a spreadsheet that you open in the Spread Designer (that is, by choosing Open from the File menu).
  - If you bind columns to data fields using the Spread Designer and then change the record source for the data control, you must refresh the data control (for example, run your project) before resetting data fields with the Spread Designer. Unless the data control is refreshed, the new data fields will not be reflected in the Spread Designer.
- 

1. Follow the steps listed for “Connecting to a Database Using Default Settings” on page 269 to bind the spreadsheet to a database.
2. Open the Spread Designer.

- From the General menu, choose Data Binding.

The Data Binding dialog box appears:



- To have the columns automatically sized based on the data in the database fields, select one of the following from the Size Column to Data drop-down list box:
  - Max Column Width to size the column to the length of the longest string in the field
  - Best Guess to size the column to the length of the database field.

---

**Note** If the bound cells are edit cells, be aware that the default setting of the **TypeMaxEditLen** property is 60 characters. You must reset the **TypeMaxEditLen** property at run time to 255 characters to accommodate a text field in the bound database.

---

- If you do not want the columns automatically sized based on the data in the database fields, from the Size Column to Data drop-down list box choose Off and if you want, size the columns manually.

For instructions for sizing columns, see “Sizing Columns and Rows” on page 166.

- ActiveX, VBX
1. Follow the steps listed for “Connecting to a Database Using Default Settings” on page 269 to bind the spreadsheet to a database.
  2. To have the columns automatically sized based on the data in the database fields, set the **DAutoSizeCols** property to either of the following values:
    - 1 (Max Col Width) to size the column to the length of the longest string in the field
    - 2 (Best Guess) to size the column to the length of the database field (the default setting)

---

**Note** If the bound cells are edit cells, be aware that the default setting of the **TypeMaxEditLen** property is 60 characters. You must reset the **TypeMaxEditLen** property to 255 characters to accommodate a text field in the bound database.

---

3. If you do not want the columns automatically sized based on the data in the database fields, in the **DataColConfig** event, set the **DAutoSizeCols** property to 0 (Off) and if you want, size the columns manually.
4. At run time, specify the column width with the **ColWidth** property or use one of the other techniques available for sizing columns, as described in “Setting the Width of Columns” on page 167.

## Assigning Cell Types for Bound Spreadsheets

By default, the spreadsheet automatically assigns the cell type for each column’s cells based on the field type of the corresponding bound database field. You can override the assigned cell type, or you can turn off the automatic assignment and assign the cell types manually.

### ► To assign cell types to a bound spreadsheet

Spread  
Designer

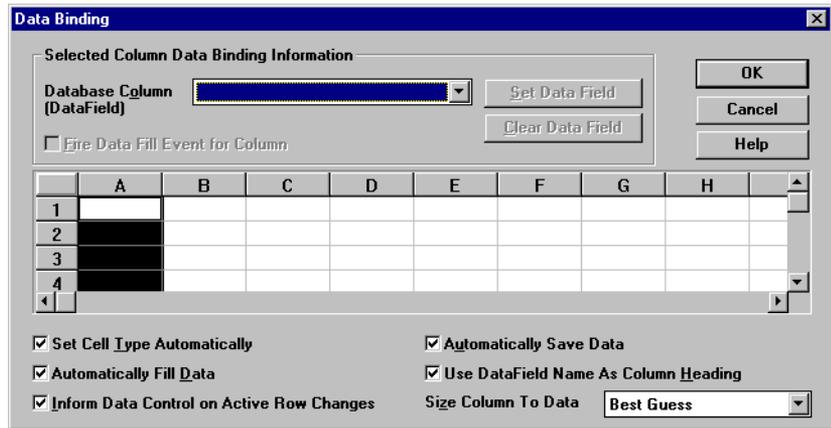
---

#### Notes

- To set data-binding properties, the Spread Designer view must be the active view. You cannot set data-binding properties for a spreadsheet that you open in the Spread Designer (that is, by choosing Open from the File menu).
  - If you bind columns to data fields using the Spread Designer and then change the record source for the data control, you must refresh the data control (for example, run your project) before resetting data fields with the Spread Designer. Unless the data control is refreshed, the new data fields will not be reflected in the Spread Designer.
-

1. Follow the steps listed for “Connecting to a Database Using Default Settings” on page 269 to bind the spreadsheet to a database.
2. If you want to override the cell types that are assigned automatically, use code in the **DataColConfig** event at run time. After each column is assigned a cell type, the **DataColConfig** event occurs. Therefore, in the **DataColConfig** event,
  - a. Return the assigned cell type using the **CellType** property at run time if you want to see what cell type the control assigned the cells in the column.
  - b. Change the assigned cell type by specifying the column with the **Col** property, setting the **Row** property to -1, and then setting the **CellType** property to the desired type.
3. If you want to set the cell types manually,
  - a. Open the Spread Designer.
  - b. From the General menu, choose Data Binding.

The Data Binding dialog box appears:



- c. In the Data Binding dialog box, to manually set the cell types,
  - i. Clear the Set Cell Type Automatically check box.
  - ii. Choose the OK button.

- ActiveX, VBX
1. Follow the steps listed for “Connecting to a Database Using Default Settings” on page 269 to bind the spreadsheet to a database.
  2. If you want to override the cell types that are assigned automatically, use code in the **DataColConfig** event to do so. After each column is assigned a cell type, the **DataColConfig** event occurs. Therefore, in the **DataColConfig** event,
    - a. Return the assigned cell type using the **CellType** property at run time if you want to see what cell type the control assigned the cells in the column.
    - b. Change the assigned cell type by specifying the column with the **Col** property, setting the **Row** property to  $-1$ , and then setting the **CellType** property to the desired type.
  3. If you want to set the cell types manually, set the **DAutoCellTypes** property to False, then assign cell types as described in Chapter 11, “Setting the Cell Type.”

## Specifying Spreadsheet and Database Interaction

You can specify whether or not the spreadsheet and database synchronize the active record. You can move through records in the spreadsheet independently of records in the database, or the spreadsheet and database can move through records synchronously.

When the user makes changes to a spreadsheet record, you can specify whether the changes are automatically written to the bound database table.

### ► To specify spreadsheet and database interaction

Spread  
Designer

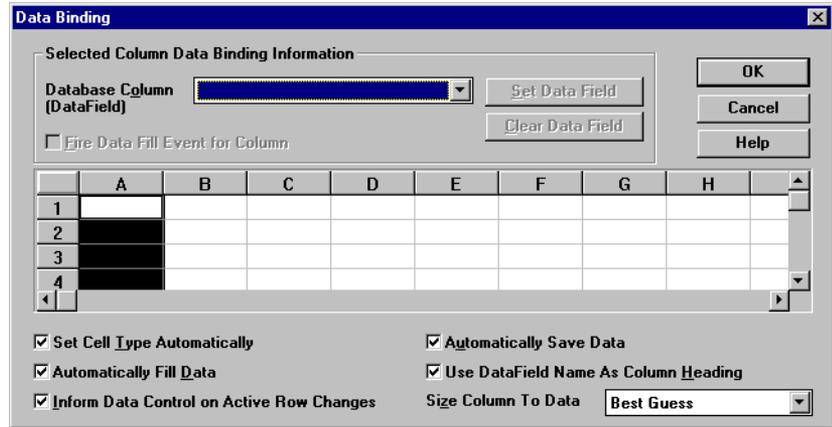
---

#### Notes

- To set data-binding properties, the Spread Designer view must be the active view. You cannot set data-binding properties for a spreadsheet that you open in the Spread Designer (that is, by choosing Open from the File menu).
  - If you bind columns to data fields using the Spread Designer and then change the record source for the data control, you must refresh the data control (for example, run your project) before resetting data fields with the Spread Designer. Unless the data control is refreshed, the new data fields will not be reflected in the Spread Designer.
- 

1. Follow the steps listed for “Connecting to a Database Using Default Settings” on page 269 to bind the spreadsheet to a database.
2. From the General menu, choose Data Binding.

The Data Binding dialog box appears:



3. In the Data Binding dialog box,
  - a. To move through records in the spreadsheet independently of records in the database, clear the Inform Data Control on Active Row Changes check box.
  - b. To manually save modified spreadsheet records to the database,
    - i. Clear the Automatically Save Data check box.
    - ii. At run time, set the **Action** property to 15 (Data Save).
  - c. To manually enter data from the database into the spreadsheet,
    - i. Clear the Automatically Fill Data check box.
    - ii. At run time, load data in the spreadsheet, using your application.
  - d. Choose the OK button.

ActiveX, VBX

1. Follow the steps listed for “Connecting to a Database Using Default Settings” on page 269 to bind the spreadsheet to a database.
2. To move through records in the spreadsheet independently of records in the database, set the **DInformActiveRowChange** property to False.
3. To manually save modified spreadsheet records to the database,
  - a. Set the **DAutoSave** property to False.
  - b. At run time, set the **Action** property to 15 (Data Save).

4. To manually enter data from the database into the spreadsheet,
  - a. Set the **DAutoFill** property to False.
  - b. At run time, load data in the spreadsheet, using your application.

## Using Virtual Mode with Bound Spreadsheets

Detailed instructions for using virtual mode are provided in “Using Virtual Mode” on page 130. However, if you are using virtual mode with a bound spreadsheet, you should be aware that the following interactions might not behave as you would expect:

- Querying the number of records

If you ask the data control how many records are in the database, it will tell you the number of records in the virtual buffer, not the number of records in the database.
- Searching the database

If you search for data in the database, the data control will not look in every record. It will only look in records that are in the virtual buffer.

As a rule, using virtual mode with a bound fpSpread control limits the data control to working with only the information in the virtual buffer, not the entire database. Keep this in mind when designing interactions with the data control.

# Chapter 13 Customizing the Spreadsheet Display

This chapter explains how to customize your spreadsheet display, including the spreadsheet border, grid lines, colors, and focus display.

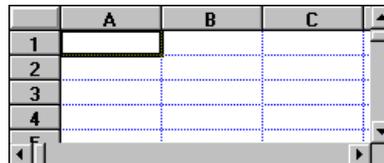
This chapter describes the tasks listed in the following table.

Topic	On Page
Customizing Grid Lines	283
Setting Spreadsheet Measurement Units	285
Preventing the Spreadsheet from Flickering	286
Setting the Gray Area Color	287
Working with the Control and Spreadsheet Borders	288
Hiding Selections When the Spreadsheet Does Not Have the Focus	290

## Customizing Grid Lines

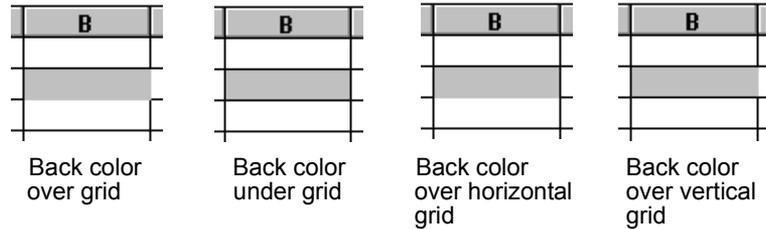
You can choose to show the horizontal grid lines, the vertical grid lines, or both. You can have the grid lines display as solid or dotted lines and you can specify the color of the grid lines.

The following figure illustrates a control with dotted grid lines.



By default, the background color of a cell overlaps (hides) the right and bottom sides of the cell's grid lines. You can specify that the complete grid lines display or that only the right or bottom sides are covered.

The following figure shows four cells in different spreadsheets, each with a different setting for how and whether the background color (green) of the cell overlaps the grid lines (red).



Individual cells or blocks of cells can also display borders, in addition to grid lines. Note that borders provide a custom appearance for cells, while grid lines serve to separate columns and rows. For more information about cell borders, see “Customizing Cell Borders” on page 202.

► **To customize spreadsheet grid lines**

Spread Designer

In the Spread Designer,

1. From the Display menu, choose Grid Lines or press the Grid Lines button () in the tool bar.
2. On the Grid Lines tab in the Display dialog box,
  - a. To hide horizontal grid lines, clear the Show Horizontal Lines check box under Settings.
  - b. To hide vertical grid lines, clear the Show Vertical Lines check box under Settings.
  - c. To display grid lines as dotted, clear the Make Grid Lines Solid check box under Settings.
  - d. To display the grid lines over the background color, select the appropriate option button under Back Color Displays.
  - e. Choose the Apply button.
3. From the General menu, choose Spreadsheet Colors or press the Spreadsheet Colors button () in the tool bar.

4. On the Colors tab in the Default dialog box,
  - a. Select the **GridColor** property from the Property drop-down list box.
  - b. Choose the Color button to display the Color dialog box.
  - c. In the Color dialog box,
    - i. Select a basic color or customize your own color.
    - ii. Choose the OK button.
  - d. Choose the OK or Apply button.

- ActiveX, VBX
1. If you want to hide the horizontal grid lines, set the **GridShowHoriz** property to False.
  2. If you want to hide the vertical grid lines, set the **GridShowVert** property to False.
  3. If you want to display the grid lines as dotted, set the **GridSolid** property to False.
  4. If you want to display the grid lines over the background color, set the **BackColorStyle** property to 1 (Under Grid), 2 (Over Horizontal Grid Only), or 3 (Over Vertical Grid Only).
  5. Set the **GridColor** property.

- DLL
1. Call the **SSSetGridType** function.
  2. Call the **SSSetGridColor** function.
  3. If you want to display the grid lines over the background color, call the **SSSetBackColorStyle** function, and set the value of the *wStyle* parameter to either `SS_BACKCOLORSTYLE_UNDERGRID`, `SS_BACKCOLORSTYLE_OVERHORIZGRIDONLY`, or `SS_BACKCOLORSTYLE_OVERVERTGRIDONLY`.

## Setting Spreadsheet Measurement Units

The column width of the spreadsheet is specified in internal units. This value is calculated as 1 unit being equal to the width of one character of the Windows system fixed font.

The row height is based on the current font. If the font is changed, the height of the row will change to the new size of the font. For more information on changing the font, see “Setting the Font” on page 207.

You can change the unit type to twips or to a constant value provided by the fpSpread control.

---

**Note** You should set the unit type before using any other spreadsheet property, function, or method. Failure to do so can produce unexpected results.

---

► **To set the spreadsheet measurement units**

Spread  
Designer

In the Spread Designer,

1. From the General Menu, choose Unit Type.
2. On the Unit Type tab in the Default dialog box,
  - a. Select one of the measurement option buttons under Settings.
  - b. Choose the OK or Apply button.

ActiveX, VBX Set the **UnitType** property.

DLL Call the **SSSetUnitType** function.

## Preventing the Spreadsheet from Flickering

By default, when you set a property or function that changes the spreadsheet's appearance or data, the spreadsheet automatically refreshes itself. However, when you make multiple changes or add large amounts of data, the spreadsheet appears as if it were flickering because the spreadsheet is redrawing itself multiple times. You can prevent these multiple spreadsheet redraws and flickering. Your application will also run faster.

---

**Note** Even if you turn off the spreadsheet refreshing, some changes might be shown on the screen, depending on the type of changes and the order in which they are made.

---

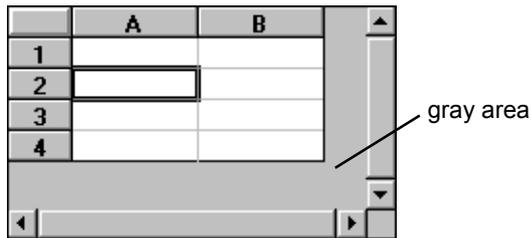
► **To prevent spreadsheet flickering**

- ActiveX, VBX
1. Set the **ReDraw** property to False.
  2. Perform the necessary operations.
  3. Set the **ReDraw** property to True.

- DLL
1. Call the **SSSetBool** function, and set the parameters as follows:
    - a. Set the value of the *nIndex* parameter to `SSB_REDRAW`.
    - b. Set the value of the *bNewVal* parameter to `FALSE`.
  2. Perform the necessary operations.
  3. Call the **SSSetBool** function, and set the parameters as follows:
    - a. Set the value of the *nIndex* parameter to `SSB_REDRAW`.
    - b. Set the value of the *bNewVal* parameter to `TRUE`.

## Setting the Gray Area Color

If you scroll beyond the maximum columns and rows in a spreadsheet, a gray area appears indicating no more cells exist, as illustrated in the following figure. In this example, the maximum number of columns is 2 and the maximum number of rows is 4.



By default, the color of this gray area is light gray. You can customize the gray area color.

### ► To set the gray area color

Spread  
Designer

In the Spread Designer,

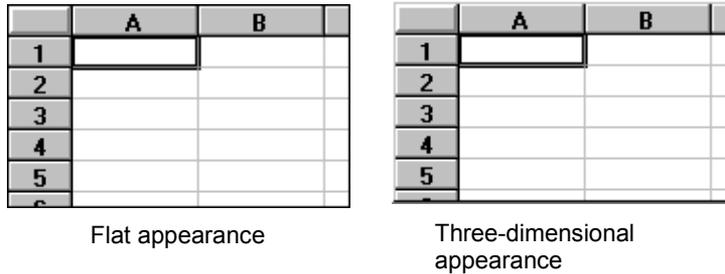
1. From the General menu, choose Spreadsheet Colors or press the Spreadsheet Colors button (  ) in the tool bar.
2. On the Colors tab in the Default dialog box, select GrayAreaBackColor from the Property drop-down list box.
3. Choose the Color button to display the Color dialog box.
4. Select a basic color or customize your own color.
5. Choose the OK button.
6. Choose the OK or Apply button.

## Chapter 13

- ActiveX, VBX Set the **GrayAreaBackColor** property.
- DLL Call the **SSSetGrayAreaColor** function.

## Working with the Control and Spreadsheet Borders

By default, the control displays either a flat or three-dimensional appearance as shown in the following figure and as explained in “Changing the Three-Dimensional Appearance.”



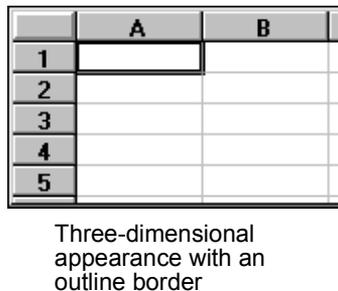
The spreadsheet also displays a border along the bottom and right edge.

## Changing the Three-Dimensional Appearance

If you use the fpSpread ActiveX control or VBX control, the control’s default border appearance is flat. If you use the fpSpread DLL control, the control’s default border appearance depends on the style `WS_BORDER` and the extended Windows styles to paint the border.

The control can display a flat appearance, a raised three-dimensional appearance, or a raised three-dimensional appearance with an outline border.

An fpSpread control with a raised three-dimensional appearance with an outline border is shown in the following figure.



► **To change the three-dimensional appearance**

Spread  
Designer

1. From the General menu, choose Spreadsheet Environment or press the Spreadsheet Environment button (  ) in the toolbar.
2. On the General tab in the Sheet Environment Settings dialog box, select a border appearance from the Appearance drop-down list box under Settings.
3. Choose the OK or Apply button.

ActiveX, VBX

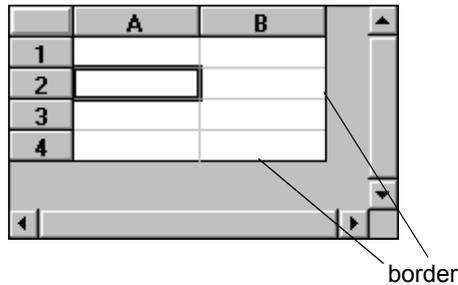
Set the **Appearance** property to either 1 (3D) or 2 (3DWithBorder).

DLL

Call the **SSSetAppearance** function and set the *wAppearance* parameter to SS\_APPEARANCE\_FLAT, SS\_APPEARANCE\_3D, or SS\_APPEARANCE\_3DWITHBORDER.

## Hiding Spreadsheet Borders

By default, the spreadsheet displays a border along the bottom and right edges of the spreadsheet, as shown in the following figure.



If you prefer, you can hide the spreadsheet border.

► **To hide the spreadsheet border**

Spread  
Designer

In the Spread Designer,

1. From the General menu, choose Spreadsheet Environment or press the Spreadsheet Environment button (  ) in the tool bar.
2. In the Sheet Environment Settings dialog box on the General tab, select the Hide Border check box under Settings.
3. Choose the OK or Apply button.

ActiveX VBX

Set the **NoBorder** property to True.

DLL Call the **SSSetBool** function, and set the parameters as follows:

1. Set the value of the *nIndex* parameter to `SSB_NOBORDER`.
2. Set the value of the *bNewVal* parameter to `TRUE`.

## Hiding Selections When the Spreadsheet Does Not Have the Focus

By default, the spreadsheet displays the focus rectangle or selection highlighting when the spreadsheet does not have the focus. If you prefer, the spreadsheet can hide the focus rectangle or selection highlighting when the spreadsheet does not have the focus.

If the spreadsheet does not display the focus rectangle or selection highlighting when the spreadsheet does not have the focus, the focus rectangle or selection highlighting will be restored when the spreadsheet regains the focus.

### ► To hide selections when the spreadsheet does not have the focus

Spread  
Designer

In the Spread Designer,

1. From the General menu, choose Spreadsheet Environment or press the Spreadsheet Environment button () in the tool bar.
2. On the General tab in the Sheet Environment Settings dialog box,
  - a. Clear the Retain Selected Block check box under Settings.
  - b. Choose the OK or Apply button.

ActiveX, VBX Set the **RetainSelBlock** property to `False`.

DLL Call the **SSSetBool** function, and set the parameters as follows:

1. Set the value of the *nIndex* parameter to `SSB_RETAINSELBLOCK`.
2. Set the value of the *bNewVal* parameter to `FALSE`.

# Chapter 14 Customizing User Interaction

You can customize many aspects of user interaction, including allowing users to drag and drop cells, displaying custom cursors, and customizing navigation.

You can customize user interaction as described in the following topics:

Topic	On Page
Allowing Users to Drag and Drop Cells	291
Displaying Scroll Bars	293
Specifying the Pointer	296
Turning Off the Spreadsheet Beep	297
Customizing Navigation	298
Using Arrow Keys to Move Between Cells	301
Turning Off the Clipboard Shortcut Keys	301
Changing or Turning Off the F2, F3, and F4 Action Keys	302
Specifying When Buttons Appear	305
Providing Text Tips	306

## Allowing Users to Drag and Drop Cells

You can specify whether the user can select a cell or range of cells and drag and drop them to a new location in the same spreadsheet. When the mouse button is released and the cell block is dropped, the **DragDropBlock** event occurs (ActiveX, VBX controls) or the spreadsheet sends the **SSM\_DRAGDROP** message (DLL control).

Whether the data in the cells is copied or moved by the drag-and-drop operation is determined by the user's actions as described in "Copying Data" on page 84 and "Moving Data" on page 91.

► **To allow users to drag and drop cells**

Spread  
Designer

In the Spread Designer,

1. From the General menu, choose Spreadsheet Environment or press the Spreadsheet Environment button () in the toolbar.
2. On the General tab in the Sheet Environment Settings dialog box,
  - a. Select the Cell Dragdrop check box under Settings.
  - b. Choose the OK or Apply button.

---

**Note** When the Operation Mode is set to Row Mode (SheetMode tab in the Sheet Environment Settings dialog box), if you select the Cell Dragdrop check box and you drag and drop a cell or block of cells, the entire row or rows containing the selected cell or cells are dragged and dropped.

---

ActiveX, VBX

Set the **AllowDragDrop** property to True.

---

**Note** When the **OperationMode** property is set to 2 (Row Mode), if you set the **AllowDragDrop** property to True and you drag and drop a cell or block of cells, the entire row or rows containing the selected cell or cells are dragged and dropped.

---

DLL

Call the **SSSetBool** function, and set the parameters as follows:

1. Set the value of the *nIndex* parameter to `SSB_ALLOWDRAGDROP`.
2. Set the value of the *bNewVal* parameter to `TRUE`.

---

**Note** When the *wMode* parameter of the **SSSetOperationMode** function is set to `SS_OPMODE_ROWMODE` and the `SSB_ALLOWDRAGDROP` boolean of the **SSSetBool** function is set to `TRUE`, and you drag and drop a cell or block of cells, the entire row or rows containing the selected cell or cells are dragged and dropped.

---

## Displaying Scroll Bars

You can specify whether scroll bars display for your spreadsheet. If you decide to display scroll bars, you can designate when the scroll bars display.

### Specifying Whether Scroll Bars Are Displayed

By default, the spreadsheet displays both a horizontal and a vertical scroll bar. You can choose to have the spreadsheet display no scroll bars, or you can have the spreadsheet display only the horizontal scroll bar or vertical scroll bar.

If you prefer, you can have the spreadsheet display the scroll bars only when the size of the spreadsheet is larger than the view, as described in “Displaying Scroll Bars Only When Needed” on page 294.

#### ► To specify whether scroll bars are displayed

Spread  
Designer

In the Spread Designer,

1. From the Display menu, choose Scroll Bars or press the Scroll Bars button () in the tool bar.
2. On the Scroll Bars tab in the Display dialog box,
  - a. Specify which scroll bars you want to display:
    - If you want to hide the scroll bars, select the None option button under Display.
    - If you want to display the scroll bars, select the Horizontal, Vertical, or Both option button under Display.
  - b. Choose the OK or Apply button.

ActiveX, VBX

Set the **ScrollBars** property as follows:

- If you want to hide the scroll bars, set the **ScrollBars** property to 0 (None).
- If you want to display the scroll bars, set the **ScrollBars** property to either 1 (Horizontal), 2 (Vertical), or 3 (Both).

DLL

Call the **SSSetBool** function, and set the parameters as follows:

1. If you want to hide the scroll bars,
  - a. Set the value of the *nIndex* parameter to `SSB_HORZSCROLLBAR` or `SSB_VERTSCROLLBAR`.
  - b. Set the value of the *bNewVal* parameter to `FALSE`.

2. If you want to display the scroll bars,
  - a. Set the value of the *nIndex* parameter to `SSB_HORZSCROLLBAR` or `SSB_VERTSCROLLBAR`.
  - b. Set the value of the *bNewVal* parameter to `TRUE`.

## Displaying Scroll Bars Only When Needed

By default, the spreadsheet displays the scroll bars you specified for it to display at all times. Alternatively, the control can display scroll bars only when the size of the spreadsheet is larger than the size of the view.

Which scroll bars are displayed at any time is determined by the setting of the **ScrollBars** property, as described in “Specifying Whether Scroll Bars Are Displayed” on page 293. For example, if the **ScrollBars** property is set to 2 (Vertical), the spreadsheet will not display the horizontal scroll bar.

### ► To display scroll bars only when needed

Spread  
Designer

In the Spread Designer,

1. From the Display menu, choose Scroll Bars or press the Scroll Bars button (  ) in the tool bar.
2. On the Scroll bars tab in the Display dialog box,
  - a. Select the Display Only if Needed check box.
  - b. Choose the OK or Apply button.

ActiveX, VBX

Set the **ScrollBarExtMode** property to `True`.

DLL

Call the **SSSetBool** function, and set the parameters as follows:

1. Set the value of the *nIndex* parameter to `SSB_SCROLLBAREXTMODE`.
2. Set the value of the *bNewVal* parameter to `TRUE`.

## Limiting the Scroll Box Range

By default, the user can use the scroll box to scroll through the entire spreadsheet, up to the maximum number of rows and columns. For example, if the maximum number of rows is 500, the user can scroll through rows 1 through 500.

You can limit how far the user can scroll using the scroll box, limiting the user to moving the scroll box through only the rows and columns that contain data, or through 20 rows and 8 columns, whichever is greater.

► **To limit the scroll box range**

Spread  
Designer

In the Spread Designer,

1. From the Display menu, choose Scroll Bars or press the Scroll Bars button () in the toolbar.
2. On the Scroll Bars tab in the Display dialog box, clear the Scroll Box Reflects Max Rows check box.
3. Choose the OK or Apply button.

ActiveX, VBX

Set the **ScrollBarShowMax** property to False.

DLL

Call the **SSSetBool** function, and set the parameters as follows:

1. Set the value of the *nIndex* parameter to `SSB_SCROLLBARSHOWMAX`.
2. Set the value of the *bNewVal* parameter to `FALSE`.

## Synchronizing the Scroll Box with Columns and Rows

By default, when the user drags the horizontal or vertical scroll box, the columns or rows do not move with the scroll box. In other words, the scroll box is not synchronized with the columns or rows. You can specify that the columns, the rows, or both are updated continuously as the user drags the horizontal or vertical scroll box.

► **To synchronize the scroll box with columns or rows**

Spread  
Designer

In the Spread Designer,

1. From the Display menu, choose Scroll Bars or press the Scroll Bars button () in the toolbar.
2. On the Scroll Bars tab in the Display dialog box, select the Vertical Scroll Bar, Horizontal Scroll Bar, or Both Scroll Bars option button under Scroll Bar Track.
3. Choose the OK or Apply button.

## Chapter 14

ActiveX, VBX Set the **ScrollBarTrack** property to 1(Horizontal), 2 (Vertical), or 3 (Both).

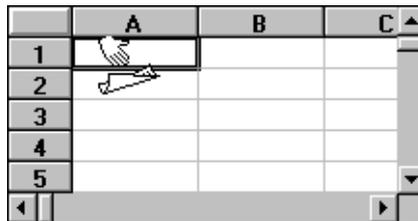
DLL Call the **SSSetBool** function, and set the parameters as follows:

1. If you want to synchronize the horizontal scroll box,
  - a. Set the value of the *nIndex* parameter to `SSB_HSCROLLBARTRACK`.
  - b. Set the value of the *bNewVal* parameter to `TRUE`.
2. If you want to synchronize the vertical scroll box,
  - a. Set the value of the *nIndex* parameter to `SSB_VSCROLLBARTRACK`.
  - b. Set the value of the *bNewVal* parameter to `TRUE`.

## Specifying the Pointer

You can specify the pointer to display for different areas of the spreadsheet. You can choose one of four pointers (default pointer, arrow, column resize, or row resize). If you are using the ActiveX control, you can define a custom icon to display as the pointer.

In the following illustration, the spreadsheet allows users to drag and drop cells using the mouse, so the pointer changes to a custom drag-and-drop pointer when it is over the cell boundary.



### ► To specify the pointer and where it is displayed

Spread  
Designer

In the Spread Designer,

1. From the Display menu, choose Pointer or press the Pointer button () in the tool bar.
2. On the Pointer tab in the Display dialog box, select an option from the Pointer Location drop-down list box to specify the area of the spreadsheet for which you want to display the pointer.
3. Select the icon from the Pointer Appearance combo box to define an icon for the pointer.

4. Choose the OK or Apply button.
5. Repeat steps 2–4 for each area of the spreadsheet for which you want to specify a pointer.

ActiveX At run time,

1. If you want to specify a user-defined icon, specify the icon with the **CursorIcon** property.
2. Specify the area of the spreadsheet for which you want to display the pointer with the **CursorType** property.
3. If you set the **CursorIcon** property in step 1, set the **CursorStyle** property to 0 (User Defined). Otherwise, set the **CursorStyle** property to a value other than 0 (User Defined) to specify the pointer for the area you set in step 2.
4. Repeat steps 1–3 for each area of the spreadsheet for which you want to specify a pointer.

VBX At run time,

1. Specify the area of the spreadsheet for which you want to display the pointer with the **CursorType** property.
2. Specify the pointer for the area you set in step 2 with the **CursorStyle** property.
3. Repeat steps 1–2 for each area of the spreadsheet for which you want to specify a pointer.

DLL Call the **SSSetCursor** function and set the *wType* and *hCursor* parameters.

## Turning Off the Spreadsheet Beep

When the user tries to perform an action that is not allowed, the spreadsheet beeps. For example, if the user tries to type a number into an edit field for which you have defined the character set to be letters only, the spreadsheet beeps.

If you prefer, you can turn off the spreadsheet beep.

### ► To turn off the spreadsheet beep

- Spread Designer In the Spread Designer,
1. From the General menu, choose Spreadsheet Environment or press the Spreadsheet Environment button () in the tool bar.
  2. On the General tab in the Sheet Environment Settings dialog box,
    - a. Select the Beep Off check box under Settings.
    - b. Choose the OK or Apply button.
- ActiveX, VBX Set the **NoBeep** property to True.
- DLL Call the **SSSetBool** function, and set the parameters as follows:
1. Set the value of the *nIndex* parameter to `SSB_NOBEEP`.
  2. Set the value of the *bNewVal* parameter to `TRUE`.

## Customizing Navigation

When the spreadsheet is in edit mode, you can specify if and where the active cell moves when the user presses the Enter key. You can also use the Tab key to move the active cell to the next cell in the spreadsheet.

You can also specify that the active cell does not change when you move the focus to the spreadsheet from another control using the mouse.

If you prefer, the spreadsheet does not have to display a focus rectangle around the active cell.

For more information on navigating within cells and a table describing navigation methods, see “Navigation” on page 27.

## Using the Enter Key to Move Between Cells

You can control where the focus rectangle moves (up, down, left, right, next cell, previous cell, or first column of the next row) when the user presses the Enter key when edit mode is on. You can also specify that the focus rectangle stays at the same cell.

By default, when the user presses the Enter key edit mode is turned off and the active cell does not change.

► **To use the Enter key to move between cells**

Spread  
Designer

In the Spread Designer,

1. From the General menu, choose Spreadsheet Environment or press the Spreadsheet Environment button (  ) in the toolbar.
2. On the Enter Key Action tab in the Sheet Environment Settings dialog box, select the appropriate option button under Moves Active Cell.
3. Choose the OK or Apply button.

ActiveX, VBX

Set the **EditEnterAction** property as follows:

- If you want to specify that the focus rectangle stays at the current cell when the user presses the Enter key, set the property to 7 (Same).
- If you want to specify that the focus rectangle moves to another cell when the user presses the Enter key, set the property to the appropriate setting.

DLL

Call the **SSSetEditEnterAction** function and set the *wAction* parameter to specify the action.

## Using the Tab Key to Move Between Cells

By default, pressing the Tab key moves the focus rectangle to the next control on the dialog. If you prefer, when the user presses the Tab key the active cell can move to the next cell to the right within the spreadsheet.

► **To use the Tab key to move the focus between cells**

Spread  
Designer

In the Spread Designer,

1. From the General menu, choose Spreadsheet Environment or press the Spreadsheet Environment button (  ) in the toolbar.
2. On the General tab in the Sheet Environment Settings dialog box,
  - a. Select the Process Tab Key check box under Settings.
  - b. Choose the OK or Apply button.

ActiveX, VBX

Set the **ProcessTab** property to True.

DLL

Call the **SSSetBool** function, and set the parameters as follows:

1. Set the value of the *nIndex* parameter to `SSB_PROCESSTAB`.
2. Set the value of the *bNewVal* parameter to `FALSE`.

## Preventing Active Cell Change on Focus

By default, when the user moves the focus to the spreadsheet from another control using the mouse, the active cell moves to the cell under the pointer. You can specify that the active cell does not change.

### ► To prevent the active cell from changing on focus

Spread  
Designer

In the Spread Designer,

1. From the General menu, choose Spreadsheet Environment or press the Spreadsheet Environment button () in the toolbar.
2. On the General tab in the Sheet Environment Settings dialog box, clear the On Focus Set Cell to Cursor check box under Settings.
3. Choose the OK or Apply button.

ActiveX, VBX

Set the **MoveActiveOnFocus** property to False.

DLL

Call the **SSSetBool** function, and set the parameters as follows:

1. Set the value of the *nIndex* parameter to `SSB_MOVEACTIVEONFOCUS`.
2. Set the value of the *bNewVal* parameter to `TRUE`.

## Hiding the Focus Rectangle

You can specify that edit mode is always on in the spreadsheet. When edit mode is always on, the spreadsheet acts like a form, where the user can enter data in any data field without turning edit mode on, and the focus rectangle is not displayed.

For more information, see “Keeping Edit Mode Permanently On” on page 200.

### ► To hide the focus rectangle

Spread  
Designer

In the Spread Designer,

1. From the General menu, choose Spreadsheet Environment or press the Spreadsheet Environment button () in the toolbar.
2. On the General tab in the Sheet Environment Settings dialog box,
  - a. Select the Make Edit Mode Permanent check box under Settings.
  - b. Choose the OK or Apply button.

ActiveX, VBX

Set the **EditModePermanent** property to True.

- DLL Call the **SSSetBool** function, and set the parameters as follows:
1. Set the value of the *nIndex* parameter to `SSB_EDITMODEPERMANENT`.
  2. Set the value of the *bNewVal* parameter to `TRUE`.

## Using Arrow Keys to Move Between Cells

By default, when the user is in a cell that is in edit mode and presses an arrow key, the cursor moves within the cell and the cell remains in edit mode. If you prefer, pressing an arrow key can exit edit mode and move the active cell in the specified direction.

### ► To use arrow keys to move between cells

Spread  
Designer

In the Spread Designer,

1. From the General menu, choose Spreadsheet Environment or press the Spreadsheet Environment button () in the tool bar.
2. On the General tab in the Sheet Environment Settings dialog box,
  - a. Select the Use Arrow Keys to Exit Edit check box under Settings.
  - b. Choose the OK or Apply button.

ActiveX, VBX Set the **ArrowsExitEditMode** property to `True`.

- DLL Call the **SSSetBool** function, and set the parameters as follows:
1. Set the value of the *nIndex* parameter to `SSB_ARROWSEXITEDITMODE`.
  2. Set the value of the *bNewVal* parameter to `TRUE`.

## Turning Off the Clipboard Shortcut Keys

Users can use Clipboard shortcut keys to cut (Ctrl+X or Shift+Del), copy (Ctrl+C or Ctrl+Ins), and paste (Ctrl+V or Shift+Ins) data in the spreadsheet. If you prefer, you can turn off these Clipboard shortcut keys.

► **To turn off the Clipboard shortcut keys**

Spread  
Designer

In the Spread Designer,

1. From the General menu, choose Spreadsheet Environment or press the Spreadsheet Environment button () in the tool bar.
2. On the General tab in the Sheet Environment Settings dialog box,
  - a. Clear the Handle Clipboard Keys check box under Settings.
  - b. Choose the OK or Apply button.

ActiveX, VBX

Set the **AutoClipboard** property to False.

DLL

Call the **SSSetBool** function, and set the parameters as follows:

1. Set the value of the *nIndex* parameter to `SSB_AUTOCLIPBOARD`.
2. Set the value of the *bNewVal* parameter to `FALSE`.

## Changing or Turning Off the F2, F3, and F4 Action Keys

The F2, F3, and F4 keys perform the following actions in the spreadsheet.

Key	Action	Cell type
F2	Clear	All
F3	Display current date or time	Date, Time
F4	Display the pop-up calendar	Date

You can turn off these keys or you can change the key sequence to perform the action.

---

**Note** The action occurs only when the cell is in edit mode.

---

► **To change or turn off the F2, F3, and F4 action keys**

Spread  
Designer

**Note** Spread provides multiple ways to navigate within the spreadsheet. We recommend that you do not choose a key sequence for these actions that Spread uses for navigation. For a list of navigation keys that Spread uses, see “Navigation” on page 27.

---

In the Spread Designer,

1. From the General menu, choose Spreadsheet Environment or press the Spreadsheet Environment button (  ) in the tool bar.
2. On the Action Keys tab in the Sheet Environment Settings dialog box,
  - a. To customize an action key to clear the active cell, complete the following steps:
    - i. Select Clear Data from the Action drop-down list box.
    - ii. Under Key Used, select an option button for the key.  
If you want to turn off the action key, select the None option button.  
If you select the Custom option button, choose the Capture Key Combination button to capture the appropriate keystrokes.
    - iii. Choose the OK or Apply button.
  - b. To customize an action key to display the current date and time in a date or time cell, complete the following steps:
    - i. Select Display Current Date/Time from the Action drop-down list box.
    - ii. Under Key Used, select an option button for the key.  
If you want to turn off the action key, select the None option button.  
If you select the Custom option button, choose the Capture Key Combination button to capture the appropriate keystrokes.
    - iii. Choose the OK or Apply button.
  - c. To customize an action key to display a pop-up calendar in a date cell, complete the following steps:
    - i. Select Popup Calendar from the Action drop-down list box.
    - ii. Under Key Used, select an option button for the key.  
If you want to turn off the action key, select the None option button.  
If you select the Custom option button, choose the Capture Key Combination button to capture the appropriate keystrokes.
    - iii. Choose the OK or Apply button.
  - d. Choose the OK or Apply button.

ActiveX, VBX

---

**Note** Spread provides multiple ways to navigate within the spreadsheet. We recommend that you do not choose a key sequence for these actions that Spread uses for navigation. For a list of navigation keys that Spread uses, see “Navigation” on page 27.

---

1. To change the F2, F3, or F4 action key, call the **SetActionKey** function or method and set the parameters as follows:
  - a. Set the *wAction* parameter to 0 (Clear), 1 (Current), or 2 (Popup).
  - b. Set the *fShift* parameter to True or False to indicate whether the Shift key is pressed.
  - c. Set the *fCtrl* parameter to True or False to indicate whether the Ctrl key is pressed.
  - d. Set the *wKey* parameter to the ASCII value of the key you want to use.

---

**Note** If you want to use alphabetical keys, you must use uppercase letters for this parameter.

---

2. To turn the F2, F3, or F4 action key off, call the **SetActionKey** function or method and set the parameters as follows:
  - a. Set the *wAction* parameter to 0 (Clear), 1 (Current), or 2 (Popup).
  - b. Set the *wKey* parameter to 0.

DLL

---

**Note** Spread provides multiple ways to navigate within the spreadsheet. We recommend that you do not choose a key sequence for these actions that Spread uses for navigation. For a list of navigation keys that Spread uses, see “Navigation” on page 27.

---

1. To change the F2, F3, or F4 action key, call the **SSSetActionKey** function and set the parameters as follows:
  - a. Set the *wAction* parameter to SS\_KBA\_CLEAR, SS\_KBA\_CURRENT, or SS\_KBA\_POPUP.
  - b. Set the *fShift* parameter to True or False to indicate whether the Shift key is pressed.
  - c. Set the *fCtrl* parameter to True or False to indicate whether the Ctrl key is pressed.

- d. Set the *wKey* parameter to the ASCII value of the key you want to use.

---

**Note** If you want to use alphabetical keys, you must use uppercase letters for this parameter.

---

2. To turn the F2, F3, or F4 action key off, call the **SSSetActionKey** function and set the parameters as follows:
  - a. Set the *wAction* parameter to SS\_KBA\_CLEAR, SS\_KBA\_CURRENT, or SS\_KBA\_POPUP.
  - b. Set the *wKey* parameter to 0.

## Specifying When Buttons Appear

When you use combo boxes, buttons, or spin buttons in cells, you can specify when the spin button (in date, integer, or time cells), button, or drop-down arrow (in combo box cells) displays: always, only in the current cell, only in the current column, only in the current row, always in button cells, or always in combo box cells. By default, all buttons are displayed all the time.

You might want to hide buttons to keep your spreadsheet from looking cluttered. For example, if your spreadsheet has an entire column of combo box cells, you might want to display the drop-down button only in the current cell.

---

**Note** When the **OperationMode** property is set to 2 (Row Mode) or the **SSSetOperationMode** function's *wMode* parameter is set to SS\_OPMODE\_ROWMODE, the button draw setting is ignored. While the row is selected and the highlight bar is visible, no buttons are drawn. While the row is being edited, all the buttons in the current row are drawn.

---

### ► To specify when buttons appear

Spread  
Designer

In the Spread Designer,

1. From the General menu, choose Spreadsheet Environment or press the Spreadsheet Environment button (  ) in the tool bar.
2. On the Button Display tab in the Sheet Environment Settings dialog box, select the appropriate check box under Display Buttons.
3. Choose the OK or Apply button.

## Chapter 14

ActiveX, VBX Set the **ButtonDrawMode** property.

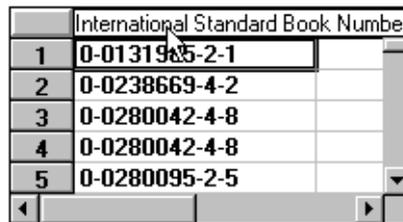
Note that you can combine values 1 (Current Cell), 2 (Current Column), 4 (Current Row), 8 (Always Button), and 16 (Always Combo) using the Or operator to limit where buttons are displayed in the spreadsheet.

DLL Call the **SSSetButtonDrawMode** function and set the *wMode* parameter.

Note that you can combine the values using the Or operator to limit where buttons are displayed in the spreadsheet.

## Providing Text Tips

The fpSpread control provides *text tips* that can be displayed for any header or any cell in the control. Text tips are displayed in ToolTip windows for the header or cell that is under the pointer. As shown in the following figure, when the pointer is over the header in column 1, the control displays the text tip for that header.



The image shows a spreadsheet control with a tooltip displayed over the header of the first column. The tooltip text is 'International Standard Book Number'. The spreadsheet data is as follows:

	International Standard Book Number	
1	0-0131985-2-1	
2	0-0238669-4-2	
3	0-0280042-4-8	
4	0-0280042-4-8	
5	0-0280095-2-5	

The fpSpread control is designed to display default text tips when you specify that you want the control to display text tips. The default text tips display the text of the header or cell for which they appear. The control displays default text tips after a short delay. The appearance of the default text tips is determined by your system settings and some other default settings.

You can customize default text tips to display a different appearance and in a different location. You can also specify alternate text to display in the text tips.

## Displaying Text Tips

Text tips are not displayed automatically for headers and cells. To have the control display text tips, set the **TextTip** property to a value other than 0 (Off) and, if the text is completely displayed in the cell, set the *ShowTip* parameter in the **TextTipFetch** event to True. When you set the **TextTip** property, the setting you choose specifies the following:

- That you want the control to display text tips
- Where the text tips are displayed
- Whether the control only displays text tips when it has the focus

Once you have specified that the control should display text tips, it will display them for any header or cell that contains text greater than the column width. You can specify that text tips appear regardless of length. If you want to display text tips for a header or cell that does not contain text, see “Customizing Text Tips” on page 309.

The control waits 500 milliseconds to display the text tip for a header or cell after the pointer is positioned over the header or cell. You can change that delay interval, if you prefer.

► **To display text tips**

Spread  
Designer

In the Spread Designer,

1. From the Display menu, choose Text Tip or press the Text Tip button (  ) in the tool bar.
2. On the Text Tip tab in the Display dialog box,
  - a. Specify that you want to display text tips, whether you want text tips to display only when the control has the focus, and the location of the text tips by setting the Display drop-down list box to a selection other than Off, as described in the following table:

<b>If you want the text tip to display . . .</b>	<b>Use this setting . . .</b>
Aligned to the top-left corner of the header or cell for which it appears at any time	Fixed
Aligned to the top-left corner of the header or cell for which it appears only when the control has the focus	Fixed Focus Only
Aligned to the pointer location at any time	Floating
Aligned to the pointer location only when the control has the focus	Floating Focus Only

- b. If you want to change the delay interval before the control displays text tips, click the arrows or type a value in the Delay spin box.
  - c. Choose the OK or Apply button.
3. At run time, set the *ShowTip* parameter to True in the **TextTipFetch** event to display text tips for cells where the text is completely displayed.

ActiveX, VBX

1. Specify that you want to display text tips, whether you want text tips to display only when the control has the focus, and the location of the text tips by setting the **TextTip** property to a value other than 0 (Off), as described in the following table:

<b>If you want the text tip to display . . .</b>	<b>Use this setting . . .</b>
Aligned to the top-left corner of the header or cell for which it appears at any time	1 (Fixed)
Aligned to the top-left corner of the header or cell for which it appears only when the control has the focus	3 (Fixed Focus Only)
Aligned to the pointer location at any time	2 (Floating)
Aligned to the pointer location only when the control has the focus	4 (Floating Focus Only)

2. If you want to change the delay interval before the control displays text tips, set the **TextTipDelay** property.
3. Set the *ShowTip* parameter to True in the **TextTipFetch** event to display text tips for cells where the text is completely displayed.

DLL

1. Call the **SSSetTextTip** function, setting the *wStatus* and *lDelay* parameters as follows:
  - a. Specify that you want to display text tips, whether you want text tips to display only when the control has the focus, and the location of the text tips by setting the *wStatus* parameter to a value other than `SS_TT_STATUS_OFF`, as described in the following table:

<b>If you want the text tip to display . . .</b>	<b>Use this setting . . .</b>
Aligned to the top-left corner of the header or cell for which it appears at any time	<code>SS_TT_STATUS_FIXED</code>
Aligned to the top-left corner of the header or cell for which it appears only when the control has the focus	<code>SS_TT_STATUS_FIXEDFOCUSONLY</code>

**If you want the text tip to display . . . Use this setting . . .**

Aligned to the pointer location at any time      `SS_TT_STATUS_FLOATING`

Aligned to the pointer location only when the control has the focus      `SS_TT_STATUS_FLOATINGFOCUSONLY`

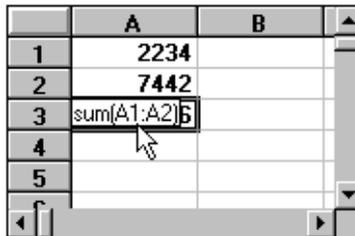
- b. If you want to change the delay interval before the control displays text tips, set the *IDelay* parameter.

2. Set the *fShow* field of the `SS_TEXTTIPFETCH` structure to True.

## Customizing Text Tips

When the control displays the text tip for a header or cell, the **TextTipFetch** event occurs, or for the DLL control, the control sends the `SSM_TEXTTIPFETCH` message. Use the event or message and corresponding structure to customize aspects of the text tip, such as whether text is displayed on multiple lines and the tip width.

You can customize the text in the text tips for the header or cell for which the text tip appears using the **TextTipFetch** event or the `SSM_TEXTTIPFETCH` message and `SS_TEXTTIPFETCH` structure. For example, you might want to have the text tip for each cell display the cell's formula rather than the cell text, as shown in the following figure.



By default, text tips are not displayed for headers or cells that do not contain text or that have text completely displayed. If you want to display a text tip for a header or cell that does not contain text, you can use the **TextTipFetch** event or the `SSM_TEXTTIPFETCH` message and `SS_TEXTTIPFETCH` structure to provide the text and tell the control to display the text tip.

Alternatively, even if you choose for the control to display text tips, you can suppress text tips for specified headers or cells.

► **To customize text tips**

ActiveX, VBX

1. Specify that you want to display text tips using the instructions in “Displaying Text Tips” on page 307.
2. In the **TextTipFetch** event, use the parameters to customize text tips as follows:
  - a. Specify custom text tips for a certain header or cell by responding to the values of the *Col* and *Row* parameters.
  - b. Specify whether text tips display text on multiple lines using the *MultiLine* parameter.
  - c. Specify the width of the text tip in pixels using the *TipWidth* parameter.
  - d. Provide custom text for the text tip using the *TipText* parameter.
  - e. If you want to display a text tip with custom text for the header or cell and the header or cell does not have text in it, set the *ShowTip* parameter to 1.

DLL

1. Specify that you want to display text tips using the instructions in “Displaying Text Tips” on page 307.
2. Use the **SS\_TEXTTIPFETCH** structure’s fields to respond to the **SSM\_TEXTTIPFETCH** message as follows:
  - a. Specify custom text tips for a certain header or cell by responding to the values of the *Col* and *Row* fields.
  - b. Specify whether text tips display text on multiple lines using the *wMultiLine* field.
  - c. Specify the width of the text tip in pixels using the *nWidth* field.
  - d. Provide custom text for the text tip using the *szText* field.
  - e. Specify the horizontal and vertical alignment using the *wAlignX* and *wAlignY* fields.
  - f. If you want to display a text tip with custom text for the header or task and the header or task does not have text in it, set the *fShow* field to 1.

► **To suppress text tips for specific headers or cells**

ActiveX, VBX

In the **TextTipFetch** event, use the parameters as follows:

1. Specify that you want to suppress text tips for a certain header or cell by responding to *Col* and *Row* parameters.

For column headers, set the *Col* parameter to 0. For row headers, set the *Row* parameter to 0.

2. Set the *ShowTip* parameter to 0.

DLL Use the **SS\_TEXTTIPFETCH** structure's fields to respond to the **SSM\_TEXTTIPFETCH** message as follows:

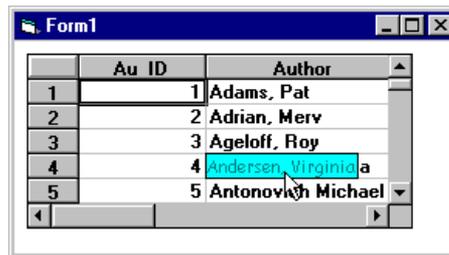
1. Specify that you want to suppress text tips for a certain header or cell by responding to the values of the *Col* and *Row* fields.

For column headers, set the *Col* field to 0. For row headers, set the *Row* field to 0.

2. Set the *fShow* field to 0.

## Specifying the Appearance of Text Tips

Default text tips display text using font and color information from your system settings and using background and text colors provided by the control. You can override these default settings, providing custom font characteristics and specifying other background and text colors. In the following figure, the text tip displays with dark cyan text, a cyan background, and custom font settings.



The settings you specify for the font characteristics and text tips background and text colors are set for all text tips in the control, providing a uniform appearance. If you want to set font characteristics and colors for text tips for specific headers and tasks, set those characteristics in response to the **TextTipFetch** event or the **SSM\_TEXTTIPFETCH** message.

### ► To specify the appearance of text tips

Spread Designer

In the Spread Designer,

1. From the Display menu, choose Text Tip or press the Text Tip button (  ) in the tool bar.
2. On the Text Tip tab in the Display dialog box,
  - a. If you want to override the system settings for font characteristics,
    - i. Choose the Font button.

- ii. In the Font dialog box, specify the font name, style, size, and effects.
- iii. Choose the OK button.
- b. If you want to change the background color for text tips,
  - i. Select BackColor from the drop-down list box under Color.
  - ii. Choose the Set button.
  - iii. Select a basic color or customize your color.
  - iv. Choose the OK button.
- c. If you want to change the text (foreground) color for text tips,
  - i. Select ForeColor from the drop-down list box under Color.
  - ii. Choose the Set button.
  - iii. Select a basic color or customize your color.
  - iv. Choose the OK button.
- d. Choose the OK or Apply button.

ActiveX, VBX Use the **SetTextTipAppearance** function or method and set the parameters as follows:

1. If you want to override the system settings for font characteristics, set the *TipFontName*, *TipFontSize*, *TipFontBold*, and *TipFontItalic* parameters as you prefer.
2. If you want to change the background color for text tips, set the *TipBackColor* parameter.
3. If you want to change the text (foreground) color for text tips, set the *TipForeColor* parameter.

DLL Use the **SSSetTextTip** function, setting the parameters as follows:

1. If you want to override the system settings for font characteristics, set the *lpLogFont* parameter as you prefer.
2. If you want to change the background color for text tips, set the *clrBack* parameter.
3. If you want to change the text (foreground) color for text tips, set the *clrFore* parameter.

# Appendix A fpSpread Control Property Summary Table

The following table provides a summary of all the fpSpread control properties including

- Property name
- Whether ActiveX and VBX controls support the property
- Whether the Spread Designer provides access to the property
- Whether the property sets (writes) or returns (reads) a value
- Whether the property is available at run time or design time
- Short description of property

Property	ActiveX	VBX	SD*	Read	Write	Run Time	Design Time	Description
<b>About</b>	✓	✓	✓	✓			✓	Provides version information
<b>Action</b>	✓	✓	✓	✓	✓	✓		Performs a specified action based on the assigned setting
<b>ActiveCol</b>	✓	✓	✓	✓	✓	✓		Returns the column number of the active cell
<b>ActiveRow</b>	✓	✓	✓	✓	✓	✓		Returns the row number of the active cell
<b>AllowCellOverflow</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether data can overflow into adjacent empty cells
<b>AllowDragDrop</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether the user can drag and drop cells
<b>AllowMultiBlocks</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether the user can select multiple, discontinuous blocks of cells
<b>AllowUserFormulas</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether the user can type formulas in float or integer cells
<b>Appearance</b>	✓	✓	✓	✓	✓	✓	✓	Specifies a predefined border style
<b>ArrowsExitEditMode</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the function of the arrow keys during edit mode
<b>AutoCalc</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether the spreadsheet recalculates each formula when the contents of dependent cells change
<b>AutoClipboard</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether the spreadsheet supports the Clipboard shortcut keys
<b>AutoSize</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether the spreadsheet is automatically sized based on the size of its parent object and on the number of visible columns
<b>BackColor</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the background color of a portion of or the entire spreadsheet

\* Spread Designer

Property	ActiveX	VBX	SD*	Read	Write	Run Time	Design Time	Description
<b>BackColorStyle</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether the background colors of cells overlap the grid lines
<b>BlockMode</b>	✓	✓	✓	✓	✓	✓		Determines whether you can modify a block of cells
<b>BorderStyle</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the style of the control border
<b>ButtonDrawMode</b>	✓	✓	✓	✓	✓	✓	✓	Specifies whether to display buttons in button or combo box cells
<b>CausesValidation</b>	✓			✓	✓	✓	✓	Determines whether the Validate event occurs when the control is losing the focus (Visual Basic 6.0 and later)
<b>CellBorderColor</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the border color of a portion of the spreadsheet
<b>CellBorderStyle</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the border style of a portion of or the entire spreadsheet
<b>CellBorderType</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether to display a border around a portion of or the entire spreadsheet
<b>CellType</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the type of data displayed in a portion of or the entire spreadsheet
<b>ChangeMade</b>	✓	✓	✓	✓	✓	✓	✓	Specifies whether the user has made changes to data in the spreadsheet
<b>Clip</b>	✓	✓	✓	✓	✓	✓	✓	Adds formatted data into or returns formatted data from a column, a row, or a block of cells
<b>ClipboardOptions</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether headers are included when data is copied to or pasted from the Clipboard
<b>ClipValue</b>	✓	✓	✓	✓	✓	✓	✓	Adds unformatted data into or returns unformatted data from a column, a row, or a block of cells

\* Spread Designer

## Appendix A

Property	ActiveX	VBX	SD*	Read	Write	Run Time	Design Time	Description
<b>Col</b>	✓	✓	✓	✓	✓	✓		Specifies a column or the first column of a block of cells on which an operation is to occur
<b>Col2</b>	✓	✓	✓	✓	✓	✓		Specifies the last column of a block of cells on which an operation is to occur
<b>ColHeaderDisplay</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether column headers display letters or numbers or are blank
<b>ColHidden</b>	✓	✓	✓	✓	✓	✓	✓	Specifies a column as hidden
<b>ColPageBreak</b>	✓	✓	✓	✓	✓	✓		Determines whether a page break occurs at the specified column when printing the spreadsheet
<b>ColsFrozen</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the number of nonscrolling columns
<b>ColWidth</b>	✓	✓	✓	✓	✓	✓		Specifies the width of the specified column
<b>Container</b>	✓	✓	✓	✓	✓	✓		Specifies the container of a control on a Form (Visual Basic 4.0 and later)
<b>CursorIcon</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the icon to display as the custom pointer
<b>CursorStyle</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the appearance of the pointer
<b>CursorType</b>	✓	✓	✓	✓	✓	✓		Specifies the area of the spreadsheet in which to display a different pointer
<b>DataBindings</b>	✓	✓	✓	✓	✓	✓		Returns the DataBindings collection object containing the available bindable properties (Visual Basic 4.0 and later)
<b>DataChanged</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether data in a bound fpSpread control has been changed by some process other than retrieving data from the current record
<b>DataColCnt</b>	✓	✓	✓	✓	✓	✓	✓	Returns the last column that contains data in the spreadsheet

\* Spread Designer

Property	ActiveX	VBX	SD*	Read	Write	Run Time	Design Time	Description
<b>DataField</b>	✓	✓	✓	✓	✓	✓		Specifies the database field name assigned to a column in a bound spreadsheet
<b>DataFillEvent</b>	✓	✓	✓	✓	✓	✓		Determines whether the <b>DataFill</b> event occurs for a specified column
<b>DataMember</b>	✓			✓	✓	✓	✓	Specifies the name of the data member to retrieve from the object referenced by the <b>DataSource</b> property (ActiveX ADO control in Visual Basic 6.0 or later)
<b>DataRowCnt</b>	✓	✓		✓		✓		Returns the last row that contains data in the spreadsheet
<b>DataSource</b>	✓	✓		✓	✓	✓	✓	Specifies the name of the data control bound to the fpSpread control
<b>DAutoCellTypes</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether a bound spreadsheet automatically sets the cell type for the cells in a column
<b>DAutoFill</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether a bound spreadsheet automatically reads in the data from the database
<b>DAutoHeadings</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether a bound spreadsheet automatically assigns the text in the column headers to be the data field names
<b>DAutoSave</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether a bound spreadsheet automatically writes modified records to the database
<b>DAutoSizeCols</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether a bound spreadsheet automatically sizes its columns based on the data in the associated database fields
<b>DestCol</b>	✓	✓	✓	✓	✓	✓		Specifies the destination column for an action
<b>DestRow</b>	✓	✓	✓	✓	✓	✓		Specifies the destination row for an action

\* Spread Designer

Property	ActiveX	VBX	SD*	Read	Write	Run Time	Design Time	Description
<b>DInformActiveRowChange</b>	✓	✓	✓	✓	✓	✓	✓	Specifies how a bound fpSpread control communicates with the Data control
<b>DisplayColHeaders</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether the spreadsheet displays column headers
<b>DisplayRowHeaders</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether the spreadsheet displays row headers
<b>DragIcon</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the icon displayed as the pointer when the user drags the fpSpread control
<b>DragMode</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the dragging mode for drag-and-drop operations
<b>EditEnterAction</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the action that occurs when the user presses the Enter key
<b>EditMode</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether a cell is in edit mode
<b>EditModePermanent</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether a cell is in edit mode whenever it becomes the active cell
<b>EditModeReplace</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether the existing text in a cell is selected when the cell enters edit mode
<b>Enabled</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether the fpSpread control is enabled at run time
<b>FileNum</b>		✓		✓	✓	✓		Specifies the file number to use when saving or loading data
<b>FloatDefCurrencyChar</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the default currency symbol for all float cells in the spreadsheet
<b>FloatDefDecimalChar</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the default decimal symbol for all float cells in the spreadsheet
<b>FloatDefSepChar</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the default separator for all float cells in the spreadsheet
<b>Font</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the font characteristics of displayed text

\* Spread Designer

Property	ActiveX	VBX	SD*	Read	Write	Run Time	Design Time	Description
<b>FontBold</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether displayed text is bold
<b>FontItalic</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether displayed text is italicized
<b>FontName</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the font used to display text
<b>FontSize</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the font size of displayed text
<b>FontStrikethru</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether displayed text is struck through
<b>FontUnderline</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether displayed text is underlined
<b>ForeColor</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the text color of a portion of or the entire spreadsheet
<b>Formula</b>	✓	✓	✓	✓	✓	✓	✓	Specifies a formula for a cell
<b>FormulaSync</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether cell references in formulas are updated when columns or rows are inserted or deleted or when blocks of cells are moved or swapped
<b>GrayAreaBackColor</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the gray area color for the spreadsheet
<b>GridColor</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the color of the grid lines
<b>GridShowHoriz</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether to display horizontal grid lines in the spreadsheet
<b>GridShowVert</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether to display vertical grid lines in the spreadsheet
<b>GridSolid</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the type of grid lines displayed in the spreadsheet
<b>hDCPrinter</b>	✓	✓	✓	✓	✓	✓	✓	Provides a device context handle (HDC) for a printer for creating custom printing configurations
<b>Height</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the height of the fpSpread control

\* Spread Designer

## Appendix A

Property	ActiveX	VBX	SD*	Read	Write	Run Time	Design Time	Description
<b>HelpContextID</b>	✓	✓	✓	✓	✓	✓	✓	Provides an associated context number for the fpSpread control
<b>hWnd</b>	✓	✓		✓		✓		Returns the window handle of the fpSpread control
<b>Index</b>	✓	✓	✓	✓	✓	✓	✓	Sets or returns the number that uniquely identifies a control in a control array
<b>IsBlockSelected</b>	✓	✓	✓	✓	✓	✓	✓	Returns whether any blocks of cells are selected
<b>Left</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the distance between the internal left edge of an object and the left edge of its container
<b>LeftCol</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the number of the column to position as the leftmost column in the displayed spreadsheet
<b>Lock</b>	✓	✓	✓	✓	✓	✓	✓	Specifies whether the specified cell or cells are marked as locked
<b>LockBackColor</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the background color of locked cells
<b>LockForeColor</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the text color of locked cells
<b>MaxCols</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the number of columns in the spreadsheet
<b>MaxRows</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the number of rows in the spreadsheet
<b>MaxTextCellHeight</b>	✓	✓	✓	✓	✓	✓	✓	Returns the minimum height necessary to display all the text within the specified cell
<b>MaxTextCellWidth</b>	✓	✓	✓	✓	✓	✓	✓	Returns the minimum width necessary to display all the text within the specified cell
<b>MaxTextColWidth</b>	✓	✓	✓	✓	✓	✓	✓	Returns the width of the widest text string in the specified column

\* Spread Designer

Property	ActiveX	VBX	SD*	Read	Write	Run Time	Design Time	Description
<b>MaxTextRowHeight</b>	✓	✓		✓		✓		Returns the height of the tallest text string in the specified row
<b>MoveActiveOnFocus</b>	✓	✓	✓	✓	✓	✓	✓	Determines the location of the active cell when the user moves the focus to the spreadsheet with the mouse
<b>MultiSelCount</b>	✓	✓		✓		✓		Returns the number of blocks selected when the spreadsheet is in normal operation mode and the <b>AllowMultiBlocks</b> property is set to True
<b>MultiSelIndex</b>	✓	✓		✓	✓	✓		Specifies the coordinates of a selected block of cells when the spreadsheet is in normal operation mode and the <b>AllowMultiBlocks</b> property is set to True
<b>Name</b>	✓	✓		✓	✓	✓	✓	Specifies the name used in code to identify the fpSpread control
<b>NoBeep</b>	✓	✓		✓	✓	✓	✓	Determines whether the spreadsheet sounds warning beeps
<b>NoBorder</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether the spreadsheet displays a border on the bottom and right edges
<b>Object</b>	✓			✓		✓		Returns an object in a control (Visual Basic 4.0 and later)
<b>OLEDropMode</b>	✓			✓	✓	✓	✓	Specifies whether the control can act as an OLE drop object
<b>OperationMode</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the operation mode of the spreadsheet
<b>Parent</b>	✓	✓		✓		✓		Returns the object on which a control is located
<b>Position</b>	✓	✓		✓	✓	✓	✓	Specifies the location at which to position a cell on the screen

\* Spread Designer

## Appendix A

Property	ActiveX	VBX	SD*	Read	Write	Run Time	Design Time	Description
<b>PrintAbortMsg</b>	✓	✓	✓	✓	✓	✓		Determines the information displayed in an abort dialog box during printing
<b>PrintBorder</b>	✓	✓	✓	✓	✓	✓		Specifies whether to print the spreadsheet border
<b>PrintColHeaders</b>	✓	✓	✓	✓	✓	✓		Specifies whether to print the column headers at the top of each page
<b>PrintColor</b>	✓	✓	✓	✓	✓	✓		Specifies whether to print the colors in the spreadsheet as they appear on the screen
<b>PrintFirstPageNumber</b>	✓			✓	✓	✓		Sets or returns the number to use to begin numbering the pages in the spreadsheet
<b>PrintFooter</b>	✓	✓	✓	✓	✓	✓		Determines the text and format of footers on printed pages
<b>PrintGrid</b>	✓	✓	✓	✓	✓	✓		Specifies whether to print the spreadsheet grid lines
<b>PrintHeader</b>	✓	✓	✓	✓	✓	✓		Determines the text and format of headers on printed pages
<b>PrintJobName</b>	✓	✓	✓	✓	✓	✓		Specifies the print job name displayed in the Print Manager when printing the spreadsheet
<b>PrintMarginBottom</b>	✓	✓	✓	✓	✓	✓		Specifies the bottom margin when printing the spreadsheet
<b>PrintMarginLeft</b>	✓	✓	✓	✓	✓	✓		Specifies the left margin when printing the spreadsheet
<b>PrintMarginRight</b>	✓	✓	✓	✓	✓	✓		Specifies the right margin when printing the spreadsheet
<b>PrintMarginTop</b>	✓	✓	✓	✓	✓	✓		Specifies the top margin when printing the spreadsheet
<b>PrintNextPageBreakCol</b>	✓			✓		✓		Returns the column number of the next column in the spreadsheet at which a page break occurs

\* Spread Designer

Property	ActiveX	VBX	SD*	Read	Write	Run Time	Design Time	Description
<b>PrintNextPageBreakRow</b>	✓			✓		✓		Returns the row number of the next row in the spreadsheet at which a page break occurs
<b>PrintOrientation</b>	✓	✓	✓	✓	✓	✓		Specifies the page orientation used to print the spreadsheet
<b>PrintPageCount</b>	✓			✓		✓		Returns the number of pages required for printing the spreadsheet with the current print settings
<b>PrintPageEnd</b>	✓	✓	✓	✓	✓	✓		Specifies the last page of the spreadsheet to print
<b>PrintPageOrder</b>	✓			✓	✓	✓		Specifies the order in which pages are printed
<b>PrintPageStart</b>	✓	✓	✓	✓	✓	✓		Specifies the first page of the spreadsheet to print
<b>PrintRowHeaders</b>	✓	✓	✓	✓	✓	✓		Determines whether to print the row headers on the left side of each page
<b>PrintShadows</b>	✓	✓	✓	✓	✓	✓		Determines whether to print the shadow effect within the column and row headers
<b>PrintSmartPrint</b>	✓			✓	✓	✓		Determines whether the spreadsheet prints using its SmartPrint capabilities
<b>PrintType</b>	✓	✓	✓	✓	✓	✓		Specifies the type of print range used to print the spreadsheet
<b>PrintUseDataMax</b>	✓	✓	✓	✓	✓	✓		Determines whether the spreadsheet prints empty cells beyond the last column and row containing data
<b>ProcessTab</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether pressing the Tab key moves the active cell to the next cell in the spreadsheet or moves the focus to the next control on the dialog
<b>Protect</b>	✓	✓	✓	✓	✓	✓	✓	Specifies whether users can edit cells marked as locked with the <b>Lock</b> property

\* Spread Designer

Property	ActiveX	VBX	SD*	Read	Write	Run Time	Design Time	Description
<b>ReDraw</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether the application immediately redraws the fpSpread control after the user makes changes
<b>RestrictCols</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether users can type data in a column that is more than one column beyond the last column containing data
<b>RestrictRows</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether users can type data in a row that is more than one row beyond the last row containing data
<b>RetainSelBlock</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether the selected block of cells remains highlighted when the spreadsheet loses the focus
<b>Row</b>	✓	✓	✓	✓	✓	✓	✓	Specifies a specific row or specifies the first row of a block of cells on which an operation is to occur
<b>Row2</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the last row of a block of cells on which an operation is to occur
<b>RowHeaderDisplay</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether row headers display letters or numbers or are blank
<b>RowHeight</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the height of the specified row
<b>RowHidden</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether a row is hidden
<b>RowPageBreak</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether a page break occurs at the specified row when printing the spreadsheet
<b>RowsFrozen</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the number of nonscrolling rows
<b>ScrollBarExtMode</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether to display scroll bars at all times or only when needed
<b>ScrollBarMaxAlign</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the alignment of the last row and column of the spreadsheet in the control

\* Spread Designer

Property	ActiveX	VBX	SD*	Read	Write	Run Time	Design Time	Description
<b>ScrollBars</b>	✓	✓	✓	✓	✓	✓	✓	Specifies whether and how the control displays scroll bars
<b>ScrollBarShowMax</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether the user can use the scroll box to scroll through all the columns and rows in the spreadsheet, or just a limited number of them
<b>ScrollBarTrack</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether the spreadsheet scrolls when the user drags the scroll box
<b>SelBackColor</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the background color of the selected cells
<b>SelBlockCol</b>	✓	✓	✓	✓	✓	✓	✓	Returns the left column of a selected block of cells
<b>SelBlockCol2</b>	✓	✓	✓	✓	✓	✓	✓	Returns the right column of a selected block of cells
<b>SelBlockRow</b>	✓	✓	✓	✓	✓	✓	✓	Returns the top row of a selected block of cells
<b>SelBlockRow2</b>	✓	✓	✓	✓	✓	✓	✓	Returns the bottom row of a selected block of cells
<b>SelectBlockOptions</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether the user can select columns, rows, blocks, all cells, or any combination of these
<b>SelForeColor</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the text color of the selected cells
<b>SelLength</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the length of selected data within the active cell when the cell is in edit mode
<b>SelModeIndex</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the number of the selected row when the spreadsheet is in single-selection operation mode
<b>SelModeSelCount</b>	✓	✓	✓	✓	✓	✓	✓	Returns the number of selected rows when the spreadsheet is in extended- or multiple-selection operation mode

\* Spread Designer

Property	ActiveX	VBX	SD*	Read	Write	Run Time	Design Time	Description
<b>SelectionMode</b>	✓	✓	✓	✓	✓	✓		Returns the selection state of a row when the spreadsheet is in extended- or multiple-selection operation mode
<b>SelStart</b>	✓	✓	✓	✓	✓	✓		Specifies the starting location of selected text within the active cell
<b>SelText</b>	✓	✓	✓	✓	✓	✓		Specifies the selected text within the active cell
<b>ShadowColor</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the background color of the column and row headers, and static text cells when the <b>TypeTextShadow</b> property or the <b>TypeTextShadowIn</b> property is set to True
<b>ShadowDark</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the color of the bottom and right borders of the column and row headers, and static text cells when the <b>TypeTextShadow</b> property is set to True
<b>ShadowText</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the text color used for the column and row headers, and static text cells when the <b>TypeTextShadow</b> property or <b>TypeTextShadowIn</b> property is set to True
<b>SortBy</b>	✓	✓	✓	✓	✓	✓		Determines whether to sort columns or rows
<b>SortKey</b>	✓	✓	✓	✓	✓	✓		Specifies the column or row to sort by
<b>SortKeyOrder</b>	✓	✓	✓	✓	✓	✓		Specifies for the column or row by which you are sorting whether to sort in ascending or descending order
<b>StartingColNumber</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the number displayed in the first column
<b>StartingRowNumber</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the number displayed in the first row

\* Spread Designer

Property	ActiveX	VBX	SD*	Read	Write	Run Time	Design Time	Description
<b>TabIndex</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the tab order of a control within its parent form
<b>TabStop</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether users can use the Tab key to move the focus to a control
<b>Tag</b>	✓	✓	✓	✓	✓	✓	✓	Specifies extra data for the application
<b>Text</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the contents of the specified cell
<b>TextTip</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether the text tips are displayed at all times, only when the control has the focus, or not at all, and when the tips are displayed and how they are positioned
<b>TextTipDelay</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the number of milliseconds the control waits before displaying the text tip for the cell under the pointer
<b>ToolTipText</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the text that appears as a ToolTip for a control (Visual Basic 4.0 and later)
<b>Top</b>	✓	✓	✓	✓	✓	✓	✓	Determines the distance between the internal top edge of a control and the top edge of its container
<b>TopRow</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the row that is currently displayed as the top scrollable row on the screen
<b>TwoDigitYearMax</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the year value used to calculate the century for two-digit year values
<b>TypeButtonAlign</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the alignment of the picture and text within a button cell
<b>TypeButtonBorderColor</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the border color of a button cell
<b>TypeButtonColor</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the background color of a button cell
<b>TypeButtonDarkColor</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the color of the shadow around a button cell

\* Spread Designer

Property	ActiveX	VBX	SD*	Read	Write	Run Time	Design Time	Description
TypeButtonLightColor	✓	✓	✓	✓	✓	✓		Specifies the color of the highlight around a button cell
TypeButtonPicture	✓	✓	✓	✓	✓	✓		Specifies the picture displayed when a button is up
TypeButtonPictureDown	✓	✓	✓	✓	✓	✓		Specifies the picture displayed when a button is down
TypeButtonShadowSize	✓	✓	✓	✓	✓	✓		Specifies the size of the shadow and highlight of a button cell
TypeButtonText	✓	✓	✓	✓	✓	✓		Specifies the text to display in a button cell
TypeButtonTextColor	✓	✓	✓	✓	✓	✓		Specifies the color of the text in a button cell
TypeButtonType	✓	✓	✓	✓	✓	✓		Determines whether the button cell behaves as a one-state or two-state button
TypeCheckCenter	✓	✓	✓	✓	✓	✓		Specifies whether the check box is centered in a check box cell
TypeCheckPicture	✓	✓	✓	✓	✓	✓		Specifies the picture used for each state of a check box
TypeCheckText	✓	✓	✓	✓	✓	✓		Specifies the text in a check box cell
TypeCheckTextAlign	✓	✓	✓	✓	✓	✓		Specifies the alignment of the picture and text in a check box cell
TypeCheckType	✓	✓	✓	✓	✓	✓		Specifies whether the check box is a two-state or three-state check box
TypeComboBoxCount	✓	✓	✓	✓	✓	✓		Returns the number of items in the combo box list
TypeComboBoxCurSel	✓	✓	✓	✓	✓	✓		Specifies the number of the item selected in the combo box list
TypeComboBoxEditable	✓	✓	✓	✓	✓	✓		Specifies whether users can edit the text in a combo box cell
TypeComboBoxHWnd	✓	✓	✓	✓	✓	✓		Specifies the window handle of the control to use as the combo box in a combo box cell

\* Spread Designer

Property	ActiveX	VBX	SD*	Read	Write	Run Time	Design Time	Description
TypeComboBoxIndex	✓	✓	✓	✓	✓	✓		Specifies the combo box item on which to perform an operation
TypeComboBoxList	✓	✓	✓	✓	✓	✓		Specifies the list of items displayed in a combo box cell
TypeComboBoxMaxDrop	✓	✓	✓	✓	✓	✓		Specifies the number of rows to display in the drop-down list of a combo box cell
TypeComboBoxString	✓	✓	✓	✓	✓	✓		Specifies a string in a combo box list
TypeComboBoxWidth	✓	✓	✓	✓	✓	✓		Specifies the width of the drop-down list portion of a combo box cell
TypeDateCentury	✓	✓	✓	✓	✓	✓		Determines whether to display the year as a four-digit value in a date cell
TypeDateFormat	✓	✓	✓	✓	✓	✓		Specifies the format used to display the date in a date cell
TypeDateMax	✓	✓	✓	✓	✓	✓		Specifies the maximum date value allowed in a date cell
TypeDateMin	✓	✓	✓	✓	✓	✓		Specifies the minimum date value allowed in a date cell
TypeDateSeparator	✓	✓	✓	✓	✓	✓		Specifies the separator used between the month, day, and year in a date cell
TypeEditCharCase	✓	✓	✓	✓	✓	✓		Specifies the case of the characters typed into an edit cell
TypeEditCharSet	✓	✓	✓	✓	✓	✓		Specifies the set of characters users are allowed to type in an edit cell
TypeEditMultiLine	✓	✓	✓	✓	✓	✓		Determines whether data is displayed on multiple lines in an edit cell
TypeEditPassword	✓	✓	✓	✓	✓	✓		Determines whether an edit cell displays asterisks instead of the characters typed by users

\* Spread Designer

Property	ActiveX	VBX	SD*	Read	Write	Run Time	Design Time	Description
TypeFloatCurrencyChar	✓	✓	✓	✓	✓	✓		Specifies the currency symbol displayed in a float cell
TypeFloatDecimalChar	✓	✓	✓	✓	✓	✓		Specifies the decimal point symbol displayed in a float cell
TypeFloatDecimalPlaces	✓	✓	✓	✓	✓	✓		Specifies the number of digits allowed to the right of the decimal point in a float cell
TypeFloatMax	✓	✓	✓	✓	✓	✓		Specifies the maximum value allowed in a float cell
TypeFloatMin	✓	✓	✓	✓	✓	✓		Specifies the minimum value allowed in a float cell
TypeFloatMoney	✓	✓	✓	✓	✓	✓		Determines whether the spreadsheet displays a currency symbol to the left of the value in a float cell
TypeFloatSeparator	✓	✓	✓	✓	✓	✓		Determines whether separators are displayed in a float cell
TypeFloatSepChar	✓	✓	✓	✓	✓	✓		Specifies the character used to separate groups of thousands digits in a float cell
TypeHAlign	✓	✓	✓	✓	✓	✓		Specifies the horizontal alignment in a cell
TypeIntegerMax	✓	✓	✓	✓	✓	✓		Specifies the maximum value allowed in an integer cell
TypeIntegerMin	✓	✓	✓	✓	✓	✓		Specifies the minimum value allowed in an integer cell
TypeIntegerSpinInc	✓	✓	✓	✓	✓	✓		Specifies the increment used in an integer cell when users press the spin button
TypeIntegerSpinWrap	✓	✓	✓	✓	✓	✓		Determines whether the value of the spin button in an integer cell wraps when the maximum or minimum values are reached
TypeMaxEditLen	✓	✓	✓	✓	✓	✓		Specifies the maximum number of characters users can enter in an edit cell or in the edit field of a combo box cell

\* Spread Designer

Property	ActiveX	VBX	SD*	Read	Write	Run Time	Design Time	Description
TypeOwnerDrawStyle	✓	✓	✓	✓	✓	✓		Determines the style assigned to an owner-drawn cell
TypePicDefaultText	✓	✓		✓	✓	✓		Specifies the default text displayed in a PIC cell
TypePicMask	✓	✓	✓	✓	✓	✓		Defines the mask used in a PIC cell
TypePicCenter	✓	✓		✓	✓	✓		Determines whether the picture in a picture cell is centered
TypePicMaintainScale	✓	✓	✓	✓	✓	✓		Determines whether the application maintains the proportions of a picture in a picture cell when it sizes the picture
TypePicPicture	✓	✓	✓	✓	✓	✓		Specifies the picture displayed in a picture cell
TypePicStretch	✓	✓	✓	✓	✓	✓		Determines whether a picture in a picture cell is sized to fit the cell
TypeSpin	✓	✓	✓	✓	✓	✓		Determines whether a spin button is displayed in a date, time, or integer cell
TypeTextPrefix	✓	✓	✓	✓	✓	✓		Determines whether an ampersand in text in a static text cell underlines the next character
TypeTextShadow	✓	✓	✓	✓	✓	✓		Determines whether a static text cell is displayed with a raised three-dimensional appearance
TypeTextShadowIn	✓	✓	✓	✓	✓	✓		Determines whether a static text cell is displayed with a lowered three-dimensional appearance
TypeTextWordWrap	✓	✓	✓	✓	✓	✓		Determines whether text in static text cells is displayed on multiple lines
TypeTime24Hour	✓	✓	✓	✓	✓	✓		Determines whether time is displayed in 12- or 24-hour format in a time cell
TypeTimeMax	✓	✓	✓	✓	✓	✓		Specifies the maximum time value allowed in a time cell

\* Spread Designer

Property	ActiveX	VBX	SD*	Read	Write	Run Time	Design Time	Description
TypeTimeMin	✓	✓	✓	✓	✓	✓		Specifies the minimum time value allowed in a time cell
TypeTimeSeconds	✓	✓	✓	✓	✓	✓		Determines whether seconds are displayed in a time cell
TypeTimeSeparator	✓	✓	✓	✓	✓	✓		Specifies the character used to separate the hours, minutes, and seconds in a time cell
TypeVAlign	✓	✓	✓	✓	✓	✓		Specifies the vertical alignment in a cell
UnitType	✓	✓	✓	✓	✓	✓	✓	Specifies the units used for the column width and row height
UserResize	✓	✓	✓	✓	✓	✓	✓	Determines whether the user can resize rows, columns, or both
UserResizeCol	✓	✓	✓	✓	✓	✓	✓	Determines whether users can resize individual columns
UserResizeRow	✓	✓	✓	✓	✓	✓	✓	Determines whether users can resize individual rows
Value	✓	✓	✓	✓	✓	✓		Sets or returns unformatted data
VirtualCurRowCount	✓	✓	✓	✓	✓	✓		Returns the number of rows in the virtual buffer
VirtualCurTop	✓	✓	✓	✓	✓	✓		Returns the current top row of the virtual buffer
VirtualMaxRows	✓	✓	✓	✓	✓	✓	✓	Specifies the maximum number of rows in the spreadsheet when virtual mode is on
VirtualMode	✓	✓	✓	✓	✓	✓	✓	Determines whether virtual mode is on or off
VirtualOverlap	✓	✓	✓	✓	✓	✓	✓	Specifies the number of retrieved rows to keep in the virtual buffer when virtual mode is on
VirtualRows	✓	✓	✓	✓	✓	✓	✓	Specifies the number of virtual rows the spreadsheet requests at a time

\* Spread Designer

Property	ActiveX	VBX	SD*	Read	Write	Run Time	Design Time	Description
<b>VirtualScrollBuffer</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether the scroll box reflects the number of rows in the virtual buffer rather than the maximum number of rows
<b>Visible</b>	✓	✓	✓	✓	✓	✓	✓	Specifies whether the fpSpread control is visible
<b>VisibleCols</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the number of columns that are fully displayed
<b>VisibleRows</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the number of rows that are fully displayed
<b>VScrollSpecial</b>	✓	✓	✓	✓	✓	✓	✓	Determines whether the spreadsheet displays a special vertical scroll bar
<b>VScrollSpecialType</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the scroll arrows to display on the special vertical scroll bar
<b>WhatsThisHelpID</b>	✓	✓	✓	✓	✓	✓	✓	Specifies an associated context number for the control (Visual Basic 4.0 and later)
<b>Width</b>	✓	✓	✓	✓	✓	✓	✓	Specifies the width of the fpSpread control

\* Spread Designer



# Appendix B fpSpreadPreview Control Property Summary Table

The following table provides a summary of all the fpSpreadPreview control properties including

- Property name
- Whether ActiveX and VBX controls support the property
- Whether the property sets (writes) or returns (reads) a value
- Whether the property is available at run time or design time
- Short description of property

## Appendix B

Property	ActiveX	VBX	Read	Write	Run Time	Design Time	Description
<b>AllowUserZoom</b>	✓	✓	✓	✓	✓	✓	Determines whether the user can use the mouse to zoom in or out while viewing pages
<b>GrayAreaColor</b>	✓	✓	✓	✓	✓	✓	Specifies the color of the preview area of the control
<b>GrayAreaMarginH</b>	✓	✓	✓	✓	✓	✓	Specifies the horizontal space between the displayed page and the edge of the control
<b>GrayAreaMarginType</b>	✓	✓	✓	✓	✓	✓	Determines whether the control uses the actual gray area margin settings or scales the settings to optimize the page view
<b>GrayAreaMarginV</b>	✓	✓	✓	✓	✓	✓	Specifies the vertical space between the displayed page and the edge of the control
<b>hWndSpread</b>	✓	✓	✓	✓	✓	✓	Specifies the window handle of the spreadsheet you want to preview
<b>PageBorderColor</b>	✓	✓	✓	✓	✓	✓	Specifies the border color of the displayed pages
<b>PageBorderWidth</b>	✓	✓	✓	✓	✓	✓	Specifies the border width of the displayed pages
<b>PageCurrent</b>	✓	✓	✓	✓	✓	✓	Specifies the page to display in the control
<b>PageGutterH</b>	✓	✓	✓	✓	✓	✓	Specifies the horizontal space in twips between displayed pages
<b>PageGutterV</b>	✓	✓	✓	✓	✓	✓	Specifies the vertical space in twips between displayed pages
<b>PageMultiCntH</b>	✓	✓	✓	✓	✓	✓	Specifies the number of pages displayed in the horizontal direction
<b>PageMultiCntV</b>	✓	✓	✓	✓	✓	✓	Specifies the number of pages displayed in the vertical direction
<b>PagePercentageActual</b>	✓	✓	✓	✓	✓	✓	Returns the size of the displayed pages as a percentage of the actual size
<b>PageShadowColor</b>	✓	✓	✓	✓	✓	✓	Specifies the color of the shadow behind the displayed pages
<b>PageShadowWidth</b>	✓	✓	✓	✓	✓	✓	Specifies the width of the shadow behind the displayed pages

Property	ActiveX	VBX	Read	Write	Run Time	Design Time	Description
<b>PagesPerScreen</b>	✓	✓	✓	✓	✓	✓	Specifies the number of pages displayed at one time
<b>PageViewPercentage</b>	✓	✓	✓	✓	✓	✓	Specifies the percent size of the displayed pages
<b>PageViewType</b>	✓	✓	✓	✓	✓	✓	Determines how pages are displayed
<b>ScrollBarH</b>	✓	✓	✓	✓	✓	✓	Determines whether and how the horizontal scroll bars are displayed
<b>ScrollBarV</b>	✓	✓	✓	✓	✓	✓	Determines whether and how the vertical scroll bars are displayed
<b>ScrollIncH</b>	✓	✓	✓	✓	✓	✓	Specifies the horizontal distance the control scrolls when the user scrolls through the displayed pages
<b>ScrollIncV</b>	✓	✓	✓	✓	✓	✓	Specifies the vertical distance the control scrolls when the user scrolls through the displayed pages
<b>ZoomState</b>	✓	✓	✓	✓	✓	✓	Specifies the way the control zooms between different views of the displayed pages



# Appendix C fpSpread and fpSpreadPreview DLL Function Summary

## fpSpread Control DLL Function Summary

The following table provides a complete list of the fpSpread DLL functions and a description of each.

<b>DLL Function</b>	<b>Description</b>
SSAddCustomFunction	Adds a custom function to the fpSpread control's equation parser
SSAddCustomFunctionExt	Adds a custom function with a variable number of parameters to the fpSpread control's equation parser
SSAddMultiSelBlocks	Lets the user select multiple blocks of cells
SSClear	Deletes the data and clears all the properties of a cell, a column, a row, or the entire spreadsheet
SSClearData	Deletes the data from the specified cell
SSClearDataRange	Deletes the data from the specified block of cells
SSClearRange	Deletes the data and clears all the properties of a block of cells
SSClipboardCopy	Copies the selected block of cells to the Clipboard
SSClipboardCut	Cuts the selected block of cells to the Clipboard
SSClipboardPaste	Pastes the Clipboard contents into the active cell or into a block of cells
SSClipIn	Loads formatted data into a column, a row, or a block of cells
SSClipOut	Copies formatted data from a column, a row, or a block of cells into a tab-delimited buffer
SSClipValueIn	Loads unformatted data into a column, a row, or a block of cells
SSClipValueOut	Copies unformatted data from a column, a row, or a block of cells into a tab-delimited buffer
SSColWidthToLogUnits	Converts column-width measurement to screen measurement
SSComboBoxSendMessage	Specifies the coordinates of the combo box cell that is to receive a message from the application
SSCopyRange	Copies a block of cells to a new location
SSDelCol	Deletes the specified column

## Appendix C

<b>DLL Function</b>	<b>Description</b>
SSDelColRange	Deletes the specified range of columns
SSDelRow	Deletes the specified row
SSDelRowRange	Deletes the specified range of rows
SSDeselectBlock	Deselects all selected cells
SSEnumCustomFunction	Returns the next custom function
SSEnumCustomFunctionLen	Returns the length of the custom function
SSExportRangeToHTML	Exports the specified range of the spreadsheet to an HTML file
SSExportToExcel	Exports the spreadsheet to a sheet in an Excel-formatted file
SSExportToHTML	Exports the portion of the spreadsheet that contains data to an HTML file
SSFloatFormat	Formats the specified floating-point value using the specified cell type
SSGetActionKey	Retrieves the shortcut key or navigation key assigned to a specific action
SSGetActiveCell	Returns the row and column numbers of the active cell
SSGetAppearance	Retrieves the predefined border style
SSGetArray	Retrieves an array containing data
SSGetAutoSizeVisible	Returns the number of visible columns and rows in the spreadsheet
SSGetBackColorStyle	Determines whether the background colors of cells overlap the grid lines
SSGetBool	Returns the value of one of the spreadsheet's booleans
SSGetBorder	Returns the position of the border, the type of border, and the color of the border for a specified cell
SSGetBorderEx	Returns the cell border settings for each side of the specified cell
SSGetBottomRightCell	Returns the row and column of the lower-right corner cell in the displayed portion of the spreadsheet
SSGetButtonDrawMode	Returns whether buttons in button and combo box cells are displayed
SSGetButtonText	Returns the text in a button cell
SSGetCallBack	Returns the optional callback function to use to receive all fpSpread control messages
SSGetCellFromPixel	Determines the column and row of a cell by using window coordinates
SSGetCellRect	Returns the rectangle coordinates of a cell relative to the upper-left corner of the spreadsheet

## fpSpread and fpSpreadPreview DLL Function Summary

DLL Function	Description
SSGetCellSendingMsg	Returns the coordinates of the cell that sent either the SSN_BUTTONDOWN or SSN_BUTTONUP notification message
SSGetCellType	Returns the type of data displayed in a cell, a column, a row, a block of cells, or the entire spreadsheet
SSGetClientRect	Returns the rectangle coordinates of the client area of the fpSpread control, excluding scroll bars
SSGetClipboardOptions	Retrieves whether row headers are included when data is copied to or pasted from the Clipboard
SSGetColHeaderDisplay	Returns whether the column header displays letters or numbers or is blank
SSGetColor	Returns the foreground and background colors of a cell, a column, a row, or the entire spreadsheet
SSGetColPageBreak	Returns whether a page break occurs at the specified column when printing the spreadsheet
SSGetColUserData	Returns a four-byte user-defined data item for the specified column
SSGetColWidth	Returns the width of the specified column
SSGetColWidthInPixels	Returns the width of the specified column in pixels
SSGetCurrSelBlockPos	Returns the position of the last cell containing the mouse pointer in a selected block
SSGetCursor	Returns the pointer the fpSpread control displays in a variety of situations
SSGetCustomFunction	Retrieves information on custom functions
SSGetCustomName	Returns a custom name and its value
SSGetCustomNameLen	Returns the length of the value of the specified custom name
SSGetData	Returns data in a cell, a column, a row, or the entire spreadsheet
SSGetDataCnt	Returns the column and row numbers of the last column and row that contain data
SSGetDataLen	Returns the length (number of characters) of the data in a cell, a column, a row, or the entire spreadsheet
SSGetDefFloatFormat	Returns the default float characters used by all float cells in the spreadsheet
SSGetEditEnterAction	Returns the action performed when the user presses the Enter key
SSGetEditMode	Returns whether a cell is in edit mode

## Appendix C

<b>DLL Function</b>	<b>Description</b>
SSGetExcelSheetList	Returns a handle to the workbook and a list of the sheets in the specified Excel-formatted file
SSGetFirstValidCell	Returns the coordinates of the first valid cell in the spreadsheet
SSGetFloat	Returns the floating-point value for a float cell
SSGetFont	Returns the font used in a cell, a column, a row, or the entire spreadsheet
SSGetFormula	Returns the formula of a cell, a column, a row, or the entire spreadsheet
SSGetFormulaLen	Returns the length (number of characters) of a formula in a cell, a column, a row, or the entire spreadsheet
SSGetFreeze	Returns the number of frozen (nonscrolling) rows or columns
SSGetGrayAreaColor	Returns the color of the gray area background color of the spreadsheet
SSGetGridColor	Returns color of the spreadsheet grid lines
SSGetGridType	Returns whether horizontal and vertical grid lines are displayed, and the style of the grid lines displayed
SSGetInteger	Returns the integer value in an integer cell
SSGetIteration	Returns the number of iterations and the change between iterations for circular references
SSGetLastValidCell	Returns the coordinates of the last valid cell in the spreadsheet
SSGetLock	Returns whether the user can type data into a cell, a row, or a column
SSGetLockColor	Returns the background and foreground colors displayed in locked cells
SSGetMaxCols	Returns the maximum number of usable columns in the spreadsheet
SSGetMaxRows	Returns the maximum number of usable rows in the spreadsheet
SSGetMaxTextCellSize	Returns the minimum height and width of the specified cell necessary to display all the text within the cell
SSGetMaxTextColWidth	Returns the width of the widest text string in the specified column
SSGetMaxTextRowHeight	Returns the height of the tallest text string in the specified row
SSGetMultiSelBlocks	Returns the selected blocks in the spreadsheet

## fpSpread and fpSpreadPreview DLL Function Summary

DLL Function	Description
SSGetNextPageBreakCol	Returns the column number of the next column marked as a page break
SSGetNextPageBreakRow	Returns the row number of the next row marked as a page break
SSGetOddEvenRowColor	Retrieves the background and text colors for odd and even rows
SSGetOperationMode	Returns the operation mode of the spreadsheet
SSGetOwner	Returns the owner window handle
SSGetPrintOptions	Retrieves the print options settings for the spreadsheet
SSGetPrintPageCount	Returns the number of pages required for printing the spreadsheet with the current print settings
SSGetRefStyle	Returns the reference style used by the spreadsheet to represent formulas and custom names
SSGetRowHeaderDisplay	Returns whether the row header displays letters, numbers, or is blank
SSGetRowHeight	Returns the height of the specified row
SSGetRowHeightInPixels	Returns the height of a specified row in pixels
SSGetRowPageBreak	Returns whether a page break occurs at the specified row when printing the spreadsheet
SSGetRowUserData	Returns a four-byte, user-defined data item for the specified row
SSGetSel	Returns the starting position and length of text that can be selected within the active cell when the cell is in edit mode
SSGetSelBlockOptions	Returns whether the user can select columns, rows, blocks of cells, all cells, or any combination of these
SSGetSelColor	Retrieves the background and text color of the selected row or rows
SSGetSelectBlock	Returns the upper-left and lower-right cell positions of a block selection
SSGetSelText	Returns a handle to the selected text
SSGetShadowColor	Returns the shadow colors displayed in the fpSpread control
SSGetStartingNumbers	Returns the starting numbers for the column and row headers
SSGetTextTip	Retrieves whether to display text tips and the location and the appearance of the tips
SSGetTopLeftCell	Returns the coordinates of the upper-left cell displayed in the scrollable section of the fpSpread control

## Appendix C

DLL Function	Description
SSGetTopLeftPrev	Returns the previous top row and left column after the user scrolls the spreadsheet
SSGetTwoDigitYearMax	Retrieves the year value used to calculate the century when year values are given as two-digit values
SSGetUnitType	Returns the unit type used to specify column width and row height
SSGetUserData	Returns a user-defined, four-byte data value for the fpSpread control to be used by the application
SSGetUserResize	Returns whether the user can resize columns, rows, both, or neither
SSGetUserResizeCol	Returns the column resize option for the specified column
SSGetUserResizeRow	Returns the row resize option for the specified row
SSGetValue	Returns unformatted data in a cell
SSGetValueLen	Returns the length of unformatted text in a cell
SSGetVisible	Returns whether a specified cell, row, or column is visible
SSImportExcelSheet	Imports the specified sheet from an Excel-formatted file
SSInsCol	Inserts a column into the spreadsheet before the specified column
SSInsColRange	Inserts multiple columns into the spreadsheet before the specified column
SSInsRow	Inserts a row into the spreadsheet before the specified row
SSInsRowRange	Inserts multiple rows into the spreadsheet before the specified row
SSIsCellInSelection	Determines whether the specified cell is within the selected block or blocks of cells
SSIsColHidden	Retrieves whether a specific column is hidden
SSIsExcelFile	Returns whether the specified file is an Excel-formatted file in Excel 97 (BIFF8) format
SSIsRowHidden	Retrieves whether a specific row is hidden
SSLoadFromBuffer	Loads a previously saved spreadsheet from a buffer
SSLoadFromFile	Loads a binary data file
SSLoadTabFile	Loads a tab-delimited file
SSLogUnitsToColWidth	Converts screen coordinates to column width coordinates
SSLogUnitsToRowHeight	Converts screen coordinates to row height coordinates
SSMoveRange	Moves the specified block of cells to another location in the spreadsheet
SSPrint	Prints the spreadsheet
SSQueryCustomName	Returns a defined custom name

## fpSpread and fpSpreadPreview DLL Function Summary

DLL Function	Description
SSQueryCustomNameLen	Returns the length of the next custom name
SSRecalc	Recalculates every spreadsheet formula
SSReCalcCell	Recalculates the formula in an individual cell
SSRemoveCustomFunction	Removes the specified custom function
SSReplaceSel	Replaces the selected text with the specified text
SSReset	Clears and resets all spreadsheet attributes to the default settings
SSRowHeightToLogUnits	Converts row height coordinates to screen coordinates for a specified row
SSSaveTabFile	Saves the spreadsheet data in a tab-delimited file
SSSaveToBuffer	Saves the spreadsheet to a buffer
SSSaveToFile	Saves the spreadsheet data only, or the data with formatting, to a binary data file
SSSelModeSendMessage	Sets or returns selection information for the spreadsheet
SSSetActionKey	Specifies the shortcut key or navigation key assigned to a specified action
SSSetActiveCell	Sets the row and column numbers of the active cell
SSSetAppearance	Specifies a predefined border style
SSSetArray	Sets an array containing data
SSSetAutoSizeVisible	Sets the number of visible columns and rows in the spreadsheet
SSSetBackColorStyle	Determines whether the background colors of cells overlap the grid lines
SSSetBool	Sets the value of one of the spreadsheet's booleans
SSSetBorder	Sets the position of the border, the type of border, and the color of the border for a specified cell
SSSetBorderRange	Sets the position of the border, the type of border, and the color of the border for a block of cells
SSSetButtonDrawMode	Sets whether to display buttons in button and combo box cells
SSSetButtonText	Sets the text in a button cell
SSSetCallBack	Sets the optional callback function to use to receive all fpSpread control messages
SSSetCalText	Specifies the text for the popup calendar in a date cell
SSSetCellType	Sets the type of data displayed in a cell, a column, a row, or the entire spreadsheet
SSSetCellTypeRange	Sets the type of data displayed in a block of cells

## Appendix C

<b>DLL Function</b>	<b>Description</b>
SSSetClipboardOptions	Specifies whether row headers are included when data is copied to or pasted from the Clipboard
SSSetColHeaderDisplay	Sets whether the column header displays letters or numbers or is blank
SSSetColor	Sets the foreground and background colors of a cell, a column, a row, or the entire spreadsheet
SSSetColorRange	Sets the foreground and background colors of a block of cells
SSSetColPageBreak	Sets whether a page break occurs at the specified column when printing the spreadsheet
SSSetColUserData	Sets a four-byte user-defined data item for the specified column
SSSetColWidth	Sets the width of the specified column
SSSetColWidthInPixels	Sets the width of the specified column in pixels
SSSetCursor	Sets the pointer the fpSpread control displays in a variety of situations
SSSetCustomName	Sets a custom name and its value
SSSetData	Sets data in a cell, a column, a row, or the entire spreadsheet
SSSetDataRange	Sets the data for the specified block of cells
SSSetDefFloatFormat	Sets the default float characters used by all float cells in the spreadsheet
SSSetEditEnterAction	Sets the action performed when the user presses the Enter key
SSSetEditMode	Sets whether a cell is in edit mode
SSSetFloat	Sets the floating-point value for a float cell
SSSetFloatRange	Sets the floating-point value for a float cell type for the specified block of cells
SSSetFont	Sets the font used in a cell, a column, a row, or the entire spreadsheet
SSSetFontRange	Sets the font used in the specified block of cells
SSSetFormula	Sets the formula of a cell, a column, a row, or the entire spreadsheet
SSSetFormulaRange	Sets the formula of a specified block of cells
SSSetFreeze	Sets the number of frozen (nonscrolling) rows or columns
SSSetGrayAreaColor	Sets the gray area background color of the spreadsheet
SSSetGridColor	Sets the color of the spreadsheet grid lines
SSSetGridType	Sets whether horizontal and vertical grid lines are displayed, and the style of the grid lines displayed

## fpSpread and fpSpreadPreview DLL Function Summary

DLL Function	Description
SSSetInteger	Sets the integer value in an integer cell
SSSetIntegerRange	Sets the integer value for the specified block of integer cells
SSSetIteration	Sets the number of iterations and the change between iterations for circular references
SSSetLock	Sets whether the user can type data into a cell, a row, a column, a block of cells, or the entire spreadsheet
SSSetLockColor	Sets the background and foreground colors displayed in locked cells
SSSetLockRange	Sets whether the user can type data into a block of cells or the entire spreadsheet
SSSetMaxCols	Sets the maximum number of usable columns in the spreadsheet
SSSetMaxRows	Sets the maximum number of usable rows in the spreadsheet
SSSetMultiSelBlocks	Sets the selected blocks in the spreadsheet
SSSetOddEvenRowColor	Specifies the background and text colors for odd and even rows
SSSetOperationMode	Sets operation mode of the spreadsheet
SSSetOwner	Sets the owner window handle
SSSetPrintOptions	Sets the print options settings for the spreadsheet
SSSetRefStyle	Sets the reference style used by the spreadsheet to represent formulas and custom names
SSSetRowHeaderDisplay	Sets whether the row header displays letters, numbers, or is blank
SSSetRowHeight	Sets the height of the specified spreadsheet row
SSSetRowHeightInPixels	Sets the height of a specified row in pixels
SSSetRowPageBreak	Sets whether a page break occurs at the specified row when printing the spreadsheet
SSSetRowUserData	Sets a four-byte, user-defined data item for the specified row
SSSetSel	Sets the starting position and length of text that can be selected within the active cell when the cell is in edit mode
SSSetSelBlockOptions	Sets whether the user can select columns, rows, blocks of cells, all cells, or any combination of these
SSSetSelColor	Specifies the background and text colors of the selected row or rows
SSSetSelectBlock	Sets the upper-left and lower-right cell positions of a block selection

## Appendix C

<b>DLL Function</b>	<b>Description</b>
SSSetShadowColor	Sets the shadow colors displayed in the fpSpread control
SSSetStartingNumbers	Sets the starting numbers for the column and row headers
SSSetTextTip	Specifies whether to display text tips and the location and the appearance of the tips
SSSetTwoDigitYearMax	Specifies the year value used to calculate the century when year values are given as two-digit values
SSSetTypeButton	Sets the style, text, picture, type, and shadow size for button cells
SSSetTypeCheckBox	Sets the style, text, and pictures displayed in check box cells
SSSetTypeComboBox	Sets the style and list items for combo box cells
SSSetTypeComboBoxEx	Sets the style, list items, drop-down list display, edit field maximum characters, and optional control to display for the drop-down list for combo box cells
SSSetTypeDate	Sets the style, format, and minimum and maximum dates for date cells
SSSetTypeEdit	Sets the style, length, and characters and cases allowed for edit cells
SSSetTypeFloat	Sets the style, number of digits, and minimum and maximum values for float cells
SSSetTypeFloatExt	Sets the style, number of digits, minimum and maximum values, and format for float cells
SSSetTypeInteger	Sets the minimum and maximum values for integer cells
SSSetTypeIntegerExt	Sets the style, maximum and minimum values, and spin button characteristics for integer cells that contain spin buttons
SSSetTypeOwnerDraw	Sets the style for owner-drawn cells
SSSetTypePic	Sets the style and mask for PIC cells
SSSetTypePicture	Sets the style and picture for picture cells
SSSetTypePictureHandle	Sets the style, handle, and palette for a picture cell
SSSetTypeStaticText	Sets parameters for static text cells
SSSetTypeTime	Sets the style, format, and minimum and maximum values for time cells
SSSetUnitType	Sets the unit type used to specify column width and row height
SSSetUserData	Sets a user-defined, four-byte data value for the fpSpread control to be used by the application
SSSetUserResize	Sets whether the user can resize columns, rows, both, or neither

## fpSpread and fpSpreadPreview DLL Function Summary

DLL Function	Description
SSSetUserResizeCol	Sets the column resize option for the specified column
SSSetUserResizeRow	Sets the row resize option for the specified row
SSSetValue	Sets unformatted data in a cell
SSSetValueRange	Sets unformatted data in a block of cells
SSShowActiveCell	Scrolls the spreadsheet so the active cell is positioned as specified
SSShowCell	Scrolls the spreadsheet so that an individual cell is positioned as specified
SSShowCol	Determines whether a column is displayed or hidden
SSShowRow	Determines whether a row is displayed or hidden
SSSort	Sorts a block of cells using the quick sort method
SSSortEx	Sets the sort parameters and performs a sort on the specified data
SSSwapRange	Swaps a block of cells with another block of cells
SSValidateFormula	Determines if a formula is valid
SSVGetBufferSize	Returns the size of the virtual buffer when virtual mode is on
SSVGetMax	Returns the maximum number of virtual rows when virtual mode is on
SSVGetPhysBufferSize	Returns the size of the virtual buffer when virtual mode is on
SSVGetPhysBufferTop	Returns the current top row of the virtual buffer when virtual mode is on
SSVGetStyle	Returns the virtual mode style for the spreadsheet
SSVRefreshBuffer	Forces the spreadsheet to re-request the data currently in the virtual buffer
SSVScrollGetSpecial	Returns whether the spreadsheet uses a special vertical scroll bar and the scroll arrows to display on that scroll bar
SSVScrollSetSpecial	Sets whether the spreadsheet uses a special vertical scroll bar and the scroll arrows to display on that scroll bar
SSVSetBufferSize	Sets the size of the virtual buffer when virtual mode is on
SSVSetMax	Sets the maximum number of virtual rows when virtual mode is on
SSVSetStyle	Sets the virtual mode style for the spreadsheet

## fpSpreadPreview Control DLL Function Summary

The following table provides a complete list of the fpSpreadPreview DLL functions and a description of each.

DLL Function	Description
SpvGetAllowUserZoom	Returns whether the user can zoom in or out while viewing pages
SpvGetGrayAreaColor	Returns the gray area background color of the fpSpreadPreview control
SpvGetGrayAreaMarginH	Returns the horizontal space between the displayed page and the edge of the control
SpvGetGrayAreaMarginType	Returns whether the control uses the actual gray area margin settings or scales the settings to optimize the page view
SpvGetGrayAreaMarginV	Returns the vertical space between the displayed page and the edge of the control
SpvGethWndSpread	Returns the window handle of the spreadsheet you want to preview
SpvGetPageBorderColor	Returns the border color of the displayed pages
SpvGetPageBorderWidth	Returns the border width of the displayed pages
SpvGetPageCurrent	Returns the page displayed in the control
SpvGetPageGutterH	Returns the horizontal space between displayed pages
SpvGetPageGutterV	Returns the vertical space between displayed pages
SpvGetPageMultiCntH	Returns the number of pages displayed in the horizontal direction
SpvGetPageMultiCntV	Returns the number of pages displayed in the vertical direction
SpvGetPagePercentageActual	Returns the actual size of the displayed pages
SpvGetPageShadowColor	Returns the color of the shadow behind the displayed pages
SpvGetPageShadowWidth	Returns the width of the shadow behind the displayed pages
SpvGetPagesPerScreen	Returns the number of pages displayed at one time
SpvGetPageViewPercentage	Returns the percent size of the displayed pages
SpvGetPageViewType	Returns the type of the pages being displayed
SpvGetScrollBarH	Returns whether and how the horizontal scroll bar is displayed
SpvGetScrollBarV	Returns whether and how the vertical scroll bars are displayed

## fpSpread and fpSpreadPreview DLL Function Summary

DLL Function	Description
SpvGetScrollIncH	Returns the distance the scroll box moves in the horizontal scroll bar when you scroll through the displayed pages
SpvGetScrollIncV	Returns the distance the scroll box moves in the vertical scroll bar when you scroll through the displayed pages
SpvGetZoomState	Returns the way the control zooms between different views of the displayed pages
SpvSetAllowUserZoom	Sets whether the user can zoom in or out while viewing pages
SpvSetGrayAreaColor	Sets the gray area background color of the fpSpreadPreview control
SpvSetGrayAreaMarginH	Sets the horizontal space between the displayed page and the edge of the control
SpvSetGrayAreaMarginType	Sets whether the control uses the actual gray area margin settings or scales the settings to optimize the page view
SpvSetGrayAreaMarginV	Sets the vertical space between the displayed page and the edge of the control
SpvSetHWndSpread	Sets the window handle of the spreadsheet you want to preview
SpvSetPageBorderColor	Sets the border color of the displayed pages
SpvSetPageBorderWidth	Sets the border width of the displayed pages
SpvSetPageCurrent	Sets the page displayed in the control
SpvSetPageGutterH	Sets the horizontal space between displayed pages
SpvSetPageGutterV	Sets the vertical space between displayed pages
SpvSetPageMultiCntH	Sets the number of pages displayed in the horizontal direction
SpvSetPageMultiCntV	Sets the number of pages displayed in the vertical direction
SpvSetPageShadowColor	Sets the color of the shadow behind the displayed pages
SpvSetPageShadowWidth	Sets the width of the shadow behind the displayed pages
SpvSetPageViewPercentage	Sets the percent size of the displayed pages
SpvSetPageViewType	Sets the type of the pages being displayed
SpvSetScrollBarH	Sets whether and how the horizontal scroll bar is displayed
SpvSetScrollBarV	Sets whether and how the vertical scroll bars are displayed
SpvSetScrollIncH	Sets the distance the scroll box moves in the horizontal scroll bar when you scroll through the displayed pages
SpvSetScrollIncV	Sets the distance the scroll box moves in the vertical scroll bar when you scroll through the displayed pages
SpvSetZoomState	Sets the way the control zooms between different views of the displayed pages



# Appendix D fpSpread Control ActiveX/VBX to DLL Cross Reference

## ActiveX/VBX Properties to DLL Functions

Property	DLL Function
About	N/A
Action	SSSetActiveCell (0) SSShowCell (1) SSSetSelectBlock (2) SSClear (3) SSClearRange (3) SSDelCol (4) SSDelRow (5) SSInsCol (6) SSInsRow (7) SSSaveToFile, SSSaveTabFile (8) SSSaveToFile (9) SSSaveTabFile (10) SSReCalc (11) SSClearData (12) SSClearDataRange (12) SSPrint (13) SSDeSelectBlock (14) N/A (15) SSSetBorder (16) SSSetBorderRange (16) SSAddMultiSelBlocks (17) SSGetMultiSelBlocks (18) SSCopyRange (19) SSMoveRange (20) SSSwapRange (21) SSClipboardCopy (22) SSClipboardCut (23) SSClipboardPaste (24) SSSort (25) SSComboBoxSendMessage (26) (SS_CBM_RESETCONTENT) SSComboBoxSendMessage (27) (SS_CBM_DELETESTRING) SSReset (28) SSVRefreshBuffer (30) N/A (32)
ActiveCol	SSGet/SetActiveCell

## Appendix D

Property	DLL Function
ActiveRow	SSGet/SetActiveCell
AllowCellOverflow	SSGet/SetBool (SSB_ALLOWCELLOVERFLOW)
AllowDragDrop	SSGet/SetBool (SSB_ALLOWDRAGDROP)
AllowMultiBlocks	SSGet/SetBool (SSB_ALLOWMULTIBLOCKS)
AllowUserFormulas	SSGet/SetBool (SSB_ALLOWUSERFORMULAS)
Appearance	SSGet/SetAppearance
ArrowsExitEditMode	SSGet/SetBool (SSB_ARROWSEXITEDITMODE)
AutoCalc	SSGet/SetBool (SSB_AUTOCALC)
AutoClipboard	SSGet/SetBool (SSB_AUTOCLIPBOARD)
AutoSize	SSGet/SetBool (SSB_AUTOSIZE)
BackColor	SSGet/SetColor SSSetColorRange
BackColorStyle	SSGet/SetBackColorStyle
BlockMode	N/A
BorderStyle	Microsoft Windows function: GetWindowLong GWL_STYLE WS_BORDER
ButtonDrawMode	SSGet/SetButtonDrawMode
CellBorderColor	SSGetBorderEx SSGet/SetBorder SSSetBorderRange
CellBorderStyle	SSGetBorderEx SSGet/SetBorder SSSetBorderRange
CellBorderType	SSGet/SetBorder SSSetBorderRange
CellType	SSGet/SetCellType SSSetCellTypeRange
ChangeMade	SSGet/SetBool (SSB_CHANGEMADE)
Clip	SSClipIn SSClipOut
ClipboardOptions	SSGet/SetClipboardOptions
ClipValue	SSClipValueIn SSClipValueOut
Col	N/A

Property	DLL Function
Col2	N/A
ColHeaderDisplay	SSGet/SetColHeaderDisplay
ColHidden	SSIsColHidden SSShowCol
ColPageBreak	SSGet/SetColPageBreak
ColsFrozen	SSGet/SetFreeze
ColWidth	SSGet/SetColWidth SSGet/SetColWidthInPixels
CursorIcon	SSGet/SetCursor
CursorStyle	SSGet/SetCursor
CursorType	SSGet/SetCursor
DataChanged	SSGet/SetBool (SSB_CHANGEMADE)
DataColCnt	SSGetDataCnt
DataField	N/A
DataFillEvent	N/A
DataRowCnt	SSGetDataCnt
DataSource	N/A
DAutoCellTypes	N/A
DAutoFill	N/A
DAutoHeadings	N/A
DAutoSave	N/A
DAutoSizeCols	N/A
DestCol	N/A
DestRow	N/A
DInformActiveRowChange	N/A
DisplayColHeaders	SSGet/SetBool (SSB_SHOWCOLHEADERS)
DisplayRowHeaders	SSGet/SetBool (SSB_SHOWROWHEADERS)
DragIcon	N/A
DragMode	N/A
EditEnterAction	SSGet/SetEditEnterAction
EditMode	SSGet/SetEditMode
EditModePermanent	SSGet/SetBool (SSB_EDITMODEPERMANENT)
EditModeReplace	SSGet/SetBool (SSB_EDITMODEREPLACE)

## Appendix D

Property	DLL Function
Enabled	Microsoft Windows function: GetWindowLong GWL_STYLE WS_DISABLED
FileNum	N/A
FloatDefCurrencyChar	SSGet/SetDefFloatFormat
FloatDefDecimalChar	SSGet/SetDefFloatFormat
FloatDefSepChar	SSGet/SetDefFloatFormat
Font	SSGet/SetFont SSSetFontRange
FontBold	SSGet/SetFont SSSetFontRange
FontItalic	SSGet/SetFont SSSetFontRange
FontName	SSGet/SetFont SSSetFontRange
FontSize	SSGet/SetFont SSSetFontRange
FontStrikethru	SSGet/SetFont SSSetFontRange
FontUnderline	SSGet/SetFont SSSetFontRange
ForeColor	SSGet/SetColor SSSetColorRange
Formula	SSGet/SetFormula SSSetFormulaRange
FormulaSync	SSGet/SetBool (SSB_FORMULASYNC)
GrayAreaBackColor	SSGet/SetGrayAreaColor
GridColor	SSGet/SetGridColor
GridShowHoriz	SSGet/SetGridType
GridShowVert	SSGet/SetGridType
GridSolid	SSGet/SetGridType
hDCPrinter	SSPrint
Height	Microsoft Windows functions: GetWindowRect MoveWindow SetWindowPos

Property	DLL Function
HelpContextID	N/A
hWnd	N/A
Index	N/A
IsBlockSelected	SSGetSelectBlock
Left	Microsoft Windows functions: GetWindowRect MoveWindow SetWindowPos
LeftCol	SSGetTopLeftCell SSShowCell
Lock	SSGet/SetLock SSSetLockRange
LockBackColor	SSGet/SetLockColor
LockForeColor	SSGet/SetLockColor
MaxCols	SSGet/SetMaxCols
MaxRows	SSGet/SetMaxRows
MaxTextCellHeight	SSGetMaxTextCellSize
MaxTextCellWidth	SSGetMaxTextCellSize
MaxTextColWidth	SSGetMaxTextColWidth
MaxTextRowHeight	SSGetMaxTextRowHeight
MoveActiveOnFocus	SSGet/SetBool (SSB_MOVEACTIVEONFOCUS)
MultiSelCount	SSGetMultiSelBlocks
MultiSelIndex	SSGetMultiSelBlocks
Name	N/A
NoBeep	SSGet/SetBool (SSB_NOBEEP)
NoBorder	SSGet/SetBool (SSB_NOBORDER)
OLEDropMode	N/A
OperationMode	SSGet/SetOperationMode
Parent	SSGetOwner
Position	SSShowActiveCell SSShowCell
PrintAbortMsg	SSPrint
PrintBorder	SSPrint
PrintColHeaders	SSPrint
PrintColor	SSPrint

## Appendix D

Property	DLL Function
PrintFirstPageNumber	N/A
PrintFooter	SSPrint
PrintGrid	SSPrint
PrintHeader	SSPrint
PrintJobName	SSPrint
PrintMarginBottom	SSPrint
PrintMarginLeft	SSPrint
PrintMarginRight	SSPrint
PrintMarginTop	SSPrint
PrintNextPageBreakCol	SSGetNextPageBreakCol
PrintNextPageBreakRow	SSGetNextPageBreakRow
PrintOrientation	SSPrint
PrintPageCount	SSGetPrintPageCount
PrintPageEnd	SSPrint
PrintPageOrder	SSGet/SetPrintOptions
PrintPageStart	SSPrint
PrintRowHeaders	SSPrint
PrintShadows	SSPrint
PrintSmartPrint	SSGet/SetPrintOptions
PrintType	SSPrint
PrintUseDataMax	SSPrint
ProcessTab	SSGet/SetBool (SSB_PROCESSTAB)
Protect	SSGet/SetBool (SSB_PROTECT)
ReDraw	SSGet/SetBool (SSB_REDRAW)
RestrictCols	SSGet/SetBool (SSB_RESTRICTCOLS)
RestrictRows	SSGet/SetBool (SSB_RESTRICTROWS)
RetainSelBlock	SSGet/SetBool (SSB_RETAINSELBLOCK)
Row	N/A
Row2	N/A
RowHeaderDisplay	SSGet/SetRowHeaderDisplay
RowHeight	SSGet/SetRowHeight SSGet/SetRowHeightInPixels
RowHidden	SSIsRowHidden SSShowRow

Property	DLL Function
RowPageBreak	SSGet/SetRowPageBreak
RowsFrozen	SSGet/SetFreeze
ScrollBarExtMode	SSGet/SetBool (SSB_SCROLLBAREXTMODE)
ScrollBarMaxAlign	SSGet/SetBool (SSB_SCROLLBARMAXALIGN)
ScrollBars	SSGet/SetBool (SSB_HORZSCROLLBAR, SSB_VERTSCROLLBAR)
ScrollBarShowMax	SSGet/SetBool (SSB_SCROLLBARSHOWMAX)
ScrollBarTrack	SSGet/SetBool (SSB_HSCROLLBARTRACK, SSB_VSCROLLBARTRACK)
SelBackColor	SSGet/SetSelColor
SelBlockCol	SSGetActiveCell SSGetSelectBlock
SelBlockCol2	SSGetActiveCell SSGetSelectBlock
SelBlockRow	SSGetActiveCell SSGetSelectBlock
SelBlockRow2	SSGetActiveCell SSGetSelectBlock
SelectBlockOptions	SSGet/SetSelBlockOptions
SelForeColor	SSGet/SetSelColor
SelLength	SSGet/SetSel
SelModeIndex	SSSelModeSendMessage (SS_SELMODE_GETCURSEL, SS_SELMODE_SETCURSEL)
SelModeSelCount	SSSelModeSendMessage (SS_SELMODE_GETSELCOUNT)
SelModeSelected	SSSelModeSendMessage (SS_SELMODE_GETSEL, SS_SELMODE_SETSEL)
SelStart	SSGet/SetSel
SelText	SSGetSelText SSReplaceSel
ShadowColor	SSGet/SetShadowColor
ShadowDark	SSGet/SetShadowColor
ShadowText	SSGet/SetShadowColor
SortBy	SSSort
SortKey	SSSort
SortKeyOrder	SSSort
StartingColNumber	SSGet/SetStartingNumbers
StartingRowNumber	SSGet/SetStartingNumbers
TabIndex	N/A

## Appendix D

Property	DLL Function
TabStop	N/A
Tag	N/A
Text	SSGet/SetData SSSetDataRange
TextTip	Get/SetTextTip
TextTipDelay	Get/SetTextTip
Top	Microsoft Windows functions: GetWindowRect MoveWindow SetWindowPos
TopRow	SSGetTopLeftCell SSShowCell
TwoDigitYearMax	SSGet/SetTwoDigitYearMax
TypeButtonAlign	SSSetTypeButton
TypeButtonBorderColor	SSSetTypeButton
TypeButtonColor	SSSetTypeButton
TypeButtonDarkColor	SSSetTypeButton
TypeButtonLightColor	SSSetTypeButton
TypeButtonPicture	SSSetTypeButton
TypeButtonPictureDown	SSSetTypeButton
TypeButtonShadowSize	SSSetTypeButton
TypeButtonText	SSGet/SetButtonText
TypeButtonTextColor	SSSetTypeButton
TypeButtonType	SSSetTypeButton
TypeCheckCenter	SSSetTypeCheckBox
TypeCheckPicture	SSSetTypeCheckBox
TypeCheckText	SSSetTypeCheckBox
TypeCheckTextAlign	SSSetTypeCheckBox
TypeCheckType	SSSetTypeCheckBox
TypeComboBoxCount	SSSetTypeComboBox
TypeComboBoxCurSel	SSSetTypeComboBox
TypeComboBoxEditable	SSSetTypeComboBox SSSetTypeComboBoxEx
TypeComboBoxhWnd	SSSetTypeComboBoxEx
TypeComboBoxIndex	SSSetTypeComboBox

Property	DLL Function
TypeComboBoxList	SSSetTypeComboBox SSSetTypeComboBoxEx
TypeComboBoxMaxDrop	SSSetTypeComboBoxEx
TypeComboBoxString	SSSetTypeComboBox SSSetTypeComboBoxEx
TypeComboBoxWidth	SSSetTypeComboBoxEx
TypeDateCentury	SSSetTypeDate
TypeDateFormat	SSSetTypeDate
TypeDateMax	SSSetTypeDate
TypeDateMin	SSSetTypeDate
TypeDateSeparator	SSSetTypeDate
TypeEditCharCase	SSSetTypeEdit
TypeEditCharSet	SSSetTypeEdit
TypeEditMultiLine	SSSetTypeEdit
TypeEditPassword	SSSetTypeEdit
TypeFloatCurrencyChar	SSSetTypeFloatExt
TypeFloatDecimalChar	SSSetTypeFloatExt
TypeFloatDecimalPlaces	SSSetTypeFloat SSSetTypeFloatExt
TypeFloatMax	SSSetTypeFloat SSSetTypeFloatExt
TypeFloatMin	SSSetTypeFloat SSSetTypeFloatExt
TypeFloatMoney	SSSetTypeFloat SSSetTypeFloatExt
TypeFloatSeparator	SSSetTypeFloat SSSetTypeFloatExt
TypeFloatSepChar	SSSetTypeFloatExt

## Appendix D

Property	DLL Function
TypeHAlign	SSSetTypeCheckBox SSSetTypeComboBox SSSetTypeComboBoxEx SSSetTypeDate SSSetTypeEdit SSSetTypeFloat SSSetTypeFloatExt SSSetTypeIntegerExt SSSetTypePic SSSetTypePicture SSSetTypeStaticText SSSetTypeTime
TypeIntegerMax	SSSetTypeInteger SSSetTypeIntegerExt
TypeIntegerMin	SSSetTypeInteger SSSetTypeIntegerExt
TypeIntegerSpinInc	SSSetTypeIntegerExt
TypeIntegerSpinWrap	SSSetTypeIntegerExt
TypeMaxEditLen	SSSetTypeEdit SSSetTypeComboBoxEx
TypeOwnerDrawStyle	SSSetTypeOwnerDraw
TypePicDefaultText	N/A
TypePicMask	SSSetTypePic
TypePictCenter	SSSetTypePicture SSSetTypePictureHandle
TypePictMaintainScale	SSSetTypePicture SSSetTypePictureHandle
TypePictPicture	SSSetTypePicture SSSetTypePictureHandle
TypePictStretch	SSSetTypePicture SSSetTypePictureHandle
TypeSpin	SSSetTypeDate SSSetTypeIntegerExt SSSetTypeTime
TypeTextPrefix	SSSetTypeStaticText
TypeTextShadow	SSSetTypeStaticText
TypeTextShadowIn	SSSetTypeStaticText
TypeTextWordWrap	SSSetTypeStaticText

Property	DLL Function
TypeTime24Hour	SSSetTypeTime
TypeTimeMax	SSSetTypeTime
TypeTimeMin	SSSetTypeTime
TypeTimeSeconds	SSSetTypeTime
TypeTimeSeparator	SSSetTypeTime
TypeVAlign	SSSetTypeCheckBox SSSetTypeComboBox SSSetTypeComboBoxEx SSSetTypeDate SSSetTypeEdit SSSetTypeFloat SSSetTypeFloatExt SSSetTypeIntegerExt SSSetTypePic SSSetTypePicture SSSetTypeStaticText SSSetTypeTime
UnitType	SSGet/SetUnitType
UserResize	SSGet/SetUserResize
UserResizeCol	SSGet/SetUserResizeCol
UserResizeRow	SSGet/SetUserResizeRow
Value	SSGet/SetData SSSetDataRange SSGet/SetFloat SSSetFloatRange SSGet/SetInteger SSSetIntegerRange SSGet/SetValue SSSetValueRange
VirtualCurRowCount	SSVGetPhysBufferSize
VirtualCurTop	SSVGetPhysBufferTop
VirtualMaxRows	SSVGetMax SSVSetMax
VirtualMode	SSGet/SetBool (SSB_VIRTUALMODE)
VirtualOverlap	SSVGetBufferSize SSVSetBufferSize
VirtualRows	SSVGetBufferSize SSVSetBufferSize

## Appendix D

Property	DLL Function
VirtualScrollBuffer	SSVGetStyle SSVSetStyle
Visible	Microsoft Windows functions: IsWindowVisible ShowWindow
VisibleCols	SSGet/SetAutoSizeVisible
VisibleRows	SSGet/SetAutoSizeVisible
VScrollSpecial	SSVScrollGetSpecial SSVScrollSetSpecial
VScrollSpecialType	SSVScrollGetSpecial SSVScrollSetSpecial
Width	Microsoft Windows functions: GetWindowRect MoveWindow SetWindowPos

## ActiveX/VBX Functions to DLL Functions

ActiveX/VBX Function	DLL Function
AddCustomFunction	SSAddCustomFunction
AddCustomFunctionExt	SSAddCustomFunctionExt
CFGetCellParam	N/A
CFGetDoubleParam	N/A
CFGetDoubleParamExt	N/A
CFGetLongParam	N/A
CFGetParamInfo	N/A
CFGetRangeParam	N/A
CFGetStringParam	N/A
CFSetResult	N/A
ColNumberToLetter	N/A
ColWidthToTwips	SSColWidthToLogUnits
DerefHlstrLen	N/A
EnumCustomFunction	SSEnumCustomFunction SSEnumCustomFunctionLen
ExportRangeToHTML	SSExportRangeToHTML

ActiveX/VBX Function	DLL Function
ExportToExcel	SSExportToExcel
ExportToHTML	SSExportToHTML
GetActionKey	SSGetActionKey
GetArray	SSGetArray
GetBottomRightCell	SSGetBottomRightCell
GetCellDirtyFlag	N/A
GetCellFromScreenCoord	SSGetCellFromPixel
GetCellPos	SSGetCellRect
GetClientArea	SSGetClientRect
GetColItemData	SSGetColUserData
GetCustomFunction	SSGetCustomFunction
GetCustomName	SSGetCustomName
GetDataFillData	N/A
GetExcelSheetList	SSGetExcelSheetList
GetFirstValidCell	SSGetFirstValidCell
GetFloat	SSGetFloat
GetFormulaSync	SSGetBool (SSB_FORMULASYNC)
GetInteger	SSGetInteger
GetItemData	SSGetUserData
GetIteration	SSGetIteration
GetLastValidCell	SSGetLastValidCell
GetMultiSelItem	SSSelModeSendMessage (SS_SELMODE_GETSELITEM, SS_SELMODE_GETSELITEMS)
GetNextPageBreakCol	SSGetNextPageBreakCol
GetNextPageBreakRow	SSGetNextPageBreakRow
GetOddEvenRowColor	SSGetOddEvenRowColor
GetPrintOptions	SSGetPrintOptions
GetPrintPageCount	SSGetPrintPageCount
GetRefStyle	SSGetRefStyle
GetRowItemData	SSGetRowUserData
GetText	SSGetInteger SSGetFloat SSGetData

## Appendix D

<b>ActiveX/VBX Function</b>	<b>DLL Function</b>
GetTextTipApperance	SSGetTextTip
GetTwoDigitYearMax	GetTwoDigitYearMax
ImportExcelSheet	SSImportExcelSheet
IsCellSelected	SSIsCellInSelection
IsExcelFile	SSIsExcelFile
IsFormulaValid	SSValidateFormula
IsVisible	SSGetVisible
LoadFromFile	SSLoadFromFile
LoadTabFile	SSLoadTabFile
QueryCustomName	SSQueryCustomName
ReCalcCell	SSReCalcCell
RemoveCustomFunction	SSRemoveCustomFunction
RowHeightToTwips	SSRowHeightToLogUnits
SaveTabFile	SSSaveTabFile
SaveToFile	SSSaveToFile
SetActionKey	SSSetActionKey
SetArray	SSSetArray
SetCalText	SSSetCalText
SetCellDirtyFlag	N/A
SetColItemData	SSSetColUserData
SetCustomName	SSSetCustomName
SetDataFillData	N/A
SetFloat	SSSetFloat
SetFormulaSync	SSSetBool (SSB_FORMULASYNC)
SetInteger	SSSetInteger
SetItemData	SSSetUserData
SetIteration	SSSetIteration
SetOddEvenRowColor	SSSetOddEvenRowColor
SetPrintOptions	SSSetPrintOptions
SetRefStyle	SSSetRefStyle
SetRowItemData	SSSetRowUserData

<b>ActiveX/VBX Function</b>	<b>DLL Function</b>
SetText	SSSetInteger SSSetFloat SSSetData
SetTextTipAppearance	SSSetTextTip
SetTwoDigitYearMax	SSSetTwoDigitYearMax
TwipsToColWidth	SSLogUnitsToColWidth
TwipsToRowHeight	SSLogUnitsToRowHeight



# Appendix E fpSpread Control DLL to ActiveX/VBX Cross Reference

## DLL Functions to ActiveX/VBX Properties and Functions

DLL Function	Property or Function
SSAddCustomFunction	AddCustomFunction
SSAddCustomFunctionExt	AddCustomFunctionExt
SSAddMultiSelBlocks	Action (17)
SSClear	Action (3)
SSClearData	Action (12)
SSClearDataRange	Action (12)
SSClearRange	Action (3)
SSClipboardCopy	Action (22)
SSClipboardCut	Action (23)
SSClipboardPaste	Action (24)
SSClipIn	Clip
SSClipOut	Clip
SSClipValueIn	ClipValue
SSClipValueOut	ClipValue
SSColWidthToLogUnits	ColWidthToTwips
SSComboBoxSendMessage	Action (26) Action (27)
SSCopyRange	Action (19)
SSDelCol	Action (4)
SSDelColRange	N/A
SSDelRow	Action (5)
SSDelRowRange	N/A
SSDeSelectBlock	Action (14)
SSEnumCustomFunction	EnumCustomFunction
SSEnumCustomFunctionLen	EnumCustomFunction
SSExportRangeToHTML	ExportRangeToHTML
SSExportToExcel	ExportToExcel
SSExportToHTML	ExportToHTML
SSFloatFormat	N/A
SSGetActionKey	GetActionKey

## Appendix E

DLL Function	Property or Function
SSGetActiveCell	ActiveCol ActiveRow SelBlockCol SelBlockCol2 SelBlockRow SelBlockRow2
SSGetAppearance	Appearance
SSGetArray	GetArray
SSGetAutoSizeVisible	VisibleCols VisibleRows
SSGetBackColorStyle	BackColorStyle
SSGetBool (SSB_ALLOWCELLOVERFLOW)	AllowCellOverflow
SSGetBool (SSB_ALLOWDRAGDROP)	AllowDragDrop
SSGetBool (SSB_ALLOWMULTIBLOCKS)	AllowMultiBlocks
SSGetBool (SSB_ALLOWUSERFORMULAS)	AllowUserFormulas
SSGetBool (SSB_ARROWSEXITEDITMODE)	ArrowsExitEditMode
SSGetBool (SSB_AUTOCALC)	AutoCalc
SSGetBool (SSB_AUTOCLIPBOARD)	AutoClipboard
SSGetBool (SSB_AUTOSIZE)	AutoSize
SSGetBool (SSB_CHANGEMADE)	DataChanged
SSGetBool (SSB_EDITMODEPERMANENT)	EditModePermanent
SSGetBool (SSB_EDITMODEREPLACE)	EditModeReplace
SSGetBool (SSB_FORMULASYNC)	FormulaSync GetFormulaSync
SSGetBool (SSB_GRIDLINES)	GridShowHoriz GridShowVert
SSGetBool (SSB_HORZSCROLLBAR)	ScrollBars
SSGetBool (SSB_HSCROLLBARTRACK)	ScrollBarTrack
SSGetBool (SSB_MOVEACTIVEONFOCUS)	MoveActiveOnFocus
SSGetBool (SSB_NOBEEP)	NoBeep
SSGetBool (SSB_NOBORDER)	NoBorder
SSGetBool (SSB_PROCESSTAB)	ProcessTab
SSGetBool (SSB_PROTECT)	Protect
SSGetBool (SSB_REDRAW)	ReDraw
SSGetBool (SSB_RESTRICTCOLS)	RestrictCols
SSGetBool (SSB_RESTRICTROWS)	RestrictRows
SSGetBool (SSB_RETAINSELBLOCK)	RetainSelBlock
SSGetBool (SSB_SCROLLBAREXTMODE)	ScrollBarExtMode

DLL Function	Property or Function
SSGetBool (SSB_SCROLLBARMAXALIGN)	ScrollBarMaxAlign
SSGetBool (SSB_SCROLLBARSHOWMAX)	ScrollBarShowMax
SSGetBool (SSB_SHOWCOLHEADERS)	DisplayColHeaders
SSGetBool (SSB_SHOWROWHEADERS)	DisplayRowHeaders
SSGetBool (SSB_VERTSCROLLBAR)	ScrollBars
SSGetBool (SSB_VIRTUALMODE)	VirtualMode
SSGetBool (SSB_VSCROLLBARTRACK)	ScrollBarTrack
SSGetBorder	CellBorderColor CellBorderStyle CellBorderType
SSGetBorderEx	CellBorderColor CellBorderStyle CellBorderType
SSGetBottomRightCell	GetBottomRightCell
SSGetButtonDrawMode	ButtonDrawMode
SSGetButtonText	TypeButtonText
SSGetCallBack	N/A
SSGetCellFromPixel	GetCellFromScreenCoord
SSGetCellRect	GetCellPos
SSGetCellSendingMsg	N/A
SSGetCellType	CellType
SSGetClientRect	GetClientArea
SSGetClipboardOptions	ClipboardOptions
SSGetColHeaderDisplay	ColHeaderDisplay
SSGetColor	BackColor ForeColor
SSGetColPageBreak	ColPageBreak
SSGetColUserData	GetCollItemData
SSGetColWidth	ColWidth
SSGetColWidthInPixels	ColWidth
SSGetCurrSelBlockPos	N/A
SSGetCursor	CursorIcon CursorStyle CursorType
SSGetCustomFunction	GetCustomFunction
SSGetCustomName	GetCustomName
SSGetCustomNameLen	N/A

## Appendix E

DLL Function	Property or Function
SSGetData	GetText Text
SSGetDataCnt	DataColCnt DataRowCnt
SSGetDataLen	N/A
SSGetDefFloatFormat	FloatDefCurrencyChar FloatDefDecimalChar FloatDefSepChar
SSGetEditEnterAction	EditEnterAction
SSGetEditMode	EditMode
SSGetExcelSheetList	GetExcelSheetList
SSGetFirstValidCell	GetFirstValidCell
SSGetFloat	GetFloat
SSGetFont	Font FontBold FontItalic FontName FontSize FontStrikethru FontUnderline
SSGetFormula	Formula
SSGetFormulaLen	N/A
SSGetFreeze	ColsFrozen RowsFrozen
SSGetGrayAreaColor	GrayAreaBackColor
SSGetGridColor	GridColor
SSGetGridType	GridShowHoriz GridShowVert GridSolid
SSGetInteger	GetInteger
SSGetIteration	GetIteration
SSGetLastValidCell	GetLastValidCell
SSGetLock	Lock
SSGetLockColor	LockBackColor LockForeColor
SSGetMaxCols	MaxCols
SSGetMaxRows	MaxRows
SSGetMaxTextCellSize	MaxTextCellHeight MaxTextCellWidth
SSGetMaxTextColWidth	MaxTextColWidth

DLL Function	Property or Function
SSGetMaxTextRowHeight	MaxTextRowHeight
SSGetMultiSelBlocks	Action (18) MultiSelCount MultiSelIndex
SSGetNextPageBreakCol	GetNextPageBreakCol PrintNextPageBreakCol
SSGetNextPageBreakRow	GetNextPageBreakRow PrintNextPageBreakRow
SSGetOddEvenRowColor	GetOddEvenRowColor
SSGetOperationMode	OperationMode
SSGetOwner	Parent
SSGetPrintOptions	GetPrintOptions PrintPageOrder PrintSmartPrint
SSGetPrintPageCount	GetPrintPageCount PrintPageCount
SSGetRefStyle	GetRefStyle
SSGetRowHeaderDisplay	RowHeaderDisplay
SSGetRowHeight	RowHeight
SSGetRowHeightInPixels	RowHeight
SSGetRowPageBreak	RowPageBreak
SSGetRowUserData	GetRowItemData
SSGetSel	SelLength SelStart
SSGetSelBlockOptions	SelectBlockOptions
SSGetSelColor	SelBackColor SelForeColor
SSGetSelectBlock	IsBlockSelected SelBlockCol SelBlockCol2 SelBlockRow SelBlockRow2
SSGetSelText	SelText
SSGetShadowColor	ShadowText ShadowColor ShadowDark
SSGetStartingNumber	StartingColNumber StartingRowNumber

## Appendix E

DLL Function	Property or Function
SSGetTextTip	GetTextTipAppearance TextTip TextTipDelay
SSGetTopLeftCell	LeftCol TopRow
SSGetTopLeftPrev	N/A
SSGetTwoDigitYearMax	GetTwoDigitYearMax TwoDigitYearMax
SSGetUnitType	UnitType
SSGetUserData	GetItemData
SSGetUserResize	UserResize
SSGetUserResizeCol	UserResizeCol
SSGetUserResizeRow	UserResizeRow
SSGetValue	Value
SSGetValueLen	N/A
SSGetVisible	IsVisible
SSImportExcelSheet	ImportExcelSheet
SSInsCol	Action (6)
SSInsColRange	N/A
SSInsRow	Action (7)
SSInsRowRange	N/A
SSIsCellInSelection	IsCellSelected
SSIsColHidden	ColHidden
SSIsExcelFile	IsExcelFile
SSIsRowHidden	RowHidden
SSLoadFromBuffer	N/A
SSLoadFromFile	Action (8) LoadFromFile
SSLoadTabFile	LoadTabFile
SSLogUnitsToColWidth	TwipsToColWidth
SSLogUnitsToRowHeight	TwipsToRowHeight
SSMoveRange	Action (20)

DLL Function	Property or Function
SSPrint	Action (13) hDCPrinter PrintAbortMsg PrintBorder PrintColHeaders PrintColor PrintFooter PrintGrid PrintHeader PrintJobName PrintMarginBottom PrintMarginLeft PrintMarginRight PrintMarginTop PrintPageEnd PrintPageStart PrintRowHeaders PrintShadows PrintType PrintUseDataMax
SSQueryCustomName	QueryCustomName
SSQueryCustomNameLen	N/A
SSReCalc	Action (11)
SSReCalcCell	ReCalcCell
SSRemoveCustomFunction	RemoveCustomFunction
SSReplaceSel	SelText
SSReset	Action (28)
SSRowHeightToLogUnits	RowHeightToTwips
SSSaveTabFile	SaveTabFile
SSSaveToBuffer	N/A
SSSaveToFile	Action (9) Action (10) SaveToFile
SSSelModeSendMessage	SelModeIndex SelModeSelCount SelModeSelected GetMultiSelItem
SSSetActionKey	SetActionKey
SSSetActiveCell	Action (0) ActiveCol ActiveRow
SSSetAppearance	Appearance

## Appendix E

DLL Function	Property or Function
SSSetArray	SetArray
SSSetAutoSizeVisible	VisibleCols VisibleRows
SSSetBackColorStyle	BackColorStyle
SSSetBool (SSB_ALLOWCELLOVERFLOW)	AllowCellOverflow
SSSetBool (SSB_ALLOWDRAGDROP)	AllowDragDrop
SSSetBool (SSB_ALLOWMULTIBLOCKS)	AllowMultiBlocks
SSSetBool (SSB_ALLOWUSERFORMULAS)	AllowUserFormulas
SSSetBool (SSB_ARROWSEXITEDITMODE)	ArrowExitEditMode
SSSetBool (SSB_AUTOCALC)	AutoCalc
SSSetBool (SSB_AUTOCLIPBOARD)	AutoClipboard
SSSetBool (SSB_AUTOSIZE)	AutoSize
SSSetBool (SSB_CHANGEMADE)	DataChanged
SSSetBool (SSB_EDITMODEPERMANENT)	EditModePermanent
SSSetBool (SSB_EDITMODEREPLACE)	EditModeReplace
SSSetBool (SSB_FORMULASYNC)	FormulaSync SetFormulaSync
SSSetBool (SSB_HORZSCROLLBAR)	ScrollBars
SSSetBool (SSB_HSCROLLBARTRACK)	ScrollBarTrack
SSSetBool (SSB_MOVEACTIVEONFOCUS)	MoveActiveOnFocus
SSSetBool (SSB_NOBEEP)	NoBeep
SSSetBool (SSB_NOBORDER)	NoBorder
SSSetBool (SSB_PROCESSTAB)	ProcessTab
SSSetBool (SSB_PROTECT)	Protect
SSSetBool (SSB_REDRAW)	ReDraw
SSSetBool (SSB_RESTRICTCOLS)	RestrictCols
SSSetBool (SSB_RESTRICTROWS)	RestrictRows
SSSetBool (SSB_RETAINSELBLOCK)	RetainSelBlock
SSSetBool (SSB_SCROLLBAREXTMODE)	ScrollBarExtMode
SSSetBool (SSB_SCROLLBARMAXALIGN)	ScrollBarMaxAlign
SSSetBool (SSB_SCROLLBARSHOWMAX)	ScrollBarShowMax
SSSetBool (SSB_SHOWCOLHEADERS)	DisplayColHeaders
SSSetBool (SSB_SHOWROWHEADERS)	DisplayRowHeaders
SSSetBool (SSB_VERTSCROLLBAR)	ScrollBars
SSSetBool (SSB_VIRTUALMODE)	VirtualMode
SSSetBool (SSB_VSCROLLBARTRACK)	ScrollBarTrack

DLL Function	Property or Function
SSSetBorder	Action (16) CellBorderColor CellBorderStyle CellBorderType
SSSetBorderRange	Action (16) CellBorderColor CellBorderStyle CellBorderType
SSSetButtonDrawMode	ButtonDrawMode
SSSetButtonText	TypeButtonText
SSSetCallBack	N/A
SSSetCalText	SetCalText
SSSetCellType	CellType
SSSetCellTypeRange	CellType
SSSetClipboardOptions	SetClipboardOptions
SSSetColHeaderDisplay	ColHeaderDisplay
SSSetColor	BackColor ForeColor
SSSetColorRange	BackColor ForeColor
SSSetColPageBreak	ColPageBreak
SSSetColUserData	SetColItemData
SSSetColWidth	ColWidth
SSSetColWidthInPixels	ColWidth
SSSetCursor	CursorIcon CursorStyle CursorType
SSSetCustomName	SetCustomName
SSSetData	SetText Text
SSSetDataRange	Clip SetArray SetText Text
SSSetDefFloatFormat	FloatDefCurrencyChar FloatDefDecimalChar FloatDefSepChar
SSSetEditEnterAction	EditEnterAction
SSSetEditMode	EditMode
SSSetFloat	SetFloat

## Appendix E

DLL Function	Property or Function
SSSetFloatRange	N/A
SSSetFont	Font FontBold FontItalic FontName FontSize FontStrikethru FontUnderline
SSSetFontRange	Font FontBold FontItalic FontName FontSize FontStrikethru FontUnderline
SSSetFormula	Formula
SSSetFormulaRange	Formula
SSSetFreeze	ColsFrozen RowsFrozen
SSSetGrayAreaColor	GrayAreaBackColor
SSSetGridColor	GridColor
SSSetGridType	GridShowHoriz GridShowVert GridSolid
SSSetInteger	SetInteger
SSSetIntegerRange	N/A
SSSetIteration	SetIteration
SSSetLock	Lock
SSSetLockColor	LockBackColor LockForeColor
SSSetLockRange	Lock
SSSetMaxCols	MaxCols
SSSetMaxRows	MaxRows
SSSetMultiSelBlocks	Action (17, 18) MultiSelCount
SSSetOddEvenRowColor	SetOddEvenRowColor
SSSetOperationMode	OperationMode
SSSetOwner	N/A

DLL Function	Property or Function
SSSetPrintOptions	PrintPageOrder PrintSmartPrint SetPrintOptions
SSSetRefStyle	SetRefStyle
SSSetRowHeaderDisplay	RowHeaderDisplay
SSSetRowHeight	RowHeight
SSSetRowHeightInPixels	RowHeight
SSSetRowPageBreak	RowPageBreak
SSSetRowUserData	SetRowItemData
SSSetSel	SelLength SelStart
SSSetSelBlockOptions	SelectBlockOptions
SSSetSelColor	SelBackColor SelForeColor
SSSetSelectBlock	Action (2)
SSSetShadowColor	ShadowColor ShadowDark ShadowText
SSSetStartingNumber	StartingColNumber StartingRowNumber
SSSetTextTip	SetTextTipAppearance TextTip TextTipDelay
SSSetTwoDigitYearMax	SetTwoDigitYearMax TwoDigitYearMax
SSSetTypeButton	TypeButtonAlign TypeButtonBorderColor TypeButtonColor TypeButtonDarkColor TypeButtonLightColor TypeButtonPicture TypeButtonPictureDown TypeButtonShadowSize TypeButtonText TypeButtonText TypeButtonTextColor TypeButtonType

## Appendix E

DLL Function	Property or Function
SSSetTypeCheckBox	TypeCheckCenter TypeCheckPicture TypeCheckText TypeCheckTextAlign TypeCheckType TypeHAlign TypeVAlign
SSSetTypeComboBox	TypeComboBoxCount TypeComboBoxCurSel TypeComboBoxEditable TypeComboBoxIndex TypeComboBoxList TypeComboBoxString TypeHAlign TypeVAlign
SSSetTypeComboBoxEx	TypeComboBoxEditable TypeComboBoxhWnd TypeComboBoxList TypeComboBoxMaxDrop TypeComboBoxString TypeComboBoxWidth TypeHAlign TypeMaxEditLen TypeVAlign
SSSetTypeDate	TypeDateCentury TypeDateFormat TypeDateMax TypeDateMin TypeDateSeparator TypeHAlign TypeSpin TypeVAlign
SSSetTypeEdit	TypeEditCharCase TypeEditCharSet TypeEditMultiLine TypeEditPassword TypeMaxEditLen
SSSetTypeFloat	TypeFloatDecimalPlaces TypeFloatMax TypeFloatMin TypeFloatMoney TypeFloatSeparator TypeHAlign TypeVAlign

DLL Function	Property or Function
SSSetTypeFloatExt	TypeFloatCurrencyChar TypeFloatDecimalChar TypeFloatDecimalPlaces TypeFloatMax TypeFloatMin TypeFloatMoney TypeFloatSeparator TypeFloatSepChar TypeHAlign TypeVAlign
SSSetTypeInteger	TypeIntegerMax TypeIntegerMin
SSSetTypeIntegerExt	TypeHAlign TypeIntegerMax TypeIntegerMin TypeIntegerSpinInc TypeIntegerSpinWrap TypeSpin TypeVAlign
SSSetTypeOwnerDraw	TypeOwnerDrawStyle
SSSetTypePic	TypeHAlign TypePicMask TypeVAlign
SSSetTypePicture	TypeHAlign TypePictCenter TypePictMaintainScale TypePictPicture TypePictStretch TypeVAlign
SSSetTypePictureHandle	TypePictCenter TypePictMaintainScale TypePictPicture TypePictStretch
SSSetTypeStaticText	TypeHAlign TypeTextPrefix TypeTextShadow TypeTextShadowIn TypeTextWordWrap TypeVAlign

## Appendix E

DLL Function	Property or Function
SSSetTypeTime	TypeHAlign TypeSpin TypeTime24Hour TypeTimeMax TypeTimeMin TypeTimeSeconds TypeTimeSeparator TypeVAlign
SSSetUnitType	UnitType
SSSetUserData	SetItemData
SSSetUserResize	UserResize
SSSetUserResizeCol	UserResizeCol
SSSetUserResizeRow	UserResizeRow
SSSetValue	Value
SSSetValueRange	Value
SSShowActiveCell	Position
SSShowCell	Action (1) LeftCol Position TopRow
SSShowCol	ColHidden
SSShowRow	RowHidden
SSSort	Action (25) SortBy SortKey SortKeyOrder
SSSortEx	Action (25) SortBy SortKey SortKeyOrder
SSSwapRange	Action (21)
SSValidateFormula	IsFormulaValid
SSVGetBufferSize	VirtualOverlap VirtualRows
SSVGetMax	VirtualMaxRows
SSVGetPhysBufferSize	VirtualCurRowCount
SSVGetPhysBufferTop	VirtualCurTop
SSVGetStyle	VirtualScrollBuffer
SSVRefreshBuffer	Action (30)
SSVScrollGetSpecial	VScrollSpecialType

DLL Function	Property or Function
SSVScrollSetSpecial	VScrollSpecialType
SSVSetBufferSize	VirtualOverlap VirtualRows
SSVSetMax	VirtualMaxRows
SSVSetStyle	VirtualScrollBuffer



# Appendix F Spread Designer Cross Reference

## Menus and Menu Items

### File Menu

Menu Item	Displays	Items in Dialog Box	Description
Print Preview			Print view of the active spreadsheet
Print	Print Active View dialog	Option buttons under Range Pages box and To box under Range Orientation group box Left box under Margins Right box under Margins Top box under Margins Bottom box under Margins Option buttons under Units Page Order group box Data Cells Only check box under Print Column Headers check box under Print Row Headers check box under Print Grid check box under Print Border check box under Print Shadows check box under Print Color check box under Print Job Name box under Job Setup Header Text box under Job Setup Footer Text box under Job Setup	Applies settings to current view only.

## Appendix F

Menu Item	Displays	Items in Dialog Box	Description
Print ( <i>continued</i> )	Print from Active View dialog	Abort Message box under Job Setup Save settings check box	Applies settings to current view only.
Apply			Applies settings depending on the following choices and the setting in the View Applied to Control group box in the General dialog.
All (Both Data and Format)			Changes made to the data and format in the Spread designer are applied back to the fpSpread control.
Format			Changes made to the format in the Spread designer are applied back to the fpSpread control.
Data			Changes made to the data in the Spread designer are applied back to the fpSpread control.
Apply & Exit			Changes made to the data and format in the Spread designer are applied back to the fpSpread control and returns you to the fpSpread control.
Exit Spread Designer			Returns to the selected fpSpread control.

## Edit Menu

Menu Item	Displays	Items in Dialog Box	Description
Clear			
Clear All			Clears data and formatting from the selected cells.
Clear Data Only			Clears data only from the selected cells.
Insert			
Column			Column is added to the left of the current column. Data in the current column shifts right.
Row			Row is added above the current row. Data in the current row shifts down.
Delete			
Column			Selected column is deleted.
Row			Selected row is deleted.

## Display Menu

Menu Item	Displays	Items in Dialog Box	Associated Property, Function, Subroutine
Scroll Bars	Scroll Bars tab in the Display dialog	Display group box	ScrollBars property
		Scroll Box Track group box	ScrollBarTrack property
		Display Only if Needed check box	ScrollBarExtMode property
		Align at Last Row and Column check box	ScrollBarMaxAlign property
		Scroll Box Reflects Max Rows check box	ScrollBarShowMax property
Grid Lines	Grid Lines tab in the Display dialog	Show Vertical Lines check box under Settings	GridShowVert property
		Show Horizontal Lines check box under Settings	GridShowHoriz property
		Make Grid Lines Solid check box under Settings	GridSolid property
		Back Color Displays group box	BackColorStyle property
Pointer	Pointer tab in the Display dialog	Pointer Location drop-down list box	CursorType property
		Pointer Appearance combo box	CursorStyle property
Text Tip	Text Tip tab in the Display dialog	Display drop-down list box	TextTipDelay property
		Delay spin box	TextTip property
		BackColor in the drop-down list box under Color	SetTextTipAppearance function/method <i>TipBackColor</i> parameter
		ForeColor in the drop-down list box under Color	SetTextTipAppearance function/method <i>TipForeColor</i> parameter
		Font button	SetTextTipAppearance function/method <i>FontName</i> , <i>FontSize</i> , <i>FontBold</i> , <i>FontItalic</i> parameters

## Cell Menu

Menu Item	Displays	Items in Dialog Box	Associated Property
Cell Type			
Button	Settings tab in the Cell Type dialog with the Cell Type drop-down list box set to Button	Cell Type drop-down list box	CellType
		Text box	TypeButtonText
		Shadow spin box	TypeButtonShadowSize
		Style drop-down list box	TypeButtonType
		Text Pos drop-down list box	TypeButtonAlign
		BorderColor in the drop-down list box under Color	TypeButtonBorderColor
		ButtonColor in the drop-down list box under Color	TypeButtonColor
		DarkColor in the drop-down list box under Color	TypeButtonDarkColor
		LightColor in the drop-down list box under Color	TypeButtonLightColor
		TextColor in the drop-down list box under Color	TypeButtonTextColor
		Picture Up in the drop-down list box under Picture	TypeButtonPicture
Picture Down in the drop-down list box under Picture	TypeButtonPictureDown		
Check Box	Settings tab in the Cell Type dialog with the Cell Type drop-down list box set to Check Box	Cell Type drop-down list box	CellType
		Text box	TypeCheckText
		Text Left of Picture check box	TypeCheckTextAlign
		Type group box	TypeCheckType
		Picture group box	TypeCheckPicture
		Horizontal Alignment group box	TypeHAlign
		Vertical Alignment group box	TypeVAlign

Menu Item	Displays	Items in Dialog Box	Associated Property
Combo Box	Settings tab in the Cell Type dialog with the Cell Type drop-down list box set to Combo Box	Cell Type drop-down list box	CellType
		Text box and Combo Box List group box	TypeComboBoxList
			TypeComboBoxString
		Editable check box	TypeComboBoxEditable
		Text Length spin box	TypeMaxEditLen
		Max Drop spin box	TypeComboBoxMaxDrop
		List Width drop-down combo box	TypeComboBoxWidth
		Horizontal Alignment group box	TypeHAlign
		Vertical Alignment group box	TypeVAlign
Date	Settings tab in the Cell Type dialog with the Cell Type drop-down list box set to Date	Cell Type drop-down list box	CellType
		Format drop-down list box	TypeDateFormat
		Min Date box	TypeDateMin
		Max Date box	TypeDateMax
		Separator box	TypeDateSeparator
		Display Century check box	TypeDateCentury
		Display Spin Button check box	TypeSpin
		Horizontal Alignment group box	TypeHAlign
		Vertical Alignment group box	TypeVAlign

## Appendix F

Menu Item	Displays	Items in Dialog Box	Associated Property
Edit	Settings tab in the Cell Type dialog with the Cell Type drop-down list box set to Edit	Cell Type drop-down list box	CellType
		Character Set drop-down list box	TypeEditCharSet
		Text Case group box	TypeEditCharCase
		Multiple Lines check box	TypeEditMultiLine
		Password check box	TypeEditPasword
		Length spin box	TypeMaxEditLen
		Horizontal Alignment group box	TypeHAlign
		Vertical Alignment group box	TypeVAlign
Float	Settings tab in the Cell Type dialog with the Cell Type drop-down list box set to Float	Cell Type drop-down list box	CellType
		Min Value box	TypeFloatMin
		Max Value box	TypeFloatMax
		Integer Digits spin box	
		Decimal Digits spin box	TypeFloatDecimalPlaces
		Money check box	TypeFloatMoney
		Money box	TypeFloatCurrencyChar
		Separator check box	TypeFloatSeparator
		Separator box	TypeFloatSepChar
		Decimal Box	TypeFloatDecimalChar
		Horizontal Alignment group box	TypeHAlign
		Vertical Alignment group box	TypeVAlign

Menu Item	Displays	Items in Dialog Box	Associated Property
Integer	Settings tab in the Cell Type dialog with the Cell Type drop-down list box set to Integer	Cell Type drop-down list box	CellType
		Min Value box	TypeIntegerMin
		Max Value box	TypeIntegerMax
		Horizontal Alignment group box	TypeHAlign
		Vertical Alignment group box	TypeVAlign
		Display Spin Button check box under Spin Settings	TypeSpin
		Increment spin box under Spin Settings	TypeIntegerSpinInc
		Spin Wrap check box under Spin Settings	TypeIntegerSpinWrap
Label(Static)	Settings tab in the Cell Type dialog with the Cell Type drop-down list box set to Label(Static)	Cell Type drop-down list box	CellType
		Text group box	Text
		Raised check box under Shadow	TypeTextShadow
		Lowered check box under Shadow	TypeTextShadowIn
		Horizontal Alignment group box	TypeHAlign
		Vertical Alignment group box	TypeVAlign
		Display Mnemonic check box	TypeTextPrefix
		Word Wrap check box	TypeTextWordWrap
Mask(Pic)	Settings tab in the Cell Type dialog with the Cell Type drop-down list box set to Mask(Pic)	Cell Type drop-down list box	CellType
		Mask Definition box	TypePicMask
		Horizontal Alignment group box	TypeHAlign
		Vertical Alignment group box	TypeVAlign

## Appendix F

Menu Item	Displays	Items in Dialog Box	Associated Property
Owner Draw	Settings tab in the Cell Type dialog with the Cell Type drop-down list box set to OwnerDraw	Cell Type drop-down list box	CellType
		Owner Draw ID spin box	TypeOwnerDrawStyle
Picture	Settings tab in the Cell Type dialog with the Cell Type drop-down list box set to Picture	Cell Type drop-down list box	CellType
		Picture button	TypePictPicture
		Stretch check box under Appearance	TypePictStretch
		Maintain Scale check box under Appearance	TypePictMaintainScale
		Horizontal Alignment group box	TypeHAlign
		Vertical Alignment group box	TypeVAlign
Time	Settings tab in the Cell Type dialog with the Cell Type drop-down list box set to Time	Cell Type drop-down list box	CellType
		Format drop-down list box	TypeTime24Hour
		Min Time box	TypeTimeMn
		Max Time box	TypeTimeMax
		Separator box	TypeTimeSeparator
		Display Seconds check box	TypeTimeSeconds
		Display Spin Button check box	TypeSpin
		Horizontal Alignment group box	TypeHAlign
		Vertical Alignment group box	TypeVAlign
Border Lines	Border Lines tab in the Cell Settings dialog	Border Around group box	CellBorderType
		Border Side drop-down list box and Color button	CellBorderColor
		Border Side drop-down list box and Style group box	CellBorderStyle
Formula	Formula tab in the Cell Settings dialog	Formula box	Formula
		Insert drop-down list box	Inserts one of the available formula functions as described in Appendix G, “Formula Operators and Functions.”

Menu Item	Displays	Items in Dialog Box	Associated Property
Font	Font dialog	Font combo box	FontName
		Font Style drop-down list box	FontBold
			FontItalic
		Size drop-down combo box	FontSize
		Strikeout check box under Effects	FontStrikethru
Underline check box under Effects	FontUnderline		
Cell Colors	Colors tab in the Cell Settings dialog	Property drop-down list box and Color button	BackColor
			ForeColor
Lock			Lock
Unlock			Lock

## General Menu

Menu Item	Displays	Items in Dialog Box	Associated Property, Function
Spreadsheet Environment	General tab in the Sheet Environment Settings dialog	Make Edit Mode Permanent check box	EditModePermanent property
		Replace Existing Text check box	EditModeReplace property
		Handle Clipboard Keys check box	AutoClipboard property
		Use Arrow Keys to Exit Edit check box	ArrowsExitEditMode property
		Retain Selected Block check box	RetainSelBlock property
		Protect Locked Cells check box	Protect property
		Multiple Block Selections check box	AllowMultiBlocks property
		On Focus Set Cell to Cursor check box	MoveActiveOnFocus property
		Clipboard Options drop-down list box	ClipboardOptions property
		Appearance drop-down list box	Appearance property
		Cell Overflow check box	AllowCellOverflow property
		Beep Off check box	NoBeep property
		Cell Dragdrop check box	AllowDragDrop property
		Hide Border check box	NoBorder property
		Process Tab Key check box	ProcessTab property
		Autosize On check box	AutoSize property
		Calculate Formulas check box	AutoCalc property
		Custom Formulas check box	AllowUserFormulas property
		Formula Sync check box	FormulaSync property
Pivot Year spin box	TwoDigitYearMax property		

Menu Item	Displays	Items in Dialog Box	Associated Property, Function
Spreadsheet Environment (continued)	SheetMode tab in the Sheet Environment Settings dialog	Spreadsheet Operation group box	OperationMode property
	BlockMode tab in the Sheet Environment Settings dialog	User Can Select group box	SelectBlockOptions property
	Enter Key Action tab in the Sheet Environment Settings dialog	Moves Active Cell group box	EditEnterAction property
	Button Display tab in the Sheet Environment Settings dialog	Display Buttons group box	ButtonDrawMode property
	Action Keys tab in the Sheet Environment Settings dialog	Action drop-down list box	SetActionKey function/method <i>wAction</i> parameter
		Key Used group box	SetActionKey function/method <i>wKey</i> parameter
Capture Key Combination button		SetActionKey function/method <i>wKey</i> parameter	

## Appendix F

Menu Item	Displays	Items in Dialog Box	Associated Property, Function
Headers	Columns tab in the Headers dialog	Display Headers check box under Settings	DisplayColHeaders property
		User Resize check box under Settings	UserResize property
		Restrict Columns check box under Settings	RestrictCols property
		User Defined Headers group box	Text property
		Start Col Num spin box under Headers	StartingColNumber property
		Header Styles drop-down list box under Headers	ColHeaderDisplay property
	Rows tab in the Headers dialog	Display Headers check box under Settings	DisplayRowHeaders property
		User Resize check box under Settings	UserResize property
		Restrict Rows check box under Settings	RestrictRows property
		User Defined Headers group box	Text property
		Start Row Num spin box under Headers	StartingRowNumber property
		Header Styles drop-down list box under Headers	RowHeaderDisplay property

Menu Item	Displays	Items in Dialog Box	Associated Property, Function
Columns and Rows	Columns tab in the Columns / Rows dialog	Max Cols box under Settings	MaxCols property
		Frozen Cols spin box under Settings	ColsFrozen property
		Visible Cols spin box under Settings	VisibleCols property
		Left Col spin box under Settings	LeftCol property
		Columns Hidden check box	ColHidden property
		Column Page Break check box	ColPageBreak property
		Col Width box under Settings	ColWidth property
		Resizable group box	UserResizeCol property
	Rows tab in the Columns / Rows dialog	Max Rows box under Settings	MaxRows property
		Frozen Rows spin box under Settings	RowsFrozen property
		Visible Rows spin box under Settings	VisibleRows property
		Top Row spin box under Settings	TopRow property
		Row Hidden check box	RowHidden property
		Row Page Break check box	RowPageBreak property
		Row Height box under Settings	RowHeight property
Resizable group box	UserResizeRow property		

## Appendix F

Menu Item	Displays	Items in Dialog Box	Associated Property, Function
Print Properties	Print Properties dialog	Option buttons under Range	PrintType property
		Pages box under Range	PrintPageStart property
		To box under Range	PrintPageEnd property
		Orientation group box	PrintOrientation property
		Left box under Margins	PrintMarginLeft property
		Right box under Margins	PrintMarginRight property
		Top box under Margins	PrintMarginTop property
		Bottom box under Margins	PrintMarginBottom property
		Units group box	
		Page Order group box	PrintPageOrder property
		Data Cells Only check box under Print	PrintUseDataMax property
		Column Headers check box under Print	PrintColHeaders property
		Row Headers check box under Print	PrintRowHeaders property
		Grid check box under Print	PrintGrid property
		Border check box under Print	PrintBorder property
		Shadows check box under Print	PrintShadows property
		Color check box under Print	PrintColor property
		Job Name box under Job Setup	PrintJobName property
		Header Text box under Job Setup	PrintHeader property
		Footer Text box under Job Setup	PrintFooter property
Abort Message box under Job Setup	PrintAbortMsg property		
Initial Page Number	PrintFirstPageNumber		

Menu Item	Displays	Items in Dialog Box	Associated Property, Function
Default Characters	Characters tab in the Default dialog	Currency Char box under Settings	FloatDefCurrencyChar property
		Decimal Char box under Settings	FloatDefDecimalChar property
		Separator Char box under Settings	FloatDefSepChar property
Spreadsheet Colors	Colors tab in the Default dialog	Property drop-down list box and Color button	GrayAreaBackColor property
			GridColor property
			LockBackColor property
			LockForeColor property
			SelBackColor property
			SelForeColor property
			SetOddEvenRowColor function/method <i>clrBackOdd, clrForeOdd, clrBackEven, clrForeEven</i> parameters
			ShadowColor property
ShadowDark property			
ShadowText property			
Unit Type	UnitType tab in the Default dialog	Settings group box	UnitType property
Spreadsheet Sort	Sort Settings dialog	Sort by group box	SortBy property
		Column or Row spin box under Settings Key <i>n</i>	SortKey property
		Option buttons under Settings Key <i>n</i>	SortKeyOrder property

## Appendix F

Menu Item	Displays	Items in Dialog Box	Associated Property, Function
Virtual Mode	Virtual Mode dialog	Virtual Mode check box under Settings	VirtualMode
		Virtual Max Rows box under Settings	VirtualMaxRows
		Memory Max Rows box under Settings	VirtualOverlap
		Rows Per Read box under Settings	VirtualRows
		Use Special Scroll bar check box under Virtual Scroll Bar	VScrollSpecial
		No Home/End under Virtual Scroll Bar	VScrollSpecialType
		No Page Up/Down under Virtual Scroll Bar	
		No Line Up/Down under Virtual Scroll Bar	
Thumb Reflects Row in Buffer under Virtual Scroll Bar	VirtualScrollBuffer		

Menu Item	Displays	Items in Dialog Box	Associated Property, Function
Data Binding	Data Binding dialog	Database Column (DataField) drop-down list box and Set Data Field and Clear Data Field buttons under Selected Column Data Binding Information	DataField
		Fire Data Fill Event for Column check box under Selected Column Data Binding Information	DataFillEvent
		Set Cell Type Automatically check box	DAutoCellTypes
		Automatically Fill Data check box	DAutoFill
		Inform Data Control on Active Row Changes check box	DInformActiveRowChange
		Automatically Save Data check box	DAutoSave
		Use DataField Name As Column Heading check box	DAutoHeadings
		Size Column To Data drop-down list box	DAutoSizeCols

## Calculation Menu

Menu Item	Displays	Items in Dialog Box	Associated Function
Custom Name	Custom Name tab in the Calculation dialog	Name drop-down combo box Value box Add and Delete buttons	SetCustomName function
Iteration	Iteration tab in the Calculation dialog	Allow Iterations check box Maximum Iterations box	SetIteration function <i>Iteration</i> parameter SetIteration function <i>MaxIterations</i> parameter
Reference Style	Reference Style tab in the Calculation dialog	Reference Style group box	SetRefStyle function

## Window Menu

Menu Item	Displays	Items in Dialog Box	Description
Cascade			Arranges the windows so they overlap
Tile Horizontal			Arranges the windows as non-overlapping horizontal tiles
Tile Vertical			Arranges the windows as non-overlapping vertical tiles
Arrange Icons			Arranges icons at the bottom of the window
Clipboard Viewer			Displays code generated in Learn Mode
Toolbar			Displays or hides the feature toolbar
Cell Type Bar			Displays or hides the cell type toolbar
Status Bar			Displays or hides the status bar
Balloons			Displays or hides the toolbar buttons' balloons
(List of available windows)			Available active windows

## Mode Menu

Menu Item	Displays	Items in Dialog Box	Description
Design Mode			Sets the Spread Designer to design mode
Run-Time Mode			Sets the Spread Designer to run-time mode

<b>Menu Item</b>	<b>Displays</b>	<b>Items in Dialog Box</b>	<b>Description</b>
Learn Mode On			Turns on Learn Mode and begins generating code
Learn Mode Off			Turns off Learn Mode and writes generated code to the clipboard
Cell Type Mode On			Turns on cell-type identification mode
Cell Type Mode Off			Turns off cell-type identification mode

## Settings Menu

Menu Item	Displays	Items in Dialog Box	Description
General	General dialog	Verify When Closing Window without Saving/Applying check box	Displays a dialog box that prompts you to apply your changes when you exit the Spread Designer
		Create a Log File when Importing/Exporting Files check box	Specifies whether to create a log file when importing Excel-formatted files or when exporting HTML or Excel-formatted files
		Create a Log File when Importing/Exporting Files box	Path and file name of the log file to create
		Type Applied by Exit Verification group box	Specifies what is applied at exit verification
		View Applied to Control group box	Specifies the view to apply to the selected control when applying changes
		Display Balloons check box under Balloon Settings	Specifies whether the balloon help is displayed for the feature and cell type toolbar buttons
		Shape group box under Balloon Settings	Customizes the shape of the balloon
		Tail Type group box under Balloon Settings	Customizes the tail of the balloon
		Shadow group box under Balloon Settings	Customizes the shadow behind the balloon
Learn Mode	Learn Mode Settings dialog	Form Name box under Object Information	Specifies the name of the form containing the fpSpread control
		Control Name box under Object Information	Specifies the name of the fpSpread control
		Data Source Name box under Object Information	If the fpSpread control is data bound, specifies the name of the data control
		Add Comments check box	Specifies whether comments are added to the generated code
		Display Message When Data Is Written to Clipboard check box	Specifies whether a message is displayed to confirm when code is written to the Clipboard

## Toolbars and Buttons

Button	Displays . . .	Properties set in dialog box
	Scroll Bars tab in the Display dialog	ScrollBarExtMode ScrollBarMaxAlign ScrollBars ScrollBarShowMax ScrollBarTrack
	Grid Lines tab in the Display dialog	BackColorStyle GridShowHoriz GridShowVert GridSolid
	Pointer tab in the Display dialog	CursorStyle CursorType
	Text Tip tab in the Display dialog	SetTextTipAppearance function/method TextTip property TextTipDelay property
	Border Lines tab in the Cell Settings dialog	CellBorderColor CellBorderStyle CellBorderType
	Formula tab in the Cell Settings dialog	Formula
	Colors tab in the Cell Settings dialog	BackColor ForeColor

## Appendix F

Button	Displays . . .	Properties set in dialog box
	General tab in the Sheet Environment Settings dialog	AllowCellOverflow AllowDragDrop AllowMultiBlocks AllowUserFormulas Appearance ArrowsExitEditMode AutoCalc AutoClipboard AutoSize ClipboardOptions EditModePermanent EditModeReplace FormulaSync MoveActiveOnFocus NoBeep NoBorder ProcessTab Protect RetainSelBlock TwoDigitYearMax
	SheetMode tab in the Sheet Environment Settings dialog	OperationMode
	BlockMode tab in the Sheet Environment Settings dialog	SelectBlockOptions
	Enter Key Action tab in the Sheet Environment Settings dialog	EditEnterAction
	Button Display tab in the Sheet Environment Settings dialog	ButtonDrawMode
	Action Keys tab in the Sheet Environment Settings dialog	SetActionKey function/method
	Columns tab in the Headers dialog	ColHeaderDisplay DisplayColHeaders RestrictCols StartingColNumber Text UserResize
	Rows tab in the Headers dialog	DisplayRowHeaders RestrictRows RowHeaderDisplay StartingRowNumber Text UserResize

Button	Displays . . .	Properties set in dialog box
	Columns tab in the Columns / Rows dialog	ColHidden ColPageBreak ColsFrozen ColWidth LeftCol MaxCols UserResizeCol VisibleCols
	Rows tab in the Columns / Rows dialog	MaxRows RowHeight RowHidden RowPageBreak RowsFrozen TopRow UserResizeRow VisibleRows
	Colors tab in the Default dialog	GrayAreaBackColor GridColor LockBackColor LockForeColor SelBackColor SelForeColor SetOddEvenRowColor function/method ShadowColor ShadowDark ShadowText
	Virtual Mode dialog	VirtualMaxRows VirtualMode VirtualOverlap VirtualRows VirtualScrollBuffer VScrollSpecial VScrollSpecialType

## Appendix F

Button	Displays . . .	Properties set in dialog box
	Cell Type dialog with the Cell Type drop-down list box set to Button	CellType TypeButtonAlign TypeButtonBorderColor TypeButtonColor TypeButtonDarkColor TypeButtonLightColor TypeButtonPicture TypeButtonPictureDown TypeButtonShadowSize TypeButtonText TypeButtonTextColor TypeButtonType
	Cell Type dialog with the Cell Type drop-down list box set to Check Box	CellType TypeCheckPicture TypeCheckText TypeCheckTextAlign TypeCheckType TypeHAlign TypeVAlign
	Cell Type dialog with the Cell Type drop-down list box set to Combo Box	CellType TypeComboBoxEditable TypeComboBoxList TypeComboBoxMaxDrop TypeComboBoxString TypeComboBoxWidth TypeHAlign TypeMaxEditLen TypeVAlign
	Cell Type dialog with the Cell Type drop-down list box set to Date	CellType TypeDateCentury TypeDateFormat TypeDateMax TypeDateMin TypeDateSeparator TypeHAlign TypeSpin TypeVAlign

Button	Displays . . .	Properties set in dialog box
	Cell Type dialog with the Cell Type drop-down list box set to Edit	CellType TypeEditCharCase TypeEditCharSet TypeMaxEditLen TypeEditMultiLine TypeEditPassword TypeHAlign TypeVAlign
	Cell Type dialog with the Cell Type drop-down list box set to Float	CellType TypeFloatCurrencyChar TypeFloatDecimalChar TypeFloatDecimalPlaces TypeFloatMax TypeFloatMin TypeFloatMoney TypeFloatSeparator TypeFloatSepChar TypeHAlign TypeVAlign
	Cell Type dialog with the Cell Type drop-down list box set to Integer	CellType TypeHAlign TypeIntegerMax TypeIntegerMin TypeIntegerSpinInc TypeIntegerSpinWrap TypeSpin TypeVAlign
	Cell Type dialog with the Cell Type drop-down list box set to Label(Static)	CellType Text TypeHAlign TypeTextPrefix TypeTextShadow TypeTextShadowIn TypeTextWordWrap TypeVAlign
	Cell Type dialog with the Cell Type drop-down list box set to Mask(Pic)	CellType TypeHAlign TypePicMask TypeVAlign
	Cell Type dialog with the Cell Type drop-down list box set to OwnerDraw	CellType TypeOwnerDrawStyle

## Appendix F

Button	Displays . . .	Properties set in dialog box
	Cell Type dialog with the Cell Type drop-down list box set to Picture	CellType TypeHAlign TypePictMaintainScale TypePictPicture TypePictStretch TypeVAlign
	Cell Type dialog with the Cell Type drop-down list box set to Time	CellType TypeHAlign TypeSpin TypeTime24Hour TypeTimeMax TypeTimeMin TypeTimeSeconds TypeTimeSeparator TypeVAlign

## Properties Cross-Referenced to Buttons and Menus

Property	Button to display dialog box	Menu item to display dialog box	To set the property in the dialog box
About	N/A	N/A	N/A
Action	N/A	N/A	Property is set in generated code when performing certain actions in Learn Mode
ActiveCol	N/A	N/A	N/A
ActiveRow	N/A	N/A	N/A
AllowCellOverflow		From the General menu, choose Spreadsheet Environment	Cell Overflow check box under Settings on the General tab in the Sheet Environment Settings dialog box
AllowDragDrop		From the General menu, choose Spreadsheet Environment	Cell Dragdrop check box under Settings on the General tab in the Sheet Environment Settings dialog box
AllowMultiBlocks		From the General menu, choose Spreadsheet Environment	Multiple Block Selections check box under Settings on the General tab in the Sheet Environment Settings dialog box
AllowUserFormulas		From the General menu, choose Spreadsheet Environment	Custom Formulas check box under Settings on the General tab in the Sheet Environment Settings dialog box

Property	Button to display dialog box	Menu item to display dialog box	To set the property in the dialog box
Appearance		From the General menu, choose Spreadsheet Environment	Appearance drop-down list box under Settings on the General tab in the Sheet Environment Settings dialog box
ArrowsExitEditMode		From the General menu, choose Spreadsheet Environment	Use ArrowKeys to Exit Edit check box under Settings on the General tab in the Sheet Environment Settings dialog box
AutoCalc		From the General menu, choose Spreadsheet Environment	Calculate Formulas check box under Settings on the General tab in the Sheet Environment Settings dialog box
AutoClipboard		From the General menu, choose Spreadsheet Environment	Handle Clipboard Keys check box under Settings on the General tab in the Sheet Environment Settings dialog box
AutoSize		From the General menu, choose Spreadsheet Environment	Autosize On check box under Settings on the General tab in the Sheet Environment Settings dialog box
BackColor		From the Cell menu, choose Cell Colors	Select BackColor from the Property drop-down list box on the Colors tab in the Cell Settings dialog box
BackColorStyle		From the Display menu, choose Grid Lines	Option buttons under Back Color Displays on the Grid Lines tab in the Display dialog box
BlockMode	N/A	N/A	Property is set in generated code when performing certain actions in Learn Mode
ButtonDrawMode		From the General menu, choose Spreadsheet Environment	Check boxes under Display Buttons on the Button Display tab in the Sheet Environment Settings dialog box
CellBorderColor		From the Cell menu, choose Border Lines	Border Side drop-down list box and Color button on the Border Lines tab in the Cell Settings dialog box
CellBorderStyle		From the Cell menu, choose Border Lines	Border Side drop-down list box and option buttons and check box under Style on the Border Lines tab in the Cell Settings dialog box
CellBorderType		From the Cell menu, choose Border Lines	Option buttons under Border Around on the Border Lines tab in the Cell Settings dialog box
CellType		From the Cell menu, choose Cell Type then select the appropriate item	Cell Type drop-down list box in the Cell Type dialog box

## Appendix F

Property	Button to display dialog box	Menu item to display dialog box	To set the property in the dialog box
ChangeMade	N/A	N/A	N/A
Clip	N/A	N/A	N/A
ClipboardOptions		From the General menu, choose Spreadsheet Environment	Clipboard Options drop-down list box under Settings on the General tab in the Sheet Environment Settings dialog box
ClipValue	N/A	N/A	N/A
Col	N/A	N/A	Property is set in generated code when performing certain actions in Learn Mode
Col2	N/A	N/A	Property is set in generated code when performing certain actions in Learn Mode
ColHeaderDisplay		From the General menu, choose Headers	Header Styles drop-down list box under Headers on the Columns tab in the Headers dialog box
ColHidden		From the General menu, choose Columns and Rows	Columns Hidden check box on the Columns tab in the Columns / Rows dialog box
ColPageBreak		From the General menu, choose Columns and Rows	Column Page Break check box on the Columns tab in the Columns / Rows dialog box
ColsFrozen		From the General menu, choose Columns and Rows	Frozen Cols spin box under Settings on the Columns tab in the Columns / Rows dialog box
ColWidth		From the General menu, choose Columns and Rows	Col Width box on the Columns tab in the Columns / Rows dialog box
CursorIcon	N/A	N/A	N/A
CursorStyle		From the Display menu choose Pointer	Pointer Appearance combo box on the Pointer tab in the Display dialog box
CursorType		From the Display menu choose Pointer	Pointer Location drop-down list box on the Pointer tab in the Display dialog box
DataChanged	N/A	N/A	N/A
DataColCnt	N/A	N/A	N/A
DataField	N/A	From the General menu, choose Data Binding	Database Column (DataField) drop-down list box and Set Data Field and Clear Data Field buttons under Selected Column Data Binding Information in the Data Binding dialog box

Property	Button to display dialog box	Menu item to display dialog box	To set the property in the dialog box
DataFillEvent	N/A	From the General menu, choose Data Binding	Fire Data Fill Event for Column check box under Selected Column Data Binding Information in the Data Binding dialog box
DataRowCnt	N/A	N/A	N/A
DAutoCellTypes	N/A	From the General menu, choose Data Binding	Set Cell Type Automatically check box in the Data Binding dialog box
DAutoFill	N/A	From the General menu, choose Data Binding	Automatically Fill Data check box in the Data Binding dialog box
DAutoHeadings	N/A	From the General menu, choose Data Binding	Use DataField Name As Column Heading check box in the Data Binding dialog box
DAutoSave	N/A	From the General menu, choose Data Binding	Automatically Save Data check box in the Data Binding dialog box
DAutoSizeCols	N/A	From the General menu, choose Data Binding	Size Column to Data drop-down list box in the Data Binding dialog box
DestCol	N/A	N/A	N/A
DestRow	N/A	N/A	N/A
DInformActiveRowChange	N/A	From the General menu, choose Data Binding	Inform Data Control on Active Row Changes check box in the Data Binding dialog box
DisplayColHeaders		From the General menu, choose Headers	Display Headers check box under Settings on the Columns tab in the Headers dialog box
DisplayRowHeaders		From the General menu, choose Headers	Display Headers check box under Settings on the Rows tab in the Headers dialog box
EditEnterAction		From the General menu, choose Spreadsheet Environment	Option buttons under Moves Active Cell on the Enter Key Action tab in the Sheet Environment Settings dialog box
EditMode	N/A	N/A	N/A
EditModePermanent		From the General menu, choose Spreadsheet Environment	Make Edit Mode Permanent check box under Settings on the General tab in the Sheet Environment Settings dialog box
EditModeReplace		From the General menu, choose Spreadsheet Environment	Replace Existing Text check box under Settings on the General tab in the Sheet Environment Settings dialog box
FileNum	N/A	N/A	N/A
FloatDefCurrencyChar	N/A	From the General menu, choose Default Characters	Currency Char box under Settings on the Characters tab in the Default dialog box

## Appendix F

Property	Button to display dialog box	Menu item to display dialog box	To set the property in the dialog box
FloatDefDecimalChar	N/A	From the General menu, choose Default Characters	Decimal Char box under Settings on the Characters tab in the Default dialog box
FloatDefSepChar	N/A	From the General menu, choose Default Characters	Separator Char box under Settings on the Characters tab in the Default dialog box
Font	N/A	From the Cell menu, choose Font	Font dialog box
FontBold	N/A	From the Cell menu, choose Font	Font Style drop-down list box in the Font dialog box
FontItalic	N/A	From the Cell menu, choose Font	Font Style drop-down list box in the Font dialog box
FontName	N/A	From the Cell menu, choose Font	Font combo box in the Font dialog box
FontSize	N/A	From the Cell menu, choose Font	Size drop-down combo box in the Font dialog box
FontStrikethru	N/A	From the Cell menu, choose Font	Strikeout check box under Effects in the Font dialog box
FontUnderline	N/A	From the Cell menu, choose Font	Underline check box under Effects in the Font dialog box
ForeColor		From the Cell menu, choose Cell Colors	Select ForeColor from the Property drop-down list box on the Colors tab in the Cell Settings dialog box
Formula		From the Cell menu, choose Formula	Formula box on the Formula tab in the Cell Settings dialog box
FormulaSync		From the General menu, choose Spreadsheet Environment	Formula Sync check box under Settings on the General tab in the Sheet Environment Settings dialog box
GrayAreaBackColor		From the General menu, choose Spreadsheet Colors	Select GrayAreaBackColor from the Property drop-down list box on the Colors tab in the Default dialog box
GridColor		From the General menu, choose Spreadsheet Colors	Select GridColor from the Property drop-down list box on the Colors tab in the Default dialog box
GridShowHoriz		From the Display menu, choose Grid Lines	Show Horizontal Lines check box under Settings on the Grid Lines tab in the Display dialog box

Property	Button to display dialog box	Menu item to display dialog box	To set the property in the dialog box
GridShowVert		From the Display menu, choose Grid Lines	Show Vertical Lines check box under Settings on the Grid Lines tab in the Display dialog box
GridSolid		From the Display menu, choose Grid Lines	Make Grid Lines Solid check box under Settings on the Grid Lines tab in the Display dialog box
hDCPrinter	N/A	N/A	N/A
IsBlockSelected	N/A	N/A	N/A
LeftCol		From the General menu, choose Columns and Rows	Left Col spin box under Settings on the Columns tab in the Columns / Rows dialog box
Lock	N/A	From the Cell menu, choose Lock or Unlock	N/A
LockBackColor		From the General menu, choose Spreadsheet Colors	Select LockBackColor from the Property drop-down list box on the Colors tab in the Default dialog box
LockForeColor		From the General menu, choose Spreadsheet Colors	Select LockForeColor from the Property drop-down list box on the Colors tab in the Default dialog box
MaxCols		From the General menu, choose Columns and Rows	Max Cols box under Settings on the Columns tab in the Columns / Rows dialog box
MaxRows		From the General menu, choose Columns and Rows	Max Rows box under Settings on the Rows tab in the Columns / Rows dialog box
MaxTextCellHeight	N/A	N/A	N/A
MaxTextCellWidth	N/A	N/A	N/A
MaxTextColWidth	N/A	N/A	N/A
MaxTextRowHeight	N/A	N/A	N/A
MoveActiveOnFocus		From the General menu, choose Spreadsheet Environment	On Focus Set Cell to Cursor check box under Settings on the General tab in the Sheet Environment Settings dialog box
MultiSelCount	N/A	N/A	N/A
MultiSelIndex	N/A	N/A	N/A
NoBeep		From the General menu, choose Spreadsheet Environment	Beep Off check box under Settings on the General tab in the Sheet Environment Settings dialog box

## Appendix F

Property	Button to display dialog box	Menu item to display dialog box	To set the property in the dialog box
NoBorder		From the General menu, choose Spreadsheet Environment	Hide Border check box under Settings on the General tab in the Sheet Environment Settings dialog box
OLEDropMode	N/A	N/A	N/A
OperationMode		From the General menu, choose Spreadsheet Environment	Option buttons under Spreadsheet Operation on the SheetMode tab in the Sheet Environment Settings dialog box
Position	N/A	N/A	N/A
PrintAbortMsg	N/A	From the General menu, choose Print Properties	Abort Message box under Job Setup in the Print Properties dialog box
PrintBorder	N/A	From the General menu, choose Print Properties	Border check box under Print in the Print Properties dialog box
PrintColHeaders	N/A	From the General menu, choose Print Properties	Column Headers check box under Print in the Print Properties dialog box
PrintColor	N/A	From the General menu, choose Print Properties	Color check box under Print in the Print Properties dialog box
PrintFirstPageNumber	N/A	From the General menu, choose Print Properties	Initial Page Number box under Job Setup in the Print Properties dialog box
PrintFooter	N/A	From the General menu, choose Print Properties	Footer Text box under Job Setup in the Print Properties dialog box
PrintGrid	N/A	From the General menu, choose Print Properties	Grid check box under Print in the Print Properties dialog box
PrintHeader	N/A	From the General menu, choose Print Properties	Header Text box under Job Setup in the Print Properties dialog box
PrintJobName	N/A	From the General menu, choose Print Properties	Job Name box under Job Setup in the Print Properties dialog box
PrintMarginBottom	N/A	From the General menu, choose Print Properties	Bottom box under Margins in the Print Properties dialog box
PrintMarginLeft	N/A	From the General menu, choose Print Properties	Left box under Margins in the Print Properties dialog box
PrintMarginRight	N/A	From the General menu, choose Print Properties	Right box under Margins in the Print Properties dialog box
PrintMarginTop	N/A	From the General menu, choose Print Properties	Top box under Margins in the Print Properties dialog box
PrintNextPageBreakCol	N/A	N/A	N/A
PrintNextPageBreakRow	N/A	N/A	N/A

Property	Button to display dialog box	Menu item to display dialog box	To set the property in the dialog box
PrintOrientation	N/A	From the General menu, choose Print Properties	Option buttons under Orientation in the Print Properties dialog box
PrintPageCount	N/A	N/A	N/A
PrintPageEnd	N/A	From the General menu, choose Print Properties	To box under Range in the Print Properties dialog box
PrintPageOrder	N/A	From the General menu, choose Print Properties	Option buttons under Page Order in the Print Properties dialog box
PrintPageStart	N/A	From the General menu, choose Print Properties	Pages box under Range in the Print Properties dialog box
PrintRowHeaders	N/A	From the General menu, choose Print Properties	Row Headers check box under Print in the Print Properties dialog box
PrintShadows	N/A	From the General menu, choose Print Properties	Shadows check box under Print in the Print Properties dialog box
PrintSmartPrint	N/A	N/A	N/A
PrintType	N/A	From the General menu, choose Print Properties	Option buttons under Range in the Print Properties dialog box
PrintUseDataMax	N/A	From the General menu, choose Print Properties	Data Cells Only check box under Print in the Print Properties dialog box
ProcessTab		From the General menu, choose Spreadsheet Environment	Process Tab Key check box under Settings on the General tab in the Sheet Environment Settings dialog box
Protect		From the General menu, choose Spreadsheet Environment	Protect Locked Cells check box under Settings on the General Tab in the Sheet Environment Settings dialog box
ReDraw	N/A	N/A	N/A
RestrictCols		From the General menu, choose Headers	Restrict Columns check box under Settings on the Columns tab in the Headers dialog box
RestrictRows		From the General menu, choose Headers	Restrict Rows check box under Settings on the Rows tab in the Headers dialog box
RetainSelBlock		From the General menu, choose Spreadsheet Environment	Retain Selected Block check box under Settings on the General tab in the Sheet Environment Settings dialog box
Row	N/A	N/A	Property is set in generated code when performing certain actions in Learn Mode

## Appendix F

Property	Button to display dialog box	Menu item to display dialog box	To set the property in the dialog box
Row2	N/A	N/A	Property is set in generated code when performing certain actions in Learn Mode
RowHeaderDisplay		From the General menu, choose Headers	Header Styles drop-down list box under Headers on the Columns tab in the Headers dialog box
RowHeight		From the General menu, choose Columns and Rows	Row Height box on the Rows tab in the Column / Rows dialog box
RowHidden		From the General menu, choose Columns and Rows	Row Hidden check box on the Rows tab in the Columns / Rows dialog box
RowPageBreak		From the General menu, choose Columns and Rows	Row Page Break check box on the Rows tab in the Columns / Rows dialog box
RowsFrozen		From the General menu, choose Columns and Rows	Frozen Rows spin box under Settings on the Rows tab in the Columns / Rows dialog box
ScrollBarExtMode		From the Display menu, choose Scroll Bars	Display Only if Needed check box on the Scroll Bars tab in the Display dialog box
ScrollBarMaxAlign		From the Display menu, choose Scroll Bars	Align at Last Row and Column check box on the Scroll Bars tab in the Display dialog box
ScrollBars		From the Display menu, choose Scroll Bars	Option buttons under Display on the Scroll Bars tab in the Display dialog box
ScrollBarShowMax		From the Display menu, choose Scroll Bars	Scroll Box Reflects Max Rows check box on the Scroll Bars tab in the Display dialog box
ScrollBarTrack		From the Display menu, choose Scroll Bars	Option buttons under Scroll Box Track on the Scroll Bars tab in the Display dialog box
SelBackColor		From the General menu, choose Spreadsheet Colors	Select SelBackColor from the Property drop-down list box on the Colors tab in the Default dialog box
SelBlockCol	N/A	N/A	N/A
SelBlockCol2	N/A	N/A	N/A
SelBlockRow	N/A	N/A	N/A
SelBlockRow2	N/A	N/A	N/A
SelectBlockOptions		From the General menu, choose Spreadsheet Environment	Check boxes under Settings on the BlockMode tab in the Sheet Environment Settings dialog box

Property	Button to display dialog box	Menu item to display dialog box	To set the property in the dialog box
SelForeColor		From the General menu, choose Spreadsheet Colors	Select SelForeColor from the Property drop-down list box on the Colors tab in the Default dialog box
SelLength	N/A	N/A	N/A
SelModeIndex	N/A	N/A	N/A
SelModeSelCount	N/A	N/A	N/A
SelModeSelected	N/A	N/A	N/A
SelStart	N/A	N/A	N/A
SelText	N/A	N/A	N/A
ShadowColor		From the General menu, choose Spreadsheet Colors	Select ShadowColor from the Property drop-down list box on the Colors tab in the Default dialog box
ShadowDark		From the General menu, choose Spreadsheet Colors	Select ShadowDark from the Property drop-down list box on the Colors tab in the Default dialog box
ShadowText		From the General menu, choose Spreadsheet Colors	Select ShadowText from the Property drop-down list box on the Colors tab in the Default dialog box
SortBy	N/A	From the General menu, choose Spreadsheet Sort	Option buttons under Sort by in the Sort Settings dialog box
SortKey	N/A	From the General menu, choose Spreadsheet Sort	Column or Row spin box under Settings Key <i>n</i> in the Sort Settings dialog box
SortKeyOrder	N/A	From the General menu, choose Spreadsheet Sort	Option buttons under Settings Key <i>n</i> in the Sort Settings dialog box
StartingColNumber		From the General menu, choose Headers	Start Col Num spin box under Headers on the Columns tab in the Headers dialog box
StartingRowNumber		From the General menu, choose Headers	Start Row Num spin box under Headers on the Rows tab in the Headers dialog box
Text	N/A	N/A	Property is set in generated code when performing certain actions in Learn Mode
TextTip		From the Display menu, choose Text Tip	Display drop-down list box on the Text Tip tab in the Display dialog box
TextTipDelay		From the Display menu, choose Text Tip	Delay spin box on the Text Tip tab in the Display dialog box

## Appendix F

Property	Button to display dialog box	Menu item to display dialog box	To set the property in the dialog box
TopRow		From the General menu, choose Columns and Rows	Top Row spin box under Settings on the Rows tab in the Columns / Rows dialog box
TwoDigitYearMax	N/A	From the General menu, choose Spreadsheet Environment	Pivot Year spin box under Settings on the General tab in the Sheet Environment Settings dialog box
TypeButtonAlign		From the Cell menu, choose Cell Type, Button	Text Pos drop-down list box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Button
TypeButtonBorderColor		From the Cell menu, choose Cell Type, Button	Select BorderColor from the drop-down list box under Color on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Button
TypeButtonColor		From the Cell menu, choose Cell Type, Button	Select ButtonColor from the drop-down list box under Color on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Button
TypeButtonDarkColor		From the Cell menu, choose Cell Type, Button	Select DarkColor from the drop-down list box under Color on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Button
TypeButtonLightColor		From the Cell menu, choose Cell Type, Button	Select LightColor from the drop-down list box under Color on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Button
TypeButtonPicture		From the Cell menu, choose Cell Type, Button	Picture Up in the drop-down list box and command buttons under Picture on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Button
TypeButtonPictureDown		From the Cell menu, choose Cell Type, Button	Picture Down in the drop-down list box and command buttons under Picture on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Button
TypeButtonShadowSize		From the Cell menu, choose Cell Type, Button	Shadow spin box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Button

Property	Button to display dialog box	Menu item to display dialog box	To set the property in the dialog box
TypeButtonText		From the Cell menu, choose Cell Type, Button	Text box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Button
TypeButtonTextColor		From the Cell menu, choose Cell Type, Button	Select TextColor from the drop-down list box under Color on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Button
TypeButtonType		From the Cell menu, choose Cell Type, Button	Style drop-down list box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Button
TypeCheckCenter		N/A	N/A
TypeCheckPicture		From the Cell menu, choose Cell Type, Check Box	Drop-down list box under Picture on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Check Box
TypeCheckText		From the Cell menu, choose Cell Type, Check Box	Text box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Check Box
TypeCheckTextAlign		From the Cell menu, choose Cell Type, Check Box	Text Left of Picture check box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Check Box
TypeCheckType		From the Cell menu, choose Cell Type, Check Box	Options buttons under Type on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Check Box
TypeComboBoxCount	N/A	N/A	N/A
TypeComboBoxCurSel	N/A	N/A	N/A
TypeComboBoxEditable		From the Cell menu, choose Cell Type, Combo Box	Editable check box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Combo Box
TypeComboBoxhWnd	N/A	N/A	N/A
TypeComboBoxIndex	N/A	N/A	N/A
TypeComboBoxList		From the Cell menu, choose Cell Type, Combo Box	Text box and Combo Box List group box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Combo Box

## Appendix F

Property	Button to display dialog box	Menu item to display dialog box	To set the property in the dialog box
TypeComboBoxMaxDrop		From the Cell menu, choose Cell Type, Combo Box	Max Drop spin box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Combo Box
TypeComboBoxString		From the Cell menu, choose Cell Type, Combo Box	Text box and Combo Box List group box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Combo Box
TypeComboBoxWidth		From the Cell menu, choose Cell Type, Combo Box	List Width drop-down list box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Combo Box
TypeDateCentury		From the Cell menu, choose Cell Type, Date	Display Century check box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Date
TypeDateFormat		From the Cell menu, choose Cell Type, Date	Format drop-down list box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Date
TypeDateMax		From the Cell menu, choose Cell Type, Date	Max Date box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Date
TypeDateMin		From the Cell menu, choose Cell Type, Date	Min Date box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Date
TypeDateSeparator		From the Cell menu, choose Cell Type, Date	Separator box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Date
TypeEditCharCase		From the Cell menu, choose Cell Type, Edit	Option buttons under Text Case on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Edit
TypeEditCharSet		From the Cell menu, choose Cell Type, Edit	Character Set drop-down list box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Edit
TypeEditMultiLine		From the Cell menu, choose Cell Type, Edit	Multiple Lines check box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Edit
TypeEditPassword		From the Cell menu, choose Cell Type, Edit	Password check box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Edit

Property	Button to display dialog box	Menu item to display dialog box	To set the property in the dialog box
TypeFloatCurrencyChar		From the Cell menu, choose Cell Type, Float	Money box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Float
TypeFloatDecimalChar		From the Cell menu, choose Cell Type, Float	Decimal box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Float
TypeFloatDecimalPlaces		From the Cell menu, choose Cell Type, Float	Decimal Digits spin box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Float
TypeFloatMax		From the Cell menu, choose Cell Type, Float	Max Value box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Float
TypeFloatMin		From the Cell menu, choose Cell Type, Float	Min Value box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Float
TypeFloatMoney		From the Cell menu, choose Cell Type, Float	Money check box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Float
TypeFloatSeparator		From the Cell menu, choose Cell Type, Float	Separator check box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Float
TypeFloatSepChar		From the Cell menu, choose Cell Type, Float	Separator box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Float
TypeHAlign		From the Cell menu, choose Cell Type, Check Box	Options buttons under Horizontal Alignment on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Check Box
		From the Cell menu, choose Cell Type, Combo Box	Options buttons under Horizontal Alignment on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Combo Box
		From the Cell menu, choose Cell Type, Date	Options buttons under Horizontal Alignment on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Date

## Appendix F

Property	Button to display dialog box	Menu item to display dialog box	To set the property in the dialog box
TypeHAlign ( <i>continued</i> )		From the Cell menu, choose Cell Type, Edit	Options buttons under Horizontal Alignment on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Edit
		From the Cell menu, choose Cell Type, Float	Options buttons under Horizontal Alignment on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Float
		From the Cell menu, choose Cell Type, Integer	Options buttons under Horizontal Alignment on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Integer
		From the Cell menu, choose Cell Type, Label(Static)	Options buttons under Horizontal Alignment on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Label(Static)
		From the Cell menu, choose Cell Type, Mask(Pic)	Options buttons under Horizontal Alignment on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Mask(Pic)
		From the Cell menu, choose Cell Type, Picture	Options buttons under Horizontal Alignment on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Picture
		From the Cell menu, choose Cell Type, Time	Options buttons under Horizontal Alignment on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Time
TypeIntegerMax		From the Cell menu, choose Cell Type, Integer	Max Value box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Integer
TypeIntegerMin		From the Cell menu, choose Cell Type, Integer	Min Value box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Integer
TypeIntegerSpinInc		From the Cell menu, choose Cell Type, Integer	Increment spin box under Spin Settings on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Integer

Property	Button to display dialog box	Menu item to display dialog box	To set the property in the dialog box
TypeIntegerSpinWrap		From the Cell menu, choose Cell Type, Integer	Spin Wrap check box under Spin Settings on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Integer
TypeMaxEditLen		From the Cell menu, choose Cell Type, Combo Box	Text Length spin box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Combo Box
		From the Cell menu, choose Cell Type, Edit	Length spin box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Edit
TypeOwnerDrawStyle		From the Cell menu, choose Cell Type, Owner Draw	Owner Draw ID spin box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to OwnerDraw
TypePicDefaultText	N/A	N/A	N/A
TypePicMask		From the Cell menu, choose Cell Type, Mask(Pic)	Mask Definition box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Mask(Pic)
TypePictCenter	N/A	N/A	N/A
TypePictMaintainScale		From the Cell menu, choose Cell Type, Picture	Maintain Scale check box under Appearance on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Picture
TypePictPicture		From the Cell menu, choose Cell Type, Picture	Picture button on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Picture
TypePictStretch		From the Cell menu, choose Cell Type, Picture	Stretch check box under Appearance on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Picture
TypeSpin		From the Cell menu, choose Cell Type, Date	Display Spin Button check box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Date
		From the Cell menu, choose Cell Type, Integer	Display Spin Button check box under Spin Settings on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Integer
		From the Cell menu, choose Cell Type, Time	Display Spin Button check box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Time

## Appendix F

Property	Button to display dialog box	Menu item to display dialog box	To set the property in the dialog box
TypeTextPrefix		From the Cell menu, choose Cell Type, Label(Static)	Display Mnemonic check box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Label(Static)
TypeTextShadow		From the Cell menu, choose Cell Type, Label(Static)	Raised check box under Shadow on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Label(Static)
TypeTextShadowIn		From the Cell menu, choose Cell Type, Label(Static)	Lowered check box under Shadow on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Label(Static)
TypeTextWordWrap		From the Cell menu, choose Cell Type, Label(Static)	Word Wrap check box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Label(Static)
TypeTime24Hour		From the Cell menu, choose Cell Type, Time	Format drop-down list box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Time
TypeTimeMax		From the Cell menu, choose Cell Type, Time	Max Time box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Time
TypeTimeMin		From the Cell menu, choose Cell Type, Time	Min Time box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Time
TypeTimeSeconds		From the Cell menu, choose Cell Type, Time	Display Seconds check box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Time
TypeTimeSeparator		From the Cell menu, choose Cell Type, Time	Separator box on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Time

Property	Button to display dialog box	Menu item to display dialog box	To set the property in the dialog box
TypeVAlign		From the Cell menu, choose Cell Type, Check Box	Options buttons under Vertical Alignment on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Check Box
		From the Cell menu, choose Cell Type, Combo Box	Options buttons under Vertical Alignment on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Combo Box
		From the Cell menu, choose Cell Type, Date	Options buttons under Vertical Alignment on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Date
		From the Cell menu, choose Cell Type, Edit	Options buttons under Vertical Alignment on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Edit
		From the Cell menu, choose Cell Type, Float	Options buttons under Vertical Alignment on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Float
		From the Cell menu, choose Cell Type, Integer	Options buttons under Vertical Alignment on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Integer
		From the Cell menu, choose Cell Type, Label(Static)	Options buttons under Vertical Alignment on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Label(Static)
		From the Cell menu, choose Cell Type, Mask(Pic)	Options buttons under Vertical Alignment on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Mask(Pic)
		From the Cell menu, choose Cell Type, Picture	Options buttons under Vertical Alignment on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Picture
		From the Cell menu, choose Cell Type, Time	Options buttons under Vertical Alignment on the Settings tab in the Cell Type dialog box with the Cell Type drop-down list box set to Time

## Appendix F

Property	Button to display dialog box	Menu item to display dialog box	To set the property in the dialog box
UnitType	N/A	From the General menu, choose Unit Type	Option buttons under Settings on the UnitType tab in the Default dialog box
UserResize		From the General menu, choose Headers	User Resize check box under Settings on the Columns tab and User Resize check box under Settings on the Rows tab in the Headers dialog box
UserResizeCol		From the General menu, choose Columns and Rows	Option buttons under Resizable on the Columns tab in the Columns / Rows dialog box
UserResizeRow		From the General menu, choose Columns and Rows	Option buttons under Resizable on the Rows tab in the Columns / Rows dialog box
Value	N/A	N/A	N/A
VirtualCurRowCount	N/A	N/A	N/A
VirtualCurTop	N/A	N/A	N/A
VirtualMaxRows		From the General menu, choose Virtual Mode	Virtual Max Rows box under Settings in the Virtual Mode dialog box
VirtualMode		From the General menu, choose Virtual Mode	Virtual Mode check box under Settings in the Virtual Mode dialog box
VirtualOverlap		From the General menu, choose Virtual Mode	Memory Max Rows box under Settings in the Virtual Mode dialog box
VirtualRows		From the General menu, choose Virtual Mode	Rows Per Read box under Settings in the Virtual Mode dialog box
VirtualScrollBuffer		From the General menu, choose Virtual Mode	Thumb Reflects Row in Buffer check box under Virtual Scroll Bar in the Virtual Mode dialog box
VisibleCols		From the General menu, choose Columns and Rows	Visible Cols spin box under Settings on the Columns tab in the Columns / Rows dialog box
VisibleRows		From the General menu, choose Columns and Rows	Visible Rows spin box under Settings on the Rows tab in the Columns / Rows dialog box
VScrollSpecial		From the General menu, choose Virtual Mode	Use Special Scroll Bar check box under Virtual Scroll Bar in the Virtual Mode dialog box
VScrollSpecialType		From the General menu, choose Virtual Mode	No Home/End, No Page Up/Down, and No Line Up/Down check boxes under Virtual Scroll Bar in the Virtual Mode dialog box

# Appendix G Formula Operators and Functions

Spread provides operators and predefined formula functions to use when you create formulas.

## Formula Operators

The following table lists the available operators.

Operator	Description	Literal / Literal	Cell Ref / Literal	Cell Ref / Cell Ref
+	Addition	2 + 2	A1 + 2	A1 + B2
-	Subtraction	4 - 2	A1 - 2	A1 - B2
^	Exponentiation	2 ^ 2	A1 ^ 2	A1 ^ B2
*	Multiplication	2 * 2	A1 * 2	A1 * B2
/	Division	2 / 2	A1 / 2	A1 / B2
&	Logical And	2 & 2	A1 & 2	A1 & B2
	Logical Or	2   2	A1   2	A1   B2
:	Creates a range			A1: B2
#	Wild card		A# * 2	A# * B#

---

**Note** The wild card operator (#) is available only when the reference style is `SS_REFSTYLE_DEFAULT`. For more information, see “Specifying Cell References” on page 103.

---

## Formula Functions

The following sections describe the formula functions available. Each function provides an example. Examples that provide results give decimal values for 10 decimal places. For most functions, the arguments can be number values, cell references, or mathematical expressions that contain cell references, or other values as described in the section.

## Appendix G

### ABS

<b>Name</b>	Absolute value
<b>Description</b>	Returns the absolute value of the specified value.
<b>Syntax</b>	ABS( <i>value</i> )
<b>Remarks</b>	The <i>value</i> argument can be any real number.  See also NEG and SIGN.
<b>Examples</b>	ABS (R1C2) ABS (B3) ABS (-4) =4

### ACOS

<b>Name</b>	Arccosine
<b>Description</b>	Returns the angle whose cosine is the specified value. The angle is in radians between 0 (zero) and $\pi$ (pi).
<b>Syntax</b>	ACOS( <i>value</i> )
<b>Remarks</b>	The <i>value</i> argument is the cosine of the angle you want to return and must be between -1 and 1.  If you want to convert the result to degrees, multiply it by $180/\pi$ .  See also ASIN, ATAN, ATAN2, COS, and COSH.
<b>Examples</b>	ACOS (R1C2) ACOS (B3) ACOS (0.5) =1.0471975512

### ADD

<b>Name</b>	Addition
<b>Description</b>	Adds the two arguments.
<b>Syntax</b>	ADD( <i>value1</i> , <i>value2</i> )
<b>Remarks</b>	See also SUM.
<b>Example</b>	ADD (R1C2, R2C3) ADD (B3, C4) ADD (-1, 9) =8

## ASIN

<b>Name</b>	Arcsine
<b>Description</b>	Returns the angle whose sine is the specified value. The angle is in radians between $-\pi/2$ and $\pi/2$ .
<b>Syntax</b>	ASIN( <i>value</i> )
<b>Remarks</b>	The <i>value</i> argument is the sine of the angle you want to return and must be between $-1$ and $1$ .  If you want to convert the result to degrees, multiply it by $180/\pi$ .  See also ACOS, ATAN, ATAN2, SIN, and SINH.
<b>Example</b>	ASIN(R1C2) ASIN(B3) ASIN(0.5)=0.5235987756

## ATAN

<b>Name</b>	Arctangent
<b>Description</b>	Returns the angle whose tangent is the specified value. The angle is in radians between $-\pi/2$ and $\pi/2$ .
<b>Syntax</b>	ATAN( <i>value</i> )
<b>Remarks</b>	The <i>value</i> argument is the tangent of the angle you want to return and must be between $-1$ and $1$ . If you want to convert the result to degrees, multiply it by $180/\pi$ .  See also ACOS, ASIN, ATAN2, TAN, and TANH.
<b>Example</b>	ATAN(R1C2) ATAN(B3) ATAN(1)=0.7853981634

## ATAN2

<b>Name</b>	Arctangent 2
<b>Description</b>	Returns the arctangent of the specified <i>x</i> - and <i>y</i> - coordinates. The result is given in radians between $-\pi$ and $\pi$ , excluding $-\pi$ .
<b>Syntax</b>	ATAN2( <i>value1</i> , <i>value2</i> )

## Appendix G

**Remarks** The arctangent is the angle from the  $x$ -axis to a line containing the origin (0, 0) and a point with coordinates ( $value1$ ,  $value2$ ). If you want to convert the result to degrees, multiply it by  $180/\pi$ .

See also ACOS, ASIN, ATAN, TAN, and TANH.

**Example** ATAN2 (R1C2, R3C5)  
ATAN2 (A1, C3)  
ATAN2 (1, 1)=0.7853981634

## AVERAGE

**Name** Average

**Description** Returns the average of the specified values.

**Syntax** AVERAGE( $value1$ ,  $value2$ , . . . )

**Remarks** Accepts up to 30 arguments. Each argument can be a cell range, a float value, or an integer value.

See also MEDIAN and MODE.

**Example** AVERAGE (A1, B3, D5, E9)  
AVERAGE (R1C2, R3C1)  
AVERAGE (A1 : A9)  
AVERAGE (98, 72, 85)=85

## CEILING

**Name** Ceiling

**Description** Rounds a number up to the nearest multiple of a specified value.

**Syntax** CEILING( $value1$ ,  $value2$ )

**Remarks** Specify the number to round using the  $value1$  argument. Specify the value to use as the rounding factor using the  $value2$  argument. Use either both positive or both negative numbers for the arguments. Regardless of the sign of number, the number is rounded away from zero.

See also EVEN, FLOOR, INT, ODD, ROUND, ROUNDUP, and TRUNCATE.

**Example** CEILING (R1C2, 1)  
CEILING (C4, B2)  
CEILING (B3, 0.05)  
CEILING (4.65, 2)=6  
CEILING (-2.78, -1)=-3

**COMBIN****Name** Combination**Description** Returns the number of possible combinations for a specified number of items.**Syntax** COMBIN(*value1*, *value2*)**Remarks** Specify the number of items using the *value1* argument. Specify the number of items in each possible combination using the *value2* argument. Both arguments must be positive integers, and the *value1* argument must be greater than or equal to the *value2* argument.

A combination is any set or subset of items, regardless of the internal order of the items. Contrast with *permutation* (PERMUT function).

The number of combinations is calculated as follows, where *value1*=*n* and *value2*=*k*:

$$\binom{n}{k} = \frac{P_{k,n}}{k!} = \frac{n!}{k!(n-k)!}$$

where:

$$P_{k,n} = \frac{n!}{(n-k)!}$$

See also PERMUT.

**Example**

```
COMBIN(R1C2, 2)
COMBIN(C4, B2)
COMBIN(B3, 5)
COMBIN(8, 2)=28
COMBIN(100, 3)=161700
```

**COS****Name** Cosine**Description** Returns the cosine of the specified angle.**Syntax** COS(*value*)

## Appendix G

**Remarks** Specify the angle in radians using the *value* argument. If the angle is in degrees, multiply it by  $\pi/180$  to convert it to radians.

See also ACOS, COSH, SIN, and TAN.

**Example** COS (R1C3)  
COS (B2)  
COS (45\*PI () /180)=0.7071067812

## COSH

**Name** Hyperbolic cosine

**Description** Returns the hyperbolic cosine of the specified value.

**Syntax** COSH(*value*)

**Remarks** You can specify any real number for the *value* argument.

The hyperbolic cosine is calculated as follows:

$$\text{COSH}(z) = \frac{e^z + e^{-z}}{2}$$

See also ACOS, COS, SINH, and TANH.

**Example** COSH (R1C2)  
COSH (B3)  
COSH (4)=27.3082328360

## DB

**Name** Depreciation (fixed-declining balance method)

**Description** Calculates the depreciation of an asset for a specified period using the fixed-declining balance method.

**Syntax** DB(*Cost, Salvage, Life, Period, Month*)

**Remarks** Arguments are as follows:

<b>Argument</b>	<b>Description</b>
<i>Cost</i>	Initial cost of the asset
<i>Salvage</i>	Value at the end of the depreciation period
<i>Life</i>	Number of periods over which the asset is being depreciated
<i>Period</i>	Period for which you want to calculate the depreciation Use the same units as the <i>Life</i> argument.
<i>Month</i>	(Optional) Number of months in the first year If omitted, the calculation assumes 12 months.

The fixed-declining balance method computes depreciation at a fixed rate. The DB function uses the following formulas to calculate depreciation for a period:

$$(Cost - \text{total depreciation from prior periods}) \times \text{rate}$$

where:

$$\text{rate} = 1 - ((Salvage / Cost)^{(1 / Life)}), \text{ rounded to three decimal places}$$

Depreciation for the first and last periods is a special case. For the first period, the DB function uses this formula:

$$Cost \times \text{rate} \times \text{month} / 12$$

For the last period, the DB function uses this formula:

$$((Cost - \text{total depreciation from prior periods}) \times \text{rate} \times (12 - \text{month})) / 12$$

See also DDB, SLN, and SYD.

**Example** DB(R1C2, 10000, 10, 1)  
DB(500000, 5000, 5, 1, 10)=250833.3333333333

## DDB

<b>Name</b>	Depreciation (double-declining balance method)
<b>Description</b>	Calculates the depreciation of an asset for a specified period using the double-declining balance method or another method you specify.
<b>Syntax</b>	DDB( <i>Cost, Salvage, Life, Period, Factor</i> )

## Appendix G

**Remarks** Arguments are as follows:

<b>Argument</b>	<b>Description</b>
<i>Cost</i>	Initial cost of the asset
<i>Salvage</i>	Value at the end of depreciation
<i>Life</i>	Number of periods over which the asset is being depreciated
<i>Period</i>	Period for which you want to calculate the depreciation Use the same units as the <i>Life</i> argument.
<i>Factor</i>	Rate at which the value declines This is an optional argument. If omitted, the calculation assumes 2 (double declining method) Change this to a value other than 2 to use a different method.

All five arguments must be positive numbers.

The DDB function uses the following formula to calculate depreciation for a period:

$$Cost - Salvage(\text{total depreciation from prior periods}) \times Factor/Life$$

See also DB, SLN, and SYD.

**Example** DDB (R1C2, 10000, 10, 1)  
DDB (500000, 5000, 5, 1, 4) = 400000

## DEGREES

<b>Name</b>	Degrees
<b>Description</b>	Converts the specified value from radians to degrees.
<b>Syntax</b>	DEGREES( <i>value</i> )
<b>Remarks</b>	See also RADIANS.
<b>Example</b>	DEGREES (R1C2) DEGREES (B3) DEGREES (PI ()) = 180

**EVEN**

<b>Name</b>	Even
<b>Description</b>	Rounds the specified value up to the nearest even integer.
<b>Syntax</b>	EVEN( <i>value</i> )
<b>Remarks</b>	Regardless of the sign of the number specified by the <i>value</i> argument, the number is rounded away from zero.  See also CEILING, FLOOR, INT, ODD, ROUND, ROUNDUP, and TRUNCATE.
<b>Example</b>	EVEN (R1C2) EVEN (A3) EVEN (5) =6 EVEN (-2.5) =-4

**EXP**

<b>Name</b>	Exponent base e
<b>Description</b>	Returns e raised to the power of the specified value ( $e^x$ ). EXP is the inverse of LN.
<b>Syntax</b>	EXP( <i>value</i> )
<b>Remarks</b>	See also LN and POWER.
<b>Example</b>	EXP (R1C2) EXP (B3) EXP (1) =2.7182818285 EXP (LN (4) ) =4

**FACT**

<b>Name</b>	Factorial
<b>Description</b>	Returns the factorial of the specified number.
<b>Syntax</b>	FACT( <i>value</i> )
<b>Remarks</b>	The <i>value</i> argument must be a nonnegative number. The factorial is calculated as $1 \times 2 \times 3 \times \dots \times \text{value}$ . If you provide a number that is not an integer for the <i>value</i> argument, the decimal portion of the number is ignored.  See also PRODUCT.

## Appendix G

**Example**     `FACT (R1C2)`  
                 `FACT (B3)`  
                 `FACT (10) = 3628800`

## FALSE

**Name**         `False`

**Description**     Specifies the logical value FALSE.

**Syntax**        `FALSE()`

**Remarks**        The function does not accept arguments.  
  
                     See also NOT and TRUE.

**Example**        `FALSE ()`

## FLOOR

**Name**            `Floor`

**Description**     Rounds a number down to the nearest multiple of a specified value.

**Syntax**          `FLOOR(value1, value2)`

**Remarks**        Specify the number to round using the *value1* argument. Specify the value to use as the rounding factor using the *value2* argument. Use either both positive or both negative numbers for the arguments. Regardless of the sign of number, the number is rounded toward zero.  
  
                     See also CEILING, EVEN, INT, ODD, ROUND, ROUNDUP, and TRUNCATE.

**Example**         `FLOOR (R1C2, 1)`  
                     `FLOOR (C4, B2)`  
                     `FLOOR (B3, 0.05)`  
                     `FLOOR (4.65, 2) = 4`  
                     `FLOOR (-2.78, -1) = -2`

# FV

<b>Name</b>	Future Value
<b>Description</b>	Returns the future value of an investment based on a present value, periodic payments, and a specified interest rate.
<b>Syntax</b>	$FV(Rate, Nper, Pmt, Pv, Type)$
<b>Remarks</b>	Arguments are as follows:

<b>Argument</b>	<b>Description</b>
<i>Rate</i>	Interest rate expressed as percentage (per period)
<i>Nper</i>	Total number of payment periods
<i>Pmt</i>	Payment made each period
<i>Pv</i>	Present value This is an optional argument. If omitted, it is assumed to be zero and the calculation is based on the <i>Pmt</i> argument.
<i>Type</i>	Indicates when payments are due This is an optional argument. If omitted, it is assumed that the payments are made at the end of the period.
	<b>Set <i>Type</i> equal to</b>
	<b>If payments are due</b>
	0                      At the end of the period
	1                      At the beginning of the period

---

**Note** You must specify a value for the *Pv* argument if you are going to specify a value for the *Type* argument.

---

Use consistent units for specifying the *Rate* and *Nper* arguments. If you make monthly payments on a five-year loan at 8 percent annual interest, use 0.08/12 for the *Rate* argument and 5\*12 for the *Nper* argument. If you make annual payments on the same loan, use 0.08 for *Rate* and 5 for *Nper*.

For all the arguments, money paid out such as deposits in an investment, is represented by negative numbers; money you receive, such as dividend checks, is represented by positive numbers.

See the PV function for the formula used to calculate financial values.

See also NPER, PMT, PV.

**Example**  $FV(R1C1/12, 48, R1C2, 1000, 0)$   
 $FV(0.005, 60, -100, 100, 1) = 6877$

## IF

<b>Name</b>	Comparison
<b>Description</b>	If <i>value1</i> is nonzero (or True), then <i>value2</i> is returned. If <i>value1</i> is zero (or False), then <i>value3</i> is returned. <i>value1</i> can contain one of the relational operators: greater than (>), less than (<), equal to (=), or not equal to (<>).
<b>Syntax</b>	IF( <i>value1</i> , <i>value2</i> , <i>value3</i> )
<b>Example</b>	IF (R1C2>65, 1000, 2000) IF (A3<>2000, 1900, 2000) IF (C4, B2, B4) IF (1>2, 5, 10)=10

## INT

<b>Name</b>	Integer
<b>Description</b>	Rounds a specified number down to the nearest integer.
<b>Syntax</b>	INT( <i>value</i> )
<b>Remarks</b>	You can use the INT function to return the decimal portion of the value in a cell by subtracting the value of INT() for the cell from the value in the cell, as illustrated in the first example.  See also CEILING, EVEN, FLOOR, ODD, ROUND, ROUNDUP, and TRUNCATE.
<b>Example</b>	R1C2-INT (R1C2) INT (A3) INT (2.85)=2 INT (-2.85)=-3

## INVERSE

<b>Name</b>	Inverse of the specified value
<b>Description</b>	Returns 1/ <i>x</i> as a decimal value.
<b>Syntax</b>	INVERSE( <i>value</i> )
<b>Example</b>	INVERSE(R1C2) INVERSE(B3) INVERSE(5)=0.20

**ISEMPTY**

<b>Name</b>	Is cell empty
<b>Description</b>	If the cell specified by the <i>value</i> argument is empty, then a 1 (True) is returned. If the cell specified by the <i>value</i> argument is not empty, then a 0 (False) is returned.
<b>Syntax</b>	ISEMPTY( <i>value</i> )
<b>Example</b>	ISEMPTY (R1C2) ISEMPTY (B3)

**ISEVEN**

<b>Name</b>	Is number even
<b>Description</b>	If the number specified by the <i>value</i> argument is even, then 1 (True) is returned. If the number specified by the <i>value</i> argument is odd, then 0 (False) is returned.
<b>Syntax</b>	ISEVEN( <i>value</i> )
<b>Remarks</b>	Specify any number for the <i>value</i> argument.  See also ISODD.
<b>Example</b>	ISEVEN (R1C2) ISEVEN (B3) ISEVEN (9) =0 ISEVEN (2 . 4) =1 ISEVEN (3 . 4) =0

**ISNONTEXT**

<b>Name</b>	Is nontext
<b>Description</b>	Returns 1 (True) if the <i>value</i> argument is or refers to any item that is not text. (If the cell is empty, 1 is returned.) Returns 0 (False) if the <i>value</i> argument is or refers to any item that is text.
<b>Syntax</b>	ISNONTEXT( <i>value</i> )
<b>Remarks</b>	See also ISNUMBER and ISTEXT.
<b>Example</b>	ISNONTEXT (R1C2) ISNONTEXT (A3) ISNONTEXT (12) =1

## ISNUMBER

<b>Name</b>	Is number
<b>Description</b>	Returns 1 (True) if the <i>value</i> argument is or refers to a number. Returns 0 (False) if the argument is or refers to a value that is not a number.
<b>Syntax</b>	ISNUMBER( <i>value</i> )
<b>Remarks</b>	You might want to use the ISNUMBER function to test whether cells contain numeric data before you perform mathematical operations on them, such as averaging the cells' contents.  See also ISNONTTEXT and ISTEXT.
<b>Example</b>	ISNUMBER (R1C2) ISNUMBER (B3) ISNUMBER (12) =1

## ISODD

<b>Name</b>	Is number odd
<b>Description</b>	If the number specified by the <i>value</i> argument is odd, then the function returns 1 (True). If the number specified by the <i>value</i> argument is even, then the function returns 0 (False).
<b>Syntax</b>	ISODD( <i>value</i> )
<b>Remarks</b>	Specify any number for the <i>value</i> argument.  See also ISEVEN.
<b>Example</b>	ISODD (R1C2) ISODD (B3) ISODD (12) =0 ISODD (2.5) =0 ISODD (3.4) =1

## ISREF

<b>Name</b>	Is reference
<b>Description</b>	Returns 1 (True) if the <i>value</i> argument refers to a reference. Returns 0 (False) if the <i>value</i> argument refers to a value that is not a reference.
<b>Syntax</b>	ISREF( <i>value</i> )

**Remarks** Use the ISREF function to determine whether the specified cell contains a reference to another cell.

**Example** ISREF (R1C2)  
ISREF (A3)

## ISTEXT

**Name** Is text

**Description** Returns 1 (True) if the *value* argument is or refers to text. Returns 0 (False) if the *value* argument is or refers to a value that is not text.

**Syntax** ISTEXT(*value*)

**Remarks** See also ISNONTEXT and ISNUMBER.

**Example** ISTEXT (R1C2)  
ISTEXT (B3)  
ISTEXT (12) =0

## LN

**Name** Natural log

**Description** Returns the natural log of the specified number ( $e^{\ln}=\text{value}$ ). LN is the inverse of EXP.

**Syntax** LN(*value*)

**Remarks** Specify a positive real number for the *value* argument.

See also EXP, LOG, and LOG10.

**Example** LN (R1C2)  
LN (B3)  
LN (10) =2 . 3025850930  
LN (EXP (1) ) =1

## LOG

**Name** Logarithm

**Description** Returns the log base *Y* of a number *X* ( $Y^{\log}=\text{X}$ ).

**Syntax** LOG(*value1*, *value2*)

## Appendix G

**Remarks** The second argument is optional. If you omit the *value2* argument, the function assumes base 10. The number specified by the *value1* argument must be a positive real number.

See also LN and LOG10.

**Example**  
LOG (R1C2, R4C4)  
LOG (B3, C5)  
LOG (255, 16) = 1.9985883592

## LOG10

**Name** Base 10 logarithm

**Description** Returns the log base 10 of the number  $X$  ( $10^{\log=X}$ ).

**Syntax** LOG10(*value*)

**Remarks** The number specified by the *value* argument must be a positive real number.

See also LN and LOG.

**Example**  
LOG (R1C2)  
LOG (B3)  
LOG (115) = 2.0606978404

## MAX

**Name** Maximum value

**Description** Returns the maximum value of all arguments.

**Syntax** MAX(*value1, value2, . . .*)

**Remarks** Accepts up to 30 arguments. Each argument can be a cell range, a float value, or an integer value.

See also MIN.

**Example**  
MAX (R1C2, 5, R3C5, 7)  
MAX (A1, B2, C3, D4, E5)  
MAX (A1 : A9)  
MAX (2, 15, 12, 3, 7, 19, 4) = 19

**MEDIAN**

<b>Name</b>	Median
<b>Description</b>	Returns the number in the middle of the provided set of numbers; that is, half the numbers have values that are greater than the median, and half have values that are less.
<b>Syntax</b>	MEDIAN( <i>value1</i> , <i>value2</i> , . . . )
<b>Remarks</b>	Accepts up to 30 arguments. Each argument can be a cell range, a float value, or an integer value. If there is an even number of arguments, the function calculates the average of the two numbers in the middle.  See also AVERAGE, MODE, and RANK.
<b>Example</b>	MEDIAN (R1C2, R3C5, R4C7, R6C7) MEDIAN (A3, B5, C1, D4, E7) MEDIAN (A1 : A9) MEDIAN (89, 95, 76, 88, 92) = 89

**MIN**

<b>Name</b>	Minimum value
<b>Description</b>	Returns the minimum value of all arguments.
<b>Syntax</b>	MIN( <i>value1</i> , <i>value2</i> , . . . )
<b>Remarks</b>	Accepts up to 30 arguments. Each argument can be a cell range, a float value, or an integer value.  See also MAX.
<b>Example</b>	MIN (R1C2, R3C5, R4C7, R6C7) MIN (A3, B5, C1, D4, E7) MIN (A1 : A9) MIN (2, 15, 12, 3, 7, 19, 4) = 2

**MOD**

<b>Name</b>	Mod
<b>Description</b>	Returns the remainder after the <i>Number</i> argument (dividend) is divided by the <i>Divisor</i> argument.
<b>Syntax</b>	MOD( <i>Number</i> , <i>Divisor</i> )

## Appendix G

**Remarks** Arguments are as follows:

<b>Argument</b>	<b>Description</b>
<i>Number</i>	Number for which you want to find the remainder (dividend)
<i>Divisor</i>	Number by which you want to divide the <i>Number</i> argument

The remainder has the same sign as the *Divisor* argument.

**Example**  
MOD (R1C2, 12)  
MOD (B3, 10)  
MOD (C4, B2)  
MOD (255, 16) = 15  
MOD (-3, 2) = 1

## MODE

**Name** Mode

**Description** Returns the most frequently occurring value in a set of data.

**Syntax** MODE(*value1*, *value2*, . . . )

**Remarks** Accepts up to 30 arguments. Each argument can be a cell range, a float value, or an integer value.

See also AVERAGE, MEDIAN, and RANK.

**Example**  
MODE (R1C2, 12, 10, R2C3)  
MODE (A3, B3, C3, D3)  
MODE (A1:A9)  
MODE (89, 95, 88, 97, 88, 74) = 88

## NEG

**Name** Negative value

**Description** Changes the sign of the value. For example, NEG (-15) = 15.

**Syntax** NEG(*value*)

**Remarks** See also ABS and SIGN.

**Example**  
NEG (R1C2)  
NEG (A3)  
NEG (81) = -81

**NOT**

<b>Name</b>	Logical Not
<b>Description</b>	Reverses the logical value of its argument.
<b>Syntax</b>	NOT( <i>value</i> )
<b>Remarks</b>	If specified value is zero, then function returns 1 (True). If specified value is not zero, then function returns zero (False).  See also FALSE and TRUE.
<b>Example</b>	NOT (R1C2) NOT (A3) NOT (12) =0

**NPER**

<b>Name</b>	Number of periods																		
<b>Description</b>	Returns the number of periods for an investment based on a present value, future value, periodic payments, and a specified interest rate.																		
<b>Syntax</b>	NPER( <i>Rate, Pmt, Pv, Fv, Type</i> )																		
<b>Remarks</b>	Arguments are as follows: <table> <thead> <tr> <th><b>Argument</b></th> <th><b>Description</b></th> </tr> </thead> <tbody> <tr> <td><i>Rate</i></td> <td>Interest rate expressed as percentage (per period)</td> </tr> <tr> <td><i>Pmt</i></td> <td>Payment made each period; cannot change over life of the annuity</td> </tr> <tr> <td><i>Pv</i></td> <td>Present value</td> </tr> <tr> <td><i>Fv</i></td> <td>Future value This is an optional argument. If omitted, it is assumed to be 0.</td> </tr> <tr> <td><i>Type</i></td> <td>Indicates when payments are due This is an optional argument. If omitted, it is assumed that the payments are made at the end of the period. <table> <thead> <tr> <th><b>Set <i>Type</i> equal to</b></th> <th><b>If payments are due</b></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>At the end of the period</td> </tr> <tr> <td>1</td> <td>At the beginning of the period</td> </tr> </tbody> </table> </td> </tr> </tbody> </table>	<b>Argument</b>	<b>Description</b>	<i>Rate</i>	Interest rate expressed as percentage (per period)	<i>Pmt</i>	Payment made each period; cannot change over life of the annuity	<i>Pv</i>	Present value	<i>Fv</i>	Future value This is an optional argument. If omitted, it is assumed to be 0.	<i>Type</i>	Indicates when payments are due This is an optional argument. If omitted, it is assumed that the payments are made at the end of the period. <table> <thead> <tr> <th><b>Set <i>Type</i> equal to</b></th> <th><b>If payments are due</b></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>At the end of the period</td> </tr> <tr> <td>1</td> <td>At the beginning of the period</td> </tr> </tbody> </table>	<b>Set <i>Type</i> equal to</b>	<b>If payments are due</b>	0	At the end of the period	1	At the beginning of the period
<b>Argument</b>	<b>Description</b>																		
<i>Rate</i>	Interest rate expressed as percentage (per period)																		
<i>Pmt</i>	Payment made each period; cannot change over life of the annuity																		
<i>Pv</i>	Present value																		
<i>Fv</i>	Future value This is an optional argument. If omitted, it is assumed to be 0.																		
<i>Type</i>	Indicates when payments are due This is an optional argument. If omitted, it is assumed that the payments are made at the end of the period. <table> <thead> <tr> <th><b>Set <i>Type</i> equal to</b></th> <th><b>If payments are due</b></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>At the end of the period</td> </tr> <tr> <td>1</td> <td>At the beginning of the period</td> </tr> </tbody> </table>	<b>Set <i>Type</i> equal to</b>	<b>If payments are due</b>	0	At the end of the period	1	At the beginning of the period												
<b>Set <i>Type</i> equal to</b>	<b>If payments are due</b>																		
0	At the end of the period																		
1	At the beginning of the period																		

---

**Note** You must specify a value for the *Fv* argument if you are going to specify a value for the *Type* argument.

---

## Appendix G

Be sure to express the interest rate as per period. For example, if you make monthly payments on a loan at 8 percent interest, use 0.08/12 for the *Rate* argument.

For all the arguments, money paid out such as deposits in an investment, is represented by negative numbers; money you receive, such as dividend checks, is represented by positive numbers.

See the PV function for the formula used to calculate financial values.

See also FV, PMT, PV.

**Example** `NPER (R1C1/12, 50, 1000, 0, 1)`  
`NPER (0.005, -790, 90000, 0, 1)=167.7227522114`

## ODD

**Name** Odd integer

**Description** Rounds the specified value up to the nearest odd integer.

**Syntax** `ODD(value)`

**Remarks** Regardless of the sign of the number specified by the *value* argument, the number is rounded away from zero.

See also CEILING, EVEN, FLOOR, INT, ROUND, ROUNDUP, and TRUNCATE.

**Example** `ODD (R1C2)`  
`ODD (A3)`  
`ODD (4) =5`  
`ODD (-2.5) =-3`

## PERMUT

**Name** Permutation

**Description** Returns the number of possible permutations for a specified number of items.

**Syntax** `PERMUT(value1, value2)`

**Remarks** Specify the number of items using the *value1* argument. Specify the number of items in each possible permutation using the *value2* argument. Both arguments must be positive integers, and the *value1* argument must be greater than 0.

A permutation is any set or subset of items where internal order is significant. Contrast with *combinations* (COMBIN function).

The number of permutations is calculated as follows:

$$P_{k,n} = \frac{n!}{(n-k)!}$$

See also COMBIN.

**Example** PERMUT (R1C2, 2)  
 PERMUT (B3, 5)  
 PERMUT (C4, B2)  
 PERMUT (8, 2)=56  
 PERMUT (100, 3)=970200

## PI

**Name** Pi value  
**Description** Provides  $\pi$  as 3.1415926536.  
**Syntax** PI()  
**Remarks** This function does not accept arguments.  
**Example** PI ( )  
 DEGREES (PI ( ))=180

## PMT

**Name** Payment calculation  
**Description** Returns the payment amount given the present value, specified interest rate, and number of terms.  
**Syntax** PMT(*Amount, Interest, Terms, TermsPerYear*)

## Appendix G

**Remarks** Arguments are as follows:

<b>Argument</b>	<b>Description</b>
<i>Amount</i>	Present value (for example, loan amount)
<i>Interest</i>	Interest rate expressed as annual percentage
<i>Terms</i>	Total number of terms ( <i>TermsPerYear</i> multiplied by the number of years of the loan)
<i>TermsPerYear</i>	Number of payments per year

Be sure to express the interest rate as per annum. For example, if the interest rate is 8 percent, use 8 for the *Interest* argument.

See the PV function for the formula used to calculate financial values.

See also FV, NPER, PV.

**Example** `PMT (R1C2, 8, 16, 4)`  
`PMT (20000, 6, 120, 12) = 222.0500000000`

## POWER

**Name** Raises number to a power

**Description** Raises the specified number to the given power.

**Syntax** `POWER(value1, value2)`

**Remarks** Specify the number to raise using the *value1* argument. Specify the power to raise it to using the *value2* argument.

You can use the ^ operator instead of the POWER function to raise a number to a power, for example,  $16^3$ .

See also EXP.

**Example** `POWER (R1C2, 3)`  
`POWER (A3, B4)`  
`POWER (16, 3) = 4096`

## PRODUCT

**Name** Product

**Description** Multiplies all the numbers given as arguments and returns the product.

**Syntax** `PRODUCT(value1, value2, . . . )`

**Remarks** Accepts up to 30 arguments. Each argument can be a cell range, a float value, or an integer value.

See also FACT.

**Example**  
 PRODUCT (R1C2, 2, 10)  
 PRODUCT (B3, B7, 12)  
 PRODUCT (C4, B2, B4, C5)  
 PRODUCT (A1 : A9)  
 PRODUCT (1, 2, 3, 5, 7, 11, 13) = 30030

## PV

**Name** Present value

**Description** Returns the present value of an investment based on the interest rate, number and amount of periodic payments, and future value. The present value is the total amount that a series of future payments is worth now.

**Syntax** PV(*Rate*, *Nper*, *Pmt*, *Fv*, *Type*)

**Remarks** Arguments are as follows:

Argument	Description
<i>Rate</i>	Interest rate expressed as percentage (per period)
<i>Nper</i>	Total number of payment periods
<i>Pmt</i>	Payment made each period; cannot change over the life of the annuity
<i>Fv</i>	Future value This is an optional argument. If omitted, it is assumed that the calculation is based on the payments.
<i>Type</i>	Indicates when payments are due This is an optional argument. If omitted, it is assumed that the payments are made at the end of the period.
	<b>Set <i>Type</i> equal to</b> <b>If payments are due</b>
	0                              At the end of the period
	1                              At the beginning of the period

---

**Note** You must specify a value for the *Fv* argument if you are going to specify a value for the *Type* argument.

---

Use consistent units for specifying the *Rate* and *Nper* arguments. If you make monthly payments on a five-year loan at 8 percent annual interest, use

## Appendix G

0.08/12 for the *Rate* argument and 5\*12 for the *Nper* argument. If you make annual payments on the same loan, use 0.08 for *Rate* and 5 for *Nper*.

For all the arguments, money paid out such as deposits in an investment, is represented by negative numbers; money you receive, such as dividend checks, is represented by positive numbers.

For all the financial functions (FV, NPER, PMT, and PV), the spreadsheet uses the following formula and solves for the required variable:

$$pv \times (1 + rate)^{nper} + pmt(1 + rate \times type) \times \left( \frac{(1 + rate)^{nper} - 1}{rate} \right) + fv = 0$$

If rate is 0, then:

$$(pmt \times nper) + pv + fv = 0$$

See also FV, NPER, PMT.

**Example** PV(R1C1/12, 48, R1C2, 0, 0)  
PV(0.005, 60, -100, 0, 1)=5198.4188554887

## RADIANS

<b>Name</b>	Radians
<b>Description</b>	Converts the specified number from degrees to radians.
<b>Syntax</b>	RADIANS( <i>value</i> )
<b>Remarks</b>	See also DEGREES.
<b>Example</b>	RADIANS (R1C2) RADIANS (B3) RADIANS (45)=0.7853981634= $\pi/4$

## RAND

<b>Name</b>	Random number
<b>Description</b>	Returns an evenly distributed random number greater than or equal to 0 and less than 1. The RAND function returns a new random number every time the sheet is calculated.

<b>Syntax</b>	RAND()
<b>Remarks</b>	<p>This function does not accept arguments.</p> <p>To generate a random real number between <math>x</math> and <math>y</math>, with <math>y</math> greater than <math>x</math>, use the following expression:</p> $\text{RAND}() * (y-x) + x$ <p>To generate a random integer between <math>x</math> and <math>y</math>, with <math>y</math> greater than <math>x</math>, use the following expression:</p> $\text{int}((y-x+1) * \text{RAND}() + x)$
<b>Example</b>	<p>RAND()</p> <p>RAND() * 100</p>

## RANK

<b>Name</b>	Ranks values
<b>Description</b>	Returns the rank of a number in a set of numbers. If you were to sort the set, the rank of the number would be its position in the list.
<b>Syntax</b>	RANK( <i>Number, Ref, Order</i> )
<b>Remarks</b>	Arguments are as follows:

<b>Argument</b>	<b>Description</b>						
<i>Number</i>	Number whose rank you want to return						
<i>Ref</i>	Reference to the set of numbers						
<i>Order</i>	How to rank the number						
	This is an optional argument. If omitted, the function ranks the number as if the set of numbers was sorted in descending order.						
	<table border="0" style="width: 100%;"> <thead> <tr> <th style="text-align: left;"><b>Set Order to</b></th> <th style="text-align: left;"><b>To rank as if numbers are sorted</b></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>In descending order</td> </tr> <tr> <td>Nonzero</td> <td>In ascending order</td> </tr> </tbody> </table>	<b>Set Order to</b>	<b>To rank as if numbers are sorted</b>	0	In descending order	Nonzero	In ascending order
<b>Set Order to</b>	<b>To rank as if numbers are sorted</b>						
0	In descending order						
Nonzero	In ascending order						

The RANK function gives duplicate numbers the same rank. The presence of duplicate numbers affects the ranks of subsequent numbers. For example, in a list of integers, if the number 12 appears twice and has a rank of 4, then 13 would have a rank of 6 (no number would have a rank of 5).

See also MEDIAN and MODE.

## Appendix G

**Example**     RANK (12, R1C2, 0)  
                 RANK (B3, B1:B8, 1)

## ROUND

<b>Name</b>	Round
<b>Description</b>	Rounds the a specified value to the nearest number, using the specified number of decimal places.
<b>Syntax</b>	ROUND( <i>value1</i> , <i>value2</i> )
<b>Remarks</b>	<p>Use the <i>value1</i> argument to specify the number to round. Use the <i>value2</i> argument to specify the number of decimal places. Set <i>value2</i> to a value greater than zero to round to the specified number of decimal places. Set <i>value2</i> to zero to round to the nearest whole number. Set <i>value2</i> to a value less than zero to round the value left of the decimal to the nearest ten, hundred, etc.</p> <p>See also CEILING, EVEN, FLOOR, INT, ODD, ROUNDUP, and TRUNCATE.</p>
<b>Example</b>	<pre>ROUND (R1C2, 2) ROUND (A3, -2) ROUND (C4, B2) ROUND (PI (), 5) = 3.14159 ROUND (29.2, -2) = 0 ROUND (-1.963, 0) = -2</pre>

## ROUNDUP

<b>Name</b>	Round up
<b>Description</b>	Rounds the specified number up to the nearest number, using the specified number of decimal places.
<b>Syntax</b>	ROUNDUP( <i>value1</i> , <i>value2</i> )
<b>Remarks</b>	<p>Use the <i>value1</i> argument to specify the number to round. Use the <i>value2</i> argument to specify the number of decimal places. Set <i>value2</i> to a value greater than zero to round to the specified number of decimal places. Set <i>value2</i> to zero to round to the nearest whole number. Set <i>value2</i> to a value less than zero to round the value left of the decimal to the nearest ten, hundred, etc. Regardless of the sign of the number specified by the <i>value1</i> argument, the number is rounded away from zero</p> <p>See also CEILING, EVEN, FLOOR, INT, ODD, ROUND, and TRUNCATE.</p>

## Appendix G

**Example**    `ROUNDUP (R1C2, 2)`  
              `ROUNDUP (A3, -2)`  
              `ROUNDUP (C4, B2)`  
              `ROUNDUP (PI (), 5) = 3.14160`  
              `ROUNDUP (29.2, -2) = 100`  
              `ROUNDUP (-1.963, 0) = -2`

## SIGN

**Name**        Sign of a number

**Description**    Returns the sign of a number.

**Syntax**        `SIGN(value)`

**Remarks**      Returns 1 if the number is positive, 0 if the number is 0, and -1 if the number is negative.

                  See also ABS and NEG.

**Example**        `SIGN (R1C2)`  
                  `SIGN (B3)`  
                  `SIGN (-5) = -1`

## SIN

**Name**        Sine

**Description**    Returns the sine of the specified angle.

**Syntax**        `SIN(value)`

**Remarks**      Specify the angle in radians using the *value* argument. If the angle is in degrees, multiply it by  $\pi/180$  to convert it to radians.

                  See also ASIN, COS, SINH, and TAN.

**Example**        `SIN (R1C2)`  
                  `SIN (B4)`  
                  `SIN (30*PI () / 180) = 0.5`

## SINH

**Name**        Hyperbolic sine

**Description**    Returns the hyperbolic sine of the specified number.

**Syntax**        `SINH(value)`

**Remarks** You can use any real number for the *value* argument.

The formula used to calculate the hyperbolic sine is:

$$\text{SINH}(z) = \frac{e^z - e^{-z}}{2}$$

See also ASIN, COSH, SIN, and TANH.

**Example**  
 SINH (R1C2)  
 SINH (B4)  
 SINH (1) = 1.1752011936

## SLN

**Name** Straight-line depreciation

**Description** Returns the straight-line depreciation of an asset for one period.

**Syntax** SLN(*Cost, Salvage, Life*)

**Remarks** Arguments are as follows:

Argument	Description
<i>Cost</i>	Initial cost of the asset
<i>Salvage</i>	Value at the end of the depreciation
<i>Life</i>	Number of periods over which the asset is being depreciated

See also DB, DDB, and SYD.

**Example**  
 SLN (R1C2, 1000, 10)  
 SLN (500000, 20000, 5) = 96000

## SQRT

**Name** Square root

**Description** Returns the positive square root of the specified number.

**Syntax** SQRT(*value*)

## Appendix G

**Remarks** You must provide a positive number for the *value* argument.

See also POWER and XROOT.

**Example** Sqrt (R1C3)  
Sqrt (B4)  
Sqrt (256) =16

## SQUARE

**Name** Square

**Description** Returns the square of the specified number.

**Syntax** SQUARE(*value*)

**Example** SQUARE (R1C3)  
SQUARE (B4)  
SQUARE (10.5) =110.25

## STDEV

**Name** Standard deviation

**Description** Returns the standard deviation for a set of numbers.

**Syntax** STDEV(*value1, value2, . . .*)

**Remarks** The standard deviation is a measure of how widely values are dispersed from the average value.

Accepts up to 30 arguments. Each argument can be a cell range, a float value, or an integer value.

The standard deviation is calculated using the “nonbiased” or “n-1” method. The function uses the following formula to calculate the standard deviation:

$$\sqrt{\frac{n \sum x^2 - (\sum x)^2}{n(n-1)}}$$

The STDEV function assumes that its arguments are a sample of the population. If your data represents the entire population, then compute the standard deviation using the STDEVP function.

See also AVERAGE, STDEVP, VAR, and VARP.

**Example** STDEV (R1C2, R3C4, R4C5, R7C2)  
 STDEV (A1, B2, C3, D4, E5, F6)  
 STDEV (A1 : A9)  
 STDEV (95, 89, 73, 87, 85, 76, 100, 96, 96) = 9.3422576382

## STDEVP

**Name** Standard deviation for population

**Description** Calculates standard deviation based on the entire population given as arguments.

**Syntax** STDEVP(*value1*, *value2*, . . . )

**Remarks** The standard deviation is a measure of how widely values are dispersed from the average value.

Accepts up to 30 arguments. Each argument can be a cell range, a float value, or an integer value.

The standard deviation is calculated using the “biased” or “n” method. The function uses the following formula to calculate the standard deviation:

$$\sqrt{\frac{n \sum x^2 - (\sum x)^2}{n^2}}$$

The STDEVP function assumes that its arguments are the entire population. If your data represents a sample of the population, then compute the standard deviation using the STDEV function.

See also AVERAGE, STDEV, VAR, and VARP.

**Example** STDEVP (R1C2, R3C4, R4C5, R7C2)  
 STDEVP (A1, B2, C3, D4, E5, F6)  
 STDEVP (A1 : A9)  
 STDEVP (95, 89, 73, 87, 85, 76, 100, 96, 96) = 8.8079649700

**SUM**

<b>Name</b>	Summation
<b>Description</b>	Sums cells or blocks.
<b>Syntax</b>	SUM( <i>value1</i> , <i>value2</i> , . . . )
<b>Remarks</b>	Accepts up to 30 arguments. Each argument can be a cell range, a float value, or an integer value.  See also ADD and SUMSQ.
<b>Example</b>	SUM(R1C2, R3C5, R6C2, R1C7) SUM(A1, B7, C11) SUM(A1:A9) SUM(95, 89, 73, 87, 85, 76, 100, 96, 96) = 797

**SUMSQ**

<b>Name</b>	Sum of the squares
<b>Description</b>	Returns the sum of the squares of the arguments.
<b>Syntax</b>	SUMSQ( <i>value1</i> , <i>value2</i> , . . . )
<b>Remarks</b>	Accepts up to 30 arguments. Each argument can be a cell range, a float value, or an integer value.  See also SUM.
<b>Example</b>	SUMSQ(R1C2, R3C5, R6C2, R1C7) SUMSQ(A1, B7, C11) SUMSQ(A1:A9) SUMSQ(95, 89, 73, 87, 85, 76, 100, 96, 96) = 71277

**SYD**

<b>Name</b>	Sum-of-years' digits depreciation
<b>Description</b>	Returns the sum-of-years' digits depreciation of an asset for a specified period.
<b>Syntax</b>	SYD( <i>Cost</i> , <i>Salvage</i> , <i>Life</i> , <i>Per</i> )

**Remarks** Arguments are as follows:

Argument	Description
<i>Cost</i>	Initial cost of the asset
<i>Salvage</i>	Value at the end of the depreciation
<i>Life</i>	Number of periods over which the asset is being depreciated
<i>Per</i>	The period for depreciation Must use the same units as the <i>Life</i> argument.

The SYD calculates the standard deviation as follows:

$$SYD = \frac{(Cost - Salvage) \times (Life - Per + 1) \times 2}{(Life)(Life + 1)}$$

See also DB, DDB, and SLN.

**Example** SYD(R1C2, 1000, 10, 1)  
SYD(100000, 10000, 5, 2) = 24000

## TAN

<b>Name</b>	Tangent
<b>Description</b>	Returns the tangent of the specified angle.
<b>Syntax</b>	TAN( <i>value</i> )
<b>Remarks</b>	Specify the angle in radians using the <i>value</i> argument. If the angle is in degrees, multiply it by $\pi/180$ to convert it to radians.

See also ATAN, ATAN2, COS, SIN, and TANH.

**Example** TAN(R1C2)  
TAN(B3)  
TAN(45\*PI()/180) = 1

## TANH

<b>Name</b>	Hyperbolic tangent
<b>Description</b>	Returns the hyperbolic tangent of the specified number.
<b>Syntax</b>	TANH( <i>value</i> )

## Appendix G

**Remarks** You can specify any real number for the *value* argument.

The formula for the hyperbolic tangent is:

$$\text{TANH}(z) = \frac{\text{SINH}(z)}{\text{COSH}(z)}$$

See also ATAN, ATAN2, COSH, SINH, and TAN.

**Example** `TANH (R1C2)`  
`TANH (B3)`  
`TANH (0.5) = 0.4621171573`

## TRUE

**Name** True

**Description** Returns the logical value TRUE.

**Syntax** TRUE()

**Remarks** This function does not accept arguments.

See also FALSE and NOT.

**Example** `TRUE ()`

## TRUNCATE

**Name** Truncate

**Description** Truncates the specified value using the specified number of decimal places.

**Syntax** TRUNCATE(*value1*, *value2*)

**Remarks** Use the *value1* argument to specify the number to truncate. Use the *value2* argument to specify the number of decimal places. Set *value2* to a value greater than zero to truncate to the specified number of decimal places. Set *value2* to zero to truncate to the nearest whole number. Set *value2* to a value less than zero to round the value left of the decimal to the nearest ten, hundred, etc.

The TRUNCATE and INT functions are similar, but the TRUNCATE function removes the decimal portion of the number, while INT rounds the number based on the value in the decimal portion.

See also CEILING, EVEN, FLOOR, INT, ODD, ROUND, and ROUNDUP.

**Example** TRUNCATE (R1C2, 2)  
 TRUNCATE (A3, -2)  
 TRUNCATE (C4, B2)  
 TRUNCATE (PI (), 5) = 3.14159  
 TRUNCATE (29.2, -2) = 0  
 TRUNCATE (-1.963, 0) = -1

## URL

**Name** URL

**Description** Creates a hypertext link in the exported HTML file.

**Syntax** URL(*value1*, *value2*)

**Remarks** Use the *value1* argument to use the specified value as a URL address in a hypertext link in the exported HTML table.

If you specify the address without a transport protocol (http://, ftp://, news://, and so on), Spread adds a default protocol of http:// when it exports the spreadsheet. For example, if *value1* is www.fpoint.com, the address is exported as http://www.fpoint.com. The browser displays www.fpoint.com. For best results, use a complete URL address.

Use the optional *value2* argument if you want to display the link as a descriptive name. For example, if *value1* is http://www.fpoint.com and *value2* is FarPoint, the browser will display the link as FarPoint.

**Example** vaSpread1.Formula = "URL (A1, A2) "  
 vaSpread1.Formula = "URL (" + Chr (34) +  
 ➔ "http://www.fpoint.com" + Chr (34) + ", " + Chr (34)  
 ➔+ "FarPoint" + Chr (34) + ") "

## VAR

**Name** Variance

**Description** Calculates variance based on a sample of a population.

**Syntax** VAR(*value1*, *value2*, . . . )

**Remarks** Accepts up to 30 arguments. Each argument can be a cell range, a float value, or an integer value.

The VAR function assumes that its arguments are a sample of the population. If your data represents the entire population, then compute the variance using the VARP function.

## Appendix G

The VAR function uses the following formula to calculate the variance:

$$\frac{n \sum x^2 - (\sum x)^2}{n(n-1)}$$

See also AVERAGE, STDEV, STDEVP, and VARP.

**Example** VAR (R1C2, 100, R2C5, 102)  
VAR (B3, C4, B2, D10, E5)  
VAR (A1 : A9)  
VAR (98, 85, 76, 87, 92, 89, 90) = 45.8095238095

## VARP

**Name** Variance of population

**Description** Calculates variance based on the entire population.

**Syntax** VARP(*value1*, *value2*, . . .)

**Remarks** Accepts up to 30 arguments. Each argument can be a cell range, a float value, or an integer value.

The VARP function assumes that its arguments are the entire population. If your data represents a sample of the population, then compute the variance using the VAR function.

The VARP function uses the following formula to calculate the variance:

$$\frac{n \sum x^2 - (\sum x)^2}{n^2}$$

See also AVERAGE, STDEV, STDEVP, and VAR.

**Example** VARP (R1C2, 100, R2C5, 102)  
VARP (B3, C4, B2, D10, E5)  
VARP (A1 : A9)  
VARP (98, 85, 76, 87, 92, 89, 90) = 39.2653061224

## XROOT

<b>Name</b>	Root Y of value X
<b>Description</b>	Returns the root, Y, of a specified number, X ( $X^{1/Y}$ ).
<b>Syntax</b>	XROOT( <i>value1</i> , <i>value2</i> )
<b>Remarks</b>	<p>Specify the number for which you want the root using the <i>value1</i> argument. Specify the root you want using the <i>value2</i> argument. If <i>value2</i> is an even number, <i>value1</i> must be a positive number.</p> <p>See also POWER and SQRT.</p>
<b>Example</b>	<pre>XROOT (R1C2, 3) XROOT (A3, 4) XROOT (C4, B2) XROOT (256, 3)=6.3496042079</pre>



# Appendix H Using DLL Controls

You can use the Spread DLL controls in the C programming language using the provided dynamic-link library (.DLL) files. You can also use the .DLL files in the C++ programming language. This appendix explains the following topics:

- Working with libraries
- Creating the Spread controls
- Retrieving and setting property values
- Working with string data

## Working with Libraries

To use the Spread controls with C, you must first load the .DLL and .LIB file for this control. Before you exit your application, you should unload the library.

### Loading the Library

To load a library, you must declare a variable of HINSTANCE and then use a Windows API call. The Windows API call loads the control's .DLL file into memory and returns the handle.

The following code loads the Spread .DLL file:

```
HINSTANCE hInstSpread;  
  
hInstSpread = LoadLibrary("spread_filename.DLL");
```

### Unloading the Library

Unload .DLL files using the FreeLibrary command, which is a Windows API call. Specify the .DLL file by using the assigned identifier from the LoadLibrary command. In the previous code example, the variable for the Spread control .DLL file is *hInstSpread*.

The following code frees the library from memory:

```
FreeLibrary(hInstSpread);
```

## Creating the fpSpread Control or fpSpread Preview Control

Once you have loaded the library, you can create the fpSpread or fpSpreadPreview control in your application. You can create the control in one of two ways:

- The **CreateWindow** function
- A **CONTROL** statement in a resource (.RC) file

### Creating Controls with the CreateWindow Function

Use the Windows **CreateWindow** function to create the control. The **CreateWindow** function specifies the window class, the window title, the window style, the initial position and size of the window, the window's parent, and the menu.

The following code creates an fpSpread control:

```
hWndSpread = CreateWindow(
    SS_CLASSNAME,           // window class name
    "",                    // window caption
    WS_CHILD | WS_VISIBLE  // window style
    Rect.left,              // initial x position
    Rect.top,               // initial y position
    Rect.right - Rect.left, // initial x size
    Rect.bottom - Rect.top, // initial y size
    hWnd,                  // parent window handle
    IDC_SPREAD,            // window ID
    hInstance,             // program instance handle
    NULL                   // creation parameter
);
```

The **CreateWindow** function creates controls one at a time. Use the AppStudio® in Visual C++® or the Resource Workshop® in Borland® C++ to create a control that you can bring up on a dialog.

### Creating Controls with a CONTROL Statement

You can create an fpSpread or fpSpreadPreview control using a **CONTROL** statement in a resource file in a dialog box. The **CONTROL** statement defines a control by using arguments separated by commas. These arguments specify the class of the control, give the control an identifier, define the control style, and define the position and dimensions of the control within the parent dialog.

The following statement creates an fpSpread control:

```
CONTROL "", IDC_SPREAD, SS_CLASSNAME, WS_BORDER | WS_CHILD |
↳WS_VISIBLE | WS_TABSTOP, 5, 35, 165, 60
```

Note that the arguments in the CONTROL statement are similar to the arguments in the **CreateWindow** function. However, you do not have to specify the parent window handle or the program instance handle in the CONTROL statement.

## Using Get and Set Functions

Once you have loaded the .DLL file and created an fpSpread or fpSpreadPreview control, you can use the specific Get and Set functions to retrieve and set characteristics of the control.

The DLL Get and Set functions are declared in the header files that accompany this product. The header files are installed when you install Spread using the installation program. For information about Get and Set functions, refer to the function reference in the online *DLL Reference Guide*.

## Working with String Data

When you use the Get or Set function for a string value, you must supply a buffer to indicate the array of available characters for the string, as in the following code:

```
char szbuffer[#];
```

Replace the # sign with the number of characters available in the array. For example, the **SSGetButtonText** function for the Spread control requires a string value.

The first statement in the following example indicates the buffer can contain up to 39 characters (with an additional character for the NULL terminator).

```
char szBuffer[40];
SSGetButtonText(hWndSS, 1, 2, szBuffer);
SSSetButtonText(hWndSS, 1, 2, "I am the button text");
```

If you do not know the exact length of the string, use code similar to the following example.

```
HANDLE hBuffer;
SHORT nLen;
nLen = SSGetButtonText(hWndSS, 1, 2, NULL);
```

## Appendix H

```
If (hBuffer = GlobalAlloc(GHND, nLen + 1))
{
    LPSTR lpszBuffer = GlobalLock(hBuffer);
    SSGetButtonText(hWndSS, 1, 2, lpszBuffer);
    GlobalUnlock(hBuffer);
    GlobalFree(hBuffer);
}
```

# Appendix I Excel-Formatted File Import/Export

When you import or export an Excel-formatted file, you are given the option to create a log file that records the manner in which Spread handled the import or export of features, including cell types and other settings. If you choose to create that log file, you can view it to find out more details about the file import or export.

The following sections provide a summary of the manner in which Spread deals with importing and exporting Excel-formatted files. The information about Excel and Excel-formatted files in the following sections is based on Excel 97 and the BIFF8 format.

For instructions on importing Excel-formatted files, see “Loading an Existing Spreadsheet” on page 81. For instructions on exporting from Spread to an Excel-formatted file, see “Exporting a Spreadsheet to an Excel-Formatted File” on page 117.

## Excel-Formatted File Import

When Spread imports an Excel-formatted file, it sets properties using the spreadsheet and cell definitions in the Excel-formatted file, as listed in the following table.

### Excel-Formatted File Import

Excel Formatting	Spread Property	Discussion	Log #
<b>Spreadsheet settings</b>			
Scroll bars	ScrollBars	ScrollBars property set to same setting as in Excel.	
Column and row headers	DisplayColHeaders DisplayRowHeaders	Excel shows or hides both column and row headers; therefore, if both are displayed in Excel, both are displayed in Spread (DisplayColHeaders and DisplayRowHeaders are both set to True).	
Grid lines	GridShowHoriz GridShowVert	Excel shows or hides both horizontal and vertical grid lines; therefore, if both are displayed in Excel, both are displayed in Spread (GridShowHoriz and GridShowVert are both set to True).	

**Excel-Formatted File Import (Continued)**

<b>Excel Formatting</b>	<b>Spread Property</b>	<b>Discussion</b>	<b>Log #</b>
Enter key action	EditEnterAction	Excel stores this setting in the system registry. Spread reads the value of this setting, and sets the EditEnterAction property. The log file issues a message.	102
Clipboard shortcut key support	AutoClipboard	Excel stores this setting in the system registry. Spread reads the value of this setting, and sets the AutoClipboard property. The log file issues a message.	102
Cell overflow	AllowCellOverflow	Spread sets the AllowCellOverflow property to True. (Cells can always overflow in Excel.)	
Drag and drop	AllowDragDrop	Excel stores this setting in the system registry. Spread reads the value of this setting, and sets the AllowDragDrop property. The log file issues a message.	102
Print headers	PrintColHeaders and PrintRowHeaders	Excel allows users either to print both the column and row headers or to not print any headers. Therefore, if the Excel file is set to print both column and row headers, the PrintColHeaders and PrintRowHeaders properties in Spread are set to True.	
Print grid lines	PrintGrid	If the Excel file is set to print grid lines, the PrintGrid property in Spread is set to True.	
<b>Columns and Rows</b>			
Default row height	RowHeight	Spread reads the default row height used in Excel, then sets the RowHeight property for each row using that value.	
Default column width	ColWidth	Spread reads the default column width used in Excel, then sets the ColWidth property for each column using that value.	

**Excel-Formatted File Import (Continued)**

<b>Excel Formatting</b>	<b>Spread Property</b>	<b>Discussion</b>	<b>Log #</b>
Row height	RowHeight	Spread reads the row heights used in Excel, then sets the RowHeight property for individual rows using those values.	
Column width	ColWidth	Spread reads the column widths used in Excel, then sets the ColWidth property for individual columns using those values.	
Hidden columns	ColHidden	Columns hidden in Excel are hidden in Spread.	
Hidden rows	RowHidden	Rows hidden in Excel are hidden in Spread.	
<b>Cell Settings</b>			
Active cell	ActiveCol ActiveRow	Sets active cell in Spread as it was when file was saved in Excel	
Cell blocks		Sets selected cell blocks in Spread as they were when file was saved in Excel	
Cell color/pattern	BackColor	Spread reads the color setting from Excel and assigns color based on the setting of each cell's individual setting, the cell's row setting, the cell's column setting, and finally, the spreadsheet setting. Spread does not support background patterns, so any background pattern is ignored.	
Fonts Font type, style, and size	Font FontBold FontItalic FontName FontSize FontStrikethru	Spread reads the font settings from Excel and assigns values to these properties based on the setting of each cell's individual setting, the cell's row setting, the cell's column setting, and finally, the spreadsheet setting.	

**Excel-Formatted File Import (Continued)**

<b>Excel Formatting</b>	<b>Spread Property</b>	<b>Discussion</b>	<b>Log #</b>
Fonts Font underline	FontUnderline	<p>Spread reads the underline settings from Excel and assigns values to these properties based on the setting of each cell's individual setting, the cell's row setting, the cell's column setting, and finally, the spreadsheet setting.</p> <p>Excel provides three underline settings that Spread does not: Double, Single Accounting, and Double Accounting. Spread sets the FontUnderline property to True for those settings.</p>	
Fonts Subscript and superscript	Not applicable	Spread does not support these font settings, so the text is displayed as normal text.	
Vertical text alignment in cells	TypeVAlign	Spread uses the Excel setting.	
Horizontal text alignment in cells	TypeHAlign	Spread uses the Excel setting.	
Locked cells	Lock Protect	Spreadsheets imported from Excel import the Lock property setting for each cell and import the Protect property setting. Note that the default value for the Protect property in Excel is False—the opposite of the Spread Protect property default setting—and the default value of the Lock property is True for all cells in Excel.	
Borders	CellBorderColor	Spread sets the color to the color used in Excel.	

**Excel-Formatted File Import (Continued)**

Excel Formatting	Spread Property	Discussion	Log #																														
Borders	CellBorderStyle	<p>Excel provides border styles that Spread does not. The following list explains how Spread sets the CellBorderStyle property for Excel border styles:</p> <table border="0"> <thead> <tr> <th>Excel:</th> <th>Spread</th> </tr> </thead> <tbody> <tr><td>None</td><td>6 - Blank</td></tr> <tr><td>Dotted</td><td>13 - Fine Dot</td></tr> <tr><td>Dashed</td><td>12 - Fine Dash</td></tr> <tr><td>Dash-dot-dot</td><td>15 - Fine Dash Dot Dot</td></tr> <tr><td>Dash-dot</td><td>14 - Fine Dash Dot</td></tr> <tr><td>Hair</td><td>12 - Fine Dash</td></tr> <tr><td>Thin</td><td>11 - Fine Solid</td></tr> <tr><td>Medium dash-dot-dot</td><td>5 - Dash Dot Dot</td></tr> <tr><td>Slanted dash-dot</td><td>1 - Solid</td></tr> <tr><td>Medium dash-dot</td><td>4 - Dash Dot</td></tr> <tr><td>Medium dashed</td><td>2 - Dash</td></tr> <tr><td>Medium</td><td>1 - Solid</td></tr> <tr><td>Thick</td><td>1 - Solid</td></tr> <tr><td>Double</td><td>1 - Solid</td></tr> </tbody> </table>	Excel:	Spread	None	6 - Blank	Dotted	13 - Fine Dot	Dashed	12 - Fine Dash	Dash-dot-dot	15 - Fine Dash Dot Dot	Dash-dot	14 - Fine Dash Dot	Hair	12 - Fine Dash	Thin	11 - Fine Solid	Medium dash-dot-dot	5 - Dash Dot Dot	Slanted dash-dot	1 - Solid	Medium dash-dot	4 - Dash Dot	Medium dashed	2 - Dash	Medium	1 - Solid	Thick	1 - Solid	Double	1 - Solid	
Excel:	Spread																																
None	6 - Blank																																
Dotted	13 - Fine Dot																																
Dashed	12 - Fine Dash																																
Dash-dot-dot	15 - Fine Dash Dot Dot																																
Dash-dot	14 - Fine Dash Dot																																
Hair	12 - Fine Dash																																
Thin	11 - Fine Solid																																
Medium dash-dot-dot	5 - Dash Dot Dot																																
Slanted dash-dot	1 - Solid																																
Medium dash-dot	4 - Dash Dot																																
Medium dashed	2 - Dash																																
Medium	1 - Solid																																
Thick	1 - Solid																																
Double	1 - Solid																																
Borders	CellBorderType	<p>Excel provides border types that Spread does not. The following list explains how Spread sets the CellBorderType property for Excel border types:</p> <table border="0"> <thead> <tr> <th>Excel:</th> <th>Spread</th> </tr> </thead> <tbody> <tr><td>None</td><td>0 - None</td></tr> <tr><td>Outline</td><td>16 - Outline</td></tr> <tr><td>Inside (block)</td><td>1 Or 2 Or 4 Or 8*</td></tr> <tr><td>Top</td><td>4 - Top</td></tr> <tr><td>Between horizontal (block)</td><td>4 Or 8*</td></tr> <tr><td>Bottom</td><td>8 - Bottom</td></tr> <tr><td>Diagonal /</td><td>0 - None</td></tr> <tr><td>Left</td><td>1 - Left</td></tr> <tr><td>Between vertical (block)</td><td>1 Or 2*</td></tr> <tr><td>Right</td><td>2 - Right</td></tr> <tr><td>Diagonal \</td><td>0 - None</td></tr> </tbody> </table> <p>* Logically Ors values together as indicated, but excludes outer edge of block</p>	Excel:	Spread	None	0 - None	Outline	16 - Outline	Inside (block)	1 Or 2 Or 4 Or 8*	Top	4 - Top	Between horizontal (block)	4 Or 8*	Bottom	8 - Bottom	Diagonal /	0 - None	Left	1 - Left	Between vertical (block)	1 Or 2*	Right	2 - Right	Diagonal \	0 - None							
Excel:	Spread																																
None	0 - None																																
Outline	16 - Outline																																
Inside (block)	1 Or 2 Or 4 Or 8*																																
Top	4 - Top																																
Between horizontal (block)	4 Or 8*																																
Bottom	8 - Bottom																																
Diagonal /	0 - None																																
Left	1 - Left																																
Between vertical (block)	1 Or 2*																																
Right	2 - Right																																
Diagonal \	0 - None																																

## Excel-Formatted File Import (Continued)

Excel Formatting	Spread Property	Discussion	Log #
<b>Cell Types</b>			
Cell formats	CellType	Excel cell formats are interpreted as a similar Spread cell type, when possible. The following rows describe specific settings for cell formats. Spread converts all non-object cell formats.	
General cell format Text cell format	TypeEditMultiLine	Spread uses the Excel setting.	
Number cell format Currency cell format Accounting cell format Scientific notation cell format Percent cell format Fraction cell format	TypeFloatDecimalPlaces	Excel rounds float values, but Spread truncates them. Therefore, be careful when working with values that might be truncated in Spread.  Excel maintains the precision of float values regardless of the cell format that displays them, but Spread does not. Therefore, be careful when working with values that might lose precision in Spread. Note that an integer format in Excel will maintain the precision of a float value in the cell, but Spread will save that value as an integer and the decimal portion will be lost. The log file issues a message	121

## Excel-Formatted File Import (Continued)

Excel Formatting	Spread Property	Discussion	Log #
Number cell format Currency cell format Accounting cell format Scientific notation cell format Percent cell format Fraction cell format	TypeFloatCurrencyChar TypeFloatMoney	Spread does not support more than one character for the currency symbol. If the Excel-formatted spreadsheet uses a currency symbol with more than one character, Spread uses the first letter of the character as the currency symbol. The log file issues message number 138.  Spread does not support displaying currency symbols anywhere other than to the left of the currency value. If the Excel-formatted spreadsheet displays the currency symbol in a place other than to the left of the currency value, the symbol is moved and displayed on the left side. The log file issues message number 139.  Spread does not allow you to align the currency symbol, and therefore it ignores the Excel alignment.	138 or 139
Number cell format Currency cell format Accounting cell format Scientific notation cell format Percent cell format Fraction cell format	TypeFloatDecimalChar	Excel reads this from the system's international or regional settings, as does Spread.	
Number cell format Currency cell format Accounting cell format Scientific notation cell format Percent cell format Fraction cell format	TypeFloatSeparator TypeFloatSepChar	If Excel does not display a separator character, the TypeFloatSeparator property is set to False, and Spread does not display a separator character. If Excel displays a separator character, it reads it from the system's international or regional settings. In that case, the TypeFloatSeparator property in Spread is set to True, and Spread reads the character from the system's international or regional settings.	

## Excel-Formatted File Import (Continued)

Excel Formatting	Spread Property	Discussion	Log #
Number cell format Currency cell format Accounting cell format Scientific notation cell format Percent cell format Fraction cell format	TypeFloatMax TypeFloatMin	<p>The maximum allowed value in Spread is 999999999999999. The maximum allowed value in Excel is 9.999999999999999E307. If the cell's value in Excel exceeds the maximum value allowed in Spread, the log file issues a message and Spread creates an empty cell.</p> <p>The minimum allowed value in Spread is -999999999999999. The minimum allowed value in Excel is -9.999999999999999E307. If the cell's value in Excel is less than the minimum value allowed in Spread, the log file issues a message and Spread creates an empty cell.</p>	137
Integer cell format		Excel maintains the precision of float values displayed as integers in the cell, but Spread will save that value as an integer and the decimal portion will be lost. Therefore, be careful when working with values that might lose precision in Spread. The log file issues a message.	121
Integer cell format	TypeIntegerMax TypeIntegerMin	<p>The maximum allowed value for integer cells in Spread is 2147483647. The maximum allowed value in Excel is 9.999999999999999E307. If the cell's value in Excel exceeds the maximum value allowed in Spread, Spread creates the cell as a float cell to accommodate the value.</p> <p>The minimum allowed value in Spread is -2,147,483,648. The minimum allowed value in Excel is -9.999999999999999E307. If the cell's value in Excel is less than the minimum value allowed in Spread, Spread creates the cell as a float cell to accommodate the value.</p>	

**Excel-Formatted File Import (Continued)**

Excel Formatting	Spread Property	Discussion	Log #
Date cell format	TypeDateCentury TypeDateFormat	Spread uses the Excel format if Spread supports it. If Spread does not support the Excel format, the date value is displayed in an edit cell as a string. The log file issues a message.	124
Date cell format	TypeDateSeparator	If the Excel spreadsheet uses a date separator, Spread uses that separator. If Excel does not use a date separator, Spread does not display a date separator.	
Time cell format	TypeTime24Hour TypeTimeSeconds	Spread uses the Excel format if Spread supports it. If Spread does not support the Excel format, the time value is displayed in an edit cell as a string. The log file issues a message.	123
Time cell format	TypeTimeSeparator	If the Excel spreadsheet uses a time separator, Spread uses that separator. If Excel does not use a time separator, Spread does not display a time separator.	
Percentage cell format		Spread converts percentage format to float cells by dividing the cell's value by 100 and displaying it without the percent symbol. The log file issues a message.  Note that Excel rounds decimal values, but Spread truncates them. Also, Excel limits decimal values to 30 decimal places, but Spread limits them to 14 places or less.	129

## Excel-Formatted File Import (Continued)

Excel Formatting	Spread Property	Discussion	Log #																														
Fraction cell format	Spread converts fraction format to float cells and displays the decimal equivalent of the fraction. The log file issues a message.  Note that Excel rounds decimal values, but Spread truncates them. Also, Excel limits decimal values to 30 decimal places, but Spread limits them to 14 places or less.  Spread interprets Excel's fraction formatting as follows:	128																															
	<table border="1"> <thead> <tr> <th>Excel format</th> <th colspan="2">Spread Digits L/R of Decimal</th> </tr> <tr> <td></td> <th>Digits Left</th> <th>Digits Right</th> </tr> </thead> <tbody> <tr> <td>??/??</td> <td>5</td> <td>9</td> </tr> <tr> <td>??/???</td> <td>5</td> <td>9</td> </tr> <tr> <td>?/2</td> <td>13</td> <td>1</td> </tr> <tr> <td>?/4</td> <td>12</td> <td>2</td> </tr> <tr> <td>?/8</td> <td>11</td> <td>3</td> </tr> <tr> <td>?/10</td> <td>12</td> <td>2</td> </tr> <tr> <td>?/100</td> <td>11</td> <td>3</td> </tr> <tr> <td>?/16</td> <td>9</td> <td>5</td> </tr> </tbody> </table>	Excel format	Spread Digits L/R of Decimal			Digits Left	Digits Right	??/??	5	9	??/???	5	9	?/2	13	1	?/4	12	2	?/8	11	3	?/10	12	2	?/100	11	3	?/16	9	5		
Excel format	Spread Digits L/R of Decimal																																
	Digits Left	Digits Right																															
??/??	5	9																															
??/???	5	9																															
?/2	13	1																															
?/4	12	2																															
?/8	11	3																															
?/10	12	2																															
?/100	11	3																															
?/16	9	5																															
Scientific notation cell format	Spread converts scientific notation format to float cells and displays the decimal equivalent of the value. The log file issues a message.  Note that Excel rounds decimal values, but Spread truncates them. Also, Excel limits decimal values to 30 decimal places, but Spread limits them to 14 places or less.	130																															
Custom cell format	Excel supports custom cell formats, including four predefined ones. Spread will convert any custom cell format into an equivalent Spread cell type, if possible. If the cell format cannot be converted, the corresponding Spread cell is created as an empty cell, and the log file issues a message.  Excel provides these predefined custom cell formats: ZIP Code, ZIP Code + 4, phone number, and Social Security Number. Spread converts these cell formats to PIC cells.	123 or 127																															

## Excel-Formatted File Import (Continued)

Excel Formatting	Spread Property	Discussion	Log #
<b>Formulas</b>			
Formulas	Formula	Formulas are imported from Excel, as noted in the following rows. When the import encounters a cell containing a formula or function, the corresponding cell in Spread is set to a float cell, and the log file issues message number 134.  If the import encounters a formula that it does not support, the corresponding cell in Spread is left empty, and the log file issues message number 133.	133 or 134
Reference style for formulas	SetRefStyle method	Sets the <i>RefStyle</i> parameter to either SS_REFSTYLE_A1 or SS_REFSTYLE_R1C1.  If Excel's reference style is R1C1, when Spread imports the spreadsheet, the ColHeaderDisplay property is set to 1 (Display Numbers).	
Whether formula iterations occur	SetIteration method	Sets the <i>Iteration</i> parameter to either True or False	
Maximum number of iterations	SetIteration method	Sets the <i>MaxIterations</i> parameter	
Maximum change per iteration	SetIteration method	Sets the <i>MaxChange</i> parameter	
Automatically calculates formulas when cells change	AutoCalc	Spread uses the Excel setting.	
Formulas referencing other sheets	Not supported	Spread does not import any value for cells containing formulas that reference other sheets. The corresponding cell in Spread is created as an empty cell. The log file issues a message.	136

## Excel-Formatted File Import (Continued)

Excel Formatting	Spread Property	Discussion	Log #
<b>Formula operations</b>			
Operators	Operators for Formula property, such as addition and logical operators	Spread behaves similarly to Excel for all operators except logical operators. Excel returns True or False from Boolean operations; Spread returns 1 (True) or 0 (False).	
Functions	Functions for Formula property, such as absolute value and summation functions	Spread supports many of the functions provided in Excel. A list of the functions Spread supports is in Appendix G, "Formula Operators and Functions." Functions not listed in the appendix are not imported into Spread. When the import encounters a cell containing a non-supported function, the log file issues a message, and no value is imported from Excel.  Functions that are supported in Spread but behave differently in Spread than in Excel are noted in the following rows.	135
Less than	Formula property < function	Excel returns True or False from comparison functions; Spread returns 1 (True) or 0 (False).	
Less than or equal to	Formula property $\leq$ function	Excel returns True or False from comparison functions; Spread returns 1 (True) or 0 (False).	
Equal to	Formula property = function	Excel returns True or False from comparison functions; Spread returns 1 (True) or 0 (False).	
Greater than or equal to	Formula property $\geq$ function	Excel returns True or False from comparison functions; Spread returns 1 (True) or 0 (False).	
Greater than	Formula property > function	Excel returns True or False from comparison functions; Spread returns 1 (True) or 0 (False).	
Not equal to	Formula property $\neq$ function	Excel returns True or False from comparison functions; Spread returns 1 (True) or 0 (False).	

## Excel-Formatted File Import (Continued)

Excel Formatting	Spread Property	Discussion	Log #
Logical IF	Formula property IF function	Excel returns True or False from comparison functions; Spread returns 1 (True) or 0 (False).	
Payment calculation	Formula property PMT function	Spread changes the PMT function's <i>TermsPerYear</i> argument to 12 and uses $-(PMT)$ function to simulate Excel's PMT function (which calculates payments as negative values). The value of the Excel PMT function's <i>Fv</i> and <i>Type</i> arguments are ignored. The log file issues a message.	131
Truncation	Formula property TRUNCATE function	Spread requires 2 arguments for the TRUNCATE function, but Excel requires only 1 argument. If the TRUNCATE function being imported from Excel has only 1 argument, Spread sets the <i>Precision</i> argument to 0 so it works similarly to Excel. The log file issues a message.	142
<b>Other Excel Features</b>			
Hypertext links	URL formula function	Spread converts the hypertext link in Excel to the URL formula function.	
Graphs/charts	Not supported	Graphs or charts in Excel files are not imported into Spread.	
Rotated text	Not supported	Spread imports text and displays it in the usual orientation.	
Indented text	Not supported	Spread imports text and displays it at the usual alignment (not indented).	
Text shrunken to fit cell	Not supported	Spread imports text and displays it at size specified by the <i>FontSize</i> or <i>Font</i> property.	
Merged cells		Spread places the value of the upper-left cell in Excel's merged cell block into the corresponding cell in the spreadsheet. All other cells in Spread remain empty.	
Password-protected files		Spread cannot open password-protected files. The log file issues a message.	144

## Excel-Formatted File Export

When Spread exports an Excel-formatted file, it translates its property settings and some function call settings to Excel-supported settings, as listed in the following sections.

---

**Note** If your spreadsheet contains numeric data and you are planning to export the spreadsheet to Excel, for best results place any numeric data in integer or float cells in Spread.

---

### Property Settings and Export

When Spread exports an Excel-formatted file, it translates its property settings to Excel-supported settings, as listed in the following table.

#### Excel-Formatted File Export—Property Settings

Spread Property	Excel Setting	Discussion	Log #
ActiveCol	Specifies active cell (with ActiveRow property).		
ActiveRow	Specifies active cell (with ActiveCol property).		
AllowCellOverflow	The setting from Spread is ignored.	Excel always allows cell overflow.	
AllowDragDrop	The setting from Spread is ignored.	Excel reads this setting from the system's registry. The log file issues a message.	102
AllowMultiBlocks	The setting from Spread is ignored.	Excel always allows multiple blocks to be selected.	
AllowUserFormulas	The setting from Spread is ignored.	Excel always allows users to enter formulas.	
ArrowsExitEditMode	The setting from Spread is ignored.	Excel always uses arrows to exit edit mode in cells.	
AutoCalc	Excel uses the Spread setting to determine whether to calculate formulas when cells change.		
AutoClipboard	The setting from Spread is ignored.	Excel reads value from the system's registry. The log file issues a message.	102

## Excel-Formatted File Export—Property Settings (Continued)

Spread Property	Excel Setting	Discussion	Log #
BackColor	Excel uses the Spread setting.	Excel sets the background color using the same precedence that Spread uses (cell, row, column, spreadsheet). The color from Spread is translated to be one of 40 Excel colors.	
CellBorderStyle	Excel uses the Spread setting as follows: <b>Spread setting:</b> 0 - Default 1 - Solid 2 - Dash 3 - Dot 4 - Dash Dot 5 - Dash Dot Dot 6 - Blank 11 - Fine Solid 12 - Fine Dash 13 - Fine Dot 14 - Fine Dash Dot 15 - Fine Dash Dot Dot	<b>Excel setting:</b> None Medium Medium dashed Dotted Medium dash-dot Medium dash-dot-dot None Thin Dashed Dotted Dash-dot Dash dot dot	
CellBorderType	Excel uses the Spread setting.	Excel supports all the Spread border types.	
CellType	Excel sets the cell format using the Spread settings when possible, as described in the following rows.		
Button	Excel uses the value of the control.	Creates general cell in Excel that contains the value from the button cell. If the button cell is a one-state button, the value is 0. If the button cell is a two-state button, the value is either 0 or 1. The log file issues a message.	171
Check box	Excel uses the value of the control.	Creates general cell in Excel that contains the value from the check box cell. If the check box cell is a two-state check box, the value is 0 or 1. If the check box cell is a three-state check box, the value is either 0, 1, or 2. The log file issues a message.	170

## Excel-Formatted File Export—Property Settings (Continued)

Spread Property	Excel Setting	Discussion	Log #								
Combo box	Excel uses the current selection in the control.	Creates general cell in Excel that contains the current selection in the combo box cell. The log file issues a message.	172								
Date	Excel creates cell with date format as described in the following rows.										
TypeDateCentury	Excel uses the Spread setting.	Determines whether the century digits are displayed in the year value.									
TypeDateFormat	Excel uses the Spread setting.	<p>Sets date format, interpreting the date format as follows:</p> <table border="1"> <thead> <tr> <th>Spread</th> <th>Excel</th> </tr> </thead> <tbody> <tr> <td>DS_DDMMYY</td> <td>“dd/mm/yy”</td> </tr> <tr> <td>DS_MMDDYY</td> <td>“mm/dd/yy”</td> </tr> <tr> <td>DS_YYMMDD</td> <td>“yy/mm/dd”</td> </tr> </tbody> </table>	Spread	Excel	DS_DDMMYY	“dd/mm/yy”	DS_MMDDYY	“mm/dd/yy”	DS_YYMMDD	“yy/mm/dd”	
Spread	Excel										
DS_DDMMYY	“dd/mm/yy”										
DS_MMDDYY	“mm/dd/yy”										
DS_YYMMDD	“yy/mm/dd”										
TypeDateMax	Excel ignores the Spread setting.	Excel does not let you change the maximum allowed date. Excel and Spread limit dates to dates before December 31, 9999.									
TypeDateMin	Excel ignores the Spread setting.	<p>Excel does not let you change the minimum allowed date. Excel limits dates to dates after January 1, 1990 (or January 1, 1904).</p> <p>Spread’s minimum allowed date is January 1, 1 A.D. If the date value in Spread is earlier than January 1, 1990, Spread writes the value to the log file and the cell in Excel displays the date value in the following format: day:dd month:mm year:yyyy</p>	161								
TypeDateSeparator	Excel uses the Spread setting.										

**Excel-Formatted File Export—Property Settings (Continued)**

Spread Property	Excel Setting	Discussion	Log #
Edit	Excel creates cell with general format as described in the following rows.	Excel creates cell with general format as described in the following rows.	
TypeEditCharCase	Excel ignores the Spread setting.	Excel does not force characters to a certain case.	
TypeEditCharSet	Excel ignores the Spread setting.	Excel does not allow you to limit the character set for edit cells.	
TypeEditMultiLine	Excel uses the Spread setting.	Excel uses the Spread setting.	
TypeEditPassword	Excel ignores the Spread setting.	Excel does not provide password field appearance.	
Float	Excel creates cell with number format as described in the following rows.	Excel creates cell with number format as described in the following rows.	
		Excel rounds float values, but Spread truncates them. Therefore, be careful when working with values that might have been truncated in Spread.	
		Excel maintains the precision of float values regardless of the type of cell that displays them, but Spread does not. Therefore, be careful when working with values that might have lost precision in Spread.	
TypeFloatCurrencyChar	Excel uses the Spread setting.	Excel uses the setting of the FloatDefCurrencyChar property if you have not defined a currency character using the TypeFloatCurrencyChar property.	
TypeFloatDecimalChar	Excel ignores the Spread setting.	Excel uses the system setting.	
TypeFloatDecimalPlaces	Excel uses the Spread setting.	Spread does not maintain precision for float values. Be aware that if the float value came from a source with greater precision, that original value is not saved.	
TypeFloatMax	Excel ignores the Spread setting.	Excel's maximum allowed value is 9.999999999999999E307. It does not let you change this value.	



## Excel-Formatted File Export—Property Settings (Continued)

Spread Property	Excel Setting	Discussion	Log #
Static text	Not supported	Excel creates a cell containing the value from the static text cell with general format as described in the following rows. The log file issues a message.	175
TypeTextPrefix	Excel ignores the Spread setting.	If the value from the Spread cell contains an ampersand, the ampersand is displayed in the Excel cell.	
TypeTextShadow	Excel ignores the Spread setting.	Excel does not support three-dimensional cell effects.	
TypeTextShadowIn	Excel ignores the Spread setting.	Excel does not support three-dimensional cell effects.	
TypeTextWordWrap	Excel uses the Spread setting.		
Time	Excel creates cell with time format as described in the following rows.		
TypeTime24Hour	Excel uses the Spread setting.		
TypeTimeSeconds	Excel uses the Spread setting.		
TypeTimeSeparator	Excel ignores the Spread setting.	Excel uses the system setting.	
ClipboardOptions	Excel ignores the Spread setting.	Excel always excludes headings from Clipboard operations.	
ColHeaderDisplay	Excel ignores the Spread setting.	Excel by default displays letters, unless the reference style is set to R1C1, in which case the column headers display numbers.  <b>Note</b> If you have provided custom text for your column headers in Spread, it is not exported to Excel. Excel does not support custom column headers. The log file issues message 173.	
ColHidden	Excel uses the Spread setting.		

## Excel-Formatted File Export—Property Settings (Continued)

Spread Property	Excel Setting	Discussion	Log #
ColsFrozen	Excel ignores the Spread setting.		
ColWidth	Excel uses the Spread setting.	Excel uses the Spread settings to set the column width for each individual column.	
DisplayColHeaders	Excel uses the Spread setting.	Excel has two choices: show column and row headers or do not show any headers. If DisplayColHeaders is set to True, Excel shows both column and row headers.	
DisplayRowHeaders	Show headers	Excel has two choices: show column and row headers or do not show any headers. If DisplayRowHeaders is set to True, Excel shows both column and row headers.	
EditEnterAction	Excel ignores the Spread setting.	Excel reads this setting from the system's registry. The log file issues a message.	102
EditModeReplace	Excel ignores the Spread setting.	Excel always places the cursor at the beginning of the cell's data.	
FloatDefCurrencyChar	Excel uses Spread setting for currency symbol.	Excel uses the setting of the FloatDefCurrencyChar property if you have not defined a currency character using the TypeFloatCurrencyChar property.	
FloatDefDecimalChar	Excel ignores the Spread setting.	Excel uses the system settings.	
FloatDefSepChar	Excel ignores the Spread setting.	Excel uses the system settings.	
Font	Excel uses the Spread settings.	Excel applies font settings to the cell, the row, the column, and the spreadsheet, using the same precedence Spread uses.	
FontBold	Excel uses the Spread settings.	Excel applies font settings to the cell, the row, the column, and the spreadsheet, using the same precedence Spread uses.	

**Excel-Formatted File Export—Property Settings (Continued)**

<b>Spread Property</b>	<b>Excel Setting</b>	<b>Discussion</b>	<b>Log #</b>
FontItalic	Excel uses the Spread settings.	Excel applies font settings to the cell, the row, the column, and the spreadsheet, using the same precedence Spread uses.	
FontName	Excel uses the Spread settings.	Excel applies font settings to the cell, the row, the column, and the spreadsheet, using the same precedence Spread uses.	
FontSize	Excel uses the Spread settings.	Excel applies font settings to the cell, the row, the column, and the spreadsheet, using the same precedence Spread uses.	
FontStrikethru	Excel uses the Spread settings.	Excel applies font settings to the cell, the row, the column, and the spreadsheet, using the same precedence Spread uses.	
FontUnderline	Excel uses the Spread settings.	Excel applies font settings to the cell, the row, the column, and the spreadsheet, using the same precedence Spread uses.	
ForeColor	Excel uses the Spread settings.	Excel uses this setting to set the font color. The color from Spread is translated to be one of 40 Excel colors.	

## Excel-Formatted File Export—Property Settings (Continued)

Spread Property	Excel Setting	Discussion	Log #						
Formula	Excel uses the formulas from Spread, except as noted.	<p>The following functions are not exported to Excel:</p> <p>PMT (log message 163)            Custom functions (log message 167)            URL (log message 174)</p> <p>When a cell containing a function that cannot be exported is encountered, the log file issues a message.</p> <p>Spread exports the INVERSE function to Excel's MINVERSE function (the matrix inverse).</p> <p>Spread supports the following functions, but Excel does not provide corresponding functions. Therefore, the functions are exported as mathematical expressions. If you subsequently import the file from Excel, Spread imports the mathematical expression, and does not resubstitute the function for the mathematical expression.</p> <table border="1"> <thead> <tr> <th>Function in Spread</th> <th>Math Expression in Excel</th> </tr> </thead> <tbody> <tr> <td>SQUARE</td> <td><math>x * x</math></td> </tr> <tr> <td>XROOT</td> <td><math>x^{1/y}</math></td> </tr> </tbody> </table>	Function in Spread	Math Expression in Excel	SQUARE	$x * x$	XROOT	$x^{1/y}$	163, 167, or 174
Function in Spread	Math Expression in Excel								
SQUARE	$x * x$								
XROOT	$x^{1/y}$								
FormulaSync	Excel ignores the Spread setting.	Excel always tries to update cell references when cells are copied. It does not updated them when cells are moved. Whether cell references are updated when cells are copied depend on the cell reference style you use.							
GridColor	Excel uses the Spread setting.	The color from Spread is translated to be one of 40 Excel colors.							
GridShowHoriz	Excel uses the Spread setting.	Excel has two choices: show horizontal and vertical grid lines or do not show any grid lines. If GridShowHoriz is set to True, Excel shows both horizontal and vertical grid lines.							

**Excel-Formatted File Export—Property Settings (Continued)**

<b>Spread Property</b>	<b>Excel Setting</b>	<b>Discussion</b>	<b>Log #</b>
GridShowVert	Excel uses the Spread setting.	Excel has two choices: show horizontal and vertical grid lines or do not show any grid lines. If GridShowVert is set to True, Excel shows both horizontal and vertical grid lines.	
Lock	Excel uses the Spread setting.	Spreadsheets exported from Spread export the Lock property setting for each cell and export the Protect property setting for the spreadsheet. Note that the default value for the Protect property in Spread is True—the opposite of the Excel Protect default setting—and the default value of the Lock property is False for all cells in Spread.	
MaxCols	Excel ignores the Spread setting.	You cannot change the maximum number of columns in Excel. The maximum number of columns Excel allows is 256. The maximum number of columns in Spread is two billion. If Spread contains more than 256 columns, the extra columns are truncated. The log file issues a message.	165
MaxRows	Excel ignores the Spread setting.	You cannot change the maximum number of rows in Excel. The maximum number of rows Excel allows is 65,536. The maximum number of rows in Spread is two billion. If Spread contains more than 65,536 rows, the extra rows are truncated. The log file issues a message.	166
PrintColHeaders	Excel uses the Spread setting.	Excel has two choices: print column and row headers or do not print any headers. If PrintColHeaders is set to True, Excel prints both column and row headers.	

## Excel-Formatted File Export—Property Settings (Continued)

Spread Property	Excel Setting	Discussion	Log #
PrintGrid	Excel uses the Spread setting.		
PrintRowHeaders	Excel uses the Spread setting.	Excel has two choices: print column and row headers or do not print any headers. If PrintRowHeaders is set to True, Excel prints both column and row headers.	
Protect	Protects cells	Spreadsheets exported from Spread export the Lock property setting for each cell and export the Protect property setting for the spreadsheet. Note that the default value for the Protect property in Spread is True—the opposite of the Excel Protect default setting—and the default value of the Lock property is False for all cells in Spread.	
RowHeaderDisplay	Excel ignores the Spread setting.	Excel's default setting is numbers, which cannot be changed.  <b>Note</b> If you have provided custom text for your row headers in Spread, it is not exported to Excel. Excel does not support custom row headers. The log file issues message 173.	
RowHeight	Excel uses the Spread setting.	Excel uses the settings to set the height for each individual row.	
RowHidden	Excel uses the Spread setting.		
RowsFrozen	Excel ignores the Spread setting.		

**Excel-Formatted File Export—Property Settings (Continued)**

<b>Spread Property</b>	<b>Excel Setting</b>	<b>Discussion</b>	<b>Log #</b>
ScrollBars	Excel uses the Spread setting.	Excel displays either no scroll bars or both horizontal and vertical scroll bars. If the ScrollBars property is set to 1 (Horizontal) or 2 (Vertical), Excel displays both horizontal and vertical scroll bars.	
SelBlockCol	Excel uses the Spread setting.	Excel spreadsheet uses SelBlockCol, SelBlockCol2, SelBlockRow, and SelBlockRow2 settings to select block that was selected in Spread	
SelBlockCol2	Excel uses the Spread setting.	Excel spreadsheet uses SelBlockCol, SelBlockCol2, SelBlockRow, and SelBlockRow2 settings to select block that was selected in Spread	
SelBlockRow	Excel uses the Spread setting.	Excel spreadsheet uses SelBlockCol, SelBlockCol2, SelBlockRow, and SelBlockRow2 settings to select block that was selected in Spread	
SelBlockRow2	Excel uses the Spread setting.	Excel spreadsheet uses SelBlockCol, SelBlockCol2, SelBlockRow, and SelBlockRow2 settings to select block that was selected in Spread	
SelectBlockOptions	Excel ignores the Spread setting.	Excel always behaves as if the SelectBlockOptions property is set to 15 (all values).	
ShadowColor	Excel ignores the Spread setting.	Excel does not support three-dimensional shading and effects. If the cell encountered is a static text cell, the log file issues message 175.	175
ShadowDark	Excel ignores the Spread setting.	Excel does not support three-dimensional shading and effects. If the cell encountered is a static text cell, the log file issues message 175.	175

## Excel-Formatted File Export—Property Settings (Continued)

Spread Property	Excel Setting	Discussion	Log #
ShadowText	Excel ignores the Spread setting.	Excel does not support three-dimensional shading and effects. If the cell encountered is a static text cell, the log file issues message 175.	175
Text		If you have provided custom text for your column or row headers in Spread, it is not exported to Excel. Excel does not support custom column or row headers. The log file issues a message.	173
TwoDigitYearMax	Excel ignores the Spread setting.	Year values are always exported as four-digit numbers from Spread. Therefore, the method specified by the TwoDigitYearMax property in Spread is used to determine the century value in Spread, and that calculated value is exported.	
TypeHAlign	Excel uses the Spread setting.		
TypeMaxEditLen	Excel ignores the Spread setting.	Excel does not let you set a maximum length for edit cells.	
TypeVAlign	Excel uses the Spread setting.		
Value		If you have provided custom text for your column or row headers in Spread, it is not exported to Excel. Excel does not support custom column or row headers. The log file issues a message.	173
VirtualMode		The export ignores the setting of the VirtualMode property and exports the entire spreadsheet.	

## Method Calls and Export

Settings exported to an Excel-formatted file will include the following settings, if they have been applied to the spreadsheet before the export.

### Excel-Formatted File Export—Methods

Spread Method	Excel Setting	Discussion
GetIteration, SetIteration	Excel uses Spread settings, if they have been applied to the spreadsheet.	Sets for the spreadsheet whether iterations occur ( <i>Iterations</i> parameter), the maximum number of iterations ( <i>MaxIterations</i> parameter), and the maximum change between iterations ( <i>MaxChange</i> parameter).
GetRefStyle, SetRefStyle	Excel uses Spread settings, if they have been applied to the spreadsheet.	Excel uses the value of the <i>RefStyle</i> parameter if value is set to <code>SS_REFSTYLE_A1</code> or <code>SS_REFSTYLE_R1C1</code> . If the parameter is set to <code>SS_REFSTYLE_DEFAULT</code> , Excel uses the R1C1 reference style, and you must update your formulas accordingly.  If Excel uses the R1C1 format, the column headers use numbers instead of letters.

## Log File Messages

When you import an Excel-formatted file or export a Spread file to Excel format, Spread can generate a log file. The log file provides an number and a short message explaining the export or import action, and the column and row number of the cell for which the action occurred, if applicable.

The following table lists the possible messages used in the log files.

### Log File Messages

Number	Log Message	Explanation
102	Excel reads the AllowDragDrop, AutoClipboard, and EditEnterAction properties from the registry. If you modify these properties in Spread, the changes are not saved at export.	Excel reads values stored in the registry for setting the listed properties. At import, Spread reads those files in turn from Excel. However, at export, the settings in Spread are ignored, and Excel reads those values from the registry.
120	Import Excel File: <i>path\filename</i>	Spread is importing the specified Excel file.
121	Rounding/Truncating difference: If a float value is placed in an integer cell, Spread truncates the value. Excel rounds float values in integer cells. Value from Excel = <i>value</i>	In Excel, an integer cell contains a float value and the value is rounded for display in the integer cell. In Spread, the value will be truncated for display. For example, if the float value is 27.9, in Excel the value is displayed as 28, and in Spread the value is displayed as 27. The float value from Excel is provided at the end of the message.
122	Spread does not support the format token. Excel token = <i>token</i>	Tokens in Excel are the pieces that define formats, such as “dd” that is part of the date format “dd-mm-yy”. Spread does not support some Excel tokens. Because Spread does not support the format token, no value is placed in the corresponding Spread cell. The format token from Excel is provided at the end of the message.
123	Spread does not support this format. Excel format = <i>format</i>	Excel supports formats that Spread does not, such as a date/time combination format or the day of the week format. Because Spread does not support the format, no value is placed in the corresponding Spread cell. The format from Excel is provided at the end of the message.
124	Spread does not support this date format. Excel format = <i>format</i>	Excel supports date formats that Spread does not, such as “mmm/dd”. Because Spread does not support the format, no value is placed in the corresponding Spread cell. The date format from Excel is provided at the end of the message.

## Log File Messages (Continued)

Number	Log Message	Explanation
125	Spread ignored zeros in format w/ leading zeros. Excel format = <i>format</i>	Excel lets you format values with leading zeros, such as the value 0123. Spread does not allow you to do this; therefore, Spread ignores the leading zeros. Spread would read in the value 0123 as 123. The format from Excel is provided at the end of the message.
126	Spread substituted zeros in format w/ leading or trailing non-zeros. Excel format = <i>format</i>	Excel lets you specify a format that acts as a template for numbers users enter. For example, such a template might be “00.###” so that when users type 12.1, they get the value 12.1. Spread does not support this capability (in Spread, the number will appear as 12.100). The format from Excel is provided at the end of the message.
127	Spread recognizes this format token, but does not support it. Excel token = <i>token</i>	Tokens in Excel are the pieces that define formats, such as “dd” that is part of the date format “dd-mm-yy”. Spread does not support some Excel tokens. Because Spread does not support the format token, no value is placed in the corresponding Spread cell. The format token from Excel is provided at the end of the message.
128	Spread does not support fraction formats; the value has been changed to a decimal representation. Excel format = <i>format</i>	Excel can represent values as fractions, but Spread cannot. The value in Excel is converted to a decimal representation of the fraction, where the number of digits in the representation is determined by Spread’s limitations (no more than 14 total digits in float values). The Excel fraction format is provided at the end of the message.
129	Spread does not support percent formats; the value has been changed to a decimal representation. Excel format = <i>format</i>	Excel can represent values as percentage values, but Spread cannot. The value in Excel is converted to a decimal representation of the percentage. The Excel percent format is provided at the end of the message.

## Log File Messages (Continued)

Number	Log Message	Explanation
130	Spread does not support scientific notation; the value has been changed to a decimal representation. Excel format = <i>format</i>	Excel can represent values using scientific notation, but Spread cannot. The value in Excel is converted to a decimal representation of the number, where the number of digits in the representation is determined by Spread's limitations (no more than 14 total digits in float values). The Excel scientific notation format is provided at the end of the message.
131	PMT() Function—TermsPerYear argument defaults to 12 when Spread imports the Excel file.	Spread changes its PMT function's <i>TermsPerYear</i> argument to 12 and uses <i>-(PMT)</i> function to simulate Excel's PMT function (which calculates payments as negative values). The value of the Excel PMT function's <i>Fv</i> and <i>Type</i> arguments are ignored.
132	Spread does not support graphic objects. The graphic object will not be represented in Spread.	Excel allows you to add graphic objects on top of the spreadsheet; however, Spread does not support graphic objects. The graphic object in the Excel spreadsheet does not import into Spread.
133	Import encountered unrecognized Excel formula format tag. An empty cell has been created in Spread. Excel format tag = <i>tag</i>	Spread does not recognize a format tag used to create the formula used in Excel. Therefore, the value from Excel is not imported into Spread. The formula format tag used in Excel is provided at the end of the message.
134	Import encountered a function or formula in the Excel file; a float cell has been created in Spread.	Whenever Spread encounters a function or formula in an Excel cell, the value from the cell is imported into a float cell in Spread.
135	Spread does not support the specified function. An empty cell has been created in the Spread file. Excel function = <i>function</i>	Excel provides functions that Spread does not support. Therefore, Spread creates an empty cell when it encounters an Excel cell that contains functions it does not support.
136	Import encountered a formula error reported from Excel. An empty cell has been created in the Spread file.	Excel allows you to create formulas that return error values, but Spread does not. An example of such a formula is one that uses an invalid value. Instead of importing the erroneous formula, Spread creates an empty cell. Spread also issues this message when it tries to import a cell that contains a formula that references other sheets.

## Log File Messages (Continued)

Number	Log Message	Explanation
137	Float value in Excel is beyond the range of Spread's float cell type. An empty cell has been created in Spread. Excel float value = <i>value</i>	<p>The maximum allowed value in Spread is 9999999999999999. The maximum allowed value in Excel is 9.999999999999999E307. If the cell's value in Excel exceeds the maximum value allowed in Spread, Spread creates an empty cell.</p> <p>The minimum allowed value in Spread is -9999999999999999. The minimum allowed value in Excel is -9.999999999999999E307. If the cell's value in Excel is less than the minimum value allowed in Spread, Spread creates an empty cell.</p>
138	Excel's currency symbol uses multiple characters. Spread does not support multiple characters; therefore, Spread uses the first character of the Excel symbol. Excel symbol = <i>symbol</i>	Excel lets you display multiple characters as the currency symbol, such as "DM". Spread does not, so it just displays the first character of the Excel symbol. The currency symbol used in Excel is provided at the end of the message.
139	Excel displayed the currency symbol somewhere other than to the left of the value. Spread does not support this; therefore, Spread placed the symbol to the left of the value. Excel format = <i>format</i>	Excel lets you display the currency symbol after the value, as is the practice in some countries. Spread does not support this, so the symbol is displayed to the left of the value. The currency format used in Excel is provided at the end of the message.
140	Spread does not support Rich Text Formats. The text is displayed in the first format encountered. String = <i>text</i>	In Excel, you can format text, such as providing boldface and normal text in the same cell. Spread does not support that capability. Therefore, when Rich Text Formats are encountered on import, Spread imports the text, but displays it with the first formatting used in the corresponding Excel cell.
142	TRUNCATE() Function—Precision argument defaults to 0 when Spread imports the Excel file.	If the Excel TRUNC function does not have a second argument, Excel uses the default value of 0. Therefore, when Spread imports the TRUNC function, if the Excel function does not have a second argument, Spread uses the value 0 for the <i>Precision</i> argument.

## Log File Messages (Continued)

Number	Log Message	Explanation
143	February 29, 1900, is not a valid date, although Excel accepts it as a valid date. An empty cell has been created in Spread.	Excel incorrectly accepts the date February 29, 1900, as a valid date; the year 1900 was not a leap year. Spread recognizes that this is not a valid date, and therefore creates the corresponding cell in the imported spreadsheet as an empty date cell.
144	This Excel file is password protected. Spread cannot open password protected files.	Spread cannot open password-protected Excel files.
160	Export Excel File: <i>path\filename</i>	Spread is exporting the specified file to an Excel-formatted file.
161	Excel dates must fall between 1/1/1900 and 12/31/9999. The date in Spread is not in that range, and has not been saved at export. Date in Spread = <i>date</i>	Excel limits dates to dates after January 1, 1990 (or January 1, 1904). Spread's minimum allowed date is January 1, 1 A.D. If the date value in Spread is earlier than January 1, 1990, the log file provides the value in Spread at the end of the message, and the cell in Excel displays the date value in the following format: day:dd month:mm year:yyyy
162	Excel does not support the specified Spread cell type; an empty cell has been created in the Excel file. Spread cell type = <i>type</i>	Excel does not support the following Spread cell types: Owner-drawn Picture  The Spread cell type is provided at the end of the message.
163	The PMT() function is not exported to Excel. An empty cell has been created in the Excel file.	Because Spread and Excel handle the PMT function differently, the Spread PMT function is not exported to Excel. The corresponding cell in the exported file is created as an empty cell.
164	Excel does not support the specified mask definition character. This mask character has not been exported to Excel. Spread mask character = <i>character</i>	Excel does not support nonnumeric mask characters. Therefore, any nonnumeric mask definition characters are not included in the mask created in the exported file. The non-supported mask character is provided at the end of the message.

## Log File Messages (Continued)

Number	Log Message	Explanation
165	The Spread control contains more columns than Excel supports. Excel supports only 256 columns; columns beyond 256 have been truncated. Number of columns in Spread = <i>columns</i>	The maximum number of columns Excel allows is 256. The maximum number of columns in Spread is two billion. If Spread contains more than 256 columns, the extra columns are truncated. The number of columns in Spread is provided at the end of the message.
166	The Spread control contains more rows than Excel supports. Excel supports only 65536 rows; rows beyond 65536 have been truncated. Number of rows in Spread = <i>rows</i>	The maximum number of rows Excel allows is 65,536. The maximum number of rows in Spread is two billion. If Spread contains more than 65,536 rows, the extra rows are truncated. The number of rows in Spread is provided at the end of the message.
167	Custom functions are not exported to Excel.	Custom functions created in Spread are not exported to Excel. The corresponding cell in Excel is created as an empty cell.
170	Export encountered a check box cell. The value of the check box cell has been placed in the corresponding Excel cell.	Excel does not support check box cells. Instead, the export creates a general cell in Excel that contains the value from the check box cell. If the check box cell is a two-state check box, the value is 0 or 1. If the check box cell is a three-state check box, the value is either 0, 1, or 2.
171	Export encountered a button cell. The value of the button cell has been placed in the corresponding Excel cell.	Excel does not support button cells. Instead, the export creates a general cell in Excel that contains the value from the button cell. If the button cell is a one-state button, the value is 0. If the button cell is a two-state button, the value is either 0 or 1.
172	Export encountered a combo box cell. The current selection in the combo box cell has been placed in the corresponding Excel cell.	Excel does not support combo box cells. Instead, the export creates a general cell in Excel that contains the current selection from the combo box cell. The log file issues a message.
173	Custom headers are not exported to Excel.	If you have provided custom text for your column or row headers in Spread, it is not exported to Excel. Excel does not support custom column or row headers.

## Log File Messages (Continued)

Number	Log Message	Explanation
174	The URL() function is not exported to Excel. The text displayed in the Spread cell has been placed in the corresponding Excel cell. The hyperlink is: <i>hyperlink_text</i>	The hypertext links created in Spread using the URL function are not exported to Excel. However, Excel does support hypertext links, which you can add to your Excel file.
175	Excel does not support the Spread static text cell type. The value of the Spread cell has been placed in the corresponding Excel cell, but the cell formatting has been discarded.	Excel does not support the Spread static text cell type. Instead, Excel creates a general cell that contains the value of the Spread static text cell. The Excel general cell will not reflect the shadow color and other three-dimensional settings for the Spread static text cell. It will use other Spread cell settings, such as word wrap and text alignment.
190	Spread 3.0 does not support multiple format sections. Spread will represent the value using the default Spread format. Format in Excel = <i>format</i>	Excel lets you define multiple formats for representing values in cells, but Spread does not. Therefore, Spread displays the value from Spread in the default Spread format. The format used in Excel is provided at the end of the message.

# Appendix J HTML File Export

This appendix provides a summary of the manner in which Spread deals with exporting spreadsheets to HTML. The first section describes how the settings in Spread are translated into an HTML table. The second section lists the messages that are provided in the log file, should you choose to create one. The log file records the manner in which Spread handled the export of features, including cell types and other settings.

---

**Note** Do not export a Spread file that uses virtual mode. Before exporting the spreadsheet, turn virtual mode off. If you do not turn virtual mode off, the HTML table will contain only the data that was in the virtual buffer at the time of the export.

---

For more information about HTML export and for instructions for exporting, see “Exporting a Spreadsheet to an HTML File” on page 119.

## File Export Information

When Spread exports an HTML file, it translates its property settings to an HTML table, as listed in the following table.

Spread Property	HTML Tag/Attribute	Discussion	Log #
AllowCellOverflow	TD and TH tags, COLSPAN and NOWRAP attributes	If the AllowCellOverflow property is set to True, cells in the spreadsheet that overflowed into adjacent empty cells become cells that span columns in the HTML table (COLSPAN attribute) and the NOWRAP attribute is set.  If the AllowCellOverflow property is set to False, cells containing long data allow data to wrap in the cells.	

## Appendix J

Spread Property	HTML Tag/Attribute	Discussion	Log #
BackColor	TH and TD tags, BGCOLOR attribute	Displays the background color assigned to the individual cell, the row, the column, or the spreadsheet, in that order. The colors in the fpSpread control are translated to the closest color of the 216 defined web-compliant colors.	
CellType		HTML tables do not provide cells that can act similarly to controls, as Spread does. However, for many cell types, the data and formatting in Spread are exported to the HTML table, as explained in the following rows.	
Button		Contents of button cells are not exported. The log file issues a message.	222
Check box		If the check box cell is a two-state check box, the value 0 or 1 is exported to the HTML table. If the check box cell is a three-state check box, the value 0, 1, or 2 is exported to the HTML table. The log file issues a message.	218
Combo box		The current value in the edit field of the combo box is exported to the cell in the HTML table. The log file issues a message.	219
Date		Date cells are exported in the format in which they are displayed, as explained in the following rows.	

Spread Property	HTML Tag/Attribute	Discussion	Log #
TypeDateCentury		<p>If the TypeDateCentury property is set to True, the displayed date displays the century digits, which are exported to the HTML table.</p> <p><b>Note</b> To avoid misunderstanding and possible miscalculations, always display century values in Spread so that they are exported to the HTML table.</p>	
TypeDateFormat		The setting of the TypeDateFormat property determines the date format in Spread, which is the format exported to the HTML table.	
TypeDateSeparator		The setting of the TypeDateSeparator property determines the date separator in Spread, which is the separator exported to the HTML table.	
Edit		Edit cells are exported in the format in which they are displayed, as explained in the following rows.	
TypeEditCharCase		The setting of the TypeEditCharCase property determines the text case in Spread, which is the text case exported to the HTML table.	
TypeEditPassword		If the TypeEditPassword property is set to True, the asterisks used to mask the data in the fpSpread control are exported to the HTML table.	
Float		Float cells are exported in the format in which they are displayed, as explained in the following rows.	

## Appendix J

Spread Property	HTML Tag/Attribute	Discussion	Log #
TypeFloatCurrencyChar		The setting of the TypeFloatCurrencyChar property determines the currency character in Spread, which is the character exported to the HTML table.	
TypeFloatDecimalChar		The setting of the TypeFloatDecimalChar property determines the decimal character in Spread, which is the character exported to the HTML table.	
TypeFloatDecimalPlaces		The setting of the TypeFloatDecimalPlaces property determines the number of decimal digits displayed in Spread, which is the number of decimal digits exported to the HTML table.	
TypeFloatMoney		The setting of the TypeFloatMoney property determines whether the cell displays a currency character in Spread; if the cell displays the character, it is exported to the HTML table.	
TypeFloatSeparator		The setting of the TypeFloatSeparator property determines whether the cell displays a separator character in Spread; if the cell displays the character, it is exported to the HTML table.	
TypeFloatSepChar		The setting of the TypeFloatSepChar property determines the separator character in Spread, which is the character exported to the HTML table.	
Integer		Integer cells are exported in the format in which they are displayed.	

Spread Property	HTML Tag/Attribute	Discussion	Log #
Owner-Drawn		Contents of owner-drawn cells are not exported. The log file issues a message.	220
PIC		The data in PIC cells is exported, as explained in the following rows.	
TypePicDefaultText		Default text is not exported unless the user has typed some data in the cell, in which case the PIC cell displays the data and the default text, and it is all exported.	
TypePicMask		The setting of the TypePicMask property determines which data users can enter in the cell. The data provided by the users is exported, as are the literal characters defined in the PIC cell.	
Picture		The icon or bitmap in the picture cell is converted to a JPEG file. If for some reason the bitmap or icon cannot be converted to a JPEG, the log file issues a message.	221, 222, 223, 224, 229
TypePictCenter	TD and TH tags, VALIGN attribute	If the TypePictCenter property is set to True, the VALIGN attribute is set to CENTER.	
TypePictMaintainScale		The picture in the cell is converted to a JPEG that is exactly the same in dimensions and appearance as the picture as it is displayed in the Spread control.	
TypePictPicture		The file specified by the TypePictPicture property is converted to a JPEG, named SPRnnnnn.JPG (where nnnnn is a random number), and placed in the same directory as the HTML file. The log file issues a message.	221, 222

## Appendix J

Spread Property	HTML Tag/Attribute	Discussion	Log #
TypePictStretch		The picture in the cell is converted to a JPEG that is exactly the same in dimensions and appearance as the picture as it is displayed in the Spread control.	
Static text		Static text cells are exported in the format in which they are displayed.	
Time		Time cells are exported in the format in which they are displayed, as explained in the following rows.	
TypeTime24Hour		The setting of the TypeTime24Hour property determines whether the cell displays time in 12- or 24-hour format in Spread; whichever format the cell displays is the format exported to the HTML table.	
TypeTimeSeconds		The setting of the TypeTimeSeconds property determines whether the cell displays the seconds value in Spread; if the cell displays seconds, the seconds value is exported to the HTML table.	
TypeTimeSeparator		The setting of the TypeTimeSeparator property determines which time separator character the cell displays in Spread; whichever separator the cell displays is exported to the HTML table.	
ColHeaderDisplay	TH tag	Sets text in column headers in table heading.	
ColHidden		Spread does not export hidden columns to the HTML file.	

Spread Property	HTML Tag/Attribute	Discussion	Log #
ColWidth	TD and TH tags, WIDTH attribute	The widths of the exported columns are set (in pixels) to create columns that are the same widths as the Spread columns.	
DisplayColHeaders	TH tag	Displays column headers if set to True; does not display them if set to False.	
DisplayRowHeaders	TH tag	Displays row headers if set to True; does not display them if set to False.	
FloatDefCurrencyChar		The setting of the FloatDefCurrencyChar property determines the default currency character in Spread, which is the character exported to the HTML table.	
FloatDefDecimalChar		The setting of the FloatDefDecimalChar property determines the default decimal character in Spread, which is the character exported to the HTML table.	
FloatDefSepChar		The setting of the FloatDefSepChar property determines the default separator character in Spread, which is the character exported to the HTML table.	
Font	See separate font setting properties		
FontBold	B tag	Sets bold face text using bold tag.	
FontItalic	I tag	Sets italicized text using italic tag.	
FontName	FONT FACE attribute	Sets font displayed using FONT FACE attribute.	

## Appendix J

Spread Property	HTML Tag/Attribute	Discussion	Log #																
FontSize	FONT SIZE attribute	<p>Translates the font size from points to a value between 1 and 7 as follows:</p> <table border="1"> <thead> <tr> <th>Font size in points</th> <th>Font size attribute</th> </tr> </thead> <tbody> <tr> <td>&lt;10</td> <td>1</td> </tr> <tr> <td>&lt;12, but ≥10</td> <td>2</td> </tr> <tr> <td>&lt;14, but ≥12</td> <td>3</td> </tr> <tr> <td>&lt;18, but ≥14</td> <td>4</td> </tr> <tr> <td>&lt;24, but ≥18</td> <td>5</td> </tr> <tr> <td>&lt;36, but ≥24</td> <td>6</td> </tr> <tr> <td>≥36</td> <td>7</td> </tr> </tbody> </table>	Font size in points	Font size attribute	<10	1	<12, but ≥10	2	<14, but ≥12	3	<18, but ≥14	4	<24, but ≥18	5	<36, but ≥24	6	≥36	7	
Font size in points	Font size attribute																		
<10	1																		
<12, but ≥10	2																		
<14, but ≥12	3																		
<18, but ≥14	4																		
<24, but ≥18	5																		
<36, but ≥24	6																		
≥36	7																		
FontStrikethru	S tag	Sets strike through using strikethrough tag.																	
FontUnderline	U tag	Sets underline using underline tag.																	
ForeColor	TH and TD tags, FONT COLOR attribute	Displays the text color assigned to the individual cell, the row, the column, or the spreadsheet, in that order. The colors in the fpSpread control are translated to the closest color of the 216 defined web-compliant colors.																	
Formula	HREF tag if URL function is in formula	<p>Current value in the cell at time of export goes into HTML file. The log file issues a message.</p> <p><b>Tip</b> Before you perform the export, recalculate the formulas in the spreadsheet to be sure you are exporting the most current value.</p> <p>If the URL function is used in the formula, the information in the function is used to create an HREF tag (hypertext link) in the exported HTML table.</p>	224																

Spread Property	HTML Tag/Attribute	Discussion	Log #
GridShowHoriz	TABLE tag, RULES attribute	If the GridShowHoriz property is set to True, the RULES attribute for the table is set to ROWS. If both the GridShowHoriz and the GridShowVert are set to False, the RULES attribute is set to NONE. Note that if you set the BORDER attribute to 0, no grid lines are displayed.	
GridShowVert	TABLE tag, RULES attribute	If the GridShowVert property is set to True, the RULES attribute for the table is set to COLS. If both the GridShowHoriz and the GridShowVert are set to False, the RULES attribute is set to NONE. Note that if you set the BORDER attribute to 0, no grid lines are displayed.	
LockBackColor	TH and TD tags, BGCOLOR attribute	Displays the locked cell background color. The colors in the fpSpread control are translated to the closest color of the 216 defined web-compliant colors.	
LockForeColor	TH and TD tags, FONT COLOR attribute	Displays the locked cell text color. The colors in the fpSpread control are translated to the closest color of the 216 defined web-compliant colors.	

Spread Property	HTML Tag/Attribute	Discussion	Log #
MaxCols		<p>Using the ExportToHTML method or the SSEXportToHTML function, the fpSpread control exports all the columns and rows that contain data in the spreadsheet to the HTML table. If you specify a range of cells using the ExportRangeToHTML method or the SSEXportRangeToHTML function, all the cells you specify are exported to the HTML table, whether those cells have data or not.</p> <p>Remember to consider your users' system resources and design your exported table's size accordingly.</p>	
MaxRows		<p>Using the ExportToHTML method or the SSEXportToHTML function, the fpSpread control exports all the columns and rows that contain data in the spreadsheet to the HTML table. If you specify a range of cells using the ExportRangeToHTML method or the SSEXportRangeToHTML function, all the cells you specify are exported to the HTML table, whether those cells have data or not.</p> <p>Remember to consider your users' system resources and design your exported table's size accordingly.</p>	
RowHeaderDisplay		Sets text in row headers in table heading.	
RowHeight	TD and TH tags, HEIGHT attribute	The heights of the exported rows are set (in pixels) to create rows that are the same heights as the Spread rows.	

Spread Property	HTML Tag/Attribute	Discussion	Log #
RowHidden		Spread does not export hidden rows to the HTML file.	
ShadowColor	TH and TD tags, BGCOLOR attribute	Displays the background color assigned to the headers and static text cells. The colors in the fpSpread control are translated to the closest color of the 216 defined web-compliant colors.	
ShadowText	TH and TD tags, FONT COLOR attribute	Displays the text color assigned to the headers and static text cells. The colors in the fpSpread control are translated to the closest color of the 216 defined web-compliant colors.	
StartingColNumber		Numbering used in Spread exports to the HTML table.	
StartingRowNumber		Numbering used in Spread exports to the HTML table.	
TwoDigitYearMax		The HTML table displays dates in the same format as they were displayed in the spreadsheet. Therefore, if the TypeDateCentury property is set to True so that dates display century digits, this value affects how the century values were determined in Spread, if the year values were provided as two-digit values.	
TypeHAlign	TD and TH tags, ALIGN attribute	Sets the horizontal alignment for the individual cell, the row, the column, or the spreadsheet, in that order.	
TypeVAlign	TD and TH tags, VALIGN attribute	Sets the vertical alignment for the individual cell, the row, the column, or the spreadsheet, in that order.	

Spread Property	HTML Tag/Attribute	Discussion	Log #
VirtualMode		Set the VirtualMode property to False before exporting the spreadsheet from Spread. If you do not turn virtual mode off before exporting from Spread, the HTML table will contain only the data that is currently in the virtual buffer.	
<b>Other settings</b>			
Control border	TABLE tag, BORDER attribute	The export reads the NoBorder property setting. If the NoBorder property is set to True, the BORDER attribute is set to 0, and the HTML table will not display grid lines. If the NoBorder property is set to False, the HTML uses the default BORDER attribute setting of 1.	
Cell spacing	TABLE tag, CELLSPACING attribute	The cell spacing is set to 0 to create a single-line grid effect similar to Spread.	
	TABLE tag, CELLPADDING attribute	The cell padding is set to 0 to appear similar to Spread (which does not provide padding capability).	
Empty cells	&nbsp;	The export inserts a nonbreaking space in each empty cell.	
Leading or trailing spaces in cells	&nbsp;	The export inserts a nonbreaking space in place of each leading or trailing space in cells. Without these nonbreaking spaces, leading or trailing spaces within the HTML tags are ignored.	

## Log File Information

When you export a Spread file to an HTML file, Spread can generate a log file. The log file provides a number and a short explanation of the export action, and the column and row number of the cell for which the action occurred, if applicable.

The following table lists the possible numbers used in the log files.

Number	Log Message	Explanation
201	Column number is invalid. Must be $-1$ or greater. Export canceled.	You have specified an invalid column number, such as $-3$ . Specify a column number between $-1$ (all columns) and the maximum column number in the spreadsheet.
202	Second column number ( <i>number</i> ) for range method is invalid. Must be $-1$ or greater. Export canceled.	While exporting a range of cells, the second column number you specified is invalid, such as $-3$ . Specify a column number between $-1$ (all columns) and the maximum column number in the spreadsheet, making sure that the column number is greater than or equal to the column number specified for the first column you are exporting.
203	Row number is invalid. Must be $-1$ or greater. Export canceled.	You have specified an invalid row number, such as $-3$ . Specify a row number between $-1$ (all rows) and the maximum row number in the spreadsheet.
204	Second row number ( <i>number</i> ) for range method is invalid. Must be $-1$ or greater. Export canceled.	While exporting a range of cells, the second row number you specified is invalid, such as $-3$ . Specify a row number between $-1$ (all rows) and the maximum row number in the spreadsheet, making sure that the row number is greater than or equal to the row number specified for the first row you are exporting.
205	The directory you specified ( <i>directory</i> ) does not exist or is invalid. Changing to the current directory ( <i>directory</i> ).	You have specified to export to or create a log file in a directory that does not exist on your system or that is invalid. The file will be created in your current directory.
206	The file you specified ( <i>filename</i> ) is invalid. Provide the name of a valid file. Export canceled.	You have specified an invalid file name. The file name might be invalid because you used incorrect syntax or punctuation. Try the export again and provide a valid file name.
207	The column range you have specified is invalid. The first column must be less than or equal to the second column ( <i>number</i> ). Export canceled.	Try the export again and specify the second column number as a column number greater than or equal to the first column number you specify.
208	The row range you have specified is invalid. The first row must be less than or equal to the second row ( <i>number</i> ). Export canceled.	Try the export again and specify the second row number as a row number greater than or equal to the first row number you specify.

## Appendix J

Number	Log Message	Explanation
209	Spread cannot open the file ( <i>filename</i> ) for writing. Export canceled.	You have specified to export to a file that cannot be opened. The export might be unable to open the file because the file is read only or another user might already be accessing the file. Check the file you specified, and then try the export again, specifying a different file name if necessary.
210	Spread has successfully created the file ( <i>filename</i> ).	The export has successfully completed and created the HTML file.
211	You specified the file ( <i>filename</i> ) to append to, but that file does not exist. Provide the name of an existing file. Export canceled.	You have tried to export and append to an existing file, but that file does not exist. Try the export again, correcting the export file name to the correct, existing file, or changing the <i>Append</i> parameter setting such that the export does not append to a file.
212	Spread has successfully exported columns <i>number</i> through <i>number</i> , rows <i>number</i> through <i>number</i> to HTML file ( <i>filename</i> ).	Your range export has completed successfully and the export has created the specified HTML file or the range has successfully been appended to the specified HTML file.
213	Spread export failed for HTML file ( <i>filename</i> ). No data present for specified columns and rows.	An HTML file was not created because the created file would not contain data. Check your settings to ensure that you have correctly specified the columns and rows you want to export.
214	The specified ending column ( <i>number</i> ) is greater than the maximum number of columns ( <i>number</i> ). Spread changed second column specified ( <i>number</i> ) to the maximum number of columns ( <i>number</i> ).	You have specified to export more columns than the spreadsheet contains. The export has adjusted the export to export the maximum number of columns that exist in the spreadsheet. For example, if you specified to export 100 columns, but the spreadsheet only contains 50 columns, the export adjusts to export just the existing 50 columns.
215	The specified ending row is greater than the maximum number of rows ( <i>number</i> ). Spread changed second row specified ( <i>number</i> ) to the maximum number of rows ( <i>number</i> ).	You have specified to export more rows than the spreadsheet contains. The export has adjusted the export to export the maximum number of rows that exist in the spreadsheet. For example, if you specified to export 100 rows, but the spreadsheet only contains 50 rows, the export adjusts to export just the existing 50 rows.

Number	Log Message	Explanation
216	The specified column has not been exported because it is hidden.	Hidden columns are not exported from Spread to the HTML table. If you wanted the specified column to be exported, change the column in Spread to no longer be hidden, then try the export again.
217	The specified row has not been exported because it is hidden.	Hidden rows are not exported from Spread to the HTML table. If you wanted the specified row to be exported, change the row in Spread to no longer be hidden, then try the export again.
218	The specified cell has not been exported because it is a check box cell.	Check box cells are not exported as check box cells to the HTML table. However, the value of the check box at the time of the export is exported to the HTML table.
219	The specified cell has not been exported because it is a combo box cell.	Combo box cells are not exported as combo box cells to the HTML table. However, the value in the edit field of the combo box at the time of the export is exported to the HTML table.
220	The specified cell has not been exported because it is an owner drawn cell.	Owner drawn cells are not exported to the HTML table.
221	The specified cell contains a bitmap. It was converted to a JPEG ( <i>filename</i> ).	Bitmap files in the spreadsheet are converted to JPEG files during export to HTML.
222	The specified cell contains an icon. It was converted to a JPEG ( <i>filename</i> ).	Icon files in the spreadsheet are converted to JPEG files during export to HTML.
223	The specified cell's picture could not be converted to a JPEG.	Spread could not convert the picture in the specified cell to a JPEG.
224	The specified cell contains a bitmap that could not be converted to a true color bitmap for JPEG conversion.	Spread could not convert the bitmap in the specified cell to a true color bitmap. You might want to use image editing software to convert the bitmap to a true color bitmap, and then try the HTML export again.
225	The specified cell contains an icon that could not be converted to a true color bitmap for JPEG conversion.	Spread could not convert the icon in the specified cell to a true color bitmap. You might want to use image editing software to convert the icon to a true color bitmap, and then try the HTML export again.
226	The specified cell has not been exported because it is a button cell.	Button cells are not exported to the HTML table.

## Appendix J

Number	Log Message	Explanation
227	The specified cell has not been exported because it is unknown.	The export encountered a cell type it did not recognize. The cell was not exported.
228	The specified cell contains a formula. The current value of the formula has been exported.	Formulas are not exported to the HTML table, because HTML tables do not support formulas. The current value of the formula is exported.  <b>Tip</b> Before you perform the export, recalculate the formulas in the spreadsheet to be sure you are exporting the most current value.
229	The picture for this cell must be either an icon or a bitmap.	The picture in the cell cannot be converted to a JPEG because it is not a bitmap or an icon.

# Index

## Numerics

12-hour format, time cells 265

24-hour format, time cells 265

## A

abort messages 60, 142

ABS function 430

absolute cell references 103

absolute value function 430

ACOS function 430

action keys, changing 302

Action property

active cell 189

cell borders 202

columns 161, 164

copying data 86, 88

cutting data 94

data binding 280

deleting

columns 164

data 98

rows 164

inserting

columns 161

rows 161

moving data 92

pasting data 88, 94

printing 130, 141

resetting spreadsheet properties 127

rows 161, 164

saving data 280

selecting blocks 88, 94, 191

sorting data 100

spreadsheet borders 202

swapping data 97

active cell

coordinates 189

focus 189

active cell (*continued*)

focus rectangle 189

preventing focus movement 300

setting 189

specifying 189

active view

applying changes 67

defined 58

printing 58

saving changes 67

ActiveCol property 189

ActiveRow property 189

ActiveX Data Objects (ADO) 269, 271

ADD function 430

AddCustomFunction function 105

AddCustomFunction method 105

AddCustomFunctionExt function 105

AddCustomFunctionExt method 105

adding

columns 161

data

formatted 75–77

notation 76

overview 75

unformatted 75, 79

formulas 102

rows 161

addition function 430

addition operator 429

address, FarPoint Technologies xx

ADO (ActiveX Data Objects) 269, 271

alignment

drop-down list, combo box cells 228

edit field, combo box cells 225

pictures

button cells 213

check box cells 220, 222

picture cells 255

## Index

### alignment (*continued*)

- text
  - button cells 211
  - check box cells 220, 221, 222
  - date cells 233
  - edit cells 238
  - float cells 242
  - integer cells 249
  - PIC (mask) cells 253
  - static text cells 258
  - time cells 264
- AllowCellOverflow property 198
- AllowDragDrop property 86, 93, 291
- AllowMultiBlocks property 191
- AllowUserFormulas property 107
- AllowUserZoom property 155
- And (logical operator) 429
- Appearance property 288
- arccosine function 430
- arcsine function 431
- arctangent 2 function 431
- arctangent function 431
- arrays
  - saving data from a spreadsheet 126
  - using to add data to a spreadsheet 75, 77
- arrow keys 301
- ArrowsExitEditMode property 301
- ASIN function 431
- ATAN function 431
- ATAN2 function 431
- AutoCalc property 107
- AutoClipboard property 301
- AutoSize property 173
- AVERAGE function 432

## B

- BackColor property 205
- BackColorStyle property 283
- background color
  - button cells 218
  - cells 205
  - headers 185
  - locked cells 195

### background color (*continued*)

- odd and even rows 186
- overlapping grid lines 283
- selected cells 207
- spreadsheet 205
- text tips 311
- balloon help, Spread Designer 71
- base e functions 437, 443
- beeping, spreadsheet 297
- binary files
  - loading 81
  - saving 115
- binding to databases
  - ADO 271
  - cell types 278
  - columns 273, 276
  - DAO 270
  - default settings 269
  - fields 269, 273
  - headers 275
  - loading data 269, 273, 278
  - overview 26
  - RDO 270
  - scrolling 280
  - setting up 269
  - virtual mode 282
- BlockMode property 190
- blocks of cells
  - copying 86
  - discontiguous selection 191
  - moving 92
  - printing 62, 147
  - restricting selection 126
  - selecting multiple 191
  - specifying 190
- booleans
  - SSB\_ALLOWCELLOVERFLOW 198
  - SSB\_ALLOWDRAGDROP 86, 93, 291
  - SSB\_ALLOWMULTIBLOCKS 191
  - SSB\_ALLOWUSERFORMULAS 107
  - SSB\_ARROWSEXITEDITMODE 301
  - SSB\_AUTOCALC 107
  - SSB\_AUTOCLIPBOARD 301

booleans (*continued*)

- SSB\_AUTOSIZE 173
- SSB\_EDITMODEPERMANENT 200, 300
- SSB\_EDITMODEREPLACE 201
- SSB\_FORMULASYNC 107
- SSB\_HORZSCROLLBAR 293
- SSB\_HSCROLLBARTRACK 295
- SSB\_MOVEACTIVEONFOCUS 300
- SSB\_NOBEEP 297
- SSB\_NOBORDER 289
- SSB\_PROCESSTAB 299
- SSB\_PROTECT 193, 194
- SSB\_REDRAW 286
- SSB\_RETAINSELBLOCK 290
- SSB\_SCROLLBAREXTMODE 294
- SSB\_SCROLLBARMAXALIGN 177
- SSB\_SCROLLBARSHOWMAX 294
- SSB\_SHOWCOLHEADERS 183
- SSB\_SHOWROWHEADERS 183
- SSB\_VERTSCROLLBAR 136, 293
- SSB\_VIRTUALMODE 131
- SSB\_VSCROLLBARTRACK 295

## borders

- button cells 215
- cells 202
- color
  - button cells 215
  - cells 202
- preview page 158
- printing 60, 142
- spreadsheet
  - hiding 289
  - three-dimensional appearance 288

## buffers

- loading files from 81
- rows in virtual 134
- saving files to 115
- virtual mode 134

## button cells

- aligning
  - pictures 213
  - text 211

button cells (*continued*)

- color
  - background 218
  - border 215
  - foreground 218
  - highlight 216
  - shadow 216
  - text 218
- creating 209
- displaying 305
- one-state 210
- overview 209
- pictures 213
- shadow size 216
- text 211
- three-dimensional appearance 216
- two-state 210

ButtonDrawMode property 305

**C**

calendar, popup 237

CEILING function 432

ceiling rounding function 432

cell reference functions

- ISEMPTY 441
- ISEVEN 441
- ISNONTEXT 441
- ISNUMBER 442
- ISODD 442
- ISREF 442
- ISTEXT 443

cell types

assigning when data binding 278

button cells

- aligning pictures 213
- aligning text 211
- background color 218
- border color 215
- creating 209
- displaying 305
- foreground color 218
- highlight color 216
- one-state 210

## Index

### cell types (*continued*)

#### button cells (*continued*)

- overview 209
- pictures 213
- shadow color 216
- shadow size 216
- text 211
- text color 218
- three-dimensional appearance 216
- two-state 210

#### check box cells

- aligning pictures 220, 222
- aligning text 221
- creating 220
- overview 219
- pictures 220, 222
- text 221
- three-state 220
- two-state 220

#### combo box cells

- creating 225
- displaying buttons 305
- drop-down list alignment 228
- drop-down list width 228
- text 225
- using the fpCombo control 230

#### date cells

- aligning text 233
- century values 233, 235
- creating 233
- formatting 235
- maximum value 233
- minimum value 233
- overview 233
- popup calendar 237
- separator 235
- spin buttons 235
- Year 2000 handling 233, 235

#### edit cells

- aligning text 238
- creating 238
- limiting user entry 240
- lowercase text 238

### cell types (*continued*)

#### edit cells (*continued*)

- overview 238
- password field 238
- text 238
- uppercase text 238

#### float cells

- aligning text 242
- creating 242
- currency symbol 244, 247
- decimal places 242
- decimal symbol 244, 247
- default formatting characters 247
- formatting 244
- maximum value 242
- minimum value 242
- separator 244, 247

#### integer cells

- creating 249
- maximum value 249
- minimum value 249
- overview 249
- spin buttons 251

#### mask (PIC) cells 253

#### owner-drawn cells 252

#### PIC (mask) cells 253

#### picture cells 255

#### setting 209

#### static text cells

- aligning text 258
- creating 258
- overview 258
- shadow color 258
- shadow text 258
- text 258
- three-dimensional appearance 261

#### time cells

- 24-hour format 265
- creating 264
- formatting 265
- maximum value 264
- minimum value 264
- overview 263

- cell types (*continued*)
  - time cells (*continued*)
    - seconds, displaying 265
    - separator 265
    - spin buttons 264
- CellBorderColor property 202
- CellBorderStyle property 202
- CellBorderType property 202
- cells
  - active
    - coordinates 189
    - focus 189
    - focus rectangle 189
    - preventing focus movement 300
    - setting 189
    - specifying 189
  - attached characteristics 32
  - blocks
    - discontiguous selection 191
    - moving 92
    - printing 62, 147
    - selecting multiple 191
    - selection, restricting 126
    - specifying 190
  - borders 202
  - color
    - background 205
    - border 202
    - foreground 205
    - locked 195
    - selected 207
  - creating different types 209
  - data 75, 110
  - data binding 278
  - dragging and dropping 291
  - font 207
  - formulas 102
  - locked
    - clearing 194
    - color 195
    - marking as locked 192
    - overview 192
    - protecting the spreadsheet 193
- cells (*continued*)
  - locked (*continued*)
    - removing protection 194
    - setting 193
    - unlocking the spreadsheet 194
  - marking as locked 192, 194
  - moving cursor inside 301
  - moving to a new location 291
  - notation 103
  - overflow 198
  - overview 22
  - precedence 32
  - preventing focus movement 300
  - reference style 103
  - references 103
  - selecting 126
  - setting properties for 31, 32, 51
  - specifying 31, 51
  - types
    - button 209
    - check box 219
    - combo box 225
    - date 233
    - edit 238
    - float 242
    - integer 249
    - mask (PIC) 253
    - owner-drawn 252
    - PIC (mask) 253
    - picture 255
    - static text 258
    - time 263
- CellType property
  - button cells 210–218
  - check box cells 220, 221, 222
  - combo box cells 225
  - data binding 278
  - date cells 233, 235
  - edit cells 238, 240
  - float cells 242, 244
  - integer cells 249, 251
  - mask (PIC) cells 253
  - owner-drawn cells 252

## Index

- CellType property (*continued*)
  - PIC (mask) cells 253
  - picture cells 255
  - static text cells 258, 261
  - time cells 264, 265
- century values 233, 235
- CFGetCellParam function 105
- CFGetCellParam method 105
- CFGetDoubleParam function 105
- CFGetDoubleParam method 105
- CFGetDoubleParamExt function 105
- CFGetDoubleParamExt method 105
- CFGetLongParam function 105
- CFGetLongParam method 105
- CFGetParamInfo function 105
- CFGetParamInfo method 105
- CFGetRangeParam function 105
- CFGetRangeParam method 105
- CFGetStringParam function 105
- CFGetStringParam method 105
- CFSetResult function 105
- CFSetResult method 105
- check box cells
  - aligning
    - pictures 220, 222
    - text 221
  - creating 220
  - overview 219
  - pictures 220, 222
  - text 221
  - three-state 220
  - two-state 220
- circular references, in formulas 109
- clearing data 98
- Clip property 75–77, 110, 111
- Clipboard
  - copying data 88
  - moving data 94
  - shortcut keys 27, 301
- ClipboardOptions property 88, 94
- ClipValue property 75–79, 110, 112
- code, generating in Spread Designer 53
- Col parameter 31
- Col property 31, 190
- Col2 property 190
- ColHeaderDisplay property 179, 275
- ColHidden property 119, 175
- color
  - background
    - button cells 218
    - cells 205
    - headers 185
    - locked cells 195
    - odd and even rows 186
    - overlapping grid lines 283
    - selected cells 207
    - spreadsheet 205
    - text tips 311
  - borders
    - button cells 215
    - cells 202
  - button cells
    - borders 215
    - text 218
    - three-dimensional appearance 216
  - column headers 185
  - gray area 287
  - grid lines 283
  - headers
    - background 185
    - text 185
    - three-dimensional appearance 185
  - highlight
    - button cells 216
    - headers 185
  - locked cells 195
  - odd and even rows 186
  - overlapping grid lines 283
  - printing 60, 142
  - row headers 185
  - selected cells 207
  - shadow
    - button cells 216
    - headers 185

color (*continued*)

## text

- button cells 218
  - cells 205
  - headers 185
  - locked cells 195
  - odd and even rows 186
  - selected cells 207
  - spreadsheet 205
  - text tips 311
- three-dimensional appearance
- button cells 216
  - headers 185

Color property page 47

ColPageBreak property 63, 149

ColsFrozen property 177

## columns

- adding 161
- background color 205
- binding to databases 273, 276
- data binding 273, 276
- deleting 164
- displaying 173–177
- freezing 177
- headers
  - color 185
  - content 179
  - height 168
  - hiding 183
  - letters 179
  - numbers 179
  - overview 179
  - printing 58, 142
  - starting letter or number 182
  - text 179, 180
  - three-dimensional appearance 184, 185
- hiding 175
- inserting 161
- leftmost displayed 174
- maximum number of 172
- number of 172
- number of displayed 173

columns (*continued*)

- overview 23
- page breaks, printing 63, 149
- position 174, 177
- precedence 32
- resizing 170
- scrolling 177
- selection, restricting 126
- setting properties for 31, 51
- sizing in bound spreadsheets 276
- specifying 31, 51
- width
  - based on text 169, 196
  - measurement unit 285
  - preventing resizing 170
  - setting 167

ColWidth property 167, 169, 196, 276

COMBIN function 433

combination function 433

combo box cells

- creating 225
- displaying buttons 305
- drop-down list
  - alignment 228
  - width 228
- text 225
- using the fpCombo control 230

comparison logical function 440

computer requirements xv

contacting FarPoint Technologies xix

CONTROL statement 468

conventions, documentation xvi

coordinates, active cell 189

copying data 84

COS function 433

COSH function 434

cosine function 433

CreateWindow function 468

currency symbol, in float cells 244, 247

CursorIcon property 296

cursors 296

CursorStyle property 296

## Index

CursorType property 296  
custom functions 105  
CustomFunction event 105

## D

DAO (Data Access Objects) 269, 270  
data

- adding
  - formatted 75–77
  - notation 76
  - overview 75
  - unformatted 79
- clearing 98
- copying 84
- deleting 98
- dragging and dropping 291
- flags for changes 128
- formatted
  - adding 75, 77
  - overview 30
  - returning 111
- monitoring changes 128
- moving 91
- overflow 198
- printing only 62, 146
- returning 110
- saving 115, 126
- sorting 100
- swapping 97
- unformatted
  - adding 75
  - overview 30
  - returning 112

Data Access Objects (DAO) 269, 270

data binding

- ADO 271
- cell types 278
- columns 273, 276
- DAO 270
- default settings 269
- fields 269, 273
- headers 275
- loading data 269, 273, 278

data binding (*continued*)

- overview 26
- RDO 270
- scrolling 280
- setting up 269
- virtual mode 282

DatabaseName property 270

DataColConfig event 276, 278

DataField property 273

DataMember property 271

DataSource property 270, 271

date cells

- aligning text 233
- century values 233, 235
- creating 233
- formatting 235
- maximum value 233
- minimum value 233
- overview 233
- popup calendar 237
- separator 235
- spin buttons 235
- Year 2000 handling 233, 235

date formats 235

DAutoCellTypes property 278

DAutoFill property 280

DAutoHeadings property 275

DAutoSave property 280

DAutoSizeCols property 276

DB function 434

DDB function 435

decimal places, in float cells 242

decimal symbol, in float cells 244, 247

default (normal) operation mode 128

default settings, resetting 127

degrees conversion function 436

DEGREES function 436

deleting

- cell properties 98
- columns 164
- data 98
- rows 164

- depreciation functions
  - DB function 434
  - DDB function 435
  - SLN function 457
  - SYD function 460
- DestCol property 86, 92, 97
- DestRow property 86, 92, 97
- DInformActiveRowChange property 280
- DisplayColHeaders property 183
- DisplayRowHeaders property 183
- division operator 429
- DLL control
  - CONTROL statement 468
  - CreateWindow function 468
  - creating a Spread control 468
  - fpSpread control components 36
  - fpSpreadPreview control components 44
  - Get functions 469
  - loading and unloading libraries 467
  - Set functions 469
  - string data 469
  - using 467
- documentation
  - conventions xvi
  - online help xvii
  - overview xvi
- double-declining balance depreciation 435
- drag-and-drop operations
  - cells 291
  - copying data 86
  - dragging and dropping cells 86, 93
  - moving data 93
- DragDropBlock event 291
- DrawItem event 252
- drop-down list boxes, providing 225
- E**
- e functions
  - EXP function 437
  - LN function 443
- edit cells
  - aligning text 238
  - creating 238
- edit cells (*continued*)
  - limiting user entry 240
  - lowercase text 238
  - overview 238
  - password field 238
  - text 238
  - uppercase text 238
- edit mode
  - exiting using an arrow key 301
  - making permanent 200
  - replacing text 201
  - turning on and off 200
  - using 199
- EditEnterAction property 298
- EditMode event 199
- EditMode property 200
- EditModePermanent property 200, 300
- EditModeReplace property 201
- elements, spreadsheet
  - precedence 32
  - specifying 31, 51
- e-mail sales, FarPoint Technologies xix
- e-mail support, FarPoint Technologies xix
- EVEN function 437
- even rounding function 437
- events
  - fpSpread control 34
  - fpSpreadPreview control 43
- Excel files
  - exporting 69, 117
  - importing 81
  - Spread conversion rules 471
- EXP function 437
- exponent base e function 437
- exponentiation operator 429
- exporting
  - Excel files 69, 117, 471
  - HTML files 69, 119, 463, 505
- ExportRangeToHTML method 119
- ExportToExcel method 117
- ExportToHTML method 119

## Index

extended-selection list box, spreadsheet operating as 128

extended-selection operation mode 128

## F

F2, F3, F4 action keys 302

FACT function 437

factorial function 437

FALSE function 438

false logical function 438

FarPoint Technologies, contacting xix

fax number, FarPoint Technologies xix

features

- fpSpread control 21

- fpSpreadPreview control 39

files

- loading 81

- saving 115

financial functions

- DB 434

- DDB 435

- FV 439

- NPER 447

- PMT 449

- PV 451

- SLN 457

- SYD 460

fixed-declining balance depreciation 434

flags, monitoring data changes 128

flickering 286

float cells

- aligning text 242

- creating 242

- currency symbol 244, 247

- decimal places 242

- decimal symbol 244, 247

- default formatting characters 247

- formatting 244

- maximum value 242

- minimum value 242

- separator 244, 247

FloatDefCurrencyChar property 244, 247

FloatDefDecimalChar property 244, 247

FloatDefSepChar property 244, 247

FLOOR function 438

floor rounding function 438

focus rectangle

- active cell 189

- hiding 290, 300

- moving 298, 301

- preventing movement 300

Font property 207

FontBold property 207

FontItalic property 207

FontName property 207

fonts, setting 207

FontSize property 207

FontStrikethru property 207

FontUnderline property 207

footers, printing 64, 151

ForeColor property 205

foreground color

- button cells 218

- cells 205

- headers 185

- locked cells 195

- odd and even rows 186

- selected cells 207

- spreadsheet 205

- text tips 311

format

- date cells

  - century values 233, 235

  - separator 235

  - spin buttons 235

- float cells

  - currency symbol 244, 247

  - decimal symbol 244, 247

  - separator 244, 247

- time cells 265

formatted data

- adding 75–77

- overview 30

- returning 111

Formula property 102

- formulas
  - absolute cell references 103
  - adding 102
  - allowing user entered 107
  - cell references 103
  - circular references 109
  - custom functions 105
  - functions 429
  - iterations 109
  - operators 429
  - overview 26, 102
  - recalculating 107
  - relative cell references 103
  - updating 107
- FormulaSync property 107
- fpSpread control
  - ActiveX control components 34
  - components 34
  - DLL control components 36
  - events 34
  - features 21
  - functions
    - DLL control 36
    - VBX control 35
  - messages 36
  - methods 35
  - Spread Designer 21
  - structures 37
  - styles 37
  - VBX control components 34
- fpSpreadPreview control
  - ActiveX control components 43
  - changing page size 154
  - DLL functions 44
  - events 43
  - gray area 41, 158
  - messages 44
  - overview 39
  - page
    - borders 41, 158
    - gutters 41, 159
    - shadow 41, 157
  - pages per screen 154
  - fpSpreadPreview control (*continued*)
    - previewing the spreadsheet 153
    - print preview 39
    - properties 43
    - property pages 45
    - scroll bars 42, 156
    - VBX control components 43
    - view, customizing 154
    - zooming 40, 155
  - freezing columns or rows 177
  - FTP site, FarPoint Technologies xix
  - functions
    - custom 105
    - DLL
      - fpSpread control 36
      - fpSpreadPreview control 44
  - financial
    - DB 434
    - DDB 435
    - FV 439
    - NPER 447
    - PMT 449
    - PV 451
    - SLN 457
    - SYD 460
  - formulas 429
  - logical
    - FALSE 438
    - IF 440
    - NOT 447
    - TRUE 462
  - mathematical
    - ABS 430
    - ADD 430
    - CEILING 432
    - DEGREES 436
    - EVEN 437
    - EXP 437
    - FACT 437
    - FLOOR 438
    - INT 440
    - INVERSE 440
    - LN 443

## Index

### functions (*continued*)

#### mathematical (*continued*)

- LOG 443
- LOG10 444
- MAX 444
- MIN 445
- MOD 445
- NEG 446
- ODD 448
- PI 449
- POWER 450
- PRODUCT 450
- RADIANS 452
- RAND 452
- ROUND 455
- ROUNDUP 455
- SIGN 456
- SQRT 457
- SQUARE 458
- SUM 460
- SUMSQ 460
- TRUNCATE 462
- XROOT 465

#### reference

- ISEMPTY 441
- ISEVEN 441
- ISNONTEXT 441
- ISNUMBER 442
- ISODD 442
- ISREF 442
- ISTEXT 443

#### statistical

- AVERAGE 432
- COMBIN 433
- MEDIAN 445
- MODE 446
- PERMUT 448
- RANK 453
- STDEV 458
- STDEVP 459
- VAR 463
- VARP 464

### functions (*continued*)

#### trigonometric

- ACOS 430
- ASIN 431
- ATAN 431
- ATAN2 431
- COS 433
- COSH 434
- SIN 456
- SINH 456
- TAN 461
- TANH 461

#### URL 463

- VBX, fpSpread control 35

- future value function 439

- FV function 439

## G

- General property page 45

- Get functions, DLL control 469

- GetActionKey function 302

- GetActionKey method 302

- GetArray function 126

- GetArray method 126

- GetCellDirtyFlag function 128

- GetCellDirtyFlag method 128

- GetExcelSheetList method 81

- GetFloat function 110

- GetFloat method 110

- GetFormulaSync function 107

- GetInteger function 110

- GetInteger method 110

- GetIteration function 109

- GetIteration method 109

- GetNextPageBreakCol function 149

- GetNextPageBreakRow function 149

- GetOddEvenRowColor function 186

- GetOddEvenRowColor method 186

- GetPrintOptions function 141, 142

- GetPrintPageCount function 142

- GetRefStyle function 103

- GetRefStyle method 103

- GetText function 110, 111

- GetText method 110, 111
- GetTextTipAppearance function 311
- GetTextTipAppearance method 311
- GetTwoDigitYearMax function 233
- gray area
  - fpSpread control 287
  - fpSpreadPreview control 41, 158
- GrayAreaBackColor property 287
- GrayAreaColor property 158
- GrayAreaMarginH property 158
- GrayAreaMarginType property 158
- GrayAreaMarginV property 158
- grid lines
  - color 283
  - displaying 283
  - hiding 283
  - overview 28
  - printing 60, 142
  - style 283
- GridColor property 283
- GridShowHoriz property 283
- GridShowVert property 283
- GridSolid property 283
- H**
- headers
  - binding to databases 275
  - color 185
  - column height 168
  - content 179
  - data binding 275
  - hiding 183
  - letters 179
  - numbers 179
  - overview 179
  - printing 58, 60, 64, 142, 151
  - row width 167
  - specifying 31, 51
  - starting letter or number 182
  - text 179
  - three-dimensional appearance 184, 185
- height
  - column headers 168
  - rows
    - based on text 169, 196
    - headers 168
    - preventing resizing 170
    - user interaction 170
- help
  - balloon, in Spread Designer 71
  - online, accessing xvii
  - technical support xvii, xix
- hiding
  - columns 175
  - focus rectangle 290, 300
  - grid lines 283
  - headers 183
  - rows 175
  - selections 290
  - spreadsheet borders 289
  - text tips 311
- highlight color
  - button cells 216
  - headers 185
- HTML files, exporting 69, 119, 463, 505
- hWndSpread property 153
- hyperbolic cosine function 434
- hyperbolic sine function 456
- hyperbolic tangent function 461
- I**
- IF function 440
- ImportExcelSheet method 81
- importing, Excel files 81, 471
- inserting
  - columns 161
  - rows 161
- installation information xv
- INT function 440
- integer cells
  - creating 249
  - maximum value 249

## Index

integer cells (*continued*)

    minimum value 249

    overview 249

    spin buttons 251

integer rounding function 440

INVERSE function 440

investment functions

    FV 439

    NPER 447

    PMT 449

    PV 451

IEMPTY function 441

ISEVEN function 441

IsExcelFile method 81

ISNONTEXT function 441

ISNUMBER function 442

ISODD function 442

ISREF function 442

ISTEXT function 443

iterations, in formulas 109

## J

Jet 3.51 OLE DB provider 271

job name, printing 60, 142

JPEG, HTML conversion 505

## L

learn mode 53

LeftCol property 174

letters, in headers 179

libraries, loading and unloading 467

link to Web function 463

list box, spreadsheet operating as 128

List Pro fpCombo control 230

LN function 443

LoadFromFile function 81

LoadFromFile method 81

loading

    binary files 81

    DLL libraries 467

    spreadsheets from a buffer 81

    tab-delimited files 81

LoadTabFile function 81

LoadTabFile method 81

Lock property 192, 194

LockBackColor property 195

locked cells

    color 195

    marking as locked 192, 194

    overview 192

    protecting the spreadsheet 193

    removing protection 194

    setting 193

    unlocking the spreadsheet 194

LockForeColor property 195

log files

    Excel 81

    HTML 81

LOG function 443

LOG10 function 444

logarithm base 10 function 444

logarithm function 443

logfiles

    Excel 471

    HTML 505

logical And operator 429

logical functions

    FALSE 438

    IF 440

    NOT 447

    TRUE 462

logical Or operator 429

long day names, popup calendar 237

long month names, popup calendar 237

## M

margins

    gray area for print preview 158

    printing 60, 142

mask (PIC) cells 253

mathematical constants, functions

    EXP function 437

    LN function 443

    PI function 449

- mathematical functions
  - ABS 430
  - ADD 430
  - CEILING 432
  - DEGREES 436
  - EVEN 437
  - EXP 437
  - FACT 437
  - FLOOR 438
  - INT 440
  - INVERSE 440
  - LN 443
  - LOG 443
  - LOG10 444
  - MAX 444
  - MIN 445
  - MOD 445
  - NEG 446
  - ODD 448
  - PI 449
  - POWER 450
  - PRODUCT 450
  - RADIANS 452
  - RAND 452
  - ROUND 455
  - ROUNDUP 455
  - SIGN 456
  - SQRT 457
  - SQUARE 458
  - SUM 460
  - SUMSQ 460
  - TRUNCATE 462
  - XROOT 465
- MAX function 444
- MaxCols property 172
- maximum
  - function 444
  - number of columns 172
  - number of rows 172
  - number of virtual rows 132
  - value in date cells 233
  - value in float cells 242
- MaxRows property 132, 172
- MaxTextCellHeight property 197
- MaxTextCellWidth property 196
- MaxTextColWidth property 169, 196
- MaxTextRowHeight property 169, 197
- mean (average) function 432
- MEDIAN function 445
- messages
  - fpSpread control 36
  - fpSpreadPreview control 44
- methods, fpSpread control 35
- MIN function 445
- minimum
  - function 445
  - value in date cells 233
  - value in float cells 242
- MOD function 445
- MODE function 446
- modes
  - edit mode
    - exiting 301
    - making permanent 200
    - replacing text 201
    - turning on and off 200
    - using 199
  - extended-selection 128
  - multiple-selection 128
  - operation mode 128
  - row mode 128
  - single-selection 128
  - virtual mode
    - maximum number of rows 132
    - number of rows in buffer 134
    - overview 130
    - rows, specifying 132, 134
    - scroll bar, customizing 134, 136
    - turning on 131
- monitoring data changes 128
- MoveActiveOnFocus property 300
- moving data 91
- moving focus
  - with the Enter key 298
  - with the Tab key 299
- multiple block selection 191

## Index

multiple-selection list box, spreadsheet operating as  
128

multiple-selection operation mode 128

multiplication operator 429

multiply function 450

## N

natural log function 443

navigation

    customizing spreadsheet 298

    keys 27

NEG function 446

negative value function 446

NoBeep property 297

NoBorder property 289

normal operation mode 128

NOT function 447

not logical function 447

notation

    cell 103

    data 76

NPER function 447

number of periods function 447

numbers, in headers 179

## O

odd and even rows, background color 186

ODD function 448

odd rounding function 448

online help xvii

operation modes 128

OperationMode property

    copying data using drag-and-drop 86

    dragging and dropping cells 291

    drawing button cells 305

    moving data using drag-and-drop 93

    specifying the operation mode 128

operators, in formulas 429

Or (logical operator) 429

overflow, cells 198

overlapping grid lines, background color 283

overview

    binding to databases 26

    button cells 209

    cells 22

    check box cells 219

    column headers 179

    columns 23

    combo box cells 225

    data

        exporting 25, 115

        formatted 30

        importing 25, 75

        loading 25, 75

        saving 25, 115

        unformatted 30

    data binding 26

    date cells 233

    documentation xvi

    edit cells 22, 238

    formatted data 30

    formulas 26, 102

    grid lines 28

    headers 179

    locked cells 192

    navigation 27

    printing 24, 139

    row headers 179

    rows 23

    Spread 3

    Spread Designer 21

    spreadsheet appearance 28

    spreadsheets 3

    unformatted data 30

    user interaction 29

    virtual mode 26, 130

owner-drawn cells 252

## P

page

    borders 41, 158

    breaks, printing 63, 149

    count 142

    gutters 41, 159

- page (*continued*)
  - order, printing 60, 142
  - orientation, printing 60, 142
  - printing 62, 147
  - shadow 41, 157
- Page property page 46
- PageBorderColor property 158
- PageBorderWidth property 158
- PageGutterH property 159
- PageGutterV property 159
- PageMultiCntH property 154
- PageMultiCntV property 154
- PageShadowColor property 157
- PageShadowWidth property 157
- PageViewPercentage property 154, 155
- PageViewType property 154, 155, 159
- password fields 238
- payment function 449
- periods in investment function 447
- PERMUT function 448
- permutation function 448
- phone number, FarPoint Technologies xix
- PI function 449
- PIC (mask) cells 253
- Picture property page 48
- pictures
  - button cells 213
  - check box cells 220, 222
  - HTML conversion 505
  - picture cells 255
- PMT function 449
- pointers 296
- popup calendar 237
- position
  - columns 174, 177
  - rows 174, 177
- POWER function 450
- precedence 32
- present value function 451
- preventing focus movement 300
- preview page
  - borders 158
  - customizing 154
- preview page (*continued*)
  - gray area 158
  - gutters 159
  - linking to the spreadsheet 153
  - scroll bars 156
  - shadow 157
  - zooming 155
- print preview 39, 65
- PrintAbortMsg property 60, 142
- PrintBorder property 60, 142
- PrintColHeaders property 60, 142
- PrintColor property 60, 142
- PrintFirstPageNumber property 60, 142
- PrintFooter property 64, 151
- PrintGrid property 60, 142
- PrintHeader property 64, 151
- printing
  - abort message 60, 142
  - active view in Spread Designer 58, 65
  - block of cells 62, 147
  - borders 60, 142
  - color 60, 142
  - data only 62, 146
  - footers 64, 151
  - grid lines 60, 142
  - headers 60, 64, 142, 151
  - job name 60, 142
  - margins 60, 142
  - overview 24, 139
  - page breaks 63, 149
  - page count 142
  - page order 60, 142
  - page orientation 60, 142
  - pages 62, 147
  - portion of spreadsheet 62, 146
  - previewing 39, 65
  - ranges 62, 147
  - three-dimensional appearance 60, 142
- PrintJobName property 60, 142
- PrintMarginBottom property 60, 142
- PrintMarginLeft property 60, 142
- PrintMarginRight property 60, 142
- PrintMarginTop property 60, 142

## Index

PrintNextPageBreakCol property 63, 149  
PrintNextPageBreakRow property 63, 149  
PrintOrientation property 60, 142  
PrintPageCount property 142  
PrintPageEnd property 62, 147  
PrintPageOrder property 60, 142  
PrintPageStart property 62, 147  
PrintRowHeaders property 60, 142  
PrintShadows property 60, 142  
PrintSmartPrint property 141  
PrintType property 62, 147  
PrintUseDataMax property 62, 130, 146, 147  
ProcessTab property 299  
PRODUCT function 450  
properties  
    fpSpreadPreview control 43  
    resetting to default settings 127  
property pages  
    Color 47  
    General 45  
    Page 46  
    Picture 48  
Protect property 193, 194  
PV function 451

## Q

QueryData event 134

## R

radians conversion function 452  
RADIANS function 452  
raise to power function 450  
RAND function 452  
random number function 452  
range operator 429  
ranges, printing 62, 147  
RANK function 453  
ranking function 453  
RDO (Remote Data Objects) 269, 270  
read only, making spreadsheet 128  
read-only operation mode 128  
ReCalcCell function 107  
ReCalcCell method 107

RecordSource property 270  
redistribution information xv  
ReDraw property 286  
reference functions  
    ISEMPTY 441  
    ISEVEN 441  
    ISNONTEXT 441  
    ISNUMBER 442  
    ISODD 442  
    ISREF 442  
    ISTEXT 443  
reference style 103  
references, circular 109  
relative cell references 103  
release notes xv  
Remote Data Objects (RDO) 269, 270  
RemoveCustomFunction function 105  
RemoveCustomFunction method 105  
resizing columns or rows 170, 276  
RetainSelBlock property 290  
root function 465  
roots of numbers functions 457, 465  
ROUND function 455  
rounding functions  
    CEILING 432  
    EVEN 437  
    FLOOR 438  
    INT 440  
    ODD 448  
    ROUND 455  
    ROUNDUP 455  
    TRUNCATE 462  
ROUNDUP function 455  
row height  
    based on text 169, 196  
    headers 168  
    preventing resizing 170  
    user interaction 170  
row operation mode 128  
Row parameter 31  
Row property 31, 190  
Row2 property 190  
RowHeaderDisplay property 179

RowHeight property 168, 169, 197

RowHidden property 119, 175

RowPageBreak property 63, 149

rows

adding 161

background color 205

buffers, virtual mode 134

color of odd and even 186

deleting 164

displaying 173–177

freezing 177

headers

color 185

content 179

hiding 183

letters 179

numbers 179

overview 179

starting letter or number 182

text 180

three-dimensional appearance 184, 185

width 167

height

based on text 169, 196

headers 168

preventing resizing 170

setting 168

unit of measurement 285

user interaction 170

hiding 175

inserting 161

maximum number of 132, 172

number of 172

number of displayed 173

odd and even background color 186

operation mode 128

overview 23

page breaks, printing 63, 149

position 174, 177

precedence 32

printing page breaks 63, 149

resizing 170

rows (*continued*)

scrolling 177

selection, restricting 126

setting properties for 31, 51

specifying 31, 51

topmost displayed 174

RowsFrozen property 177

## S

SaveTabFile function 115

SaveTabFile method 115

SaveToFile function 115

SaveToFile method 115

saving

files 115

Spread Designer changes 68

scroll bars

displaying 293

displaying only when needed 294

fpSpreadPreview control 42, 156

preview page 156

scroll box range 294

specifying when to display 293

synchronizing with columns and rows 295

vertical 134, 136

virtual mode 134, 136

scroll box

range 294

synchronizing with columns and rows 295

virtual mode 134

ScrollBarExtMode property 294

ScrollBarH property 156

ScrollBarMaxAlign property 177

ScrollBars property 136, 293

ScrollBarShowMax property 294

ScrollBarTrack property 295

ScrollBarV property 156

ScrollIncH property 156

ScrollIncV property 156

scrolling, columns or rows 177

SelBackColor property 207

SelectBlockOptions property 126

## Index

- selecting
  - cells 126
  - multiple blocks 191
- selection
  - discontiguous blocks 191
  - extended 128
  - hiding 290
  - multiple 128
  - multiple blocks 191
  - restricting 126
  - rows only 128
  - single 128
- SelForeColor property 207
- separator character
  - float cells 244, 247
  - time cells 265
- Set functions, DLL control 469
- SetActionKey function 302
- SetActionKey method 302
- SetArray function 75, 77
- SetArray method 75, 77
- SetCalText function 237
- SetCalText method 237
- SetCellDirtyFlag function 128
- SetCellDirtyFlag method 128
- SetFloat function 76
- SetFloat method 75, 76, 79
- SetFormulaSync function 107
- SetInteger function 76
- SetInteger method 75, 76, 79
- SetIteration function 109
- SetIteration method 109
- SetOddEvenRowColor function 186
- SetOddEvenRowColor method 186
- SetPrintOptions function 141, 142
- SetRefStyle function 103
- SetRefStyle method 103
- SetText function 75, 76, 77
- SetText method 75, 76, 77
- SetTextTipAppearance function 311
- SetTextTipAppearance method 311
- settings, default, resetting 127
- SetTwoDigitYearMax function 233
- shadow color
  - button cells 216
  - preview page 157
- shadow size, button cells 216
- shadow, preview page 157
- ShadowColor property 185, 258
- ShadowDark property 185, 261
- ShadowText property 185, 258
- short day names, popup calendar 237
- short month names, popup calendar 237
- shortcut keys 27, 301
- SIGN function 456
- SIN function 456
- sine function 456
- single-selection list box, spreadsheet operating as 128
- single-selection operation mode 128
- SINH function 456
- SLN function 457
- SmartPrint 141
- SortBy property 100
- sorting data 100
- SortKey property 100
- SortKeyOrder property 100
- spin buttons
  - integer cells 251
  - time cells 264
- Spread
  - changes from previous versions 8
  - document conventions xvi
  - documentation xvi, xvii
  - installation information xv
  - online help xvii
  - overview 3
  - redistribution information xv
  - release notes xv
  - system requirements xv
  - updates 8
- Spread Designer
  - active view 58
  - applying changes 67
  - balloon help 71
  - cell type mode 55

- Spread Designer (*continued*)
  - changes to spreadsheet 67
  - code, generating 53
  - colors representing cell types 55
  - design mode 52
  - exporting
    - Excel files 69
    - HTML files 69
  - generating code 53
  - help, balloon 71
  - learn mode 53
  - opening 49
  - opening multiple spreadsheets 57
  - overview 21
  - print preview 65
  - printing the active view 58
  - run-time mode 52
  - saving changes 67
  - saving changes to a file 68
  - Spread Designer view 58
  - spreadsheets, opening 57
  - starting 49
  - views of spreadsheets 58
  - visually indicating cell types 55
- Spread Designer view
  - applying changes 67
  - defined 58
  - saving changes 67
- SpreadAddCustomFunction function 105
- SpreadAddCustomFunctionExt function 105
- SpreadCFGetCellParam function 105
- SpreadCFGetDoubleParam function 105
- SpreadCFGetDoubleParamExt function 105
- SpreadCFGetLongParam function 105
- SpreadCFGetParamInfo function 105
- SpreadCFGetRangeParam function 105
- SpreadCFGetStringParam function 105
- SpreadCFSetResult function 105
- SpreadGetActionKey function 302
- SpreadGetArray function 126
- SpreadGetCellDirtyFlag function 128
- SpreadGetFloat function 110
- SpreadGetFormulaSync function 107
- SpreadGetInteger function 110
- SpreadGetIteration function 109
- SpreadGetNextPageBreakCol function 149
- SpreadGetNextPageBreakRow function 149
- SpreadGetOddEvenRowColor function 186
- SpreadGetPrintOptions function 141, 142
- SpreadGetPrintPageCount function 142
- SpreadGetRefStyle function 103
- SpreadGetText function 110, 111
- SpreadGetTextTipAppearance function 311
- SpreadGetTwoDigitYearMax function 233
- SpreadLoadFromFile function 81
- SpreadLoadTabFile function 81
- SpreadReCalcCell function 107
- SpreadRemoveCustomFunction function 105
- SpreadSaveTabFile function 115
- SpreadSaveToFile function 115
- SpreadSetActionKey function 302
- SpreadSetArray function 75, 77
- SpreadSetCalText function 237
- SpreadSetCellDirtyFlag function 128
- SpreadSetFloat function 75, 76, 79
- SpreadSetFormulaSync function 107
- SpreadSetInteger function 75, 76, 79
- SpreadSetIteration function 109
- SpreadSetOddEvenRowColor function 186
- SpreadSetPrintOptions function 141, 142
- SpreadSetRefStyle function 103
- SpreadSetText function 75, 76, 77
- SpreadSetTextTipAppearance function 311
- SpreadSetTwoDigitYearMax function 233
- spreadsheets
  - abort message, printing 60, 142
  - active cell
    - coordinates 189
    - focus 189
    - focus rectangle 189
    - preventing focus movement 300
    - setting 189
    - specifying 189
  - adding
    - columns 161
    - data 75

## Index

### spreadsheets (*continued*)

#### adding (*continued*)

formulas 102

rows 161

background color 205

beeping 297

binary files

loading 81

saving 115

binding to databases

ADO 271

cell types 278

columns 273, 276

DAO 270

default settings 269

fields 269, 273

headers 275

loading data 269, 273, 278

overview 26

RDO 270

scrolling 280

setting up 269

virtual mode 282

blocks of cells

copying 86

discontiguous selection 191

moving 92

printing 62, 147

restricting selection 126

selecting multiple 191

specifying 190

borders

button cells 215

cells 202

hiding 289

printing 60, 142

three-dimensional appearance 288

buffers

loading files from 81

rows in virtual 134

saving files to 115

virtual mode 134

### spreadsheets (*continued*)

#### button cells

aligning pictures 213

aligning text 211

background color 218

border color 215

color 215–218

creating 209

displaying 305

foreground color 218

highlight color 216

one-state 210

overview 209

pictures 213

shadow color 216

shadow size 216

text 211

text color 218

three-dimensional appearance 216

two-state 210

cell borders 202

cell references 103

cell types

button 209

check box 219

combo box 225

date 233

edit 238

float 242

integer 249

mask (PIC) 253

owner-drawn 252

PIC (mask) 253

picture 255

setting 209

static text 258

time 263

cells

color 195

formulas 102

locking 192

marking as locked 192, 194

notation 103

spreadsheets (*continued*)cells (*continued*)

- overflow 198
- overview 22
- precedence 32
- selection 126
- setting type 209
- specifying 31, 51
- types 209
- unlocking 194

## check box cells

- aligning pictures 220, 222
- aligning text 221
- creating 220
- overview 219
- pictures 222
- text 221
- three-state 220
- two-state 220

## circular references, formulas 109

## clearing data 98

## Clipboard operations 88, 94

## Clipboard shortcut keys 27, 301

## color

- background 205
- gray area 287
- grid lines 283
- headers 185
- locked cells 195
- odd and even rows 186
- overlapping grid lines 283
- printing 60, 142
- selected cells 207
- text 205
- text tips 311

## column headers 179

## columns

- adding 161
- background color 205
- binding to databases 273, 276
- data binding 273, 276
- deleting 164
- displaying 173–177

spreadsheets (*continued*)columns (*continued*)

- freezing 177
- headers 168, 179, 182, 183, 184, 185
- hiding 175
- inserting 161
- leftmost displayed 174
- maximum number of 172
- number of 172
- number of displayed 173
- overview 23
- position 174, 177
- precedence 32
- resizing 170
- scrolling 177
- selection 126
- setting properties for 31, 51
- sizing in bound spreadsheets 276
- specifying 31, 51
- unit of measurement 285
- width 167, 169, 170, 196, 285

## combo box cells

- creating 225
- displaying buttons 305
- drop-down list alignment 228
- drop-down list width 228
- text 225
- using the fpCombo control 230

## copying data 84

## cursors 296

## custom functions 105

## data

- adding formatted 75, 77
- adding unformatted 79
- adding, notation 76
- adding, overview 75
- clearing 98
- copying 84
- deleting 98
- dragging and dropping 291
- flags for changes 128
- formatted 30, 77, 111
- monitoring changes 128

## Index

### spreadsheets (*continued*)

#### data (*continued*)

- moving 91
- notation 76
- overflow 198
- returning 110
- saving 115, 126
- sorting 100
- swapping 97
- unformatted 30, 79, 112

#### data binding

- ADO 271
- cell types 278
- columns 273, 276
- DAO 270
- default settings 269
- fields 269, 273
- headers 275
- loading data 269, 273, 278
- overview 26
- RDO 270
- scrolling 280
- setting up 269
- virtual mode 282

#### date cells

- aligning text 233
- century values 233, 235
- creating 233
- formatting 235
- maximum value 233
- minimum value 233
- overview 233
- popup calendar 237
- separator 235
- spin buttons 235
- Year 2000 handling 233, 235

#### date formats 235

#### decimal places in float cells 242

#### decimal symbol in float cells 244, 247

#### default (normal) operation mode 128

#### default settings 127

### spreadsheets (*continued*)

#### deleting

- cell properties 98
- columns 164
- data 98
- rows 164

#### drag-and-drop operations

- cells 291
- copying data 86
- dragging and dropping cells 86, 93
- moving data 93

#### edit cells

- aligning text 238
- creating 238
- limiting user entry 240
- lowercase text 238
- overview 238
- password field 238
- text 238
- uppercase text 238

#### edit mode 199

#### elements

- precedence 32
- specifying 31, 51

#### Excel files

- exporting 69, 117, 471
- importing 81

#### exporting

- Excel files 69, 117, 471
- HTML files 69, 119, 463, 505

#### extended-selection list boxes 128

#### extended-selection mode 128

#### F2, F3, F4 action keys 302

#### features 21

#### files

- loading 81
- saving 115

#### flickering 286

#### float cells

- aligning text 242
- creating 242

- spreadsheets (*continued*)
  - float cells (*continued*)
    - currency symbol 244, 247
    - decimal places 242
    - decimal symbol 244, 247
    - formatting 244
    - formatting characters 247
    - maximum value 242
    - minimum value 242
    - separator 244, 247
  - focus rectangle
    - active cell 189
    - hiding 290, 300
    - moving 298, 301
    - preventing movement 300
  - fonts 207
  - footers, printing 64, 151
  - formatted data
    - adding 75–77
    - overview 30
    - returning 111
  - formulas
    - absolute cell references 103
    - adding 102
    - allowing user entered 107
    - cell references 103
    - circular references 109
    - custom functions 105
    - functions 429
    - iterations 109
    - operators 429
    - overview 26, 102
    - recalculating 107
    - relative cell references 103
    - updating 107
  - fpCombo control 230
  - freezing columns or rows 177
  - functions
    - formulas 429
    - VBX control 35
  - gray area
    - fpSpread control 287
    - fpSpreadPreview control 41, 158
- spreadsheets (*continued*)
  - grid lines
    - color 283
    - displaying 283
    - hiding 283
    - printing 60, 142
    - style 283
  - headers
    - binding to databases 275
    - color 185
    - column height 168
    - content 179
    - data binding 275
    - hiding 183
    - letters 179
    - numbers 179
    - overview 179
    - printing 58, 60, 64, 142, 151
    - row width 167
    - specifying 31, 51
    - starting letter or number 182
    - text 179
    - three-dimensional appearance 184, 185
  - height
    - column headers 168
    - rows, based on text 169, 196
    - rows, headers 168
    - rows, preventing resizing 170
    - rows, user interaction 170
  - hiding
    - borders 289
    - columns 175
    - focus rectangle 290, 300
    - grid lines 283
    - headers 183
    - rows 175
    - selections 290
    - text tips 311
  - HTML files, exporting 69, 119, 463, 505
  - importing Excel files 81, 471
  - inserting
    - columns 161
    - rows 161

## Index

### spreadsheets (*continued*)

- integer cells
  - creating 249
  - maximum value 249
  - minimum value 249
  - overview 249
  - spin buttons 251
- iterations in formulas 109
- job name, printing 60, 142
- letters, in headers 179
- list box, spreadsheet operating as 128
- loading
  - binary files 81
  - spreadsheets from a buffer 81
  - tab-delimited files 81
- locked cells
  - color 195
  - marking as locked 192, 194
  - overview 192
  - protecting the spreadsheet 193
  - removing protection 194
  - setting 193
  - unlocking the spreadsheet 194
- margins, printing 60, 142
- mask (PIC) cells 253
- maximum
  - number of columns 172
  - number of rows 172
  - number of virtual rows 132
- maximum value
  - date cells 233
  - float cells 242
- minimum value
  - date cells 233
  - float cells 242
- modes
  - edit 199
  - operation 128
  - row 128
  - virtual 130
- monitoring data changes 128
- moving data 91

### spreadsheets (*continued*)

- moving focus
  - with the Enter key 298
  - with the Tab key 299
- multiple block selection 191
- multiple-selection list boxes 128
- navigation 27, 298
- notation
  - cell 103
  - data 76
- number of rows 132
- number of rows in virtual buffer 134
- numbers, in headers 179
- opening 57, 81
- operation modes 128
- operators, formulas 429
- overflow in cells 198
- overview 3
- owner-drawn cells 252
- page breaks, printing 63, 149
- page count 142
- page order, printing 60, 142
- page orientation, printing 60, 142
- pages, printing 62, 147
- password fields 238
- PIC (mask) cells 253
- picture cells 255
- pointers 296
- popup calendar 237
- position
  - columns 174, 177
  - rows 174, 177
- precedence of elements 32
- previewing 65, 153
- printing
  - abort message 60, 142
  - active view in Spread Designer 58, 65
  - block of cells 62, 147
  - borders 60, 142
  - color 60, 142
  - data only 62, 146
  - footers 64, 151

- spreadsheets (*continued*)
  - printing (*continued*)
    - grid lines 60, 142
    - headers 60, 64, 142, 151
    - job name 60, 142
    - margins 60, 142
    - overview 24, 139
    - page breaks 63, 149
    - page count 142
    - page order 60, 142
    - page orientation 60, 142
    - pages 62, 147
    - portion 62, 146
    - ranges 62, 147
    - three-dimensional appearance 60, 142
  - properties, resetting to default settings 127
  - ranges, printing 62, 147
  - read only 128
  - reference style 103
  - relative cell references 103
  - resetting to default settings 127
  - resizing columns or rows 170, 276
  - retrieving data 110
  - returning data 110
  - row headers 179
  - row mode 128
  - rows
    - adding 161
    - background color 205
    - buffers, virtual mode 132, 134
    - color of odd and even 186
    - deleting 164
    - displaying 173–177
    - freezing 177
    - headers 167, 179, 184
    - height 168, 169, 170, 196, 285
    - hiding 175
    - inserting 161
    - maximum number of 132, 172
    - number of 172
    - number of displayed 173
    - odd and even background color 186
    - operation mode 128
- spreadsheets (*continued*)
  - rows (*continued*)
    - overview 23
    - page breaks, printing 63, 149
    - position 174, 177
    - precedence 32
    - resizing 170
    - scrolling 177
    - selection, restricting 126
    - setting properties for 31, 51
    - specifying 31, 51
    - topmost displayed 174
    - virtual mode 132, 134
  - saving
    - files 115
    - Spread Designer changes 68
  - scroll bars
    - displaying 293
    - displaying only when needed 294
    - scroll box range 294
    - specifying when to display 293
    - synchronizing with columns and rows 295
    - virtual mode 134, 136
  - selecting
    - cells 126
    - multiple blocks 191
  - selection
    - discontiguous blocks 191
    - extended 128
    - hiding 290
    - multiple 128
    - multiple blocks 191
    - restricting 126
    - rows only 128
    - single 128
  - separator character
    - float cells 244, 247
    - time cells 265
  - setting properties for 31, 51
  - shortcut keys 27, 301
  - single-selection list boxes 128
  - single-selection operation mode 128
  - SmartPrint 141

## Index

- spreadsheets (*continued*)
  - sorting data 100
  - spin buttons
    - integer cells 251
    - time cells 264
  - static text cells
    - aligning text 258
    - creating 258
    - overview 258
    - shadow color 258
    - shadow text 258
    - text 258
    - three-dimensional appearance 261
  - swapping data 97
  - tab-delimited files
    - loading 81
    - saving 115
  - text
    - color 205
    - column width 169, 196
    - overflow 198
    - row height 169, 196
  - text tips 306
  - three-dimensional appearance
    - button cells 216
    - headers 184, 185
    - printing 60, 142
    - spreadsheets 288
    - static text cells 261
  - time cells
    - 24-hour format 265
    - creating 264
    - formatting 265
    - maximum value 264
    - minimum value 264
    - overview 263
    - seconds, displaying 265
    - separator 265
    - spin buttons 264
  - unformatted data
    - adding 75, 79
    - overview 30
    - returning 112
- spreadsheets (*continued*)
  - unit of measurement
    - column width 285
    - row height 285
  - user interaction
    - column width 170
    - overview 29
    - preventing resizing of columns and rows 170
    - resizing columns and rows 170
    - row height 170
  - vertical scroll bar, virtual mode 134, 136
  - views in Spread Designer 58
  - virtual mode
    - binding to databases 282
    - maximum number of rows 132
    - number of rows in buffer 134
    - overview 26, 130
    - scroll bars 134, 136
    - turning on 131
    - width, columns 167, 169, 170, 196, 285
    - year 2000 handling 233, 235
- SpvGetAllowUserZoom function 155
- SpvGetGrayAreaColor function 158
- SpvGetGrayAreaMarginH function 158
- SpvGetGrayAreaMarginType function 158
- SpvGetGrayAreaMarginV function 158
- SpvGethWndSpread function 153
- SpvGetPageBorderColor function 158
- SpvGetPageBorderWidth function 158
- SpvGetPageGutterH function 159
- SpvGetPageGutterV function 159
- SpvGetPageMultiCntH function 154
- SpvGetPageMultiCntV function 154
- SpvGetPageShadowColor function 157
- SpvGetPageShadowWidth function 157
- SpvGetPageViewPercentage function 154
- SpvGetPageViewType function 154
- SpvGetScrollBarH function 156
- SpvGetScrollBarV function 156
- SpvGetScrollIncH function 156
- SpvGetScrollIncV function 156
- SpvGetZoomState function 155

- SpvSetAllowUserZoom function 155
- SpvSetGrayAreaColor function 158
- SpvSetGrayAreaMarginH function 158
- SpvSetGrayAreaMarginType function 158
- SpvSetGrayAreaMarginV function 158
- SpvSetHWndSpread function 153
- SpvSetPageBorderColor function 158
- SpvSetPageBorderWidth function 158
- SpvSetPageGutterH function 159
- SpvSetPageGutterV function 159
- SpvSetPageMultiCntH function 154
- SpvSetPageMultiCntV function 154
- SpvSetPageShadowColor function 157
- SpvSetPageShadowWidth function 157
- SpvSetPageViewPercentage function 154
- SpvSetPageViewType function 154
- SpvSetScrollBarH function 156
- SpvSetScrollBarV function 156
- SpvSetScrollIncH function 156
- SpvSetScrollIncV function 156
- SpvSetZoomState function 155
- SQRT function 457
- SQUARE function 458
- square function 458
- square root function 457
- SS\_DATE structure 233
- SS\_DATEFORMAT structure 235
- SS\_FLOATFORMAT structure 247
- SS\_PRINTFORMAT structure
  - print options 142
  - print portion of spreadsheet 146
  - range 147
  - SmartPrint 141
  - virtual mode 130
- SS\_SORT structure 100
- SS\_SORTKEY structure 100
- SS\_TEXTTIPFETCH structure 307, 309
- SS\_VQUERYDATA structure 134
- SSAddCustomFunction function 105
- SSAddCustomFunctionExt function 105
- SSAddMultiSelBlocks function 191
- SSB\_ALLOWCELLOVERFLOW boolean 198
- SSB\_ALLOWDRAGDROP boolean 86, 93, 291
- SSB\_ALLOWMULTIBLOCKS boolean 191
- SSB\_ALLOWUSERFORMULAS boolean 107
- SSB\_ARROWSEXITEDITMODE boolean 301
- SSB\_AUTOCALC boolean 107
- SSB\_AUTOCLIPBOARD boolean 301
- SSB\_AUTOSIZE boolean 173
- SSB\_EDITMODEPERMANENT boolean 200, 300
- SSB\_EDITMODEREPLACE boolean 201
- SSB\_FORMULASYNC boolean 107
- SSB\_HORZSCROLLBAR boolean 293
- SSB\_HSCROLLBARTRACK boolean 295
- SSB\_MOVEACTIVEONFOCUS boolean 300
- SSB\_NOBEEP boolean 297
- SSB\_NOBORDER boolean 289
- SSB\_PROCESSTAB boolean 299
- SSB\_PROTECT boolean 193, 194
- SSB\_REDRAW boolean 286
- SSB\_RETAINSELBLOCK boolean 290
- SSB\_SCROLLBAREXTMODE boolean 294
- SSB\_SCROLLBARMAXALIGN boolean 177
- SSB\_SCROLLBARSHOWMAX boolean 294
- SSB\_SHOWCOLHEADERS boolean 183
- SSB\_SHOWROWHEADERS boolean 183
- SSB\_VERTSCROLLBAR boolean 136, 293
- SSB\_VIRTUALMODE boolean 131
- SSB\_VSCROLLBARTRACK boolean 295
- SSClear function 98
- SSClearData function 98
- SSClearDataRange function 98
- SSClearRange function 98
- SSClipboardCopy function 88
- SSClipboardCut function 94
- SSClipboardPaste function 88, 94
- SSClipIn function 75, 76, 77
- SSClipOut function 110, 111
- SSClipValueIn function 75, 76, 79
- SSClipValueOut function 110, 112
- SSCopyRange function 86
- SSDelCol function 164
- SSDelColRange function 164
- SSDelRow function 164
- SSDelRowRange function 164
- SSExportRangeToHTML function 119

## Index

- SSExportToExcel function 117
- SSExportToHTML function 119
- SSGetActionKey function 302
- SSGetActiveCell function 189
- SSGetAppearance function 288
- SSGetArray function 126
- SSGetAutoSizeVisible function 173
- SSGetBackColorStyle function 283
- SSGetBorder function 202
- SSGetButtonDrawMode function 305
- SSGetCellType function 209
- SSGetClipboardOptions function 88, 94
- SSGetColHeaderDisplay function 179
- SSGetColor function 205
- SSGetColPageBreak function 149
- SSGetColWidth function 167, 169, 196
- SSGetColWidthInPixels function 167, 169, 196
- SSGetCursor function 296
- SSGetData function 110, 111, 180
- SSGetDefFloatFormat function 247
- SSGetEditEnterAction function 298
- SSGetEditMode function 200
- SSGetExcelSheetList function 81
- SSGetFloat function 110
- SSGetFont function 207
- SSGetFormula function 102
- SSGetFreeze function 177
- SSGetGrayAreaColor function 287
- SSGetGridColor function 283
- SSGetGridType function 283
- SSGetInteger function 110
- SSGetIteration function 109
- SSGetLock function 192, 194
- SSGetLockColor function 195
- SSGetMaxCols function 172
- SSGetMaxRows function 132, 172
- SSGetMaxTextCellSize function 196, 197
- SSGetMaxTextColWidth function 169, 196
- SSGetMaxTextRowHeight function 169, 197
- SSGetNextPageBreakCol function 149
- SSGetNextPageBreakRow function 149
- SSGetOddEvenRowColor function 186
- SSGetOperationMode function 128
- SSGetPrintOptions function 142, 147
- SSGetPrintPageCount function 142
- SSGetRefStyle function 103
- SSGetRowHeaderDisplay function 179
- SSGetRowHeight function 168, 169, 197
- SSGetRowHeightInPixels function 168, 169, 197
- SSGetRowPageBreak function 149
- SSGetSelBlockOptions function 126
- SSGetSelColor function 207
- SSGetShadowColor function 185, 258, 261
- SSGetStartingNumbers function 182
- SSGetTextTip function 307
- SSGetTwoDigitYearMax function 233
- SSGetUnitType function 285
- SSGetUserResize function 170
- SSGetUserResizeCol function 170
- SSGetUserResizeRow function 170
- SSGetValue function 110, 112
- SSImportExcelSheet function 81
- SSInsCol function 161
- SSInsColRange function 161
- SSInsRow function 161
- SSInsRowRange function 161
- SSIsExcelFile function 81
- SSLoadFromBuffer function 81
- SSLoadFromFile function 81
- SSLoadTabFile function 81
- SSM\_DRAGDROP message 291
- SSM\_EDITMODEOFF message 199
- SSM\_EDITMODEON message 199
- SSM\_TEXTTIPFETCH message 309
- SSM\_VQUERYDATA message 134
- SSMoveRange function 92
- SSPrint function
  - headers and footers 151
  - portion of spreadsheet with data 146
  - print options 142
  - print range 147
  - sending to the printer 141
- SSReCalcCell function 107
- SSRemoveCustomFunction function 105
- SSReset function 127
- SSSaveTabFile function 115

- SSSaveToBuffer function 115
- SSSaveToFile function 115
- SSSetActionKey function 302
- SSSetActiveCell function 189
- SSSetAppearance function 288
- SSSetArray function 75, 77
- SSSetAutoSizeVisible function 173
- SSSetBackColorStyle function 283
- SSSetBool function
  - active cell 300, 301
  - allowing cells to overflow 198
  - automatic formula calculation 107
  - automatic formula updates 107
  - autosizing 173
  - beeps 297
  - blocks of cells 191
  - borders 289
  - cells
    - locking 193
    - overflow 198
    - unlocking 194
  - Clipboard shortcut keys 301
  - column headers 183
  - drag-and-drop operations 86, 93, 291
  - edit mode 200, 201, 300, 301
  - flickering 286
  - focus movement 299, 300, 301
  - focus rectangle 300
  - headers 183
  - highlighting 290
  - horizontal scroll bar 293
  - locking the spreadsheet 193
  - multiple blocks 191
  - redrawing the control 286
  - replacing existing text 201
  - row headers 183
  - scroll bars 136, 293, 294
  - scrolling 177
  - selection 191, 201, 290
  - shortcut keys 301
  - SSB\_ALLOWCELLOVERFLOW boolean 198
  - SSB\_ALLOWDRAGDROP boolean 86, 93, 291
- SSSetBool function (*continued*)
  - SSB\_ALLOWMULTIBLOCKS boolean 191
  - SSB\_ALLOWUSERFORMULAS boolean 107
  - SSB\_ARROWSEXITEDITMODE boolean 301
  - SSB\_AUTOCALC boolean 107
  - SSB\_AUTOCLIPBOARD boolean 301
  - SSB\_AUTOSIZE boolean 173
  - SSB\_EDITMODEPERMANENT boolean 200, 300
  - SSB\_EDITMODEREPLACE boolean 201
  - SSB\_FORMULASYNC boolean 107
  - SSB\_HORZSCROLLBAR boolean 293
  - SSB\_HSCROLLBARTRACK boolean 295
  - SSB\_MOVEACTIVEONFOCUS boolean 300
  - SSB\_NOBEEP boolean 297
  - SSB\_NOBORDER boolean 289
  - SSB\_PROCESSTAB boolean 299
  - SSB\_PROTECT boolean 193, 194
  - SSB\_REDRAW boolean 286
  - SSB\_RETAINSELBLOCK boolean 290
  - SSB\_SCROLLBAREXTMODE boolean 294
  - SSB\_SCROLLBARMAXALIGN boolean 177
  - SSB\_SCROLLBARSHOWMAX boolean 294
  - SSB\_SHOWCOLHEADERS boolean 183
  - SSB\_SHOWROWHEADERS boolean 183
  - SSB\_VERTSCROLLBAR boolean 136, 293
  - SSB\_VIRTUALMODE boolean 131
  - SSB\_VSCROLLBARTRACK boolean 295
  - synchronizing with columns and rows 295
  - Tab key 299
  - turning on edit mode 200
  - unlocking the spreadsheet 194
  - user formulas 107
  - vertical scroll bar 293
  - virtual mode 131
- SSSetBorder function 202
- SSSetBorderRange function 202
- SSSetButtonDrawMode function 305
- SSSetCalText function 237
- SSSetCellType function
  - button cells 210–218
  - check box cells 220, 221, 222

## Index

- SSSetCellType function (*continued*)
  - combo box cells 225, 228, 230
  - date cells 233, 235
  - edit cells 238, 240
  - float cells 242, 244
  - integer cells 249, 251
  - owner-drawn cells 252
  - PIC (mask) cells 253
  - picture cells 255
  - static text cells 184, 258, 261
  - time cells 264, 265
- SSSetCellTypeRange function
  - button cells 210–218
  - check box cells 220, 221, 222
  - combo box cells 225, 228, 230
  - date cells 233, 235
  - edit cells 238, 240
  - float cells 242, 244
  - integer cells 249, 251
  - owner-drawn cells 252
  - PIC (mask) cells 253
  - picture cells 255
  - static text cells 184, 258, 261
  - time cells 264, 265
- SSSetClipboardOptions function 88, 94
- SSSetColHeaderDisplay function 179
- SSSetColor function 205
- SSSetColorRange function 205
- SSSetColPageBreak function 149
- SSSetColWidth function 167, 169, 196
- SSSetColWidthInPixels function 167, 169, 196
- SSSetCursor function 296
- SSSetData function 75, 76, 77, 180
- SSSetDataRange function 75, 76, 77, 180
- SSSetDefFloatFormat function 247
- SSSetEditEnterAction function 298
- SSSetEditMode function 200
- SSSetFloat function 75, 76, 79
- SSSetFloatRange function 76
- SSSetFont function 207
- SSSetFontRange function 207
- SSSetFormula function 102
- SSSetFormulaRange function 102
- SSSetFreeze function 177
- SSSetGrayAreaColor function 287
- SSSetGridColor function 283
- SSSetGridType function 283
- SSSetInteger function 75, 76, 79
- SSSetIntegerRange function 76
- SSSetIteration function 109
- SSSetLock function 192, 194
- SSSetLockColor function 195
- SSSetLockRange function 192, 194
- SSSetMaxCols function 172
- SSSetMaxRows function 132, 172
- SSSetOddEvenRowColor function 186
- SSSetOperationMode function
  - copying data using drag-and-drop 86
  - dragging and dropping cells 291
  - drawing button cells 305
  - moving data using drag-and-drop 93
  - specifying the operation mode 128
- SSSetPrintOptions function 142, 147
- SSSetRefStyle function 103
- SSSetRowHeaderDisplay function 179
- SSSetRowHeight function 168, 169, 197
- SSSetRowHeightInPixels function 168, 169, 197
- SSSetRowPageBreak function 149
- SSSetSelBlockOptions function 126
- SSSetSelColor function 207
- SSSetShadowColor function 185, 258, 261
- SSSetStartingNumbers function 182
- SSSetTextTip function 307
- SSSetTwoDigitYearMax function 233
- SSSetTypeButton function 210–218
- SSSetTypeCheckBox function 220, 221, 222
- SSSetTypeComboBox function 225, 230
- SSSetTypeComboBoxEx function 225, 228, 230
- SSSetTypeDate function 233, 235
- SSSetTypeEdit function 238, 240
- SSSetTypeFloat function 242, 244
- SSSetTypeFloatExt function 242, 244
- SSSetTypeInteger function 249
- SSSetTypeIntegerExt function 249, 251
- SSSetTypeOwnerDraw function 252
- SSSetTypePic function 253

- SSSetTypePicture function 255
  - SSSetTypePictureHandle function 255
  - SSSetTypeStaticText function 184, 258, 261
  - SSSetTypeTime function 264, 265
  - SSSetUnitType function 285
  - SSSetUserResize function 170
  - SSSetUserResizeCol function 170
  - SSSetUserResizeRow function 170
  - SSSetValue function 75, 76, 79, 258
  - SSSetValueRange function 75, 76, 79
  - SSShowCell function 174
  - SSShowCol function 119, 175
  - SSShowRow function 119, 175
  - SSSort function 100
  - SSSortEx function 100
  - SSSwapRange function 97
  - SSVGetBufferSize function 134
  - SSVGetMax function 132, 134
  - SSVGetStyle function 134
  - SSVScrollGetSpecial function 136
  - SSVScrollSetSpecial function 136
  - SSVSetBufferSize function 134
  - SSVSetMax function 132, 134
  - SSVSetStyle function 134
  - standard deviation for population function 459
  - standard deviation function 458
  - StartingColNumber property 182
  - StartingRowNumber property 182
  - static text cells
    - aligning text 258
    - creating 258
    - overview 258
    - shadow color 258
    - shadow text 258
    - text 258
    - three-dimensional appearance 261
  - statistical functions
    - AVERAGE 432
    - COMBIN 433
    - MEDIAN 445
    - MODE 446
    - PERMUT 448
    - RANK 453
  - statistical functions (*continued*)
    - STDEV 458
    - STDEVP 459
    - VAR 463
    - VARP 464
  - STDEV function 458
  - STDEVP function 459
  - straight-line depreciation 457
  - string data, DLL control 469
  - structures, fpSpread control 37
  - styles
    - fpSpread control 37
    - grid lines 283
  - subtraction operator 429
  - SUM function 460
  - sum of squares function 460
  - summation function 460
  - sum-of-years' digits depreciation 460
  - SUMSQ function 460
  - SUPERBTNCOLOR structure 215, 216, 218
  - support
    - contacting FarPoint Technologies, Inc. xvii
    - online help xvii
  - supported properties
    - Excel 81
    - Excel import and export 471
    - HTML 81
    - HTML export 505
  - swapping data 97
  - SYD function 460
  - symbols, documentation xvi
  - synchronizing the scroll box 295
  - system requirements xv
- ## T
- Tab key, moving focus 299
  - tab-delimited files
    - loading 81
    - saving 115
  - TAN function 461
  - tangent function 461
  - TANH function 461
  - technical support xvii, xix

## Index

- text
  - alignment
    - button cells 211
    - check box cells 221
    - combo box cells 225
    - date cells 233
    - edit cells 238
    - float cells 242
    - integer cells 249
    - PIC (mask) cells 253
    - picture cells 255
    - static text cells 258
    - time cells 264
  - button cells 211
  - check box cells 221
  - color
    - button cells 218
    - cells 205
    - headers 185
    - locked cells 195
    - odd and even rows 186
    - selected cells 207
    - spreadsheet 205
    - text tips 311
  - column width 169, 196
  - edit cells 238
  - headers 179
  - height 169, 197
  - overflow 198
  - replacing in edit mode 201
  - row height 169, 196
- Text property
  - adding data 75, 76, 77
  - headers 180, 275
  - returning data 110, 111
  - static text cells 258
- text tips
  - colors 311
  - customizing for cells and headers 309
  - delay 307
  - displaying 307
  - focus 307
  - hiding 311
  - text tips (*continued*)
    - location 307
    - multiple lines of text 309
    - overview 306
    - suppressing 311
    - turning on 307
    - wait 307
    - width 309
  - TextTip property 307
  - TextTipDelay property 307
  - TextTipFetch event 307, 309
  - three-dimensional appearance
    - button cells 216
    - headers 184, 185
    - printing 60, 142
    - spreadsheets 288
    - static text cells 261
  - time cells
    - 24-hour format 265
    - creating 264
    - formatting 265
    - maximum value 264
    - minimum value 264
    - overview 263
    - seconds, displaying 265
    - separator 265
    - spin buttons 264
  - TopRow property 174
  - trigonometric functions
    - ACOS 430
    - ASIN 431
    - ATAN 431
    - ATAN2 431
    - COS 433
    - COSH 434
    - SIN 456
    - SINH 456
    - TAN 461
    - TANH 461
  - TRUE function 462
  - true logical function 462
  - TRUNCATE function 462
  - truncation function 462

- TwoDigitYearMax property 233
- TypeButtonAlign property 211, 213
- TypeButtonBorderColor property 215, 216
- TypeButtonColor property 218
- TypeButtonDarkColor property 216
- TypeButtonLightColor property 216
- TypeButtonPicture property 213
- TypeButtonPictureDown property 213
- TypeButtonShadowSize property 216
- TypeButtonText property 211
- TypeButtonTextColor property 218
- TypeButtonType property 210
- TypeCheckCenter property 220
- TypeCheckPicture property 222
- TypeCheckText property 221
- TypeCheckTextAlign property 221, 222
- TypeCheckType property 220
- TypeComboBoxEditable property 225
- TypeComboBoxHwnd property 230
- TypeComboBoxList property 225
- TypeComboBoxMaxDrop property 225
- TypeComboBoxWidth property 228
- TypeDateCentury property 235
- TypeDateFormat property 235
- TypeDateMax property 233
- TypeDateMin property 233
- TypeDateSeparator property 235
- TypeEditCharCase property 238
- TypeEditCharSet property 240
- TypeEditMultiLine property 240
- TypeEditPassword property 238
- TypeFloatCurrencyChar property 244, 247
- TypeFloatDecimalChar property 244, 247
- TypeFloatDecimalPlaces property 242
- TypeFloatMax property 242
- TypeFloatMin property 242
- TypeFloatMoney property 244, 247
- TypeFloatSeparator property 244, 247
- TypeFloatSepChar property 244, 247
- TypeHAlign property
  - cell overflow 198
  - check box cells 220
  - combo box cells 225
- TypeHAlign property (*continued*)
  - date cells 233
  - edit cells 238
  - float cells 242
  - integer cells 249
  - mask (PIC) cells 253
  - PIC (mask) cells 253
  - picture cells 255
  - static text cells 258
  - time cells 264
- TypeIntegerMax property 249
- TypeIntegerMin property 249
- TypeIntegerSpinInc property 251
- TypeIntegerSpinWrap property 251
- TypeMaxEditLen property 225, 240, 270, 271, 276
- TypeOwnerDrawStyle property 252
- TypePicDefaultText property 253
- TypePicMask property 253
- TypePictCenter property 255
- TypePictMaintainScale property 255
- TypePictPicture property 255
- TypePictStretch property 255
- TypeSpin property 235, 251, 264
- TypeTextPrefix property 258
- TypeTextShadow property 184, 261
- TypeTextShadowIn property 184, 261
- TypeTextWordWrap property 258
- TypeTime24Hour property 265
- TypeTimeMax property 264
- TypeTimeMin property 264
- TypeTimeSeconds property 265
- TypeTimeSeparator property 265
- TypeVAlign property
  - check box cells 220
  - combo box cells 225
  - date cells 233
  - edit cells 238
  - float cells 242
  - integer cells 249
  - mask (PIC) cells 253
  - PIC (mask) cells 253
  - picture cells 255

## Index

TypeVAlign property (*continued*)

static text cells 258

time cells 264

## U

unformatted data

adding 75, 79

overview 30

returning 112

unit of measurement

column width 285

row height 285

UnitType property 285

unloading DLL libraries 467

URL function 463

user interaction

column width 170

overview 29

preventing resizing of columns and rows 170

resizing columns and rows 170

row height 170

UserResize property 170

UserResizeCol property 170

UserResizeRow property 170

## V

value

absolute 430

negative 446

Value property

adding data 75, 76, 79

returning data 110, 112

static text cell 258

VAR function 463

variance function 463

variance of population function 464

VARP function 464

versions, changes from previous 8

vertical scroll bars, virtual mode 134, 136

virtual mode

binding to databases 282

number of rows, specifying

in a buffer 134

maximum 132

overview 26, 130

rows, specifying

maximum number 132

number in the buffer 134

scroll bar, customizing 134, 136

turning on 131

VirtualMaxRows property 132, 134

VirtualMode property 131

VirtualOverlap property 134

VirtualRows property 134

VirtualScrollBuffer property 134

Visible property 230

VisibleCols property 173

VisibleRows property 173

VScrollSpecial property 136

VScrollSpecialType property 136

## W

width

columns

based on text 169, 196

preventing resizing 170

setting 167

unit of measurement 285

user interaction 170

row headers 167

wild card operator 429

Windows appearance 288

World Wide Web site, FarPoint Technologies xix

## X

XROOT function 465

## Y

year 2000 handling 233, 235

## Z

zooming, preview page 40, 155

ZoomState property 155

