



Affordable, Adaptable ERP Software



VDT Reports
Course Workbook
Version 5.40

***Fourth Generation
Fitrix Visual Development Tool (VDT)
Reports Course Workbook***
Version 5.40

Restricted Rights Legend

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS252.227-7013. Fourth Generation Software Solutions, 700 Galleria Parkway, Suite 480, Atlanta, GA 30039.

Copyright

Copyright (c) 1988-2011 Fourth Generation Software Solutions All rights reserved No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language in any form by any means without the written permission of Fourth Generation Software Solutions

Software License Notice

Your license agreement with Fourth Generation Software Solutions, which is included with the product, specifies the permitted and prohibited uses of the product Any unauthorized duplication or use of Fitrix, in whole or in part, in print, or in any other storage and retrieval system is forbidden

Licenses and Trademarks

Fitrix is a registered trademark of Fourth Generation Software Solutions. Informix is a registered trademark of Informix Software, Inc. UNIX is a registered trademark of AT&T.

FITRIX MANUALS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, FURTHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE FITRIX MANUALS IS WITH YOU SHOULD THE FITRIX MANUALS PROVE DEFECTIVE, YOU (AND NOT FOURTH GENERATION SOFTWARE OR ANY AUTHORIZED REPRESENTATIVE OF FOURTH GENERATION SOFTWARE) ASSUME THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR, OR CORRECTION IN NO EVENT WILL FOURTH GENERATION BE LIABLE TO YOU FOR ANY DAMAGES, INCLUDING ANY LOST PROFITS, LOST SAVINGS, OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF OR INABILITY TO USE SUCH FITRIX MANUALS, EVEN IF FOURTH GENERATION OR AN AUTHORIZED REPRESENTATIVE OF FOURTH GENERATION HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY IN ADDITION, FOURTH GENERATION SHALL NOT BE LIABLE FOR ANY CLAIM ARISING OUT OF THE USE OF OR INABILITY TO USE SUCH FOURTH GENERATION SOFTWARE OR MANUALS BASED UPON STRICT LIABILITY OR FOURTH GENERATION'S NEGLIGENCE SOME STATES DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THE ABOVE EXCLUSION MAY NOT APPLY TO YOU THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS AND YOU MAY ALSO HAVE OTHER RIGHTS, WHICH VARY FROM STATE TO STATE

Fourth Generation Software Solutions
700 Galleria Parkway, Suite 480
Atlanta, GA 30339
<http://www.fitrix.com>

Corporate: (770) 432-7623
Fax: (770) 432-3447
E-mail: info@fitrix.com

Copyright (c) 1988-2009 - Fourth Generation Software Solutions Corporation - All rights reserved.

No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system or translated

Welcome to the Fitrix , Visual Development Tool (VDT) Report Tools Course Workbook. This manual is designed for use in the Fitrix VDT Training class. We hope that you find all of this information clear and useful.

The Fitrix 'Report Writer' is a character based end user report writer that can be used either for end users to create simple reports, or as a front-end for the Fitrix 'Report Code Generator'. The Fitrix Report Writer is not available in Windows Graphical Mode, but can be run from a Windows desktop in character mode.

The Fitrix Report Code Generator creates report and posting programs in FourJ's Genero Business Development Language. This is a fourth Generation Language and is shortened to just 4GL in this workbook.

If you have any questions about the Fitrix Report Code Generator or the Fitrix Report Writer please consult your manual or contact the Fitrix help desk at 1(800)374-6157. You can also contact us by email: support@fitrix.com. Please be prepared to offer your name, your company, telephone number, the product you are using, your Fitrix serial number and your exact question. We hope you enjoy using our products and look forward to serving you in the future.

Table of Contents

Preface	1
General Workbook Information.....	2
Coverage	2
Audience	2
Chapter 1.....	1
<i>Introduction to Fitrix Report</i>	1
Starting the Fitrix Report Writer	2
Fitrix Report Writer's Menus.....	2
File Pull-Down Menu	2
Report Pull-Down Menu.....	3
Delete Pull-Down Menu	4
Options Pull-Down Menu	5
Help Pull-Down Menu.....	5
Quit Pull-Down Menu.....	6
Creating a Simple Report.....	7
Report Description Window	7
Automatic Report Template.....	7
Print the Report	8
Displaying a Report to the Screen.....	9
Section Summary	10
Lab Exercise 1	11
Start Fitrix Report Writer	11
Create a Simple Report	11
Display the Report to the Screen.....	12
Chapter 2.....	13
<i>Creating Reports</i>	13
Creating a New Report	14
Defining the Report Description	14
Choosing the Columns to Appear on the Report	15
Arranging the Columns to Display on the Report.....	15
Printing Reports	17
Manipulating Report Data.....	18
Loading a Saved Report.....	19
Editing the Report Data	19
Grouping and Sorting.....	19

Selection Sets.....	21
Defining Selection Sets.....	21
Selection Data.....	22
Section Summary.....	23
Lab Exercise 2a.....	24
Defining the Report Description.....	24
Choosing the Columns to Appear on the Report.....	24
Adding Columns and Rearranging the Column Order.....	24
Lab Exercise 2b.....	25
Grouping Report Data.....	25
Sorting Data.....	25
Selection Criteria.....	27
Entering an SQL Filter Statement.....	28
Chapter 3.....	29
<i>Editing Reports</i>.....	29
Using the Fitrix Report Editor.....	30
Editing the Report Template.....	30
Column Detail Zoom.....	31
Adding a New Column.....	32
Creating "Hidden" Columns.....	32
Math Calculations.....	34
Formatting Data in the Column.....	35
Runtime Report Values.....	35
Report Blocks.....	37
Type of Report Blocks.....	38
Adding a New Report Block.....	38
Adding a Before or After Group Report Block.....	39
Adding Last Groups and Trailers.....	41
The Fitrix Report Runner.....	42
Section Summary.....	43
Lab Exercise 3a.....	44
Adding Columns in the Report Editor.....	44
Lab Exercise 3b.....	46
Math Formulas.....	46
Lab Exercise 3c.....	47
Define Groups in the Fitrix Report Editor.....	47
Lab Exercise 3d.....	48
Report Runner.....	48

Chapter 4.....	49
<i>Managing the Data Dictionary</i>	49
Descriptive Table and Column Names	50
Data Groups	50
Table Relationships.....	51
One-to-One Relationships.....	51
One-to-Many Relationships	52
Outer Joins	52
Section Summary	54
Lab Exercise 4a.....	55
Adding the Table Through dbaccess.....	55
Give Your Tables Descriptive Names	55
Lab Exercise 4b	57
Adding Your New Table to a Data Group	57
Adding a Column to the trcustomer Table.....	57
Adding the Table Relationship	58
Add Credit Information to Your "Customer Phone Numbers" Report.....	59
Using an Outer Join	59
Chapter 5.....	61
Fitrix Report Code Generator.....	61
Fitrix Report Code Generator Overview.....	62
Sample report.ifg File	63
Files Generated by the Fitrix Report Code Generator	64
Compiling Programs	64
Passing Criteria at Run-Time.....	64
Report Data Flow	65
Report Functions.....	66
Initialization Functions:	66
Loop on Detail Functions.....	67
End of Job Functions	67
Report Functions Commonly Modified	67
Referencing Fields not Printed in the Report.....	68
Basic Library Philosophy.....	69
Directory Structure.....	70
Section Summary	71
Lab Exercise 5a.....	72
Make a Simple Report	72

Export the Report from the Report Writer	72
Generate Code on the Report	73
Compile the Code	73
Run the Program	73
Lab Exercise 5b	74
Add the New Field with the Fitrix Report Writer	74
Generate Code on the Report	74
Compile the Code	74
Run the Program	75
Diff the Files	75
Chapter 6.....	77
<i>Using the Featurizer.....</i>	<i>77</i>
Report Code Generator Overview	78
A Function with Blocks	79
Block Command Examples.....	79
Where Do Block Commands Go?.....	80
The <code>start file</code> Command.....	80
Extension Files.....	81
Feature Set Files.....	81
Plugging in Features with a <code>.set</code> File.....	82
Custom Block tags in <code>report.4gl</code>	82
Section Summary	84
Lab Exercise 6a.....	85
Determine Where to Put Your Code	85
Use the Featurizer to Place Custom Logic into Code	85
Merging with the <code>.set</code> File	86
Merge and Compile with <code>fg.make</code>	86
Run the Program	87
Lab Exercise 6b	87
Adding in Report Selection Criteria.....	87
Create the Selection Criteria Form.....	87
Add the Selection Criteria Using an Extension File	88
Chapter 7.....	90
<i>Creating Posting Programs</i>	<i>90</i>
What is a Posting Program?.....	91
Creating the Posting Program	92
Creating the Report	92

Inserting the SQL Statements	92
Using prepare and execute	94
Using begin work, commit work, and rollback work.....	95
Section Summary	96
Lab Exercise 7	97
Create a New Program Directory	97
Add a Column with dbaccess.....	97
Generate Code.....	98
Add an SQL Statement with an .ext File	98
Merging with the .set File	99
Merge and Compile with <code>fg .make</code>	99
Run the Program	99
Verify Through dbaccess	99
Chapter 8.....	101
<i>Other Features</i>	101
Exporting Report Data	102
Post Processor Flexibility	103
Section Summary	104

Preface

The Fitrix Report Tools Course Workbook is an extensive aid for training on the products that make up the Fitrix Report Writer and the Fitrix Report Code Generator.

General Workbook Information

The Report Tools Course Workbook is an aide for training on the products that make up Fitrix Report. This workbook serves as a supplement to classroom training offered by Fourth Generation Software. This workbook is not designed to be used by itself as a self study tutorial.

Coverage

This workbook covers the following CASE Tools products:

- *Report* (Fitrix Report Writer, Fitrix Report Code Generator, and Featurizer)
- 132 Column Pager

This workbook does *not* cover the following CASE Tools products:

- VDT Application Code Generator (VDT Form Designer, User Control Libraries, VDT Application Code Generator)
- Fitrix Viual Menus

Audience

This workbook is divided into three parts; each part addresses a separate audience:

- Sections 1 through 3 address end-users with no technical knowledge, who use Fitrix Report Writer to create reports.
- Sections 1 through 4 address system implementers with some technical knowledge, who set up and support end-users.
- Sections 1 through 8 address programmers with extensive technical knowledge, who modify report code for end-users.

Chapter 1

Introduction to Fitrix **Report**

Major topics in this chapter include the following:

- Starting the Fitrix Report Writer
- Using Fitrix Report Writer's menus
- Creating a simple report

Starting the Fitrix Report Writer

Start Fitrix Report Writer by typing **fg.writer** at the LINUX prompt. If DBNAME is set in your .profile, there is no need to specify the name of the database prior to entering the program. If you need to, this is done by passing a "-d" flag to fg.writer, e.g., `fg.writer -d student{}`. For now and through out this workbook, replace the curly braces {} with the number of your student database.

Fitrix Report Writer's Menus

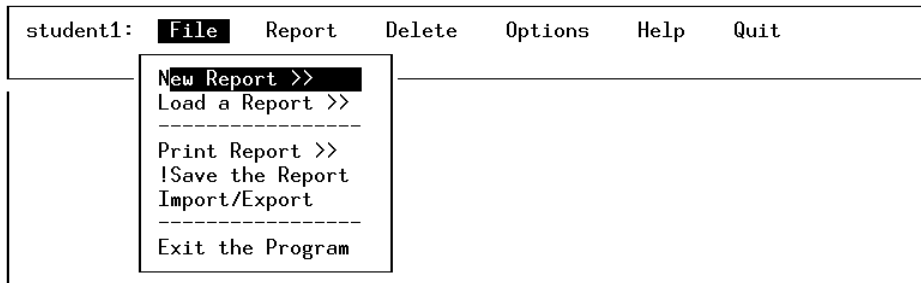
Fitrix Report Writer ring menu is the user interface for reporting information stored in the database without entering structured query language code. Notice that the database name, in this case "student1," is identified to the left of the ring menu options.

```
student1: File  Report  Delete  Options  Help  Quit
Load, save, and print reports
```

Each option on the main ring menu identifies a pull-down menu with additional options for activities in Fitrix Report Writer. To move between the menu options, type the first letter of the menu option or press the arrow keys and press [ENTER] to display the selected option's pull-down menu.

File Pull-Down Menu

The File menu lets you create and maintain report files.



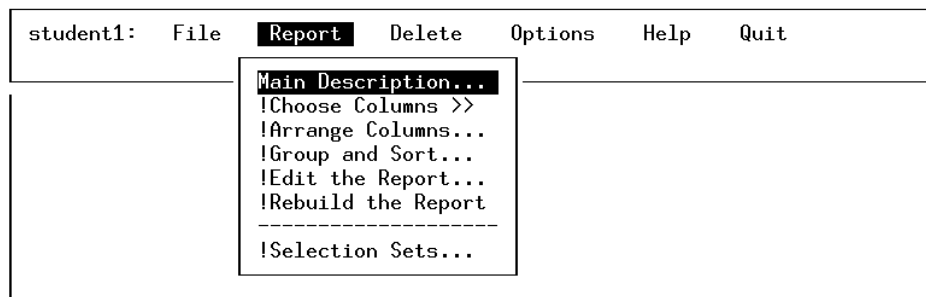
To select an option on the File pull-down menu, press the first letter of the option or use the arrow key to move to the option and press [ENTER]. The pull-down menus contain two characters which either proceed or follow the menu option. When an option is preceded with a !, the menu option is not available at the current time. When an option is followed by », selecting this option causes additional options to appear.

The File pull-down menu options include the following:

- **New Report** leads you through the Report Definition and creates a new report file.
- **Load a Report** provides a list of existing report files and allows the user to load an existing report.
- **Print a Report** lists existing report files and prints the selected report.
- **Save the Report** saves a newly created or edited report file.
- **Import/Export** allows the user to store report definitions in a text format. The format is saved in a file, which can be imported into another database.
- **Exit the Program** confirms that the user is ready to quit and exits to the LINUX prompt.

Report Pull-Down Menu

The Report menu option lets you build a new report or modify a current report.



You must be working with a current report or have begun a new report description before these options are available.

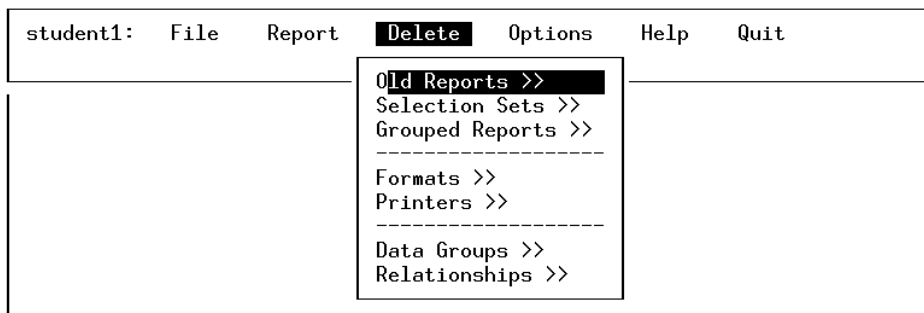
The Report menu options include the following:

- **Main Description** brings up the Report Description Form, where you define the report parameters. Parameters include database information, report format, who created the report, and when it was created.

- **Choose Columns** lists the available columns for your report based upon the database information you enter into the Report Description form.
- **Arrange Columns** lets you define the order that the columns will appear on the report and mark which columns are totaled.
- **Group and Sort** allows you to specify a column or sequence of columns by which to sort the report.
- **Edit the Report** allows you to make modifications to the report template that is created by the Report Writer.
- **Rebuild the Report** removes any modifications performed during the current work session by reverting to the most recently saved copy of the report.
- **Selection Sets** allows you to define selection criteria to define the data displayed in the report.

Delete Pull-Down Menu

The Delete menu lets you remove old report files and other data files used by the Fitrix Report Writer.

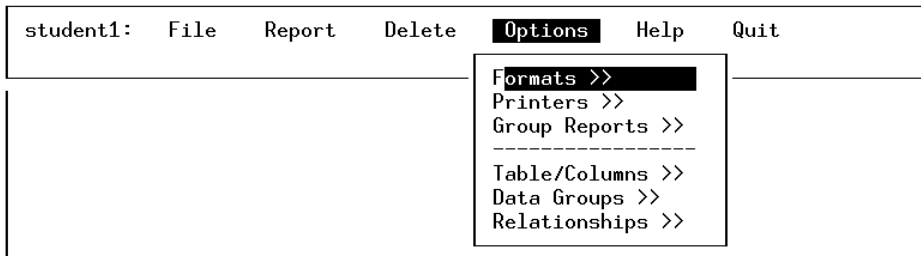


The Delete pull-down menu options include the following:

- **Old Reports** lets you delete a saved report.
- **Selection Sets** lets you delete selection sets.
- **Grouped Reports** lets you delete selected groups of reports defined under the Options menu.
- **Formats** lets you delete selected formats defined under Options.
- **Printers** lets you delete printer definitions defined under Options.
- **Data Groups** lets you delete selected data groups defined under Options.
- **Relationships** lets you delete selected table relationships defined under Options.

Options Pull-Down Menu

The Options menu lets you define and maintain report and database files.



The first three options let you set up reference files.

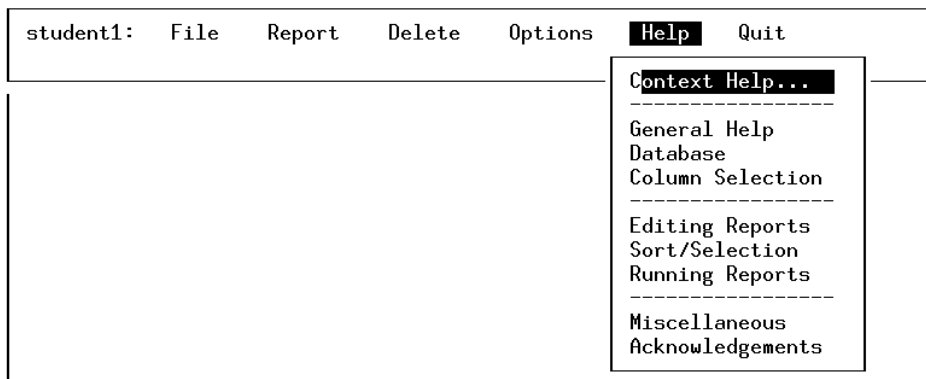
- **Formats** allows you to define multiple page templates for the report's output. Formats define the column width, length, and margins.
- **Printers** is the form on which you set up the printers to be used to print reports.
- **Group Reports** allows you to organize reports into like groups.

The last three options let you perform data administration tasks.

- **Table/Columns** allows you to assign descriptive names to the tables and columns in your database.
- **Data Groups** allows you to organize tables into logical groups.
- **Relationships** allows you to define relationships between tables in the database. These relationships, or "joins," should be defined by the data dictionary manager.

Help Pull-Down Menu

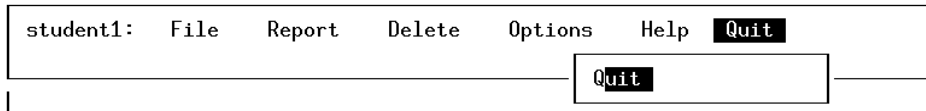
The Help menu lets you view detailed help information.



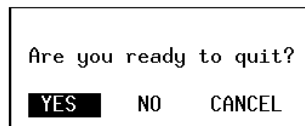
This includes context-sensitive help and general discussions on database administration, creating reports, and editing reports.

Quit Pull-Down Menu

The **Quit** menu lets you quit Fitrix Report Writer.



After selecting the Quit option, you are asked to confirm your request. Select YES if you want to quit, NO or CANCEL if you want to continue using Fitrix Report Writer.



Creating a Simple Report

You can create several types of reports using the Fitrix Report Writer.

When creating a report, follow these basic steps:

1. Identify the data to be used in the report.
2. Define any totaling, or data manipulation, that results in a display element.
3. Format how the data appears on the report.
4. Print or save the report.

Report Description Window

The Fitrix Report Writer automatically leads you through the steps of creating a new report. The first steps to creating a new report include creating the report definition:

1. Select the Data Group
2. Choose a Table
3. Choose a Report Format
4. Name the Report

Fitrix Report Writer then leads you through the steps to creating a specific report.

5. Select the columns in the report
6. Arrange the columns in the report.

Once the base report description information is defined, Fitrix Report Writer automatically creates a report template.

Automatic Report Template

The report template appears on the screen in the Fitrix Report Editor.

```

[ESC] Save [DEL] Cancel "[ " New Column "{ " New Group
Report Name: Customer Information
===== (1,1) ===== (Zoom) =
      10      20      30      40      50      60      70
.....:.....:.....:.....:.....:.....:.....:.....:
HDL date      ]           [title      ]           Pa
HD-----:-----:-----:-----:-----:-----:-----:
Cust No      [_____]
5 First Name  [_____]
  Last Name   [_____]
  Company Name [_____]
  Address Line #1 [_____]
  Address Line #2 [_____]
10 City       [_____]
  St          [ ]
  Zip         [____]
  Phone       [_____]
:
15 :
:
:

```

This template displays a preview of your report layout and how your report will appear when it is printed.

Print the Report

To print the report, choose Print Report from the File pull-down menu. Then you choose a destination to which to send the report. You have the following options:

```

[ENT] Select [ESC] Exit
=====
  Choose a Destination
-----
Display to Screen
Output to Printer
Write to File
Transmit FAX
Archive to Database
-----
Export Data >>
Reporting Options >>
Quit

```

- **Display to Screen** displays the report to the screen. You can move through the report using the options on the Pager ring menu.
- **Output to Printer** prints a hardcopy of your report to the selected printer.
- **Write to File** saves the report to a file.
- **Transmit FAX** transmits the report via fax if a fax server is setup.
- **Archive to Database** stores the text of your report in tables in the database.

You can also select one of the following:

- **Export Data** prints the report to an ASCII file.

- **Reporting Options** allows you to define several options to control the detail of the report. These options include Summary Only, Test Length, Form Alignment, Data Selection, and Store Raw Data.
- **Quit** returns you to the File pull-down menu.

Displaying a Report to the Screen

If you choose to display the report to the screen, the report appears in the Report Pager.

```

Pager: Next Prev Right Left Top Bottom Scroll Up-scroll Quit
View the next page.
=====
08/11/2010                Customer Information                Page:
=====
Cust No           101
First Name        Ludwig
Last Name         Pauli
Company Name      All Sports Supplies
Address Line #1   213 Erstwild Court
Address Line #2
City              Sunnyvale
St                CA
Zip              94086
Phone            408-789-8075

Cust No           102
First Name        Carole
Last Name         Sadler
Company Name      Sports Spot
Customer Information: columns 1 to 76 lines 1 to 18 of 396

```

The Pager ring menu options help you to view different sections of the report. Move the cursor to the self-explanatory ring menu selections using the arrow keys or by keying the first letter of the selection, which is capitalized. Press [ENTER] to execute the command.

Section Summary

- From the Fitrix Report Writer ring menu, you perform most reporting activities using the File and Report menu options.
- You manage your data dictionary management and perform delete activities from the Delete and Options selections on the ring menu.
- To create a simple report, identify the data to be used in the report, define manipulations to the data, format how the data appears on the report and print or save the report.
- Fitrix Report Writer gives you the ability to set up and maintain a data dictionary using the options for data administration under the Options menu.

Lab Exercise 1

Objective: Create a simple report and display it to the screen.

Start Fitrix Report Writer

1. **Execute Fitrix Report Writer using the `student{}` database by typing:**

```
fg.writer -d student{}
```

Substitute the name of your database for `student{}` or set `$DBNAME` to the name of your database and leave off the `-d` flag.

Create a Simple Report

The first step to creating a new report is to define the report description. Fitrix Report Writer will lead you through this process.

1. **From the File pull-down menu, select "New Report."**
2. **Choose "Training Data Set" as the Data Group.**
3. **Choose the "Customer Information" table.**
If the Report Format box appears, press [DEL].
4. **Name the report "Customer Phone Numbers."**
5. **Press [ESC] to store the Report Description.**
6. **Use the arrow keys to move to the following columns and press [ENTER] to choose the columns to appear on the report.**

Customer Number
Company Name
Phone Number

Note

When you choose a column, an asterisk will appear next to the column name indicating this column is selected. To remove a column, press [ENTER] again to toggle off the column selection.

7. **Press [ESC] to store the columns in their default order.**
8. **Accept the sequence shown by pressing the [ESC] key.**

Notice how the report template is displayed on the screen behind the menus.

9. Press [CTRL]-[c] to avoid choosing a Selection Set.
10. Press [CTRL]-[c] again to clear any open pull-down menus and view the entire report template.

At this time a preview of the report template is visible on the screen. As you are working with more detailed reports, you can view the effects of any changes made in your report definitions as you are making them.

```

student1:  File  Report  Delete  Options  Help  Quit
Load, save, and print reports

          10      20      30      40      50      60      70
.....:.....:.....:.....:.....:.....:.....:.....:
HD[date  ]          [title          ]          Pa
HD=====
HD Cust No   Company Name   Phone
HD-----
5 [         ] [             ] [             ]
:
:
:
:
10 :
:
:
:
:
15 :
:
:
:

```

Display the Report to the Screen

1. From the File pull-down menu, select "Print Report."
2. Press [ENTER] to save the report, with the default name: "Customer Phone Numbers."
3. If prompted for a Selection Set, press [CTRL]-[c] as you have yet to define any selection criteria.
4. Select "Display to Screen" to display the "Customer Phone Numbers" report to the screen.
5. Use the Pager ring menu options to move through the report.
6. Select Quit to return to the Report Destination window.
7. Press [CTRL]-[c] or Quit to return to the Main Report Writer menu and exit the program.

Chapter 2

Creating Reports

Major topics of this chapter include the following:

- Manipulating Report Data
- Printing Reports
- Grouping Data in a Report
- Creating Selection Sets

Creating a New Report

The Fitrix Report Writer leads you through the basic steps of creating a report. As you begin creating a report, you define the report description, choose the columns to appear on the report, and specify the order in which they should be arranged.

Defining the Report Description

To define a report description:

1. Choose a data group. A data group is a predefined group of related tables in your database. Defining data groups allows you to organize related tables in your database from which you can create your report. Data groups are defined by the database administrator using the Options selection from the main ring menu.
2. Choose a Table. The list of tables displayed are familiar names describing all the tables in the selected data group.
3. Complete the Report Description. Once you have selected the data group and main table, you are led to the Report Description window. This form is divided into three sections.

Report Description			
[ESC] to Store		[DEL] to Cancel	
		[CTRL]-[w] Help	
Database	:	student1	
Report Name	:	██	
Data Group	:	Training Data Set	
Table Name	:	Customer Information	
Table	:	trcustomer	
Selection Set	:		

Report Format	:		
Report Width	:	80	Left Margin : 0
			Right Margin : 0
Report Length	:	66	Top Margin : 0
			Bottom Margin : 0

Author	:	kittyk	Private Report? :
Date Changed	:		Date Created : 08/12/10

Enter the name of this report.			

4. The top section contains report file information. You assign a name to the report here. This name prints at the top of your reports.
5. The middle section contains the report format information. Report formats are predefined under Options. A format is a predefined page size for your report. The type of information set in the report format is the report column width, report length in number of lines, and margin spacing.
6. The bottom section identifies who created the report, when it was created, and when it was last changed. You also have the ability to define the report as a private report available only to the author.

Choosing the Columns to Appear on the Report

You are automatically led to the next logical step in creating a report: selecting which columns the report will use. The columns listed are the columns found in the main table defined in the report description. You are not limited to reporting from these columns. You can also select columns from tables that are defined as being related to or joined to the main table. The columns are listed by their familiar, or descriptive, names.

```

[ENT] Select [ESC] Exit [ESC] Done
===== (Zoom) =
  Choose the Columns for the Report
-----
*Customer Number      Zip Code
First Name            Phone Number
Last Name              Credit Code
Company Name           Sales Person Code
Address Line #1        Card Number
Address Line #2        Expiration Date
City                   Card Holder
State

```

From this screen, there are two ways to select columns to appear on the report.

1. Move to the column name using your arrow keys and press [ENTER] to select the column. Note that an asterisk (*) appears in front of the column name indicating that this column has been selected.
2. Press [CTRL] - [z] and you will be prompted to select a data file. A list of columns is displayed for you to select from. The data files are other tables that have a relationship to the main table defined in the report description. In addition to selecting columns from the main table, you have the option of selecting columns from related tables. Table relationships are defined by the database administrator from the Options pull-down menu.

```

[ENT] Select [ESC] Exit
=====
  Choose a Data File
-----
Sales Orders
State Definitions
-----
Order Line Items
Manufacturer Definitions

```

Arranging the Columns to Display on the Report

The next step is to arrange the columns in the order you want them to appear on your report. In addition to arranging the order of the columns for the report, you can add a column and total numeric column values.

The columns appear in the order they were selected. To change their order on the reports, alter the number in the Sequence field. For example, entering "1" (for "Company Name") in the Sequence field indicates

that it will be the first column of the report. It will appear either as the column on the far left of the report or as the first column if the report data is stacked.

```

Arrange Columns to Display on the Report
[ESC] to Store [DEL] to Cancel

- Sequence - Column ----- Total? -
  1 Company Name
  2 Customer Number
  3 Stock Number
  4 Quantity Y
  5 Line Extension Y

-----
Enter the order of the column on the report.
  
```

You can add another column to the report from this entry screen by typing a column name under the Column field. You also have the option of totaling a column by entering "Y" in the Total field.

Once you have stored your column arrangements, the Fitrix Report Writer displays the template in the background of the pull-down menus, allowing you to preview the report layout as you create it.

```

student1:  File  Report  Delete  Options  Help  Quit
Build and modify a report

    10      20      30      40      50      60      70
  .....:.....:.....:.....:.....:.....:.....:.....:
  HD[ date   ]      [title]                      ] Pa
  HD-----
  HDCompany Name   Cust No   Stk No   Qty   Extension
  HD-----
  5 [ _____ ] [ _____ ] [ ____ ] [ ____ ] [ _____ ]
  LR
  LRGrand Totals
  LR                [ ____ ] [ _____ ]
  :
10 :
  :
  :
  :
  :
15 :
  :
  :
  :
  
```

Printing Reports

When you select the Print Report option, you will be prompted for a name. You will be asked to specify a selection set. Once you have named the report and specified the selection set, you can choose a destination for the report. You have the following options:

```
[ENT] Select [ESC] Exit
=====
Choose a Destination
-----
Display to Screen
Output to Printer
Write to File
Transmit FAX
Archive to Database
-----
Export Data >>
Reporting Options >>
Quit
```

Display to the Screen

This option sends the output of your report to the screen. Use your arrow keys to highlight an option and press [ENTER] to select it. You can also type the first letter in the option's name, such as "p" for Prev (Previous) to highlight it.

```

Pager:  Next Prev Right Left Top Bottom Scroll Up-scroll Quit
View the next page.
=====
08/12/2010                Customer Phone Numbers                Page:
=====
Cust No   Company Name   Phone
-----
    103 Phil's Sports   415-328-4543
    106 Watson & Son   415-389-8789
    107 Athletic Supplies 415-356-9876
    108 Quinn's Sports  415-544-8729
    109 Sport Stuff     408-723-8789
    110 AA Athletics    415-743-3611
    111 Sports Center   408-277-7245
    112 Runners & Others 415-887-7235
    113 Sportstown     415-356-9982
    114 Sporting Place  415-886-6677
    115 Gold Medal Sports 415-356-1123
    116 Olympic City    415-534-8822
    117 Kids Korner     415-245-4578
    118 Blue Ribbon Sports 415-655-0011
Customer Phone Numbers: columns 1 to 76 lines 1 to 18 of 132

```

Report Pager Ring Menu Options

- **Next** allows you to page down and view the next page.
- **Prev** allows you to move back up to the previous page.
- **Right** allows you to move the screen display to the right, should you have more columns across the report than are displayed to the screen.
- **Left** allows you to move the screen display of the report back to the left.

- **Top** allows you to move to the top of the report.
- **Bottom** allows you to move to the end of the last page of the report.
- **Scroll** allows you to move down through the report one line at a time.
- **Up-scroll** allows you to move up towards the beginning of the report one line at a time.
- **Quit** exits from the report display and returns to the Select Destination menu.

Output to Printer

Prompts you to choose from the printers set up to print reports, and routes your report to the printer.

Write to File

Prompts you to enter a file name for the report's destination, and saves your report to the file.

Archive to Database

Stores the text of your report in two tables in the Fitrix Report Writer database.

Export Data

Outputs the detail of your report to an ASCII file, which can be imported by other software programs.

Reporting Options

Allow you to define several options to control the report output.

- **Detail or Summary** toggles between printing detail only or summary only. Detail prints out every row of the report. Summary prints only group headings and their corresponding subtotals.
- **Test Length** allows you to define how many rows to print. Use this to preview a sample before printing the entire report.
- **Form Alignment** allows you to send positioning marks to the printer so you can see if the forms are aligned properly.
- **Data Selection** allows you to select multiple selections sets for a report.
- **Create Data Table** stores the raw data of your report to a table, allowing subsequent reports from this table to run faster.

Manipulating Report Data

Report data can be manipulated by the following methods: changing the Arrange column sequence, grouping data and calculating subtotals, and sorting the data.

Once you have created a report you may want to make editing changes to that report or to another saved report.

Loading a Saved Report

To load an existing report, from the File pull-down menu select Load a Report. A screen is displayed prompting you to Choose a Report. The options are divided into two parts:

1. The first contains Current Report or Last Report. Selecting Current Report will return you to the report on which you are working. Selecting Last Report loads the last report you were working with.
2. The second portion of this screen displays a list of all reports created and saved for this database:

```

[ENT] Select  [ESC] Exit
=====
      Choose a Report
-----
Current Report
Last Report
-----
Customer Information
Customer Orders
Customer Phone Numbers

```

Editing the Report Data

Editing options can be accessed from the Report pull-down menu. The first three options on the Report pull-down menu access the input programs that the Fitrix Report Writer automatically leads you through when creating a new report. Changes to column information can be entered by selecting one of the first three options.

- **Main Description** allows you to modify the report file definition.
- **Choose Columns** allows you to change the selection of columns for the report. If changes are made to the columns selected for the report, the program automatically brings up the Arrange Column screen.
- **Arrange Columns** allows you to change the column display on the report or define a column for totaling.

The four selections following allow you to create more complex reports.

- **Group and Sort** allows you to define criteria to group data, and to sort the display order of the data.
- **Edit the Report** allows you to access the report template viewed on the screen and make changes to the report through the template.
- **Rebuild the Report** recalculates the math functions and redraws the report template display.
- **Selection Sets** allows you to create a new selection set or choose from defined selection sets. A selection set specifies the selection criteria for data included on the report.

Grouping and Sorting

More functionality can be added to your report by defining a column to group data by, and to sort the data in ascending or descending order. When Group and Sort is selected, the list of columns chosen for this

report is displayed. When a column is chosen as a group, that column will appear as a subheading for the group. If any columns in the report are marked for totaling, a subtotal will be calculated for each group defined.

Once a column is selected, the Sort the Report by these Columns screen is displayed.

```

Sort the Report by these Columns
[ESC] to Store [DEL] to Cancel

- Seq - Sort? - Group? - Page Bk.- Column -----
   1   A     Y     A     Customer Number
   2   A     Y     N     Stock Number

-----
Group report by this column (for totals)? (Y/N)
    
```

- **Seq** is an abbreviation for sequence. If you have data grouped by more than one column or criteria, define the sequence by which to sort the columns. In the above example, data will first be grouped by the Customer Number then by Order Number.
- **Sort?** allows you to sort a column in “A” (ascending) or “D” (descending) order.
- **Group?** allows you to mark columns that you want subtotaed. A “Y” indicates the column will be subtotaed.
- **Page Bk.** is an abbreviation for page break. You can define a page break before or after a grouping, or select “N” for no page break at all. The above example has an “A” defined for Customer Number. This will print a new page for each customer number group.
- **Column** displays the name of the column which is to be grouped. By using [CTRL]-[z], you Zoom to additional columns that can be added from this input screen.

The following report template has two groups defined. The first group appears with the subheading Customer Number. The second group, indented under the Customer Number subheading, is Order Number. Each grouping has a corresponding subtotal calculated for the Extension column, which is marked for totaling.

```

Sort the Report by these Columns
[ESC] to Store [DEL] to Cancel

- Seq - Sort? - Group? - Page Bk.- Column -----
   1   A     Y     A     Customer Number
   2   A     Y     N     Order Number

-----
Group report by this column (for totals)? (Y/N)
    
```

Below is the report displayed to the screen:

```

Pager: Next Prev Right Left Top Bottom Scroll Up-scroll Quit
View the next page.
=====
08/12/2010 Customer Orders Page:
=====
Cust No: 101
Order No: 1002
Company Name Cust No Order No Stk No Qty Extension
-----
All Sports Supplies 101 1002 3 1 $240.00
All Sports Supplies 101 1002 4 1 $960.00
Subtotals for 1002 -----
2 1200.00
Subtotals for 101 -----
2 1200.00
Customer Orders: columns 1 to 76 lines 1 to 18 of 792
    
```

Selection Sets

A selection set is a group of columns and their selection criteria that determine what data is displayed on the report. Selection sets are not tied directly to any given report. A report and a selection set are two independent elements. All reports that share the same main table (as defined in the report definition) can use any selection set that uses the same table as the main table.

Defining Selection Sets

From the Report pull-down menu, Selection Sets can be chosen to define a new selection set or modify an existing set. If no previous selection sets have been defined, the only option on the Choose Selection Set menu is New Selection Set.

The first step to defining a selection set is to choose the column by which to select or sort the data.

Once a column has been chosen, the Sorting and Data Selection Columns window appears.

```

Sorting and Data Selection Columns
[ESC] to Store [DEL] to Cancel

- Sequence - Sort? - Selection Data ----- Column -----
  1         A         Customer Number
  2         A         Company Name

-----
Enter the order of the column for sorting (if any).
    
```

Note that this screen looks similar to the Arrange Columns screen and the Group and Sort screen, but there is a new column, Selection Data.

- **Sequence** lets you prioritize the columns for sorting.
- **Sort?** allows you to sort in "A" (ascending) or "D" (descending) order.
- **Selection Data** lets you define your selection criteria.
- **Column** displays, adds, or updates columns for selecting and sorting.

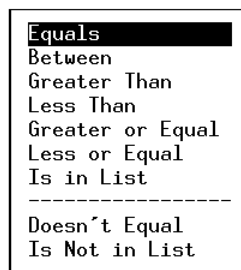
Selection Data

The Selection Data field allows you to build simple or complex selection clauses to limit the data displayed on your report. There are two ways to enter a selection clause.

1. If you are familiar with SQL "select" statements, you may manually enter your SQL selection clause by typing it directly into the Selection Data field.

The Selection Data field can store an SQL statement up to 100 characters long. If your statement requires more than 100 characters, then you can continue on the next row of the Data Selection field. The following key strokes can be used for editing your select statement:

- [CTRL] - [a] inserts a space
 - [CTRL] - [x] deletes a character
 - [CTRL] - [d] delete to end of line
 - [CTRL] - [u] undo an edit
2. If you are not familiar with SQL selection statements, there is a Zoom on the Data Selection field which helps you build your selection statement.



A list of operators is displayed for you to choose from. Once you have selected an operator, you are prompted for more specific selection criteria. Comparison operators differ depending upon the type of column, that is, whether it is character, numeric, or date column.

Once you have entered the criteria, another window appears allowing you to define an "And" clause, which further restricts the selection. You can also choose an "Or" clause, which expands the criteria.

Section Summary

- When creating a new report, the Fitrix Report Writer automates the steps to create a basic report. These steps include the following:
 - Defining the report file description.
 - Selecting the columns to appear on the report.
 - Arranging the columns to appear on the report.
 - Viewing the report template.
- Column templates can be modified and edited using the options under the Report pull-down menu.
- The report template, which appears in the background of the pull-down menus, allows you to view the changes made to the report definition while you are making them. You do not have to print or display the report to view your changes.
- Reports can be:
 - displayed to the screen,
 - sent to a defined printer,
 - written to a file,
 - archived to the database,
 - exported to an ASCII File.
- Report data can be grouped by a specified column or by multiple columns. Sub-totals are printed for grouped data columns.
- Selection sets are defined to limit which data appears on the report. Selection sets are independent of individual reports. They are tied to the main table defined in the report description and can be used for any report with the same main table definition.
- Selection data can be entered in the following two ways:
 - Typing an SQL select statement in the Selection Data field,
 - Zooming to define selection criteria.

Lab Exercise 2a

Objective: To create a new report that will display all sales orders and their items. The total of all orders will appear at the bottom of the report.

Defining the Report Description

1. Create a new report using the Training Data Set.
2. Select the "Sales Orders" table.
If the Report Format box appears, press [CTRL]-[c].
3. Name the report "Orders."

Choosing the Columns to Appear on the Report

1. Choose the following columns to appear on the report and arrange them in the following order:
 - Order Date
 - Quantity (From the Order Line Items table)
 - Line Extension (From the Order Line Items table)
2. Total the amounts in Line Extension column.
3. Save and display the report to the screen.

Adding Columns and Rearranging the Column Order

1. Return to the Choose Columns window and add the following columns to the report. (These columns are in the Items in Inventory table.)
 - Item Number
 - Item Description
 - Unit Price
2. Arrange the columns in the following sequence:

```

: Arrange Columns to Display on the Report
: [ESC] to Store [DEL] to Cancel
:
: - Sequence - Column ----- Total? --
:   1      Order Date
:   2      Item Number
:   3      Item Description
:   4      Unit Price
:   5      Quantity
:   6      Line Extension           Y
:
: -----
: Enter the order of the column on the report.

```

1. Order Date
2. Item Number
3. Item Description
4. Unit Price
5. Quantity
6. Line Extension

3. Display the report to the screen.

Lab Exercise 2b

Objective: To expand the Orders report by adding subtotal logic to the Order Number and Customer Number columns; and by calculating a total for all Customer Orders.

Grouping Report Data

1. From the Report pull-down menu, select the Group and Sort option.
2. Group your report by the Order Number column and sort the orders in ascending order.

Notice the change in your report by viewing the report template behind the pull-down menu.

Sorting Data

1. Add an additional group, Customer Number, with a grouping sequence of 1.

Orders on the report will be grouped within customer numbers, so customer numbers will be your first group.

2. Save this report under the new name "Customer Orders."
3. Add a page break so a different page is printed for each customer.
4. Add a third column, Stock Number, for sorting only.

It is not necessary to group orders by the Stock Numbers as they are already grouped by order number.

5. Save this report under a new name: "Customer Orders Summary."
6. Display the report to the screen and page down through each customer order.

Pager: <input type="checkbox"/> Next <input type="checkbox"/> Prev <input type="checkbox"/> Right <input type="checkbox"/> Left <input checked="" type="checkbox"/> Top <input type="checkbox"/> Bottom <input type="checkbox"/> Scroll <input type="checkbox"/> Up-scroll <input type="checkbox"/> Quit Go to the top.						
08/12/2010		Customer Order Summary			Page:	
Cust No: 101						
Order No: 1002						
Stk No: 3						
Order Date	Item No	Description	Price	Qty	Extension	
06/01/2009	3	baseball bat	\$240.00	1	\$240.00	
Subtotals for 3					240.00	
Stk No: 4						
Order Date	Item No	Description	Price	Qty	Extension	
06/01/2009	4	football	\$960.00	1	\$960.00	
Customer Order Summary: columns 1 to 76 lines 1 to 18 of 462						

Lab Exercise 2c

Objectives: You will define two Selection Sets, first, to select only customer orders for footballs, then you will be entering an SQL filter statement directly in the Selection Data field.

Selection Criteria

1. Load the "Orders" report.
2. From the Report pull-down menu choose "Selection Sets."
3. Add a new selection set.
4. Deselect all columns except the Item Description column in the Items in Inventory table. You deselect a column with an asterisk (*) by using the arrow key to move to the column and pressing [ENTER]. Deselect the Order Date column.
5. Zoom to the Items in Inventory Table.
6. Deselect Item Number and Unit Price. Leave Item Description selected. Press [ESC] to save your changes.
7. Zoom to the Items on Order Lines table. Deselect Quantity and Line Extension. Press [ESC] to save your changes.
8. Press [ESC] again to save your changes.
9. For Item Description, define the Sequence as 1 and sort in ascending order.
10. Zoom on the Selection Data column and select Equals.
11. Enter "football" as the data criteria. Press [ESC] to save your changes. The chose Done.
12. Save the selection set as "Footballs," but do not define this as the default for the Orders report.
13. Print the Orders report with Footballs as the selection set.

Entering an SQL Filter Statement

1. Load the "Orders" report and choose "Selection Sets" from the Report pull-down menu.
2. Choose a new selection set.
3. Choose the Payment Date column as the column by which to select data.
4. In the Selection Data column, type the following SQL select statement: `@ is null`
5. Save the Selection Set under the following name: "Unpaid Orders." Do *not* define this as the default Selection Set for the "Orders" report.
6. Print the "Orders" report so that it shows unpaid orders only.
7. Print the "Orders" report so that it shows all orders.
8. Print the "Customer Orders Summary" report of all unpaid orders.
9. Print the "Customer Orders Summary" so that it shows all orders.

Chapter 3

Editing Reports

Major topics in this chapter include the following:

- Using the *Report* Editor
- The Column Detail Zoom
- Modifying Report Blocks
- Using the *Report* Runner

Using the Fitrix Report Editor

The Edit the Report option from the Report pull-down menu allows you to make changes directly to the report template generated by the automatic column formatter. Once you modify your report using the Edit the Report option, you lose the ability to make changes using the pull-down menus.

In the Report Editor, you have four options displayed at the top of the screen:

- [ESC] saves your report
- [CTRL][C] returns you to the pull-down menu
- "[" allows you to add a column
- "{" allows you to add new groups of data.

You can also scroll up and down and to the left and right using the arrow keys. Rulers along the top and left edges of the report indicate the column and line numbers. Toward the upper right-hand corner on the double dotted line, the cursor location is displayed in parentheses. For example, as displayed in the example above, (1 , 1) means that the cursor is located at line 1 and column 1.

Editing the Report Template

Once you are in the Report Editor, you can add, delete, or change columns, groups, and text by typing in the report template. The following 4GL-defined editing keys are also available to use while editing your report.

Key(s)	Action
[CTRL]-[a]	insert space
[CTRL]-[x]	delete character
[CTRL]-[d]	deletes to end of line
[CTRL]-[p]	pastes a line
[CTRL]-[u]	undo an edit
[F1]	insert a blank line
[F2]	delete a line
[define a new column group
{	define a new group
[ESC]	save
[CTRL]-[c]	cancel

You can also cut and paste a line in the report by selecting [F2] to delete a line, which actually puts it to a temporary buffer. Then paste the line by pressing [CTRL]-[p].

Column Detail Zoom

The Column Detail Zoom can be accessed directly from the Report Editor by pressing [CTRL]-[z]. When your cursor is placed on a column in the report template, (Zoom) appears in the upper right-hand corner on the double dotted line, indicating that the Zoom option is available for accessing the Column Detail Zoom screen.

```

[ESC] Save  [DEL] Cancel  "[" New Column  "{" New Group
Report Name: Orders
===== (5,14)===== (Zoom)=
                               70
                               .....
                               Pa
-----
[ESC] Save  [Del] Cancel  [CTRL]-[w] Help
Column Detail Zoom
-----
Column Name : trstock.stock_num
              Item Number
5  Format      : right
   Hidden?    : N
   Table Name : Items in Inventory
              trstock
10  Report Block : Report Body
   Column Type : column
   Position Row : 5  Column : 13  Length : 6
-----
15  Enter a unique name for this column.

```

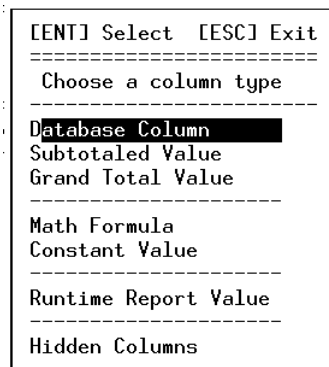
The first portion of the screen contains detailed information about the column:

- **Column Name** displays the 4GL table name and column name on the first line and the familiar name on the second line.
- **Format** allows you to define the column justification or numerical formatting.
- **Hidden?** allows you to mark "Y" or "N" to print or hide a column.
- **Table Name** displays first the Fitrix Report Writer descriptive name, and below it the 4GL table name.
- **Report Block** displays where this column is in the report block definition. This is not an entry field. It is governed by the location of your new column and where you pressed "[".
- **Column Type** displays the type of column. Column types are column, Subtotaled, Grand Total, Math Formula, Constant Value, Runtime Report Value, and Hidden.
- **Position Row/Column/Length** gives the physical location and size of the column as it appears on the report.

The bottom portion of the screen displays the math formula if the column is a formula column. To edit an existing math formula, press [CTRL]-[z].

Adding a New Column

To add a new column press "[", which brings up the Choose a Column Type menu. You will be prompted to select a column type.



There are several different types of columns including the following:

- **Database Column** displays a list of columns in the main database table for the report. Zoom is available to list the columns in related tables. If a column is selected, you will be prompted with the Column Detail Zoom to make any changes to the column detail information.
- **Subtotaled Value and Grand Total Value** are columns that can be added to the report when you have added report groups through the Report Editor.
- **Math Formula** lets you add a field to the screen to display the value of a math formula.
- **Constant Value** allows you to define a constant value to be displayed in that column.
- **Runtime Report Value** allows you to place special values on the report that are specific to the time the report is printed. These special values include system-generated page number, time and date stamping, etc.
- **Hidden Columns** displays a list of all columns marked to not display on the report.

Once the column type is selected and the column detail is saved, the column is activated on the report.

Creating "Hidden" Columns

The Report Editor allows you to hide a column that you do not want to print on your report. This is commonly used in creating math formulas with column values you do not want displayed on the report.

To hide a column:

1. **Place the cursor on the column.**
2. **Press [CTRL]-[z] to modify the column detail.**

3. Enter "Y" in the Hidden? field.

4. Press [ESC] to save.

Once you have saved the column detail information, the column is removed from the Report Editor screen.

To list all hidden columns on your report:

1. Press "[" to display the Choose a Column Type menu.

2. Press "H" to display a list of all hidden columns on your report.

If you wish to reveal a column that has been hidden on your report or to modify the hidden column detail:

1. List all hidden columns on your report.

2. Select the column from the window of the column you wish to reveal.

3. The Column Detail Zoom appears for you to change the Hidden? field to "N."

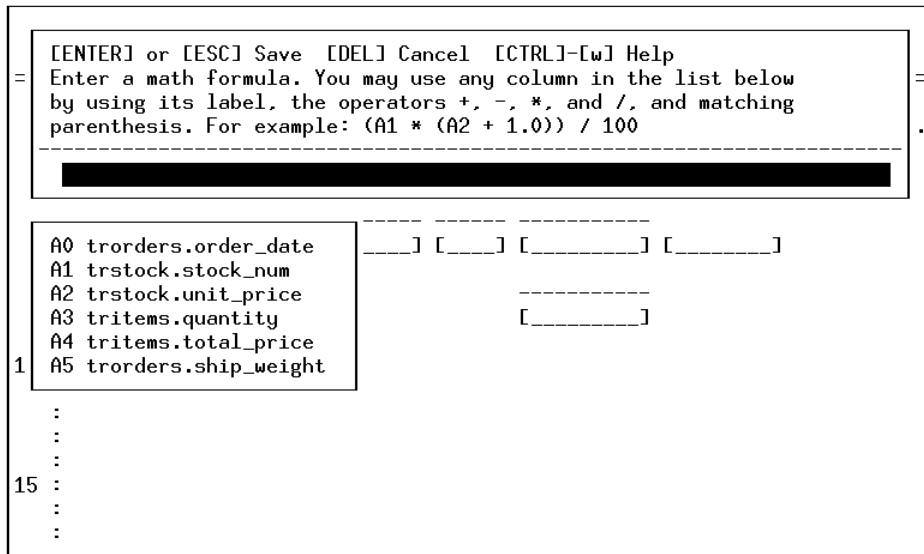
4. Press [ESC] to return to the Report Editor and view the column.

Math Calculations

The Fitrix Report Writer allows you to create math columns. You must be in the Report Editor to define math formulas.

To create math formulas:

1. Press "[" to add a new column.
2. Select Math Formula from the menu.



3. Enter the math formula in the scrolling window.

Columns that are available to perform calculations within the report block will be listed in a window below the main formula window. The columns available to be used in the math formula are numeric columns in that Report Block only.

Once the math formula is saved, the Column Detail Zoom screen appears for you to define the numeric format of the new math column.

To edit an existing math formula:

1. Press [CTRL]-[z] to display the math column detail.
2. Press [ENTER] to move to and highlight the block at the bottom of the window.
3. Press [CTRL]-[z] to display the scrolling math formula definition window.
4. Edit the math formula.

If you want to use a subtotal, grand total, or math formula amounts in your math formula, you must place the new math formula after (either to the right on the same row or on a row below) the calcu-

lated value. The Fitrix Report Writer reads values from left to right. For a calculated value to be included in a formula, its value must first be calculated.

Formatting Data in the Column

The text justification or numbered formatting can be defined in the Column Detail Zoom screen in the Format field.

Justification choices include the following:

- **right** moves the text to the right edge of the column display.
- **center** centers the text within the column.
- **left** moves the text to the left edge of the column display.
- **pushleft** moves the data in the field to the left and inserts a leading space. This is helpful when making address labels. It leaves only one space between the city and the state even though the length of the city column varies.
- **flushleft** moves the data in the column to the left leaving no spaces between columns. This is useful when two columns are displayed as one item with no spaces between them.
- **flushright** moves the data in the column to the right leaving no spaces between columns. This is useful when two columns are displayed as one item with no spaces between them.

Numeric formatting choices include any 4GL numeric format for formatting numeric data. Refer to the 4GL Reference Manual for more information on numeric formats. Examples appear below.

- \$\$\$,\$\$\$.\$\$
- ###,###.##
- <<,<<<

Runtime Report Values

Runtime report values are special column types that display system defined values at the time the report is printed. You can specify the location of these columns on your report when you add the column.

To add a runtime report value:

1. **Move the cursor to where you want to add the column.**
2. **Press "[" to display the Select Column Type window.**
3. **Select Runtime Report Values. The available columns or values you can add include the following:**

- **Page Number** prints the page number on the report. It only prints the number; if you want "Page 2" printed you must add "Page" before the special column in the Fitrix Report Editor.
- **Time of Printing** prints the time the report was printed.
- **Date of Printing** prints the day the report was printed according to the DBDATE definitions variable.
- **Report Title** displays the name of the report file on the report.
- **Selection Set Name** prints the name of the chosen selection set.
- **Report User Name** prints the login name of the report author.
- **Row Count** counts the total number of rows within the report. This does not count rows broken down by subgroups.
- **Row Number** assigns a number to the current row of the report.

Report Blocks

The Fitrix Report Writer places block tags on the left edge of the Report Editor to identify the roles that particular blocks of text play in the report. The block tags are generated automatically based on the report definitions. The blocks can be modified using the Report Editor. You can insert or delete rows and add new report blocks.

```

[ESC] Save [DEL] Cancel "[ New Column "{" New Group
Report Name: Customer Orders
===== (12,18) ===== (Zoom) =
      10      20      30      40      50      60      70
.....:.....:.....:.....:.....:.....:.....:.....:
HD[date ] [title ] Pa
=====
B1
B1Cust No: [ ]
5B2
B2 Order No: [ ]
B2 Order Date Item No Description Price Qty Extension
B2 [ ] [ ] [ ] [ ] [ ] [ ] [ ]
10A2 Subtotals for [ ]
A2 [ ]
A1Subtotals for [ ]
A1 [ ]
LR
15LRGrand Totals
LR [ ]
:
    
```

If the columns in the report extend beyond the width of the report format, the columns are automatically stacked rather than displayed across the report. The following report shows stacked columns.

```

[ESC] Save [DEL] Cancel "[ New Column "{" New Group
Report Name: Customer Information Detail
===== (1,1) ===== (Zoom) =
      10      20      30      40      50      60      70
.....:.....:.....:.....:.....:.....:.....:
HD[date ] [title ] Pa
=====
HD
5 Cust No [ ]
Company Name [ ]
First Name [ ]
Last Name [ ]
Address Line #1 [ ]
Address Line #2 [ ]
10 City [ ]
St [ ]
Zip [ ]
Phone [ ]
Credit Code [ ]
15 Sales Person Code [ ]
Card Number [ ]
Expiration Date [ ]
    
```


Type of Report Blocks

- **FR** is the First Row Block Tag. This report block will print once at the very beginning of the report.
- **HD** is the Header Block Tag. The header information is printed once at the top of every page of the report.
- **B1, B2...** are Before Block Tags. The Before block tags identify blocks of text which are printed on the report before the detailed data rows. Examples of before blocks of data are Grouped information. In the example above there is a B1 and a B2.
- **B1, or Before 1**, indicates the block of information will print before the group by Company Name information. **B2, or Before 2**, identifies a subgrouping of information based upon a group definition by the column Order No. Before groupings are numbered according to the sequence specified in the Group and Sort definition. These are group headings, which will print at the beginning of each group break.
- **Blank Tag** The row directly following the Before Tag which has no identifier identifies where the report detail data rows will print.
- **A2, A1...** are After Block Tags. The After block tag identifies information that is printed after the report detail data rows. Subtotal groupings are numbered according to the corresponding Before block tag. The sequencing is relative to the sequence defined in the Group and Sort definition.
- **LR** is the Last Row Block Tag. It is printed on the final row of the report. It contains column information such as Grand Totals.
- **TR is the Trailer Block Tag.** It is printed at the bottom of every page and can be defined to contain footer information.

Note

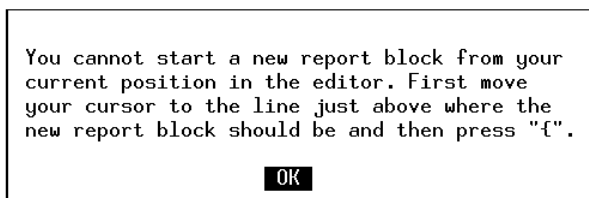
When defining a math formula, the columns that can be used in the formula are displayed in a window below the formula entry field. The columns that are available are numeric columns located in *that* Report Block.

Adding a New Report Block

The steps to adding a Report Block to your report are dependent upon the type of report block you are adding. In the Report Editor you can add the following types of report blocks:

- B# or Before Group
- A# or After Group
- LR or Last Group
- TR or Trailer

The type of block you can add is sensitive to the line the cursor is on in relation to the existing report blocks and predefined report groups. If you are not on a row to which a report block can be added, you will get the following message:



Adding a Before or After Group Report Block

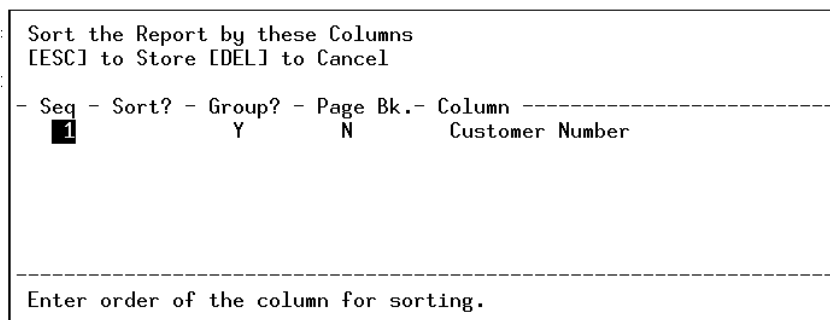
Once you make changes to your report in the Report Editor, the automatic column formatter no longer updates the report template. Any additional data groupings defined in the Group and Sort will not appear on the report; adding new groups must be done through the Report Editor.

The steps for adding a new B or A group are as follows:

1. **Select Group and Sort from the Report ring menu option.**
2. **Enter and save the column grouping information.**

When a new report group is defined there are two types of report blocks that can be added to the report, a Before Group or an After Group. The number associated with the B or A block tag will be dependent upon the grouping sequence defined for the column.

Since Before and After Blocks refer to a specific Group, the reference is Group specific. The following template and sequence definition displays one block:



The Customer Number field defines the only grouping, and can have items placed before (B1) or after (A1) the area where the data representing one customer number is presented.

The following report template has more than one group defined:

```

Sort the Report by these Columns
[ESC] to Store [DEL] to Cancel

- Seq - Sort? - Group? - Page Bk.- Column -----
  1   A     Y     N     Customer Number
  2   A     Y     N     Order Number
  3   A     N     N     Stock Number

-----
Page break on this group? (B-before, A-after,N(or null)-none
    
```

Even though there are three fields in the Sort consideration, only two are specified to have data grouped. This instructs the Report Writer to create blocks for each, B1, B2, A1, and A2:

```

[ESC] Save [DEL] Cancel "[" New Column "{" New Group
Report Name: Customer Orders
------(1,1)------(Zoom)=
      10      20      30      40      50      60      70
.....:.....:.....:.....:.....:.....:.....:
HD[ date ] [title ] Pa
HD-----
B1
B1 Cust No: [_____]
5B2
B2 Order No: [_____]
B2 Order Date Item No Description Price Qty Extension
B2 -----
B2 [_____] [_____] [_____] [_____] [_____] [_____]
10A2 Subtotals for [_____]
A2 -----
A2 [_____]
A1 Subtotals for [_____]
A1 -----
A1 [_____]
LR
LR
15LR Grand Totals
LR -----
LR [_____]
:
    
```

The Order Number Block (the second group) can have other information placed before it (between B1 and B2) and after it (between A2 and A1).

The Customer Number Block (first group) can have information before (between HD and B1) and after it (between A1 and LR).

In order to add information to a report template:

1. Select Edit the Report from the Report ring menu option.
2. Place your cursor on the row above where you wish to insert the group.
3. Press "{". A window appears with the report blocks that are available to add to the screen.
4. Once the report block is added, you will have a blank row to add further text and column information.

Adding Last Groups and Trailers

The steps for adding LR or TR groups are as follows:

- 1. Place the cursor on the line above where you wish to add the new LR or TR group.**
- 2. Press "{".**
- 3. Confirm you wish to add the report block.**
- 4. A new line is created for the new report block.**
- 5. Add text or columns on that line.**

The Fitrix Report Runner

The Fitrix Report Runner allows you to print, from a LINUX prompt, reports created with the Fitrix Report Writer. This shortcut eliminates the need to execute the Fitrix Report Writer program for the sole purpose of printing, thus saving time. By using the Fitrix Report Runner, the first request for input is at the **Choose a (print) Destination** window.

In order to run reports with the Fitrix Report Runner, you must know the name of the report and the name of the database. If you are going to print the report with a selection set, you must know the name of that selection set.

To print a report using the Fitrix Report Runner the syntax is:

```
fg.runner -d database -r report_name -s selection_set_name
```

To print a report using the "student1" database called "Orders" with the selection set "Unpaid Orders" you would type:

```
fg.runner -d student1 -r "Customer Orders" -s "footballs"
```

Note

To specify a report or selection set that has more than one word, you must put quotes around the report or selection set's name.

Once you enter the `fg.runner` command, the Choose a Destination menu appears

```
[ENT] Select  [ESC] Exit
=====
  Choose a Destination
=====
Display to Screen
Output to Printer
Write to File
Transmit FAX
Archive to Database
-----
Export Data >>
Reporting Options >>
Quit
```

From this window, the Reporting Options selection lets you further define your reporting options.

If you make an error typing the name of the database, report, or selection set, the Fitrix Report Runner will display an error message notifying you which part of the statement it was not able to understand.

Section Summary

- The Fitrix Report Editor allows you to make modifications directly to the report template displayed on the screen. Once you edit your report in the Fitrix Report Editor, the Fitrix Report Writer will not automatically format the report template.
- Column detail information is accessed by editing within the Report Editor, allowing you to change the column format, hide columns, and perform math calculations on column values.
- Report Blocks are tags located along the left edge of the Report Editor screen which identify the role of a group of lines within a report.
- New Report Groups can be added within the Report Editor only after they have been defined in the Group and Sort window.
- The Fitrix Report Runner allows you to print a report anytime, anywhere, from a LINUX prompt, if you know the names of the database, report, and optional Selection Set.

Lab Exercise 3a

Objective: To add columns to the report from the Report Editor.

Adding Columns in the Report Editor

1. Load the Customer Orders Summary report.
2. From the Report pull-down menu, select Edit the Report.
3. Using the edit keys, add the following columns below the Customer Number column:

Company Name
 Address
 City, State, Zip
 Contact Name (first and last)
 Phone Number

4. Add them to report Block 1 and format the fields to "pushleft."

```
[ESC] Save [DEL] Cancel "[ " New Column "{ " New Group
Report Name: Cusotmer Order Summary
===== (1,1) ===== (Zoom) =
      10      20      30      40      50      60      70
.....:.....:.....:.....:.....:.....:.....:.....:
HD]date ] [title ] Pa
HD
B1
B1Cust No: [_____]
5B1Company Name: [_____]
B1 Address: [_____]
B1 : [_____]
B1 : [_____] , [ ] [_____]
B1 Contact: [_____] [_____] Phone: [_____]
10B2
B2 Order No: [_____]
B2 Order Date Item No Description Price Qty Extension
B2
[_____] [_____] [_____] [_____] [_____] [_____]
15A2 Subtotals for [_____]
A2 [_____]
A1Subtotals for [_____]
-----
```

5. Display the report to the screen.

```

Pager: Next Prev Right Left Top Bottom Scroll Up-scroll Quit
View the next page.
=====
08/14/2010                Cusotmer Order Summary                Page:
=====
Cust No: 101
Company Name: All Sports Supplies
Address: 213 Erstwild Court
      :
      : Sunnyvale      , CA 94086
Contact: Ludwig Pauli                Phone: 408-789-8075

Order No: 1002
Order Date Item No Description          Price    Qty    Extension
-----
06/01/2009     3 baseball bat      $240.00    1    $240.00
06/01/2009     4 football          $960.00    1    $960.00
Subtotals for 1002                                -----
                                                    1200.00
Subtotals for 101
Cusotmer Order Summary: columns 1 to 76 lines 1 to 18 of 330

```


Lab Exercise 3b

Objective: To use math columns to calculate the order sales tax and to create a new column called Order Total.

Math Formulas

1. Load the Customer Orders Summary report and, from the Report pull-down menu, select Edit the Report.
2. Insert three rows to the bottom of Report Block A2.
3. On the line directly below the orders subtotal amount, insert a row of "----" to create a line across the report.

Refer to the report image on the next page.

4. On the row below the line, add the text "Shipping Charge" and add the database column Shipping Charge to the report.
5. Next to the Shipping Charge column add the text "Sales Tax" and press "[" to add a new column to the report.
6. Select Math Formula from the column type window and enter a math formula to calculate 8.2% sales tax on the order subtotal.

The formula to make this calculation is $(.082 * A1)$.

Note

On your report, the variable for the subtotal of items.total_price may be different than A1. If it is, use the variable displayed in the variables window that identifies SUBT_AF_2_items.total_price.

-
7. Enter a numeric format of \$\$\$.\$\$ for this math column and reduce the field size to six characters using the [CTRL]-[x] key combination.
 8. To the right of the sales tax column, add the text "Order Total" and press "[" to add a new column to the report.
 9. Select Math Formula and enter a math formula to add the Orders Subtotal, Shipping Charge, and Sales Tax. Reduce the field size to nine positions.

Your report should appear as illustrated below:

```

[ESC] Save [DEL] Cancel "[" New Column "{" New Group
Report Name: Cusotmer Order Summary
===== (18,3) =====
          10      20      30      40      50      60      70
.....:.....:.....:.....:.....:.....:.....:.....
B1Cust No: [_____]
5B1Company Name: [_____]
B1   Address: [_____]
B1   : [_____]
B1   : [_____] , [ ] [___]
B1   Contact: [_____] [_____] Phone: [_____]
10B2
B2  Order No: [_____]
B2  Order Date Item No Description      Price Qty  Extension
B2  -----
B2  [_____] [____] [_____] [_____] [____] [_____]
15A2 Subtotals for [_____]
A2  -----
A2  Shipping Charge:[_____] Sales Tax:[_____] Order Total:[_____]
A2  -----
20A1 Subtotals for [_____]
A1  -----
    
```

Lab Exercise 3c

Objective: To add a new report group to the Customer Orders Summary report without using the Fitrix Report Writer to automatically format the report template.

Define Groups in the Fitrix Report Editor

1. Load the Customer Orders Summary report.
2. From the Report pull-down menu select Group and Sort.

Before you can add a new group in the Report Editor, you must first define it in the Group and Sort window from the Report pull-down menu.
3. Change Stock Number to Group as well as Sort.
4. Select Edit the Report.
5. Move to the last row of the Report Block B2 and press "{" to add a new group.
6. Add two columns in the new B3 report block to display the Stock # and Manufacturer's Code (located in the "Items on Order Lline" Data File).
7. Print the Customer Orders Summary report.

```

Pager: Next Prev Right Left Top Bottom Scroll Up-scroll Quit
View the next page.
=====
08/14/2010                Cusotmer Order Summary                Page:
=====
Cust No: 101
Company Name: All Sports Supplies
Address: 213 Erstwild Court
      :
      : Sunnyvale      , CA 94086
Contact: Ludwig Pauli                Phone: 408-789-8075

Order No: 1002
Order Date Item No Description          Price Qty Extension
-----
Stock #:      3 Manufacturer's Code: HSK
06/01/2009    3 baseball bat          $240.00 1    $240.00
Stock #:      4 Manufacturer's Code: HSK
06/01/2009    4 football          $960.00 1    $960.00
Subtotals for 1002
Cusotmer Order Summary: columns 1 to 76 lines 1 to 18 of 462

```

Lab Exercise 3d

Objective: To print the Customer Orders Summary report from a LINUX prompt with the Unpaid Orders selection set.

Report Runner

1. From a LINUX prompt, enter the command to print the Customer Orders Summary report using the Unpaid Orders selection set.
2. Display the report to the screen.
3. Without exiting back to a LINUX prompt, reprint the report with no selection set.
4. From the Reporting Options choose Data Selection.
5. Modify an existing selection set to contain no selection criteria and save it upon exiting as "No Selection Set."
6. Display the new report to the screen.

Chapter 4

Managing the Data Dictionary

Major topics in this chapter include the following:

- Naming Tables and Columns
- Adding Tables to a Data Group
- Table Relationships

Descriptive Table and Column Names

You have the ability to give familiar names to the tables you create. Tables and columns created in dbaccess are assigned default names which are often not descriptive enough for your users.

The Table/Columns option on the Options pull-down menu allows you to give tables and columns more descriptive or familiar names. Here, a table called "trcredit" is being assigned a familiar name. Each of its columns may also be assigned more descriptive names:

Table Name		
[TAB] to Column Window [ESC] to Save [DEL] to Cancel [CTRL]-[w] Help		
Table: trcredit	Name: Credit Table	

Enter a familiar name for the table.		

- Column -----	Column Name -----	Column Label -----
credit_code	Credit Code	
credit_desc	Credit Description	
credit_amt	Credit Amount	

"Column Label" is the heading for the column as it appears on a report. The Column Name will appear as the default column heading if the Column Label field is blank. Column labels can be changed for individual reports within the Report Editor.

Data Groups

The Data Groups option on the Options pull-down menu allows you to put tables into a data group, or to create new data groups. Both Table Description and Table Name are Zoom-capable fields. Here is an example of the "trcredit" table being added to the "Training Data Set:"

```

Update: [ENT] to Store, [ESC] to Cancel
Enter changes into form
                                     (Zoom)
-----
                        Define Data Groups
-----
Group Name: Training Data Set

- Table Description ----- Table Name -----
Sales Person Information   trsalesrep
Sales Orders               trorders
Manufacturer Definitions   trmanufact
Sales Person Information   trsalesrep
State Definitions          trstate
Items on Order Lines       tritems
Items in Inventory         trstock
Credit Table               trcredit
-----
Enter the name of the database table.

```

Table Relationships

After a data group has been added, you should define how the new table relates to the other tables in the database, so that the Fitrix Report Writer knows how to incorporate this new table, and what the join column(s) are. The Table Relationships option on the Options pull down menu allows you to establish this relationship to the other tables in your database. Here is an example of how the trcustomer table relates to the trorders table:

```

Update: [ENT] to Store, [ESC] to Cancel
Enter changes into form
                                     (Zoom)
----- Define the Table Relationship -----
From Table: trcustomer
to Table: trorders

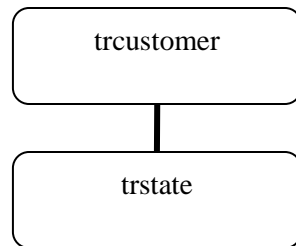
Relationship Type: 2
-----
Enter the main table for the relationship.

-----
-- From Column ----- To Column -----
customer_num          customer_num
-----

```

One-to-One Relationships

In a relational database, there are various ways that tables relate to each other. In the one-to-one relationship illustrated below, each order can have only one payment method attached to it. Therefore, a one-to-one relationship exists between the customer table and the state table.



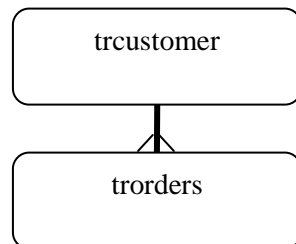
The relationship is specified as 1-1 in the Relationship Type field of the Define the Table Relationship window. The join column(s) are also specified in the Define the Table Relationship window.

One-to-Many Relationships

Another way that tables relate to each other is the one-to-many relationship.

One customer can appear on several orders. Therefore, a one-to-many relationship exists between the trcustomer table and the trorders table.

This relationship is specified as "1-many" in the Relationship Type field of the Define the Table Relationship window. The filled-in Define the Table Relationship window would appear as follows:



Notice the join column(s) are specified in the bottom portion of the Define the Table Relationship window.

Outer Joins

The purpose of defining table relationships is to present the data on a report correctly. The Fitrix Report Writer will take into account how tables relate to each other at the time the report is printed. If table relationships are improperly defined, the data on the report will not be correct.

For "1-1" and "1-many" relationships, data will appear on the report only if a value exists in the join column of the table listed in the to Table field in the Define the Table Relationships window. If the join column's value is null, nothing will appear on the report.

Outer joins placed on a table relationship allow you to see all data on a report regardless of whether values in the join columns of the table listed in the to Table field are null or not. You specify outer joins with "1-0/1" or "1-0/many" in the Relationship Type field of the Define the Table Relationships window.

Section Summary

- Each table added to a database can be given a descriptive name. A descriptive name is familiar to the user of the table. Likewise, each column can have a familiar name.
- Each new table should be added to a data group.
- Each table's relationship to other tables in the database must be known to the Firtix Report Writer. Types of relationships are "1-1," "1- many," "1-0/1," and "1-0/many."
- Outer join relationships display all data on a report regardless of whether there is a match in the join columns.

Lab Exercise 4a

Objective: To introduce a new table to the Report Writer. The table will hold credit code information, and each customer will have an individual credit code. Give this table a familiar name, include it in a data group, and establish its relationship with the customer table.

Adding the Table Through dbaccess

1. Add a table called `credit` to your student {} database.

(If you already have this table, you may proceed to Exercise 4b).

Use dbaccess to add your "credit" table. The "credit" table will hold the following columns:

Column Name	Type
credit_code	char(3)
credit_desc	char(10)
credit_amt	decimal(10,2)

2. Once the table has been created, add the following rows to your "credit" table with ISQL:

credit_code	credit_desc	credit_amt
AAA	Excellent	\$10,000
BBB	Good	\$5,000
CCC	Fair	\$1,000
DDD	Poor	\$250

3. When you add a table or column to your database, you must update the schema files in \$fg/data. You do this with the following commands.

```
cd $fg/data
sch.sh student{}
```

Give Your Tables Descriptive Names

Next you can give your tables familiar names so that your users do not have to remember technical names.

1. Start the Fitrix Report Writer with:

```
fg.writer -d student{}
```

2. **Select Table/Columns from the Options pull-down menu and pull up your credit table.**
3. **On the window that comes up, assign a familiar name to your credit table, such as Credit Code Information.**
4. **Give each column a descriptive, familiar name, if names have not been assigned yet.**
5. **Press [ESC] to save your work in this window.**

Lab Exercise 4b

Next you will add your new "trcredit" table to the "Training Data Set" data group.

Adding Your New Table to a Data Group

1. **Select Data Groups from the Options pull-down menu. Select the "Training Data Set" data group.**
2. **Under Table Description or Table Name, enter either the familiar name or technical name of your credit table. You may use the Zoom feature to help you.**

Note

If you enter the familiar name, the technical name is looked up for you after you press [ENTER]. If you enter the technical name, the familiar name is looked up for you after you press [ENTER].

3. **Press [ESC] when done.**
4. **Exit the Report Writer.**

Adding a Column to the trcustomer Table

Each customer can have one credit code. Therefore, you will add a column to the trcustomer table to hold a particular customer's credit code. (If your customer table already has this column, you may proceed to "Adding the Table Relationship," below.)

1. **Using dbaccess, add a Credit Code column to the customer table.**

Make sure the Credit Code column is the same name and type as the Credit Code column in your credit table.

2. **In dbaccess, enter some credit codes for some of the customers.**

Be sure the credit code values are the same as the values for credit code in the credit table.

3. **Exit dbaccess.**
4. **Update the schema file in \$fg/data using the sch.sh utility.**

Adding the Table Relationship

Next you will make known to the Firtrix Report Writer how the trcredit table relates to other tables in the student{} database.

1. Start the Fitrix Report Writer:

```
fg.writer -d student{}
```

2. Select Relationships from the Options pull-down menu.

3. Select Add a New Relationship at the Choose a Table Relationship window.

4. Enter "trcustomer" in the From Table field of the Define the Table Relationship window.

5. Enter "trcredit" in the to Table field. Then determine the relationship type that exists between these two tables.

Ask yourself, how many credit codes can one customer have? For example, you can have many orders for one customer. Therefore, you would enter a "2" in the Relationship Type field for the "trcustomer" - "trorders" relationship. "2" is a "1-many" relationship.

6. Tab to the From Column - To Column area, and enter the join information.

Both fields are Zoom capable.

7. Press [ESC] to save your work.

8. Again, select Add a New Relationship at the Choose a Table Relationship window and define the relationship in reverse for the two tables.

9. Press [ESC] to save your work.

Add Credit Information to Your "Customer Phone Numbers" Report

Finally you will see your new table and relationship in action. Add a credit description field to your "Customer Phone Numbers" report.

1. In the Fitrix Report Writer, load the "Customer Phone Numbers" report.
2. Select Edit the Report from the Report pull-down menu.
3. Add a credit description field label to the heading line (HD) and the field to the report in the on every row (blank) block.

The screenshot shows the Fitrix Report Writer interface. At the top, there are menu options: [ESC] Save, [DEL] Cancel, "[" New Column, and "{ " New Group. Below this, the report name is displayed as "Customer Phone Numbers". A status bar at the top right shows "(5,57)=====(Zoom)=". The main area displays a table structure with columns defined by their widths: 10, 20, 30, 40, 50, 60, and 70. The heading line (HD) is defined as follows:

HD	[date]	[title]	[]	[]	[]	[]
HD	Cust No	Company Name	Phone		Credit Description	
HD	[]	[]	[]	[]	[]	[]

The data block starts at line 5 and contains several rows of data, with the first row being a header row for the data block.

4. Save and print your report with no selection set.

Using an Outer Join

If the relationship types you chose were "1-1" or "1-many," you will only see those customers that have credit codes in the database. The customers that do not have credit codes are not on the report.

If you would like to see all customers regardless of the existence of a credit code, change the relationship types to "1-0/1" or "1-0/many." This is how you indicate that there is an outer join relationship between tables.

1. In the Report Writer, select Relationships from the Options pull-down menu.

2. Change the two relationships you entered above to indicate “1/0-1” or “1/0-many” for their respective relationship types.

3. Re-print the report.

What prints now?

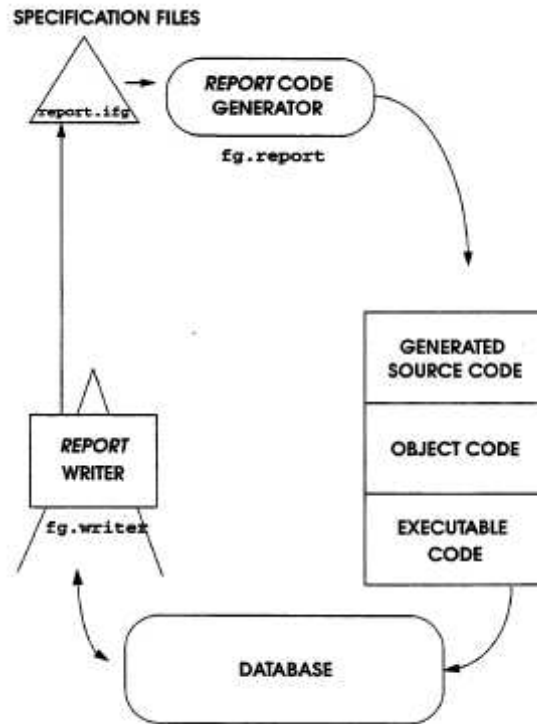
Chapter 5

Fitrix Report Code Generator

Major topics in this chapter include the following:

- Working with the Fitrix Report Code Generator
- Specification Files for Reports
- Report Code
- Compiling and Running Report
- Editing Generated Reports
- Directory Structure

Fitrix Report Code Generator Overview



Sample report.ifg File

```
#####
# Copyright 2000 Fourth Generation Software Solutions
# All rights reserved.
# Use, modification, duplication, and/or distribution of this
# software is limited by the software license agreement.
# Sccsid:  @(#)  ../train.4gm/report1.bak/report.ifg  1.1  Delta: 4/21/94
#####

database student1

output
  top margin      0
  bottom margin   0
  left margin     0
  right margin    80
  page length     66

page header
{
      +C1
<A0      +A1                      Page: >A2
=====

Customer No Order No   Order Date   Description   Item No Mfct   Price
-----
}

on every row
{
[A3      [A6      [A7      [A8      [A9      [A  [AA      [*
}

on last row
{
      Grand Totals
      -----
      [AD
}

attributes
  A0  = today using "MM/DD/YY"
  A1  = constant "Customer Listing"
  A2  = pageno using "<<<<<"
  A3  = trcustomer.customer_num
  A6  = trorders.order_num
  A7  = trorders.order_date using "MM/DD/YY"
  A8  = trstock.description
```

```
AC = sum(tritems.total_price)
AD = sum(tritems.total_price)
A  = trstock.manu_code
A9 = trstock.stock_num
AA = tritems.total_price
C1 = constant "Report Demo 1"

select
  tables = trcustomer, trorders, tritems, trstock
  join   = trstock.manu_code = tritems.manu_code and trstock.stock_num = tritems.stock_num and trorders.customer_num = trcustomer.customer_num and tritems.order_num = trorders.order_num
end

defaults
  progname = report1
  destin   = report.out
end
```

Files Generated by the Fitrix Report Code Generator

The Fitrix Report Code Generator creates the following files:

```
globals.4gl      main.4gl
midlevel.4gl     lowlevel.4gl
report.4gl       Makefile
```

A Makefile is simply a list of files to compile and libraries to link. FourGen uses the LINUX make utility to compile files.

The `report.4gl` takes the WYSIWYG image in the `report.ifg` file and maps it out as follows.

Compiling Programs

The generated 4GL source code must then be compiled into object code. Also, all calls to functions in libraries must be resolved. Fitrix Report programs are compiled with a script called `fg.make`. The `fg.make` script will compile each `.4gl` file and link in library functions that are called within the generated code.

Passing Criteria at Run-Time

Once you have compiled a report program, you can pass criteria to the report on the LINUX command line.

The filter flag passes an SQL "where" clause. For example:

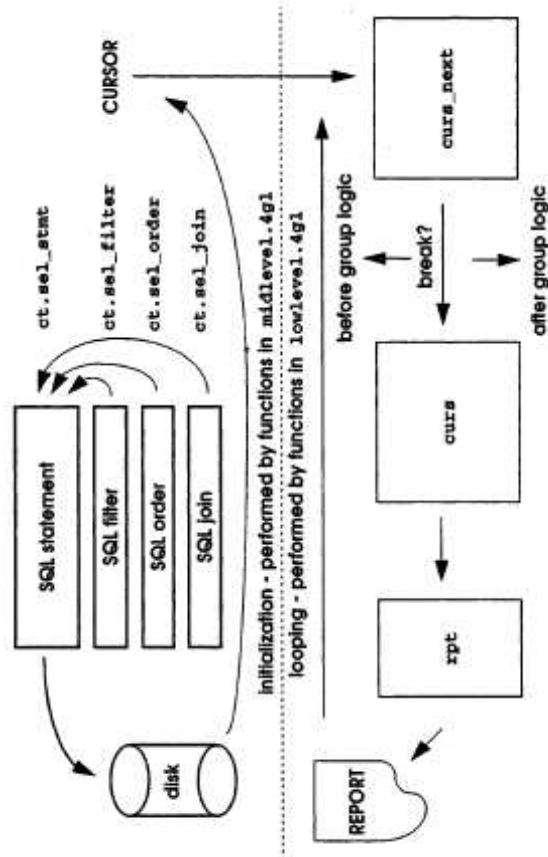
```
fglrun *.42r -d student1 filter "trcustomer = 101"
```

The order flag passes an SQL "order" clause. For example:

```
fglrun *.42r -d student1 order "trorder.customer_num"
```

Most of the other variables in the "defaults" section of a `report.ifg` file can be overridden at run-time. Another often-used variable is `dest.in`. This variable is used to specify the report's destination at run-time (e.g., screen, file, or printer).

Report Data Flow



Report Functions

The following tables describe some of the key functions used by generated report programs.

Initialization Functions:

<code>ct_defaults()</code>	initialize all "ct" global variables with hardcoded default values
<code>ml_defaults()</code>	initialize "ct" variables with values from the <code>report.ifg</code> file
<code>ct_parse()</code>	reads command line input from report, if any
<code>ct_display()</code>	border info for pre-report screen called with seven different flags
<code>ct_display()</code>	Gathering elements message
<code>ml_getcount()</code>	number of records fitting selection criteria; terminates if no rows
<code>ct_display()</code>	Sorting ## Elements message
<code>ml_define_cur()</code>	run select statement with all elements required to load cursor
<code>ct_start_rpt()</code>	destination output initialize
<code>ct_display()</code>	"Processing element # of #" message
<code>ml_fetch()</code>	get first row of cursor into <code> curs_next </code> record
<code>ml_curs_prep()</code>	put data into <code> curs </code> record from <code> curs_next </code>
<code>before_group()</code>	"first row" initialization of report break logic; additional code can be added
<code>ml_curs_null()</code>	clears <code> curs </code> buffer; <code> curs_next </code> record contains first row; <code> curs </code> buffer is null

Loop on Detail Functions

<code>ml_before_break()</code>	check to see if <code> curs = curs_next</code> ; if false then there is a break; break logic is also done by "Report;" custom code can be added here
<code>ml_curs_prep()</code>	<code> curs_next</code> row moves to <code> curs</code> record
<code>ml_fetch()</code>	if not last row, get next row into <code> curs_next</code> record
<code>on_every_row()</code>	moves <code> curs</code> data into report record
<code>ml_after_break()</code>	check to see if group finished
<code>after_group()</code>	last row of table processing
<code>Ml_output()</code>	report line is output to "Report;" "Report" is report name generated
<code>ct_display()</code>	"Processing # of #" message

End of Job Functions

<code>ct_finish()</code>	"Finish Report" message
<code>ct_display()</code>	"Processing Complete" message
<code>clear_screen()</code>	clears the screen display

Report Functions Commonly Modified

There are some functions in `midlevel.4gl` that you will frequently modify. These functions give you control over the SQL statement that is executed to return rows from the disk. You can modify these functions to pass criteria to the SQL statement:

- `ml_filter()`: manages the final "where" criteria that are used in the SQL statement. The `ml_filter()` function manages the `sel_filter` variable, and is typically modified to pass selection criteria to a program. An example of how you can modify `ml_filter()` to put a selection criteria screen into a modified program is included in this section's Lab exercises.
- `ml_order()`: manages the "order" criteria that are used in the final SQL statement. It manages the `sel_order` variable.

- `ml_join()`: manages the "join" criteria that are used in the SQL statement. It manages the `sel_join` variable.

Referencing Fields not Printed in the Report

On occasion, you need to select data to be included on a report, or "order" report output based on a field that does not appear on the report. The optional more line defines the columns that are not included on the report but are needed in the select or order by statements.

If a report needed to be arranged by credit code, salesperson code, or another field that does not print, a modification to the select section of the .ifg file would allow that functionality. As the following example illustrates, each field to be included is defined with its own more statement.

```
select
more = trcustomer.credit_code
more = trcustomer.sales_code
tables = trorders, trcustomer
join = trcustomer.customer_num = trorders.customer_num
order = trorders.customer_num
```

Basic Library Philosophy

A library holds source code that is shared by other programs. Envision several lines of source code that check the form to be printed to, lines per page, carriage control, etc. Every print program will use this source code. Does it make any sense to copy the same lines of source code into every program directory that uses these lines of code? No. Therefore, the lines of source code are placed into a function, and that function is placed into a library. Any program that needs to consider these options simply calls that function in the library.

When placing a function that will be shared by programs into a library, we first ask:

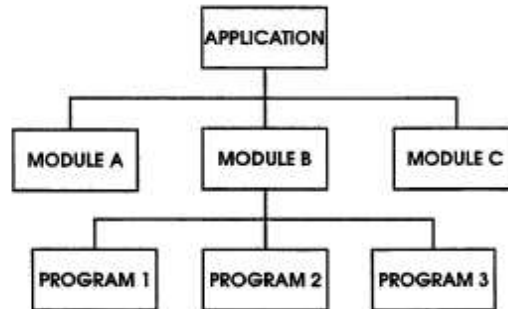
- What types of programs will use this library function?
- Will it be used only by report programs?
- Will it be used by products other than report programs, such as input programs?

Based on what types of programs will use the function, places it into an appropriate library directory.

Libraries exist as library directories. A library directory is simply a .4gs directory that contains .4g1 files with code shared by multiple programs.

Directory Structure

- **Program Directories** contain all the files that make up a program. Program directories are the lowest level of directories.
- **Module Directories** contain program directories. A "module" contains a group of programs that are somehow alike.
- **Application Directories** contain module directories. An "application" contains modules that are alike.



Section Summary

- You have the ability to create 4GL source code on reports that you develop with the *Report Writer*. One advantage of creating code is that you can modify the code; that is, any capabilities that cannot be performed with the Fitrix Report Writer may be coded into 4GL source code.
- Export a report specification file called `report.ifg`. The `report.ifg` file is read by the Fitrix Report Code Generator, and 4GL source code is created based on instructions in the `report.ifg` file.
- Two special files created by the Fitrix Report Code Generator are `report.4gl` and `Makefile`. The first, `report.4gl`, is the WYSIWYG image of the report in the `report.ifg` file mapped out in 4GL language. The `Makefile` contains a list of files to compile and libraries to link.
- The source code is compiled with `fg.make` script. Compiling converts each source code file into object code. The `fg.make` utility also performs linking, which means that calls to library functions in the generated source code are resolved. The script `fg.make` uses the LINUX make utility to perform its tasks.
- You can pass report criteria from the command line.
- Fields used to select or order report output, that are not included on the report, are declared by adding a more statement in the select section of the `report.ifg` file.
- The flow of a generated report is very simple. The first half of the logic handles setting up the main SQL statement and executing it. The second half of the logic handles processing each row returned by the SQL statement.
- Commonly modified functions are `ml_filter`, `ml_order`, and `ml_join`. These functions control the final filter, order, and join criteria, respectively, that are attached to the main SQL statement.
- When creating a directory to hold an exported `report.ifg` file, the same directory structure that is followed with input and edit programs applies to report programs.

Lab Exercise 5a

Objective: You will create 4GL source code for a report you made in the Fitrix Report Writer.

Make a Simple Report

First, you will make a simplified "customer orders" report. It is helpful to work with a simplified report when learning how to modify the code.

1. **Start the Fitrix Report Writer with**

```
fg.writer -d student{}
```
2. **Load the "Orders" report.**
3. **Under the Report pull-down menu select Group and Sort.**
4. **Group the "Orders" report by Customer Number.**
5. **Save as "Orders Listing."**
6. **Quit out of the Report Writer.**

Export the Report from the Report Writer

1. **If \$HOME/labs/aw.4gm does not exist, from your HOME directory, make a directory called "labs" with an aw.4gm directory under labs.**
2. **cd to \$HOME/labs/aw.4gm**
3. **Make a directory to contain the report program.**

```
mkdir o_orders.4gs
```
4. **cd to your newly created directory.**

```
cd o_orders.4gs
```
5. **Start the Fitrix Report Writer**
6. **Load your "Orders Listing" report.**
7. **Under the File pull-down menu, select Import/Export and then select Export Report.**

You import and export reports for two purposes: To move reports from one database to another and to generate code from a report stored in the Fitrix Report Writer. In this exercise, you will generate code from the "Orders Listing" report.

8. Under "Export File Name," change `report.txt` to `report.ifg`.
9. Press [ESC] to export the report.
10. Exit the Fitrix Report Writer when finished.

Generate Code on the Report

1. List the files in your report directory. Notice the `report.ifg` file.

The `report.ifg` file alone will not make up an entire report; it is simply a specification file. You must create and compile source code to run your program. The Fitrix Report Code Generator will do this for you.

2. Start the Fitrix Report Code Generator with the following command:

```
fg.report -d student{ }
```

The Fitrix Report Code Generator reads the instructions in the `report.ifg` file and creates 4GL source code. This source code is the code that will drive your report.

3. When the Fitrix Report Code Generator is done, list the files in the directory.

The newly-created `.4gl` files contain 4GL source code.

Compile the Code

After generating code, you must convert this code to object code, and then link in the libraries. These processes are called "compiling" and "linking," and you use the following steps:

1. From your report directory, compile your code by typing

```
fg.make
```

2. When compiling and linking is done, list the files.

All the object code exists in `.42m` files. The executable file has a `.42r` extension.

Run the Program

1. Run the program with the following command:

```
fglrun *.42r -d student{ }
```

After the report runs, the results are placed in a file called `report.out`.

2. Use `vi` to view the `report.out`.

This is your final report.

Lab Exercise 5b

Objective: To add a new field using the Report Writer and to generate code to see the result of adding a field.

Add the New Field with the Fitrix Report Writer

1. Start the Fitrix Report Writer.
2. Load your "Orders Listing" report.
3. Under the Report pull-down menu, select Edit the Report.
4. Add the "Company Name" column to the right of the Customer Number column.
5. Under the File pull-down menu, select Save the Report to save your work.
6. Select Import/Export and then select Export Report.
7. In the Export File Name field, change `report.txt` to `report.ifg`. Overwrite the existing `report.ifg`.
8. Press [ENTER] and [ESC] to export your report with the new field on it.
9. Exit the Fitrix Report Writer when finished.

Generate Code on the Report

1. Use `vi` to edit the `report.ifg` file and see if you can spot the changes.
2. Start the Fitrix Report Code Generator.
3. Before the Fitrix Report Code Generator is finished, the Overwrite menu appears. When it appears, answer choice "3" for each file.

If the Fitrix Report Code Generator finds a `.4gl` file that already exists and its newly generated `.4gl` file is different from the existing `.4gl` file, it will ask you what to do with the existing `.4gl` file.

Generally you will answer choice "1" to overwrite the file. However, for this exercise you will be viewing the differences between newly generated `.4gl` files and old `.4gl` files. Choice "3" gives the existing `.4gl` file an ".old" extension and generates a new `.4gl` file. Your existing file is preserved in the `.old` file.

Compile the Code

The newly-generated `.4gl` files now must be compiled and linked.

Use the following command to compile and link your code:

fg.make

Run the Program

1. **Run the program with the following commands:**

```
fglrun *.42r -d student{}
```

The report will run and the results will be placed in a file called `report.out`.

2. **Use vi to view the `report.out`.**

This is your final report. Is your new field there?

3. **Quit out of `report.out`.**

Diff the Files

Now use the LINUX `diff` utility to see how adding a new field affected the generated code.

1. **List the files in the directory.**
2. **For each `.4g1` file in your directory, run the `diff` command against the `.old` file.**

The `diff` command syntax is:

```
diff [first_file] [second_file]
```

For example, to see the differences between `report.old` and `report.4g1`, you would use `diff` as follows:

```
diff report.old report.4g1
```

What is the difference between the old `.4g1` file and the newly generated `.4g1` file?

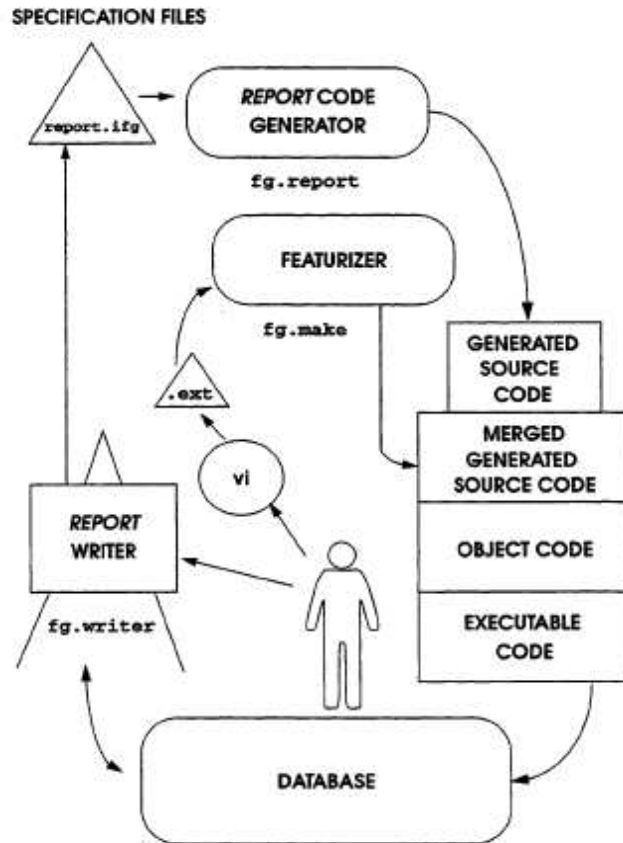
Chapter 6

Using the Featurizer

Major topics included in this chapter are the following:

- Using Blocks to Modify Generated Code
- Block Command Placement
- Extension Files
- Feature Sets

Report Code Generator Overview



A Function with Blocks

Every line of code created with the Fitrix Report Code Generator is within a block. Here is an example of blocks within a generated function:

```
#####
function ml_defaults()
#####
#

    #_define_var - Define local variables

    #_def_prcname - Default "prcname" value
    let ct.prcname = "Orders Listng"

    #_def_destin - Default "destin" value
    let ct.destin = "report.out"

end function
# ml_defaults()
```

The block id is the part that follows #_. There are 2 blocks in the above example, one block id is 'def_prcname' and the other is 'def_destin'. You use these block ids in block commands.

Block Command Examples

Block command syntax:

```
Block_command [function_name] [block_ID]
```

Example 1:

```
after block ml_defaults def_destin
```

This block command places code after the `def_destin` block in the `ml_defaults` function.

Example 2:

```
before block ml_defaults def_destin
```

This block command places code before the `def_destin` block in the `ml_defaults` function.

Example 3:

```
replace block ml_defaults def_destin
```

This block command replaces the entire `def_destin` block in the `ml_defaults` function.

Where Do Block Commands Go?

Block commands for reports go into `.ext` files, which are specification files where you place your block commands and the custom logic you wish to add. The Featurizer then reads the `.ext` files and places your logic into the source code. This allows you to make further modifications to your `report.ifg` file, such as added fields, etc., and regenerate without having to manually re-insert your custom logic.

The `start file` Command

Any block command placed in a `.ext` file must be under a start file command. The start file command tells the Featurizer to which file to apply the block command.

Here is the syntax of the start file command:

```
start file "file_name"
```

For example, a block command to manipulate a block in `midlevel.4gl` would be preceded by the following line:

```
start file "midlevel.4gl"
```

Here is an example of start file with a block command under it. Note that quotes are required around a file name.

```
start file "midlevel.4gl"
  after block ml_defaults def_destin
    for n = 1 to num_args()
      if arg_val(n) = "-e"
        then
          let check_post = "CHECK"
        end if
      if arg_val(n) = "-P"
        then
          let check_post = "POST"
        end if
      end for ;
    end for ;
```

You can have multiple start file commands in an `.ext` file. Here is an `.ext` file with more than one start file command:

```
start file "globals.4gl"
  before block NUL define
    check-post char(5) ;

start file "midlevel.4gl"
  after block ml_defaults def_destin
    for n = 1 to num_args()
      if arg_val(n) = "-e"
        then
          let check_post = "CHECK"
```

```
end if
if arg_val(n) = "-p"
then
    let check_post = "POST"
end if
end for ;
```

The logic that you see here parses the LINUX command line when the report is run and accepts additional arguments. These arguments then set a variable. .

Note

There are some reserved words for blocks. Since there are no functions in `globals.4gl`, the block command uses the reserved word "NUL" in place of the function name.

Extension Files

The logic within blocks may be specific to one feature. You may set up one `.ext` file for each feature that you have. This allows you to selectively plug and unplug features from the source code of a program.

For example, a feature that parses the command line for additional flags could go into a file named:

```
parse.ext
```

A feature that could put a selection criteria screen into a report could go into a file called:

```
select.ext
```

You can have as many features as you want.

Feature Set Files

Feature set (`base.set`) files allow you to tell the Featurizer which features to plug and unplug. The names of the features are simply the prefixes of the `.ext` files.

Here is an example of an entire `.set` file:

```
parse - looks for added command line input
select - pulls up selection criteria screen
```

This `.set` file instructs the Featurizer to insert the custom logic found in the `parse.ext` file and the `select.ext` file. The Featurizer reads this `.set` file and plugs in the `parse` and `select` features.

The Featurizer only reads the first word on a line in a `.set` file. Anything following the first letter space on a line can be descriptive comments, as in the above example.

Plugging in Features with a .set File

The Featurizer will, by default, find and read a .set file called base.set.

Custom Block tags in report.4gl

The block ids in the report.4gl source file are numbered according to the block in which they occur. Below is an example of this.

```
before group of rpt.customer_num
    #_b_customer_num - Before group customer_num.

    #_need_before_group_customer_num
    need 4 lines

    #_B_customer_num_1 - Print statement
    print

    #_B_customer_num_2 - Print statement
    call put_vararg(rpt.customer_num using usg.customer_num)
    call put_vararg(rpt.company[1,20])
    let line_image = imageManager_getLine(3)
    print line_image clipped

    #_B_customer_num_3 - Print statement
    let line_image = imageManager_getLine(4)
    print line_image clipped

    #_B_customer_num_4 - Print statement
    let line_image = imageManager_getLine(5)
    print line_image clipped

#_end - End of report block.
```

Notice that there are 4 lines printed in before group of customer. If you add code to the third line (B_customer_num_3) and latter add a line to the before group of customer above the third line, the block tags are renumbered (B_customer_num_1, B_customer_num_2, B_customer_num_3, B_customer_num_4, B_customer_num_5). Your code would be added to the new line 3 instead of the old line 3. To get around this you can add your own custom block tags directly to the report.ifg as illustrated below.

```
before group of trorders.customer_num
{

#_Customer_Number
Cust No: [A3 ] Company Name: [A4 ]

#_Detail_heading
Order Date Item No Description Price Qty Extension
#_Heading_lines
-----
```

}

The above custom block tags result in the following code in report.4gl:

```

before group of rpt.customer_num
  #_b_customer_num - Before group customer_num.

  #_need_before_group_customer_num
  need 4 lines

  #_B_customer_num_1 - Print statement
  print

  #_Customer_Number
  call put_vararg(rpt.customer_num using usg.customer_num)
  call put_vararg(rpt.company[1,20])
  let line_image = imageManager_getLine(3)
  print line_image clipped

  #_Detail_heading
  let line_image = imageManager_getLine(4)
  print line_image clipped

  #_Heading_lines
  let line_image = imageManager_getLine(5)
  print line_image clipped

#_end - End of report block.

```

To add code to the third line you would use the block id of 'Detail_heading' which does not change when you add lines above it. When you need to add code to report.4gl, using custom block tags can save you time.

Section Summary

- The Featurizer handles block merging. Use a block when making a modification to report code.
- Blocks can be manipulated with block commands. Block commands go into .ext files.
- The start file command applies blocks to a specified file.
- Extension (*.ext) files are selectively merged into code.
- You specify what .ext files merge into code via a .set file.
- Feature set (base.set) files allow you to specify what .ext files get merged into code.
- Custom block tags in report.ifg can save you time when you add code to report.4gl.

Lab Exercise 6a

Objective: To merge your own custom code into the generated code with the Featurizer. The code that you will merge will simply display a message to the user before the program runs.

Determine Where to Put Your Code

The one of the first things a report program does is display a "Report Status" window. This Report Status window is updated as the program runs.

1. **move to your labs/aw.4gm/o_orders.4gs directory.**
2. **Use vi to open main.4gl and find the block tag #_after_init.**

Notice that this block is after the Report Status window is displayed.

3. **Quit out of main.4gl**

Use the Featurizer to Place Custom Logic into Code

Next use the Featurizer to place your message after the Report Status window is displayed.

1. **Create a message.ext file.**

The prefix to the .ext file should be indicative of the "feature" you are incorporating

Next you will use the Featurizer to place custom logic into code.

2. **Use vi to open your message.ext file and insert a start file command at the top.**

When manipulating a block, you must tell the Featurizer where the block and its function are. You do this by inserting a start file command in your .ext file. All block commands placed under that `start file "file_name"` command are applied to that file. The syntax for the start file command is:

```
start file "file_name"
```

At the top of your newly created .ext file, add in the `start file "file_name"` command, specifying the .4gl file which contains the block you are going to manipulate. This .4gl file must be in double quotes as above.

3. **Under start file, place the block command that will place your custom logic into code.**

Any of the following block commands will place code differently, but the result in the final compiled program could be the same:

- replace block
- after block
- before block

Unless you need to remove code in a block it is best to use the after block or before block commands if you can. Using the before or after block commands improves the maintainability of your programs.

Block command syntax is:

```
Block_command function_name block_id
```

The function name is main and the block id is after_init. Be sure your block command is under your start file command.

4. Insert the 4GL logic to display your custom message.

Now add the actual custom 4gl logic that will be placed into the code. In your .ext file under the block command, add the following code.:

```
IF gn_yesno("Warning","Be sure orders are posted first. Continue?","N") = "N"
THEN
    CALL exit_program(0)
END IF
```

5. End the block with a semi-colon (“;”), and save and exit of the message.ext file.

You can have multiple block commands in an extension file. Block commands are delimited with a semi-colon.

Merging with the .set File

Next you will place your modification into code.

1. Create a .set file with the exact name base.set.
2. Use vi to create the base.set file.
3. On the first line, add the prefix to the .ext file you created.

Example base.set file contents:

```
message
```

Merge and Compile with fg.make

1. Compile and link the source code with the following command again:

```
fg.make
```

When you type fg.make at the LINUX prompt, you may get a message such as:

```
Compiling source files.
main.4gl: failed.
ERROR: /usr/ /Make/program.sh:
Unable to compile these 4gl files:
main.4gl
while compiling
```

This indicates that a syntax error has occurred. You can examine the `main.err` file for the syntax error and then correct the problem in `message.ext`

2. Use **vi** to open `main.4gl`.
3. Find the message you added and inspect the change.

The Featurizer should have incorporated your block command into the code.

Run the Program

- Run the program.

Does your message appear? Notice the default answer of “N” exits the program.

Lab Exercise 6b

Objective: To create and incorporate a selection criteria screen into your report so that the user may limit which customers appear on the report. You will create the selection `custsel.per` file, code a function that handles the selection criteria screen, and place the function call into code with the Featurizer.

Adding in Report Selection Criteria

Many reports require the end user to enter selection criteria prior to running a report. Using the VDT Application Code Generator and VDT Form Designer, you can use the following extension file to automatically hook in these prompts. Follow the general steps outlined below to link in your query screens:

1. Create a query screen in the VDT Form Designer.
2. Run the VDT Application Code Generator.
3. Create an image (`report.ifg`) file for your report.
4. Run the Fitrix Report Code Generator.
5. Create the extension file (shown below) and call it in the `base.set` file.
6. Run `fg.make` to build the report program.

Create the Selection Criteria Form

1. Use the VDT Form Designer to create the following form:

```
Form Editor: [ESC] or [DEL] Command Line          [CTRL]-[w] Help
Update data entry image
===== (student1) ===== (custsel) ===== (Zoom) ===== (1,1) =====

Customer Number: [          ]
Company Name: [          ]
```

Call the form `custsel.per`. Use the screen type of query. Use the `trucustomer` for the main table. Use the `customer_num` and `company` for the fields.

2. Run the VDT Application Code Generator.

```
fg.screen -d student{ }
```

3. Run the Fitrix Report Code Generator.

This step regenerates a report type Makefile

Add the Selection Criteria Using an Extension File

1. Add the following code to an extension file. Call the extension file `selcrit.ext`.

```
start file "Makefile"

libraries
    $(fg)/lib/scr.a
;

start file "main.4gl"

switchbox_items
    custsel S_custsel
;

socket_items
    query
;

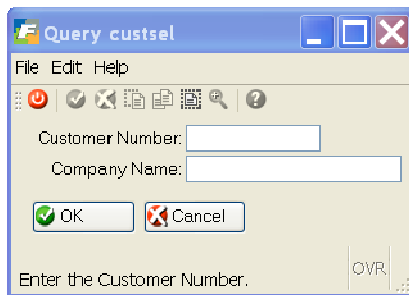
start file "midlevel.4gl"

function_define ml_filter
```

```
n smallint,  
m smallint  
;  
  
after block ml_filter sel_filter  
  call socketManager("custsel", "query", "default")  
  let n = fgStack_pop()  
  if n = 0  
  then  
    let int_flag = true  
  else  
    for m = 1 to n  
      let sel_filter = sel_filter clipped, " and ", fgStack_pop()  
    end for  
  end if  
;  
;
```

2. Add selcrit.ext to the base.set file.
3. Merge and compile the code using fg.make.
4. Run the program.

After the initial warning message, should you choose to continue, the following selection criteria screen appears.



Chapter 7

Creating Posting Programs

Major topics in this chapter include the following:

- Types of Posting Programs
- Creating Posting Programs
- Posting Program Techniques

What is a Posting Program?

A posting program is a report program that performs some special processing on each row as it prints on the report. This special processing is in the form of SQL activity.

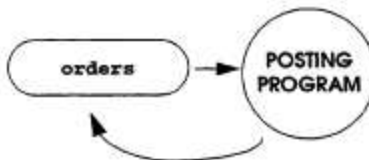
Posting programs can take data from one table and insert into another:



Posting programs can take data from more than one table and insert into more than one table:



The two "pre-posting" tables have a one-to-many relationship between each other, and the two "post-posting" tables also have a one-to-many relationship. Posting programs can also update a table:



Creating the Posting Program

There are two main steps to creating a posting program with Fitrix Report.

1. Create a report with the Fitrix Report Writer that will select all rows that you wish to post.
2. Use the Featurizer to insert SQL statements that act on each row as it is processed.

Creating the Report

Create a report that will pull from a table and display all rows to be posted. The report output itself is ideal because the user can see what data are to be posted.

Keep in mind when creating a report with the Fitrix Report Writer:

- All values to be posted must be brought into the program. Therefore, you design your report so that all values to be posted are brought in by the main SQL statement in `midlevel.4gl`.
- All values to be posted do not have to be on the report itself. You may use the "Hidden Fields" capability in the Fitrix Report Writer to pull data out of columns that do not appear on the final report.
- When posting from two tables, clarify the one-to-many relationship between the two tables. Design your report so there is a group break on the "one" table.
- Filtering criteria: How will you ensure that rows are not reposted? Are there some rows that you never want posted?

Inserting the SQL Statements

After you have created the report with the *Report* Writer and have generated code for it, use the Featurizer to insert SQL statements into the appropriate places in code that will post your rows as they are processed. Typically you will place SQL insert and update statements into `lowlevel.4gl` to post each row as it passes through.

Items to keep in mind:

- If posting from one table to another table, place your SQL statement into the `on_every_row` function.

Here is an example of an SQL insert placed at the end of the `on_every_row` function. It is placed here via a block command in an `.ext` file.

```
#_unit_price - insert "on every row unit_price" code here
  let rpt.unit_price = curs.unit_price
#_total_price - insert "on every row total_price" code here
  let rpt.total_price = curs.total_price
  insert into pst_items values
    (curs.item_num,
     curs.order_num,
     curs.stock_num,
     curs.manu_code,
```

```
        curs.quantity,  
        curs.total_price)  
end function  
# on_every_row()
```

Here, as items table rows pass through the looping phase of the report, they are inserted into a table called `pst_items`.

- If posting from two tables that have a one-to-many relationship, you have two SQL statements: one to post the "one" table's data, and one to post the "many" table's data.
- The "many" table's data are processed by the `on_every_row` function, so you place one SQL statement in `on_every_row` to post.
- The "one" table's data are handled by the `after_group` function or the `a_g_[grouping column]` function, so you place your SQL statement here to post the "one" table's data.

Here is an example that places code into the `a_g_order_num` function. `a_g_order_num` holds code that is executed after the order number changes. Again, code is placed here via a block command in an `.ext` file.

```
#_a_order_num - insert after group code here  
insert into pst_orders values  
    (curs.order_num,  
     curs.order_date,  
     curs.customer_num,  
     curs.ship_instruct,  
     curs.backlog,  
     curs.po_num,  
     curs.ship_date,  
     curs.ship_weight,  
     curs.ship_charge,  
     curs.paid_date,  
     curs.pay_code,  
     curs.ship_method)
```

Here, as orders table rows pass through the looping phase of the report, they are inserted into a table called `pst_orders`.

Note

If you use the `curs` record as you see in the examples, make sure it contains all the variables needed for posting. It typically *will not*. You can add additional variables into your `curs` record by placing hidden fields on the report with the *Report Writer*. When you export the report, a hidden field shows up as a "more =" parameter in the select section of the `report.ifg` file. When you re-generate code, all supporting logic for hidden fields is created for you in code, including logic to place extra variables in the `curs` record.

Using prepare and execute

One key posting technique is to use the 4GL prepare and execute commands in lieu of insert and update.

When you run an SQL insert or update, not only are these statements executed but they are also preprocessed each time they are run.

The preprocessing and execution stages can be separated, so that the insert or update is preprocessed only once, and then executed for each row of the report. You do so with the 4GL prepare and execute commands, which preprocess and run respectively.

Using prepare and execute is ideal for posting programs, which are notoriously slow due to the large number of SQL statements in them. With prepare and execute, you preprocess an SQL statement at the beginning of the program; then you simply execute the preprocessed SQL statement when you need it.

Instead of coding the 4GL insert, you use a prepare and execute as follows:

```
start file "lowlevel.4gl"

function_define b_g_first_row
    sql_str char(200)
;
after block b_g_first_row after_first_row
    let sql_str = "insert into pst_items values (?, ?, ?, ?, ?, ?)"
    prepare prep_stmt from sql_str
;

after block on_every_row after_every_row
    execute prep_stmt using
        curs.item_num,
        curs.order_num,
        curs.stock_num,
        curs.manu_code,
        curs.quantity,
        curs.total_price
;
```

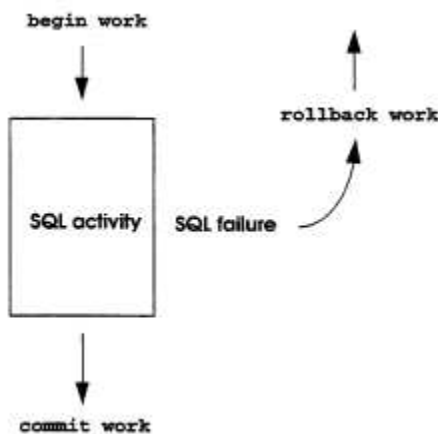
The question marks that you see are placeholders in the preprocessed SQL statement; they are filled with values when the statement is executed. Notice that the cursor preparation code is placed in the `b_g_first_row`. The `b_g_first_row` function is executed once at the beginning of the program. The actual insert is executed at the end `on_every_row` as each row is processed. Executing a prepared statement takes considerably less time than directly inserting into a table. Use prepared SQL statements to speed up processing.

Using begin work, commit work, and rollback work

Another posting technique ensures database integrity.

When you are using more than one SQL statement in a posting program, there is always a danger that one SQL statement executes successfully while another one fails. This means that you have partially posted a record; for example, a "one" row is inserted but its "many" rows are not.

Fortunately 4GL allows you to define the boundaries of a record. You can issue an SQL begin work command as the record begins processing, and then a commit work command when the record has finished processing. After each SQL statement is executed, a test for successful execution is made. If an SQL statement fails, then a rollback work command is issued and any SQL statements after the begin work command that have been executed are undone.



A begin work is placed in the `before_group` function or the `b_g_[grouping column]` function before any SQL statement. This is where a record starts processing.

```
#_b_order_num - insert before group code here
begin work
```

A commit work is placed in the `after_group` function or the `a_g_[grouping column]` function after any SQL statements. This is where a record completes its processing.

```
commit work
end function
# a_g_order_num()
```

Section Summary

- The following are various types of posting programs:
 - Posting programs that take data from one table and insert it into another table.
 - Posting programs that take data from multiple tables and insert the data to multiple tables.
 - Posting programs that update values in columns.
- Creating a posting program is a two-step process: use the Fitrix Report Writer to create a report that pulls all values to be posted into the program, and then use the Featurizer to place SQL statements that do the actual posting into the code.
- It is critical to ensure that all values to post are selected into the posting program. Use of the "Hidden Fields" capability in the Fitrix Report Writer is especially useful for accessing values that do not appear on a report.
- The performance of posting programs may be improved by using 4GL prepare and execute commands in lieu of insert or update.
- When posting to multiple tables, it is important to use the begin work/commit work/rollback work commands to ensure database integrity.

Lab Exercise 7

Objective: To create an inactivation program that inactivates customers. This type of posting program prompts the user for a customer to inactivate, and then inactivates the customer by setting a flag in the customer table.

This lab uses the Selection Criteria screen you created in the previous section's lab. The customer that the user enters into the Selection Criteria screen is the customer to be inactivated.

Create a New Program Directory

1. `cd $HOME/labs/aw.4gm`
2. **Make a new program directory prefixed with a "p_" and give it a .4gs extension. Call it p_cust.4gs**
When making program directories, make sure the entire prefix does not exceed eight characters. Posting programs are prefixed with a "p_".
3. **cd into your new posting directory.**
4. **Copy in all files from your last lab.**
You will convert the prior lab's report to a posting program.
5. **Remove the executable .42r file.**

Add a Column with dbaccess

1. **Enter dbaccess and add a column called "inactive" to the trcustomer table.**
2. **Make its data type char (1) and exit out of dbaccess.**
3. **Update the schema file in \$fg/data.**

```
cd $fg/data
sch.sh student{ }
cd $HOME/labs/aw.4gm/p_cust.4gs
```
4. **Run the Fitrix Report Writer.**
5. **Load the "Orders Listing" report.**
6. **In the Fitrix Report Writer, change the name of the report to "Customer Inactivation Report."**
7. **Save the report.**

8. **Export the Report. Make sure to change the export name to report.ifg.**
9. **Exit the Fitrix Report Writer.**

Generate Code

Generate code on your modified `report.ifg` file using the Fitrix Report Code Generator.

Add an SQL Statement with an .ext File

Next use the Featurizer to place an SQL statement into code. The SQL statement will be an update which places a "Y" value into the `inactive` column for the customer that the user selects to inactivate.

1. Use `vi` to open `lowlevel.4gl`.
2. Find the function that handles after group logic for the customer number field (`a_g_customer_num()`).
3. Find the block within the above function that allows you to place custom after group logic (`#_a_customer_num`).

This is the block that you will manipulate with the Featurizer.

4. **Exit** `lowlevel.4gl`.
5. **Create an .ext file with a descriptive prefix.**

Now create an .ext file to hold a block command with the modification you wish to make. Create an .ext file now (just the file). The prefix to the .ext file should be indicative of the "feature" you are incorporating, such as:

```
inact.ext
```

6. **In your newly created .ext file, insert a start file command at the top.**

Use the following start file command:

```
start file "lowlevel.4gl"
```

Which .4gl file contains the block you are going to manipulate?

7. **Under `start file`, place the block command that will place your update statement into code.**

The block command should look as follows:

```
after block a_g_customer_num a_customer_num
```

8. **Insert an SQL update statement to set the current customer's inactive column to "Y."**

For example, you could use the following SQL statement:

```
UPDATE trcustomer SET trcustomer.inactive = "Y"
WHERE trcustomer.customer_num = curs.customer_num
```

9. End the block with a semi-colon (“;”) and save and quit out of the .ext file.

Merging with the .set File

1. vi your base.set file.
2. On the first line, add the prefix to the .ext file you created.

Add only the prefix. Make sure that your selection criteria screen is kept in the .set file, because your inactivation feature will depend on the selection criteria feature being in code.

Merge and Compile with fg.make

1. Start the Featurizer:

```
fg.make -mf
```

The fg.make command invokes the Featurizer and merges your logic into the code. The “-mf” flag instructs fg.make to only invoke the Featurizer and not compile and link.

2. Use vi to open lowlevel.4gl and notice your change.

Is your change there?

3. Complete the compile, and link the source code with the following command again:

```
fg.make
```

Run the Program

```
fglrun p_cust.42r -d student{}
```

Enter a single customer number.

Verify Through dbaccess

Enter dbaccess and select the customer you updated. Is the value of the inactive column “Y” for this customer?

Chapter 8

Other Features

Major topics in this chapter include the following:

- Report Exporting
- Post Processor Flexibility

Exporting Report Data

The Fitrix Report Code Generator can generate code for exporting report data to comma delimited (.csv) files. A .csv (comma separated values) file is useful for creating spreadsheets and importing data into other application.

To use this report exporting feature, you add an export section to the `report.ifg`. The following example is from `$fg/accounting/ar.4gm/o_custls.4gs`

```
attributes
  h1 = date
  h2 = constant "Customer Information Summary"
  h3 = time
  h4 = formonly.co_name type like stxcntrc.co_name
  h5 = pageno using "Page: <<<<"
  e1 = strcustr.cust_code
  e2 = strcustr.bus_name
  e3 = strcustr.contact
  e4 = strcustr.phone
end

select
  tables = strcustr

defaults
  progname = o_custls
  prcname  = Customer Information Summary
  rtmargn  = Fourth Generation Accounting
  destin   = report.out
  quiet    = 10
  allow_int = Y

export
  columns = strcustr.cust_code "customer code", strcustr.bus_name "business name"
  columns = strcustr.contact "contact", strcustr.phone "phone"
```

You list all the columns you wish to export in the `columns =` statement in the export section. To customize the headings in the export file you add the column heading after the column name. You may list multiple columns in a `columns =` statement by separating the columns with a comma. If you do not add column headings, the name of the column is used for the column heading.

The environment variable `fx_export` determines if a .csv file is generated. Set `fx_export=1` to generate a .csv file. Set `fx_export=0` to not generate a .csv file.

Post Processor Flexibility

The Fitrix Report Code Generator lets you customize generated source code with a post processor. A post processor is useful for many tasks, such as making global changes to code, replacing or altering code blocks, and implementing bug fixes. The advantage of using the post processor is that you can make global changes without adding extension files to every report program.

The `fg.report` program runs a post processor on the local application if the environment variable `$local_rpt` is set. Use this variable to point to the name of the program you wish to run on the generated 4GL code.

The same arguments that you pass to `fg.report` get passed to your post-processor program.

For example, assume you write an initialization function (say, `my_init()`) that is more relevant than the generic `init()` function that the Fitrix Report Code Generator creates. You may want `main.4gl` to call `my_init()` rather than the `init()` function. You can set up a post processor to change the initialization call in `main.4gl` to `my_init()`.

To change `init()` in `main.4gl` to `my_init()`:

1. **Write a shell script** (`chg_init`, for example) that uses the `sed` utility to remove `init()` and add a call to `my_init()`, such as:

```
#chg_init
sed "s, call init, call my_init, " main.4gl > main.tmp &&
mv main.tmp main.4gl
```

2. **Set your `$local_rpt` environment variable to the name of the post-processor script** (you might want to set this variable in your `.profile` file), for example:

```
local_rpt=chg_init; export local_rpt
```

Once the Fitrix Report Code Generator completes creating source (`.4gl`) files, your local `main.4gl` file contains the function call `my_init()` rather than `init()`.

Section Summary

- Comma delimited files (.csv) files can easily be generated using the Report Export feature of the Fitrix Report Code Generator.
- To generate the code required for the report exporting you add an export section to the report.ifg file.
- Use the `fx_export` variable to turn the .csv file generation on and off. `fx_export=1` turns report exporting on, `fx_export=0` turns report exporting off.
- The Fitrix Report Code Generator post processor can be used to make global changes to report programs. You set the `$local_rpt` environment variable the name of a shell script. The `$local_rpt` shell script is executed automatically after the Report Code Generator is run.

Exercise 8A

1. In `$HOME/labs/aw.4gm` create a new report directory `o_cust.4gs` and move to that directory.
2. Start Fitrix Report Writer.
3. Load the Customer Phone Numbers report.
4. Export the report making sure to call the export report.ifg.
5. Exit the Fitrix Report Writer.
6. Use `vi` to edit the `report.ifg`. You must add an export section. Add the export section below.

```
select
  name =
  tables = trcustomer, outer(trcredit)
  join = trcredit.credit_code = trcustomer.credit_code
end

export
  columns = trcustomer.customer_num "Customer Number"
  columns = trcustomer.company "Company Name",trcustomer.phone "Phone"
end

defaults
  progname = o_cust
  prcname = Customer Phone Numbers
  destin = report.out
end
```

7. Generate the code using the Fitrix Report Code Generator
8. `echo fx_export`. Make sure `fx_export=1`. `fx_export` must be set to 1 in order for the `report.csv` file to be created.
9. Run the program.
10. Use `vi` to examine the `report.csv`. See how the labels you entered in the export section become the heading in the `report.csv` file.