

F9TM

“more than just financial reporting”



Disclaimer

Synex Systems Corp makes no representation or warranty with respect to the contents or use of this manual, and specifically disclaims any express or implied warranty or merchantability or fitness for any particular purpose.

Further, Synex Systems Corp reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Synex Systems Corp, makes no representation or warranty with respect to the Product, and specifically disclaims any express or implied warranty of merchantability or fitness for any particular purpose. Further, Synex Systems Corp reserves the right to make changes to any and all parts of the Product, at any time, without obligation to notify any person or entity of any changes.

Assignment

The User may not assign this Agreement, or its rights and obligations under this Agreement, without the prior written consent of the Company.

Entire Agreement

This arrangement represents the entire Agreement between the Company and the User with respect to the Product. No alteration or modification of this Agreement will be valid unless made in writing and signed by both parties.

Exclusions of Warranty

The Company does not make any express or implied warranties, including but not limited to the implied warranties of merchantability and fitness for a particular purpose, and the Company shall not be liable for consequential damages or loss of any kind whatsoever.

Trademarks

F9 and The Financial Reporter are trademarks of Synex Systems Corp.

Excel and Windows are registered trademarks of Microsoft Corporation.

Lotus 1–2–3 is a registered trademark of Lotus Development Corporation.

Quattro Pro is a registered trademark of Corel Corporation, Novell and/or Borland.

Acrobat is a registered trademark of Adobe Corporation.

Other brand and product names are trademarks of their respective holders.

This product is not manufactured, approved, or supported by Microsoft, Lotus Development Corporation, Corel Corporation, Novell, or Borland except as otherwise noted.

Publication Date Feb 1, 1999

Contents

1. Introduction	11
F9 Version 4	11
What I Should Read?	12
Where can I get help?	13
2. Installation and Setup	15
Installation From The F9 CD	15
Getting Up and Running	16
3. F9 — An Overview	19
F9 - The Concept	20
Building a Profit and Loss Report By Hand	21
Building a Profit and Loss Report With the GL Wizard	22
The GL Function	24
One Step at a Time, a Spreadsheet Based Tutorial	24
Creating Great Spreadsheets	29
Design for Simplicity	29
Use Cell References	29
Performance Techniques	30
Example Spreadsheets and Templates	30
E-Mailing or Distributing F9 Spreadsheets	31
Smart Conversions	31
Manual Conversions	32
The F9 Viewer	33
Installing the Viewer	33
Financial Entities	34
4. Reporting Wizards	39
The GL Wizard	40
A GL Wizard Example	41
The Report Wizard	45
The Five Steps to a Report	45
A Report Wizard Example	47
Editing a Report Wizard Report	52

5. F9 Functions 53

Conventions Used in this Section	54
The Absolute Minimum	55
GL(Acct,[Period[,Company[,Year[,Type[,Currency[,Topic]]]])	55
NGL(Acct,[Period[,Company[,Year[,Type[,Currency[,Topic]]]])	55
BSPEC([TSO Mask,]Segment 1[, Segment 2 [, ...Segment N]])	56
SDESC(Account,Section[,Company[,Topic]])	56
CODATA(Data[,Company[,Topic]])	56
Summary of Functions	57
Producing Account Balances	57
GL(Acct,[Period[,Company[,Year[,Type[,Currency[,Topic]]]])	57
NGL(Acct,[Period[,Company[,Year[,Type[,Currency[,Topic]]]])	57
GLTRAN(Acct,FromDate,ToDate[,Company[,Type[,Currency [,Topic]]])	57
Producing Other Account Data	58
DESC(Acct[,Company[,Year[,Type[,Topic]]])	58
SDESC(Account,Section[,Company[,Topic]])	58
Multiple segment Descriptions and SDESC	59
ACCTDATA(Data,Account[,Company[,Topic]])	60
Producing Company Information	60
CODATA(Data[,Company[,Topic]])	60
OD(...) or OTHERDATA(...)	61
Producing Non-GL Data	61
F9DATA(Code, Column [, DataMart [, Year [, Type [, Currency [, Topic]]]])	61
The Write Functions	62
Limitations of Write Functions	62
WGL(Amount, Account, Period, Company, Year , Type [, Currency [,Topic]])	63
WDESC(Description, Account [, Company [,Type [, Year [,Topic]]])	65
Account Specifiers	66
Function Parameters	67
Using Default Parameter Values	68
Accounts Specifiers	69
Using Single Account Numbers	70
Using Multiple Account Specifiers	71
Using Lists of Accounts	71
Using Wild Card Characters	72
Specifying a Range	72
Using Financial Entities	73
Natural and User Defined Entities	74
Multiple Segment Descriptions	75
Financial Entity Restrictions	76
Referencing Cells—The Preferred Method	77
BSPEC([TSO Mask,]Segment 1[, Segment 2 [, ...Segment N]])	78
The Text Substitution Operator	81
An Illuminating BSPEC Case Study	83
Period Specifiers	85
The Period Wizard	85
Periods Quick Start	86
Time Specifiers	87

Duration Specifiers	88
Future Specifiers	88
Value Type Specifiers	88
Balance Time Specifiers	88
Period Lists	89
Budget Specifiers	89
Year Specifiers	89
Additional Keywords	90
Example Period Specifiers	90
Company Specifiers	91
Using Single Company Specifiers	91
Specifying Multiple Companies	91
Year Specifiers	92
Type Specifiers and Budgets	93
Currency Specifiers	93
Specifying the Topic	93

6. Commands and F9 Windows 95

Server Commands	96
The Topic Command	96
The Default Topic	97
Quit Server	98
Help	98
DLL Commands	99
The Setup Window	99
The Chart Window	102
The Drill Down Window	104
Drilling by Account Segment	105
Drilling to Transactions	108
Transaction Fields	110
The Period Wizard and Transactions	110
The Budget Window	112
The Lists Window	113
Creating Lists of Segment Values	114
List Options	116
Financial Entities Editor	116
The Password Window	118
The Period Wizard Window	119
Getting Help	120
Client Commands	121
Zero Suppress	121
Column-wise zero suppress	122
The Zero Suppress Macro	123
Zero Suppression and 1-2-3 R5	123
Zero Suppress	123
Undo Zero Suppress	123
To E-Mail/Viewer	124

GL Paste	124
Options	125
The Analyze Command	127
Analyze Account Access	127
Analyze Re-Calc Time	130

7. Advanced Features 131

Changing the Way Cells are Recalculated	131
How F9 Works	132
Client Add-ins	133
The Server	134
Accounting System DLLs	134
DataPreLoad	134
Automating F9 Functionality	137
Silent Running	137
Using the Lists Facility with Silent Running	142
Production Reporting with F9	144
Creating F9 Dynamic Data Exchange Requests	145

8. Editing Support Files 147

Financial Entities	148
Editing Column Names	150
F9DATA(Code, Column [, DataMart [, Year [, Type [, Currency [, Topic]]]])	150
NGL(Code, Column [, DataMart [, Year [, Type [, Currency [, Topic]]]])	150
GL(Code, Column [, DataMart [, Year [, Type [, Currency [, Topic]]]])	150
The Tree file	152
The MAP File	153
Editing Currency Conversion Tables	156
Editing Consolidation Conversion Rates	156
Editing Transaction Conversion Rates	157

9. Budget Writeback 159

Budget Writeback with the Clipboard	159
Budget Writeback with WGL and WDESC	161
Common Problems with Budget Write Back	163

A. F9 Tips and Tricks 165

Consolidations with Differing Charts of Accounts	165
Performance Enhancing Formulae	166
Control Totals	167
Using Cell References	168
Named Ranges	168

B. Frequently Asked Questions **171**

C. Using F9 with Microsoft Excel **181**

Installing the Add-in	181
Loading F9 Automatically	181
Customizing the F9 Tool Icons	182
Excel 97 and “Balloon Help”	182
The Add-in Menu	182
F9 Functions	184
AutoSum	185
Reading Your Lotus 1-2-3 F9 Spreadsheets	185
Writing Macros in Excel	185
Manual Recalculation in Excel 5 and ‘95	187
Manual Recalculation in Excel ‘97	187
Range Recalculations	188
Other Tips	188
Excel FAQ	189

D. Using F9 with Lotus 1-2-3 **193**

Installing the Add-in	193
Installing and Customizing the Tool Icons	193
Loading F9 Automatically	194
The Add-in Menu	195
Invoking F9 Commands	197
F9 Functions	197
All Lotus Users	198
1-2-3 Release 4 Users	198
Writing Macros in Lotus 1-2-3 (v.4 or v.5)	199
Setting the DDE Time Out	200
Lotus 1-2-3 FAQ	200

E. Using F9 with Quattro Pro Vers 5 to 7 **203**

Installing F9 for Quattro Pro	203
Report Wizard and Analyze	204
Recalculation Modes	205
Speed Bar Icons	205
The Array Function (pre Version 3.1)	205
Error Messages (post Version 3.1)	206
Reading Your Lotus 1-2-3 F9 Spreadsheets	206
Writing Macros in Quattro Pro for Windows	206
Setting the DDE Time Out	207
Quattro Pro FAQ	207

F. F9 Templates	209
<hr/>	
Accounting System Specific Templates	209
Generic Templates	209
Supplied Templates	210
Making the Templates Work With Your Data	211
Changing the BSPEC Function	212
The Button Spreadsheet	213
G. Troubleshooting Btrieve	215
<hr/>	
F9 and Btrieve	215
Unknown Btrieve Error	215
Btrieve error 4 or 11 or 12	216
Btrieve error 20 and a DOS accounting system	216
Btrieve Error 20	218
Btrieve error 88	218
H. Glossary	219
<hr/>	
I. Sample Reports	229
<hr/>	
Statement of Changes in Financial Position	230
Budget Variance Analysis	232
Proforma Profit and Loss Statement	234
Multi Currency Reporting	236
Weekly Income Statement	238
Running Year P&L	240
Index	243
<hr/>	

Chapter

1

1. Introduction

F9 brings your accounting data into your spreadsheet for analysis and reporting. Instant access to your financial data is available to users of Excel, Quattro Pro and Lotus 1-2-3 for Windows.

This manual is “generic”. It is not specific to any one accounting system DLL, spreadsheet, or operating system. Most of the examples in this manual use a four segment account code belonging to a hypothetical accounting system. When using F9 with *your accounting system*, you will need to use account codes that are appropriate for that system. For more information on using F9 with *your particular spreadsheet*, see the appendices. We have chapters on Microsoft Excel, Lotus 1-2-3, and Quattro, as well as ‘Frequently Asked Questions.’

F9 Version 4

Version 4 has several key additions in functionality that make F9 an even more powerful business solution:

- A full **32-bit** implementation which updates a variety of controls and functionality.
- Enhanced **Drilldown** capabilities allowing for display of all fields with sizeable, sortable columns among other improvements.
- **Lotus 1-2-3 for SmartSuite ‘97** add-in with SmartIcons is now available.
- A new **Account Enquiry/GL Wizard** for a quick balance query from any account in the GL *and* produce a properly formatted GL formula with appropriate cell references ready for copying.

- **Financial entities** work in your terms, not the GL's. Queries can now refer to "Vancouver-Revenue-All_Dept" instead of "A-3*-#".
- A **smarter BSPEC function** that makes the *Text Substitution Operator* (TSO) optional.
- A new **Export to E-mail and Viewer** command to remove F9 formulae from sheets.
- A new **F9 Viewer** option which allows F9 reports to be viewed and to be drilled by users without a full F9 implementation at a lower cost.
- A new **F9 Edit** spreadsheet add-in which simplifies all DataMart editing tasks.
- Improved **Silent Running** for more stable scripted operation and automation in Executive Information Systems (EIS) applications.
- New write-back functions **WGLO** and **WDESCO** functions allow for direct updates from a spreadsheet to your GL to store information such as Budgets or consolidated books. Some capabilities depend on your accounting system.

What I Should Read?

This manual assumes that you are familiar with Excel, Quattro Pro or Lotus 1-2-3 for Windows. If you have never used F9 before, read *F9 - An Overview*, especially the section entitled *F9 - The Concept*. Then read one of *Using F9 with Excel*, *Using F9 with Lotus 1-2-3*, or *Using F9 with Quattro Pro*.

If you have used F9 before, read one of *Using F9 with Excel*, *Using F9 with Lotus 1-2-3*, or *Using F9 with Quattro Pro*. You should also read the section *The BSPEC Function* as this describes an important F9 function.

We have included a complete manual on disk with your product in the form of an Adobe Acrobat document. The file has the extension .pdf, and requires installing the Acrobat viewer. It contains a complete and up-to-date version of the manual. Please note that both the index and the table of contents are hyperlinked to the text of the document.

Finally, browse through the help system associated with your DLL. To do this, double click on the F9 Server icon, select a topic*, then select Help and finally DLL Help.

***Note:**

Most users will not need to “select a topic.” Selecting a topic simply tells F9 which accounting system is the default and is required only if you have more than one F9 accounting system DLL installed.

Where can I get help?

Our home on the electronic frontier is a Web site at www.F9.com. Please visit the site for the latest versions of all our software. Upgrades for your accounting system DLL are available on our web site and the passwords for the DLLs are available from our Technical Support Desk.

Technical Support:

The F9 Technical Support desk can be reached at phone: (604) 688-8271, or by E-mail: synexsys@synex.com.

F9 has some aspects which are particular to the accounting system you are using; these are documented in the help file associated with the DLL — F9 uses a *Dynamic Link Library* (DLL) to access each accounting system. For example, the OD (Other Data) function is always DLL-specific and is documented in the DLL's help.



Chapter

2

2. Installation and Setup

F9 Version 4 can only be installed on 32-bit versions of Windows (Win95 or later, or Windows NT) because the F9 Server is now a 32-bit executable. The F9 Server will not work with the Win32s extensions to Windows 3.x. However, F9 includes add-ins that are compatible with 16-bit spreadsheets.

Installation has two basic steps:

- **Installation From the F9 CD** will guide you through putting the F9 program files onto your computer or network server.
- **Getting Up and Running** will instruct you on how to load the correct add-in into your spreadsheet.

When completed, you will be able to use all the F9 functions to start building reports.

Installation From The F9 CD

The installation program can be started by any one of these methods:

- The F9 Setup window which Autoruns from the CD, or
- Use the “Install new software” icon in the Control Panel, or
- Use Windows Explorer or **Start | Run** to run SETUP.EXE on the F9 CD.

Typically, the F9 components of are installed in their own directory (*e.g.*, C:\F9) with exceptions of:

- the spreadsheet add-in(s) and
- the Adobe Acrobat Reader.

F9 detect the locations of most current spreadsheets and installs the add-ins to be most available to your spreadsheet(s). The Acrobat Reader will suggest a preferred location which you should accept.

Network Windows Consideration: If you have purchased a multiple user license of F9, you may install F9 to a directory on the network. If the spreadsheet software is also on the network, you can install the add-in files to the shared copy which then only needs to be done once. If each user has a local copy of the spreadsheet software, the add-in files must be installed to each machine.

Getting Up and Running

Check the Readme For Your Components

After installing your software, the F9 directory may contain a file named README.TXT. This file contains information about changes made to your software after the documentation was printed. Be sure to read these file(s) before using the software. You can open and print the README.TXT file using Microsoft Notepad or Wordpad.

Installing the Add-in

In most cases, you will be using one of the add-ins provided to access the Server. F9 supplies add-ins that work with Excel (Versions 5 and later), Lotus 1-2-3 for Windows (all versions), and Quattro Pro (Versions 5 to 7).

Network Windows Consideration: After installing the software, *each user* needs to load the F9 Add-in into their spreadsheet as described below.

Excel add-in

- To load the Excel add-in, use the **File | Open** command to load the file F9.XLL.
- After loading F9.XLL for the first time, perform the following one-time-only task: load the spreadsheet called F9BMP.XLS (Excel 5) or F9BMP32.XLS (Excel 95 or later) from the F9 directory and press the button labelled "Press Me." This will update the F9 tool bar button faces.

Excel can be set to auto-load F9 whenever it is started. To enable this option:

- Select **Tools | Add-Ins** from the Excel menu.
- If “F9 - The Financial Reporter” is not shown in the list of add-ins, press the Browse button and use the file dialog to find F9.XLL.
- Click on the checkbox next to “F9 - The Financial Reporter.” The add-in will now load every time Excel loads. If the addin was not loaded, you F9 will confirm that it is now loaded.

Lotus 1-2-3 '97 add-in

- Select **File | Add-ins | Manage Add-ins** from the Lotus 1-2-3 menu. Select the file F9.12A which should be in the Add-in directory under Lotus\123.
- Auto-loading is enabled by setting a check mark in the left margin of the dialog.

Lotus 1-2-3 v.5 add-in

- Select **Tools | Add-ins** from the Lotus 1-2-3 menu. Load the file F9.ADW.
- Alternatively, to load the Lotus 1-2-3 add-in, press Alt+F10 to open the “1-2-3 Classic” menu. Select **Options**, then **Load**. Load the file F9.ADW.

Lotus 1-2-3 can automatically load the add-in each time it is loaded. To enable this feature:

- Press Alt+F10 to open the “1-2-3 Classic” menu. Select the options **Settings, System, Set** to reach the add-in manager. Select **Add** from the menu, and then F9.ADW file from the file list, to make the add-in auto-loading.

Quattro Pro add-in

- Refer to the Appendix called *Using F9 with Quattro Pro* for instructions on enabling the Quattro Pro client add-in.

Tip:

A set of *Frequently Asked Questions* (an FAQ) is included as an Appendix to this manual for issues that are common to all the spreadsheets. Each of the spreadsheets has an Appendix in this manual which addresses version issues and common problems more thoroughly.



Setup Tool button

Tip:

Accessing Your Data

To complete the installation, you need to change some setting within F9 which are appropriate for your site. After you have loaded the add-in into your spreadsheet (as detailed above), the F9 Setup window can be opened from Excel, Lotus 1-2-3, or Quattro Pro by selecting **F9|Setup** from the menu or pressing the **Setup** button on the toolbar.

Generally speaking, it is rarely necessary to load and run F9.EXE manually. When using F9 with a spreadsheet, the spreadsheet client add-in will manage the loading and unloading of F9 automatically.

The primary setting required by each F9 DLL is a path to your accounting data. You may wish to refer to the DLL help system for details on the various options in the Setup window. Usually, all you will have to do is enter the directory where your accounting system is installed and select a default company.



Setup Tool button

In Excel or in Lotus 1-2-3, you can open the Setup window by pressing the Setup tool button or by selecting **F9|Setup** from the spreadsheet menu. In Quattro Pro, you can open the Setup window by selecting the tool button that looks like a disk with F9 written on it.

Once you are satisfied with the Setup options, you should confirm that you are successfully linked. The easiest way to do this is to press the **Profile** button. If you are correctly configured, a window will appear with some basic data about the default company displayed in it.

Once F9 is successfully linked to your data, you may begin creating spreadsheets.

Chapter

3

3. F9 — An Overview

There are a number of basic ways of developing reports using F9:

1. You can start from a blank spreadsheet and build reports “by hand.” This is the most basic and the most powerful way to exploit F9’s features. This technique is described briefly in this section.
2. The GL Wizard can quickly build the first GL function and control block. Just add an account list and you have an F9 spreadsheet with all the power of a “by hand” sheet with less struggle.
3. Alternately, you can use the Report Wizard. The Report Wizard can create many relatively simple reports and can be a good way to get started. It always creates well-formulated GL functions.
4. Or you can load one of the several sample spreadsheets (*e.g.*, Income or Balance) and modify them to match your chart of accounts.
5. Finally, you can use F9 supplied templates. You should be familiar with the GL function before using F9’s templates.

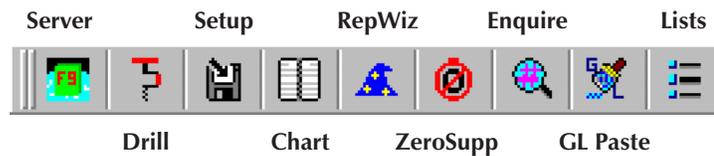
The choice of which method you use depends on how complex or unusual your requirements are. The more complex the requirements, the less likely you will be satisfied with the results of the templates, samples, or report wizard. These tools have been designed with new users and standard reports in mind. Even in the most demanding applications, you will find that using either a pure “by hand” report or one started by the GL Wizard will prove to be fast, easy, and not nearly as labor intensive as they look.

F9 - The Concept

To use F9 effectively, you need to understand what it is and how it brings data to your spreadsheet. The power of F9 built on two concepts:

- F9 is an add-in to your spreadsheet and extends the spreadsheets' functionality by providing several more formulae and commands which retrieve numbers directly from your GL.
- Your spreadsheet allows you to build reports from these numbers. All your spreadsheet skills can be brought to bear on F9 reports.

Once F9 is installed and directed to your general ledger data (this is done in the Setup window), it will be able to access the necessary files to retrieve any requested financial information. You will be unaware of F9's presence except for a menu item (F9) and a few new icons on the tool bar.



Tip:

If you are upgrading from Version 3 of F9 and not all the buttons appeared, check the *Frequently Asked Questions* appendix of this manual for steps to force the toolbar to update.

F9 brings live accounting data to the spreadsheet using F9 specific functions, the most important of which is the GL function. They are the same as any other spreadsheet function, such as SUM, in that they take a number of parameters which are separated by commas and enclosed in parentheses.

To get started with F9 you should become familiar with three functions: GL, BSPEC, and SDESC.

- GL will return amounts from your GL database,
- BSPEC makes account specifiers from cell references, and
- SDESC returns account descriptions from the segments.

If you use F9 to develop reports for several companies, you will also want to become familiar with CODATA which returns company details. These topics are also covered in more details in the topic *The Absolute Minimum* in *Chapter 5, F9 Functions*.

The GL function parameters include the account number, the period, and the company in that order. The syntax is as follows:

```
=GL("1000-100-A", "Month 2", "ABC")
```

Tip:

The syntax written above is correct when placing the parameters explicitly into the formula. However, it was written this way for illustration only. Also, note that this is an Excel formula, both Quattro Pro and Lotus 1-2-3 would use @GL rather than =GL.

You would enter this formula into a cell of the spreadsheet report where you want the balance for account 1000-100-A for Month 2 for company ABC to appear. For every cell in your report where you want a balance to appear, you will use a similar formula.

To build the most intelligent F9 reports you should always use cell references with appropriate use of the absolute reference tag (\$) to provide the various parameters. By doing so you will build in maximum flexibility to your work.

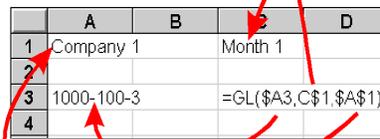
Building a Profit and Loss Report By Hand

Here is the typical process for building a multiple period P&L using the GL function with cell references.

- In Column A, type in the account numbers down the column, one on each row. F9 can also create a list of these codes for you using the **Lists** command which is described later.
- In row 1, type the periods of interest across the row as column headers. F9 allows fiscal periods like "month 1", "month 2", "month 3", *etc.*, or month names like "January", "February", "March", *etc.*
- In the top left corner of the sheet type in the company identifier for your accounting system.
- Build a GL Formula: in the cell that intersects the first account number row and the first column header, type in the GL formula and reference (in order) the cells containing the row header, the column header,

and the company identifier. Finish the formula by setting of absolute and relative cell references. The finished formula should look something like this:

`GL($A3,C$1,A1)`



	A	B	C	D
1	Company 1		Month 1	
2				
3	1000-100-3		=GL(\$A3,C\$1,\$A\$1)	
4				

When you press the F9 key (the recalculation key for all three spreadsheets, and hence the name of our product) a number will appear. The number is retrieved from the accounting system, and each time you press F9 it will be updated with the current value from the general ledger. Once the GL formula is returning the correct value, copy that formula to every other line adjacent to an account number. Press the F9 key and the cells will fill with the appropriate balances. Now spend a little time inserting totals, lines, *etc.*. When you are satisfied with the format, copy that column to the remaining columns. Press F9 again and you have a multiple period P&L. You may now want to spend some time changing fonts, inserting titles, adding color and graphs, or dressing up the report.

That's it! If you are using departments, budgets, or multiple years of data, the process of creating F9 reports is only slightly more involved. To get a better understanding of the various parameter options for the GL formula, refer to the chapter called *The GL Function - One Step at a Time*. For an interactive view, load the spreadsheet called ONESTEP located in the F9 directory. Once you feel you have a clear understanding of the GL formula, load the sheets called INCOME and BALANCE and refer to the section called *Creating Great Spreadsheets* later in this chapter. Each of these will illustrate effective techniques for building your own spreadsheet reports.

Building a Profit and Loss Report With the GL Wizard

The second method of getting a spreadsheet started is with the GL Wizard. The GL Wizard requires just one step: the **GL Paste**.



GL Paste Button

The **GL Paste** is the GL Wizard at work. Pressing the **GL Paste** button (a glue brush) on the F9 toolbar interprets the result of the last Account Enquiry dialog and makes an F9 formula from it. The **GL Paste** should not be confused with **Edit | Paste** which simply drops the clipboard onto your sheet.

The GL Wizard tries to place the GL formula in the same cell your cursor was in *when the GL Paste button was pressed*. F9 will move the cursor when F9 needs to make space for the parameters. After pasting, your spreadsheet will look like:

	A	B	C	D	E	F	G	H
1	Location	All Locations	All Departments					
2								
3								
4	Period	Month						
5	Company	Demo						
6	Year	1997						
7	Type	ACTUAL						
8	Currency	Home						
9								
10								
11	10005			-2010.96				
12								



Enquire Button

An optional step of the GL Wizard is to perform an Account enquiry before the GL Paste. Select **F9|Enquire** from the menu or press the Enquire button (magnifying glass) on the F9 toolbar to open the Account Enquiry dialog. Set each of the controls to a parameter value. The main setting for a good report is the **Create GL function by:** option. This controls whether the headers will be a company, an account number segment, or a period. When the controls are filled, press the Quit button on the dialog and return to your spreadsheet. Pressing **GL Paste** at this point will use the settings just configured to make the GL formula.

Summary

The preceding text described in overview the “core” of using the F9 GL function to create financial reports. While all F9 spreadsheets use the GL function, you can create reports without starting from a blank sheet or typing formulae. You can use either the GL Wizard to start a report, or the Report Wizard to create a broad range of simple reports. The Wizards will create the formulae for you. As the next topic will demonstrate, once a GL formula works, you to copy it to get the remaining GL balances!

F9 also provides templates that can quickly be populated with codes from your chart of accounts using facilities such as the Lists window to reduce the amount of typing involved.

The GL Function

F9 ships with an example spreadsheet called ONESTEP that closely parallels the text of this section. If you are unfamiliar with the GL function, read this section carefully, and then load the ONESTEP sheet and work through it. The examples in the following text are generic and will probably not match your accounting system. The examples in ONESTEP will match your accounting system.

As always, to enter a formula in Lotus 1-2-3 or Quattro Pro, precede the formula with an "@" sign. In Excel, precede it with an equals sign ("="). For the following discussion, neither is used.

One Step at a Time, a Spreadsheet Based Tutorial

The GL function is the heart of F9. Almost all of the important results that F9 produces are produced by the GL function. Although there are a number of ways of creating reports without having to construct GL functions yourself (such as the Report Wizard, templates or Excel Smart Templates), to get the most out of F9, it is important you understand how the GL function works.

The simplest GL function consists of an account specifier only:

```
GL("0-0-1000-0")
```

This function produces the balances for one account for the default company and year (as per the Setup window) for the current period. A slightly more complex function might return the sum of two account balances:

```
GL("0-0-1000-0,0-0-1050-0")
```

Listing accounts this way is inefficient, so F9 provides a number of methods for specifying multiple accounts in an easier way. One way is the asterisk wildcard character:

```
GL("0-0-105*-0")
```

This would return all balances for any account that starts with 105. Similarly, if the first part of the account code is the "location", this function returns all locations for the same accounts:

```
GL("*-0-105*-0")
```

Depending on your GL, this function could be summing hundreds of accounts to return a balance. Ranges of accounts are possible using the Lotus-style range notation:

```
GL( "*-0-1000..1050-0" )
```

Of course, you will not always want to get data for the current period. To get other periods, add a period specifier:

```
GL( "*-0-105*-0" , "Month 12" )
```

The meaning of this parameter should be clear. The “month” specifier is based on the fiscal year. Month names can be used for calendar year reports, too. Other common variations are:

```
This Month
Month 12 Last Year
Month 12 Budget
Running Quarter Month 10
```

This last example is the total transactions for months 8, 9, and 10.

Budgets are also accessible from F9. A function that will return the the total year’s budget would be:

```
Budget Year to Date Month 12
```

Note:

“Budget Year” will return the budget up to the current period. “Budget Last Year” will not work since budgets are not actually related to years (See the Appendix *Frequently Asked Questions*).

GL functions can be more complex still. If there is any chance of you using more than one company, your GL function should include a company parameter:

```
GL( "*-0-105*-0" , "Month 12" , "Lesson" )
```

Here, “Lesson” is the name of the company. To consolidate two companies, just list them out:

```
GL( "*-0-105*-0" , "Month 12" , "Lesson, Sample" )
```

When doing consolidations like this, the two companies should share nearly identical charts of accounts and the same fiscal years.

If they do not share the same fiscal years, use Period Specifiers such as “January.” Current versions of F9 can correctly report calendar months regardless of the fiscal years of the companies in question (prior to version 3 of F9, January always meant “month 1”).

When companies do not share a chart of accounts, you can avoid “no matching account found” errors while consolidating by setting the “Return Zero for Account not Found” option in the Setup window.

Two more parameters can be used: the Year and the Type. The meaning of the Year is obvious; the meaning of Type varies from accounting system to accounting system but is usually “ACTUAL” or a budget type.

```
GL(“*-0-105*-0”, “Month 12 budget”, “Lesson”, “1999”, “Pessimistic”)
```

This function returns data from month 12 of 1999 for the “pessimistic” budget.

This function is getting pretty big! A typical sheet might contain hundreds of such formulae. That’s a lot of typing! And if you wanted to run the same report *versus* a different company, it could take hours to edit each cell. Cell references solve this problem.

Tip:

As we have emphasized, you should always use cell references and should almost never enter parameters directly into your formulae as we have here.

The following is a typical use of cell references. Notice the use of the dollar signs to indicate absolute *versus* relative cell references:

```
GL($A12, C$3, $B$2)
```

Correct use of cell references means that you normally enter only one GL formula and then use the Copy command to duplicate it all over the sheet. Here, of course, cell A12 contains an account specifier, C3 a period specifier and B2 the company specifier. Now to create a report for a different company you only have to edit cell B2 and recalculate the sheet.

Using BSPEC

To maximize the power of the GL Formula, only one stumbling block remains: the Account Specifier. There are two basic problems:

1. The GL Function only allows one parameter for the account, but an account has multiple segments.
2. It is very desirable to build reports which compare different values of an account specifier, *e.g.*, comparing departments or locations.

The BSPEC Function is designed to over come these problems by simply **B**uilding account **S**PECifiers from cell references. Using BSPEC, you should make each part of the account code a separate parameter. This means that

not only is there just one formula for a report, but *each aspect* of that report appears only once. For example, the company name only appears once. Or the current year appears only once.

The solution which F9 brings to these problems is “cell references.” When each segment value is a cell reference in BSPEC, your spreadsheets become significantly more flexible since you can construct a spreadsheet where each value appears only once.

For example, if you want to create a report for one department where the department is the second part of the account code, and you then decide to report on a different department, you have a problem. Every place where the department’s number was used, it would need to be changed *without changing* uses of the same number in other segments.

For example, if the location number 100 and the department number 100 both appear on a report, a search and replace must change one, but not the other. These changes may be needed in hundreds of cells. Each time you want another version for another department, the same changes must be done, yet again.

BSPEC does one thing and does it well; it converts a list into an usable account specifier for F9. The formula:

```
BSPEC("0","0","1000,1050","100,200")
```

simply manipulates these numbers and returns the text:

```
(0)(0)(1000,1050)(100,200)
```

All the F9 functions which take an account specifier recognize this format. And, while it’s possible to make the same text using the spreadsheet’s built in functions, BSPEC is much simpler to use.

The BSPEC function can be inserted directly into your GL function, as well. This allows you to create queries like:

```
GL(BSPEC("0","0","1000,1050","100,200"),"This month")
```

Or, using cell references:

```
GL(BSPEC($B$3,$B$4,$A10,C$4),$B$6)
```

This last formula would allow you to make report using the fourth segment of the account code in columns. This would allow a department store to evaluate a sales report comparing Men’swear, Women’s Wear, and Kitchen appliances, for example. Or a hardware store to compare plumbing, lumber, and masonry products.

Tip:

In Lotus 1-2-3, both the GL and the BSPEC functions must be preceded by an "@" character when used together like this. In Excel, only the GL function required requires an "=" character.

Version 4:

In prior versions of F9, an Account Mask was required. This parameter is now optional and, for most uses, we recommend you leave it out. We cover this parameter in the discussion of BSPEC in *Chapter 5: F9 Functions*.

The Key Concept in using BSPEC is that *no account segment value should appear more than once on a report*. BSPEC can use the value hundreds of times, but it only appears once. This way, any change to any segment value need to only happen once per sheet. No more "Search and replace and hope its right."

Summary

This topic is intended to give you a quick overview of the core F9 features. Each of the functions is discussed later in *Chapter 5: The F9 Functions*. Please read the remainder of this chapter for more information on *Creating Great Spreadsheets*.

Creating Great Spreadsheets

The best F9 spreadsheets use certain “tricks of the trade” that can greatly improve your productivity both now, and in the future. You should become familiar with these techniques and use them in all your spreadsheets.

Design for Simplicity

Plan your reports. In particular, know the horizontal and vertical aspects. Most F9 reports use the natural account on the vertical axis. The Horizontal axis is periods, companies, another account segment (*e.g.*, department, location, *etc.*), years, or “types” such as budgets, actual.

Consistently use control areas. Put all the F9 parameters that aren’t on an axis of the report into a control block in one area of the report. This makes it very easy to reuse the report. For example, when a manager asks for a report by department for his location, you will often need to use a similar report for other locations later. Control areas allow you to build flexibility into your reports.

If you do consolidations or reports for several different companies, then **always** use cell references for the company as well. Just place the company label off in some corner (*e.g.*, the “control area”) of your sheet and then reference the cell absolutely (*e.g.*, \$A\$6).

Always use the **Copy|Paste** functionality of your spreadsheet software to develop additional rows or columns in your sheet. This is especially powerful when used with the relative and absolute cell referencing techniques described below. Remember: reports should only need one GL formula; the rest are simple copies.

Lotus 1-2-3:

If you are using Lotus 1-2-3 for Windows, we recommend you use the Lotus 1-2-3 classic menu Copy command rather than the Copy|Paste menu. Both will work, but the former is faster.

Use Cell References

In Real Estate, the top three contributors to the value of property are determined by “Location, location, location.” When reporting with F9, the top three “power techniques” are “Cell references, cell references, and cell references. Absolutely nothing will give you as much control and power.

Always use cell references rather than explicit values in your formulae. That is, instead of a formula like `GL("1000..1050","Month 2")`, use a formula such as `GL($A5,C$3)`.

Here, the cell A5 would contain the *label* '1000..1050' and the cell C3 would contain the label 'Month 2'. Use the same approach for DESC functions as well. With this approach, you can build a report template by listing the accounts and account ranges you are interested in down the left-hand side of your sheet and the periods across the top of the sheet, and then reference these cells in your functions.

Furthermore, to create really flexible spreadsheets, make use of the BSPEC function to make it easy to take any "slice" out of your GL that you like. Using BSPEC to build your account helps make your reports work harder than you do; you should never need to edit more than a few cells to make almost any report.

Remember to make use of relative and absolute row and column references (absolute references are indicated by the dollar sign). This allows you to take full advantage of copying cell formulae.

Performance Techniques

Avoid asking for more results than you need. If you want to add two departments into a single result, use a single GL function with a list of segment values rather than multiple GL formulas added together. For example, rather than adding:

```
GL(BSPEC("100", "10100", "231"), "This Month") +
GL(BSPEC("100", "10200", "231"), "This Month")
```

you get the same result faster with:

```
GL(BSPEC("100", "10100,10200", "231"), "This Month")
```

Avoid asking for numbers twice. Whenever the values you want are already in the spreadsheet, **always** use the SUM function to generate totals rather than re-specifying the range in a GL function. SUM is **much** more efficient in this circumstance.

Example Spreadsheets and Templates

We have included at least two F9 spreadsheets so you can see how to design and produce powerful F9 reports. The spreadsheets are `income.wk?` (`income.xls`) and `balance.wk?` (`balance.xls`). Load each sheet and review them carefully. These spreadsheets are designed to be able to recalculate

“out of the box” against the sample data for your accounting system. Remember, your spreadsheet should be in manual recalculation mode before you load these samples.

These are relatively simple examples of F9 spreadsheets. We have also included more complex reports and templates that show how you can use F9 to build any type of report you can imagine.

Note how the functions are formulated. Cursor to cells containing GL and DESC functions. These functions make heavy use of cell references. Most of these functions use a special F9 function called BSPEC (for **B**uild **S**PECifier) to construct account specifiers. *These three functions are all that is required to create almost any financial report.*

Use these sheets as a model for your own sheets.

We have also supplied a number of template spreadsheets. See the section called *F9 Templates* for more information.

E-Mailing or Distributing F9 Spreadsheets

F9 spreadsheets are just Excel, Lotus 1-2-3 or Quattro Pro files. The only problem with giving a spreadsheet file that uses F9 functions to someone else is that sheet can no longer be recalculated because there is no longer an F9 add-in around to calculate the functions. Net result: the sheet fills with errors and all the F9 figures are destroyed the moment the sheet is recalculated. In Excel, this often happens when the sheet is loaded. To stop this from happening, you need to convert the formulae to values.

Smart Conversions

The Spreadsheet add-ins for Excel and Lotus 1-2-3 provide a means of automatically converting F9 formulae to values.

Excel Users

The F9 menu now offers an option: **To Viewer/E-Mail**. This option processes all the F9 formulae in the sheet and converts them to values while saving the formula as a comment in the cell. For E-Mail distribution. The process eliminates errors for users who do not have F9 who open the sheet.

Users who have the F9 viewer add-in can drill the formulae in comments to see where the numbers came from. Users who have the regular F9 *will not* be able to drill the values in a converted report.

Note:

The F9 add-in for Excel will only convert formulas which have an F9 function as their first argument. If you include other calculations, they must *follow* an F9 function. Also notice *the entire* formula is reduced to a value, not just the F9 functions. Finally, only one F9 function will be preserved in the cell's comment.

Lotus 1-2-3 Users

Lotus users were introduced to the **Value** option on the F9 menu in Version 3. This option processes all the F9 formulae in the sheet and converts them to values. The **Value** process removes the F9 formula and replaces it with an explicit value. The parameters of the formula are not saved (a limitation within 1-2-3) and not available to an F9Viewer add-in.

Manual Conversions

The other way to prepare a spreadsheet for E-Mail is to use **Paste as Values** to eliminate the formulae. This process has one major difference from the **To Viewer/E-Mail** in that it converts *every* formula in the copied area or worksheet. If you want to preserve formulae that do not require F9, *do not* use **Paste as Values**.

Excel Users

Select the entire sheet or an area, do an **Edit | Copy**, and then an **Edit | Paste Special**. Select **Values** and press OK. Now do a **File | Save As** and save the workbook to a new file. Send this file to your associate.

In Excel, the **Edit** menu has an option to **Move or Copy Sheets** from one workbook to another. This can also be useful in distributing reports.

Lotus 1-2-3 Users

Select the entire sheet or an area, do an **Edit | Copy**, in **Edit | Paste Special**, select **Formulae as Values**. Now do a **File | Save As** and save the workbook to a new file. Send this file to your associate.

Quattro Pro Users

Select the entire sheet or an area, do an **Edit | Copy**, then **Edit | Paste Formulae as Values**. Now do a **File | Save As** and save the workbook to a new file. Send this file to your associate.

The F9 Viewer

The F9 viewer is a less expensive and simplified method for viewing F9 spreadsheets. Once a spreadsheet has been prepared for the viewer [*required by Excel only*], a user can view an F9 report anywhere. If the user has access to the source GL, the Viewer has the capability to drill the original formula just as if they were using a regular add-in.

What the viewer can **not** do is update an F9 report from the source GL. In other words, none of the F9 functions are actually “live” anymore. This add-in is advantageous if you have:

- users that should only have a limited view of the GL, or
- a report that would be portable outside the office, or
- you require reports which reflect balances as of “close of business on” specific dates.

Our clients have expressed an interest in having a small group of employees design and distribute reports, and a wider audience being able to view them with stronger limitations than the full viewer. These reports make excellent *executive information system* (EIS) reports.

Installing the Viewer

The F9 Viewer add-in is installed from the CD using the same process as with the other F9 components, by providing an ID/Password pair in the install dialog. The add-in should be installed on each client machine to your spreadsheet add-in directory, or in the network Add-in directory. There are separate versions of the viewer for Excel and Lotus 1-2-3.

The F9Viewer add-in can be loaded into your spreadsheet using the same methods as the F9 add-in earlier.

Financial Entities

Financial Entities are being introduced with Version 4 of F9. These are names of items included in your accounting system *or business*. Financial Entities are a new application of *subaccount descriptions* which allow you to refer to your *cash* account as “Cash” rather than 100010. Subaccount descriptions are still available with the SDESC function described later.

There are actually two kinds of financial entities: “natural” and “user defined.” Natural Entities are normally imported from your accounting system, and may be added to or edited. The SDESC function can return any natural financial entity, which is one of their advantages. Now, F9 will recognize the account specifier 101* as “Cash in Banks.”

The second kind of Financial Entity, “user defined” is quite a bit more flexible and can be significantly larger than a natural entity. User defined entities are not accessible to the SDESC function.

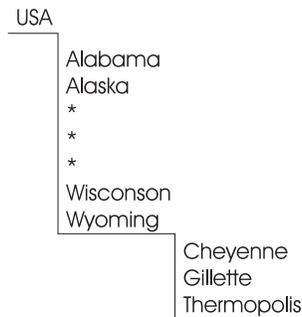
Financial Entities also extend subaccount descriptions in several ways. First, you can use a financial entity anywhere a segment value can be used. A financial entity is a named value of a segment specifier. This means that if segment 2 is “location” in your chart of accounts, and an entity for “Los Angeles” exists in that segment, then “Los Angeles” can be used anywhere you might provide the segment value. This means your GL function reads much more intuitively:

```
GL(BSPEC("Los Angeles", "All Cash", "All Departments"), "this month", "ACompany")
```

Second, you can define new Financial Entities for any account segment. If you have a range of accounts which includes all your sales income, you can define an entity “Sales-Gross” for the specifier 3002*. This entity can be used anywhere a segment value “3002*” would be used. Financial Entities allow “shorthand” for segments. For example, you may have a segment value and description for “Los Angeles” (which is imported as a “natural” financial entity), Financial Entities lets you create another entity for “LA,” too.

Finally, Financial Entities can contain other Financial Entities. That is, a Financial Entity can be composed of a collection of other financial entities. For example, in an accounting system where each location has a special value for a segment, you can also create a user defined entity for the state which includes all the locations in that state. But, going another step along, you can define a series of sales regions where each is a group of states.

Now, when a new location is opened, it is added to the state entity's list, and the sales region immediately sees it too. For example, a set of Financial Entities could model your geographic distribution:



Using this model, adding a location in Rawlins, Wyoming, you would:

- Create or import a “natural” Financial Entity of Rawlins which is a specific value of the “location” segment.
- Edit the Financial Entity for Wyoming so it now specifies: Cheyenne, Gillette, *Rawlins*, and Thermopolis.
- The Financial Entity USA then inherits the members of Wyoming so it requires no changes.

Tip:

This section of the manual discusses what Financial Entities are and how they are used. For information on how to edit them, see *Chapter 8: Editing Support Files*.

For some accounting systems or charts of accounts, it may be even simpler. If your chart of accounts uses two digits to encode the states (*e.g.*, 01-50), and two digits to encode the city (*e.g.*, 01-99) in a single segment, then Wyoming could simply be 50*. In this case, all that needs to occur is to create Rawlins as a Financial Entity.

Within F9, there are two types of Financial Entities: “natural” and user defined. In simplest terms, “natural” refers to those entities which are imported from the accounting system, whereas a user defined entity is not. The number of natural

Advantages of Financial Entities

- Both “natural” and user defined Financial Entities may have 50 character descriptions, but a “natural” Financial Entity may only use 30 characters in the segment value. A user defined Financial Entity may use 130 characters.
- “Natural” Financial Entities should be simple segment values without ranges, lists, or wildcards. User defined Financial Entities may use *any form* of specifier including ranges, lists, wildcards, or other financial entities.
- User defined Entities are listed first in the **Lists**, **Report Wizard**, and **GL Enquire** windows. Entities are simply easier to use.

Multiple segment Descriptions

The SDESC function can provide descriptions for multiple segments. Essentially, some accounting systems use subaccount codes which are completely dependent on the account number and now F9 can return descriptions for them.

In most accounting systems, each subaccount value always has the same meaning such as “location” or “department” and the individual values have exactly one meaning: 100 = Vancouver. This is not true for companies which require multiple segment descriptions.

Consider the following accounts:

<u>Account</u>	<u>Description</u>
10000-100	Cash - Credit Union
10000-200	Cash - Petty Cash
10000-300	Cash - Treasury Bills
⊕	⊕
⊕	⊕
⊕	⊕
20000-100	Mortgage - Los Angeles Office
20000-200	Mortgage - Atlanta Office
20000-300	Mortgage - Montreal Office

In this example, the subaccount 100 has a meaning of “Credit Union” or “Los Angeles” depending on whether the account is Cash or Mortgage. A similar problem can be seen with subaccounts 200 and 300. With multiple segment entities, the account specifier 10000-100 can be treated as an entity. 20000-100 is another entity.

SDESC now recognizes these multiple segment descriptions and will correctly return a fully qualified description for them. The following rules apply:

- the “account specifier” parameter has more than one segment *and* must not be a complete account specifier.
- multiple segment descriptions can **not** use lists, ranges, or wildcards; they must have a one-to-one match.

That is, these descriptions do not use SDESC in either of the previously supported ways.

When creating multiple segment descriptions, you create a Financial Entity for the *first segment* which the entity represents. For example, in a four segment account code the account entity may be segments three and four, so a two-segment entity could be created in segment three. In this way:

```
SDESC("2000-200", 3, "ACompany")
```

would return “Mortgage - Atlanta Office.” Of course, since we keep telling you to use cell references, this will probably look more like:

```
SDESC(BSPEC("^-^", $D5, $D6), 3, "ACompany")
```

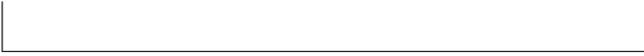
Note:

The parameter “^-^” in the BSPEC expression is an Account Mask. Multiple Segment Descriptions is one of the rare times where an account mask is required by BSPEC because the company’s Text Substitution Operator (TSO) has four segments and we only want to use two segments here.

In some cases, multiple segment descriptions will be able to be used as Financial Entities if they meet the following criteria:

- The segments used must be in consecutive positions within the account number.
- The multiple segment descriptions must use the segments in the same order as the accounting system.
- Due to their nature, a multiple segmented Financial Entity can not be used as an element in a segment list or range.

That is, given an account in the order A-B-C-D, you can use a description of B-C as a Financial Entity, but not one of B-D. Also, while it is possible to create a description for C-B, the resulting description would not be a valid Financial Entity.



Chapter

4

4. Reporting Wizards

In Windows terms, a wizard is simply a tool that automates any process into simpler, ordered steps. F9 has two wizards for reporting: the elder is the “Report Wizard” which builds complete reports including numerous formatting styles. Version 4 of F9 introduces a new “wizard” to help you build reports: The GL Wizard.

The Report Wizard is a comprehensive tool to build almost any report complete with extensive formatting. After using the five steps of this wizard, you will have a complete, formatted report. F9 is shipped with a number of report formats, such as the plain one suitable for laser printing, or a livelier layout for on-screen or colour printing.

The GL Wizard provides all the basics of a powerful F9 report without all the “bells and whistles” of the Report Wizard. The GL Wizard is more modest in that it provides just enough information to build one GL formula. But, as we keep saying, you only need one formula copied everywhere else!

Both of the F9 Reporting Wizards have one goal: to let you write a report with as little typing as possible. If you can build a report with *no typing* we have succeeded.

The GL Wizard



Enquire Button

The GL Wizard uses two steps to build a minimal F9 Worksheet with a single GL function.

First, select **F9|Enquire** from the menu or press the Enquire button (magnifying glass) on the F9 toolbar to open the Account Enquiry dialog. Set each of the controls to a reasonable value. After making selection in each control and pressing the **Get Balance** button, the window will resemble:

Location	Account	Department	Balance
All Locations	10005	All Departments	2,010.96

Enquiry Result

This simple request is for a balance totalling All Locations (a financial entity) an account, totalling All Departments. The main setting for a good report is the “Create GL function by” option. This controls whether the headers will be a company, an account number segment, or a period. When the controls are filled, press the Quit button on the dialog and return to your spreadsheet.

For any report, we recommend knowing in advance what is on the vertical and horizontal axis. To make the GL Wizard report as simple as possible, query on the first account number, and set the “Create GL function by” option to the horizontal axis.



GL Paste Button

The second step, which is the GL Wizard at work, is to press the **GL Paste** button (a glue brush) on the F9 toolbar. The **GL Paste** should not be confused with **Edit|Paste** which simply drops the clipboard onto your sheet. The **GL Paste** function interprets the result of the last Account Enquiry dialog and makes an F9 formula from it. The GL Wizard tries to place the GL formula in the same cell your cursor was in *when the GL Paste button was pressed*. The cursor should move only when F9 needs to make space for the parameters. If you expect to add row descriptions or section headers, moving the cursor to another cell before pressing the button will let you add rows or columns easily.

The GL Paste Command works by interpreting the Account Enquiry result on the clipboard. If an Enquiry result is not on the clipboard, then the Enquiry command will be run in the background using the settings from the last Account Enquiry to fill the clipboard.

A GL Wizard Example



Enquire Button



GL Paste Button

The first step in building a report with the GL Wizard is to perform an account enquiry to get a single balance. To do this, select **F9 | Enquire** from the menu or press the Enquire button. Press **Quit** to close the window and return to your spreadsheet.

On an empty worksheet, press the GL Paste button, and F9 will change the sheet will resemble:

	A	B	C	D	E	F	G	H
1	Location	All Locations		All Departments				
2								
3								
4	Period	Month						
5	Company	Demo						
6	Year	1997						
7	Type	ACTUAL						
8	Currency	Home						
9								
10								
11	10005			-2010.96				
12								



Lists Button

One number is hardly a report, but so far you haven't typed anything into your spreadsheet. In keeping with that, select **F9 | Lists** or press the Lists button. The first axis to be completed is the vertical list of accounts; this list dialog is set to return a list of segment values for the natural account code:

Lists [Univ]

Available Lists:

- Segments
- Companies
- Budgets
- B/WB Headers
- T/WB Headers
- Lists
- Periods
- Years
- Currencies
- Accounts

List Segment: Account

Where: Account Is: 10000.10299

Transpose List?

Generate Descriptions?

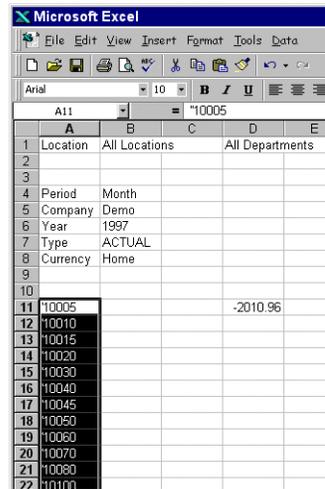
Use the Financial Entities File for Segments?

Buttons: Send to Clipboard, Quit, Help

4. Reporting Wizards

A GL Wizard Example

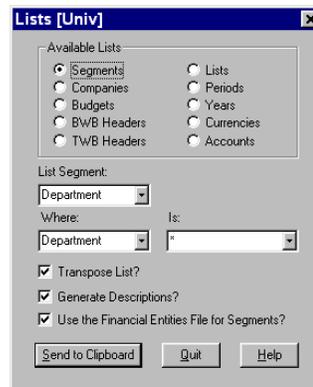
Pressing the **To Clipboard** button will send the list to the clipboard. Return to your spreadsheet and place the cursor on the first account number in column A then simply select **Edit | Paste** or press **Ctrl-V**. The list will be pasted into the column under the current cell and your sheet will now contain:



The screenshot shows a Microsoft Excel spreadsheet with the following data:

	A	B	C	D	E
1	Location	All Locations		All Departments	
2					
3					
4	Period	Month			
5	Company	Demo			
6	Year	1997			
7	Type	ACTUAL			
8	Currency	Home			
9					
10					
11	"10005			-2010.96	
12	"10010				
13	"10015				
14	"10020				
15	"10030				
16	"10040				
17	"10045				
18	"10050				
19	"10060				
20	"10070				
21	"10080				
22	"10100				

The final axis is the horizontal list of departments; set the list dialog to return a list of segment values (e.g., we'll use department names) *and transpose the list*. Transposing the list makes the result horizontal. The Lists dialog will look like:



The screenshot shows the Lists [Univ] dialog box with the following settings:

- Available Lists: Segments, Lists, Companies, Periods, Budgets, Years, B/WB Headers, Currencies, TWB Headers, Accounts
- List Segment: Department
- Where: Department
- Is: *
- Transpose List?
- Generate Descriptions?
- Use the Financial Entities File for Segments?
- Buttons: Send to Clipboard, Quit, Help

Again, place the cursor in your spreadsheet on the cell with the first item, in this case cell D1 which reads “All Departments.” Select **Edit | Paste** or press **Ctrl-V** to paste the clipboard contents to your spreadsheet. The list will be pasted into the row beginning with the current cell and your sheet will now look like:

	A	B	C	D	E	F	G
1	Location	All Locations		All Departments	Sales	Shipping	
2							
3							
4	Period	Month					
5	Company	Demo					
6	Year	1997					
7	Type	ACTUAL					
8	Currency	Home					
9							
10							
11	*10005			-2010.96			
12	*10010						
13	*10015						
14	*10020						
15	*10030						
16	*10040						
17	*10045						
18	*10050						
19	*10060						
20	*10070						
21	*10080						
22	*10100						

Everything is now in place to actually fill in the body of your report, to copy that one GL function everywhere. Start by placing your cursor on the cell where the GL formula resides and auto copy the formula.

Excel: Drag the small handle in the bottom right corner of the cell either horizontally or vertically to copy the current selection (one cell) into adjacent cells. Once you’ve dragged in one direction, release the handle, then click on it again and drag it the other way.

	A	B	C	D	E	F	G
1	Location	All Locations		All Departments	Sales	Shipping	
2							
3							
4	Period	Month					
5	Company	Demo					
6	Year	1997					
7	Type	ACTUAL					
8	Currency	Home					
9							
10							
11	*10005			-2010.96	-1005.48	-1005.48	
12	*10010			-963121.88	-93518.83	-869603.05	
13	*10015			-11140.76	-9552.09	-1588.67	
14	*10020			21466.16	-4021.93	25488.09	
15	*10030			-6977.93	-549.46	-6428.47	
16	*10040			-3601508.71	-2403466.51	-1198052.2	
17	*10045			148811.35	106592.07	40219.28	
18	*10050			-63023.62	0	-63023.62	
19	*10060			-44689.94	-27034.88	-17855.06	
20	*10070			0	0	0	
21	*10080			-4725.78	-2010.97	-2714.81	
22	*10100			-146303.16	0	-146303.16	

Lotus 1-2-3: To auto copy in 1-2-3, hold the CTRL key down while dragging the cell and the cell will be copied into adjacent cells. Unlike Excel, Lotus can copy to two dimension at once so only one drag is needed.

Tip:

We recommend using cell references *a lot* (and for good reasons). The key to cell references is correctly making them absolute (\$) and relative. When auto copy fails, the most likely problem is a relative cell reference.

Press F9 to recalculate the spreadsheet. *Now* its almost a report. Account descriptions are the only thing left to add. The fast and easy way to insert descriptions is to use the SDESC function which is as simple to use as the GL function:

SDESC(AccountSpec, SegmentNum, Company)

or, using cell references:

SDESC(\$A11, 2, \$B\$5)

The account specifier in an SDESC function is normally either a complete account code or just one segment value. The segment number determines which segment the value is from. For example, working within the same example, we would like descriptions of the main account segment (vertical axis). So, the sheet then looks like:

	A	B	C	D	E
1	Location	All Locations			
2					All Departm...
3					
4	Period	Month			
5	Company	Demo			
6	Year	1997			
7	Type	ACTUAL			
8	Currency	Home			
9					
10					
11	'10005	Petty Cash		-2010.96	-
12	'10010	Cash in Bank - Account #1		-963122	-
13	'10015	Cash in Bank - Account #2		-11140.8	-
14	'10020	Cash in Bank - Payroll Account		21466.16	-
15	'10030	Marketable Securities		-6977.93	-
16	'10040	Accounts Receivable - Trade		-3601509	-
17	'10045	Allowance for Bad Debts		146811.4	1
18	'10050	Notes Receivable		-63023.6	-
19	'10060	Interest Receivable		-44869.9	-
20	'10070	Other Receivables		0	-
21	'10080	Accounts Receiv - Employees		-4775.78	-

Notice that, in this SDESC function, the segment value is from column \$A, and the Company is always from cell \$B\$5. When cell references are done right, you will make reports where all the GL formulas are pasted from the first one and all the SDESC formulas are pasted from the first one. Now, with the GL Wizard, you only need to write the SDESC function.

The Report Wizard



Report Wizard

The F9 Report Wizard has two primary functions. First, it can be used to create a wide variety of reports automatically. Second, the Report Wizard can be a useful aid to understanding how to create F9 formulae because reports created by the Wizard always make the best use of cell references possible.

The Report Wizard is not available for users of Quattro Pro at this time.

The Report Wizard presents your general ledger to you as if it were a multiple dimensional database. In a multiple dimensional database, a report can be thought of as a two dimensional slice out of the database. The process of creating a report consists of selecting values from two of the dimensions to act as the horizontal and vertical dimensions (or axes) of your report, and providing values for the remaining dimensions. The Report Wizard can take any two-dimensional slice out of your multiple dimensional general ledger database to create a report.

In much of the following text, we use the phrase “axis” rather than “dimension” but they are generally synonymous.

The dimensions of an accounting system database are companies, budgets, years, periods, and the various segments of your account code. Your account code consists of the natural account code (*e.g.*, 1000 for “cash”) plus other segments such as subaccount, department, location, *etc.*. Which segments are available varies from accounting system to accounting system, and is often configurable within an accounting system. That is, it is possible that company “A” may have a 5 segment account code and company “B” a six segment account code.

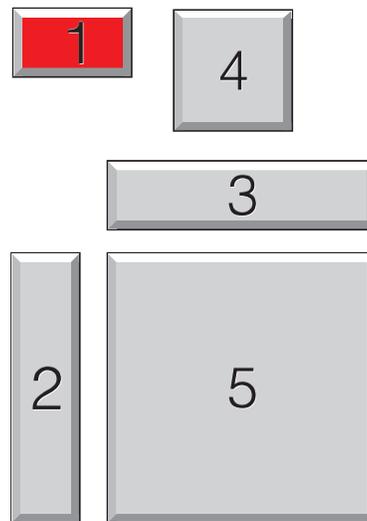
The Five Steps to a Report

There are five steps to this wizard which are the same steps that are required to create *any* report using F9. The principle difference is that while many F9 reports evolve from other reports, Report Wizard reports are all created from scratch using the information selected from the lists the Wizard provides. In general, if you wish to use the Report Wizard, we recommend that you start it on a blank spreadsheet.

Each of the steps in creating a report are illustrated here and in the Report Wizard Dialog.

① *The Report Wizard will need to know what company you want to act as the Model Company. The Model Company will determine which axes the multi-dimensional database will have and the chart of accounts which will be used for all subsequent steps in the report creation process.*

② *In Step 2, you build the vertical axis, usually Accounts, but not necessarily. If you select Accounts, you must then select which accounts. If you select departments, for example, you must then say which departments.*



③ *In Step 3, you build the horizontal axis, usually periods, but not necessarily. After selecting what data will be on the axis, you select which items are used. If you select periods, you must then select which periods are present. If you want to use departments, you must then select which departments.*

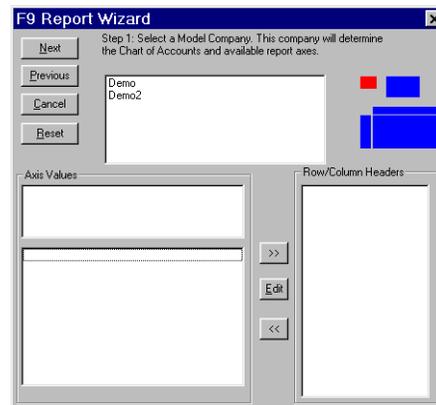
④ *In step 4, you must select a value for the remaining axes... that is, the axes which do not make up either the vertical or the horizontal axes. Where ever possible, F9 will provide a list of available values.*

⑤ *In this step, you can provide a report title and set a number of other options. You can request that the report be created at the current cursor position, or on a new sheet. You can request that the report be created at the current position or on a new sheet. You can request that the Zero Suppress function be invoked on the report body after creation. You may also request that the Report Wizard execute a formatting macro*

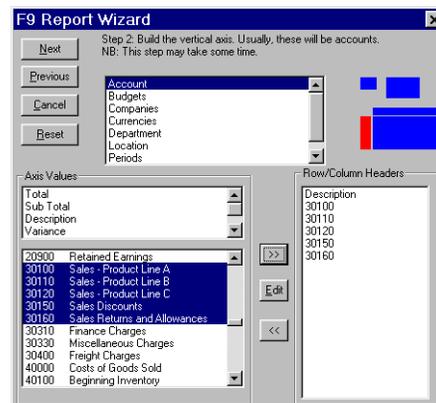
Creating a report with the Report Wizard is a five step process. The report diagram on the report wizard dialog shows this same illustration. As you work through each of the steps of the report creation process, the report diagram change colors indicating the step in the process you are performing.

A Report Wizard Example

When you first select the Report Wizard option, the Report Wizard dialog box will appear similar to the following figure. Note the text at the top of the dialog box describing what you need to do to perform **Step 1: Select a Model Company**. We will select the F9 Demo company and then press Next.

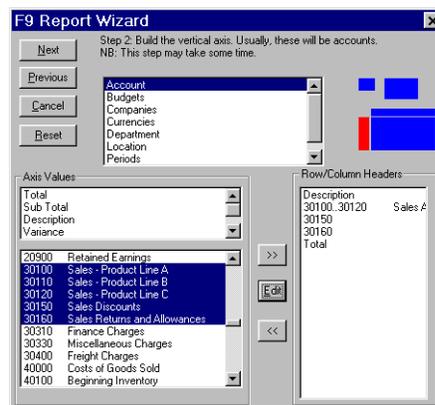


This advances us to **Step 2: Creating The Vertical Report Axis**. The axis selection box at the top of the dialog box now contains a list of the various dimensions available for reporting. You might build a report of companies by period, or departments by product (where product might be a segment of the account code, or just a set of accounts, *etc.*) but most reports will use natural accounts as the vertical axis. For example, you might be preparing a monthly sales analysis. In this case, sales accounts would be the vertical axis and periods the horizontal. We will select “account” for the vertical axis.



At this point, you will notice a significant pause as the Report Wizard builds a list of accounts for you to select from. When the list is built, the left side Select From list box (actually two list boxes labelled as “Axis Values”) will contain a list of all the unique natural accounts from the Model Company.

Now we will select the accounts we are interested in. After selecting the accounts, we press the right pointing arrow button to move them to the Selected List (this is the right hand box labelled “Row/Column Headers”). We might also add a total line by selecting Total from the Standard Selections list box (the top list box in the Axis Values group on the left). Now the Report Wizard will appear as follows:



Copying, Inserting and Removing Lines

Select the lines desired from the left hand Select From box and copy them to the right hand Selected box. New lines are added to the end of the list. You may insert lines above any Selected box line by selecting that line before copying the desired lines from the Select From box. Lines may be removed by selecting them in the Selected box and pressing the left pointing button.

Standard Lines

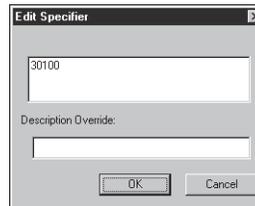
Select Standard lines to add totals, subtotals, descriptions, variance, percentage variance and blank lines. To add a standard item to the bottom of the list box highlight the line and press the right arrow button. To add a standard item to the top, or other position within the list box, highlight an item in the list box, then highlight the standard line item you require, then press the right arrow button. The standard line will appear above the originally highlighted line in the list box.

Tip:

Description, as a row or column header, will include SDESC functions for account segments on the *other* axis of the report. So a Description column will make an SDESC function for the segment on each row. If there is no segment in the other axis, the cells will be blank.

Collapsing and Editing Lines

Once values have been selected and moved to the Selected box on the right, you may edit them if desired. Select a line on the right hand Select box and double-click on it (or press the **Edit** button). The Edit dialog box will appear.



You may now modify the line. For example, perhaps you selected the account “10100” but you would really like to see all accounts that begin with “101”. To do this, change the selection to “101*”. When the axis is a segment of the account code, ranges, lists and wildcards (*, ?) may be used. You may also provide a new description for this line if desired. This description will override the description F9 would otherwise return.

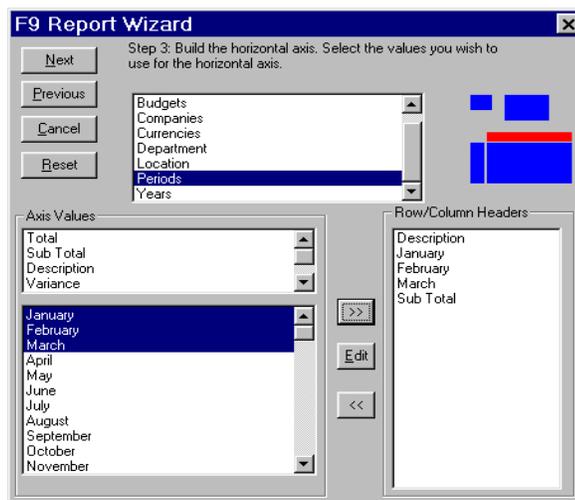
If you select more than one consecutive line, and the axis is an account code segment, and you then press the Edit button, a dialog box will appear asking if you want to make the selection into a range. If you answer yes to it, the Report Wizard will turn your selection into the appropriate range automatically. You may also add an override description to grouped accounts.

Expanding Lines

When working with the vertical axis, you may also expand a line based on one of the remaining axes. For example, suppose you were creating a sales report by period and you have a single natural account 4000 for revenue. And suppose further that the subaccount breaks sales down by product. You could expand the sales line by selecting it in the right hand Selected box and then selecting Sub-Account in the Axes Selection box at the bottom of the Report Wizard. All the Sub-Accounts that are associated with account 4000 will appear in the Select From box on the left. You can then insert those lines under the Sales line.

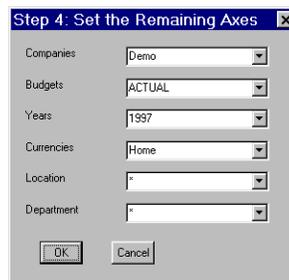
We now press the Next button and advance to **Step 3: Constructing The Horizontal Axis**. This step is identical to Step 2 in concept. This time, we will select Periods as the horizontal axis. A list of many (but not all) possible periods appears in the Select From box.

First we will select Description from the list of Standard elements. This will cause an account description to be the first column of our report. Then we select our periods (in this case, the first 3 months of the year) and a subtotal as the last column. The Period Wizard dialog box now appears as follows:



Step 3 is now complete and pressing **Next** will move us on to **Step 4: Determining the Remaining Axes**. (e.g., the axes that are not part of the vertical or horizontal part of our report.) We will need to provide the company we wish to use. In most cases, this will just be the Model Company selected in Step 1. We also need to provide the year and budget (if any). If there are other segments of the account code, such as location, subaccount or department, we must provide values for them as well. These will usually default to the "*" wildcard indicating "all values". That is, in this example, the report will be for all locations and departments. Where ever possible, F9 will provide a list of available values. It is not terribly important to get these values "right" as you can always change them on the resulting report. All the values set here just wind up in a collection of cells created outside the main body of your report. However, we might provide a value for "100" for the department indicating that we want this sales report to be only for one department. The dialog box that will appear will be different depending on which axes you selected in the previous two steps.

In our example, it will look like this:



Having selected values for the remaining axes, we can move on to **Step 5: Complete the Report**. Step 5 lets you give a title to your report, select where the report should be built, and also lets you provide the name of a macro you would like to use to format the report. We provide a library of formatting macros you can use, or you may build your own.



In this step, you can provide a report title and set a number of other options. You can request that:

- the Report Wizard execute a formatting macro.
- the report be created at the current cursor position, or on a new sheet [Lotus 1-2-3 only].
- the ZeroSuppress function be invoked on the report body after creation.
- the Report Wizard either generate GL functions (best if you are working with balance sheet accounts) or NGL functions (best if you are working with income statement accounts).

Formatting macros are a collection of macros found in the spreadsheet called REPWIZ in the F9 directory. We supply a number of macros you can choose from or you can construct your own.

To construct your own macros, load the file REPWIZ, make a copy of one of the macros we have created, and customize it. Any macros you create will be automatically available to you in the Report Wizard after you save the spreadsheet.

If your spreadsheet is blank, then you should let the Wizard create the report at the current cursor location. If your spreadsheet is not blank, we strongly recommend that you select the Create Report on a New Sheet option. Otherwise the Wizard may destroy part of the current sheet's contents.

Finally, we can let the Report Wizard build the report. Our final report looks like this:

Quarterly Sales Report				
Description	January	February	March	Sub Total
Sales, All Products	-500,755.29	-505,262.08	-509,809.45	-\$1,515,826.82
Sales Discounts	33,620.00	33,922.58	34,227.88	\$101,770.46
Sales Returns and Allowances	3,510.00	3,541.60	3,573.46	\$10,625.06
	-\$463,625.29	-\$467,797.90	-\$472,008.11	-\$1,403,431.30

The report the Report Wizard builds is hot-linked. That is, all the numbers are generated by GL functions created by the Wizard.

Editing a Report Wizard Report

If you are not happy with the final results, you can either edit the report using standard spreadsheet and F9 functions, or you can re-invoke the Report Wizard and change some or all of your choices. In general, **we recommend that if you wish to change your report after the fact that you use your spreadsheet in the usual way to make changes.** That is, if you want to add a new row, use spreadsheet functions to copy and insert another row.

When the Report Wizard is re-invoked on a spreadsheet that contains a Report Wizard report, the Wizard will read all the settings you originally made off the spreadsheet so that you do not have to make your selections all over again. **If you then make changes to your selections, make sure you choose the Create Report on a New Sheet option in the Step 5 Report Creation dialog box.**

Chapter 5

5. F9 Functions

This section describes F9's functions and how to use them. The spreadsheet Add-in adds these functions *into your spreadsheet*. They can be used just like any other formula or function in your spreadsheet.

Functions are also distinct from either commands or an F9 window; a function will only return a result or an error to a cell in your spreadsheet. Functions will not present a window. Commands are available from the F9 menu and they can present a variety of windows to you (See the next chapter).

F9 is most likely to be used with a windows spreadsheet; an add-in is available for recent versions of Microsoft Excel, Lotus 1-2-3, or Quattro Pro. In each of these cases, F9's functions are accessed by using spreadsheet functions that are provided by the F9 client add-in. Functions retrieve data from one or more general ledger companies for a particular topic (*i.e.*, accounting system or DLL).

The rest of this manual will document each F9 function as if it were being accessed using either an Excel function or a Lotus 1-2-3 / Quattro Pro function. In Excel, a formula is preceded by an equal-sign(=) while both Lotus 1-2-3 and Quattro Pro precede functions by an at-sign (@). Other than this minor difference, functions in Excel, Quattro and Lotus 1-2-3 are identical.

Each F9 DLL supports almost the same set of functions. All DLLs will support the GL, NGL, DESC and ACCTDATA and CODATA functions. This chapter documents the "generic" set of F9 functions. **Variations specific to each DLL are documented in the DLL's help.**

The last parameter to every function is the topic and is the name of the DLL or accounting system that you wish to access. If you are using only one DLL (that is, you are only using one accounting system), then you can ignore this parameter. If no topic is provided, then the client add-ins will use the

topic “F9”. This instructs the Server to use the default topic. In the case where only one DLL is installed, the Server’s default topic is set to that DLL automatically.

Note:

Behind the scenes, each function is converted into a DDE request to the F9 server. If you are using a spreadsheet with a client add-in (e.g., F9.XLL or F9.ADW), all of these DDE issues are taken care of for you and you need not worry about the formatting of the DDE item. Users who wish more information should see *Chapter 7: Advanced Features*, or contact Synex Systems for more information.

This section includes:

1. Summary of F9 functions and function syntax
2. Specifying accounts in functions
3. Specifying periods in functions
4. F9 functions used to produce account balances
5. F9 functions used to produce descriptive account information
6. F9 functions used to produce department information
7. F9 functions used to produce company information

Conventions Used in this Section

This section uses the following conventions to describe function parameters:

<i>italics</i>	You must supply the value or set of values for any of the parameters shown in italics. For example, when <i>acct</i> appears, you should specify the account(s) you want or the worksheet cells that contain the account(s).
[] brackets	Parameters within brackets are optional. Specify the parameter only. Do not type the square brackets themselves.
() parentheses	You must enclose function parameters within parentheses for the function to work.
, commas	You must separate function parameters with commas.

“” quotes You must enclose function parameters in double quotation marks unless the parameter is expressed as a reference to a cell or range of cells containing labels.

The Absolute Minimum

In keeping with the 80/20 rule (80% of the work is performed by 20% of the product), the following four functions will allow you to produce the vast majority of your reports. The four functions are GL/NGL, BSPEC, SDESC, and CODATA. These functions are covered in detail later in the chapter.

The GL and NGL functions allow you to bring specific account balances into your worksheet. They use the following parameters:

GL(Acct,[Period],[Company],[Year],[Type],[Currency],[Topic]]])

NGL(Acct,[Period],[Company],[Year],[Type],[Currency],[Topic]]])

These functions produce balances for the accounts, period, and company or companies specified in the formula or the Setup Dialog. The difference between NGL and GL is just that the balance is returned with the sign reversed (negative becomes positive and vice versa).

The BSPEC function's sole purpose is to allow you to construct account specifiers *using cell references* — typically one cell reference for each segment of the account specifier. It achieves this by manipulating strings — specifically account specifier ranges and substitution lists — to add considerable power. For example, if you wish to create a column that summed two departments, all you would have to do is change the contents of a cell from “100” (for example) to “100,150”. BSPEC successfully permits lists as a segment value into F9 formulas. The BSPEC takes a list of segment values and an optional TSO Mask (explained later in this chapter):

BSPEC([TSO Mask],[Segment 1[, Segment 2 [, ...Segment N]])

BSPEC allows you to deal with each account segment independently. Any account segment can be:

- a cell reference.
- a financial entity.
- a range of values in a segment.
- a comma separated list of segment values.
- multiple specifiers in a cell range.
- wild card characters in any account part.
- almost any combination of the above.

The SDESC function produces the descriptions associated with segment values of the account code. Most account codes consist of at least two segments, and usually more. Often the account segments have specific meanings. For example, a four part account code might consist of the location, profit center, account and subaccount. In this example, each location code and profit center code may have a description associated with it (usually a subaccount will not have a description, but it might).

SDESC(Account,Section[,Company[,Topic]])

The CODATA function returns a variety of data about the company. Using CODATA to build report headings, for example, allows you to build reports which automatically display the company's name, current period, *etc.*. These results allow you to maximize the usefulness of your reports.

CODATA(Data[,Company[,Topic]])

The CODATA function normally accepts requests for the following data:

Name	Main
Period	TSO
EndN	StartN
Year	Date
Origin (Accounting system name)	

These requests allow you to include data which changes for each company. For example, the full company name can be included in the report title.

Summary of Functions

Note:

Not all functions are supported by all DLLs (*i.e.*, accounting systems). Some DLLs will support additional functions. These are documented in the DLL's help system. See "F9 Functions". Instructions for opening the DLL Help file are part of *Chapter 6: F9 Commands and Windows*.

Producing Account Balances

The GL and NGL functions allow you to bring specific account balances into your worksheet.

GL(Acct,[Period],[Company],[Year],[Type],[Currency],[Topic]]])

NGL(Acct,[Period],[Company],[Year],[Type],[Currency],[Topic]]])

GLTRAN(Acct,FromDate,ToDate,[Company],[Type],[Currency],[Topic]]])

These functions produce the balance for accounts, period(s), and company or companies as specified in the formula or the Setup Dialog. NGL differs from GL in that it returns the balance with the sign reversed (negative becomes positive and vice versa). For example:

```
GL("0-0-1110-0", "Current month starting balance", "abc")
```

would produce the opening balance for the current month for account number "0-0-1110-0" in the GL of company "abc." The GLTRAN function also produces account balances but it differs from the GL function in two respects: rather than using a period specifier, it uses a "From" and "To" date to indicate the period desired; and GLTRAN calculates balances from account transactions, and not period balances.

Do not use the GLTRAN function to produce net transactions for a fiscal period. The net transactions (or net change in balance) for any *fiscal* period can be obtained from the GL function by using the word "transactions" or "change" in the period specifier. The GL function is much faster than the GLTRAN function for this purpose. Only use the GLTRAN function when you need data for a period *other than* a fiscal period. By supplying a Type parameter of "Debit" or "Credit", the GLTRAN function can return just debit totals or just credit totals.

The GLTRAN function can potentially be extremely slow compared to the GL function, but it does allow you to create financial reports for *any* period (e.g., you could produce a 1 week income statement).

Producing Other Account Data

The functions described below allow you to bring specific descriptive account information into your spreadsheet. They are listed in alphabetical order. If an account specifier is used for any of these functions that matches more than one account, the descriptive information for the first matching account is returned.

DESC(Acct[,Company[,Year[,Type[,Topic]]]])

The DESC function produces the description of the specified account in the specified company. For example:

```
DESC("0-0-1110-0", "abc")
```

would produce the description of account number "0-0-1110-0" in company "abc" (e.g., "Cash In Banks"). If you specify more than one account, the description for the first account will be displayed.

SDESC(Account,Section[,Company[,Topic]])

This function produces the descriptions associated with segment values of the account code. Most account codes consist of at least two segments, and usually more. Often the account segments have specific meanings. For example, a four part account code might consist of the location, profit center, account and subaccount. In this example, each location code and profit center code may have a description associated with it (usually a subaccount will not have a description, but it might). To get the description of the location, you would use the SDESC function as follows:

```
SDESC("A-100-1000-0", "1")
```

where the "1" indicates that the SDESC function should return the description for the first segment of the account — in this case the location. Similarly, to get the profit center description, you would enter:

```
SDESC("A-100-1000-0", "2")
```

Most accounting systems will not have descriptions for all segments of the account code. F9 Financial Entities allow you to create new segment descriptions for your accounting system.

Note:

The DESC and SDESC functions are *not the same*. The results of

```
DESC("A-100-1000-0")
```

and

```
SDESC("A-100-1000-0", "3")
```

are *not* the same. The DESC function returns information about *all* segments (e.g., "Los Angeles - Administration - Cash + Megabank #300 ") while SDESC will only return one segment's meaning (e.g., "Cash + Megabank #300 ").

An alternate use of SDESC is to provide a single segment value and specify which field this value is from. For example,

```
SDESC("100", "2")
```

will return the same result as:

```
SDESC("A-100-1000-0", "2")
```

Multiple segment Descriptions and SDESC

Version 4:

Some accounting systems allow subaccount codes which have a different meaning dependent on the main account segment. In these systems, a particular account code and its subaccount codes are highly interdependent. The implementation of Financial Entities allows SDESC to recognize multiple segment descriptions to provide accurate descriptions in this case.

The only requirements that must be observed when using SDESC with multiple segment descriptions is that the "account specifier" parameter:

- must have more than one segment *and*
- must not include a complete account specifier.

In other words, a request for a description of a multiple segment value must not satisfy either of the other two acceptable uses of SDESC described above.

For example, if your accounting system uses two-segment description from a four segment account number, to retrieve an SDESC description, you must provide at least two segments but can not provide all four.

When creating multiple segment descriptions, the financial entity should be created for the *first segment* which the combined segment represents. For example, in a four segment account code the account entity may be segments three and four, so a request for a multiple segment description could be created in segment three. In this way:

```
SDESC("20000-200", 3, "Demo")
```

would return "Mortgage - Atlanta Office."

Since a multiple segment description will never use the company's default mask, the BSPEC requires a Text Substitution Operator such as:

```
SDESC( BSPEC( "^-^" , $A4 , $B4 ) , 3 , "Demo" )
```

For a more complete discussion of multiple segment descriptions, refer to "Financial Entities" in *Chapter 3: F9— An Overview*. For more information on creating multiple segment entities, see *Chapter 8: Editing Support Files*.

ACCTDATA(Data,Account[,Company[,Topic]])

The ACCTDATA function returns other data about accounts. The meaning of the data is returned is determined by the "Data" parameter. See the DLL help for a list of any special values this function might return (See "Getting Help" in *Chapter 7: DLL Commands and Windows*). The following values for Data are supported by most DLLs:

Description	Returns the same string as the DESC function.
Type	<i>e.g.</i> , Asset
Typical Balance	<i>e.g.</i> , DR
Origin	Balance or P&L

Producing Company Information

CODATA(Data[,Company[,Topic]])

The CODATA function is similar to the ACCTDATA function. Data about the company is returned. Values of "Data" that are supported by most DLLs are:

Name	A user friendly name for the company. Suitable for report titles <i>etc.</i> , <i>e.g.</i> , The XYZ Company
Period	This is the current fiscal period
Origin	The name of the accounting system
Date	The current fiscal date
Year	The current fiscal year

Main	The segment number of the main or natural account code segment
TSO	A TSO mask appropriate for the company, <i>e.g.</i> , ^-^-^
StartN	The start date of fiscal period N
EndN	The end date of fiscal period N

The CODATA function allows your reports to include the current values being used by F9 for a company (*e.g.*, the current period on a report for "this month"). If no company is supplied, F9 returns the values associated with the default company.

The online help for your accounting system DLL has more information about the information which can be returned by this function (See "Getting Help" in *Chapter 7: DLL Commands and Windows*).

OD(...) or **OTHERDATA(...)**

This function, if implemented, is always DLL specific and is only documented in the online help file (See "Getting Help" in *Chapter 7: DLL Commands and Windows*). For most accounting systems, this function is not available.

Producing Non-GL Data

F9 version 4 introduces a new function, F9DATA for accessing non-GL information available to some accounting systems. These non-GL databases are often referred to as "column-wise" data. This function is not available for most accounting system DLLs. If used with a system that does not implement the function, F9 returns an error message.

F9DATA(Code, Column [, DataMart [, Year [, Type [, Currency [, Topic]]]])

Code	This is a user defined selector (<i>e.g.</i> , a part number, a client number, <i>etc.</i>).
Column	A reference to one of thirteen named columns of data available for the code.

DataMart	The name of the database, company, or other type of data collection being accessed (See the online help for specifics of this parameter).
Type	This is the type of data being accessed. The meaning of this term will depend on the database being accessed. For example, in an inventory control database, this may be “On Hand,” “On Order,” or “Used.” (See the online help for specifics of this parameter).
Currency	The meaning of this term will depend on the database being accessed. (See the online help for specifics of this parameter).
Topic	The Topic refers to which accounting system DLL this request should be passed to. Most users only have one system available and do not need to provide this parameter.

The Write Functions

We don't know all the uses you'll find for the GL writing functions. We do know that your spreadsheet can create a set of books, *any set of books*, and then you can manipulate and analyse the numbers with the F9 GL function and your spreadsheet.

Tip:

Before you use the WGL and WDESC functions extensively, be sure to read the notes about limitations below, and the use of account masks and specifiers which follow.

Limitations of Write Functions

Both WGL and WDESC share some important restrictions:

- the account specifier can not include ranges or wildcards.
- only the **first** matching account is written to when a list of accounts is provided.
- the Type parameter can not be "ACTUAL" as this has special status to accounting systems. Type can be any other valid type *for your*

accounting system. So, if your system supports one to three budgets, so do these functions. If your system allows almost any type (e.g., the F9 Datamart), so do these functions.

The WDESC function also has the following restriction:

- The WDESC function is not implemented for accounting systems which have a “global” chart of accounts; in these systems, changing an account description would effectively change an ACTUAL account and the write functions are not permitted to do that.

Note:

While all the Version 4 implementations of F9 are expected to implement these two functions, some implementations will require additional steps to enable the functions. These steps will be documented in the DLL help file installed with F9. Select F9|DLL Help to open the help window. Select Functions within the help file and GL Specific Help to see information regarding your GL's implementation.

The WGL Function

The WGL function can write an amount to a pre-defined or user defined general ledger. The syntax is identical to an F9 GL function except it adds an amount reference as the first parameter:

WGL(Amount, Account, Period, Company, Year, Type [, Currency [, Topic]])

The normal appearance of this function will be identical to a GL function except that the “amount” would normally be an “off sheet” cell reference. Using the sample worksheet “Write.xls,” the WGL functions all appear on the sheet “WGL” and they refer to values on sheet “*pro forma*.” You can compare the GL and WGL functions by looking at the same cells on the sheet “GL.”

The parameters of the WGL function are intentionally similar to the GL function, with the added "Amount" parameter. The write functions have fewer optional parameters because neither will write to "actual" which is the default type for most users. It is possible to use the default company and year by simply providing blank values. *i.e.:*

```
WGL($A3,A$2,E$11,"","","Budget")
```

The WGL function has restrictions on the Account Specifier as described below. Financial Entities can be used within existing companies.

Note:

Depending on the accounting system, account descriptions and Financial Entities may not exist for future years. See the DLL Help file.

The most natural use of the WGL function is to prepare budgets. Your spreadsheet now allows you to read a budget in, revise the numbers appropriately, and write the budget into your GL. See also *Chapter 9: Budget Writeback* for more information.

Setting the Account Origin in WGL

The account origin defines whether an account is intended to be used on the balance sheet or income statement. This implies whether the account has a balance or is understood as net transactions. Accounts default to P&L origin.

A balance sheet account can be specified by adding ",B" to the type parameter of the WGL function. This means a type of "Budget" is from the P&L, and "Budget,B" is from the balance sheet.

Tip:

Since the most common use of the WGL is expected to be budgeting, the default of P&L as an origin minimizes the need for this parameter. Remember to use a comma and not to use any spaces.

Using BSPEC with WGL

BSPEC functions may require the Text Substitution Operator (TSO) under some circumstances when used with the WGL function. In particular, when a company is created, it *generally* does not have a default TSO which is a requirement of the WGL; that is, WGL requires either BSPEC or the accounting system to provide a proper TSO. If the company does not exist in advance, BSPEC must provide a TSO.

Period Specifiers with WGL

The WGL function will accept a comma separated list of period specifiers just as the GL function will. The result is that the amount specified is written to *each* of the periods in the list. This allow you to calculate a quarterly budget and write the amount back to the GL with a function like:

```
WGL(Round(GL2/3),BSPEC($A$3,$A12,$a4$), "Month 1
Budget,Month 2 Budget,Month 3 Budget")
```

This will take a quarterly budgeted figure, divide it by three, round off the pennies, and write the result into each of the three periods. **Note:** the amount provided is written to each period, *not* a portion of it to each period.

The WDESC Function

The WDESC function writes an account description to a user defined general ledger. The syntax is identical to an F9 DESC function except it adds an account description as the first parameter:

WDESC(Description, Account [, Company [, Type [, Year [, Topic]]]])

The sample worksheet “Write.xls” includes a column of WDESC functions. The normal appearance of this function will be identical to a DESC function except that the description should be an “off sheet” cell reference. This write function will allow you to write a description that applies *only* to the complete account number.

Multiple segment account codes can be built using BSPEC. Most users will use at least two segments (department and account) and a simplified chart of accounts. The WDESC function respects financial entities so an account specifier can be created using the standard approach: BSPEC(“^-^”, “OfficeSupplies”, “Production”). F9 provides other methods to maintain and create Financial Entities which apply to discrete segment values.

One method of creating generic account descriptions is to include the financial entities as part of the worksheet, and then append them as qualifiers:

```
Data!B4 -> "Production"
Data!D4 -> "5072"
Data!D5 -> "OfficeSupplies"
```

Finally, in cell Write!D7, add the formula:

```
WDESC(Data!$D5 & " - " & Data!$B$4, BSPEC("^-^",Data!$D4,Data!$B$4),...)
```

Note:

While financial entities can be used to create descriptions, the WDESC function can not be used to create financial entities which are related to the SDESC function.

Account Specifiers

Account specifiers for the Write functions are limited in several significant ways since the functions only write to a single account. This means that:

- specifiers which use ranges and/or wildcards return errors.
- specifiers which use one or more lists will write **only** to the first matching account. The first segment value in each list is used. *No other error message is returned.*

These limitations remove the ambiguity of writing data.

The Write functions can use Financial Entities which exist for the company they are writing to. When you are creating or revising a budget, for example, all of the Financial Entities for that company can be used. When creating a company however, no Entities generally exist and account segment values must be used. The Financial entities used by Write functions must be simple, Natural Entities; they must resolve to simple segment values without wildcards or ranges.

Account specifiers are limited by the source accounting system's rules for the company you are writing to. For example, the F9 Professional GL allows a company to be created with a great deal of flexibility. Many other accounting systems limit the number of budgets, segments and/or characters per segment. The rules of your system apply.

Account specifiers must always be consistent within a company. It is possible to create account specifiers for a Write function which are not consistent with the company's account specifier. If a company is being created using Write functions, the first cell calculated by the spreadsheet will determine the account number style; any account numbers which are calculated later and do not conform to the account number style will fail. If a company already exists, the account number style is already defined and non-conforming Write functions will fail.

Function Parameters

The parameters used in F9 functions are as follows:

Acct	Account specifier (For details, see <i>Specifying Accounts</i> following).
Range	A valid range of cells or the name of a named range of cells. Can also refer to a segment value which uses a range in the form of start..end . See <i>Specifying Accounts</i> below for more on this latter type.
List	A comma-separated list of text for substitution into account specifiers. See <i>Specifying Accounts</i> below.
Period	Period specifier (For details, see <i>Specifying Periods</i> following).
Dates	The GLTRAN function takes two dates as parameters: a From date and a To date. These dates may either be labels or cell references to cells containing either a label or an actual spreadsheet date (<i>i.e.</i> , a Julian date). If you are using a label, the date must use the date format DDMmmYY (or DDMmmYYYY). That is, use a date of the form 21Jan95.
Data	The CODATA and ACCTDATA functions take a string that identifies the data to be returned. What data is available depends on the accounting system you are using. For companies, Data might be “Name”, “Date”, <i>etc.</i> For Accounts, Data might be “Type”, “Description”, “Typical Balance”, <i>etc.</i> Refer to the DLL’s Help for a complete list of supported data items.
Company	Company specifier (this is usually a directory entry, a file name or possibly a company number) — review the help system of the appropriate DLL.
Year	Year to be considered the base year (<i>i.e.</i> , this determines what “this year” means in a period specifier).

Type The account type to access (e.g., this is usually a budget identifier and it determines what the word “budget” refers to in the period specifier.) In GLTRAN, the Type should be “debit” or “credit” or blank (to produce the net of debits and credits).

Text Either text surrounded by quotation marks, or a reference to a cell containing a label. This is used by the WDESC function.

Currency An accounting system specific value to handle multiple currency general ledgers.

Cell values which do *not* begin with a math operator or a number are assumed by your spreadsheet to be text and do not need explicit quotes; if the cell begins with a number or a mathematic operator add either one single quote or a pair of double quotes to the text. For example, when specifying account number 0-0-4550-0, enter “0-0-4550-0”.

Tip:

Use caution when using double quotes: always use pairs. Double quotes are useful because your spreadsheet totally ignores them, but F9 expects them to be paired. Double quotes are only *needed* in the case of financial entities which have hyphen characters within the entity. e.g., "Mortgage - Los Angeles" requires double quotes when submitted to F9 functions.

Refer to the individual function descriptions for the default values and parameters available for each function.

Using Default Parameter Values

Function parameters tell F9 what information you desire. For example, a basic GL formula would be:

	A	B	C	D
1	Company 1			Month 1
2				
3	1000-100-3			=GL(\$A3,C\$1,\$A\$1)
4				

In some situations, it is desirable to use default values. To use the default for a parameter, you do one of two things:

- If you don't need to specify any *following* parameters, just leave it out. [except Lotus 1-2-3 '97].
- If you need to specify *following* parameters, enter a pair of empty double quotation marks then F9 will use its default value for the parameter. Note that a single quote will not work in this instance.

Default values can be manipulated in the Setup windows for each DLL. For example, it is permissible to use the default period of "this period" by:

```
GL($A3, "", $A$1)
```

Or, you may want to compare the balances for an account between this year and last year using the default period and company:

```
GL($A3)
```

```
GL($A3, "", "", "Last Year")
```

If you do not specify an parameter, F9 will either use an available default or return an error message.

1-2-3 '97:

This version (and later, we expect) does not allow variable parameter lists. That means that each parameter must be present *even if it is blank* unlike F9 for Excel, Quattro, or previous 1-2-3. To use the default value for a parameter to any function, you must provide a dummy parameter of "". See also "F9 Functions" in *Appendix D: Using F9 With Lotus 1-2-3*.

Accounts Specifiers

Each accounting system (DLL) will use a different account code(s). Some accounting systems will use a two part code, some a four part code. Some will be all numeric, others will allow a mix of alphabetic and numeric characters. Still other accounting systems allow the account code to be customized on a company by company basis.

This manual will use a four part code in all of the examples that follow. You should use a code appropriate for the accounting system (DLL) and company you are referencing.

You can specify an account parameter in an F9 function in one of three ways:

- Directly enter one or more account specifiers in double quotes into the formula, *e.g.*, `GL("A-1000-100",...)`.
- Enter one or more account specifiers into a cell and reference the cell then reference the cell in your GL formula, *e.g.*, `GL($A21,...)`.
- Build one or more account specifiers from several cells using BSPEC, *e.g.*, `GL(BSPEC(B4, $A21,$B$5),...)`.

Of the three, referencing cells with BSPEC is by far the best method; it allows you to more easily edit and update a report. With Version 4 of F9, we also recommend you use Financial Entities rather than discrete segment values.

Using Single Account Numbers

You can enter single account codes just as in the accounting system. You must supply all parts of the account code. For example, you might enter:

```
GL("100-0-1110-0")
```

or, enter "100-0-1110-0" into cell A5, and use:

```
GL($A5)
```

This may be appropriate for some reports such as a balance sheet where there is little chance of the report needing significant change. For most reports, however, this leads to a situation where the resulting report is inflexible and limited. A more useful style is to use BSPEC to reference a different cell for each segment rather than using complete account codes.

Using Multiple Account Specifiers

F9 provides several tools for specifying multiple accounts. These include:

- Lists of account specifiers. Using BSPEC, lists can be applied at the segment level.
- Wildcards allow selections based on specific characters in a segment.
- Ranges allow selection of consecutive accounts based on a start and end value for a segment.

Note:

Lists actually apply to complete specifiers, but wildcards and ranges apply to segments. To use lists with segments, you must use BSPEC. The BSPEC result allows other F9 functions to generate all of the permutations of the account number using the list(s). See *The BSPEC Function* later in this chapter for more information.

Using Lists of Accounts

You can also enter combinations of the above types of specifiers separated by commas in a function. For example, the following is allowed:

```
GL("0-0-110-0,*--1210..1299-*,*--510*--")
```

The entire combination of account numbers must be enclosed in quotation marks. Once again, cell references are always preferable to entering account specifiers directly into a function. You may enter account specifiers as a comma separated list into one cell, or enter them into a series of cells and refer to the range.

For example, if you entered the following as a label into cell A10:

```
0-0-110-0,*--1210..1299-*,*--510*--
```

then you would use a function such as:

```
GL(A10,"this month")
```

to access the data desired. Equivalently, you could enter the following labels into cells A10, A11, and A12:

```
0-0-110-0
*--1210..1299-*
*--510*--
```

and use a function such as the following to access the same balance:

```
GL(A10..A12,"this month")
```

Using Wild Card Characters

Both the asterisk (*) and question mark (?) wild card characters can be used in an account part. An asterisk means that any combination of characters (including no characters at all) can occupy that position in the specifier. For example: 2*2 matches 22, 202, 212, 222, 232, *etc.* as well as 2002, 20012, *etc.*

A question mark means that any single character can occupy that position in the specifier. For example: 1?1 matches 101, 111, 121, 131, 141, *etc.* It does not match 11, 1231, or 12341 as an asterisk would.

For example:

```
"100-*--41*-0"
```

means match all accounts where part 1 equals 100, part 2 can be anything, part 3 must begin with the digits 41, and part 4 must be 0.

Specifying a Range

When you specify a range of accounts, F9 searches through the chart of accounts for any that match and totals the balances for the period requested. The final total is returned as the result of the function. Any part of the account code may be a range.

To specify a range, separate the first and last parts in the range with two periods. For example, "1110..1150" means match all accounts where the value of that segment falls between 1110 and 1150. When inserted into a complete account specifier, the range is inserted as a segment value. For example:

```
"0-0-1110..1150-100..200"
```

This specifies all the accounts where part 1 is zero, part 2 is zero, part 3 falls between 1110 and 1150 and part 4 falls between 100 and 200.

Note:

Do not use wild card characters in any segment that uses a range. That is, the following is disallowed: "0-0-1110..11*-0" You can use wildcards and ranges within a specifier, but not within a segment. For example:

```
"0-*-1110..1150-100..200"
```

is allowed because the wildcard is in a different segment than either of the ranges.

Using Financial Entities

Financial Entities are names for individual and collected segment values in your accounting systems. For example, many accounting systems have a segment used for “Location” or “Project” which subdivides the chart of accounts. Each value in these segments for location is an “entity” which the accounting system accumulates information about. The F9 implementation of Financial Entities simply allows you to refer to the entities within the accounting system by a name rather than their code.

Tip:

This discussion of Financial Entities is limited to how they are used. For information on creating or editing them, see the chapter *Editing Support Files*.

Financial Entities have several varieties. The summaries which follow are expanded in the following sections of this chapter.

The simplest Financial Entity is referred to as a *natural Financial Entity* which generally indicates the entity was imported from the accounting system as a *subaccount description* for a particular segment value. Natural Financial Entities can be manually created for systems which do not support subaccount descriptions.

The next step in entities are *user defined Financial Entities* which are entities which add new concepts to the system. A user defined financial entity can define:

- an alternate (or new) description for any segment value so your reports become more legible.
- a list of other entities. *e.g.*, an entity of “Texas” can include all the locations in that state.
- a segment specifier with a range. *e.g.*, an entity of “Shareholder Equity” can be defined as “20700..29999”
- a segment specifier with a wildcard. *e.g.*, an entity of “Cash” can be defined as “101*” or “Assets” can be defined as “1*.”

The third and final type of financial entity is a *multiple segment description*. This is useful for a limited number of accounting systems and should be very useful to the few who need it. Essentially, these systems have subaccount segments which change meaning depending on the main account. The SDESC function now allows partial account codes to match several segments to return descriptions.

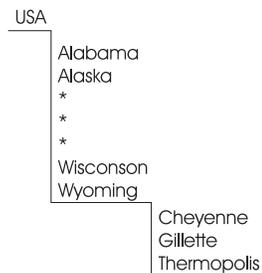
Natural and User Defined Entities

An Entity of "Assets" can be defined for a segment value of 1*. Using this to its logical extent, an entity can be defined for each of the major Balance and P&L sections. Then a spreadsheet can perform a trial balance with either of these calculations:

	GL(*-Assets-*)	GL(*-1*-*)
	GL(*-Liabilites-*)	GL(*-2*-*)
	GL(*-Equity-*)	GL(*-3*-*)
	GL(*-Income-*)	GL(*-4*-*)
+	<u>GL(*-Expenses-*)</u>	+ <u>GL(*-5*-*)</u>
	0	0

This simple example shows how Financial entities can improve the quality of your reports. Financial entities can contain any abstract collection or group of specifiers. For example, in an accounting system where each location has a special value for a segment, then you can define a state as including all the location within the state. But, going another step along, you can then define a series of sales regions which are each a group of states. Now, when a new location is opened, it is added to the state entity's list, and the sales region immediately includes it too.

For example, a hierarchy of Financial entities might be:



Using this model, adding a location in Rawlins, Wyoming, you would:

- Create a Financial Entity of Rawlins which is a specific value of the “location” segment.
- Edit the Financial Entity for Wyoming so it now specifies: Cheyenne, Gillette, Rawlins, and Thermopolis.
- The Financial Entity USA then inherits the members of Wyoming so it requires no changes.

For some accounting systems or charts of accounts, it may be even simpler. If your chart of accounts uses two digits to encode the states (*e.g.*, 01-50), and two digits to encode the city (*e.g.*, 01-99) in a single segment, then Wyoming could simply be 50* in which case all that needs to occur is the creation of one record.

Multiple Segment Descriptions

The last type of financial entity is generally the least used. In some accounting systems, the subaccount has a meaning specific to the account. For example, the subaccount value “100” may mean “Soy beans” for a particular asset account and “Pick-up Trucks” in an expense account. For these systems, F9 now supports multiple segment entities so users can provide the account and subaccount to either GL or SDESC functions and have the segments return useful values.

The following should be observed:

- when using a multiple segment description with SDESC, you must provide more than one segment *and* less than the complete account number and you must provide the segment number of the *first* segment of the entity.
- To use a multiple segment description as a Financial Entity, the segments of the description must be consecutive segments of the account code, and the segments must be in the same order as the account code. *e.g.*, For account A-B-C-D, the description of B-C would be useful as a financial entity, but B-D or D-C would not.
- when using a multiple segment description as a Financial Entity, BSPEC will often require an adjusted TSO mask since two segments of the accounting system are being provided as one

parameter. See *The BSPEC Function* later in this chapter for more information.

Multiple segment entities are user defined. See *Chapter 8: Editing Support Files* for information on editing Financial Entities.

Financial Entity Restrictions

Financial Entities have some restrictions, depending upon the accounting systems.

Financial entities are often sensitive to the default types and years in the F9 DLL Setup dialog. If a default year is setup for a company, the only entities imported for that accounting system will be those accounts with balances in the current year. Similarly, if an account type is specified which has only a few entries (*e.g.*, using the type "Budget A") where only a few accounts have actually been entered, the list of entities will be much shorter than if ACTUAL is the default type and the current or past year is used for the year parameter.

Financial Entities can be divided into natural and user defined. Natural entities are imported from the accounting system as subaccount descriptions and may be very numerous. Some accounting systems do not store descriptions at this level, but do store descriptions for the complete account. There are few, if any, natural Financial Entities in this case. Financial entities *do* allow you to define entities for the missing subaccount descriptions.

Financial entities *should not* use certain characters including hyphens. When an Entity is defined using one of these characters, F9 tries to interpret the Entity quite differently than it does without these characters. For example, F9 considers "Sales - Electronics" to be *two* segments of data rather than one and generates errors about segments being too long, invalid characters, or too many segments. *If* you must use entities with these characters, the strings must be enclosed in double quotation marks -- the single quote used to make labels is not enough.

This will not work in a cell referenced by F9:

```
'Sales - Electronics
```

This value will work correctly in a cell referenced by F9:

```
"Sales - Electronics"
```

Tip:

Use caution when using double quotes: always use pairs. Double quotes are useful because your spreadsheet totally ignores them, but F9 expects them to be paired. Double quotes are only *needed* in the case of financial entities which have hyphen characters within the entity. e.g., "Mortgage - Los Angeles" requires double quotes when submitted to F9 functions.

Referencing Cells—The Preferred Method

The most efficient way to specify accounts is to reference a cell or range of cells containing account specifiers. Using a single cell or two to specify accounts allows several columns to contain GL functions and use the account in several ways (e.g., comparative or multiperiod reports). The single list of accounts is certainly less error prone than trying to use the same specifier several times in a spreadsheet.

As well as providing a visual reference point, referencing existing cells minimizes the number of changes to the worksheet that will be required if you want to use the same worksheet for reporting on different accounts, departments and/or companies.

You can use absolute, relative and mixed cell referencing. As well, you may use the concatenation operator to concatenate two cells. This is especially useful for developing period specifiers. See your spreadsheet manual for details of the referencing methods available.

The next level of control for a spreadsheet comes when you use BSPEC and cell references.

The BSPEC Function

The GL function has only one parameter for the account specifier. This is often inconvenient. For example, if you wished to construct a departmental income statement where each column in your report is a separate department, having only one parameter for the account specifier creates a problem: the account specifier must be assembled *before* an F9 function can use it.

If we apply the usual logic to building the report, part of the account specifier will be in one cell (usually the “natural” account code in a cell on the same row and to the left) and part will be in another cell (the department number at the top of the column). This is where the BSPEC function comes in to play.

The BSPEC (**B**uild **S**PECifier) function has one primary role: **it lets you use cell references for each segment of the account specifier.**

This can now be replaced with the more flexible:

```
GL(BSPEC($A10,$C$2,$B10),"This Month")
```

BSPEC([TSO Mask,]Segment 1[, Segment 2 [, ...Segment N])

Using this function you can:

- use a separate cell for each of the segment values.
- specify any single segment value only once on a spreadsheet.
- specify a financial entity.
- specify a range for any segment.
- specify a comma separated list of segment values (lists of values is a major feature of BSPEC).
- specify multiple specifiers in a cell range.
- use wild card characters in any account part.
- almost any combination of the above.

BSPEC adds considerable power to your specifiers. The BSPEC function also manipulates strings — specifically account specifier ranges and substitution lists. Its sole purpose is to allow you to construct account specifiers *using cell references* — typically one cell reference for each segment of the account specifier. For example, if you wish to create a column that summed two departments, all you would have to do is change the contents of a cell from “100” (for example) to “100,150”. BSPEC successfully permits lists as a segment value into F9 formulas.

Another example of how BSPEC makes your reports more efficient is seeing multiple lists in action. Suppose you are using a four part account code. The parts are the location, department, account and subaccount. You want the balance for two locations (A, and 7), for three departments (100, 225, and 777), for two products (accounts 4001 and 4127) for three sales representatives (sub-accounts 202, 305, and 666) to appear as a single number in your sheet. Because of the “choppy” nature of this request (that is, the accounts, sub-accounts, locations, *etc.* are not easily represented using ranges or wildcards), you would normally have to list out each of the accounts you need. The complete list of $2 \times 3 \times 2 \times 3 = 36$ accounts would be:

```
A-100-4001-202
A-100-4001-305
A-100-4001-666
A-100-4127-202
```

A-100-4127-305
A-100-4127-666
A-225-4001-202
A-225-4001-305
A-225-4001-666
A-225-4127-202
A-225-4127-305
A-225-4127-666
A-777-4001-202
A-777-4001-305
A-777-4001-666
A-777-4127-202
A-777-4127-305
A-777-4127-666
7-100-4001-202
7-100-4001-305
7-100-4001-666
7-100-4127-202
7-100-4127-305
7-100-4127-666
7-225-4001-202
7-225-4001-305
7-225-4001-666
7-225-4127-202
7-225-4127-305
7-225-4127-666
7-777-4001-202
7-777-4001-305
7-777-4001-666
7-777-4127-202
7-777-4127-305
7-777-4127-666

This would be a difficult list to enter, you would run the risk of overflowing F9's internal buffers, and the risk of error is high. The same accounts can be accessed using the following account specifier:

```
(A,7)(100,225,777)(4001,4127)(202,305,666)
```

which can be returned from BSPEC with:

```
BSPEC("A,7","100,225,777","4001,4127","202,305,666")
```

or better, from a BSPEC which references four cells with those values.

Tip:

By using the TSO and BSPEC functions together, you can build any account specifier imaginable. Adding the power of Financial Entities makes your formulas faster, more understandable, and more powerful.

To use the full power of BSPEC, if your company used a four segment account mask:

^_^_^_^

and cells E1, and E2, are a named range called "Locs" and contain:

A
7

and cells F1, F2 and F3 are a named range called "Depts" and contain:

100
225
777

and cells G1 and G2 are a named range called "Accts" and contain:

4001
4127

and finally, cells H1, H2 and H3 are a named range called "SalesPersons" and contain:

202
305
666

then this BSPEC function:

`BSPEC(Locs,Depts,Accts,SalesPersons)`

would return the string:

`"(A,7)(100,225,777)(4001,4127)(202,305,666)"`

This string is a valid account specifier consisting of four corresponding substitution lists to be used a GL function such as:

`GL(BSPEC(Locs,Depts,Accts,SalesPersons),"this month")`

When used this way, the GL function provides the correct (four segment) TSO mask for the company. The result of this GL function is the same as if you had used the same list of 36 account specifiers listed above. And now, if you want to add a salesperson, *edit one cell*. When you consider that all F9 reports can easily use hundreds of accounts, the power of cell references becomes very clear.

Note:

The important thing to note here is that each segment of the account specifier is now a separate cell reference.

In our example, we used separate cells for each segment value (*e.g.*, the sales persons requested were in cells H1, H2 and H3) and then referred to the cell range. Usually it is more convenient to list the values desired in a single cell (*e.g.*, cell H1 could have contained the string “202,305,666”) and then just refer to that one cell, rather than a range.

The Text Substitution Operator

The combination of the TSO and the BSPEC function can make the creation of complex account requests easy. To use the text substitution features of F9, you must supply account specifiers with TSOs inserted at appropriate places, and one or more substitution lists of text to be substituted into the specifier.

Version 4:

The TSO is now an optional parameter of the BSPEC function (described in this chapter). This means that the majority of our users may never need to provide a TSO again. Some of our users can still benefit greatly from using a TSO, and all users should read this the following discussion. See *An Illuminating BSPEC Case Study* later in this chapter for why you may still want to provide a TSO.

Tip:

The TSO can be queried using the CODATA function using the keyword “mask”.

The following account specifiers are valid and equivalent parameters to GL or SDESC:

```
A-100-4000-200
(A)(100)(4000)(200)
^_^_^^(A)(100)(4000)(200)
```

The first example is a “normal” account specifier. The second is the result of BSPEC *without* providing a mask parameter, and the last is the result of BSPEC with a mask parameter. The mask for the second usage is implied by the company specified.

What should be clear is that an account specifier can consist of a mask (“^_^_^”) and a set of values to be substituted into the mask, one set per TSO. F9 resolves this internally into useful account specifiers.

The values in each of the lists (called substitution lists) are substituted into the TSOs (carets) in turn producing a kind of multiplier effect. The first substitution list “(A,7)” is associated with the first caret; the second list with the second caret, *etc.* Each substitution list is surrounded in parentheses. In this case, the list of lists is surrounded by parentheses as well, although this is not strictly necessary. This specifier will cause F9 to generate the exact same list of account specifiers

as you see above and, of course, F9 would ultimately produce the same balance. There should always be as many lists as there are TSOs. A list may be only one item long. That is:

$$\wedge-\wedge-\wedge-\wedge(A)(100,225,777)(4001,4127)(202,305,666)$$

and

$$A-\wedge-\wedge-\wedge(100,225,777)(4001,4127)(202,305,666)$$

are equivalent. In these examples, the carets represent full parts of the account code, but that is also not necessary. For example, the following is legal:

$$A-\wedge-4-\wedge(100,225,777)(001,127)(202,305,666)$$

You may also use ranges and/or wildcards in either the specifier, or in the substitution lists. The following is legal:

$$A-\wedge-4-\wedge(2^*)(001,127)(202..208,305)$$

Using the TSO facilities of F9 in conjunction with the BSPEC function virtually eliminates the need to use the string concatenation operator(&) to build account specifiers. The BSPEC function is used to build account specifiers that use TSOs.

It is your responsibility to ensure that the account specifiers that are created using F9's TSO functionality are acceptable to the accounting system being accessed.

The first parameter to the BSPEC function is the optional "mask". The mask contains TSOs, usually one per segment of the account code. The remaining parameters of the BSPEC function are substitution lists, one per TSO in the mask. In general, as with all F9 functions, the parameters to the BSPEC function will be cell references. **We strongly recommend that you use the BSPEC function with all your GL functions.** Doing so will ensure that your spreadsheet reports are as flexible as possible.

Note:

Do not create named ranges or macros with the same names as F9 functions or commands. They may have the effect of disabling the function or command.

See the example sheets shipped with F9 for examples of TSOs and BSPECs in action. The Report Wizard also produces GL functions that use the TSO and the BSPEC function.

An Illuminating BSPEC Case Study

Susan is the marketing manager of Best Way Furniture Corporation. Best Way constructs a full line of office and home furniture with factory/stores in Vancouver and Montreal. Arthur Dent is her accountant. Arthur and Susan use a popular PC based accounting system to manage the business. Years before, they had constructed an appropriate chart of accounts for the GL in order to model their company. The chart of accounts had, of course, the usual balance sheet accounts for assets and liabilities and the profit and loss portion of the chart of accounts had accounts defined for each furniture product that Best Way produced.

Susan had a meeting with Arthur to discuss ways that they could make their general ledger work better for them. Specifically, Susan had determined that there were a number of key variables about their products that she thought would be useful in analysing product sales. They were:

- Color;
- Covering material (leather or fabric);
- Location (Vancouver or Montreal);
- Construction material (*e.g.*, steel or wood);
- And if the purchase was for home or office use.

Currently, they had one account for each of the thirty products they produced. Her thought was to use the subaccount of their two segment account code to encode the above information, using one character in the code for each of the items above. This way, she could then ask Arthur to create various reports such as a report on color by location for a specific product, or for all products.

Arthur was reluctant. He explained that while yes he could create the accounts in question (some 3,800 accounts might be required for every possible combination) and post to them, but getting the reports that Susan wanted would be extremely expensive. Certainly, the accounting system's report writer would never be able to do it. The only way that Arthur could think of to get the report described above would involve importing all the P&L accounts into a spreadsheet, sorting the entries, and then using sum functions to create the 20 numbers that would be needed for the report cited above. But each of those numbers would be the sum of 16 balances. Each report might take hours to create and each would have to be recreated from scratch each time because new accounts and occasionally new products are added to the general ledger, throwing off the cell references.

The proposal was abandoned.

Some months later, Best Way purchased F9 and started using it to develop a suite of financial reports. F9's reports are developed in the spreadsheet and are essentially "set and forget". After setting up all the usual financial reports, they read deeper into the F9 manual and started reading about the TSO (Text Substitution Operator) and how it worked with the BSPEC function.

Specifically, they discovered that using a TSO mask of "^-^" (for a two segment account code) was not the only option. The mask can be used to in effect artificially segment a segment of an account code. That is, using a TSO of the form:

^_^^^^

would allow them to treat the second segment of the account code (the subaccount) as if it consisted of multiple segments. The BSPEC function can then use a separate cell reference for each individual character (in our example above) of the subaccount.

Susan and Arthur suddenly realized that the reporting and analysis goal that they were seeking was now within their grasp. Using F9 with a TSO as noted above, they could construct the color versus location report for one or many products easily. The report would be "set and forget". It would automatically pick up every new account as it was added. In fact, any analysis of sales by color, material, location, *etc.* would be easy. All they would have to do is place, as either row or column headers, the individual characters for the attribute they were interested in (just like they would with a Period Specifier). For example, they might put as column headers "V" for Vancouver and "M" for Montreal. Then they would reference those cells in their BSPEC function. Unlike the BSPEC functions they used for their more prosaic F9 reports which had three cell references (one for the mask and one for each segment of the account code), their new BSPEC functions would use the five-caret TSO mask illustrated above and have six cell references: one for the mask, one for the main account code segment, and one for each character in the segmented subaccount.

In fact, they realized that with F9 available to answer the questions, there were many possible ways they could enhance their chart of accounts to model their company in more detail. Of course, one has to be a little cautious here. In our example above, each character had only a few unique values (*e.g.*, "V" for Vancouver and "M" for Montreal). If the number of characters and the number of unique values for each character is raised only a little, the number of possible unique sub-accounts, and therefore accounts in the chart of accounts, can quickly explode into the tens or hundreds of thousands.

The point is that there is no point in gathering data, in this case as sales breakdown information in the chart of accounts, if there is no practical way to query that data and improve your business. With F9 in place, Best Way was able to turn their GL into more than just a source of financial information, but a sales and marketing database as well, all using the tools they already had: F9 and their favorite spreadsheet.

Period Specifiers

You can specify periods in F9 functions either by:

- entering a period specifier directly in the function;
- referencing a label cell or range of label cells that contain the period specifier.

Period specifiers use a subset of the English language to describe a value in the general ledger. You can specify the length of the period for which you want a value, the type of value you want, and several other parameters, all using standard accounting terms. Each parameter has a default value, so you only need to specify those parameters which deviate from the default.

You can enter capitalized or lower case characters. Also, any period specifier keyword can be abbreviated as much as you want, as long as it is still unique among the recognized keywords. For example, you could not use “b” for “budgeted” because “b” could also mean “balance”. You could, however, use “bu”, “bud”, “budget”, *etc.* for “budgeted”, since there are no other specifiers that start with these sequences of characters. This is the reason why many of the keywords, such as “months”, are in plural form. You can use either the plural or the singular, since the singular is just an abbreviation for the plural. If a period is not specified F9 defaults to “balance for the current period of this year.”

The Period Wizard

The easiest way to explore period specifiers is to use the Period Wizard. A **Period** button which invokes the Period Wizard is available in the Chart and Drill windows. When you invoke the Period Wizard, the following window will appear.



Period Wizard

Period Specifier: This Month

Balance Type

Actual

Budget

Year

This Year

Last Year

Values

Opening Balance

Ending Balance

Change

Duration

Month

Quarter

Half Year

Year

Running

Period

1 5 9 13

2 6 10 This Period

3 7 11 Last Period

4 8 12 All Periods

Ago

From: 01Mar1997 To: 31Mar1997 To Clipboard OK

Dates above used for Transaction Drilldown and by CoData only.

The Period Wizard displays all the possible allowable combinations of period keywords. (Some DLLs support special accounting system specific key words. Refer to the DLL help for additional allowable period keywords.) After selecting the options desired, you may send the period specifier(s) to the Clipboard, or press OK, in which case the data will go to the period specifier of the current window.

Periods Quick Start

The following discussion assumes 12 or 13 periods per year.

- By default, F9 produces balances for balance sheet accounts and transactions for income statement accounts.
- For P&L accounts, if you want an amount for a period other than a month, use the keywords “quarter”, “half year” or “ year”. **The word “balance” has no meaning for a P&L account.** For example, if you want the total income for the year (say up to month 6), use the period specifier “year to date month 6”.
- For an income statement account, transactions and balances are equivalent. Thus, the keyword “transactions” is useful only to produce changes in balance for balance sheet accounts.
- Using the keyword “running” causes F9 to ignore year boundaries. Thus, “running quarter” would sum the current month and the preceding two months whereas “quarter” would sum up to three months of the current fiscal quarter.
- The keyword “running” **always** implies transactions.
- The keyword “year” causes F9 to use the entire year up to and including the present period unless another period is used to override.

Some examples of period specifiers and their interpretations are:

this month	⇒	<u>Balance Sheet Account</u> : balance for this month <u>P&L Account</u> : transactions for this month
prior month	⇒	balance or transactions for last month
month 5 change	⇒	transactions for month 5
budget month	⇒	transactions or balance for budgeted account this month (if available)
last year	⇒	balance at the end of, or total transactions during, the whole of last year

last year this month	⇒	balance or transactions for this month last year
running quarter month 6	⇒	total transactions during the months 4, 5, and 6
running halfyear month 3	⇒	total transactions for periods 10, 11, and 12 of last year plus periods 1, 2, and 3 of this year
year	⇒	<u>Balance Sheet account</u> : the balance for this year, this month <u>P&L account</u> : the total transactions for this year to date
year to date last year	⇒	balance or transactions for all of last year

In addition to the keywords listed below, F9 recognizes at least the numbers 1 through 13 as referring to periods 1 through 13. Note that in the following lists, the default keyword is in **boldface**. For example, if you don't specify a period type specifier (below), the type will default to **periods**.

Time Specifiers

Use these keywords to indicate which period you want to take data from, if it is not the current period. Note that in the following, *period* is month, period, quarter, halfyear, or year.

<i>period</i>	⇒	the current calendar period of size period
last <i>period</i>	⇒	the period before the current one (<i>i.e.</i> , "last year" or "last month")
<i>n periods ago</i>	⇒	n periods before the current one, where n is 1 through 13
<i>period n</i>	⇒	period number n in the fiscal year, where n is 1 through 13
<i>January, February</i>	⇒	the names of the months may be used to specify periods
running <i>period</i>	⇒	the last period, regardless of calendar boundaries (for example, in month 11, "quarter" would give you months 10 and 11 to date, whereas "running quarter" would give you months 9, 10, and 11 to date)

Tip:

Notice the difference between specifiers in the forms of "month 1" and "1 month." "1 month" *implies* the modifier "ago" and is relative to the current fiscal *period*. "Month 1" is the first period of the current fiscal year.

Duration Specifiers

Use these keywords to indicate to F9 the size of the period for which you want data.

periods	⇒	the length of time your company uses as its base fiscal period
months	⇒	a period of time one month in length
quarters or qtrs	⇒	a period of time three (or four) months in length
halfyears or semiannually	⇒	a period of time six (or seven) months in length
years or yrs	⇒	a fiscal year, regardless of the number of periods it comprises annually

Future Specifiers

Many accounting systems allow transactions to be posted to “future periods”. This happens when transactions are posted to the first (or later) period of the next fiscal year, before the current fiscal year has been closed. F9 handles this by treating these future periods as periods 14, 15, *etc.*. These periods can normally only be accessed absolutely and individually. That is, F9 does not support periods specifiers such as “running quarter period 15”. As well, “this month” and other relative period specifiers always refer to periods between 1 and 13. Thus, to access period 14, you must use a simple period specifier such as “month 14”.

Value Type Specifiers

Use these keywords to indicate to F9 whether you want an account balance or the transactions for an account.

balance	⇒	balance at a certain point in time in the period
transactions or change	⇒	net transactions (change in balance) for the period

Balance Time Specifiers

If you are requesting an account balance from F9 (see above), use these keywords to indicate whether you want the balance at the start or end of the specified period.

ending or closing	⇒	balance at the end of the period (current balance if the current period)
starting or opening	⇒	balance at the beginning of the period
beginning or running	⇒	change over requested period without regard to year end.

Period Lists

The GL function also supports lists of period specifiers. For example, the period specifier “3,4” would return the sum of periods three and four. Period lists are a convenient way of getting data for an unusual duration.

The following cautions should be observed:

- Do not use a period list unless absolutely necessary. For example, rather than using a period specifier of “1,2,3”, you should use “year to date month 3” (or “quarter 1”). The latter will execute **much** faster.
- If you request periods one and two for a balance sheet account, you will get the sum of two closing balances.; this figure is largely meaningless. Typically, you should only use period lists for transactional values, which will usually mean P&L accounts.
- When listing period specifiers, do not combine actual requests with budget requests. The results of such a request are not defined (that is, F9 does not support this functionality).

Budget Specifiers

Use these keywords to indicate whether you want to see an actual account value or a budgeted value.

actual	⇒	the actual account value for the period
budgeted	⇒	the budgeted account value for the period
Budget Year	⇒	the total budget <i>up to the current period</i> for the current year
Budget Year to Date Month 12	⇒	the total budget for the year
Budget Last Year	⇒	Invalid. Accounting systems don't normally associate “budgets” with years

Year Specifiers

You may want to take data from last year as well as the current year. If so, use these keywords.

year	⇒	the current fiscal year
last year	⇒	the fiscal year before the current one
previous year	⇒	1 year ago

Additional Keywords

These keywords do not carry any special meaning, and they are ignored by F9. They can be useful when you want to make an F9 period specifier read more like natural English.

at	before	current	date
for	in	net	of
on	present	the	this
to	total		

Example Period Specifiers

net change for the current month	3 running quarters ago
quarter to date	year to date last year
halfyear to date	this quarter last year
transactions this month last year	year to date
month 2	month 3 last year
quarter 1 this year	transactions last month
last year last month	transactions last quarter
last year last halfyear	last quarter balance
running halfyear last year	current balance
current budgeted transactions	current budgeted balance
balance 3 months ago	balance in month 5 last year
budget for year to date	running quarter
budgeted balance end of quarter 3	year to date for month 5
opening balance for the quarter	last running halfyear
running quarter period 5 last year	

Company Specifiers

A company specifier is usually a sub-directory name or a file name. See the DLL's help for a description of company specifiers for the DLL you are using. You can specify a company in an F9 function in one of three ways:

- specify no company, in which case the default company is used.
- specify a single company.
- specify a list of companies.
- specify a range of cells containing company specifiers.

A given company specifier can be referenced within the F9 function by either placing the company specifier(s) between double quotation marks or by referencing cell addresses. Of the two, referencing cells is by far the best method.

Using Single Company Specifiers

Single company specifiers can be used in all F9 functions. For example, to obtain the account type for account 0-0-1000-0 and company "ABC", enter:

```
ACCTDATA("Type", "0-0-1000-0", "ABC") .
```

As always, using cell references rather than literal parameters is preferred.

Specifying Multiple Companies

This option can only be used within the F9 functions GL and NGL. When you specify more than one company, F9 searches through each chart of accounts for any accounts that match and totals the balances for the period requested. The final consolidated total will then appear as the result of the function.

The F9 *Return Zero for Account Not Found* option can have some effect on the return value where not all the accounts within a specified list necessarily exist within every company indicated. If this option is not on, and one or more of the account specifiers within the function do not exist for all the companies specified, then F9 will return an error message. If, however, this option is on, F9 will always return an accumulated balance for all accounts it could find within each of the companies listed. Thus, it is not necessary to have identically defined charts of accounts for all companies to generate consolidating reports.

If F9 is unable to locate or access a company's data files within a multiple company specifier, an appropriate error will still be returned indicating the problem.

You can enter company specifiers separated by commas or blanks in GL and NGL formulae.

For example, entering:

```
GL("0-0-1100-*", "Month 5", "comp1, comp2, comp3")
```

would bring into a single cell the sum of the balances for accounts matching 0-0-1100-*, within month 5, for three companies. The entire combination of company specifiers must be enclosed in quotation marks. To specify a cell range, reference the range of cells containing company specifiers within the formula as follows:

```
GL("1110-*", "Current Month", B$3..E$3)
```

In this example, cells B3, C3, D3, and E3 would contain single company specifiers. By specifying a range of cells, F9 will find all account balances possible in each of the company chart of accounts and return with the accumulated balance.

Some accounting systems support consolidations and maintain consolidation "trees" of companies that indicate how a consolidating company is calculated from various "leaf" companies. For the accounting systems, F9 can read these trees and do the consolidation automatically. To have F9 calculate consolidation balances, precede the company identifier with an exclamation mark. That is, a company specifier of the form "DEMO" will pull balances from company Demo, but a specifier of the form "!DEMO" will calculate balances from the GLs of the companies that make up demo. Of course, this is a more expensive operation and may take longer to calculate.

Year Specifiers

If you wish to access data from some year other than the default base year as set in the DLL Setup window, you may supply a year (all four digits) specifier. The year specifier is only used in GL and NGL functions. The base year determines which two years of data the period specifier will access.

The following function will access month 2 for the year 1995, regardless of the default year in the Setup window. If a period specifier such as "month 2 last year" were used, it would refer to the year 1994.

```
GL("1110-*", "month 2", "Comp1", "1995")
```

To default the company to the contents of the Setup window, enter:

```
GL("1110-*","month 2","","1995")
```

Type Specifiers and Budgets

The Type provides other data about the accounts you wish to access. The Type field may have different meanings, depending on the accounting system you are using and on which function you are using. Usually, the Type is a budget identifier. If you wish to access budget data from other than the default type asset in the DLL Setup window, you may supply a budget specifier. This specifier is only used by the GL, NGL and DESC functions.

For example, the following would get budget data for the “Pessimistic” budget for the default company and year:

```
GL("1110-*","month 2 Budget",,"","Pessimistic")
```

In the GLTRAN function, the Type should be “Credit” (for credit totals only), “Debit” (for debit totals only) or blank for the total of debits and credits.

Tip:

Though all budgets are addressed as “this year,” the budget specifier for most accounting systems allows you to select the correct budget for any fiscal period. *i.e.*, if a budget “23” or “pessimistic” is known to be for the 1998, using that specifier in any other year returns the same numbers.

Currency Specifiers

If your accounting system supports multiple currencies, then you may indicate the currency desired here. Currencies are typically indicated in the same way as in your accounting system (*e.g.*, the F9 Datamart allows a code of up to four characters, such as “USD” for US dollars, or “CAD” for Canadian dollars).

Specifying the Topic

Unless you are using more than one accounting system, you need not concern yourself with this parameter.

The last parameter to every @function is the topic (or, equivalently, the accounting system or DLL). If you are only using one accounting system with F9, you may omit this parameter. In this case, the add-in will use the topic “F9”. This special topic instructs the Server to use the default topic.

If, however, you have access to more than one DLL, you may wish to supply the topic explicitly. If you wish to retrieve data from more than one accounting system into the same sheet then it is required that each function supply an explicit topic.

The following function retrieves data from an F9 Professional database, regardless of the default topic selected:

```
GL("1110-*","month 2","","","","Univ")
```

Chapter

6

6. Commands and F9 Windows

The F9 Server is the interface between clients (usually add-ins for Excel, Lotus 1-2-3 for Windows, or Quattro Pro Version 5 for Windows) and one or more general ledger applications. Each general ledger is accessed using a specific DLL (Dynamic Link Library).

This section documents the basic Server commands. For typical F9 installs, **it is not necessary for you to understand the relationship between the Client, Server and DLL.** Feel free to skip this section and go directly to *DLL Commands*.

Each DLL is known to the Server by its file name — and this DLL file name is also the “topic” parameter of an F9 function.

This chapter describes the windows and menu items available for the F9 Server and its DLLs. In most cases, the Server simply delegates requests to the DLL. For example, if you select Setup from the F9 menu in your spreadsheet, the Server will ask the appropriate DLL to present its Setup window.

The F9 Functions are also performed almost exclusively by the DLL but no windows appear; a result is placed in the spreadsheet cell. The F9 functions and their respective parameters are discussed in detail in *Chapter 5: F9 Functions*.

Note:

Not all Setup windows are identical, nor do all DLLs support all functions. However, in most cases the variations are slight. This chapter describes functionality that is common, in one form or another, to all F9 DLLs. For specific information about your DLL, select DLL Help from your spreadsheet's F9 menu, or the DLL window's Help menu.

Each DLL has a Help command on its main menu. Select Help to review DLL (and therefore accounting system) specific information. The DLL's help system will document differences between the functionality of F9 as documented in this manual and the functionality offered by the DLL.

Server Commands



The F9 Server can be invoked either by double-clicking on the F9 icon, or by selecting the appropriate command in your spreadsheet or another DDE client. When in the Server, you may select any of the following commands:

- **Topic** -- Sets the Default Topic and opens the DLL menu.
- **Default Topic** -- opens the DLL menu.opens the DLL menu.
- **Quit Server** -- Forces the Server to quit.
- **Help** -- Opens the common help for F9.

Generally, you will only need to access the Server directly to perform configuration tasks or if you wish to use F9 to fill the Clipboard with data for your spreadsheet or some other application.

When a spreadsheet add-in is loaded, it begins a conversation with the server. When the spreadsheet is closed, the add-in ends its conversation with the server. The server maintains a count of conversations while it is active and will automatically close when the number of conversations drops to zero. F9 normally runs minimized and this has no effect on F9 processing.

The Topic Command

If you select Topic, the main menu for the appropriate DLL will appear (if there is more than one DLL, you will be presented with a list). The spreadsheet add-in sends a “Topic” command when it is loaded.



The main menus for each DLL vary slightly from one to another and the next portion of this manual gives greater detail these commands. Most F9 DLLs will have the following menu items:

- **Setup** -- Opens the configuration dialog to direct F9 to your accounting system and set the defaults.
- **Chart** -- Opens a window to work with your Chart of accounts. (We recommend “lists”).
- **Drill Down** -- Opens the drill down window to examine the last GL formula.
- **Budgets** -- Opens the Budget Write Back window.
- **Lists** -- Opens a window that can provide lists of specifiers such as segment values.
- **Transactions** -- Some F9 implementations can now prepare transaction batches for submission to data-entry.
- **Utilities** -- This menu item collects several commands which are less used, such as:
 - **Account Enquiry** -- Opens a window which allows you to quickly and easily query a GL amount and/or configure the GL Paste command.
 - **Passwords** -- *If* security is implemented in your accounting system, this will allow you to enter passwords and gain access.
 - **Maintain Financial Entities** -- Opens a dialog to create, edit or delete F9 Financial Entities.
- **Help** -- Opens the Help file for the current DLL (*i.e.*, covers F9 support of your accounting system features).

The Default Topic

You only need concern yourself with this option if you have more than one accounting system DLL. When a DDE request is sent to the Server and no topic is specified, the Server will use the client default topic. If there is no client default topic, the Server will use the Server default topic. If desired, you can change the Server default topic by selecting this menu item.

Quit Server

Unless you are sure that you want to terminate the Server, do not use this command! If you do, F9 will stop working and your sheet will be filled with DDEML errors.

However, if the need arises, you may force the Server down by selecting this option. Cases where you may wish to force the server to close include when your spreadsheet has closed from an application error.

Selecting this option while there are active conversations taking place will almost certainly generate errors (usually DDEML errors) in applications that use F9. In general, you should not need to use this option.

Help

Select this option for help on using F9 spreadsheet add-ins and the Server. The contents of the principle help file are for a “generic” F9 DLL and the features supported by virtually all accounting system DLLs.

Tip:

After selecting a Topic, the DLL window has an option to view the DLL’s Help file for details about differences between the current DLL and the “generic” F9 DLL. The current DLL is always accounting system specific.

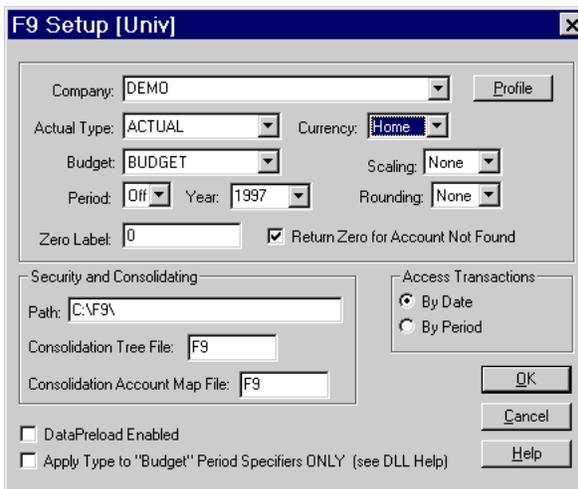
DLL Commands

Each DLL has a main menu containing a number of commands. Each command invokes an appropriate DLL window. At a minimum, each DLL has the following commands on its main menu: Setup, Drill Down, Chart, Lists, Budgets, and Help. Some DLLs will have additional commands. These are documented in the DLL's help file. The DLL windows can be accessed in one of two ways:

- by bringing up (*e.g.*, maximizing) the Server (which will cause the DLL menu to appear) and selecting the appropriate menu item to invoke the desired window;
- by selecting the appropriate command on the F9 drop-down menu in the spreadsheet.



The Setup Window



Use the Setup window to:**Specify the location of your accounting data;****Control how general ledger data is returned to your worksheet;****Change function defaults.**

When you select Setup from the DLL menu, or through a client add-in command (or by sending the DDE [Setup] command from a DDE compliant application), a Setup window will appear. The Setup window contains all the site-specific setup information for the selected DLL (*i.e.*, for the selected accounting application). Each field in the Setup window is described in the following text.

System Directory: Many DLLs will require a “System Directory.” Usually the System Directory is combined with the Default Company to create a complete path or location for the general ledger data desired. This is usually the first (and often only) field you will need to set up in order to configure F9 to your system. Review the DLL’s help for specifics about the System Directory. (Not shown in the above illustration.)

Company: All DLLs will have a default company (or set of books). This default takes different forms, depending on the accounting system. For example, the F9 Professional GL uses either a fully qualified file path, name and extension *or* a logical Datamart name; other accounting systems identify companies by an eight character directory name; some use a number such as “12”. If an F9 formula does not specify a company, the default company specified here is used. Most F9 setup dialogs use a combo-box to allow you to select from a list of available companies. Review the DLL’s help for specifics about the Default Company.

Tip:

The Profile button in the Setup Window should be used to test the association with the default company. If you press this button, a display will appear showing basic company information. At a minimum, the company description will appear, along with the number of fiscal periods, and the current fiscal year and period.

Year: Enter the year you wish to make the current or default year. The year entered here is used by functions as the default. Period specifiers in F9 GL functions allow access to “this year” and “last year” data. The default year determines which year “this year” is. An incorrect year may result in “no matching accounts found” error messages. For many DLLs, setting the year to zero will set the year to the current year. If the accounting system (and therefore the DLL) only supports two years of data, the Year item will not appear.

Type: This is the account type. This parameter contains other qualifying information about a GL or DESC request. Usually this is the budget value desired. Functions that refer to budgets will use this budget. If the accounting system only supports one budget, this item will usually not appear.

Period : In most F9 setup dialogs, you can override the current period (*i.e.*, the accounting system's current period) with a different value. The current period determines which period(s) are accessed when you use period specifiers such as "This Month" or "Running Quarter." Select "Off" to use the accounting system's current period.

Value Scaling : Select the level of scaling desired. No precision is lost with the use of this option. If you select None (the default) then values are returned with the same magnitude they have in the general ledger. For example, 19,551.02 is returned as the value 19,551.02.

If you select any other value, numbers returned will be divided by the selected amount. For example, if you select 1,000, then 19,551.02 will be returned as the value 19.55102. If the numbers in your worksheet do not seem to add up correctly, check the display options currently in use with your spreadsheet. Values are rounded first (if you are using rounding) and then scaled.

Value Rounding: Select the level of rounding desired. If you select None – the default – then returned values are not rounded. Otherwise, numbers will be rounded to the nearest value selected. Values are rounded first and then scaled (if you are using scaling).

Zero Label : F9 defaults to returning the number zero when a zero value is found. You may enter a different zero label to be returned if desired. For example, you may want to have "n/a" or "-" appear in a report instead of the number 0. This command only affects the return of zero values returned by the GL and NGL functions. To return to displaying zero, clear the alternate label and enter the digit zero. Only the special value "0" will return the number zero.

Return Zero for Account Not Found F9 defaults to returning an error message when no matching accounts are found. You may have F9 return zero when no matching accounts are found by selecting this option.

For example, you may want to have the number 0 appear instead of an error message when generating the same report for a number of companies with different Charts of Accounts. If possible, you should build your sheets with F9 returning error messages when accounts are not found. Otherwise it may be difficult to tell when you have entered an erroneous account specifier.

Please note that this option only affects the display of values returned by GL and NGL functions.

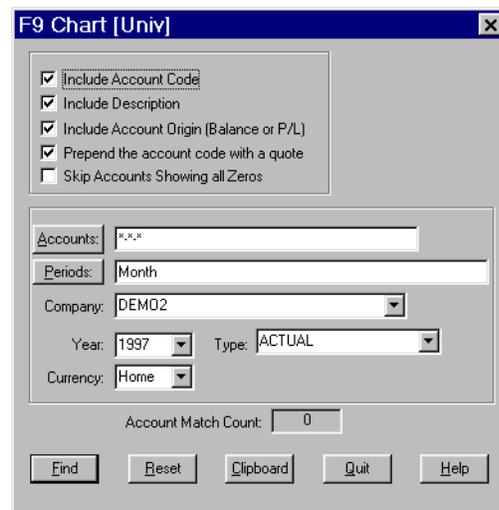
DataPreLoad Enabled: Use this to enable or disable automatic DataPreLoad. Only automatic DataPreLoad is disabled when this option is turned off. Manual DataPreLoad is always available.

Tip:

DataPreload is highly memory intensive: do not use this option without reading the notes in the chapter *Advanced Features*. While DataPreload will always improve performance with some accounting systems, and can improve the performance of others, it can also slow down F9 and your spreadsheet, as well as other applications.

Other Setup Options Other Setup Windows options specific to the accounting system you are using will be described in the DLL Help file.

The Chart Window



Use the Chart window to:

Review your chart of accounts;

Create a trial balance;

Examine balances for individual accounts.

When you select Chart from a DLL menu, the DLL will invoke the Chart window. Like the Setup window, the Chart window is virtually identical from one DLL to the next. This section will describe the “generic” Chart window and its functionality. Review the DLL’s help for specifics about the Chart window for your accounting system.

Tip:

It is not necessary to use Chart to create reports. The GL and NGL functions are best for this purpose.

The Chart window allows you to specify the accounts and the account information to copy to the Windows clipboard, which you can then paste into your spreadsheet (or any other Windows application). It is appropriate to use the Chart window to create a trial balance. It is **not** appropriate to use Chart to dump your GL into a spreadsheet and then create reports by referencing the various cells containing balances. Once again, use the GL function for this purpose.

There are several check boxes that allow you to select several options. All DLLs will allow you to **Include Account Code** and **Include Description**. Check boxes are usually provided for two other options:

Select **Prepend the Account Code with a Quote** to have F9 place a leading quote in front of the account code. This is sometimes necessary when your account code is all-numeric and the data is to be pasted into a spreadsheet. For example, it is possible for the spreadsheet to interpret the account code “1000-400” as the number 600. This will not happen if the account code appears in the clipboard as “1000-400”.

Select **Skip Accounts Showing all Zeros** to have Chart skip those accounts for which the periods you have specified have zero values.

The Chart window also contains several edit boxes.

Accounts: Enter a standard F9 account specifier for the account you wish to match and send to the clipboard.

Periods: Enter one or more standard F9 period specifiers to view as well. Separate each specifier with a comma. For example, if you wish to view the periods “this month” and “this month last year”, enter the following into the Periods list:

```
this month,this month last year
```

Press the **Period** button to invoke the Period Wizard. The Period Wizard is a convenient way of creating period specifiers without having to remember the key words that F9 recognizes. Selecting “All Periods” can be especially handy for dumping budgets to the spreadsheet for editing and subsequent write-back to the accounting system’s budget files. A trial balance can be created in seconds by requesting all accounts for all periods. Select Actual; This Year; Ending Balances; Period; All Periods in Period Wizard to do this.

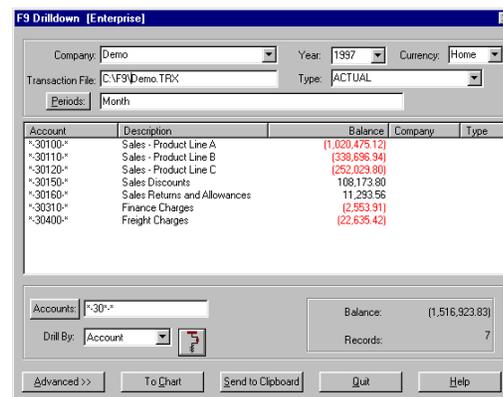
Company: Enter the name of the company or the path to the company. In some cases you will be presented with a Select button.

Year and Type: The base year and the type (usually a budget identifier) desired can also be selected.

Version 4:

F9 can copy lists of accounts and transactions to the clipboard with or without formulae from the Drilldown window using the Advanced options. This function is no longer available from Chart.

The Drill Down Window



Use the Drill Down window to:

Examine how a GL result was derived; E.g., breakdown an account balance by segment, company, or type;

Examine how a balance is built up;

Access detailed transactions.

The Drill Down window is similar to the Chart window in that both functions can be used to explore balances in your chart of accounts. The primary difference is that Chart is “single-tiered” and always displays detailed accounts, whereas Drill Down is “multitiered” and will display balances across one or more companies, types or account specifiers that use wildcards. The Drill Down window will also let you drill down to individual transactions (**Note:** access to transactions is not available with all DLLs).

When you point to a cell in your sheet that contains a GL (or NGL) function and select Drill Down (for example, by pressing the Drill Down button), you will launch the Drill Down window. The Drill Down window contains

entries for (at least) an account specifier, period specifier and company specifier. The initial specifiers will all be taken from the GL function that the cell cursor was pointing at.

Only the first account and company specifiers are used.

Version 4

The Drill windows now displays data using multiple companies, types or accounts. Also, the columns may be sorted on any column by pressing the header button (clicking again results in reverse order). This may not be available for your DLL. See the help file for your DLL.

Each Drill Down window has a **To Chart** button. If you wish to see **all** detailed accounts that match the current specifiers, press this button. The current specifiers will be loaded into the Chart window and that window will be launched. You may then press **Find** to locate all matching line items in the Chart of Accounts.

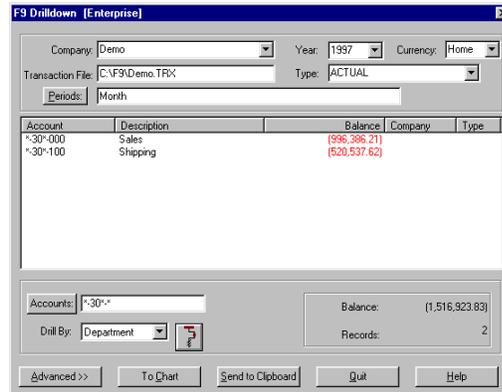
Drilling by Account Segment

The top “tier” in any accounting system is represented by the specifier *-*-*-* (for a four part account code). In most systems, this specifier will have a balance of zero, that is, the sum of every income account, which is net income for the period, plus the sum of the balance sheet accounts should make the balance sheet balance and produce a zero result. It is not generally recommended that you use this specifier, however, as it might take a while to calculate.

From this example, you may now drill into any segment of your chart of accounts by selecting the segment and pressing the drill button.

The screenshot shows the 'F9 Drilldown [Enterprise]' window. At the top, there are fields for 'Company: Demo', 'Year: 1997', and 'Currency: Home'. Below these are 'Transaction File: C:\F9\Demo.TRX' and 'Type: ACTUAL'. A 'Periods: Month' field is also present. The main area contains a table with columns for 'Account', 'Description', 'Balance', 'Company', and 'Type'. The table lists several accounts with their respective descriptions and balances. At the bottom, there is a section for 'Accounts: *-30*-' with a 'Balance: (1,516,923.83)' and 'Records: 7'. A 'Drill By:' dropdown menu is set to 'Account', and a 'Drill' button is visible. Other buttons at the bottom include 'Advanced', 'Department', 'Transactions', 'Send to Clipboard', 'Quit', and 'Help'.

Account	Description	Balance	Company	Type
30100	Sales - Product Line A	(1,020,475.12)		
30110	Sales - Product Line B	(338,636.84)		
30120	Sales - Product Line C	(252,029.80)		
30150	Sales Discounts	108,173.80		
30160	Sales Returns and Allowances	11,293.56		
30310	Finance Charges	(2,553.91)		
30400	Freight Charges	(22,635.42)		



When the drill button is pressed, there will be a pause as F9 calculates the balance for the current request. Typically, this will be the same balance that appears in your spreadsheet. After the balance appears, F9 will begin the drill process. A record count will be displayed as each matching segment is found and its associated balance calculated.

You may interrupt a drill at any time by pressing a key or mouse button.

Let's assume the first account segment represents the profit center and this chart of accounts has six profit centers numbered from 100 to 600. When you drill this specifier on the first segment, Drill will display the balances for six specifiers: 100-*-*-* , 200-*-*-* *etc.*. The drill display might show:

100-*-*-*	Office Supplies	0
200-*-*-*	Furniture	0
300-*-*-*	Wrenches	0
400-*-*-*	Widgets	0
500-*-*-*	Tape	0
600-*-*-*	Piping	0

This example does not produce particularly meaningful results because, once again, F9 is summing all accounts for each profit center. A more meaningful result might be obtained if we drilled the specifier *-*-*4*-* where the third segment is the account and accounts beginning with 4 are revenue accounts. In this case, the total that drill will display will be gross revenue and, by drilling segment one, we are asking "Show me revenue by profit center." The display might then show:

100-*-*4*-*	Office Supplies	5,000
200-*-*4*-*	Furniture	1,000

300-* -4* -*	Wrenches	50
400-* -4* -*	Widgets	10,000
500-* -4* -*	Tape	15,000
600-* -4* -*	Piping	1,200

for a total of \$32,250. You could then select one of these specifiers by clicking on it with the mouse. If you selected 300-***-4***-* and drilled on segment 3, you would in effect be asking “How is revenue for Wrenches broken down?” The display might now show:

300-* -4010 -*	Big Wrenches	10
300-* -4020 -*	Little Wrenches	15
300-* -4030 -*	Socket Wrenches	25

for a total of \$50.

Alternately, let’s suppose the second segment of the account code is the location and chose to drill the original specifier ***-*-4***-* by segment two. Now you are asking “Show me revenue by location instead.” Your display might show:

* -100-4010 -*	San Francisco	15,000
* -200-4010 -*	Los Angeles	10,000
* -300-4010 -*	New York	7,250

for the same total revenue of \$32,250. In this way, you may drill down through the four tiers (four segments in the account code implies four tiers) until you get to detailed accounts — that is, accounts with no wild cards in them — similar to those that the Chart window generates.

At any time, you may press the **Send to Clipboard** button to copy the accounts and balances to the Clipboard. Values or formulae will be generated using the settings as from the **Advanced...** dialog. You may send more than one set of data to the Clipboard if desired. The Clipboard is only emptied on entry to the Drill Down window.

The various parameters to the GL function, except the segment that was drilled, are set up as titles over the generated formulae. The drilled segment is created as the first column, the segment title is the second column, and the formula is the third. If you are generating formulae in Excel, the created formula will use the BSPEC function to recombine the drilled account segment with the account specifier to create a valid account specifier.

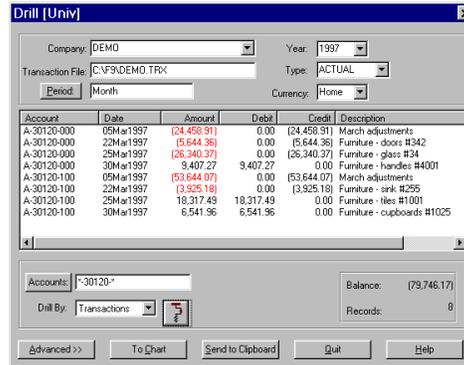
Many accounting systems support the ability to name the various codes that occupy a segment. For example, profit center 100 might be called “Office Supplies”. If your accounting system supports segment names, then the names will also be displayed along with the balances. Otherwise, NA is displayed.

Drilling to Transactions

At any time, you can Drill Down to the transaction level by selecting the special segment “Transactions” from the segment drop-down list. For those DLLs that support access to transactions, the last segment listed will always be the special segment “Transactions”. If you do not see this segment, your DLL does not support access to transactions.

Account	Date	Amount	Debit	Credit	Description
A-30100-000	05Mar1997	(82,876.55)	0.00	(82,876.55)	Dak cabinets
A-30100-000	22Mar1997	(13,125.40)	0.00	(13,125.40)	March adjustments
A-30100-000	29Mar1997	(83,251.70)	0.00	(83,251.70)	Furniture - dining table #1200
A-30100-000	30Mar1997	31,875.61	0.00	31,875.61	Furniture - chairs #500
A-30100-100	05Mar1997	(178,166.67)	0.00	(178,166.67)	March adjustments
A-30100-100	22Mar1997	(13,621.30)	0.00	(13,621.30)	Coffee maker #1096
A-30100-100	29Mar1997	1,563.09	1,563.09	0.00	Furniture - chairs #521
A-30100-100	30Mar1997	32,703.25	32,703.25	0.00	Dish washer #111001
A-30110-000	05Mar1997	(55,727.86)	0.00	(55,727.86)	March adjustments
A-30110-000	29Mar1997	(17,661.30)	0.00	(17,661.30)	Furniture - table #546

When you have selected Transactions and then press the Drill button, F9 will begin searching for transactions that match the current account specifier and period. Matching transactions are displayed in the balances window. The balance displayed is the total of all the matching transactions. In the case of P&L accounts, this will typically be the same balance that appeared when drilling by segment. This is because F9 defaults to returning the total transactions for a period for P&L accounts. For Balance Sheet accounts, the value returned will typically be the change in balance for the period in question.



In most cases, F9 only provides access to transactions for the current year. This means, for example, if your period specifier uses the phrase “last year”, it will be ignored for the purposes of a transaction drill down. In addition, the phrase “budget” in a period specifier is also ignored since budgets do not have transactions.

As before, once the transactions have been displayed, you may press the **Send to Clipboard** button and then paste the results into your spreadsheet (or anywhere else, for that matter).

Due to limitations in some windows controls which F9 uses to store the data you have selected, there is a finite limit to the data which retrieved. If you exceed the limit (technically 64Kb of data), a warning message will be displayed to inform you that some transactions were ignored. You will need to request fewer transactions or fewer fields per transaction. If you are only fetching the account code, date, amount and description, F9 will typically be able to store approximately one thousand transactions.

Tip:

Use the advanced dialog to configure what style of formulae, if any, should be returned in the clipboard data. See the discussion below for more information.

Transaction Fields

F9 defaults to both fetching and displaying each transaction's account code, date and amount. By pressing the **Transaction Fields** or **Advanced >>** button, you can manipulate both what data is displayed and what data is fetched.



F9 always fetches the account code, date and amount. You can cause F9 to fetch any other fields that are available by checking the appropriate box.

Win16:

F9 can only display either the date or description (in addition to the account code and amount) in the balances list box but it can fetch any fields you desire. To view the other fields (e.g., journal numbers, source values, etc.) you will have to send the data to the clipboard and paste it into the application (e.g., your spreadsheet).

Options you select here are stored and re-used next time.

The Period Wizard and Transactions

If you are drilling a number on a spreadsheet, when you enter the Drill Down window the account specifier and period specifier will be set to: the account and period parameters of the GL function being drilled. If you then select the special Transactions account segment and press the drill button, you will be shown transaction data for the account(s) and period specified. If the number you are drilling was created by a GLTRAN function, the Drill window will be set with the appropriate dates and the drill "segment" will be set to Transactions automatically.

F9 reads the start and end dates for each fiscal period and displays transactions that fall between the dates for the selected period. However, you can request to view transactions for any date period. To do this, press the **Period** button

to invoke the Period Wizard. As you select various periods, you will notice the From and To dates displayed at the bottom of the window change. If desired, you may enter any From and To dates you wish. Use the date format DDMmmYYYY. As long as the From and To dates are the last fields you touch in the Period Wizard, then the dates you enter will be respected and used as the From and To dates when you do the next transaction drill down. These dates are ignored for a normal account segment drill. When you exit the Period Wizard, you will notice that the Period edit box is blanked out, indicating that the From and To dates you entered will be used.

F9 Professional:

Drill by period is dependent on the dates in Company Edit. If the dates in this Dialog are incorrect, the results of the drill will also be incorrect.

Generating Formulae

By pressing the **Advanced >>** button in the Drilldown window, another dialog is displayed:



The option to paste simple spreadsheets was one of the features of **Chart**, but by moving it into Drilldown, several new capabilities were added including greater ability to preview the result of selections. This dialog allows you to configure which spreadsheet is the target and what fields to provide.

You can paste data into your spreadsheet as fixed numbers or as formulae. Select **None** to generate raw data (recommended). You may also select either **Lotus/Quattro** formulae (begin with @) or **Excel** formulae (begin with =). If you select to create formulae, GL formulae for the periods you request will be created. These formulae will be evaluated by the spreadsheet when you paste the clipboard data into the spreadsheet.

Tip:

The best ways to generate formulae is through either the GL Wizard or the Report Wizard, and not through the Drilldown window. Please see Chapter 4. *Reporting Wizards* for more information.

Lotus 1-2-3 and Quattro Pro formulae are created without using cell references — each parameter is supplied to each formula explicitly. Also, some newer versions of 1-2-3 no longer accept the formulae as produced by the drill down window.

Excel formulae are created using Excel's Row-Column relative cell references. **Before pasting Excel formulae into the spreadsheet, Excel must be in "RC" mode.** You may select this mode by:

- Excel 4 — menu **Options | Workspace**
- Excel 5, 7 or '97— select **Tools | Options | General**

After you have pasted the data, you may return to the "A1" style of cell reference notation.

When you are satisfied with the parameters of your request, press the **Drill** button. A counter will be displayed showing the number of accounts as they are matched.

You may interrupt a drill at any time by pressing a key or mouse button on the dialog.

When the request is complete, your data will be in the Windows clipboard. Press the Clipboard button to invoke the standard Windows Clipboard Viewer and review the data that was fetched. After reviewing the Clipboard, we recommend that you close it, rather than minimizing it or clicking on the sheet. Your data is now in the clipboard and can be pasted into your application, if desired.

The **Chart** button opens the Chart Window (discussed earlier in this chapter).

The Budget Window

Use the Budget window to:

Write budget data from the Windows Clipboard back to your accounting system.

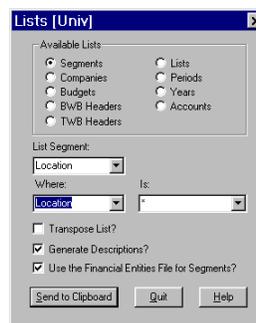
Most F9 products support budget write-back. Budget write-back allows you to create or edit budgets inside your spreadsheet and then write that budget information back to the accounting system for subsequent reporting.

Note:

Budget write-back is the only write access to your general ledger data that F9 supports. All other access is read-only. F9 Professional and Enterprise support Transaction Write Back, but this does not affect the source accounting system.

The budget window is documented extensively in *Chapter 9: Budget Write Back* which also discusses the use of WGL/WDESC as an alternative.

The Lists Window



Use the Lists window to:

Create lists in the Clipboard of accounts, department, companies, budgets etc..

There are many ways to construct an F9-based financial report. F9 provides a number of templates and sample spreadsheets that can make developing reports easier. However, whether you start from an empty spreadsheet or from one of our templates, you will be always faced with one basic task: entering your account numbers into the spreadsheet. That's where the Lists window can help.

For example, let's consider the basic problem of creating an Income Statement. An Income Statement consists of a number of sections, such as revenue, cost of goods, expenses, other income, and taxes. Each of these sections consist of a number of line items and each line item will have its associated account code or range of account codes. (When we use the phrase "account code" in this section, we will generally mean the segment of the complete account code that is the "natural" or "main" segment.) Thus, a major part of

the work in developing an Income Statement is to get these codes into your spreadsheet. Of course, one way to do it is to simply type them into your sheet, usually as a column of numbers or codes down the left side of the sheet. These codes would then be “cell-referenced” in the usual way.

Alternatively, you can get F9 to do much of this work for you using the Lists window. As the name implies, this window generates lists. The Lists window is another way to access the same functionality in F9 that is used by the Report Wizard. The example of the GL Wizard used in the chapter *Report Wizards* demonstrates how to generate complete and accurate reports using a single GL function and the lists window.

The actual lists that can be generated by the Lists window varies between accounting systems but, at a minimum, you will be able to generate lists of companies, budgets, periods, years and values for each segment of your account code. The lists you generate are always sent to the Windows Clipboard. Once in the Clipboard, it is a simple matter to paste the list into your spreadsheet at an appropriate place.

The Lists window has many potential applications, especially when used in conjunction with F9’s Silent Running feature (see *Chapter 7: Advanced Features* for more on this). For example, you could use the Lists window to create a list of departments. You could then paste this list into your spreadsheet and then perhaps use the list to drive the creation of a series of Income Statements using the RunF9Report macro we provide. The simplest use, of course, is to help you construct financial reports without having to type all of your account codes into the spreadsheet manually.

The Lists window can generate lists of values for each segment of your account code — including the main segment, of course — and it can create lists of other values, such as companies, as well. In this latter mode, the bottom three edit boxes (“List Segment”, “Where” and “Is”) will be greyed out. All lists created by the Lists window will, where applicable, reflect the default company in the Setup window. That is, if you ask for a list of budgets, you will get a list of budgets for the current default (*e.g.*, Setup window) company. Creating a list of values for anything other than segment values is trivial. Select the appropriate radio button and press OK. If all went well, a message will appear saying that the list was sent to the Clipboard.

Creating Lists of Segment Values

Creating lists of unique segment values is a tiny bit more complex, and more powerful. First, whenever you create lists of segment values, there may be a significant pause when you create the first list. This is because F9

will possibly have to scan your entire chart of accounts to create the list. Generally, F9 will only do this once per company and subsequent requests will execute quickly.

You can create segment lists in one of two ways: all unique values for a particular segment, or all values that meet a certain criteria. To create a list of all main account segments, you would do the following:

- Select the “Segments” radio button;
- Select your main account segment from the “List Segment” drop down list;
- Select “(None)” as the value for the “Where” edit box;
- And press OK.

F9 will send the list to the Clipboard. This technique is useful for creating lists of departments or profit centers. If you ask for all segment values, as in this example, the values will be sent to the Clipboard along with appropriate descriptions (if available). When asking for segment lists that use a “Where” clause (as in the following example), values are sent to the Clipboard without descriptions.

In many cases, however, you will only want a subset of the available segment values. The most common example of this is when you are constructing a financial report such as the Income Statement example we started out with. Suppose that your chart of accounts is organized such that all revenue accounts begin with a “4”, all cost of goods accounts begin with a “5”, *etc.*. In this case, to build the revenue portion of the report, you want a list of all natural account codes that begin with the digit “4” to appear as a column in your spreadsheet. To do this, think of the three edit boxes as making up the following sentence: “List all (some segment) where the (some segment) is (some value).” (where the three edit boxes are the three phrases in parentheses). To get our revenue accounts we want the sentence to read (assuming that the main segment is called “account”): “List all accounts where the account is 4*.” That is, set the List Segment edit box to “account”, the Where edit box to “account” and the Is edit box to the string “4*”. The Is edit box can use Financial Entities, ranges and wildcards for one segment value according to the normal F9 rules.

The result will be that all unique “account” values that begin with the digit “4” will be sent to the Clipboard and it is now a simple matter to paste them into your spreadsheet using the normal Windows **Paste** commands. Now to create a complete financial report, you just need to repeat the above process for each section of the report. Using the Lists dialog, it is possible to create reports where the data entry chore is dramatically reduced.

Your request can be more complex than this. For example, you could ask “List all accounts where the department is 100.” This would list all the accounts that are used by department 100. While generating lists such as this is possible, the prior example is far more common and useful.

List Options

Turn **Transpose List** on to send the list of values out as a row rather than a column of data.

Turn **Generate Descriptions** off to improve List and Report Wizard performance. This can be especially useful for larger charts of accounts or for account codes with many segments.

Using the Lists Facility with Silent Running

The Lists window follows the same rules as most F9 dialogs with respect to Silent Running. Silent Running is especially useful when used to drive the Lists window. Please read the section called *Silent Running* in *Chapter 8: Advanced Features*.

Financial Entities Editor

Use the Financial Entity Editor to:

Edit Sub-Account Descriptions imported from the Accounting System, i.e., “Natural” Financial Entities.

Edit User Defined Financial Entites.

In the DLL Window, select **Utilities | Maintain Financial Entities** to open this window. This editor has two views on the financial entities depending on whether or not **Edit Financial Entities** is checked or not. The labels for two controls change depending on the state of this control.

Controls (Edit Financial Entities not checked):

The screenshot shows a dialog box titled "Maintain Financial Entities File [Univ]". It contains the following controls:

- Segment:** A dropdown menu with "Segment 1" selected.
- Segment Value:** A dropdown menu.
- Segment Description:** A text input field.
- Edit Financial Entities** (unchecked)
- Note:** Avoid using these characters in entities: '!"?[]^
- Update/Create Entities File** button
- Add Record** button
- Delete Record** button
- Quit** button
- Help** button

This view of financial entities allows you to see and/or edit the “natural” Financial Entities which are commonly imported directly from your accounting system.

Segment Which segment will this entity exist in.

Segment Value Any segment valid specifier such as “1*” (Assets), “20000..20899” (Liabilities), or “*” (Any Location).

Segment Description This control holds the description of the specifier in the **Segment value** control.

Controls (Edit Financial Entities checked):

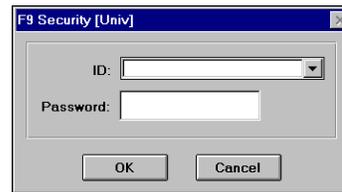
The screenshot shows the same dialog box as above, but with the following differences:

- Edit Financial Entities** (checked)
- Financial Entity:** A dropdown menu.
- Segment Specifier:** A text input field.
- Note:** Avoid using these characters in entities: '!"?[]^
- Update/Create Entities File** button
- Add Record** button
- Delete Record** button
- Quit** button
- Help** button

This view of financial entities allows you to see and/or edit the User Defined Financial Entities.

- Segment** Which segment will this entity exist in.
- Financial Entity** This control can contain a 30 character Financial Entity name such as “Assets,” “Liabilities” or “Expenses.”
- Segment Specifier** This control can contain 130 characters of specifier such as “1*” (Assets), “20000..20899” (Liabilities), or “032” (Houston). It can also contain a list such as “LA, Portland, Seattle, Vancouver” (Westcoast).

The Password Window



Note:

Most DLLs do not implement security, so this menu item and window will not exist. Security implementations are documented in the DLL help file.

When you select **Password** from a DLL menu, or from the F9 menu in your spreadsheet, an appropriate window dealing with security for your accounting system will appear. Like the Setup window, the Password window is virtually identical from one DLL to the next. This section will describe the “generic” Password window and its functionality.

For most accounting systems, passwords are associated with each company or set of books. Thus, you will normally have to specify the company you wish to access and then enter the appropriate password for that company or set of books.

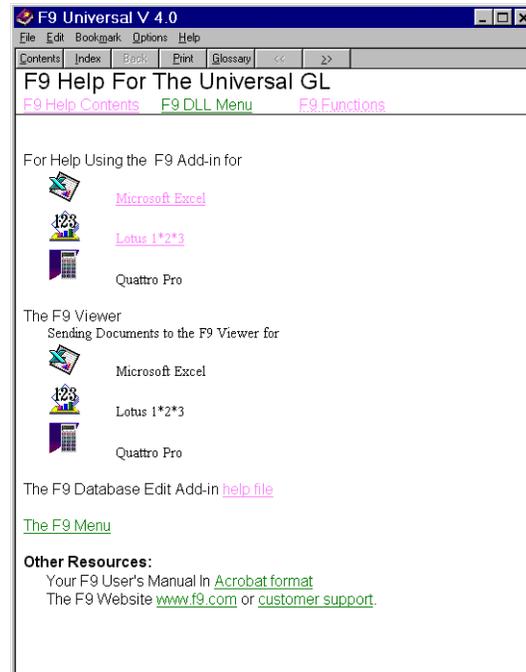
If you attempt to access general ledger data and you have not entered the required password information, F9 will return an appropriate error message, rather than the data desired.

The Period Wizard Window



When you select **Period Wizard** from a the F9 menu of your spreadsheet, or from a **Period** button in another F9 window, the Period Wizard window will appear. When invoked in this way, the normal usage is to create a desired period specifier and then send it to the Clipboard (by pressing the **To Clipboard** button) for pasting into the application (*e.g.*, spreadsheet). Using the Period Wizard initially in this way is a good way to learn the various period specifiers supported by F9. As you become more familiar with them, you will probably prefer to simply type the desired specifier into the appropriate cell in your spreadsheet.

Getting Help



Select Help and About from the DLL menu to get the version number and copyright information about the DLL.

Select **Help | DLL Help** from the F9 menu to get help about this DLL. Each DLL has help that describes the unique aspects of the DLL (that is, how it differs from the “generic” F9 DLL as described in the manual) and any special information about the accounting system as it relates to F9 that you need to know.

F9 version 4 introduces extensive context sensitive help from dialogs. All dialogs which support help have a **Help** button.

Client Commands

For the most part, the purpose of the clients (in most cases, spreadsheet add-ins) is to pass requests to the F9 Server and its DLLs. However, our clients for Lotus 1-2-3, Quattro Pro and Excel have several features that are independent of the of the Server and the DLLs: These features vary as little as possible between the spreadsheets.

Zero Suppress

The Excel, Lotus 1-2-3, and Quattro clients have a Zero Suppress command. The Excel client also creates a toolbar button for Zero Suppress and Quattro and Lotus 1-2-3 are supplied with a Zero Suppress Smart Icon.



Zero Suppress

The buttons and commands do the same thing and work almost identically in Lotus 1-2-3, Quattro Pro and Excel. Highlight a range of cells in your spreadsheet and select Zero Suppress (or press the appropriate button). The following will happen:

1. First, all rows in the range will be unhidden. In both Lotus 1-2-3 and Excel, a hidden row is essentially a row that is 0 (or 1 in 1-2-3 R5) pixels tall. In Excel and Quattro, a row actually has a hidden attribute and that attribute is turned off. In Lotus 1-2-3 R5, the rows are unhidden by resizing on a best-fit basis.
2. Next, each row in the highlighted range is scanned from left to right. If the row contains at least one number, and all the numbers that the row contains are zero, then the row is hidden. Thus, rows that contain only labels or empty cells are not affected.

Book2						
	A	B	C	D	E	F
1		Income				
2						
3		Sales - Widgets			\$6,645.69	
4		Sales - Doodads			945.46	
5		Sales - Thingys			5842.06	
6		Sales - Whatchamacallits			6997.80	
7		Sales - Brown Grommets			7456.82	
8		Sales - Red Grommets			5180.59	
9		Sales - Green Grommets			0.00	
10		Sales - Yellow Grommets			0.00	
11		Sales - Purple Grommets			0.00	
12		Sales - Brown Grommets			724.24	
13				Total	\$33,794.66	
14						

You can use Zero Suppress to quickly reduce the size of your financial reports to only include non-zero data. Sum rows (*e.g.*, rows that sum up detailed values returned by F9) will be eliminated automatically as well, as long as you include the sum row in the range to Zero Suppress. It is easy to hide title rows that are associated with all-zero details as well. To hide title rows, do the following:

- Create a hidden column in your report or use a column that is outside the printing range of your report.
- In the title row in this column, place a “sum of sums” function that sums either all the detailed values returned by F9 or the sums of those detailed values.

Finally, make sure this column is included in your Zero Suppress range. Because the title row now contains a number, and that number is zero, it too will be hidden.

To un-Zero Suppress data, you may use the Format Row Un-hide commands (Excel), re-size the rows (Excel or Lotus 1-2-3 '97), or you can use Zero Suppress.

To use Zero Suppress to Un-hide rows, highlight a range of cells that span the rows that are hidden, but that contain no zeros and then select Zero Suppress. Since the first thing Zero Suppress does is Un-hide all rows in the range, this will have the desired effect.

Column-wise zero suppress

Excel and Lotus 1-2-3 can perform column wise zero suppression. Quattro does not support this operation.

Column-wise zero suppression means a multiperiod report with columns of future periods (which should always be zero) can be suppressed until amounts are posted to these periods.

Column-wise suppression can be enabled or disabled by a check-box in the Options window. When the option is disabled column will not be suppressed. Hidden columns will be exposed whether column-wise zero suppression is on or not.

The Zero Suppress Macro

Zero Suppress is also available as a macro command called ZeroSuppress (one word). This macro takes one optional parameter: the range of cells to Zero Suppress. In Excel, the command can only function on the currently active sheet. If no parameter is supplied, the macro will Zero Suppress the currently selected cell or range of cells.

The macro formats in Lotus 1-2-3 are {UNSUPPRESS} and {ZEROSUPPRESS} or {ZEROSUPPRESS range}

Zero Suppression and 1-2-3 R5

Zero Suppress

Zero Suppress allows you to hide rows in your spreadsheets that contain only zero values. To use this command you select a range to be processed and select the Zero Suppress menu item, press the command button, or invoke the macro. Rows that have non-zero values and rows that contain no numbers are ignored. Rows that contain one or more numbers that are all exactly zero in value will have the selected cells set to a hidden format and row height set to 1.

Zero Suppress will remove any formatting in the selected cells.

1-2-3 R5 will not hide an entire row as either Excel or Quattro will. Only the area highlighted will be hidden, even though the entire row will be collapsed. Not including all of the numbers and text in a row may result in an area of collapsed but unhidden cells. To use this command you select a range which spans the cells to be exposed, and select the Undo Zero Suppress menu item, press the command button, or invoke the macro.

Undo Zero Suppress

1-2-3 '97 will **Undo Zero Suppress** automatically before attempting a zero suppression. This option is not on the menu for the add-in.

1-2-3 R5 uses the Undo Zero Suppress command to remove the Zero Suppression. Undo Zero Suppress will reset the format of the selected cells to comma punctuated numbers with two decimal places and will restore the row height to a best fit only.

To E-Mail/Viewer

Excel only.

F9 Version 4 is introducing a new spreadsheet add-in: the F9 Viewer. Selecting **F9 | To E-mail/Viewer** will cause a recalculation which will retrieve the current result of an F9 formula, insert the result into the cell (replacing the formula) and paste the query string into the cell's comment. This has the advantage of allowing users to receive and view the report without requiring full access to the original GL database.

For users who have access to the original GL database, the viewer is able to interpret the query string stored in F9 cells, and perform drill down analysis on the result. This allows users to analyze the results while limiting their ability to alter the report.

Note:

F9 will only convert cells to values if an F9 function immediately follows the '='. Normally this is not a problem since most uses of F9 are exclusive of other calculations. The few common exceptions are using CODATA to retrieve a company name for the report title or providing a scaling factor for budgeting. When an F9 function is found immediately following the "=", or "@", the entire cell is changed to a value, not just the function result.

To Value

Lotus 1-2-3 only.

To email a spreadsheet, select an area and select **F9 | To Value** to convert all of the F9 formulae in the selection to values. Due to limitations within 1-2-3, the formula is not preserved in the conversion and future drill down is not possible.

GL Paste



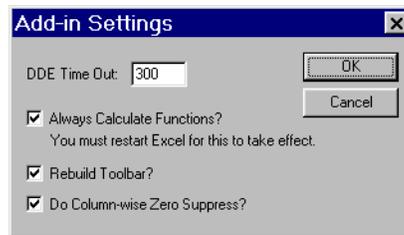
GL Paste

The GL Paste command uses the results of an Account Enquiry to format a basic F9 worksheet and the first GL Formula. If an Account Enquiry isn't available when the command is issued, an Account Enquiry is performed in the background; the result will be the equivalent of the last GL function evaluated.

Options

The options dialog offers several choices which were previously set by editing INI files. These options have well defined effects and reasons for choosing to turn them either on or off.

The dialog will look like:



These four options currently configure:

DDE Timeout The DDE Timeout controls how long a client should wait before assuming the server has “failed.” In this case, the client is the Spreadsheet add-in, and the server is F9.EXE. The default of 300 seconds (5 minutes) may seem excessive, but reducing this number may cause other applications to fail. For most accounting systems, F9 can respond considerably faster. In the event the F9 server *is* shutdown, each F9 formula in the spreadsheet will take this time to fail.

Always Calculate Functions?

This option affects the way F9 functions are registered in your spreadsheet. The default setting (On) instructs the spreadsheet not to attempt to optimize this formula out of recalculations. By turning this setting Off, the spreadsheet will recalculate the formula only if a parameter has been changed.

In simple terms, F9 functions are generally considered “dirty” and need recalculation because the results are primarily dependent on the database and not the function parameters. Alternatively, if your GL is not changing frequently, or you don’t care to see every change, turning this option Off will improve your recalculation times by reducing the number of calculations performed.

Note:

Your spreadsheet is informed about the F9 Functions when the Add-in is loaded. Changing this option will not change the operation of your spreadsheet until the next time the add-in is loaded. You should restart your spreadsheet after changing any of these options.

Rebuild Toolbar?

When your spreadsheet loads F9, the Add-in will normally attempt to create a toolbar. There are several scenarios, such as Executive Information Systems (EIS) or simple user preference, where the toolbar is not desirable. Turning this option **off** will prevent the toolbar from being recreated.

Do Column-wise Zero Suppress?

Zero Suppress will normally suppress both columns and rows. By turning this option off, columns will not be zero suppressed. This is desirable when including future periods as column in a report, for example.

Always Recalculate Functions and Analyze

The Always Recalculate option can be used to enhance the operation of the “Analyze for Access” or it may make the results almost meaningless. It is important to understand how these options interact. F9 will display a warning if an analyze is attempted while “Always Recalculate Functions” is Off.

The enhancement is from the reduction of calls being analyzed. For example, a multiple period report will access accounts up to twelve times. By turning the recalculation option off, an analysis can process just one column of the report by editing a column heading (*e.g.*, the period) before the analysis.

Tip:

The simplest way to “edit” a cell and mark it as changed or “dirty” is to place the cursor on the cell using the keyboard or mouse, then pressing the F2 key and Enter.

The detraction in analysis is that only “dirty” functions are recalculated; that is, your spreadsheet recalculates only functions which depend on a cell that has changed since the last recalculation. For example, if you only edit a cell with a DESC function, your report would report that no accounts were accessed, but by editing the “Company” cell, almost every F9 cell would recalculate making it difficult to find variances. As a general rule, each account should be accessed the same number of times.

Thus, for best results when using analyze while **Always Recalculate Functions** is off:

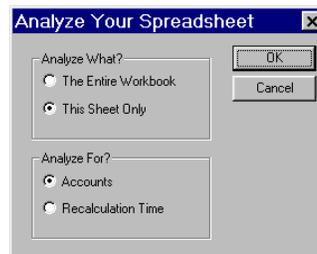
- Do a Recalculate (Press F9),
- Edit a column header (*e.g.*, click on the header, press **F2**, then **Enter** as above), and then
- Perform the **Analyze** command by selecting **F9|Analyze**.

This should result in the simplest and most usable report as possible.

The Analyze Command

When building a typical financial report with F9, it can often be difficult to determine why, for example, a balance sheet doesn't balance, or why an income statement doesn't add up to net income. The most common cause for such problems is the accidental inclusion or exclusion of one or more accounts. F9's Analyze function can make solving these problem much easier.

There are two significant analysis operations: Account Access and Recalculation Time. Each has its place in resolving problems with an F9 report. In addition, an analysis can include a complete workbook, or just the current sheet. All of the options are displayed on the Analyze dialog which is reached by selecting **F9|Analyze**.



Note:

The results of the Analyze function is heavily influenced by the Always Recalculate Functions option in the F9|Options dialog. See discussion below.

Analyze Account Access

An analysis for Account Access is useful in tracking down why an F9 report may not agree with your accounting system report. It is designed to show you which accounts are evaluated how often accounts are accessed: the vast majority of reports either shouldn't access an account or should access it a fixed number of times. Problems are shown when an unexpected account is included or excluded, or when a few accounts are accessed a different number of times.

After creating a financial report in Lotus 1-2-3 or Excel, select Analyze from the F9 menu. When you do this, the following happens:

1. A message is sent to the DLL indicating a recalculation is about to begin.
2. F9 then triggers a recalculation of the current workbook or the active spreadsheet.
3. During the recalculation F9 “takes notes” as to what accounts are being accessed.
4. When the recalculation is complete, another message is sent to the DLL indicating that the Analyze report should be created and sent to the Clipboard.
5. Finally, F9 will create a new spreadsheet and paste the results into it. In Excel, you will be prompted for the type of new object to be created... select Worksheet. In Lotus 1-2-3, you may have to paste the report into a spreadsheet manually.

A typical (truncated) F9 Analyze report looks like the following:

```
My Company Inc.  
The current sheet does not use the following accounts:  
From 000-1100-00 To 000-3030-00  
The current sheet uses the following accounts:  
000-4100-00    Sales                                4  
000-4110-01    US Sales - Retail/Parts                          4  
000-4110-02    US Sales - Finished Goods                        4  
000-4111-01    Canadian Sales - Retail/Parts                    4  
000-4111-02    Canadian Sales - Finished Goods                  4  
000-4112-01    Australian Sales - Retail/Parts                  4  
000-4112-02    Australian Sales - Finished Goods                4  
000-4114-01    Germany Sales - Retail/Parts                     4
```

The report is broken into two sections: Which accounts did the spreadsheet not access and which accounts the spreadsheet did access along with the number of times they were accessed.

In the preceding P&L example, the report indicated that accounts 000-1100-000 through 000-3030-00 were not accessed at all. These accounts correspond to the balance sheet as expected. The next section shows all the accounts that were accessed and the number of times these were accessed. The

following is another Analyze example report, except this time the report indicates a problem, one account has been added to the report twice and another was omitted:

```

My Company Inc.
The current sheet does not use the following accounts:
From 000-1100-00 To 000-3030-00
From 000-4600-00 To 000-4600-00
The current sheet uses the following accounts:
000-4100-00 Sales 3
000-4110-01 US Sales - Retail/Parts 3
000-4110-02 US Sales - Finished Goods 3
000-4111-01 Canadian Sales - Retail/Parts 3
000-4111-02 Canadian Sales - Finished Goods 3
000-4112-01 Australian Sales - Retail/Parts 3
000-4112-02 Australian Sales - Finished Goods 3
000-4114-01 Germany Sales - Retail/Parts 3
000-4114-02 Germany Sales - Finished Goods 3
. . . . .
000-4510-02 Cost of Goods Sold - Finished Goods 3
000-4601-00 Purchases Trade Discounts 6
000-4700-00 Shrinkage and Waste 3
000-4710-00 Freight and Handling 3
000-4720-00 International Freight and Handling 3
000-5100-00 Salaries and Wages 3
100-5100-00 Salaries and Wages - Administration 3

```

In this report we see that the account 4600 from the P&L was omitted and account 4601 was accessed twice as many times as any other. The problem is clear... we have accidentally entered account 4601 twice; one of the references is almost certainly a typo of 4600.

Using Analyze makes diagnosing problems with a report a snap.

Tip:

If you are using a GL function to calculate net income in order to make sure that the number you have calculated in your detailed P&L report is correct, we recommend that you disable this function (perhaps by turning it into a label) before running Analyze. Removing validation functions from the analysis improves the results, in particular the "omitted accounts" section of the Analyze report is more meaningful.

Analyze Re-Calc Time

A recalculation time analysis is most useful in resolving problems with spreadsheets that take a very long time to recalculate.

After creating a financial report in Lotus 1-2-3 or Excel, select **F9|Analyze** from the menu and select the “Recalc Time” option in the Analyze dialog. When you do this, the following happens:

1. A message is sent to the DLL indicating an analyzed recalculation is about to begin.
2. F9 then triggers a recalculation of the current spreadsheet (in Excel, only the active sheet is recalculated).
3. During the recalculation F9 notes the start time, performs the calculation, then calculates the elapsed time *and returns the elapsed time* instead of the calculation result.
4. When the recalculation is complete, another message is sent to the DLL indicating that the Analysis is complete.

At this point, your spread sheet is filled with recalculation times for every F9 function. By looking at the results it should be evident which calculations are responsible for slowing the spreadsheet down.

Tip:

Be sure to show all columns and rows. It is not uncommon for the error to appear in a row or column that has been zero suppressed or hidden.

If you arrive at a situation where your sheets are taking an excessive amount of time to recalculate and none of the performance suggestions help (see “Performance Enhancing Formulae” in *Appendix A: F9 Tips and Tricks*), contact our technical support staff.

Chapter

7

7. Advanced Features

Changing the Way Cells are Recalculated

Note:

This option applies to Excel and Lotus 1-2-3 only.

The default behaviour for Excel and Lotus 1-2-3 is to recalculate every GL function every time a recalculation (*i.e.*, pressing F9) is requested. It is possible to place Excel or Lotus 1-2-3 in a mode where GL functions are only recalculated when the spreadsheet believes they should be. That is, the spreadsheet will only request F9 to calculate a GL function if and when the cells the function refers to change.

To do this, add the following line to the Excel or Lotus 1-2-3 sections of the F9 INI file using a text editor:

```
Functions Dirty=N
```

Alternately, you can use the F9 Options dialog to set this same option. (See *Chapter 6: Commands and F9 Windows* for more information). Be sure to restart your spreadsheet after changing this option.

The next time you bring up F9, the spreadsheet will only recalculate GL functions that are “dirty”. This mode can make spreadsheet construction more pleasant, especially for large charts of accounts. When a spreadsheet is initially loaded in Lotus 1-2-3, all functions are tagged as “dirty”. This is not the case in Excel.

Caution: While the functions are marked as "not dirty," or the INI file reads "Functions Dirty=N," your spreadsheet will not recalculate F9 functions when changes are made to the Setup window that would change the values that F9 returns (*i.e.*, changing the default company). *To force cells to be recalculated, you need to “change” a dependency.* Usually, the easiest way to do this is to press **F2** and **Enter** on the cell containing the Company

Specifier (assuming that there is only one). This will cause the spreadsheet to treat all the F9 functions that refer to this cell as “dirty” and they will be recalculated the next time you press F9.

Note:

“Functions dirty=N” is a concern when using *Silent Running* where it is common to alter many dialog settings and recalculate the sheet. These operations may return misleading results if “Functions Dirty=N” is not taken into account.

Existing Lotus 1-2-3 Spreadsheets

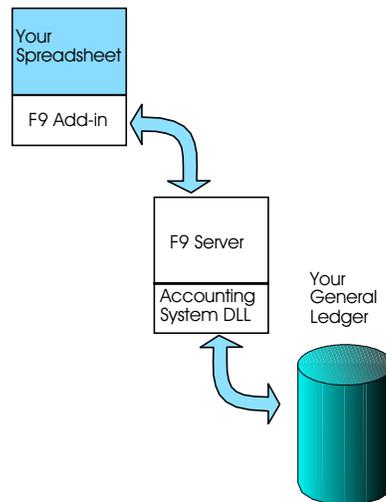
In Lotus 1-2-3, changing the *Functions Dirty* entry will have no effect on existing spreadsheets. To get the behavior described above in Lotus 1-2-3, you will need to rebuild your spreadsheets. The easiest way to do this is to press F2 and Enter on a model GL function, and then copy that function back through your spreadsheet.

How F9 Works

F9 consists of three components: a Client Add-in, a Server, and a DLL. (“DLL”, or Dynamic Link Library, is just a fancy name for a program.) The Client Add-in and the Server are generic (*i.e.*, they are not specific to any one accounting system). The DLL is accounting system specific. Each component is discussed briefly below.

Please note that for typical F9 installations, you do not need to be aware of the product organization described below. As a user of F9, you will load the Client Add-in (which can be done automatically or manually) from within

your spreadsheet. Everything else is handled automatically by F9. If you are not interested in the details of how F9 works, just read the text on Client Add-ins below and skip the rest of this section.



Client Add-ins

If you are using Excel, Lotus 1-2-3 or Quattro Pro, you will use a special client add-in that will manage information requests to the F9 Server. The add-in accepts requests in the form of spreadsheet formulae and replies with numbers or text.

Once attached, the F9 Add-in extends the power of your spreadsheet by adding a set of special functions and commands. This makes access to your general ledger balances as simple as entering a formula into your spreadsheet.

For Excel users, you will be using an Excel add-in — a file with the extension .XLL. The F9 add-in for Excel is called F9.XLL.

In Lotus 1-2-3, you will be using a Lotus add-in — a file with the extension .ADW or .12A. The F9 add-in for Lotus 1-2-3 is called F9.ADW, except 1-2-3 '97 and later which is called F9.12A.

For Quattro Pro users, you will be using a Quattro Pro add-in DLL called F9.DLL. This add-in will usually be loaded automatically by Quattro Pro when Quattro is launched.

More detailed instructions on loading an add-in into your spreadsheet is included in the *Chapter 2: Installation* or in the Appendix about using F9 with your spreadsheet.

The Server

Every copy of F9 will have a “server” and it is documented in a *Chapter 6: Commands and F9 Windows*. The F9 Server (the Server) provides access to accounting information to fill any client requests. To access the accounting system, the Server uses another DLL which is designed to read one accounting system. If you have several accounting systems on your computer, they are all accessed through one Server.

Accounting System DLLs

Your copy of F9 will include at least one accounting system specific DLL. Each accounting system DLL receives requests from the Server (and hence from the client) for data specific to its associated accounting system. F9 uses the DLL name as the “topic” of each F9 function. For example, the DLL called UNIV.DLL returns data contained in an F9 Professional general ledger. If you provide a “topic” parameter of “univ” to F9, F9 will attempt to use UNIV.DLL to retrieve the result.

When only one DLL is present, F9 uses that DLL as the topic. So, if you are using only one DLL, then you can largely ignore the Accounting System DLLs. For example, if you are only accessing data from an F9 Professional Datamart, then you can ignore the issue completely.

When more than one DLL is present, F9 will ask which is the default topic. Most people use only one DLL, so very few people will never see the request for a default topic.

For the latest version of the F9 software, visit our Web site at **www.F9.com**. The accounting system DLLs require a password which our Technical Support people can give you. You can contact Technical Support at **(604) 688-8271** or by e-mail at **support@synex.com**.

DataPreLoad

F9 version 3 introduced a feature called DataPreLoad that can significantly improve recalculation times. DataPreLoad allows you to load all or part of your General Ledger data into Windows memory before you recalculate your spreadsheet. With the data stored in memory, recalculation times can decrease by an order of magnitude or more for some databases.

A typical DataPreLoad window looks like the following:



You can pre-load data in one of three ways:

1. Manually by entering the DataPreLoad window from the Setup window. Once there, select the data desired and press the Load button. Selecting no accounts (*i.e.*, leaving the Accounts field blank) selects all accounts. In the simplest scenario, you could load all data for the current and prior year.
2. Automatically by setting up one of more sets of data to be loaded every time you run F9. After loading the data desired, press the button labeled ADPL (Automatic DataPreLoad). This will tell F9 to load the selected data every time you run F9. This is the recommended mode if your data sets are relatively small and/or you are working with only one or two companies or sets of books.
3. Using Silent Running Macros from within your spreadsheet. The various values in the DataPreLoad window can be set in the usual Silent Running way. The F9DPL command can then be executed inside your macro to actually load the data. F9 also provides an F9Reset command which resets F9. This has the side effect of unloading all DataPreLoad data. For example, if you had a number of companies, you could create spreadsheets for each; create Silent Running macros to load the data for the particular sheet as a macro that will run automatically every time the sheet is loaded; and then **link the sheets together via macros again to do production reporting.**

Tip:

There is more documentation on Silent Running in the next topic of this manual.

Correct use of the Silent Running feature will allow you to fine tune F9 on a sheet by sheet basis so that the data F9 requires to calculate the spreadsheet is loaded once and only one. Thus F9 can be made to not only run faster, but to run as fast as is theoretically possible!

Note:

Once data has been pre-loaded, it is no longer "hot", but is frozen as at the time it was loaded. If you wish the data to be refreshed, unload it (by pressing the Unload button) and load it again.

DataPreLoad allows you to load data on a company by company basis for This Year (where "This Year" is determined by the Year field, if present); Last Year and budgets. If the account specifier is left blank, the entire chart of accounts will be loaded. If a specifier is provided, however, only the accounts requested will be pre-loaded. For example, if you were working with a four segment account code where the second segment is the department, and you knew that the spreadsheet you were going to work with was a departmental income statement for department 100, you might enter an account specifier that looked something like this:

-100--*

In this way, DataPreLoad will work faster without sacrificing recalculation times. You may load as much data as desired. If you were going to work with two departments, change the account specifier and load the second department as well. The same is true for budgets.

Automating F9 Functionality

As the title of this Section implies, the goal is to tell you how you can automate various aspects of F9's functionality for applications such as Executive Information Systems (EIS). The methods we suggest are:

- Silent Running allows you to use spreadsheet macros to invoke F9 commands.
- Production reporting describes how to use macros to alter and recalculate reports so a single report generates multiple printouts for distribution with a minimum of interaction.
- Dynamic Data Exchange (DDE) requests from any other application.

Silent Running

If you are using F9 with a spreadsheet, you are aware that you can access various F9 windows by either pressing a toolbar button (*a.k.a.* smart icons) or by selecting an appropriate menu option. For example, you might select Setup from the F9 drop-down menu in Excel. The result would be that F9 would “wake up” and display the F9 Setup window.

There is another way to invoke these F9 functions. They can be invoked through a macro in your spreadsheet.

For example, if you were to execute the following macro in Excel, the result would be the same as selecting Setup from the menu as described above.

```
=Setup()  
=Return()
```

With this knowledge in hand, you can now create your own buttons to access the various F9 facilities by associating a button with a macro that contains commands that are recognized by F9.

As described in “How F9 Works” earlier in this chapter, F9 accepts a number of Dynamic Data Exchange commands, each of which has its corresponding macro. The commands and their macros are:

<u>DDE Command</u>	<u>Excel Macro</u>	<u>Lotus Macro</u>
[Setup]	=SetupO	{Setup}
[Budget]	=BudgetO	{Budget}
[Chart]	=ChartO	{Chart}
[Drilldown]	=DrillDownO	{DrillDown}

[PeriodW]	=PeriodWO	{PeriodW}
[Passwords]	=PasswordsO	{Passwords}
[F9Server]	=F9ServerO	{F9Server}
[F9Com1]	=F9Com1O	{F9Com1}
[F9Com2]	=F9Com2O	{F9Com2}
[F9Com3]	=F9Com3O	{F9Com3}
[F9Reset]	=F9ResetO	{F9Reset}
[F9DPL]	=F9DPLO	{F9DPL}
[F9List]	=F9ListO	{F9List}

There is, of course, little point in running these macros in this form since, in most cases, there are toolbar buttons or menu items to do the same thing already. Once an F9 window appears, you are free to enter values, press buttons and so on.

The last three F9 commands (F9Reset, F9DPL, and F9List) perform functions that are DLL specific. Their function is documented in each DLL's help.

If you try to run one of the above macros, such as the Excel example using Setup, you will notice that as soon as the macro encounters the F9 Setup command, the macro is suspended and the F9 Setup window appears. When you return to the spreadsheet, the macro picks up where it left off. **So the first problem is how can we invoke F9 functionality through a macro in such a way that F9 does not “wake up”, thereby allowing the macro to continue to run uninterrupted.**

The solution to this problem is to provide a special parameter to the F9 command. The special “silent running” parameter is:

NoShow=1 or NoShow=2

In Lotus 1-2-3 for Windows, the syntax for supplying parameters to an F9 command is as follows:

{Chart "NoShow=1" }

In Excel macro language, it is as follows:

=Chart ("NoShow=1")

In Visual Basic in Excel, use the following syntax:

Application.ExecuteExcel4Macro "Chart (" "Noshow=1" " ")"

Note:

For the NoShow parameter and all other DDE command parameters, *case is not important.*

If you were to execute a macro with the above syntax, the result would be that Chart would execute with its current settings and send the result to the Clipboard. While the Chart was running, the hourglass cursor would appear.

When the Chart completes, control would return automatically to the macro which could then, for example, paste the results into a spreadsheet somewhere.

A NoShow value of 1 tells F9 not to show any of its windows while serving the command. The only F9 windows that might appear will display error messages if the request failed. A NoShow value of 2 will stop F9 windows from appearing regardless of the success or failure of the request.

The primary distinction between F9 functions such as GL and F9 commands such as Chart is that functions interact directly with cells in the spreadsheet, and commands interact with Excel or Lotus 1-2-3 through the Clipboard.

Not all F9 commands respect the NoShow parameter (such as Password), and for some commands there is little or no point in running them from a macro (such as F9 Server and PeriodW). The commands that are of most interest are:

- **Setup** sets various options in the Setup window and other windows
- **Chart** sends account details to the Clipboard
- **Budget** writes budget data in the Clipboard back to accounting system
- **DrillDown** drills through various account segments or account transactions and sends the results to Clipboard
- **F9List** produces lists of accounts, companies, budgets, *etc.* and sends them to the Clipboard. The F9List command is really a subset of the Report Wizard functionality.
- **F9DPL** invokes a DataPreLoad
- **F9Reset** resets F9, closing all files and re-initializing all data, including any DataPreLoad data

The second problem to be solved with respect to F9 commands is providing a way to set the various options associated with each F9 window. Without a method to do this, F9 macros would not be particularly useful since they could only operate with current window settings — settings that can only be changed by hand.

The solution, once again, is to provide parameters to the F9 commands to set the various options associated with each window in F9.

Most of the options in each F9 window are remembered from run to run in the “INI file” F9.INI. This file is kept in your Windows directory. While every F9 DLL shares certain INI settings (such as rounding and scaling), they all have settings that are unique to the DLL as well. For those options

that are stored in the F9 INI file, the option is set through parameters to the F9 commands using almost the exact INI file text and syntax associated with the option.

The following is a fragment of a typical DLL's F9.INI file:

```
Chart Company=F:\F9U\UTIL\DEMO.BTR
Chart Account=*-*-*
Chart Period=1 Budget,2 Budget,3 Budget
Chart Type=BUDGET
Rounding=0
Scaling=0
Current Year=1992
Current Period=0
Chart Year=1992
Chart Show Account=1
Chart Show Description=1
Chart Show Origin=1
```

For each DLL, we have tried to make it obvious which INI file text line is associated with which setting. If you plan to write F9 macros, we suggest you print your INI file and have it handy as a reference.

An example should make this clear. Suppose you wish to create a button in Excel that, when pressed, will dump all the "1000" accounts for the current period out to the current sheet at the current cell. The macro associated with this button might look like this:

```
=Setup("NoShow=1,Chart Account='1000-*-*',Chart
Period='This Month'")
=Chart("NoShow=1")
=Paste()
=Return()
```

The Lotus 1-2-3 equivalent of the above example would be:

```
{Setup"NoShow=1,ChartAccount='1000-*-*',Chart Period='This Month'"}
{Chart"NoShow=1"}
{Edit-Paste}
{Return}
```

Equivalently, this example could be rewritten as:

```
=Setup("NoShow=1,Chart Account='1000-*-*'")
=Setup("NoShow=1,Chart Period='This Month'")
=Chart("NoShow=1")
```

```
=Paste()  
=Return()
```

Or as:

```
=Chart("NoShow=1,Chart Account='1000-*-*',Chart  
Period='ThisMonth'")  
=Paste()  
=Return()
```

Of the three Excel forms illustrated above, we recommend the second. We shall explain why in a moment.

Notice the use of single quotes around some of the parameters. These quotes are not absolutely necessary, but are always allowed. The use of single quotes **is required** if the parameter value includes commas. Thus, the quotes in the following example are not required:

```
Chart Period='This Month'
```

but **are required** in this example:

```
Chart Period='1,2,3'
```

This is the only significant difference between the parameter text and the INI file text.

Tip:

All F9 commands should include the NoShow Parameter as the first parameter in the list unless you want to edit the values in the F9 window controls.

Of course, the macros you write can be much more sophisticated than this simple example. It is beyond the scope of this manual to document in detail how macros are created in Lotus 1-2-3, Quattro Pro and Excel. Refer to the manuals associated with each of these products for more information. Some basic introductory information about macros is documented in the spreadsheet-specific chapters in this document.

For the greatest flexibility in editing and creating macros, we recommend setting one value per line and using and using cell reference such as:

```
=Setup("NoShow=1,Chart Period=" & Sheet1!A7)
```

This notation allows you to change what your macro does without having to edit the macro. Instead, you change the macro's behavior by editing cells in the sheet you are working with.

Your F9 product is supplied with many examples of spreadsheets that use macros for various purposes. Refer to the Button sample spreadsheet for examples of F9 command macros. This sheet includes macros that will (at the push of a button) create a Trial Balance; fetch budget data; write budget data back to the accounting system; drill a number by any segment of the account code; drill a number to fetch transactions, *etc.*, all without interacting directly with F9 at all!

Certain windows values and settings are not saved by F9 from run to run. Therefore, they do not have INI file settings. The most common ones are associated with the Drill window and are documented below. Any others will be documented in the DLL's help file.

```
Drill Account
Drill Budget
Drill Company
Drill Period
Drill Segment
Drill Year
Drill From
Drill To
```

The meanings of most of these values are self-explanatory. The Drill window segments are numbered from **zero**. Thus, supplying the parameter

```
Drill Segment=0
```

will drill by the first segment. To drill by transaction, you may either supply the value of the number of segments in the account code (*e.g.*, if the account code has three segments, then supplying the value three will drill by transactions, once again, because the segments are numbered from 0) or alternatively, you may supply any larger value (*e.g.*, 10) to drill by transactions.

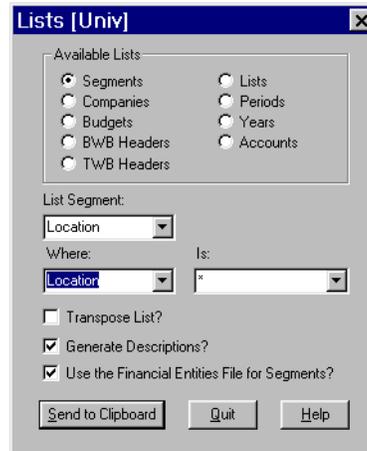
If you wish to automate drilling transactions, you may set the drill From and To dates using the Drill From and Drill To parameters. These dates must use the form DDMmmYY or DDMmmYYYY.

Using the Lists Facility with Silent Running

The Lists window follows the same rules as all F9 dialogs with respect to Silent Running. Silent Running is especially useful when used to drive the Lists window.

As always, you can review your F9.INI file to determine the names of the controls in an F9 window, and the values of those controls can be set using Visual Basic or macros in your spreadsheet and F9's Silent Running feature.

The typical Lists window resembles:



The main four controls (e.g., the radio button group and the three edit boxes) have the following names:

```
Selected List
Segment Segment
List Segment Where
List Segment Is
```

You select a particular radio button by number, starting from 0. That is, to select Segments as the type of list desired, you would enter into a Silent Running command the following string:

```
Selected List=0
```

Segments can be selected by number or by name. Typically, you would select them by name. For example, if your main or natural account segment was called "account", you could select it as the list to generate by including the following string in a Silent Running command:

```
List Segment Segment=account
```

Thus to create the list in our example, the set of Silent Running commands for Excel would read:

```
=Setup("NoShow=1,Selected List=0")
=Setup("NoShow=1,List Segment Segment=account")
=Setup("NoShow=1,List Segment Where=account")
=Setup("NoShow=1,List Segment Is=4*")
=F9List("NoShow=1")
```

and for Lotus:

```
{Setup "NoShow=1,Selected List=0"}
{Setup "NoShow=1,List Segment Segment=account"}
{Setup "NoShow=1,List Segment Where=account"}
{Setup "NoShow=1,List Segment Is=4*"}
{F9List "NoShow=1"}
```

Again, we comment specially on the Lists window with respect to Silent Running because it is so powerful. Using Silent Running and the Lists window, you can potentially automate many aspects of your spreadsheets. For example, suppose you construct an Income Statement and you wish to allow the user to run the report against a particular department. One option is to instruct the user to edit the appropriate cell in their spreadsheet and enter the appropriate department code before recalculating and printing the report. Alternately, you could create a button on the sheet that, when pressed, would display a dialog box listing available departments. The macro or Visual Basic procedure attached to the button would use Silent Running to create the list and it could, of course do much more, such as recalculate the report, zero suppress it, and then print it. Now you no longer need to remember department codes or which department is which.

Note:

It is beyond the scope of this manual to document how to develop macros or Visual Basic procedures. To access F9 commands inside Visual Basic, you need to use the `Application.ExecuteExcel4Macro` function.

Production Reporting with F9

Macros can be used to perform any repetitive reporting task. For example, you may have created an income statement for a specific department, profit center, location or any other segment of the account code. To run the same report for 15 other departments, you have a number of choices:

1. You could do it manually by editing the cell containing the department (there should be only one!), recalculating the spreadsheet, zero-suppressing the report area if desired, and then printing the report. This, of course, would take some time.
2. You could create a workbook with 15 pages in it. This will work fine, but adding departments would be a pain, and changing the report would be an even bigger pain because you would have to edit 15 (or more) pages.
3. Finally, you could use a macro. This is definitely the way to go! The Button sample sheet provides a macro called *RunF9Report* to accomplish this task. You may use the macro as is, or modify it

for your own purposes. You may want to copy the macro into your personal workbook (in Excel) so that it is always available. Here's how it works.

To use the RunF9Report macro, you will need to define three named ranges in your report spreadsheet. An example of this macro in action can be found in the Button spreadsheet.

The *SegmentValues* named range should indicate the top cell of a column of cells containing segment values (e.g., departments) that should be substituted into the report. This column of segments would typically be on the current sheet, but outside the printed area of the report.

The *SegmentTarget* named range should indicate the cell in your report containing the segment to be used in that particular report.

And finally, the named range called *ReportArea* should include the entire print area of the report.

When you run the report, the macro will start at the top of the list of account segments (*SegmentValues* named range) and, in turn, substitute each one into the cell containing the segment in the actual report (the *SegmentTarget* named range). For each segment value, it will recalculate the spreadsheet, ZeroSuppress the report area, and then print the area.

Creating F9 Dynamic Data Exchange Requests

If you are using a spreadsheet client add-in, you do not need to read this section. If however, you are not using Excel, Lotus 1-2-3, or Quattro Pro, you will have to formulate DDE requests yourself.

Each DDE request consists of a server (in our case, this is always "F9"), a topic string (one of Synex's DLL names, or the string "F9" which requests the default topic), and an item. The item is a string consisting of either a command (one of [Drilldown], [Setup], [Password], [Chart], [PeriodW], [Budget], [F9Server], or [Close]) or a function. The functions supported vary slightly from DLL to DLL, but all support GL, NGL, CODATA, ACCTDATA, and DESC at a minimum. Each function takes a number of parameters, each separated by a semicolon. For example, a valid item might be:

```
GL;0-0-1000-0;this month
```



Chapter

8

8. Editing Support Files

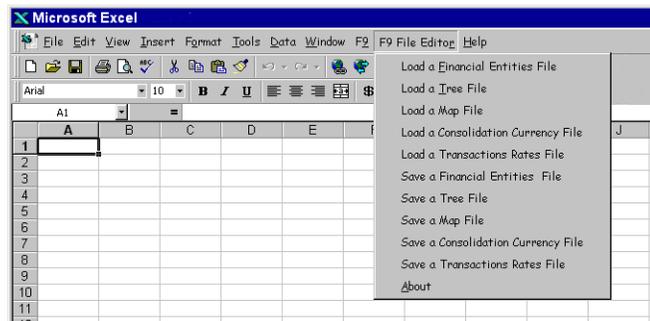
The File Edit Add-in (F9EDIT) is designed to allow extensive editing of several support files for F9's Datamart. With the exception of Financial Entities, most of these features are only supported by F9 Professional and Enterprise. For most editing of Financial Entities, we recommend the Financial Entities Editor window discussed in *Chapter 6: Commands and F9 Windows*.

The F9EDIT add-in can modify the Financial Entities used by Version 4 DLLs. The Financial Entity database (SAD file) contains the "natural" and user defined Financial entities. In addition *if you have either F9 Professional or Enterprise*, you will be able to edit:

- The MAP file which contains the account associations between companies.
- The TREE file which contains the financial ownership between companies.
- The CURRENCY.F9 file which contains rates of conversion and exchange between currencies for a series of dates. F9 supports a wide variety of systems for currency conversions depending on the date, currency, and account specifier.

These files are Btrieve database files and can not be edited except with F9 utilities. The MAP, TREE, and Currency files are specific to the F9 Professional and Enterprise Datamart systems. The SAD file is a new feature with version of F9.

The add-in is distributed on the CD with other Excel components. It is loaded like all other add-ins. Specific instructions for loading an add-in for your spreadsheet was given in *Chapter 2: Installing F9*. Once installed into your spreadsheet, the File Edit Add-in adds a menu item:



The File Edit Add-in uses a similar process to the Budget Write Back procedure (See “Budget Write Back with the Clipboard” in *Chapter 9: Budget Write Back*). When a file is loaded, the table is imported into your spreadsheet with headings. The table, including headers, is exported by placing the cursor in cell with the first header on the worksheet (normally A1) and selecting the “Save” option for the table.

Financial Entities

Note:

The use of Financial Entities in F9 Formulas is discussed in *Chapter 5: F9 Functions* within the discussion of *Function Parameters*. The rules regarding their sizes and types are discussed as part of *Financial Entity Editor* in *Chapter 6: Commands and F9 Windows*. Before using the F9Edit add-in to edit financial entities, you should be familiar with both of these previous topics.

Using the Financial Entity Editor

We highly recommend using the *Financial Entity Editor* window. It is reached by selecting **F9|Server** from your spreadsheet, then **Utilities|Maintain Financial Entities** in the DLL Window. See *Financial Entity Editor* in *Chapter 6: Commands and F9 Windows*. At the *very least* you should visit this window to initialize your database with all the known entities from your accounting system. Pressing the button **Update/Create Entities File** in the editor window will perform the initial import of all the “natural” financial entities.

For most accounting systems, financial entities are stored with the Sub-Account Descriptions in a Btrieve database file with the extension SAD.

Tip:

Your Accounting system may vary. Please see the DLL help file.

Financial Entity Edit in a Spreadsheet

Loading the Financial Entities imports data from a file with the extension SAD. This creates several columns of information in the current spreadsheet. The first row is occupied by headings. Since the Financial Entities are stored in a Btrieve database file, DO NOT attempt to open it in a text editor. The columns have the following meanings:

Segment Value The description below will apply to this value of the segment. Must be a valid segment specifier *or blank*. Blank is a special case where the Description is the Segment Description and not a description of the specifier.

e.g., segment two may have:

a blank value to describe the segment as “Location”
the value “*” described as “All Locations”.

Segment Number The segment of the account code being described. Must be a number between one and the number of segments.

Description The actual description text seen by F9. If the Segment Value is blank, this is the Segment description. If the Segment Value is not blank, the description is a financial entity

Is Financial Entity?

A **Yes** | **No** option which controls whether this is a Sub-Account Description (*i.e.*, a “natural” financial entity) or a User Defined Financial Entity. A “Yes” in this column indicates a User Defined Financial Entity.

When deciding to make a User Defined Financial Entity, remember to consider each of these side effects:

- User Defined Financial Entities are listed before natural entities in most listboxes.
- User Defined Financial Entities may also be significantly longer than descriptions. See sizes in topic *Financial Entity Editor* in *Chapter 6: Commands and F9 Windows*.

- User Defined Financial Entities are **not** accessible using the SDESC function.

Note the special meaning when the “Segment Value” is empty: the description is a Segment Name, not a description of a specifier.

Saving Financial Entities

Saving Financial Entities creates a file with the extension SAD. To do this, place your cursor in the cell containing the first header of the spreadsheet (normally A1) and then select **F9 File Editor | Save Financial Entities**. A standard file save dialog will allow you to replace existing file or create a new one.

Editing Column Names

Column names are used by the F9Data NGL, and GL functions with the syntax of:

F9DATA(Code, Column [, DataMart [, Year [, Type [, Currency [, Topic]]]])

NGL(Code, Column [, DataMart [, Year [, Type [, Currency [, Topic]]]])

GL(Code, Column [, DataMart [, Year [, Type [, Currency [, Topic]]]])

Note:

While this function accepts the Topic parameter, only the F9 Professional and Enterprise DataMarts support column names at this time (*i.e.*, the only useful topic is "UNIV").

Up to thirteen column names are available for any DataMart. When using a column based DataMart, the “column” parameter of the F9DATA, NGL and GL functions is not optional as it is in a period based DataMart (*i.e.*, a general ledger).

In an F9 Professional or Enterprise DataMart, each row may also be qualified by a year, type and currency. Column names can be edited as a part of the Company edit dialog (select **Utilities | Company Edit** in the F9 Univ DLL Window):

The screenshot shows the 'F9 (Univ) Company Edit' dialog box. It contains the following fields and controls:

- Company: Demo
- Company Name: F9 Demonstration Company
- Accounting System: In-house
- BS Amounts: Balances
- Current Year: 1997
- PL Amounts: Transactions
- Current Period: 3
- File Label: (empty)
- Currency: HOME
- Open Balance from Prior Year
- Main Segment: 2
- Use Column Names
- Edit Columns
- Fiscal Periods section with format MM/DD/YYYY and radio buttons for Start Dates (selected) and End Dates.
- Periods 1 through 13, each with a date field (e.g., Period 1: 01Jan1997).
- Buttons: Exit, Advanced..., Apply

Check the option to **Use Column Names** and press the button **Edit Column Names** to open this editor window:

The screenshot shows the 'F9 (Univ) Edit Column Names' dialog box. It contains the following controls:

- Column Names: (empty)
- Listbox: Gross Wages
- Add
- Delete
- Insert
- Change
- Save
- Quit

The controls in this window have the following functions:

Column Names This listbox allows editing and is used to select a column *and* input changes to the list of column names.

Add This button will add the current column name to the list by applying the name to the first unnamed column.

Delete	This button will remove the current name from the list of columns.
Insert	This button will insert the current item <i>after</i> the item which was selected before the edit. <i>e.g.</i> , if you select column 3, "Sales Tax Paid" and then edit the Column Name control to be "State Levy Paid" and press INSERT, column 4 will be renamed and all of the other column names below it will be moved down one.
Change	This button will replace the current item <i>with</i> the item which was selected before the edit. <i>e.g.</i> , if you select column 3, "Sales Tax Paid" and then edit the Column Name control to be "State Levy Paid" and press CHANGE, column 3 will be renamed.

The Tree file

The Tree file is not supported by all accounting systems. The tree file describes the relationship between two companies as a scaling factor of ownership. One assumption in this relationship is that the companies will have highly common charts of accounts.

Editing The Tree file

The contents of the Tree file is very simple:

Parent Company Subsidiary Control Interest

These fields can be described as:

Parent Company This is the company which *directly* owns the subsidiary to follow. Indirect ownership is explained below.

Subsidiary This is the company that is owned by the parent.

Ownership interest
A percentage of ownership interest.

To express multiple levels of ownership, which is the purpose of this file, several companies should be provided as "Parent Company" such as:

Company A Company B 75

Company B	Company C	40
Company C	Company D	90

In this way, Company A indirectly owns 30% of Company C by way of its ownership of Company B (*i.e.*, 75% of 40%) and 27% of Company D (*i.e.*, 30% of 90%). By providing only the direct ownership interests, editing is minimized prior to consolidation reporting since each ownership is present only once.

Saving The Tree file

Saving a Tree file requires placing the cursor in the cell containing the first header (normally A1) of the spreadsheet and then selecting **F9 File Editor | Save Tree File**. The Tree file has the same base name as the GL file with an extension of TRE. The company alias entry in the security file has no relationship with the file name (*i.e.*, the company alias “Bravo” can refer to the company database named “C:\F9\Data\FooBar.BTR”).

The MAP File

The Map file is not supported by all accounting systems.

The Map file is used to consolidate companies with very limited similarity in their chart of accounts. This would be the case if two companies in very different businesses were owned by a single company. The result of a MAP consolidation is a chart of accounts with only one or two segments in the account number. A chart of accounts made of just a natural account code will be common. How simple or complex the final chart is depends on how fully the accounts are merged. In general, the consolidated chart is simpler than the input charts.

The Map file is arguably the most complex of the three files because it begins with a theoretical set of books since the two companies being consolidated are dissimilar in their charts of accounts. If they were similar in their chart of accounts, they could be consolidated with a Tree file.

A second challenge is the potential length of the file; where the SAD and the TREE files could be as much as a few hundred lines, the MAP file can run into the thousands of lines. The length of a MAP file is determined by how many accounts are the result of a consolidation.

The most common use of a map file should be to collect all the similar accounts in the source GLs into broader meaning accounts in the new chart. For example, all the “revenue” accounts from each source company may be

mapped to one of about five accounts in the new chart based on whether it is a cash or credit sale, product return, or income other than sales (*e.g.*, capital gains, interest, or sublease, *etc.*). Similarly, expenses can be mapped into a small number of accounts and summarised. In this way, the output chart of accounts will be smaller and simpler than any of the source GLs.

Whether a consolidation is a detailed one-to-one map or a summary consolidation, each account in the source GLs should be included in the map file. The source accounting system will normally be rolled up during the consolidation so that while each company may maintain numerous sales accounts, a significantly reduced number will be reported in the parent companies books. This is commonly accomplished by using lists, wildcards or ranges during the consolidations.

For example, company A may have five sales accounts, and company B has seven sales accounts. The consolidated company would probably have only two sales accounts which are both gross sales for the two subsidiaries. A GL function with a MAP file would look like:

```
GL("3000", "Month 1", "Master&!minion,!peon")
```

A map file introduces a new syntax to the GL formula by introducing two new operators:

- &** The ampersand follows a consolidation company name to indicate companies to draw results from. That is, in the example above the company "Master" does not exist except as consolidated accounts from "minion" and "peon".
- !** The exclamation mark is often called "bang!" when used in formulae. This operator tells F9 to retrieve the values from this company directly rather than expect "Master" to exist or to have data from the constituent companies.

A MAP consolidated company can be pre-calculated by F9. F9 Enterprise users should see "Pre-Calculating Companies" in the chapter *F9 Enterprise Consolidations* in the F9 Professional and Enterprise *User's Manual*.

Editing the MAP file

The MAP file, while it may be very long, is a simple structure. Each line has four fields which are:

Consolidated Company

This is the company name used for the consolidated set of books. There can be several Consolidations described within a single MAP file.

Subsidiary Company

This is a real set of books accessible to F9. Map files do not support multiple levels of ownership. You can however map accounts differently into different parents.

Consolidated Account

This is an account code for the resulting consolidated books. Can not contain wildcards, lists, or ranges.

Subsidiary Account

This is the source account code in the subsidiary. The account code for the subsidiary account can be any standard F9 specifier including ranges, lists, and wildcards.

When supplying a “subsidiary account specifier,” you should use the BSPEC style of providing each segment value in parentheses. *E.g. rather than supplying:*

```
*-30100..30199-100..200
```

we suggest you use:

```
(*)(30100..30199)(100..200)
```

This style of specifier supports comma separated lists. Thus

```
(*)(30100..30199)(100,200)
```

is acceptable. There is no other acceptable form for this particular specifier.

When the map file is imported into excel by the F9 File Edit add-in, headers are inserted into the worksheet to mark the purpose of each column. When the file is saved, the headers are also saved.

The recommended method to consolidate multiple levels of ownership is to use pre-calculated companies.

Note:

Consolidation Pre-calculation is only supported by the F9 Enterprise DataMart.

Saving the MAP file

Saving a map file requires placing the cursor in cell A1 of the spreadsheet and then selecting **F9 File Editor | Save Map File**. The MAP file is a Btrieve database which can only be saved with the F9EDIT.XLL add-in.

Editing Currency Conversion Tables

The F9 Enterprise DataMart supports two distinct conversion tables: Consolidation Conversions and Transactional Conversions. These two tables have two specific applications:

Consolidation Conversions

This table lists conversion rates applied on a fiscal period basis for consolidation and reporting purposes. *E.g.*, Cash accounts can be reported in another currency at a rate appropriate for these accounts.

Transactional Conversions

This table lists daily rates for converting transactions to another currency *while the transaction is posted to the DataMart GL*. This allows a GL to originate in one currency, but be posted to the DataMart GL in additional currencies.

These tables can be edited using the F9Edit Add-in or using the File Maintenance Utilities (FMU).

Dates are saved as julian dates (a numeric format) and must be formatted as a date format your spreadsheet recognizes.

Editing Consolidation Conversion Rates

This table of rates can store a unique rate for each combination of: company, account, period, source and target currency.

Consolidation conversions are normally symmetrical so source->target = target->source; if F9 can not find a rate for a source currency to the target, it will look for a rate from the target to the source and reverse it.

Editing Transaction Conversion Rates

This table of rates can store a unique rate for each combination of: company, account, period, source and target currency. Transaction conversions are not assumed to be symmetrical where source->target = target->source, so the conversion rate is expected to be used.

When a Target company is created, it will be created in the F9 Security and consolidations directory and is added to the security file. If the Target company exists, the consolidation must have the same account number format and the records will be placed in the existing file.

Note:

While currency conversion allows you to use the same DataMart for both source and target, a roll-up will not; the purpose of a roll-up is to eliminate segments and to store data in a DataMart, the structure of the account number must be consistent.

Note:

When you process data from a source DataMart into a Rolled-up DataMart where the target DataMart has data for the same year, account type and the account format, the records from the source DataMart will *replace data* in the Target. There is NO confirmation that the new data actually comes from the same company as the existing data. For example, you can pre-calculate a company using a consolidation map and then roll-up data into a target company. The Roll-up process assumes that if the account format matches, YOU have confirmed that no one is using the old data and that it is okay to overwrite it.

By selecting the option to “Create Empty Roll-up (Budget)” and pressing Ok, a new chart of accounts is created, but no GL amounts are computed or stored in the database. Using CHART, WGL, and WDESC, this database can be populated with budget values.

How to Perform Currency Conversions

Currency conversions are performed in the Roll-up dialog in the F9 Enterprise DLL by selecting a home and target currency which requires a conversion (*i.e.*, the currencies are different). When you change the Target currency so that it is different from the Source, all the segment selectors should be disabled and checked since a currency conversion requires all of the segments.

For more information on currency conversions, F9 Enterprise users should see the chapter *Currency Conversions* in the F9 Professional and Enterprise *User's Manual*.



Chapter

9

9. Budget Writeback

Budget write Back has been a feature of F9 for sometime. While our use of the clipboard was not the most straightforward, it was effective and allowed us access to write budgets for almost every GL that F9 could read.

F9 Version 4 introduces another milestone: the WGL and WDESC functions. The combination of these two functions allow you to write your entire budget with the same ease that you currently author your financial reports. *In fact*, the best use of WGL is for budgeting; prepare your budget in a report style, then write it to the GL using a second sheet in the same workbook.

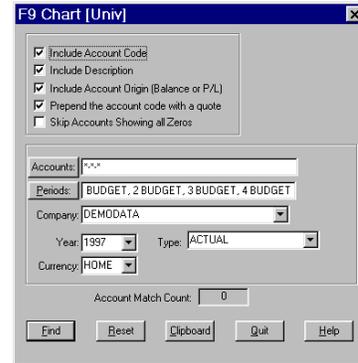
Budget Writeback with the Clipboard

The simplest and most effective way to use the budget write-back feature is in conjunction with the Chart window as follows:

- Go to the Chart window. Select appropriate accounts. For example, often P&L revenue accounts will begin with the digit “4” and an appropriate account specifier might then be “4*-*”.
- Then enter the periods you wish to budget. Usually you will want to budget all twelve periods and so an appropriate period specifier would be “1 budget,2 budget,3 budget...”. One way to generate all budgets is to use the Period Wizard. Press **Period** to invoke the Period Wizard. Select **Budget** (rather than **Actual**) and **All Periods** for the period entry.

9. Budget Writeback

Budget Writeback with the Clipboard



Tip:

Press Period to invoke the Period Wizard. Select 'Budget' (instead of Actual) and 'All Periods' for the period entry.

- If your accounting system supports multiple budgets, you will also want to select the appropriate budget.
- When you are ready, select Find and Chart will place the selected budget information into the Clipboard.
- Go back to your spreadsheet and Paste the budget data into the sheet. **Important:** Re-size the columns that contain numbers so that all significant digits display. You may now edit your budget data as desired. Feel free to use the facilities of your spreadsheet for this purpose.

	A	B	C	D	E	F
1	C:\TEMP\DEMO.BTR		1992			
2	Account	Description	1 Budget	2 Budget	3 Budget	4 Budget
3						
4	'A-30100-000	Sales - Product Line A	-156,547.50	-157,956.43	-159,378.04	0
5	'A-30100-100	Sales - Product Line A	-180,612.20	-162,057.71	-163,516.23	0
6	'A-30110-000	Sales - Product Line B	-105,265.70	-106,213.09	-107,169.01	0
7	'A-30110-100	Sales - Product Line B	0	0	0	0
8	'A-30120-000	Sales - Product Line C	-46,201.01	-46,616.82	-47,036.37	0
9	'A-30120-100	Sales - Product Line C	-32,128.88	-32,418.03	-32,709.80	0

When your budget is complete, highlight it (*i.e.*, select it) and select **Copy** to copy your budget back to the Clipboard. Be sure that you include the column titles as generated by the Chart window.

Account	Description	1 Budget	2 Budget	3 Budget	4 Budget
A-30100-000	Sales - Product Line A	-156,547.50	-157,956.43	-159,378.04	0
A-30100-100	Sales - Product Line A	-160,612.20	-162,057.71	-163,516.23	0
A-30110-000	Sales - Product Line B	-105,265.70	-106,213.09	-107,169.01	0
A-30110-100	Sales - Product Line B	0	0	0	0
A-30120-000	Sales - Product Line C	-46,201.01	-46,616.82	-47,036.37	0
A-30120-100	Sales - Product Line C	-32,128.88	-32,418.03	-32,709.80	0

Warning:

Be sure that all the numbers being copied to the Clipboard display properly in the spreadsheet *before* the copy operation. Each spreadsheet copies numbers differently depending on how they are displayed. For example, if your columns are too narrow, all three spreadsheets will display “#” signs. If you copy these numbers to the Clipboard in Excel, “#” signs are copied, but in Lotus 1-2-3, the correct numbers are copied.

Finally, open the Budget window by selecting **F9 | Budget Write Back**:

and press the **Write** button to store your budget data back to the accounting system.

Budget Writeback with WGL and WDESC

Tip:

The syntax and basic use of the WGL and WDESC functions are discussed in “The Write Functions” in *Chapter 5: F9 Functions*.

The advantages of using WGL for budgeting are:

- Higher level of budgeting [accounting system dependent].
- Simpler method of creating and entering data.
- You can use all the power of a spreadsheet to analyze the budget (*e.g.*, graph, calculate changes, *etc.*).
- Use the same “presentation report” for data entry and approval.

- Use groupware methods to collect sheets from each department head and enter budgets with diverse inputs. *E.g.*, Send each contributor a spreadsheet workbook with their budget allocation and instructions to allocate their distribution; when the workbooks are collected, the WGL formulas write the budget into the central GL.

The simplest way to use the write functions is using a report based approach. The core steps are:

- Prepare a report with the new budget as contents. This can be derived from the previous years figures (using GL) or a revision of this years budget (if one already exists).
- Adjust the figures to your satisfaction; *e.g.*, adjust forecasts up or down, remove discontinued sales lines, add new expenses, *etc.*
- Build a “shadow” page which includes all the WGL and WDESC functions.
- Recalculate the worksheet.

Tip:

The text in the following steps refers to the use of the WDESC function. However, the WDESC function is not implemented for all of the accounting systems that F9 can access since many accounting systems use a chart of accounts that must be considered a part of *actual* accounts, and F9 will not write to anything of type *actual*.

Step 1. Prepare a budget for writing. Simply design a report that contains all of the accounts that will be included in the budget. You can use GL functions to retrieve last year’s numbers or this year’s budget.

Step 2. Adjust the figures to your satisfaction. For the report just created using a GL formula to get last year or the current budget, the first task is to convert all the cells to values. You can simply select **F9 | To Email /Viewer** in Excel, or **F9 | To Value** in Lotus 1-2-3.

Step 3. Build the WGL sheet. In a blank sheet of the same workbook, start a new report with one distinction:

Almost every *constant* on the write back worksheet should be a reference to the budget report sheet. That is, if the budget worksheet was named “Budget 98,” then the cell A1 on the write back sheet should refer to “Budget 98!A1”. The easiest way to achieve this is with these steps:

1. On a blank worksheet, place the cursor in cell A1 and press “=”.
2. Use your mouse to select the report sheet and click in cell A1 on this page.
3. Press the **Return** or **Enter** key. Your workbook should now be open to the write back page with an off-sheet reference.
4. Drag the formula across and then down your write back worksheet until the constants on the report worksheet are all included.

The principle exceptions to the write back constants are the YEAR and TYPE parameters. Where the first page will read either last year’s ACTUAL amounts, or this year’s budget, the write back page should always refer to the current year’s budget type. With the formula copied as above, you should be able to see all the constants so should edit the cells containing the Year and Type parameters to make the budget write to the correct year.

Next, you need to make one GL and DESC function on the budget report into a WGL or a WDESC on the write back sheet.

Common Problems with Budget Write Back

The Most common problems with budget write back using the clipboard are:

- Not selecting the “headers” when copying to the clipboard.
- Non-numeric data in the area (e.g., the “####” sequences that Excel copies) when copying to the clipboard.

The Most common problems with budget write back using WGL are:

- Not specifying a valid budget as the **Type**.
- Trying to write with an imprecise account code (can’t use lists, wildcards, or ranges).
- Mixing account number formats.

F9 Univ:

Some problems with budget write back can only be corrected by deleting data files. For example, if a company is created with a poorly formatted account number, the only way to correct the number is to delete the files. Fortunately, most accounting systems will prevent these sorts of errors.

Appendix

A

A. F9 Tips and Tricks

This appendix contains some hints for using F9 to produce reports.

Consolidations with Differing Charts of Accounts

To perform consolidations of companies with differing charts of accounts follow these steps:

- Set the **Return Zero for Account not Found** check-box in the Setup window on. This will allow companies to be consolidated even if the account structure is not common to both companies.
- If the companies in question do not share the same fiscal years, use Period Specifiers such as “January” rather than “month 1”. Avoid the use of Period Specifiers such as “month 1” which always means fiscal period 1, and relative Period Specifiers such as “3 months ago” or “this month”.
- If account descriptions vary from ledger to ledger, decide on which ledger’s descriptions to use (the DESC function can retrieve descriptions from the chosen set of general ledger data) or enter the descriptions you want to use directly into cells as labels.
- For each set of general ledger data, list the account numbers in separate columns outside of the area which will contain the actual report. Enter the numbers that correspond to the “generic” account

descriptions you've entered in the same order as the account descriptions.

When consolidating a ledger's accounts, reference the account number column for that ledger's set of data.

If you wish to bring balances from multiple ledgers into one cell you can easily do so by including more than one company in the company specifier. For example, the formula below would place the balance from company Terra and company Prod in a single cell.

```
GL("1030..2100","","terra,prod")
```

Using a formula like this assumes the two companies have a similar chart of accounts and are both using the same accounting system (F9 Topic). An alternative which would be marginally slower is to simply use your spreadsheet to add the two companies:

```
GL("1030..2100","","terra") +  
GL("1030..2100","","prod")
```

Performance Enhancing Formulae

The above suggestions illustrates one performance issue with F9: F9 performance is optimum of using the fewest possible requests to F9. For example, the single request from two companies will be faster than the two requests because *it takes time to communicate the request* and *it takes time to calculate a result*. Seen this way, it should be clear that it takes longer to make two requests than to make one and, for F9, the amount of calculation is virtually identical.

A second source of request reduction is to use another F9 result in spreadsheet formulas rather than request the numbers again. For example, if you request a balance for accounts 10100, 10200, 10300 and 10500, and then want a subtotal, add the results using SUM or addition rather than:

```
GL("10100,10200,10300,10500","This Month")
```

Optimizing Spreadsheets

The best way to begin optimizing your spreadsheet is to use the **Analyze** command within F9 to "Analyze for Account Access." In an ideal spreadsheet, you should expect to not access any number more than the number of periods + 1. For example, if your report is for a quarter, you would expect to access each account three times, or possibly four. The four would be the case if you implement control totals (see the next topic).

Analyze also provides a report of the times required to calculate each F9 function. Use this to find time consuming requests and optimize the ones that take too long. For example, if you use a wildcard like 1010? in a chart of accounts where only one account matches (*i.e.*, 10100 exists, but 10101..10109 do not), the request for 1010? will take *considerably longer* than a request for 10100 but return the same amount.

Depending on your reporting needs, you can choose to build “full reports” and then convert some elements to values rather than request them repeatedly. For example, the Chart of Accounts for most companies are quite stable so each company has portions of their reports “fixed.” After building and saving an F9 report “template,” you could convert the requests to SDESC, CODATA, and DESC to values and save a company version. This would be a simple annual task and would save some time.

Using Cell References

As we have tried to emphasize, it is much better to use cell references for items such as account numbers, and period specifications in order to be able to easily update future reports. By referencing, you are creating a “template” report that can be modified by only changing the referenced cells.

This is a good worksheet creation technique rather than something specific to F9. Therefore, just as it is better to place your assumption in worksheet cells and reference these cells rather than integrating these assumptions into every cell calculation, it is the same with F9.

Control Totals

Control totals provide a way of ensuring that the information in your reports is as reliable and accurate as possible. You can include control totals in your reports using F9. A variety of control totals are possible, but three of the most commonly used are:

$$\text{Assets} = \text{Liabilities} + \text{Shareholders' Equity}$$

$$\text{Closing Retained Earnings} = \text{Opening Retained Earnings} + \text{Net Income for the Year}$$

$$\text{Net Income for the Year per Income Statement} = \text{Net Income for the Year per the Balance Sheet}$$

Another method of ensuring that report amounts are reliable is to use a percentage change threshold for accounts in your income statement. For

example, you can use an IF function to display a message if let's say the advertising expense account has increased 50 percent over the previous month.

In addition, use IF functions or macros to display a message if account balances are questionable. For example:

```
IF(Assets=(Liab+SHE),"Balances","Doesn't Balance")
```

where Assets, Liab and SHE are range names for the total assets, total liabilities and total shareholder's equity cells in your worksheet. This function will display either:

Balances

or

Doesn't Balance

Place this function in an appropriately noticeable cell.

Note:

In many cases, due to truncation and rounding errors, two values that appear to be equal may not be. Often, values returned by F9 will be close, but not equal. For example assets may exceed liabilities plus shareholder equity by 1 part in 1,000,000. To correct for this, subtract one value from the other, take the absolute value of the result, and compare it to a very small number. Thus your formula might look like:

```
IF(ABS(Assets-(Liab+SHE))<.00001,"Balances","Doesn't  
Balance")
```

Named Ranges

To make this a little more sophisticated, you can name each of the cells above using "named ranges". Now your formula might look like:

```
GL(Accounts,Period,Company)
```

This makes your formulae that much more readable.

Note:

Do not create named ranges or macros with the same names as any F9 functions or commands. They may have the effect of disabling the function or command.

If we use the range-name trick we learned before, the BSPEC function now looks like:

```
BSPEC(Locations,Departments,Accounts,SubAccounts)
```

This BSPEC function does not have to be in a separate cell either. It can be right inside your GL functions. And so, finally, putting it all together, we get the imposing:

```
GL(BSPEC(Locations,Departments,Accounts,SubAccounts),Period,Company,Year,Type)
```



Appendix

B

B. Frequently Asked Questions

(and answers. Please review before you call for support!)

I want to create a rolling income statement that shows income for this month, last month, the month before that, etc. going back 12 months. How can I do this?

As you know, the period specifier “month 1” gives you the month 1 balance. The period specifier “1 month” (with the period number first) means “1 month ago.” You can also include the word “ago” in the period specifier to make this clearer, although in fact, F9 ignores it.

I want to get the month 12 ending balance for one of my P&L accounts, but I keep getting transactions for month 12. What am I doing wrong?

The word “balance” is often used loosely by accountants. A balance represents an accounts value at a *point* in time. Strictly speaking, only balance sheet accounts have balances. When accessing a P&L account, one is always asking for *transactions* over a period of time. Often, what one really wants when asking for the month 12 balance of a P&L account is the **year to date month 12 transactions** and this is the period specifier you should use. For P&L accounts, the word “transactions is implied and can be omitted, as can the words “to date”.

I want to give a copy of an F9 spreadsheet to an associate who does not have access to F9. Every time he loads the sheet, all the formulae turn to errors and the numbers go away. What can I do?

In Excel, select the entire sheet, do an **Edit | Copy**, and then an **Edit | Paste Special**. Select **Values** and press OK. Now do a **File | Save As** and save the workbook to a new file. Send this file to your associate. For Lotus 1-2-3, do the same thing except in **Edit | Paste Special**, select **Formulae as Values**.

New V4:

The Add-in for Excel has a new menu item To Viewer/E-Mail and Lotus 1-2-3 has To Value which convert all the F9 formulae to values for you. The conversion will leave all formulae which do not begin with an F9 function alone. The other benefit of the conversion in Excel is that the formula data is saved in a cell comment. Both Excel and Lotus 1-2-3 can use an F9 Viewer add-in to drill the original formula.

What has F9 done to be Y2K compliant?

The official statement is: F9 is as compliant with Y2K as your accounting system. What that means is that F9 uses all the date information in the accounting system so we are Y2K compliant with systems that use four digit years, and non-compliant with systems that only use two digit years.

I don't know if my Accounting system is Y2K compliant. How does F9 behave with non-compliant accounting systems?

Very well, we must admit. The only place where F9 becomes sensitive to actual dates is in **Transaction Drill Down**. Every other F9 function is fiscal period oriented and, in this area, dates are just a convention. Virtually every accounting system uses a fiscal period database where "month 1" of the year is totally independent of calendar dates.

Transaction drill Down on the other hand uses a linear database where each transaction is recorded in the order of entry. During a drill down, F9 selects transactions based on whether the date of the transaction is within a particular fiscal period.

For the accounting systems we have encountered which record a two year date, F9 assumes years less than 50 are after year 2000, and years greater than 50 are prior to 1999.

How can I get account descriptions out using the Report Wizard?

When defining your horizontal dimension (columns), select "Description" from the list box on the left and send it to the right hand "selected" list box. Usually, you will want it to be the first item on the selected list.

Some of my functions generate a "Buffer Overflow" error. What does it mean?

It means you have overflowed a windows buffer used to pass the F9 function information. For technical reasons that are not worth going into, a function cannot be longer than 256 characters. By a function, we mean the function name plus the Account Specifier plus the Period Specifier plus the Company Specifier, *etc.* viewed as a long string of text.

The usual cause of this is either a long account specifier or a long period specifier. In the latter case, you are probably using a comma separated list

of period specifiers which is rarely necessary. For example, you should enter “year to date month 12 budget” rather than “month 1 budget, month 2 budget,...”. Long Account Specifier can best be avoided by using wildcards and ranges to reduce the total length. You should use the BSPEC function to reduce the total length to create permutations. As a last resort, break the GL function up into two or more functions and add them together (this can all be done in one cell).

New V4:

Financial Entities let you make really large account specifiers. By making an entity which is a collection of entities, only one entity is passed in the windows buffer, but hundreds can be included in the query. For example, a “sales region” can include three states which include a total of 150 sales outlets under a divisional sales manager. By requesting sales for the region (one entity), F9 expands it into three entities and those into 150 specifiers, before evaluating the query and returning a result.

I am getting a bunch of DDEML errors. What are they?

DDEML refers to the *Dynamic Data Exchange Management Library*. You will usually get these errors when the F9 Server isn't running. The server normally starts and quits automatically when the F9 add-in is loaded by your spreadsheet.

The most common *other* way for the Server to close, and cause the errors, is for you to shut it down manually by selecting “Quit Server” from the Server main menu. In general, you should **never** do this. If you want the Server out of the way, minimize it but do not shut it down. The Server will be shut down automatically when you exit your spreadsheet.

Tip:

DDE Timeout errors are specifically managed by the Options Dialog. See *Client Commands* in *Chapter 6: Commands and F9 Windows* for more information.

Can I get total debits and total credits for a period?

Yes, but not with the GL function. Most accounting systems do not carry debit and credit values in the period “buckets” file. However, the GLTRAN function takes a Type parameter that can indicate either debits or credits. To get total debits or credits for a period, set the start and end dates to match the fiscal period and supply a Type parameter of either “debit” or “credit”. Caution: the GLTRAN function is substantially slower than the GL function.

I am trying to get a range of accounts. My account code has two parts. I enter an account specifier of the form:

4000-* . . 4999-*

and I get an error. What am I doing wrong?

This is a common error (and so we included your question in this section). F9 uses a more sophisticated approach to ranges of accounts. You must specify ranges on a segment-by-segment basis. The above example is correctly entered as:

4000..4999-*

This notation is more flexible because it allows you to use an independent range on the sub-account as well. The following example produces a balance for a range of accounts *and* a range of sub-accounts:

4000..4999-100..200

I have created an income statement for one department. I have 15. I want to run this income statement for all 15 departments in a row. How can I do this without having to go through a manual edit/re-calc/print cycle 15 times?

If you load the Button sample spreadsheet, you will find a macro called RunF9Report. This macro is designed to do exactly what you have just asked. It will take a list of departments (or any segment of the account code) and repeatedly substitute them into a report, zero-suppress the report area, and then print the report. This macro is discussed in more detail in the chapter called *F9 Templates*.

I want to consolidate two companies that have differing fiscal years. If I use a period specifier like “Month 3” I get balances for March from one company and June for another. How can I get March from both companies?

Use the Period Specifier “March”. In F9 versions prior to 3.0, March would always map to fiscal period 3. In Version 3 of F9, as long as both companies have 12 or 13 periods and a fiscal period is typically a month long, you should get what you want.

How should F9 be installed on a network with Windows for Networks?

Install all F9 components in an F9 directory on the network (F9 Clients will be installed under the spreadsheet directories). You require a multi-user F9 license for this installation.

How can I get an income figure for just two months? Using the word “Quarter” gets me three months.

One way is to sum two GL functions in one cell. The simplest way, however, is to use a period list. That is, use a period specifier of the form:

3,4

Do not use period lists unless absolutely necessary. A GL function using the period specifier “1,2,3,4,5,6” will take six times longer to calculate than one using “year to date month 6”.

Why doesn't the CONAME function work?

The CONAME has been discontinued since Version 3. CONAME(company) should now become CODATA('Name',company). After setting up a few prototypical functions, you should be able to copy your functions around the sheet to recreate reports.

How can I get budget figures?

Add the word “budget” to your period specifier. For example, you might ask for “Month 5 Budget” or “This Month Budget”. This will return the default budget for your accounting system. If more than one budget is available, you can also specify a specific budget using the TYPE parameter in the GL function.

How can I get out the total annual budget; “Year Budget” doesn't seem to work?

The phrase “Year” is the same as “Year to Date”. Since you have not specified a month, the current period is assumed. To get the annual budget, regardless of the current fiscal period, use the period specifier “Year to Date Budget Month 13”.

How do I get last year's budget numbers? F9 returns \$0.00 for “Budget Last Year”?

With one or two exceptions, accounting systems don't associate a particular budget with a year. Most accounting systems treat a budget as an arbitrary set of numbers that you may edit. When an accounting system does not provide associations between years and budgets, F9 returns \$0.00 for all requests that are not the “this year.” For these same systems, F9 generally provides access to all the budgets within this year; it is your responsibility to keep track of which budget was for “Last Year” and to provide a specifier as the “TYPE” parameter to the GL/NGL functions.

How can I get the net change in balance for a period for a balance sheet account? I could subtract the ending balance for the prior period from the ending balance for the current period, but that seems like a lot of work!

Use the word “Transactions” or the word “change” in the GL function period specifier. **Do not use the GLTRAN function for this purpose!** The GLTRAN function is very slow compared to the GL function.

Every time I load my client add-in, a Server window pops up and makes me select a particular DLL. I only use one. How can I get the Server to default to selecting the one I use?

To disable an accounting system under F9, you must move a DLL out of the server directory. We suggest you make a directory named “disabled” under the F9 directory for the files. When the server asks you to select an accounting system, it displays a list of valid DLLs without their extension. Move the DLLs on this list that you do NOT use into the “disabled” directory.

Tip:

Under Win95/98 and Nt4/5, there is an option which hides certain files that most users generally prefer to ignore. One of the hidden file types is DLL. To make them visible, in Windows Explorer, select View|Options and then View Tab. In the current window, an option to “hide system files” must be turned off; the exact steps vary depending on the version of Windows and Internet Explorer you have installed. Also note that these versions of windows don’t display most file extensions, and refer to DLLs as “Application extension” in the Type column of the Explorer.

I am using Budget Write Back. I get the numbers using Chart, and then write them back. When I get the budget again using Chart or the GL function, some of the numbers are out, usually by a few cents. What is going on?

You must make sure to re-size the columns holding the numbers to display all significant digits *before you copy them to the Clipboard*. Otherwise, depending on the spreadsheet you are using, some accuracy may be lost.

How can I access the “future periods” in my accounting system? We have not closed the last fiscal year, but we are still posting data and I would like to access it.

F9 calls the future periods Months 14, 15, 16, *etc.*.

Is there a fast way to double check a spreadsheet and make sure we have a balance?

The best way to check a group of F9 equations is to use a Check total; that is, use one calculation method to generate the report, and another to generate a similar number for comparison. Please read *Check Totals* in the appendix *F9 Tips and Tricks* for more information.

The second step should be to to run an *Analyze for Access* on the spreadsheet. The report this generates should expose where accounts are repeated or omitted in the calculations. See *The Analyze Command* for more information.

Re-calc times seem very long. How can I improve them?

There is no simple answer to this question. The most important steps are to:

- Avoid the use of gratuitous wildcard characters. For example, if your account code includes a location, but you only have one location zero, then ask for location zero, not location "*" (all locations). The answer will be the same but the first function may re-calc much faster.
- Be sure your sheet is in manual re-calculation mode.
- Avoid getting the same number twice. For example, after using the GL function to get balances for a series of months, don't ask F9 to compute the total for the same period; let the spreadsheet sum the monthly values you already have.

New V4:

One of the principle rules of problem solving is to get the best focus on the problem as possible. Version 4 introduces a new analysis tool to evaluate Recalculation Times. Using this tool, all the F9 functions return an elapsed time for each call which allows you to see exactly where F9 spent "all that time." This tool should quickly expose the problem calls. See also *The Analyze Command*.

You can minimize the amount of time waiting for recalculation by taking total control over recalculation. If you are using Lotus 1-2-3, you can re-calc only portions of your sheet at a time. If you are using Excel, you can re-calc only the current active sheet by pressing Shift-F9. Thus, if you isolate your GL functions to one sheet in a workbook, you can control when they are recalculated.

Anything you can do to speed up disk access will help. This includes defragmenting your hard drive regularly. Disk caching programs can help a lot. Using a fast machine with a fast hard drive can make a large improvement.

While working in a windows environment, users must manage their virtual memory and swap file space. Under Windows 3.x, the recommendation was to use a virtual memory size that was twice the size of your RAM. Under Windows 95, the swap file is managed dynamically and commonly approaches 50-75 Mb, and the recommendation changed to specify a minimum free space *after the swap file* of approximately 50-60 Mb. Under Windows NT, the swap file commonly approaches 100 Mb and the recommended free space *after the swap file* is approximately 70-100 Mb.

If you are running a Btrieve-based accounting system on a Netware server, you might consider using the Btrieve NLM (ask your dealer about this).

Read the section on DataPreLoad in *Chapter 8: Advanced Features*. Depending on the individual situation, enabling DataPreload may speed calculation times dramatically.

Remove all zero suppression from the sheet so you can see how many formulae actually exist. Some users have found that a zero suppressed sheet looked incomplete, so someone inserted the “missing” functions which were actually hidden. After a few sessions, the worksheet made numerous accesses to accounts which return zero. In other cases, a worksheet was moved between companies and the hidden rows were never edited, so the sheet was always trying to access non-existent accounts.

How can I do scaling and rounding on a sheet by sheet basis?

Use your spreadsheet to do it! Put 1,000 (for example) in a control cell on your sheet and then divide the values returned by the GL function by that value. All three spreadsheets applications support a rounding function as well.

I can't seem to get the opening balance or net transactions for “period 1 last year”. Why?

F9 holds the closing balances for each period for two years only. Therefore, you can't get the opening balance for period 1 last year, or net transactions for that period. If your accounting system supports multiple years, however, supply a year parameter equal to the last year's number and ask for “period 1 this year” (the phrase “this year” is not actually required) where the year is the year supplied in the year parameter.

The balance value displayed by Drill and the value in the cell are different. Why?

There are a number of possible explanations: Drill only works on one account specifier at a time, so if you are using a comma-separated list of account specifiers, only the first is drilled. Drill only works on one company at a time (see previous explanation). If you are using the text substitution operator and have accidentally requested the same account twice, Drill will only display it once. In this scenario, the Drill balance displayed will be correct, but the Drill detail will not add up to it. Finally, if your Chart of Accounts has balance sheet accounts “mixed in” with P&L accounts, Drill and GL may not agree. As a rule, the GL function will not sum balance sheet accounts with P&L accounts.

When I type a simple function into a cell (like BSPEC or GL), I get a #REF error... even when the parameters are very simple. What is going on?

You have probably created a macro or a named range with the same name as an F9 function. Pressing F5 will show you a list of current named ranges. In general, do not create a named range or macro with the same name as an F9 function or command (*i.e.*, Setup and Chart are not allowed either).

Why do we drive on parkways and park on driveways, and for that matter, how come “slow up” and “slow down” mean the same thing?

That’s always bugged us here at Synex too.

If all else fails, and my problems still don’t go away, what should I do?

Read the manual. We have a separate appendix for each of the spreadsheets and answer some more FAQ and spreadsheet issues in those appendices.

Okay, but if I’ve already done that, then what should I do?

Some of our products are supported by the accounting system developer. If your product is one of those, contact your accounting system developer support staff. Even if your product is supported by the accounting system developer, you can still purchase support from us.

Visit us on the World Wide Web at www.f9.com and find the Support page. We keep an updated version of this document there, along with bug fixes and ideas, suggestions, templates. You can also call Synex for technical support and, one way or another, we will fix your problem — as long as it is F9 related, of course. Our numbers are:

(604) 688-8271 in North America.

Extension 347 or 333 for direct support; extension 314 for sales

Please be ready with your serial number. You will probably find it on the back of your manual and in the title bar of the DLL Help About box.



Appendix



C. Using F9 with Microsoft Excel

Installing the Add-in

An add-in in Excel is loaded just like an Excel spreadsheet. The F9 add-in for Excel is called F9.XLL. To load the add-in, select **File | Open** from the Excel main menu and load F9.XLL. It will usually have been installed in your Excel directory.

Tip:

Always have your F9 sheets in manual recalculation mode.

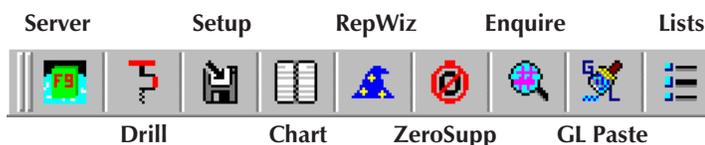
New sheets default to automatic. Use the **Options | Calculation** menu selection to set the calculation mode (**Tools | Options** in Excel 5). Excel has a Calculate Document (Shift-F9) option that only recalculates the active sheet. For larger GLs, this can be a handy feature. It allows you to place all your GL functions in one sheet in a workbook, and access the data the GL functions return in other sheets in your workbook. This gives you complete control over when the sometimes time-consuming F9 GL functions are executed. For more on recalculation modes and how to set them, see the section called *Manual Recalculation in Excel 5* later in this chapter.

Loading F9 Automatically

You can cause Excel to load F9 automatically by selecting **Tools | Add-ins | Browse** from the main menu. Locate the F9 add-in, named F9.XLL, and add it to the list of automatically loaded add-ins.

Customizing the F9 Tool Icons

The default icons used by the F9 Add-in for Excel are:



These may need to be corrected if the F9 Toolbar is “recreated” with this procedure:

1. Launch Excel and load F9.XLL (the F9 menu is available).
2. Use **File|Open** and load the workbook called F9BMP.XLW (Excel 5) or F9BMP32.XLW (Excel 7 and later) which you will find in the directory where you installed F9.
3. Click on each icon object on the sheet to change the appearance of the F9 toolbar.
4. Select **File|Close** Workbook

We also recommend that you move the Excel Recalculate tool to the F9 toolbar. This button recalculates your sheets when pressed. The Excel FAQ, later in this chapter, also includes tips on correcting problems with the toolbar.

Excel 97 and “Balloon Help”

The F9 toolbar buttons should now display their function when the mouse pauses over them. If they do not, you should load F9BMP32.XLS and press the “Press Me!” button. This spreadsheet can set the button hint text; F9BMP.XLS will not.

The Add-in Menu

Once the add-in is loaded, you have access to all the functions provided by any F9 DLL you have installed. In addition, you have access to the following commands under the F9 menu item:

F9 Server Select this option to access the F9 Server.

Drill Down	Position the cell cursor on a GL, NGL or GLTRAN function and select Drilldown to invoke Chart and view the accounts that make up a balance.
Setup	Select Setup to access the window for the default topic.
Chart	Select Chart to run Chart and send account data to the Clipboard.
Passwords	Select Passwords to enter password information for the default topic.
Toolbar	Unlike other commands, this command does not appear on the tool bar nor is it a DDE command to the current default topic or DLL. It is a toggle command to the Excel add-in to stop (or start) creating the F9 tool bar. When you select this command, a message will appear in the Excel status line indicating the tool bar create function status. In general, the tool bar will be created automatically only once. Excel will then “remember” it indefinitely. This command’s only purpose is to allow you to move the F9 tool bar icons to some other tool group if desired. If tool bar creation is on (the default) and the add-in does not detect a tool bar called “F9 Tools”, it will create one automatically. If you do wish to remove or hide the F9 toolbar, toggle tool bar creation off after you have done so.
Options	The Options dialog configures several F9/Excel features and communications parameters. See <i>Client Commands</i> in <i>Chapter 6: Commands and F9 Windows</i> .
DLL Help	Select this option to invoke the default DLL’s help system.
Server Help	Select this option to invoke the Server help system.
Set Topic	Select this command to set or reset the client default conversation to some other DLL.

Version	This command opens an "about" style dialog which reports the versions of the F9 Add-in, Server, and accounting system DLL. This information is useful for Technical Support.
Period Wizard	The Period wizard will leave a correctly formatted period specifier on the clipboard.
Report Wizard	The Report wizard is well documented in <i>Chapter 4: The Reporting Wizards</i> .
Budget	This command invokes a Budget Writeback window to write information from the clipboard to the General Ledger.
Zero Suppress	Highlight a range of cells and select this option to suppress (hide) zero rows in the range selected. F9 also provides a toolbar button for this same function.
Analyze	Analyze completes reports as a result of a recalculation: it verifies that accounts are being accessed properly or provide a report of the amount of time used for each F9 function call.
Enquire	The account enquiry window allows you to make a "snapshot" query of an account in your GL. It also leaves information on the clipboard for the GL Paste command to interpret and begin a report.
To Viewer/E-mail	This option invokes a re-calculation and conversion on the <i>currently selected area of the spreadsheet</i> .

F9 Functions

F9 adds a number of functions to those normally available within Excel. The most useful of these are GL, NGL and DESC. These functions can be reviewed using the Function Paste command or the Function Wizard and selecting functions from the F9 function set.

Tip:

Remember, in Excel you precede formulae with an equal sign (=). See the DLL's help for the general ledger you are using and the general remarks in this manual about available functions.

AutoSum

Don't use AutoSum (this is the button with the sigma or large 'E' on it) to create Sum functions in Excel. Excel recalculates the AutoSum area and with F9 functions this can take some time. In some versions of Excel, AutoSum triggers a sheet-wide recalculation. Users often mistake the sudden slowdown in response time as Excel freezing. This often leads to attempts to scroll or page around, which in turn often leads to a GPF (General Protection Fault) or some other catastrophic failure.

Reading Your Lotus 1-2-3 F9 Spreadsheets

Excel will not recognize GL and other F9 functions in a 1-2-3 spreadsheet. When you read the WK? file in, Excel will throw these functions away. Here is a suggestion for making migrating from 1-2-3 to Excel easier. Before you read the sheets with Excel, use 1-2-3 to edit the WK? sheet, and for some, or all, of your F9 functions, turn them into labels by editing the cell and adding a single quote to the front of the formula. For example, use Search and Replace to change @GL(into '@GL(.

Then, when you read the sheet into Excel, replace the quote with an equal sign and remove any "@" signs from the formula. DESC and GL functions will work without additional modification.

Another approach is the following: Use the Lotus Range function to change all GL functions into the function MAX, and all NGL functions into the function MIN and save your sheet.

Now read the sheet with Excel and the corresponding feature in Excel to change each MAX function back into a GL function and each MIN function back into an NGL function.

Writing Macros in Excel

In Excel, macros exist on separate sheets called: Macro or Module sheets. The following documentation explains in more detail how to write an F9 macro in Excel 4 format.

First, in Excel 5, select: **Insert | Macro**; and **Excel 4 Format** to create a macro sheet. Begin writing the calls to F9 in column B and make sure the last line of the macro ends with the statement:

```
=RETURN( )
```

This enables Excel to complete the macro successfully. To simplify running the macro, your next step is to give the macro a name. A trick in Excel that you can use is to place a logical name for the macro in the left adjacent cell. For example, if the first line of your macro begins at cell B5, then place its name in cell A5 as a label. Assuming the name is “JanSales,” set the cell pointer to the first cell (line) of the macro and then select: **Insert | Define Name**. Excel will automatically guess that the name you want to use is in the left adjacent cell which, in this example, is “JanSales”. Also click on the option **Command macro** in this window so that Excel knows it is a macro. Now, you can run this macro easily from any sheet as it has an identity with Excel.

But, you can go another step further. It is easy to draw buttons on sheets and assign macros to them. This simplifies the entire process of running several macros on the sheet or sheets. The Excel Drawing toolbar contains an icon to draw and size buttons. Once drawn, you can select **Assign Macro to Object** with the right mouse button and since you have already named the F9 macro, it will show up in a list of selections. Formatting the button or changing the macro it is assigned to is easy. Just point your mouse over the button and click on the right mouse button; a menu will then pop up allowing you to make changes.

In summary:

- macros in Excel exist on macro sheets and can access data on worksheets with external cell references;
- each macro is followed by an Excel return statement;
- macros can be given logical names which simplifies running them;
- you can add buttons to your worksheets and assign F9 macros to them;

F9 is supplied with a sample sheet called **F9Buttons.XLS** that contains many examples of F9 functions being run by macros. Many of our template spreadsheets have examples of macros that change the contents of cells for the purposes of creating new reports. This is a common way of implementing Executive Information System (EIS) functionality inside your F9 spreadsheets.

Note:

Do not create named ranges or macros with the same names as F9 functions or commands. They may have the effect of disabling the function or command.

Refer to your Excel manual for more information on creating and running macros.

Manual Recalculation in Excel 5 and '95

Note:

Executing the Report Wizard will also place the spreadsheet into manual recalculation mode.

You can create a macro that is always available that will put Excel into manual recalculation mode as follows: Select **Tools | Record Macro | Record New Macro | Options | Store in** in your Personal Macro Workbook. Also assign a key to the macro (for example, Ctrl-E).

Now record your macro by selecting **Tools | Options** and set the recalculation mode to manual and set the “Recalculate before Saving” option off.

Turn OFF the macro recording by selecting **Tools | Record Macro | Stop Recording** (or by hitting the Stop Recording button).

This macro can now be run anytime by hitting the assigned hot-key combination.

You can make the macro you have created run at sheet load time as follows: create a named range (**Insert | Name | Define**) that starts with “Auto_Open” and set the “Refers to” to the macro you have just created. It will have a name like “Personal.xls!Macro1”. This macro will now execute when the spreadsheet is loaded, forcing Excel into manual recalculation mode.

Manual Recalculation in Excel '97

Note:

Executing the Report Wizard will also place the spreadsheet into manual recalculation mode.

You can create a macro that is always available that will put Excel into manual recalculation mode as follows: Select **Tools | Macro | Record New Macro**. This will open the “Record Macro” dialog in Excel. You should enter a name for the macro, set the **Store Macro in** control to “Personal Macro Workbook.” Also assign a key to the macro (for example, Ctrl-E).

Now record your macro by selecting **Tools | Options | Calculation** and set the recalculation mode to manual and set the “Recalculate before Saving” option off.

Turn OFF the macro recording by selecting **Tools | Macro | Stop Recording** or by hitting the “Stop Recording” tool button.

This macro can now be run anytime by hitting the assigned hot-key combination.

You can make the macro you have created run at sheet load time as follows: create a named range (**Insert | Name | Define**) that starts with “Auto_Open” and set the “Refers to” to the macro you have just created. It will have a name like “Personal.xls!Macro1”. This macro will now execute when the spreadsheet is loaded, forcing Excel into manual recalculation mode.

Range Recalculations

It is handy to be able to only recalculate a range of cells when using F9 (Excel 7 or later only). To do this, create the following Visual Basic module:

```
Sub RecalcRange()  
    Selection.Calculate  
End Sub
```

We recommend that you place this module in your *Personal.xls* spreadsheet so that it is always available. Once created and saved, you can access this module in a number of ways to recalculate only a range of cells:

- You can run it with **Tools | Macro | Run**.
- You can add access to the module to the Excel Tools menu.
- You can create a new button on the F9 Toolbar and assign the module to it (recommended). To do this in Excel 7, select **View | Toolbars | Customize**. Then drag a new button up to the F9 Toolbar. Next, assign the *RecalcRange* module to the button. You may also wish to change the button’s “face” (right-click on the button).

Other Tips

Pressing F9 in Excel will recalculate all open workbooks in the Excel session. To recalculate the active sheet press Shift-F9 instead.

If F9 is not loaded when a spreadsheet using F9 is opened and recalculated, the cells containing F9 formulae will return #VALUE!. To attach F9 go to **Tools | Addins | Browse** and choose F9.XLL. This will load F9 as an add-in to Excel. Having done this, recalculate the spreadsheet.

Loading F9 through the **Tools | Add-ins** menu will put F9 in Excel's add-in list and F9 will be loaded into memory with Excel without reducing any of Excel's functionality. F9 takes very little memory, so it's a good idea to have F9 attaching automatically to Excel regardless of whether or not it will be used.

Excel FAQ

My Excel F9 toolbar contains a bunch of cards (i.e. the symbols for hearts, clubs, diamonds and spades), what gives?

These are the correct initial default tool faces in Excel. To make them more representative of their true functions, do the following:

1. Load the sheet called F9BMP.XLS (Excel 5) or F9BMP32.XLS (Excel 95 or later).
2. Press the large "Press me!" button.

You can also set the tool faces to any desired bit map manually. In Excel 5, click on the tool bar with the right mouse button; select **Customize**; click on the tool button you wish to change with the right mouse button; and select **Edit Button Image**.

I upgraded to version 4 and installed the add-in, but I don't have enough buttons. How do I fix it?

Two things are working against you: Excel remembers the settings you had on the last run including the toolbars. And, F9 only builds a new F9 Toolbar if one doesn't exist.

So, Excel remembers the V. 3.x toolbar and F9 won't create a V. 4.x toolbar while an F9 toolbar exists. Your buttons still work because the F9 functions do exist.

In **Excel 97**, the following steps will complete the update.

- Open your spreadsheet and delete the F9 toolbar by selecting **Tools | Customize** and then the **Toolbars** tab. In the listed toolbars, select the "F9 Tools." Press the **Delete** button to completely destroy the old toolbar. Press **Close** to close this dialog.

- Select **Tools | Add-ins** to open the Excel Add-in manager. In this list, select “F9 — the Financial Reporter” and *clear* the checkbox next to it. Press **OK** to close the dialog. You should now receive a “F9.XLL has been removed” dialog which you can close.
- Select **Tools | Add-ins** to open the Excel Add-in manager. In this list, select “F9 — the Financial Reporter” and *set* the checkbox next to it. Press **OK** to close the dialog. You should now receive a “Thank you for Adding F9.XLL” dialog which you can close. You should now have a new toolbar with hearts, spades, and other odd icons.
- Open F9BMP32.XLS from the F9 directory. *Do not disable macros* for the sheet. Press the button that says “Press me!” The macro activated by the button will replace all the images in the F9 Toolbar to match those on the spreadsheet and illustrations in the manual.

When I load the Excel add-in F9.XLL, Excel gives an error saying “Cannot read this binary file.” What have I done wrong?

You have installed the 32 bit add-in for Excel 7 in the Excel 5 directory, or the 16 bit add-in for Excel 5 in the Excel 7 directory. Either reinstall the Server being careful to indicate the correct Excel location and version, or you can copy the F9.XLL file manually from the appropriate subdirectory on the Server disk.

Note:

The spreadsheet and the Add-in must use the same memory model (*i.e.*, 16- or 32-bit). Version 4 of F9 *will* work with 16-bit spreadsheets using version 4 16-bit add-ins. The Version 4 F9 Server and your accounting system DLL are 32-bit.

Since I installed F9, Excel seems to GPF a lot more. Why?

There are two possible causes: using the mouse during recalculation and the AutoSum function. (AutoSum is actually covered in the next question.)

Excel is sensitive to the mouse during recalculations, specifically during external functions like F9 uses. The problem is that F9 spreadsheets can take a long time to recalculate, certainly longer than any of the internal Excel functions and during this time, a user may idly click the mouse in Excel which causes a GPF. During a recalculation, which can be a matter of a few seconds or minutes depending on your GL and the number of formulas, if the mouse is clicked on the Excel window it *may* cause a GPF. Sometimes it doesn't, but often it does. What can we say? Don't use the mouse during a recalculation because it causes GPFs.

Note:

This is a problem for all Excel add-ins, not just F9.

When I use the AutoSum feature in Excel, strange things happen. The machine seems to slow down or sometimes “crash”. What am I doing wrong?

Don't use it. When you do an AutoSum in Excel using the sigma button, Excel recalculates the cells being summed. This is unexpected for the user and can make Excel appear to lock up. User's often respond by trying to scroll or page around. This can make Excel GPF.

I just opened a spreadsheet in Excel and all the F9 formulae are returning #VALUE! in the cells where the numbers should appear. What did I do wrong?

You have opened an F9 spreadsheet without first attaching F9, so Excel does not recognize the formulae. Attach F9 and recalculate the sheet. If you thought F9 was attached but it no longer is, close the file and exit Excel. Then go back in and attach F9. Now open the spreadsheet and recalculate it. This should fix the problem.

This is also a common problem when you e-mail an F9 report to a user who doesn't have F9 or doesn't have the original GL data. In these circumstances, you should use the new F9 command **To E-mail/Viewer** to prepare the spreadsheet for sending. This will intelligently convert all the F9 formulas to values and save the formula parameters in the cell's comment. Users with the F9 Viewer and access to the original data can perform Drill-down analysis of the spreadsheet.



Appendix

D

D. Using F9 with Lotus 1-2-3

Installing the Add-in

A 1-2-3 for Windows add-in is loaded through the **Tools | Add-ins** menu or by using the ALT-F10 key combination. The F9 add-in for 1-2-3 for Windows is called F9.ADW. To load the add-in with the menu select the F9.ADW file from the list box and click on the load button. Using the old style Add-in ALT-F10 key combination, choose Load from the menu, select F9.ADW from the list and select Load. The F9.ADW add-in file will usually have been installed in your 1-2-3 for Windows Add-ins directory.

Installing and Customizing the Tool Icons

The current versions of the Add-in for 1-2-3 versions 5 and '97 include an F9.SMI file which installs and configures the icons.

To install any F9 icon in 1-2-3 version 5, follow this procedure:

- Launch 1-2-3 for Windows.
- From the menu, select **Tools | SmartIcons**.
- Use the mouse to press the Customize button.
- Choose icons from the Custom Icons Display and select the Add button to place them into the Current Palette.
- Use the Assign Macro button to add or change the macro the SmartIcon will invoke. The macros used for F9 are:

F9 Server

{F9}

F9 Setup

{SETUP}

F9 Chart

{CHART}

F9 Drilldown

{DRILL}

F9 Password

{PASSWORD}

F9 Topic

{SETTOPIC}

And, finally, press each OK button until you're back to the spreadsheet.

To create your own customized icon you can use the Windows Paint program to edit an existing icon or create a new one for Lotus 1-2-3 for Windows release 1 or use the new icon editing tool in release 4.0. If using Paint the image attributes have to be changed to a 21 by 21 PELS size. Use the View Zoom Command to do the editing. The image must be saved in windows bitmap (BMP) format.

Loading F9 Automatically

By selecting ALT-F10 and **Settings | System** Lotus 1-2-3 will let you specify up to 15 add-ins that you wish to load automatically every time you run Lotus 1-2-3. Select **Set**, and then select F9.ADW.

Note: Auto-invoke will be set to No. This is normal.

Loading F9

{ADDIN}L{ESC}F9~~~ (note: you may need two or none of the {ESC} commands and a complete path to the F9.adw add-in file)

Removing F9

{ADDIN}RF9~

The Add-in Menu

Once the add-in is loaded, you have access to all the functions provided by any F9 general ledger DLL you have installed. In addition, you have access to the following commands under the F9 menu item:

F9 Server	Select this option to access the F9 Server.
Setup	Select Setup to access the Setup window for the default topic.
Drilldown	Position the cell cursor on a GL, NGL or GLTRAN function and select Drilldown to invoke the Drilldown View and examine the accounts included by F9.
Chart	Select Chart to run Chart View and send account data to the Clipboard.
Password	Select Passwords to enter password information for the default topic.
Toolbar	Builds the F9 Toolbar by installing the F9.SMI file.
Set Topic	Set the default GL topic for the client add-in.
Version	Select Version to display a dialog with version numbers and dates for troubleshooting.
Recalc All	Recalc All will recalculate all of the F9 functions in the current workbook. The two Recalc commands are required to address instability issues with the later versions of Lotus 1-2-3.
Recalc Range	Recalc range will recalculate all the F9 functions in the current selection. The two Recalc commands are required to address instability issues with the later versions of Lotus 1-2-3.
Period Wizard	The period wizard assists you to build properly formatted period specifiers.

Report Wizard	The Report Wizard is a tool for building well formatted F9 reports with a minimum of typing. See <i>Chapter 4: Reporting Wizards</i> .
Budget	Select Budget to open the Budget Write Back dialog. See <i>Chapter 10: Budget Write Back</i> .
Zero Suppress	Highlight a range of cells and select this option to suppress (hide) zero rows in the range selected. See notes below.
Undo Zero Suppress	This function is re-opens all the rows in the spreadsheet. It is designed to remove the effect of the Zero Suppress command (above).
Analyze	This command performs one of two analysis functions. This command can return a report of account accesses performed during a recalculation <i>or</i> each F9 function can return a processing time for each request.
Value	This function removes F9 formulae from the current selection. That is, cells containing GL and other F9 formulae are “range-valued”. Use this function to prepare a spreadsheet for use by someone who does not have F9.
123 Setup	123 Setup (with 1-2-3 ‘97) allows you to configure column- and row-wise zero suppression.
Lists	The lists dialog is a tool for extracting various lists from your accounting system.
Enquire	Account Enquiry allows you to check a GL balance or configure the result of a GL Paste. See <i>Chapter 4: Reporting Wizards</i> .
GL Paste	Interprets the contents of the clipboard to form a proper GL formula. May execute an Account Enquiry in the background. See <i>Chapter 4: Reporting Wizards</i> .
Help Add-in Help	Opens the generic help file for using F9.

Help | GL DLL Help Opens the help file specific to the current default GL (Topic).

Zero Suppress

Highlight a range of cells and select this option to suppress (hide) zero rows in the range selected. F9 also provides a toolbar button for this same function.

Please Note: Unlike Excel or Quattro, Lotus 1-2-3 *prior to 1-2-3 '97* will not hide an entire row. Only the area highlighted will be hidden, even though the entire row will be collapsed. Not including all of the numbers and text in a row may result in an area of collapsed but unhidden cells. All Lotus 1-2-3 versions hide Zero Suppressed columns.

When using 1-2-3 for '97, the menu item **F9 | 1-2-3 Setup** allows you to configure whether F9 suppresses columns or rows or both. This is a simple selection in a dialog.

Invoking F9 Commands

To invoke an F9 command, either select F9 from the 1-2-3 for Windows main menu and then select the desired command; or select the appropriate tool icon. The F9 tool icons either display the first letter of the name of the function they invoke or a symbol such as the drill for drilldown.

F9 Functions

F9 adds a number of functions to those normally available within 1-2-3 for Windows. The most useful of these are GL, NGL and DESC.

See the DLL Help for the general ledger you are using and the general remarks in this manual about available functions.

1-2-3 '97:

Under Lotus 1-2-3 '97, all functions must provide all ten parameters. The parameters may be empty, but must be present.

Lotus 1-2-3 *prior to '97* added a tag to functions which indicated the add-in providing the function. *i.e.*,

```
<<F9>>@GL("100010", "This month", "My Company")
```

This has normal behavior and is harmless. With 1-2-3 '97, however, Lotus has discontinued this behaviour.

When importing older spreadsheets, the Lotus 1-2-3 '97 Add-in will edit the F9 formulae to provide all of the parameters correctly and removes the tag, so older worksheets should function without editing.

All Lotus Users

Lotus 1-2-3 for Windows is unstable during cell edits and copy operations when F9 functions are involved. The errors occur due to the way Lotus1-2-3 interacts with F9 and Dynamic Data Exchange. To address this instability, we have added two menu items to the Lotus 1-2-3 F9 menu: **Recalculated** and **Toggle Hold Recalculated**.

Recalculated allows you to only recalculate a cell or range of cells on your sheet. This can be very useful as your sheets get larger.

Selecting **Toggle Hold Recalculated** to turn Hold Recalculated ON causes F9 to return the string “[F9] Recalculated” for all functions until either the sheet is recalculated by pressing the F9 key, or until the cell is recalculated using the **Recalculated** range menu item. When this option is ON, the instabilities of Lotus 1-2-3 during cell editing and copying of F9 formulae is addressed. Selecting this option again turns the option OFF. When this option is OFF, F9 recalculates a formula on entry and copying.

Please Note: Make sure that F9 is attached before opening a file which uses F9. Failing to do so will generate errors (@? formulae) in the cells containing F9 formulae. If this occurs, close the file without saving it, attach F9 and reopen the file. It is important that the file is not saved. If the file is saved with the @? functions, all F9 functions (@GL) will be deleted and the spreadsheet will no longer recalculate.

1-2-3 Release 4 Users

In Release 4 of 1-2-3 for Windows, if you use the Lotus 1-2-3 menu **Edit|Copy** and **Edit|Paste** to copy F9 functions in your sheet, Lotus leaves the cursor as an hourglass after the Paste operation and it will no longer function on the sheet until you: press the Escape key; press any arrow key; or click on the menu bar with the mouse. This does not happen if you use the “Lotus Classic” function /COPY (and this method is also a lot faster)!

Writing Macros in Lotus 1-2-3 (v.4 or v.5)

A macro is a series of commands placed in a column of cells in a worksheet. When invoked these command will execute until an empty cell or a {RETURN}macro command is encountered. It is best to have a macro terminate with the {RETURN} command as it makes sure a definite ending point of the macro has been defined.

Typically, if you are defining a frequently used macro, the macro should be named by applying a block\range name to the entire range of cells that the macro occupies, or the first cell in the macro. It is a good idea to place the macro name beside the first cell of the macro for documentation purposes. Naming a macro allows you to call the macro by name from other macros. That is, if you call your macro ABC, another macro can call it with the command {ABC}.

Macros are run in two ways: You can use the Tools menu and select the Macro submenu. Alternatively, the macro can be assigned to a toolbutton or SmartIcon (a symbol that represents spreadsheet information or function(s)). When you click on the icon, the macro will be run.

To assign a macro to an icon you can use the SmartIcon dialog found under the Tools menu to assign the macro to an icon. See the 1-2-3 manual for instructions on creating smart icons and assigning macro commands to them.

In summary:

- macros are groups of commands that execute sequentially in a column of cells from top to bottom;
- each macro ends with a {RETURN} statement or an empty cell;
- macros can be given logical names which simplifies running them;
- you can add icons to your setup and assign F9 and other macros to them;

Refer to your spreadsheet manual for more information on creating and running macros.

Setting the DDE Time Out

By default, Dynamic Data Exchange requests sent by the client add-in to the Server will time out after approximately 300 seconds if no response is received. For large charts of accounts, this value may be too low.

When using 1-2-3 '97, the DDE timeout can be configured easily by selecting **F9 | 1-2-3 Setup** and editing the amount in the edit control. This number is then saved in the correct location.

You can set the DDE Time Out by adding the following line to the Lotus 1-2-3 section of the F9.INI file:

```
DDE Time Out=nnn
```

where nnn is the time value in seconds.

Lotus 1-2-3 FAQ

I am converting my Lotus 1-2-3 for DOS sheets into Excel sheets but when Excel reads the Lotus 1-2-3 sheets, all my GL functions disappear. What can I do?

Refer to the Excel chapter of in this manual (*i.e.*, see *Appendix C: Using F9 with Excel*) for an explanation on how you can work around this problem.

I have just pasted some formulae into Lotus 1-2-3, but the hourglass cursor won't go away. What is going on?

We don't know. This is a Lotus bug, but pressing the Escape key fixes it!

When I type a formula into Lotus 1-2-3, this shows up in it: <<F9>>. What is going on?

Don't Panic. Lotus is "tagging" the formula as belonging to F9. It is normal and harmless. This behavior is discontinued in Lotus 1-2-3 '97.

I upgraded to Lotus 1-2-3 '97 (or later) and it won't take the usual GL formulae (or any other F9 formulae). What's the problem?

Lotus 1-2-3 for '97 made some radical changes and imposes a few more restrictions on F9 users. The biggest one is that for the almost all of the F9 functions, ALL ten parameters must be in the formula, though they can be empty. Some functions take fewer parameters (*i.e.*, CODATA takes only two).

What you are probably trying to do is use the “old way” of putting just the first few parameters in and Lotus is saying “No, you need to tell me more.” The correct method is to enter:

```
@GL(Account,Period,Company,Year,Type,Currency,Topic,"","")
```

The only *required* parameter is the first, the Account specifier. For 1-2-3 '97, you need to pad the function out to:

```
@GL(Account,"","","","","","","","")
```

or, for functions such as CODATA, which take fewer parameters:

```
@CODATA("Name","")
```

What can we say? You must provide ALL parameters to Lotus '97 and later.

The F9 spreadsheet that I created in Lotus now has @? functions instead of @GL functions in it. What happened to the F9 formulae?

You opened a spreadsheet and recalculated it without attaching F9. Close the spreadsheet and DO NOT save it (since once you save a spreadsheet with @? functions you lose all the @GL functions). Attach F9, open the file that had the @? functions in it and recalculate the spreadsheet. This should take care of the problem.

Sometimes Lotus 1-2-3 crashes with a General Protection Fault, especially when I am copying F9 formulae. Why does this happen?

We believe this is caused by problems with “re-entrancy” (*i.e.*, code that is re-entered at an inappropriate time) that are associated with Dynamic Data Exchange. If, however, you use the Lotus 1-2-3 classic menus to copy data, not only does it work faster, it seems more stable. To completely eliminate these problems, toggle Hold Recalculated off from the F9 pull down menu. This causes cells to only be calculated during a re-calc, where this problem does not occur.



Appendix

E

E. Using F9 with Quattro Pro Vers 5 to 7

Note:

F9 Version 4 add-in is not available for Quattro Pro at this time. Only version 3 features are made available within the spreadsheet.

You can use Quattro Pro for Windows immediately by identifying the F9 functions with the prefix “F9.” That is, instead of entering GL or BSPEC you would enter: **F9.GL** and **F9.BSPEC**

In using the F9. prefix Quattro Pro will automatically load the F9 add-in (actually a DLL) and access the appropriate functions. However, we strongly recommend that you take the following steps to complete the installation of F9 for Quattro Pro.

Installing F9 for Quattro Pro

If you installed F9 using the install procedure, your QPW.INI file will have been modified automatically. If the install completed properly, when you launch Quattro Pro in Windows, you should see the F9 speedbar in Quattro Pro, and you should see the F9 Server get launched (the F9 icon will appear at the bottom of the screen) as well. If the installation did not complete properly, you may modify the QPW.INI manually as follows:

Add the following lines to the QPW.INI file you will find in your Windows directory:

```
InitMacro={DLL.LOAD F9}  
CustomSpeedBar1=C:\QPW\F9MENU.BAR  
SpeedBar=F9MENU.BAR
```

The first line ensures that the F9 DLL is loading automatically when you launch Quattro Pro. A major advantage of loading F9 in this way is that you will not have to prefix F9 functions with “F9” as described above. If you already have an InitMacro line, append the F9 macro command to it.

The next two lines load the F9 speed bar which contains buttons for accessing the various F9 functions and menus. You **must include** these second and third setup lines in this order for the F9 speed bar icons to work. The “CustomSpeedBar” line should be modified according to where your copy of F9 for Quattro Pro was installed (usually in the Quattro directory). These initialization lines may already exist in your QPW.INI file. In this case add a new custom speedbar as: “CustomSpeedBar2=” or a number one greater than the last listed. If a speedbar entry already exists, add the F9 custom speedbar with a preceding comma after it:

```
SpeedBar=FIRST.BAR,F9MENU.BAR
```

These lines can be added to the QPW.INI file with any text editor. If you are running DOS 5 or greater use EDIT. That is, change directories to your Windows directory and type:

```
EDIT QPW.INI or EDIT QPW6.INI
```

and then add the above initialization lines near the end of the QPW.INI file.

To attach the F9 speedbar menu to the Quattro menu do the following:

1. Right click a non icon portion of the speedbar.
2. Select Append from the menu that pops up.
3. Select F9MENU and select the F9MENU.BAR file from the display. If the F9MENU item does not appear, you may have to change directories (*i.e.*, if the F9MENU.BAR file is not located in Quattro’s default directory).

Report Wizard and Analyze

These functions are not available with the Quattro Pro add-in.

Recalculation Modes

We recommend that you make sure that recalculation is set to manual for all your spreadsheets. Click the right mouse button with the mouse pointer on the notebook (spreadsheet) title bar and set the recalculation mode to manual. If the recalculation mode is not manual, Drill will not work and sheets will be recalculated each time an edit is made.

Speed Bar Icons

The speed bar icons perform the following functions (from left to right):



Invoke the F9 Server;



Invoke the default DLL Setup window;



Invoke the default DLL Passwords window;



View the most recent function error;



Invoke the default DLL Chart window;



Invoke the default DLL Drill window;



Invoke the F9 Command Menu.

The Array Function (pre Version 3.1)

Although you need not concern yourself with this, the GL function actually returns a so-called “array value”. Quattro Pro is aware of this and, in order to interpret it properly, it automatically wraps an Array function around the GL function whenever you enter one. Don’t worry about it, and don’t try to delete it. Just enter GL functions into each cell without regard for the Array function.



Most Recent Error

Error Messages (post Version 3.1)

For post-Version 3.1 Quattro Pro Add-ins, unlike the add-ins for Lotus 1-2-3 and Excel, the GL function will not return an error message into a cell when something goes wrong. Rather, an ERR value is returned. A button is supplied on the F9 toolbar that will display the most recent GL function error message.

Reading Your Lotus 1-2-3 F9 Spreadsheets

Quattro Pro will not recognize @GL and other F9 functions in a 1-2-3 spreadsheet. When you read the WK? file in, Quattro Pro will throw these functions away. Here is a suggestion for making migrating from 1-2-3 to Quattro Pro easier. Before you read the sheets with Quattro Pro, use 1-2-3 to edit the WK? sheet and, for some or all of your F9 functions, turn them into labels by editing the cell and adding a single quote to the front of the formula.

Then, when you read the sheet into Quattro Pro, just delete the quote. DESC and GL functions will work without additional modification. CONAME and other functions will require some additional changes. For example, CONAME(Company) becomes CODATA("Name",Company) . After setting up a few prototypical functions, you should be able to copy your functions around the sheet to recreate your reports.

Writing Macros in Quattro Pro for Windows

A macro is a series of commands placed in a column of cells in the worksheet. When invoked, these commands will execute until an empty cell or a {RETURN} macro command is encountered. It is best to have a macro terminate with the {RETURN} command as it makes sure a definite ending point of the macro has been defined.

Typically, if this is a macro you will be using frequently, the macro should be named by applying a block/range name to the entire range of cells that the macro occupies or the first cell in the macro. It is a good idea to place the macro name beside the first cell of the macro for documentation purposes. Naming a macro allows you to call the macro by name from other macros. If you call your macro ABC another macro can call your macro with the command {ABC}.

Macros are run in two ways: First, you can use the Tools menu and select the Macro submenu.

or

Second, the macro can be assigned to an icon (a symbol that represents spreadsheet information or function(s)) or a control such as a button. When you press on the icon/control the macro will be run.

To assign a macro to an icon you can use the UI Builder tool. See *Chapter 6, Dialog Boxes and Speedbars* in the Quattro Pro for Windows *Building Spreadsheet Applications* manual for instructions.

In summary:

- macros are groups of commands that execute sequentially in a column of cells from top to bottom;
- each macro ends with a {RETURN} statement or an empty cell;
- macros can be given logical names which simplifies running them;
- you can add icons and controls to your setup and assign F9 and other macros to them;

Refer to your spreadsheet manual for more information on creating and running macros.

Setting the DDE Time Out

By default, Dynamic Data Exchange requests sent by the client add-in to the Server will time out after approximately 300 seconds if no response is received. For large charts of accounts, this value may be too low. You can set the DDE Time Out by adding the following line to the Quattro Pro section of the F9 INI file:

```
DDE Time Out=nnn
```

where nnn is the time value in seconds.

Quattro Pro FAQ

When I type a GL formula into Quattro Pro and press enter, suddenly an Array function appears wrapped around it. What is going on?

Although you need not concern yourself with this, the GL function actually returns a so-called “array value”. Quattro Pro is aware of this and, in order to interpret it properly, it automatically wraps an Array function around the GL function. Don’t worry about it, and don’t try to delete it.



Appendix**F**

F. F9 Templates

Your F9 product is normally shipped with up to three types of templates:

Accounting System Specific Templates

The first set of templates are actual working F9 spreadsheets that can be recalculated versus the sample data of the accounting system in question. At a minimum, you will be supplied with four sheets: BALANCE (a balance sheet), INCOME (a simple P&L), TIMESLIC (a P&L using the GLTRAN function) and ONESTEP (an education sheet illustrating the use of the GL function). If the accounting system's sample data supports it, an additional sheet is supplied called DEPT_INC (a departmental income statement).

As mentioned, after you have installed and set up F9, and assuming that you installed the accounting system's sample data, these sheets can be recalculated against that sample data to produce reasonable results. These sample sheets were produced specifically for the accounting system you are using.

You may use these sheets (except ONESTEP) as templates for constructing your own reports. In order to make these sheets function with your data, you must follow the procedures (with a few exceptions) as outlined below for the "generic" templates.

Generic Templates

The second set of templates are the "generic" F9 spreadsheets (as opposed to the accounting system specific spreadsheets described above). They will **not** recalculate "out of the box", but in most cases they are not very far away from being functional spreadsheets that will work with your data.

The generic templates are designed to work with a hypothetical accounting system that uses a seven part account code. Most of the templates can be thought of as one slice through your general ledger.

Tip:

Most financial reports will present the Account as the first vertical dimension of the slice and one of the other dimensions for the horizontal dimension.

Your reports will examine accounts with axes like: Company, Actual/Budget; Period; Year; Product; Department; *etc.*. For example, if you report amounts by Account and Department, you might be building a departmental income statement.

To make this template work with your data, the major part of the work is to indicate the accounts you want down the side of the report and the departments you want across the top of the report. These account and department codes will, of course, be unique to the chart of accounts you are using.

Each report contains an input area (input areas are usually highlighted in blue) near the top of the sheet that contains entries for the other dimensions of the general ledger data. Using our example above, this area would encompass cells that contain values for the company, year, period, *etc.*

Supplied Templates

When you installed F9, all templates were installed by default (you may have stored them elsewhere) in a subdirectory called F9.

The accounting system specific templates are called BALANCE, INCOME, ONESTEP, TIMESLIC and (possibly) DEPT_INC.

Various other generic templates are provided. We are always working on refining these templates and so they are not documented in detail in this manual. At a minimum however, you will find six templates that represent the six slices that are the combination of the main account code with the remaining six segments of the example seven part account code.

Each segment of the account code can, of course, represent anything. Our example templates have text in the spreadsheet that implies that the segments are Account, Location, Department, Group, Product, Profit Center, and Sales Person. But that is all it is — just text. After you have followed the directions in the next section to make a particular template work with your data, change the text in the spreadsheets to accurately reflect your account code structure.

The F9 subdirectory will also contain a file called TEMPLATE.WRI which contains a list of the supplied templates and a brief description of each one.

Making the Templates Work With Your Data

The generic templates we supply, and the instructions that follow, make one assumption about your accounting system: that the main account code (sometimes called the natural account code) is the first segment of the full account code. *If this is not the case, an additional step is required in order to make the template sheets function.* That extra step is described later in this section.

The general steps necessary to make these sheets work with your data are as follows:

1. Make sure the spreadsheet is in manual recalculation mode.
2. Enter the appropriate company code.
3. Next, you should fill in the column headings of the report. If the report you are creating is a multi-period income statement, you may not have to change anything. In our example above, the columns were departments. The easiest way to create a list of departments to occupy the horizontal row is to use the Lists window (see the description of the Lists window in the *Server and DLL Windows* section of the manual).
4. You now need to create the row headings: a list of main segment values. Again, use the Lists window to create a list of the

appropriate natural account codes and paste them into the appropriate column of the template (e.g., normally column A).

5. You will also need to modify the values contained in the main input area to values that are appropriate for your site. Generally, the other account segments will default to “*” for all values.
6. Now recalculate the spreadsheet. If you have done everything correctly, correct numbers should appear.
7. Finally, you may wish to do some “clean up”. Some examples of clean up are:
 - Edit the BSPEC function to only include cell references to those segments of the account code you are using and then copy the new BSPEC function through the sheet.
 - Delete the rows or cells in the input area at the top of the spreadsheet that you are not using.
 - Delete any unwanted columns in the spreadsheet.
 - Consolidate lines in your report if desired. The technique just described will produce a detailed P&L (in this example). That may not be what you want. For example, you may have accounts in your chart of accounts for pencils and erasers, but may only wish to see “Office Supplies” in your report.

Lines can be consolidated in many ways including using wild cards and ranges. In the most simple case, you could edit the spreadsheet so that instead of having two lines that read:

5010	Pencils	\$100
5020	Erasers	\$200

you would have one line that reads:

5010,5020	Office Supplies	\$300
-----------	-----------------	-------

Changing the BSPEC Function

If your accounting system does not have the main account code as the first segment of the account code, you will have to change the BSPEC functions we supply. Each BSPEC function has as its second parameter a cell reference to the account code at the left of the spreadsheet. A typical function for a three part account code might be (using named ranges for clarity):

`BSPEC(Mask,Account,Department,SubAccount)`

If, however, the main segment of the account code is the middle segment, this must be changed to:

`BSPEC(Mask,Department,Account,SubAccount)`

After this change is made to one function, you may copy it through the spreadsheet in the usual way.

The Button Spreadsheet

Buttons can be associated with all kinds of functionality. Many of our templates contain buttons to perform simple manipulations in the spreadsheet (such as changing the word Actual to Budget).

Buttons can also be associated with F9 functionality using the Silent Running mode. For example, a button can be designed to fetch all your budget data and place it in the current spreadsheet, and another to send that data back again. Or a button can be built to drill a particular cell containing a GL formula and display its transactions.

Examples of these types of buttons and their associated macros are contained in the sheet called BUTTON.



Appendix



G. Troubleshooting Btrieve

F9 and Btrieve

Version 4 of F9 introduces a number of features which rely on the Btrieve database engine. All products under version 4 will ship with Btrieve components. This appendix is assembled to help resolve the most common Btrieve errors.

The instructions that follow will help with most situations. In the cases that do not work, there are additional resources at two websites:

Pervasive Software	http://www.pervasive-sw.com
Smithware	http://www.smithware.com

Links to these sites can be found at our site, <http://www.F9.com>, on our support page. We highly recommend the configuration matrix at the Smithware site.

All of the Btrieve files required to use F9 are installed into the F9 installation directory. The default location for this is C:\F9.

Unknown Btrieve Error

One of the most common errors is UNKNOWN BTRIEVE ERROR. The cause of this message is F9 finding the wrong version of WBTRCALL.DLL. This usually means at least one incompatible version is available in the path. F9 can work with a number of Versions of this file:

- 16KB file works.
- 42Kb file works
- 51Kb file *will not work*.

If you are using one of the Smithware matrices, it is recommended that you use the same or later version of the file listed.

To resolve any unknown Btrieve errors search the system path for WBTRCALL.DLL and any duplicate or old copies of the file rename. Once you have renamed the extraneous copies, restart windows. The extra copies of WBTRCALL.DLL can be usually found in the windows and windows\system directories

Tip:

You can find all the versions using the Start|Find|Files or Folders feature of Windows. Avoid searching My Computer when you are on a network as this is time consuming and generates a very significant amount of network traffic.

Btrieve error 4 or 11 or 12

Btrieve error 4 refers to Btrieve being unable to find a key value within the index.

Btrieve error 11 means the specified value is invalid.

Btrieve error 12 means Btrieve cannot find the specified file.

The above errors can be resolved by checking the path to the data. The most common cause of these messages is that the path specified in F9 Setup is incorrect. If the file or directory is present, (*i.e.*, it can be seen through windows explorer or the DOS prompt), ensure that you have rights to the data files or directories.

Btrieve error 20 and a DOS accounting system

The following procedure is for Btrieve configuration for the following accounting systems.

- Solomon III (Solb DLL)
- AMSI
- GP7, 8 or 9
- MACOLA 6
- Platinum
- Realworld 6.5 and 7 (Btrieve only)

The accounting systems use Btrieve for DOS. There should be no WBTRCALL.DLL in the accounting system directory.

All Btrieve for Windows components are shipped with F9.

Make sure the BTI.INI file has requester set to YES.

Load Brequest from the F9 directory in WINSTART.BAT

Search the hard drive so that there are no WBTRCALL files in the Windows or Windows\System directories. If there are, simply rename them (ie. Wbtrcall.old). DO NOT DELETE THEM.

Restart the machine after making all these changes.

This will take care of Btrieve error 20 and any lockout issues.

Editing the BTI.INI file

The BTI.INI file resides in the Windows directory on your local hard drive. In the F9 directory, there is an utility called DBSETUP. DBSETUP allows you to edit your BTI.INI file. Double click on Dbsetup through Windows Explorer. Select Options, then CHANGE BTRIEVE SETTINGS. You will find a section called ENGINE USAGE. It is here that you change the settings for the LOCAL (client) and REQUESTER.

Save the changes and restart your machine.

Editing the registry

The Btrieve utility called "BTI Database Setup Utility" is recommended when changing the 32-bit Windows registry , W32MKSET.EXE. The program can be found in the F9 directory. Double click on this program.

In the first window select the COMPONENT drop down list. In the list select MKDE INTERFACE.

The window will change. In the area that has SETTINGS select either LOCAL (client) or REQUESTER. You then have the option to change the setting from YES to NO. Make the changes according to the matrix or instruction from F9 Technical Support. Restart Windows after you have made ALL the changes.

Lockouts

Occasionally some users will be prevented from accessing the accounting system while someone else is using F9. The cause of the situation can most frequently be tracked to differences in the BTI.INI file and in some cases the registry. Btrieve files have to be accessed *in the same way by all users*.

Ensure that all people are accessing the Btrieve files in the same manner by ensuring they are all using the same setting for LOCAL or REQUESTER. Simply check the settings in the BTI.INI file and registry on ALL machines accessing the Btrieve files.

Btrieve Error 20

The Record Manager Is Inactive or Requester Is Inactive

The error messages mean either that the record manager is not loaded or F9 was unable to establish a connection with the Requester.

To resolve the error 20, we recommend a matrix compiled by Smithware.

The matrix is extremely successful in resolving Btrieve problems. The matrix can be found at the Smithware site, www.smithware.com, or specifically http://www.smithware.com/support/btr_matrix.html. A link to this matrix is maintained on our website www.F9.com. Follow the Support link and then the link to Btrieve. Another matrix can be found at <http://www.pervasive-sw.com/>.

Reading the matrix

First select the operating environment:

- Windows 3.1,
- Windows 95, or
- Windows NT

Select the the network operating system:

- Novell Netware or
- Microsoft Windows NT.

Last, Select the 32 bit Btrieve Application: F9

The matrix lists a number of files that have to be in the F9 directory, also referred to as “the CLIENT.” It also lists settings for the REGISTRY and BTI.INI file. If you have the files in the F9 directory and the settings in the registry, and BTI.in are correct, your Btrieve problem should be solved.

Btrieve error 88

Btrieve error 88 can be resolved by changing the following line in the BTI.INI file. The lines should be as follows.

```
[Btrieve Client]
SharingOnLocalFiles=SingleEngine
SharingOnRemoteFiles=MultiEngine
```

See also:

<http://www.pervasive-sw.com/support/technical/papers/btrcitr.html>

Appendix

H

H. Glossary

Absolute Cell Reference	Returns the contents of a cell identified by row and column coordinates. For example, an absolute cell reference appearance might be \$A\$1 or \$D\$18. Absolute cell references do not change when the formula is copied.
Acrobat Reader	Adobe Acrobat is a type of graphics-rich electronic document. F9 ships with the latest copy of the manual on the CD or a diskette in Acrobat format (with the extension of PDF). By installing the Acrobat Reader from the Reader Disk , this document can be read, searched or printed. The latest Acrobat reader can be found at http://www.adobe.com/acrobat .
Ampersand operator (&)	This operator is used to dynamically recalculate a company using a chart of accounts from the MAP file. The name to the left must be a parent company in the Map file and the operator is followed by a list of the parent's subsidiaries
Analyze Command	Allows you to evaluate errors and omissions in the accounts accessed by a report. Produces a report of how many times each account was used and a list of unused accounts.
Block	A group of data with a beginning and an end usually occurring within a larger, sorted amount of data; an area of columns and rows in a spreadsheet, for example.

Budget Write-Back	Lets you create and/or edit budgets in the spreadsheet and then write that budget data back to the accounting system.
Cell	The intersection of a column and a row in a spreadsheet. For example, the intersection of column C and row 15 is referred to as C15. Each cell is a box that can hold text, a numerical value, or a formula.
Cell Reference	A cell reference is using the name of a cell in place of the value in the referenced cell. The advantage, within F9, is that any value needs to exist once and many cells can refer to that cell; this allows the single cell to control the entire sheet. <i>See also</i> BSPEC in <i>Chapter 5: F9 Functions</i> .
Character String	Also called a String, Text String or a Label. A series of characters treated as text. Character strings <i>can</i> contain numbers as well as letters but you don't generally perform math with them.
Chart Window	Lets you view your chart of accounts. Lets you send all or part of your chart of accounts to the Clipboard for subsequent pasting into your spreadsheet. Useful for diagnosing problems.
Client Add-Ins	An add-in is a special file which is attached to your spreadsheet program that extends the power of that spreadsheet by adding a set of special functions and commands. The F9 client add-in manages the DDE conversations with your Server while it works in the background.
Clipboard	Temporary storage in Windows which accepts data that may be moved or copied to an active document, another document or other Windows applications. Very useful when transferring information within a document or between documents, however, the clipboard only holds one item (<i>e.g.</i> , block of text) at a time.

Command	Well defined methods which do not return a result but are used to perform actions. For example, a command can select a region of a spreadsheet, print a report, or open a dialog. <i>See also</i> Function.
Consolidate	To combine amounts from the chart of accounts of two or more companies for financial statements.
Conversion Rate	<i>See Rate of Conversion</i>
Data Manipulation	The processing of information. The retrieval, sorting, modifying, filtering and querying of data are a few examples. Data must be present within a file or database for data manipulation to occur; it does not involve entering new data. The creation or deletion of new files, however, is considered part of data manipulation.
DataPreLoad	Improves recalculation speed by pre-loading extended portions of the G/L data into your computer's memory. It gives you the option to fine tune recalculation to take advantage of your hardware configuration.
DDE	Stands for <i>Dynamic Data Exchange</i> . It is the way in which Windows applications talk to each other. The information linked using this feature will update the destination document any time the information changes in the source document. Allows you to use one program to manipulate data in another program.
Default Topic	Required when you have more than one accounting system DLL. When a DDE request is sent to the Server and no topic is specified, the Server will use the client default topic.
DLL	Stands for <i>Dynamic Link Library</i> . This is your topic or accounting system. Each DLL receives requests from the Server (and hence from the client) for data specific to its associated accounting system.

DLL Disk	This disk contains all the accounting system specific files required to communicate GL data to your spreadsheet application. See also Server Disk.
Drill Down	This F9 window lets you “slice and dice” any F9 GL/NGL result in your spreadsheet. You can determine the origin of any number by segments of the account code (<i>i.e.</i> , department, product, or location) or transaction.
Exchange Rate	See <i>Rate of Exchange</i>
Explorer	The Windows '95 and later tool for managing files. This tool unifies the Program Manager and File Manager from Windows 3.x.
exclamation operator (!)	This operator is used to dynamically recalculate from a company in the Tree file. The company name following the operator must be a master in the Tree file.
Executive Information System (EIS)	A tool which automates a great deal of the process of viewing, analyzing, or printing reports. F9 supports EIS applications through spreadsheet macros.
F9	The Key which invokes recalculation in Excel, 1-2-3, and Quattro Pro. It's also the name of our product.
Function	<p>A well defined method which returns a result and is used to perform mathematical, financial or statistical calculations. For example, the SUM function adds all the cells in an area together, the AVG function returns the average number in an area, and the GL function returns an amount from your General Ledger. <i>See also</i> Command.</p> <p>F9 functions never cause a window to appear or open any type of dialog box.</p>
FAQ	The most Frequently Asked Questions. FAQ can be pronounced as an acronym (<i>ef-eh-que</i> , our preference), or like “fact.” An FAQ is a document which originated on the World Wide Web (WWW). Wherever users

congregate, it is suggested that new users “*lurk*” (look, listen, and learn) until they get the FAQ and read it. Questions addressed in the FAQ are often either answered with “Read the FAQ” (no direct answer), or simply a cut and paste from the FAQ and the admonition to “Read the FAQ.”

This manual includes a General FAQ and one for using F9 with Excel, Lotus 1-2-3, and Quattro Pro.

Frequently Asked Questions	See <i>FAQ</i> above.
GL Function	Lets you hot-link any cell in your spreadsheet to any balance in your general ledger.
Label	This is a spreadsheet value which is evaluated as text rather than a number. The text may <i>be</i> numeric, but is treated as text. <i>e.g.</i> , an account number must be text
Leaf Company	A leaf company is any company which has a parent(s). It is permissible for a company to have multiple parents in a Tree. Leaf companies without subsidiaries must exist.
Lists Window	This F9 window produces lists of accounts, companies, budgets, <i>etc.</i> and sends them to the Clipboard.
Logical database name	Commonly called an “alias,” this is a simplified name for a company database within the F9 security system. F9 looks up a logical database name to find a path to the actual company database.
Macros	A series of F9 and/or spreadsheet commands grouped together as a single command to make repetitive reporting tasks easier.
Main Account Segment	See the <i>Natural Account Code</i> .
Manual Disk	This diskette contains the full manual(s), about 1 megabyte, and is installed in the F9 folder. The F9 Manual(s) are also available on the CD. See also Acrobat Reader.

Natural Account Code	That segment of the “full” account code that is the financial accounting entity (<i>e.g.</i> , 1000 for Cash).
Operator	A “symbol” for a mathematic function, <i>e.g.</i> , “+” for ADD or “*” for MULTIPLY.
Parameter	A parameter is a value sent to the GL. Each GL function takes one or more parameters (<i>e.g.</i> , accounts, periods, companies).
Password	A password is a secret word or character sequence that is needed to gain access to a computer or to files and programs within your system. Passwords ensure that only authorized users can reach particular information. The best passwords are: memorable (<i>i.e.</i> , something you know well) and unpredictable (<i>i.e.</i> , other people wouldn't know it). Passwords are improved if they use a mixture of languages (<i>i.e.</i> , “ <i>beuno frits</i> ” is “good fries” and a decent password) or numbers and letters (<i>i.e.</i> , not2common).
Paste	To insert from the Clipboard the last text/value/graphic that was Cut or Copied. The pasted information appears in the active cell.
PDF	Acronym for “Portable Document Format” and the extension used by files prepared for the Adobe Acrobat Reader.
Period Wizard	When this is invoked, a dialog box will appear where you create a desired period specifier and send it to the Clipboard. This is a good way, initially, to learn the various period specifiers supported by F9.
Pre-calculate	To build a database for a mapped or tree company from its constituent parts (subsidiaries) . Increases the speed of calculations considerably.
Range	A selection of cells in a spreadsheet that are manipulated as a group – the selected cells must be adjacent to one another. A group consists of a row, a column or a block made up of both.

Rate of Conversion	A factor used to compute the value of one currency in terms of another. The same factor is used for conversions in both directions so the value is never depleted by conversion.
Rate of Exchange	A factor used when exchanging one currency for another. This rate generally favors one of the participants so they profit from each exchange.
Reader Disk	This diskette contains the Adobe Acrobat Reader and has its own install procedure. The Reader is used to read the on-line F9 manual. The reader is also distributed on the CD. See also Acrobat Reader.
Relative Cell Reference	A cell identification based upon how it is located relative to a predetermined starting point on a spreadsheet. In a formula, a relative cell reference identifies the address of a different cell in relation to the cell containing the formula. For example, a relative cell reference might look like A1 or D18. Relative cell references change when the formula is copied. See <i>Cell Reference</i> , <i>Absolute Cell Reference</i> .
Report Wizard	Creates reports for you based on a series of choices made by you.
Roll-up	To remove account segment information from a chart of accounts. For example, a chart of accounts with seventy natural accounts for each of the fifty states has 35,000 accounts. By rolling-up the account segment which indicates the <i>State</i> , only the seventy GL accounts would remain and each would be the sum all of the states.
Server	The F9 server provides accounting information to any DDE client.
Server Disk	Installation diskette for all the shared portions of F9. See also DLL Disk.

Setup Window	This dialog box is used to specify the location of your accounting data, control how your general ledger data is returned to a worksheet and/or used to change function defaults.
Smart Templates	<p>Smart Templates are a collection of special workbooks which combine Visual Basic or Macros to allow you to create a broad range of standard financial reports without requiring any knowledge of your chart of accounts. See <i>Chapter 7: Smart Templates</i> or <i>Appendix G: F9 Templates</i>.</p> <p>A way of invoking F9 functionality from within a macro. Implies the Use of the NoShow=1 parameter to prevent F9 from displaying dialogs.</p>
Text Substitution Operator	See <i>TSO</i> .
Templates	A blueprint for the data, graphics and formatting in a worksheet. See also Smart Templates.
Topic Command	When selected, the main menu for the appropriate DLL will appear (if there is more than one DLL, a list is displayed).
TSO	The <i>Text Substitution Operator</i> is the caret (^) and is used to build account specifier masks. A TSO is an optional parameter for the BSPEC function.
Wildcard	Both the asterisk (*) and the question mark (?) are wild card characters that can be used in an account segment specifier. An asterisk means that any combination of characters (including no characters at all) can occupy that position in the specifier. A question mark means that any single character can occupy that position in the specifier. Very useful when you want to apply one command to several account segments or where you do not remember the full account codes.
Y2K	Acronym for the problems of software expected to occur as a result of the transition between the years 1999 and 2000. See below for information about compliance.

Year 2000 Compliant	These refers to an applications ability to correctly manage the transition between the years 1999 and 2000. <i>F9 is as compliant as the source accounting system.</i> If the accounting system uses four digit years internally, it is Y2K compliant and so is F9 for that system. More details are discussed on this topic in the FAQ.
Zero Suppress	This F9 command hides rows that contain only zero balances and un-hides them just as easily.



Appendix

I. Sample Reports

The following appendix contains a small sampling of the reports that F9 can construct. Because F9 works inside your favorite spreadsheet, the actual number of reports you can produce is limited only by your imagination and the amount of information contained inside your general ledger.

There simply is no financial report you cannot create using F9 that is based on the information contained in your general ledger.

Not only does the combination of F9 and your favorite spreadsheet let you create financial reports, but you can create a complete EIS and/or a production reporting system if you desire. Many of the sample spreadsheets (in F9, a report and a spreadsheet are the same thing) give hints of the automation and customization possible in Microsoft Excel, Lotus 1-2-3, and Quattro Pro.

Most of the reports in this book were created using Microsoft Excel. All of the automation examples (e.g.: buttons, drop-down lists, and dialog boxes) are taken from Excel. Most of this functionality is also available in the other spreadsheets as well.

Statement of Changes in Financial Position

The Statement of Changes in Financial Position is a common report produced with many financial statements (Balance Sheet and Income Statement). This statement “removes” all aspects of accrual accounting and gives an accurate picture of a company’s cashflow during the year. If there is a problem with cashflow it can be used to pinpoint specific problem areas.

Because the Statement of Changes is essentially calculated from operating income and the changes between Balance Sheet items from period to period, F9 can easily be used to create this report.

Statement of Changes in Financial Position
Vancouver Life Insurance Co.

August-95

Operating Activities

<i>Operating Income (Loss)</i>	28,916.79
Depreciation and amortization	1,388.85
Gain (Loss) on sale of Capital Assets	0.00
(Increase)Decrease in Investment	0.00
(Increase)Decrease in A/R	142.00
(Increase)Decrease in Prepaids	0.00
Increase(Decrease) in Bonus Payable	(10,000.00)
Increase(Decrease) in A/P	(1,240.57)
Increase(Decrease) in Dividends	0.00
Changes in non-cash working capital balances	<u>(11,098.57)</u>
<i>Changes in Cash from Operations</i>	<u>19,207.07</u>
<i>Financing Activities</i>	
Repayments of Long-Term Debt	(1,218.74)
Repayments on Shareholder Loans	(4,181.08)
Redemption of Common Shares	0.00
<i>Changes in Cash from Financing Activities</i>	<u>(5,399.82)</u>
<i>Investing Activities</i>	
Purchases of Capital Assets	0.00
Proceeds on disposal of Capital Assets	0.00
<i>Changes in Cash from Investing Activities</i>	<u>0.00</u>
<i>Increase (Decrease) in Cash</i>	13,807.25
<i>Cash, beginning of period</i>	26,402.02
<i>Cash, end of period</i>	<u>40,209.27</u>

Budget Variance Analysis

In this report, we have a standard income statement that compares this year's actuals for a specified month and Year to Date, to the budget and last year's figures. The columns shaded show variances.

This type of report also allows you to pick which specific period you wish to report on. This is done through a series of macros that are provided with F9's "Smart Templates". These macros are very simple. All they do is show a list of choices, allow you to pick a value, and they shove that value into a cell in the spreadsheet's control area. The list of values the macro displays can be dynamic (i.e.: generated by F9 on the fly) or static.

This report also contains different views of the same spreadsheet. The views created are an Excel feature that allows you to see a spreadsheet in different ways.

Remember, using views, one spreadsheet can be many reports.

Budget Variance Analysis

Bob's Sample Company
Actual vs Budget
Comparative
Profit and Loss Statement

Pick a Department

Pick a Company

Print Report

	Current Month Actual	Current Month Budget	Variance	Year to Date Actual	Year to Date Budget	Variance
Income						
Sales						
Sales	89,820.62	67,000.00	32%	1,181,131.77	814,800.00	45%
Sales Returns and Allowances	(1,536.38)	(2,000.00)	-23%	(33,437.04)	(24,000.00)	39%
Sales Discounts	(988.63)	1,000.00	0%	(26,447.19)	(12,000.00)	120%
Net Sales	92,355.53	70,900.00	30%	1,241,016.90	850,800.00	46%
Cost of Goods						
Cost of Goods Sold	5,527.42	5,100.00	8%	71,856.42	61,200.00	17%
Cost Variance	359.28	200.00	80%	4,670.67	2,400.00	95%
Total Costs Of Goods Sold	5,886.70	5,300.00	11%	76,527.08	63,600.00	20%
Gross Profit	86,468.83	65,600.00	32%	1,164,489.82	787,200.00	48%
Expenses						
Operating Expenses						
Accounting and legal fees	133.25	150.00	-11%	2,819.71	1,840.00	53%
Amortization	208.20	250.00	-17%	3,644.18	2,875.00	27%
Amortization of leasehold	124.92	150.00	-17%	2,798.11	1,725.00	58%
Amortization of org. costs	208.20	150.00	39%	3,644.18	2,875.00	27%
Automotive	333.12	380.00	-12%	5,018.28	4,800.00	9%
Bad debts	166.56	200.00	-17%	3,186.14	2,300.00	39%
Commissions	333.12	380.00	-12%	5,018.28	4,800.00	9%
Depreciation	166.56	190.00	-12%	3,186.14	2,300.00	39%
Dues and subscriptions	99.94	115.00	-13%	2,453.29	1,380.00	78%
Moving	208.20	250.00	-17%	3,644.18	2,875.00	27%
Office supplies	274.82	315.00	-13%	4,377.03	3,775.00	15%
Postage	291.48	340.00	-14%	4,560.25	4,025.00	13%
Promotion and entertainment	83.28	95.00	-12%	2,270.07	1,150.00	97%
Rent, office	2,914.77	3,500.00	-17%	33,416.48	40,250.00	-17%
Repairs and maintenance	333.12	385.00	-13%	5,018.28	4,600.00	9%
Shipping supplies	208.20	250.00	-17%	3,644.18	2,875.00	27%
Shop supplies	374.76	450.00	-17%	5,476.32	4,600.00	9%
Travel	166.56	190.00	-12%	3,186.14	2,300.00	39%
Telephone, tele. fax	374.76	450.00	-17%	5,476.32	4,600.00	9%
Utilities	224.65	250.00	-10%	3,670.52	3,050.00	29%
Insurance	166.56	250.00	-33%	3,186.14	2,300.00	39%
Total Operating Expenses	7,395.19	8,690.00	-15%	109,781.09	102,120.00	8%
Payroll Expenses						
Employee benefit plan	666.23	750.00	-11%	8,682.57	9,200.00	-6%
Employee benefits, direct	416.40	470.00	-11%	5,934.35	5,750.00	3%
Wages & benefits, direct	17,488.63	20,000.00	-13%	193,728.88	241,500.00	-20%
Wages & benefits, indirect	12,491.88	14,000.00	-11%	138,764.63	172,500.00	-20%
Wages casual, direct	208.20	250.00	-17%	3,644.18	2,875.00	27%
Subcontract costs	2,914.77	3,500.00	-17%	33,416.48	40,250.00	-17%
Commissions	416.40	500.00	-17%	5,934.35	5,750.00	3%
Total Payroll Expenses	34,602.49	39,470.00	-12%	390,105.43	477,825.00	-18%
Total Expenses	41,997.68	48,160.00	-13%	499,886.52	579,945.00	-14%
Other Income						
Delivery revenue	541.31	650.00	-17%	7,388.46	7,475.00	-2%
Interest income	666.23	750.00	-11%	8,682.57	9,200.00	-6%
Miscellaneous income	74.90	90.00	-17%	2,178.46	1,035.00	110%
Total Other Income	1,282.50	1,490.00	-14%	18,169.49	17,710.00	3%
Net Income Before Taxes	45,753.65	18,930.00	142%	682,772.79	224,965.00	204%
Corporate income taxes	83.28	100.00	-17%	2,270.07	1,150.00	97%
Corp. income taxes - deferred	41.64	50.00	-17%	1,812.04	575.00	215%
Net Income	45,628.73	18,780.00	143%	678,690.68	223,240.00	204%

Proforma Profit and Loss Statement

In this statement, we have 12 columns of data. Note that the headings in the columns go from specifying actuals for a period to specifying budgets. This is done by using a simple formula in the spreadsheet that will return the correct heading based on the current period in the GL system. For example, if the current period in the GL is month 5, then this report will change the column headings to Month 1 through to Month 5 and Budget 6 through to Budget 12 or if the current period is 7, then it will change the column headings to Month 1 through Month 7 and Budget 8 through to Budget 12. So as the current period changes inside your GL, this report will change the headings to reflect actual or budget information accordingly.

In this report we have also created different views of the same spreadsheet.

We have also included macros to print the report and view the bar graph attached to the report. The buttons in the report run these macros. These macros are also provided in several of our “Smart Templates”.

Proforma Profit and Loss Statement

Bob's Sample Company
Proforma Report
for Period Ending May 31/96

	Month 1	Month 2	Month 3	Month 4	Month 5	Budget 6	Budget 7	Budget 8	Budget 9	Budget 10	Budget 11	Budget 12
Income												
Sales												
Sales	534,119.22	469,718.33	441,343.49	426,867.25	402,695.30	450,000.00	450,000.00	450,000.00	450,000.00	450,000.00	450,000.00	450,000.00
Sales Returns and Allowances	5,498.35	4,978.53	4,985.17	4,232.85	4,232.85	4,500.00	4,500.00	4,500.00	4,500.00	4,500.00	4,500.00	4,500.00
Sales Discounts	2,700.66	2,558.69	2,302.56	2,052.75	1,791.75	2,500.00	2,500.00	2,500.00	2,500.00	2,500.00	2,500.00	2,500.00
Net Sales	542,226.23	477,232.56	450,611.22	433,559.07	408,720.00	457,000.00						
Cost of Goods												
Cost of Goods Sold	33,726.47	33,664.84	32,118.81	28,927.03	26,622.58	30,000.00	30,000.00	30,000.00	30,000.00	30,000.00	30,000.00	30,000.00
Cost Variance	3,345.69	3,366.05	3,233.35	3,189.56	3,412.11	3,000.00	3,000.00	3,000.00	3,000.00	3,000.00	3,000.00	3,000.00
Total Costs Of Goods Sold	37,072.16	37,030.88	35,352.26	32,116.59	29,234.69	33,000.00						
Gross Profit	505,154.07	440,201.67	415,258.96	401,442.48	379,485.31	424,000.00						
Expenses												
Operating Expenses												
Advertising and sales	147.69	139.92	162.73	104.89	89.10	200.00	200.00	200.00	200.00	200.00	200.00	200.00
Advertising	175.63	139.09	176.63	129.15	89.67	300.00	300.00	300.00	300.00	300.00	300.00	300.00
Amortization of leasehold	234.82	234.82	212.41	176.83	134.45	150.00	150.00	150.00	150.00	150.00	150.00	150.00
Amortization of org. costs	225.96	225.96	181.54	147.99	113.99	150.00	150.00	150.00	150.00	150.00	150.00	150.00
Bad debts	122.58	101.48	77.00	55.50	41.82	300.00	300.00	300.00	300.00	300.00	300.00	300.00
Commissions	226.33	162.15	141.46	127.33	110.46	250.00	250.00	250.00	250.00	250.00	250.00	250.00
Depreciation	463.33	463.33	463.33	463.33	463.33	500.00	500.00	500.00	500.00	500.00	500.00	500.00
Director subscriptions	144.10	116.77	83.76	61.84	48.86	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Printing	362.50	362.50	362.50	362.50	362.50	150.00	150.00	150.00	150.00	150.00	150.00	150.00
Office supplies	163.74	161.83	191.73	157.54	157.54	150.00	150.00	150.00	150.00	150.00	150.00	150.00
Postage	233.65	233.65	233.65	233.65	233.65	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Promotion and entertainment	2,100.00	1,652.26	1,172.16	896.56	613.84	2,000.00	2,000.00	2,000.00	2,000.00	2,000.00	2,000.00	2,000.00
Rent, office	1,211.32	1,089.32	722.81	606.01	473.38	250.00	250.00	250.00	250.00	250.00	250.00	250.00
Repairs and maintenance	326.59	301.35	221.74	217.75	205.12	400.00	400.00	400.00	400.00	400.00	400.00	400.00
Shipping	150.00	131.42	82.83	72.58	65.44	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Travel	188.70	150.57	118.15	87.54	67.66	200.00	200.00	200.00	200.00	200.00	200.00	200.00
Utilities, phone, tele. fax	112.50	75.25	53.65	38.86	30.96	150.00	150.00	150.00	150.00	150.00	150.00	150.00
Insurance												
Total Operating Expenses	7,141.60	5,825.90	4,733.82	3,909.95	3,465.20	5,575.00						
Payroll Expenses												
Employer's share	1,319.46	868.84	843.93	719.26	569.51	1,000.00	1,000.00	1,000.00	1,000.00	1,000.00	1,000.00	1,000.00
Employee's share	252.19	2,119.06	1,727.27	1,202.54	921.64	1,500.00	1,500.00	1,500.00	1,500.00	1,500.00	1,500.00	1,500.00
Wages & benefits, direct	10,765.40	8,319.15	6,337.09	4,759.07	3,200.78	9,000.00	9,000.00	9,000.00	9,000.00	9,000.00	9,000.00	9,000.00
Wages & benefits, indirect	6,176.84	4,785.78	3,531.05	2,818.76	2,111.48	5,000.00	5,000.00	5,000.00	5,000.00	5,000.00	5,000.00	5,000.00
Salaries & benefits, indirect	3,751.36	2,845.67	2,027.16	1,473.38	1,069.33	3,000.00	3,000.00	3,000.00	3,000.00	3,000.00	3,000.00	3,000.00
Subcontractors	1,223.8	918.91	724.59	637.89	500.02	900.00	900.00	900.00	900.00	900.00	900.00	900.00
Commissions												
Total Payroll Expenses	26,334.56	20,285.50	15,495.94	11,851.40	8,551.40	21,000.00						
Total Expenses	33,406.76	26,111.40	20,229.77	15,761.34	12,016.61	26,575.00						
Other Income												
Delivery revenue	3608.99	2,995.94	2,035.22	1,702.54	1,352.39	2,500.00	2,500.00	2,500.00	2,500.00	2,500.00	2,500.00	2,500.00
Interest income	4427.78	3,055.09	2,584.23	1,721.79	1,448.56	3,500.00	3,500.00	3,500.00	3,500.00	3,500.00	3,500.00	3,500.00
Miscellaneous income	7546.08	5,619.98	4,446.61	2,818.81	2,122.97	3,000.00	3,000.00	3,000.00	3,000.00	3,000.00	3,000.00	3,000.00
Total Other Income	15,482.85	11,270.98	9,006.05	6,343.14	4,924.93	9,000.00						
Net Income Before Taxes	487,158.76	425,361.26	404,035.24	392,024.28	372,383.63	406,425.00						
Income taxes - federal	112.16	142.16	170.41	122.16	92.16	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Income taxes - state	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Income taxes - deferred	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Net Income	487,046.60	425,219.10	403,789.32	391,804.30	372,226.71	406,425.00						

Multi Currency Reporting

Creating reports in multiple currencies is trivial with F9. Consolidating companies in different currencies can be as simple as having F9 generate balances in the foreign currency and multiplying them by the appropriate currency conversion factor in the spreadsheet.

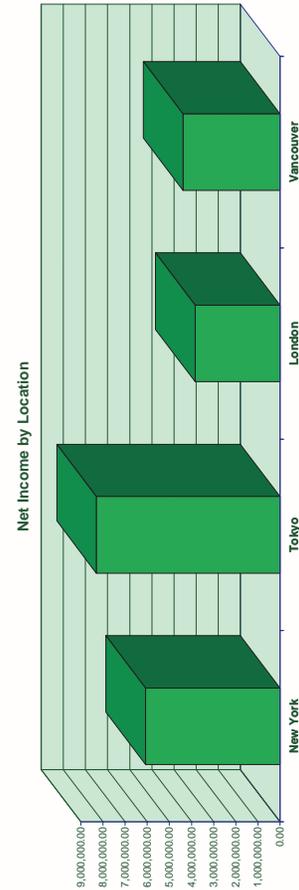
If you are using a true multi-currency accounting system where transactions can be processed in different currencies, then F9 can create reports in both home and foreign currencies. F9 can drill down to transactions in both home and foreign currencies.

As always, if the data is in your GL, F9 can get to it.

WorldWide MegaProducts
Multi Currency Income Statement
For 1st Quarter Ending July 31, 1996

Conversion Rates	
\$ US	1.363
£ UK	2.30
¥ Yen	0.0105

	New York			Tokyo			London			Vancouver			Total (HOME)
	\$ US	HOME (\$ CDN)	¥ Yen	HOME (\$ CDN)	£ UK	HOME (\$ CDN)	\$ CDN	HOME (\$ CDN)	¥ Yen	HOME (\$ CDN)	\$ CDN		
Income													
Sales	4,892,560.33	6,668,559.73	866,006,421.73	9,102,517.43	1,851,634.99	4,295,793.19	4,801,812.90	24,868,683.24					
Sales Returns and Allowances	33,598.25	46,794.41	5,971,771.95	62,703.04	18,576.04	43,096.40	35,477.18	187,071.03					
Sales Discounts	12,456.36	16,978.02	2,211,110.64	23,216.86	5,965.71	12,912.45	12,790.27	66,897.40					
Net Sales	4,938,614.94	6,731,332.16	875,089,250.32	9,188,437.13	1,875,776.74	4,351,802.04	4,850,080.34	25,121,651.68					
Cost of Goods	3,389.04	4,632.89	602,263.01	6,323.76	1,703.62	3,892.16	3,285.91	18,194.72					
Cost of Goods Sold	204.68	278.98	51,242.06	338.04	128.08	280.54	241.75	1,344.31					
Total Costs Of Goods Sold	3,603.72	4,911.87	653,505.07	6,861.80	1,826.60	4,237.71	3,527.65	19,539.03					
Gross Profit	4,935,011.22	6,726,420.29	874,435,745.25	9,181,575.33	1,873,950.14	4,347,564.33	4,846,552.69	25,102,112.64					
Expenses													
Operating Expenses													
Accounting and legal fees	1,654.22	2,254.70	308,111.53	3,235.17	637.22	1,478.36	1,528.66	8,486.89					
Advertising	2,583.96	3,521.39	463,206.20	4,811.17	1,576.14	3,666.66	2,373.23	14,362.44					
Bad debts	1,890.88	2,583.96	342,875.00	3,045.84	1,000.00	2,310.00	2,310.00	11,810.00					
Commission of leasehold	3,899.98	5,451.97	716,977.73	7,538.27	2,728.27	5,285.60	3,723.03	21,994.87					
Depreciation	2,153.25	2,934.88	381,793.61	4,008.63	719.23	1,689.61	1,841.04	10,553.37					
Office supplies	3,300.33	4,498.35	584,812.89	6,140.54	1,423.69	3,302.97	3,315.14	16,937.64					
Rent, office	35,458.77	48,330.30	6,283,336.41	65,975.03	19,442.27	45,106.06	45,106.06	192,562.53					
Repairs and maintenance	3,965.23	5,404.61	704,953.38	7,402.01	1,651.09	3,930.53	3,663.06	20,300.21					
Shipping	2,098.70	2,845.19	370,048.04	3,985.44	1,097.32	2,445.78	2,160.80	11,457.29					
Shipping supplies	2,785.66	3,786.85	500,963.45	5,255.92	1,511.10	3,605.75	2,682.29	15,240.80					
Utilities	1,966.52	2,680.37	350,557.53	3,680.85	921.80	2,138.59	1,912.97	10,412.77					
Total Operating Expenses	67,485.74	91,983.06	12,011,105.38	126,116.61	34,807.02	80,752.29	63,516.23	362,368.19					
Payroll Expenses													
Employee benefit plan	8,758.33	11,989.40	1,558,371.39	16,373.40	5,005.31	11,612.32	9,282.41	48,257.53					
Employee benefits, direct	5,002.70	6,818.68	886,006.91	9,312.52	2,696.13	5,853.43	4,720.25	26,804.88					
Wages & benefits, direct	210,000.54	286,230.74	37,249,194.48	391,116.54	96,213.99	227,665.32	206,905.81	1,110,008.01					
Wages & benefits, indirect	35,089.83	47,827.17	6,238,566.89	65,515.44	14,434.92	33,489.01	34,973.52	181,811.14					
Commissions	5,745.50	7,831.12	1,017,805.05	10,686.95	2,430.10	5,637.84	6,046.26	30,202.17					
Total Payroll Expenses	421,434.70	574,415.50	74,767,421.57	784,952.93	193,513.15	448,950.51	411,272.39	2,219,591.32					
Total Expenses	488,920.44	666,398.56	86,768,526.96	911,069.53	228,320.17	529,702.80	474,788.62	2,581,959.51					
Other Income													
Delivery revenue	6,568.87	8,953.37	1,172,297.56	12,388.61	2,914.15	6,700.83	6,528.74	34,551.76					
Interest	801.36	1,228.55	159,769.09	1,677.66	300.55	908.07	909.83	4,722.02					
Miscellaneous income													
Total Other Income	16,344.78	22,277.94	2,915,126.44	30,608.83	7,270.30	16,867.10	16,286.70	86,040.56					
Net Income Before Taxes	4,462,435.56	6,082,299.67	790,582,344.73	8,301,114.62	1,652,900.27	3,834,728.63	4,388,050.77	22,606,193.69					
Corporate income taxes	1,000.00	1,363.00	186,337.97	1,966.65	428.07	983.12	949.21	5,261.88					
Corp. income taxes - deferred	500.00	681.50	96,483.78	1,013.08	285.42	682.16	505.57	2,862.31					
Net Income	\$4,460,935.56	\$6,080,255.17	¥790,299,522.98	\$8,298,144.99	£1,652,186.79	\$3,833,073.34	\$4,386,595.99	\$22,598,069.49					



Weekly Income Statement

In this report, we have a standard income statement that compares This Year vs. Last Year Net Income *on a weekly basis*. This illustrates the F9 GLTRAN function which allows you to report on any “slice” of time you wish.

By simply specifying the starting date and the ending date of a slice of time, the GLTRAN function will return the net total for all the transactions that occurred during that specified time.

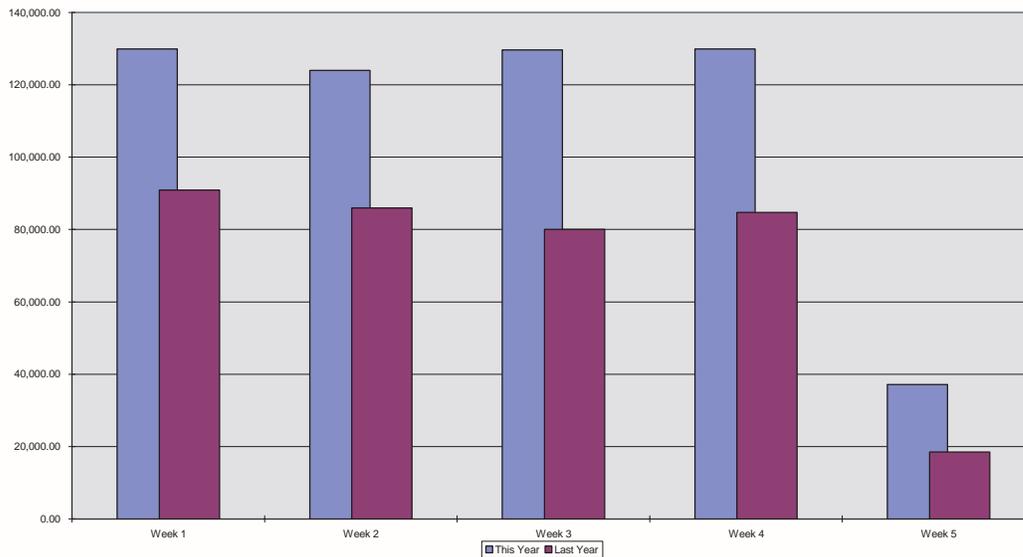
Bob's Snowcap Ski Resort, December

This Year

	From:	1-Dec	8-Dec	15-Dec	22-Dec	29-Dec	Dec - 96
	To:	7-Dec	14-Dec	21-Dec	28-Dec	31-Dec	Total
		Week 1	Week 2	Week 3	Week 4	Week 5	
Income							
Net Sales		160,687.79	146,913.77	148,445.51	145,937.97	45,910.80	647,895.83
Gross Profit		155,765.05	142,148.02	143,945.82	141,230.30	44,504.30	627,593.48
Expenses							
Total Other Income		823.86	736.83	576.69	484.07	235.39	2,856.84
Net Income Before Taxes		130,042.69	124,204.97	129,817.04	130,064.85	37,155.05	551,284.60
Net Income This Year		129,892.69	124,023.54	129,662.06	129,931.55	37,112.20	550,622.02

Last Year

	From:	1-Dec	8-Dec	15-Dec	22-Dec	29-Dec	Dec - 95
	To:	7-Dec	14-Dec	21-Dec	28-Dec	31-Dec	Total
		Week 1	Week 2	Week 3	Week 4	Week 5	
Income							
Net Sales		112,481.45	103,163.29	94,703.00	96,611.58	22,906.04	462,002.92
Gross Profit		109,035.53	100,208.70	91,995.62	94,167.37	22,204.31	448,764.54
Expenses							
Total Other Income		576.70	434.13	473.42	420.69	117.44	2,187.15
Net Income Before Taxes		91,029.88	86,140.46	80,134.66	84,867.67	18,537.58	386,718.79
Net Income Last Year		90,924.88	85,967.31	79,995.07	84,744.84	18,516.20	386,126.83



Running Year P&L

In this report, we have Running Year P&L.

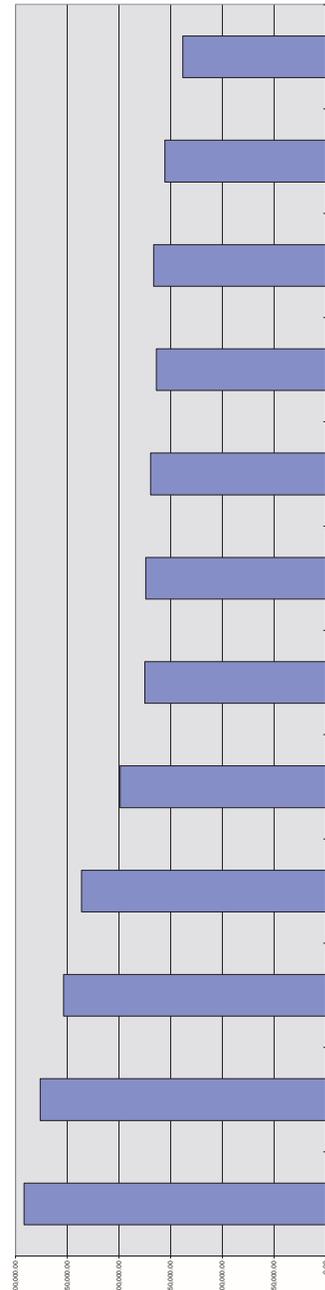
This type of report also allows you to pick which period you wish to define as your current period and report on all prior months accordingly. This is done through a series of macros that are provided with F9's "Smart Templates".

Macros can also be written to show charts and graphs that are dynamically linked to the report, so as the report changes, so do the graphs.

The Teeract Group

Running Profit and Loss Statement for Period Ending May 31/99

	This Month May-99	1 Month Ago Apr-99	2 Months Ago Mar-99	3 Months Ago Feb-99	4 Months Ago Jan-99	5 Months Ago Dec-98	6 Months Ago Nov-98	7 Months Ago Oct-98	8 Months Ago Sep-98	9 Months Ago Aug-98	10 Months Ago Jul-98	11 Months Ago Jun-98
Income												
Sales	339,119.10	315,343.30	287,037.97	268,042.87	225,970.84	201,337.33	198,941.11	188,612.04	181,332.14	182,040.00	179,232.57	159,824.46
Sales Returns and Allowances	0.00	0.00	0.00	0.00	0.00	-538.00	0.00	-496.00	0.00	463.00	-677.00	-683.00
Sales Discounts	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Net Sales	339,119.10	315,343.30	287,037.97	268,042.87	225,970.84	200,800.33	199,400.11	188,116.04	181,332.14	181,427.00	169,660.57	159,151.66
Cost of Goods Sold												
Cost of Goods Sold	21,727.41	19,543.45	18,026.50	19,016.62	17,536.41	17,907.75	15,666.70	13,195.44	12,881.76	11,277.86	10,589.71	9,178.74
Cost Variance	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Total Costs of Goods Sold	21,727.41	19,543.45	18,026.50	19,016.62	17,536.41	17,907.75	15,666.70	13,195.44	12,881.76	11,277.86	10,589.71	9,178.74
Gross Profit	317,391.69	295,800.45	269,008.47	249,026.25	208,434.43	182,892.58	183,733.41	174,920.61	168,450.38	170,149.13	159,070.86	149,972.92
Expenses												
Operating Expenses												
Accounting and legal fees	120.00	141.38	127.66	110.03	126.89	107.56	149.70	166.42	139.97	83.49	72.89	54.87
Advertising	1,000.00	1,000.00	1,000.00	1,000.00	1,000.00	1,000.00	1,000.00	1,000.00	1,000.00	1,000.00	1,000.00	1,000.00
Amortization of leasehold	84.32	86.77	88.11	90.41	92.71	95.01	97.31	99.61	101.91	104.21	106.51	108.81
Amortization of org. costs	177.29	203.07	190.01	196.38	190.17	149.34	109.73	89.08	68.47	53.16	41.25	30.92
Bad debts	122.58	93.53	74.29	51.45	37.89	88.99	69.55	47.68	35.60	23.47	18.43	12.71
Commissions	226.33	189.80	197.09	128.09	95.41	85.60	79.02	114.11	74.91	33.19	110.00	83.40
Debt and subscriptions	168.33	141.58	104.82	104.82	104.82	104.82	104.82	104.82	104.82	104.82	104.82	104.82
Moving	144.10	116.75	143.20	99.19	77.88	60.88	56.26	72.55	47.76	71.01	71.01	87.91
Office supplies	150.00	150.00	150.00	150.00	150.00	150.00	150.00	150.00	150.00	150.00	150.00	150.00
Promotion and entertainment	52.50	41.69	59.78	98.06	81.79	81.34	44.12	33.09	56.82	54.93	54.93	41.57
Rent office	1,500.00	1,500.00	1,500.00	1,500.00	1,500.00	1,500.00	1,500.00	1,500.00	1,500.00	1,500.00	1,500.00	1,500.00
Repairs and maintenance	341.30	191.81	159.70	124.86	100.92	49.24	38.18	38.18	26.74	80.93	116.84	76.66
Shipping supplies	177.30	138.33	106.84	72.92	65.21	110.92	80.07	59.72	77.71	65.55	52.27	41.04
Telephone, tele. fax	333.65	270.26	210.49	188.64	148.60	136.35	126.96	74.16	62.32	62.32	65.72	49.34
Travel	1,100.00	1,100.00	1,100.00	1,100.00	1,100.00	1,100.00	1,100.00	1,100.00	1,100.00	1,100.00	1,100.00	1,100.00
Utilities	132.00	102.47	108.44	86.07	72.46	80.00	48.80	36.26	38.07	108.68	72.72	55.93
Total Operating Expenses	5,590.66	4,617.37	4,014.11	3,360.39	2,925.80	2,562.58	2,184.04	2,104.71	1,934.94	1,707.91	1,741.27	1,418.32
Payroll Expenses												
Employee benefits, direct	1,935.06	2,045.85	2,068.00	1,831.03	1,765.89	1,658.99	1,585.69	1,578.93	1,514.95	1,477.85	1,411.95	1,311.11
Wages & benefits, direct	252.97	181.75	138.17	95.37	135.40	129.41	111.41	87.39	62.47	46.89	63.19	121.82
Wages & benefits, indirect	10,765.40	7,986.08	6,101.37	4,950.37	3,240.77	2,506.53	2,101.01	1,618.84	1,164.54	873.48	680.41	486.89
Wages indirect, direct	715.83	519.18	418.92	318.62	218.62	158.62	118.62	88.62	58.62	28.62	18.62	8.62
Subcontract costs	1,751.36	1,246.86	808.46	660.06	476.85	384.84	285.24	230.62	155.16	100.00	50.00	25.00
Commissions	2,253.00	2,045.85	1,831.03	1,561.44	1,248.88	1,036.32	824.72	613.16	401.60	190.10	78.60	37.10
Total Payroll Expenses	20,665.56	15,511.72	11,997.17	9,674.37	6,991.65	5,313.26	4,516.73	3,690.93	2,823.11	2,286.35	1,826.15	1,336.04
Total Expenses	26,246.22	20,129.09	15,921.28	13,244.77	9,887.45	7,875.84	6,702.77	5,711.64	4,758.05	3,994.46	3,567.42	2,954.36
Other Income												
Interest income	465.21	326.52	278.51	211.73	149.25	108.89	139.27	187.22	101.45	171.00	151.52	84.84
Miscellaneous income	46.08	32.38	29.78	23.78	184.66	127.96	74.74	60.05	54.31	98.38	111.55	86.52
Total Other Income	824.08	679.81	551.18	460.47	347.67	238.15	264.98	295.01	285.04	378.22	370.67	173.46
Net Income Before Taxes	291,969.55	276,351.17	253,638.36	236,251.05	198,894.65	175,153.79	174,926.52	169,593.97	163,957.97	166,532.89	155,874.11	138,375.48
Corporate income taxes	0.00	111.87	111.87	62.02	107.53	182.18	178.66	137.38	76.87	62.33	68.23	113.39
Corp. income taxes - deferred	0.00	99.45	46.19	25.22	50.32	60.56	78.51	56.32	49.44	37.10	98.04	75.85
Net Income	291,819.55	276,179.85	253,509.51	236,154.72	198,736.79	174,941.05	174,942.45	169,310.26	163,801.84	166,418.92	155,710.84	138,181.24



Index

Symbols

! operator	222
tree file	154
	<i>See also</i> Support Files: Tree
\$ operator	
absolute cell reference	30
& operator	
map file	154, 219
	<i>See also</i> Support Files: Tree
1-2-3	
	<i>See</i> Lotus 1-2-3

A

Absolute Cell Reference	219
Account	
Analyze Access Command	127
Artificial segmentation	84
Description	58
Multi-segment descriptions	75
Origin	60
Origin using WGL	64
referencing cells	77, 168
Segments	224
	<i>See also</i> Cell references: Using
Segments, main	223, 224
Segments, natural	224
Specifying	67, 69
BSPEC function, and the	77
BSPEC with WGL functions	64
Lists	24, 71
Multiple accounts	71
Permutations of lists	78
Ranges	25, 72
Single accounts	70
Wildcards	24, 72
Account Enquiry	184
	<i>See also</i> Enquire
Account Map	
	<i>See</i> Support Files: MAP
Accounting system	
Help	98, 120

Origin / CODATA function	56, 60
ACCTDATA function	60
Acrobat Reader	219, 225
Online Manual	223
Actual	
	<i>See</i> Type specifier
Add-in	
F9EDIT.XLL	147
loading in 1-2-3	17, 133
loading in Excel	16, 133
Menu	182, 195
Versions	184
Always Calculate Functions	
Options Dialog	125, 126, 131
Ampersand operator	154
Analyze command	126, 127, 184, 219
Account Access	127
Recalculation Time	130
Artificially segmenting accounts	84
Automatic Datapreload	134
Automating F9	114, 137
Automating reporting	144
Autosum causes Error	
Excel	185

B

Balance Time	
opening/ending	88
Bang! Operator	
	<i>See</i> ! operator
BSPEC function	55, 64, 77, 78
Case Study	83, 84
Cell References	220
Named ranges	213
Templates, using with	212
Text substitution operator	226
BTI.INI	217
Btrieve database	147, 149, 177, 215
error 20	216, 218
error 4 or 11 or 12	216
error 88	218
Logical name	223

Unknown Error	215	Passwords	183, 195
W32MKSET.EXE	217	Zero Suppress	121, 126, 184, 197
Budget	101, 113	Clients	133
	<i>See also</i> Type specifier	Clipboard	119
Accessing	175	Paste	224
Default	101	CODATA function	56, 60
Specifying	26, 89	Column Names	150
Write-back	220	Columns	
Excel	184	F9DATA function	61, 150
write-back	113	F9Data Function	61
Buttons		Zero Suppress	122
examples of	213	Command, F9	221
C		Company	
Case Study		Budgets	101
BSPEC function	83, 84	Report Wizard Model	47
Cell Range	224	type parameter	101
Cell reference	220	Company Edit	151
Absolute	30, 219	Company Profile	18, 100
Introduction	22	Company specifier	67, 91
Relative	30, 225	lists of	91
Using	30, 77, 168	multiple	91
Cell references		single	91
Using with Auto-copy	44	CONAME Function	175
Chart of accounts		Consolidate	221
Map file consolidation	153	Consolidations	
Chart window	102, 183, 195, 220	Account Map	153, 155
Clipboard, viewing	112	Ownership Tree	152
Interrupting	112	Precalculation	156, 224
Client add-ins	95, 220	Currency Conversion	156
Client Commands		simple	25
Account Enquire	184	tips for performing	165
Analyze	127	Control totals, using	167
Budget writeback	184	Conversion Rate	221, 225
Chart	183, 195	Converting Currency	
Drill down	183, 195	Consolidation	156, 157
Excel		Editing rates	156, 157
To E-mail/Viewer	124, 172, 184	Transactions	156
Toolbar	183	Currency	
F9 Setup	183, 195	Conversion Rate	221, 225
GL Paste	124	Exchange Rate	222, 225
Lotus 1-2-3		Currency conversion	
To Value	124, 172, 196	Editing Rates	156, 157
Options	125, 183	Transactions	156
		Currency specifier	68, 93

D

Database name, Logical	223
DataPreLoad	102, 134, 139, 177, 221
Automating	135
Silent running, and	135
Date Specifiers	67
Dates	67
DDE	145, 221
F9 Client Add-ins use	54
Topic	93, 95, 96, 97
DDE Time Out	125, 200, 207
DDEML errors	98, 173
Default	
Parameter values	
CODATA Function	56, 60
Specifying	18, 63, 68
Period	101
Default Settings	100
DESC function	58
Descriptions	
DESC Function	58
multi-segment	36, 59
SDESC Function	56, 58
WDESC function	65
Distributing F9 reports	31
DLL	221
Help	95, 120, 183
DLLs	134
defined	132
dollar sign	
	<i>See</i> \$ operator: absolute cell reference
Double Quotes	
Financial Entites and Hyphens	68, 77
Drill Down	
interrupting	106
Drill Down window	104, 105, 183, 195, 222
accounts, drilling by	105
Clipboard, sending data to	107
interrupting	106
segments, drilling by	105
to chart	105
transaction fields	110
transaction period	110

transactions, drilling by	108
Dynamic Data Exchange	145, 221
	<i>See also</i> DDE
Dynamic Link Library	
	<i>See</i> DLL

E

E-mail	
From Excel	31, 124
From Lotus 1-2-3	32, 124, 196
Preparing F9 reports	124
Sending F9 spreadsheets	31, 172, 196
Technical Support	13, 134
Editing Support Files	155
	<i>See also</i> Support Files
Enquire	40, 41
Excel	184
Entities, Financial	
	<i>See</i> Financial Entities
Errors	
Excel	
Autosum cases Error	185, 190, 191
Excel	
1-2-3 sheets, reading	185
Autofill and reports	43
Autosum cases Error	185, 190, 191
Budget Writeback	184
buttons, creating	186
Can not read this binary file	190
F9.XLL	16
F9BMPXLW	16, 182, 189
F9BMP32.XLW	16, 182, 189, 190
FAQ	189
loading F9 automatically	181
macros	185
manual recalc mode	181
Paste as Values	32
RC mode	112
To E-Mail/Viewer	31
tool bar	183
tool button faces, customizing	182
zero suppress	227
Exchange Rate	222, 225
Exclamation Operator	154, 222

- See also ! operator*
- ExecuteExcel4Macro 138, 144
- Executive Information Systems 12, 33, 126, 137
- F**
- F9
- Security 223
 - F9 Enterprise 147
 - F9 key, the 22
 - F9 Professional 66, 94, 100, 134, 150
 - F9 Server 182, 195, 225
 - F9 Setup 100
 - F9 Viewer 33
 - Preparing F9 reports 124
 - F9.INI 140, 142
 - F9BMPXLW
 - Excel Toolbar 16, 182, 189
 - F9BMP32.XLW
 - Excel Toolbar 16, 182, 189, 190
 - F9Data
 - Introducing 61
 - F9Data Function 61, 150
 - Column 61, 150
 - FAQ 222
 - Excel 189
 - General Questions 171
 - Lotus 1-2-3 200
 - Quattro Pro 193
 - File Security 223
 - Password 224
 - Financial Entities 34, 35, 36, 37, 60, 73
 - Editing 116, 148
 - Hyphens 68, 77
 - Multi-segment 36, 75
 - Natural 74
 - Restrictions 76
 - User Defined 74
 - See also Support Files: SAD*
 - Financial Roll-up 225
 - Frequently Asked Questions
 - See FAQ*
 - From Date 67
 - Function, F9 222
 - Paste as Values 32
 - Function Parameters 67, 224
 - Functions
 - ACCDATA 60
 - Account balances 55, 57
 - Account data 60
 - Account descriptions 56, 58
 - account descriptions 58
 - Account ranges 72
 - Account Specifiers 67, 69
 - Financial Entities 73
 - Account specifiers
 - Using BSPEC 77
 - additional keywords 90
 - Arguments 68
 - Balance time specifiers 88
 - BSPEC 64, 77
 - Budget specifiers 89
 - CODATA 56, 60
 - Column specifier 61
 - combining account types 72
 - company data function 56, 60
 - Company specifier 67, 91
 - CONAME 175
 - consolidating companies 91
 - Currency Specifier 93
 - Currency specifiers 68
 - Date specifiers 67
 - Default values
 - Setup Window 100
 - DESC 58
 - F9Data 61, 150
 - format conventions 54
 - GL 55, 57, 223
 - GLTRAN 57
 - Multiple accounts 71
 - NGL 55, 57
 - OD 61
 - Other Data 61
 - Parameter lists in Lotus 1-2-3 69
 - Period specifiers 67, 85, 88, 89
 - Examples 90
 - Future 88
 - referencing existing cells 77
 - SDESC 56, 58
 - Multi-segment descriptions 59, 75

- | | | | |
|--|--------------------|--|--|
| Single account numbers | 70 | | |
| Single company numbers | 91 | | |
| Time specifiers | 87 | | |
| Topic | 13, 93, 95, 96, 97 | | |
| Type specifier | 93 | | |
| value type specifiers | 88 | | |
| WDESC | 65 | | |
| WGL | 63 | | |
| wildcard characters | 72 | | |
| Year specifier | 67, 89, 92 | | |
| Functions Dirty | | | |
| Always Calculate Functions | 125, 126, 131 | | |
| Future periods | 88 | | |
| G | | | |
| Generate Descriptions | 116 | | |
| GL function | 24, 55, 57, 223 | | |
| introducing | 21, 24 | | |
| GL Paste | 40, 124 | | |
| GL Wizard | 40 | | |
| GL Paste | 124 | | |
| GLTRAN Function | 57 | | |
| GLTRAN function | 57 | | |
| H | | | |
| Help | | | |
| About | 120 | | |
| Accounting DLL | 120, 183 | | |
| F9 Server | 183 | | |
| Hyphens | 68, 77 | | |
| I | | | |
| Information Systems | | | |
| Executive | 33 | | |
| INI file | 140, 142 | | |
| Options Dialog | 125 | | |
| L | | | |
| Leaf Company | 223 | | |
| Lists, generate descriptions | 116 | | |
| Lists, transpose | 116 | | |
| Lists window | 113, 211, 223 | | |
| clipboard, and the | 115 | | |
| companies, listing | 114 | | |
| is clause | 114 | | |
| list segment | 114 | | |
| main account segment, listing | 113 | | |
| natural account segment, listing | 113 | | |
| silent running | 116, 142 | | |
| where clause | 114 | | |
| Logical database name | 223 | | |
| Lotus 1-2-3 | 193 | | |
| <<F9>> | 200 | | |
| Auto-fill and reports | 44 | | |
| F9.ADW | 193 | | |
| FAQ | 200 | | |
| loading F9 automatically | 194 | | |
| macros | 199 | | |
| Parameter lists | 69 | | |
| Paste as Values | 32 | | |
| recalcs, holding | 198 | | |
| smart icons | 193 | | |
| To Value | 32 | | |
| undo zero suppress | 123 | | |
| zero suppress | 123, 197, 227 | | |
| M | | | |
| Macros | 144, 199, 206, 223 | | |
| <i>See also</i> Silent Running | | | |
| Excel | 185 | | |
| Macros, using | 137 | | |
| Main segment, Accounts | 223 | | |
| Manual | | | |
| Acrobat Online | 12, 223 | | |
| Manual Recalculation | 187 | | |
| Map File | 153, 155 | | |
| <i>See also</i> Support Files: MAP | | | |
| Mask | | | |
| <i>See</i> TSO operator | | | |
| Menu | | | |
| F9 Add-in | 182 | | |
| Microsoft Excel | | | |
| <i>See</i> Excel | | | |
| Model Company | 47 | | |
| Multi-segment Descriptions | 59 | | |
| Multi-segment Financial Entities | | | |
| <i>See</i> Financial Entities: Multi-segment | | | |

Multidimensional databases 45

N

Named Ranges 187

NGL function 55, 57

No-Account-Found
 Replacing 101, 102

NoShow parameter 138

O

OD Function 61

Online Manual 12, 223

Options Dialog

 Always Calculate Functions 125, 126

 DDE Timeout 125

 Do Column-wise Zero Suppress 126

 Rebuild Toolbar 126

Options windows 125, 183

Origin

 Account Data function 60

 Company Data function 56, 60

 WGL function 64

Other Data Function 61

Ownership interest 152

See Support Files: Tree

Ownership Tree 152

P

Parameter

See Functions

Parameter, Function 224

Parameter lists

 Lotus 1-2-3 69

Parent Company 152

Password 224

Password window 118, 183, 224

Paste as Values 32

Performance

 Analyze command 127

 Datapreload 134

Period lists 89

Period override 101

Period specifiers 67, 85, 88

balances, obtaining 86

budgets, obtaining 89

changes in balance, obtaining 86

durations 87

examples 90

future periods 88

future periods, obtaining 88

introducing 25

listing 89

opening balances 88

 with WGL function 64

Period Wizard 85, 184

Period Wizard window 50, 110, 119, 160, 224

 Clipboard, sending data to 119

Precalculate Consolidation 157, 224

See also Consolidations:

 Precalculation

Profile button 18

Q

Quotation Marks

 Financial Entites and Hyphens 68, 77

Quattro

 zero suppress 227

Quattro Pro 203

 1-2-3 sheets, reading 206

 Error messages 206

 FAQ 193

 installing F9 203

 macros 206

 manual recalc 205

 Paste as Values 33

 Report Wizard 45

 speed bar buttons 205

R

Range, Cells 224

Range Recalc in Excel 188

Range Recalculations 188

Ranges

 Named 187

RC mode in Excel 112

Reader, Acrobat 219

Recalc, in Lotus 1-2-3 198

Recalcing a range of cells	198	Password Window	118, 183, 195
Recalculation		segment	
Manual	187	Account Main	223
Recalculation mode		Segmenting accounts	84
Excel	187, 188	Segments	
Quattro	205	Descriptions	56, 58
Recalculation Time		Multi-segment descriptions	59, 75
Analyze Command	130	Server	
Recalculations		Help	183
Range	188	Server, the F9	182, 195, 225
Referencing Cells		commands	96
<i>See</i> Cell reference: Using		introducing	134
Relative Cell Reference	225	quitting	98
Report Wizard	39, 45, 46, 112, 114, 184, 225	topics, default	97
axis, defined	45	Setup window	100, 183, 195, 226
dimensions, defined	45	DataPreLoad	102
expanding lines	49	default company	100
formatting macros	51	period override	101
horizontal axis	50	return zero for account not found	101
lines, editing	49	rounding	101
lines, expanding	49	scaling	101
lines, selecting	49	system directory	100
model company	46	type	101
remaining axes	50	year	101
selecting lines	48	Silent Running	116, 132, 137, 142
standard lines	48	buttons	140
vertical axis	47	drill down window	142
Reports		ExecuteExcel4Macro	138, 144
creating	19	INI files	142
RepWiz Spreadsheet	51	macros	142
Return zero for account not found	101	NoShow parameter	138
Roll-up	225	Smart Templates	226
Rounding Values	101	Specifying	
RunF9Report	145	Accounts	67, 69
RunF9Report macro	114	Financial Entities	73
		Using BSPEC	77
S		Using BSPEC with WGL	64
SAD File	148	Balance time	88
<i>See also</i> Support Files: SAD		Budgets	89
Scaling Values	101	Columns (F9Data function)	61
SDESC function	56, 58	Companies	67, 91
Multi-segment descriptions	59, 75	Currency	68, 93
Security	118	Dates	67
Security in F9	223	Default parameter values	18, 63, 68
Password	224	CODATA Function	56, 60

Default parameters			
Setup Window		100	
Duration		88	
Future periods		88	
Periods		67, 85, 89	
Future		88	
with WGL function		64	
Wizard		119	
Time		87	
Topic		13, 93, 95, 96, 97	
Type		93	
Value type		88	
Year		67, 89, 92	
Spreadsheets			
block		219	
cell		220	
Character string		220	
creating		19	
e-mailing		31, 172, 184	
samples		30	
Subaccount Descriptions		34, 56, 58	
Multi-segment descriptions		36, 59	
Subsidiary Company			
Ownership		152	
Support Files			
Column Names		150	
Currency Conversion		156	
F9EDIT.XLL		147	
MAP		153, 155	
SAD		116, 148	
Tree		152	
Swap file, Windows		177	
T			
Technical Support			
E-mail		13, 134	
Templates		226	
accounting system specific		209	
button spreadsheet		213	
generic		209	
Text Substitution Operator		55, 56, 61, 78, 81, 226	
Time out, DDE		125, 200, 207	
Time specifier		87	
To Date		67	
To E-mail/Viewer			124
To Viewer/E-Mail			184
Excel			184
Toolbar			
Excel			
Creating			183
Images			189
Topic		96, 183, 195, 226	
DDE, Specifier		93, 95, 96, 97	
Specifier, DDE		93, 95, 96, 97	
Topic, default		97, 221	
Topic, Specifying		13	
Transaction Post			
Currency Conversion			156
Transactions			
accessing			108
between certain dates			110
Currency conversion			156
dates, selecting			110
displaying fields			110
fetching fields			110
periods			110
Transpose List			116
Tree File			152
			<i>See also</i> Support Files: Tree
Troubleshooting			
Btrieve			215
TSO operator		55, 56, 61, 78, 81, 226	
Type specifier		26, 68, 93, 101	
V			
Value Type			
Function Parameter			88
Values			
rounding			101
scaling			101
Version number			
viewing			120, 184
Viewer Add-in			
			<i>See</i> F9 Viewer
Virtual Memory			177
Visual Basic			138, 144
			<i>See also</i> Silent Running
ExecuteExcel4Macro			138, 144

W

W32MKSET.EXE	217
WBTRCALL.DLL	<i>See</i> Btrieve database: Unknown Error
WDESC Function	65
WGL Function	63
Using BSpec	64
Using Period Specifiers	64
Wildcard characters	72, 226
Windows	
Clipboard	220
Swap file	177
Virtual Memory	177
Wizard	
GL	40, 124
Period	50, 85, 110, 119, 160, 184, 224
Report	39, 45, 46, 112, 114, 184, 225
Write functions	
WDESC	65
WGL	63
Using BSPEC	64
Using Period Specifiers	64

Y

Y2K Compliance	226, 227
Year	100
Year 2000 Compliance	226, 227
Year Specifier	67, 89, 92
Year specifier	26

Z

Zero suppress	121, 184, 196, 197, 227
1-2-3, undoing with	123
1-2-3, using with	123
Column wise option	122, 126
Zeros	
replacing	101

