Contents

Preface ................................................................. 9
  Documentation Conventions ........................................ 10
  Related Publications ............................................. 11
  Customer Support .................................................. 11
  Information You Should Have ..................................... 12
  User Feedback ..................................................... 13
  Information Builders Consulting and Training .................. 13

1. Financial Reporting and Analysis for the Enterprise .............. 15
  WebFOCUS Financial Reporting Benefits .......................... 15
  Enterprise Financial Reporting: A Critical Cornerstone of Business ............................................. 16
  Financial Reporting Within and Beyond an Organization .......... 17
    Internal Financial Reports ........................................ 17
    External Financial Reports ....................................... 17
  Financial Reporting and Government Regulations ................ 17
  The WebFOCUS Financial Reporting Platform ..................... 18
    Data Consolidation and Aggregation ............................ 18
    Report Generation ............................................... 19
    Financial Analysis .............................................. 19
    Distribution .................................................... 20
    Archiving ...................................................... 20
  WebFOCUS Financial Report Painter ............................... 21
    Advantage Over SQL-Based Financial Reporting Tools .......... 22
    Advantages for Data Access ..................................... 23
  Learning More About the WebFOCUS Financial Report Painter .... 23

2. Creating Reports With the Financial Report Painter ............ 25
  Financial Report Painter Overview ................................ 26
  Starting a Financial Report ....................................... 26
    Procedure: How to Start a Financial Report .................... 27
  Designing Your Financial Report .................................. 28
  Viewing Source Code .............................................. 35
    Procedure: How to View Source Code ............................ 35
    Previewing Your Report ......................................... 36

Creating Financial Reports 3
<table>
<thead>
<tr>
<th>Procedure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Procedure: How to Toggle Between Design and Preview Modes.</td>
<td>36</td>
</tr>
<tr>
<td>Running a Financial Report</td>
<td>37</td>
</tr>
<tr>
<td>Procedure: How to Run a Financial Report.</td>
<td>37</td>
</tr>
<tr>
<td>Creating a Financial Report Using a Cube Data Source</td>
<td>38</td>
</tr>
<tr>
<td>Procedure: How to Create a Financial Report Using a Cube Data Source.</td>
<td>38</td>
</tr>
<tr>
<td>Retrieving FOR Field Values From a Data Source</td>
<td>39</td>
</tr>
<tr>
<td>Procedure: How to Specify a Tag for a Single Data Value.</td>
<td>39</td>
</tr>
<tr>
<td>Procedure: How to Specify a Tag for Multiple Data Values.</td>
<td>40</td>
</tr>
<tr>
<td>Procedure: How to Specify a Tag for a Range of Data Values.</td>
<td>42</td>
</tr>
<tr>
<td>Procedure: How to Specify a Tag for a Related Group of Data Values.</td>
<td>43</td>
</tr>
<tr>
<td>Reference: TAG Dialog Box.</td>
<td>44</td>
</tr>
<tr>
<td>Reporting Dynamically From a Hierarchy</td>
<td>47</td>
</tr>
<tr>
<td>Procedure: How to Specify Tags for Data Values in a Hierarchy.</td>
<td>49</td>
</tr>
<tr>
<td>Procedure: How to Adjust Indents for Captions in a Hierarchy.</td>
<td>51</td>
</tr>
<tr>
<td>Displaying an FML Hierarchy</td>
<td>52</td>
</tr>
<tr>
<td>Consolidating an FML Hierarchy</td>
<td>57</td>
</tr>
<tr>
<td>Procedure: How to Consolidate FML Hierarchy Data.</td>
<td>63</td>
</tr>
<tr>
<td>Reference: Sample Data for FML Hierarchy Examples.</td>
<td>68</td>
</tr>
<tr>
<td>Supplying Data Directly</td>
<td>69</td>
</tr>
<tr>
<td>Procedure: How to Add Data Values Directly to the Report.</td>
<td>69</td>
</tr>
<tr>
<td>Reference: DATA Dialog Box.</td>
<td>70</td>
</tr>
<tr>
<td>Performing Inter-Row Calculations</td>
<td>71</td>
</tr>
<tr>
<td>Procedure: How to Create a Recap Expression in Financial Report Painter.</td>
<td>71</td>
</tr>
<tr>
<td>Reference: RECAP Dialog Box.</td>
<td>73</td>
</tr>
<tr>
<td>Reference: Format Dialog Box.</td>
<td>78</td>
</tr>
<tr>
<td>Procedure: How to Create a RECAP Summary Row.</td>
<td>81</td>
</tr>
<tr>
<td>Creating RECAP Expressions Using a Drag-and-Drop Operation.</td>
<td>81</td>
</tr>
<tr>
<td>Procedure: How to Create RECAP Expressions Using a Drag-and-Drop Operation.</td>
<td>82</td>
</tr>
<tr>
<td>Inserting Rows of Text</td>
<td>82</td>
</tr>
<tr>
<td>Procedure: How to Add Text to a Financial Report.</td>
<td>82</td>
</tr>
<tr>
<td>Reference: TEXT Dialog Box.</td>
<td>83</td>
</tr>
<tr>
<td>Suppressing the Display of Rows</td>
<td>83</td>
</tr>
<tr>
<td>Procedure: How to Suppress Rows.</td>
<td>83</td>
</tr>
<tr>
<td>Procedure: How to Suppress Rows With No Data.</td>
<td>84</td>
</tr>
</tbody>
</table>

Create a Project .................................................. 115
Add Master Files to Your Project .................................. 119
Contents

Formatting an FML Report ............................................................. 234
  Syntax: How to Add an Underline Character for Columns. ....................... 235
  Syntax: How to Specify a Page Break in an FML Report. .......................... 236
  Syntax: How to Format a Row, Column, or Cell in an FML Report. ......... 237
  Syntax: How to Add and Format Row and Cell Borders. ......................... 246
  Syntax: How to Specify an Indent for an FML Label, Tag, or Caption. ...... 250
Indenting Row Titles in an FML Hierarchy ........................................... 253
  Syntax: How to Indent FML Hierarchy Captions in an HTML Report. ......... 254
Suppressing the Display of Rows ...................................................... 256
  Suppressing Rows With No Data. ...................................................... 257
Saving and Retrieving Intermediate Report Results ................................. 258
  Posting Data .............................................................................. 258
  Syntax: How to Post Data to a File. .................................................. 259
  Syntax: How to Pick Up Data From a Work File. .................................... 260
  Creating HOLD Files From FML Reports ............................................. 261
B. Describing Data for an FML Hierarchy ............................................. 263
  Data Requirements for FML Hierarchies ............................................. 263
  Coding an FML Hierarchy in a Text Editor .......................................... 264
    Syntax: How to Specify a Hierarchy Between Fields in a Master File. .... 264
    Syntax: How to Assign Descriptive Captions for Hierarchy Field Values. 264
  Defining an FML Hierarchy in the Synonym Editor ................................. 266
    Reference: Synonym Editor: Column and Field Attributes. ................. 267
C. Glossary ...................................................................................... 271
Customer Connections ................................................................. 285
Preface

This documentation describes how to use the Financial Report Painter. It is intended for application developers and others who are responsible for creating financial reports.

Note: The WebFOCUS toolset generates the rich FOCUS fourth generation language. While this language is very extensive, the WebFOCUS toolset only supports a subset of the language and specific syntax constructs. While the user can manually modify the content of these WebFOCUS procedures and files, there is no guarantee that the user will be able to open the modified procedure in the tool.

How This Manual Is Organized

This manual includes the following chapters:

<table>
<thead>
<tr>
<th>Chapter/Appendix</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Financial Reporting and Analysis for the Enterprise</td>
<td>Introduces the WebFOCUS Financial Reporting Platform, Information Builders single vendor enterprise business intelligence solution, and describes the role of the WebFOCUS Financial Report Painter. The Financial Report Painter is designed for the special needs associated with creating, calculating, and presenting financially oriented data, such as income statements, balance sheets, consolidations, profit and loss statements, budgets, and certain government mandated financial reports.</td>
</tr>
<tr>
<td>2 Creating Reports With the Financial Report Painter</td>
<td>Describes how to create a report to accommodate the special needs associated with creating, calculating, and presenting financially oriented data.</td>
</tr>
<tr>
<td>3 Tutorial: Creating a Financial Report Using the Financial Report Painter</td>
<td>Provides step-by-step instructions for using the Financial Report Painter to create an income statement, through a pre-consolidated data mart that defines a Chart of Accounts hierarchy. The tutorial introduces basic concepts in financial reporting and presents a methodology for when you are working with your own data.</td>
</tr>
<tr>
<td>Chapter/Appendix</td>
<td>Contents</td>
</tr>
<tr>
<td>-----------------</td>
<td>----------</td>
</tr>
<tr>
<td>4 Tutorial: Creating an Income Statement Using Unconsolidated Data</td>
<td>Provides step-by-step instructions for creating an income statement using joined data sources: a Chart of Accounts hierarchy and a General Ledger of accounts that contains unconsolidated detail-level data. This tutorial introduces a variety of consolidation methods, as well as a technique that resolves differences between internal data storage conventions in a typical General Ledger system and user expectations for how those values should be displayed in a report.</td>
</tr>
<tr>
<td>B Describing Data for an FML Hierarchy</td>
<td>Documents the syntax required in a Master File to supportFML hierarchies. It provides the syntax for manual coding, as well as the corresponding entries in the Synonym Editor.</td>
</tr>
<tr>
<td>C Glossary</td>
<td>Describes key terms found in this manual.</td>
</tr>
</tbody>
</table>

**Documentation Conventions**

The following table describes the documentation conventions that are used in this manual.

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>THIS TYPEFACE</strong> or this typeface</td>
<td>Denotes syntax that you must enter exactly as shown.</td>
</tr>
<tr>
<td>this typeface</td>
<td>Represents a placeholder (or variable) in syntax for a value that you or the system must supply.</td>
</tr>
<tr>
<td>underscore</td>
<td>Indicates a default setting.</td>
</tr>
<tr>
<td>this typeface</td>
<td>Represents a placeholder (or variable), a cross-reference, or an important term. It may also indicate a button, menu item, or dialog box option that you can click or select.</td>
</tr>
</tbody>
</table>
### Convention | Description
--- | ---
Key + Key | Indicates keys that you must press simultaneously.
{ } | Indicates two or three choices. Type one of them, not the braces.
[ ] | Indicates a group of optional parameters. None are required, but you may select one of them. Type only the parameter in the brackets, not the brackets.
| | Separates mutually exclusive choices in syntax. Type one of them, not the symbol.
... | Indicates that you can enter a parameter multiple times. Type only the parameter, not the ellipsis (...).
. | Indicates that there are (or could be) intervening or additional commands.

### Related Publications

Visit our Technical Content Library at [http://documentation.informationbuilders.com](http://documentation.informationbuilders.com). You can also contact the Publications Order Department at (800) 969-4636.

### Customer Support

Do you have any questions about this product?

Join the Focal Point community. Focal Point is our online developer center and more than a message board. It is an interactive network of more than 3,000 developers from almost every profession and industry, collaborating on solutions and sharing tips and techniques. Access Focal Point at [http://forums.informationbuilders.com/eve/forums](http://forums.informationbuilders.com/eve/forums).

You can also access support services electronically, 24 hours a day, with InfoResponse Online. InfoResponse Online is accessible through our website, [http://www.informationbuilders.com](http://www.informationbuilders.com). It connects you to the tracking system and known-problem database at the Information Builders support center. Registered users can open, update, and view the status of cases in the tracking system and read descriptions of reported software issues. New users can register immediately for this service. The technical support section of [www.informationbuilders.com](http://www.informationbuilders.com) also provides usage techniques, diagnostic tips, and answers to frequently asked questions.
Call Information Builders Customer Support Services (CSS) at (800) 736-6130 or (212) 736-6130. Customer Support Consultants are available Monday through Friday between 8:00 a.m. and 8:00 p.m. EST to address all your questions. Information Builders consultants can also give you general guidance regarding product capabilities. Please be ready to provide your six-digit site code number (xxxx.xx) when you call.

To learn about the full range of available support services, ask your Information Builders representative about InfoResponse Online, or call (800) 969-INFO.

**Information You Should Have**

To help our consultants answer your questions effectively, be prepared to provide the following information when you call:

- Your six-digit site code (xxxx.xx).
- Your WebFOCUS configuration:
  - The front-end software you are using, including vendor and release.
  - The communications protocol (for example, TCP/IP or HLLAPI), including vendor and release.
  - The software release.
  - Your server version and release. You can find this information using the Version option in the Web Console.
- The stored procedure (preferably with line numbers) or SQL statements being used in server access.
- The Master File and Access File.
- The exact nature of the problem:
  - Are the results or the format incorrect? Are the text or calculations missing or misplaced?
  - Provide the error message and return code, if applicable.
  - Is this related to any other problem?
- Has the procedure or query ever worked in its present form? Has it been changed recently? How often does the problem occur?
- What release of the operating system are you using? Has it, your security system, communications protocol, or front-end software changed?
- Is this problem reproducible? If so, how?
Have you tried to reproduce your problem in the simplest form possible? For example, if you are having problems joining two data sources, have you tried executing a query containing just the code to access the data source?

Do you have a trace file?

How is the problem affecting your business? Is it halting development or production? Do you just have questions about functionality or documentation?

**User Feedback**

In an effort to produce effective documentation, the Technical Content Management staff welcomes your opinions regarding this document. You can contact us through our website, [http://documentation.informationbuilders.com/connections.asp](http://documentation.informationbuilders.com/connections.asp).

Thank you, in advance, for your comments.

**Information Builders Consulting and Training**

Interested in training? Information Builders Education Department offers a wide variety of training courses for this and other Information Builders products.

For information on course descriptions, locations, and dates, or to register for classes, visit our website ([http://education.informationbuilders.com](http://education.informationbuilders.com)) or call (800) 969-INFO to speak to an Education Representative.
WebFOCUS provides a total enterprise financial reporting solution that significantly reduces development time and maintenance costs, while providing flexibility, accessibility, and accuracy in enterprise-wide financial activities.

WebFOCUS real-time, web-based financial reporting delivers accurate and timely knowledge of current and future operations, performance, and financial status for the mission-critical financial needs of an organization, including:

- Budgeting
- Planning
- Consolidation
- Forecasting

**In this chapter:**

- WebFOCUS Financial Reporting Benefits
- Financial Reporting Within and Beyond an Organization
- Financial Reporting and Government Regulations
- The WebFOCUS Financial Reporting Platform
- WebFOCUS Financial Report Painter
- Learning More About the WebFOCUS Financial Report Painter

**WebFOCUS Financial Reporting Benefits**

From accountants to financial analysts to IT personnel, whether users are technical or nontechnical, the entire organization will benefit. CFOs will find WebFOCUS financial reporting especially useful for generating balance sheets, and CEOs will appreciate its ability to focus the organization on growth and cash flow to help fund a continual stream of product innovations, acquisitions, and to meet government regulations.
Integral to it all, the powerful WebFOCUS Financial Report Painter enables organizations to build these specialized financial reports that are critical to a wide range of departments, knowledge workers, managers, suppliers, lenders, investors, and government agencies.

**Enterprise Financial Reporting: A Critical Cornerstone of Business**

Financial information is a critical cornerstone of business. Nearly every organization needs to be able to access financial data and turn it into useful financial information in order to understand where their business has been, know where it is now, and plan for where it will be in the future.

To achieve this, organizations must have effective technology and financial tools to track income, expenses, assets, and liabilities in an organized manner. They must have ways to convert data to information that can provide frequent, consistent, and accurate insight into financial operations and performance. They must have financial information that enables them to look forward in order to plan and forecast for the future. They must also have financial information that enables them to understand what went wrong in the past without losing valuable time and resources validating, justifying and auditing previous financial situations.

Efficient and effective financial reporting solutions depend on cost effectiveness, usefulness, and flexibility. To be cost effective, the benefits of information must outweigh the cost of providing it. To be useful, information must be understandable, reliable, relevant, accurate, and timely.

Such financial reporting solutions need to accommodate a wide range of users and be able to efficiently and flexibly handle changing information requirements. In addition, and no less important, financial reporting solutions must be able to provide reliable data consolidation and aggregation, easy report generation, distribution and archiving, and financial analysis.

Today, mergers, acquisitions, global expansion, investor concerns, and government regulations have made the need for accurate and real-time delivery of financial information that meets these requirements even more critical for business operations.

From accounting to decision making, WebFOCUS financial reporting provides organizations with the solutions that meet their varied needs for obtaining, evaluating, and delivering a total view of their financial status, including:

- Greater availability of financial information.
- More timely reporting processes.
- More detailed financial analysis.
- More individuals interacting with financial data.
- Increased responsibility of sharing and management of financial information.
Financial Reporting Within and Beyond an Organization

Financial reports impact businesses, especially the financial and managerial accounting needs of organizations. Organizations may need financial reports to describe their financial condition and provide a framework for examining past performance, current financial situations, possible future performance, and the financial positions and performances of various employees, groups, and departments within and outside of these organizations. Organizations may require reports that describe their financial position at a particular point in time and their financial performance during a specified range of time.

Internal Financial Reports

Internally, financial reports may be required by various levels of managers within the organization to evaluate and make decisions about current and future operations. The reports may require highly detailed or summarized financial information, depending on management requirements for decision making. These internal financial reports may include sales reports, production cost reports, or other detailed financial reports.

External Financial Reports

Externally, financial reports may be required by people or agencies outside the organization who have an interest in the business activities of that organization. For example, suppliers, lenders, investors, and government agencies may need to access the financial status of a business for any number of reasons. External financial reports may include balance sheets, income statements, or cash flow statements.

Financial Reporting and Government Regulations

While financial reporting is critical to the inner workings of every business, it is becoming even more critical to the outside stakeholders of business. Evidence of this can be seen in the demands of investors and government for financial reporting reforms in the wake of recent corporate accounting scandals and financial improprieties. Thus, the demand for better, more timely financial information tailored to a wider audience is more urgent today than ever, and by all indications, the trend will continue.
For example, the Sarbanes-Oxley Act contains key provisions to prevent corporate fraud and restore investor confidence in financial reports, highlighting the need for organizations to produce better financial information and be more accountable. To meet the demands of government agencies, investors, creditors, and others for accurate and current financial information, organizations must be able to disclose correct and timely financial information in varied report styles and formats for distribution, not only within the organization, but outside of it. While financial manipulation resides in the province of ethics, which no financial reporting solution can change, financial reporting solutions must be effective to help eliminate misleading financial statements and valuations. Organizations must put in place financial reporting solutions that, for example, promptly recognize reversals of income that had been booked on the basis of projections, which later may be proven to have been inaccurate. Financial reporting solutions must be able to generate objective and accurate valuations of assets and clear provable balance sheets in a timely manner.

The WebFOCUS Financial Reporting Platform

The WebFOCUS Financial Reporting Platform is an enterprise business intelligence solution that can access, develop, forecast, archive, and report on vital financial information from a wide range of data sources, across many departments and business units.

Data Consolidation and Aggregation

Consolidating data based on Charts of Accounts or other hierarchies is a powerful feature that enables real-time data access for timely and accurate financial reports. Typically, data is stored at the posted account level in the database. However, if you want to report on assets or liabilities, you first must consolidate the lower-level accounts to higher levels. For example, if you want to print a report, such as an income statement, you need the numbers for Total Revenue. Total Revenue is not typically stored in the database. However, the revenue numbers are all posted in the database at low levels. Thus, some sort of mechanism is required so that at report time, these lower-level numbers show up as consolidated numbers, or numbers rolled up to a higher level. These consolidated (rolled up) numbers can then be shown as Total Revenue in the financial statement.

WebFOCUS consolidation of financial data provides the benefits of quickly and easily managing financial reporting without the overhead involved in cube creation. It also speeds the creation of data warehouses to make it easier to generate financial reports with more flexibility when dealing with various business mergers, acquisitions, and new business units.
Report Generation

WebFOCUS report generation makes it easy to build a wide range of financial reports for many uses and many audiences. Organizations can get a worldwide picture of financial and business activities because WebFOCUS financial reports can handle international currency conversion, reading and interpreting national characters, and handle sorting, case conversions and formatting of dates, currencies, and numbers. Multiple currencies can be shown on a single report with appropriate symbols, and reporting applications can be developed that provide users with the ability to select the language in which they are most comfortable working.

Additionally, dynamic posting capability allows for generating summary statements from individual, detailed reports. Additionally, the ability to dynamically roll up financial data that has been consolidated simplifies fast generation of a range of financial reports. For more information, see WebFOCUS Financial Report Painter on page 21.

Financial Analysis

WebFOCUS financial reports that contain active drill downs can take advantage of WebFOCUS OLAP. Within a financial report, for example, you can drill down on a field, such as Income Statement. The drill down takes you to a non-financial report, from which you can access a WebFOCUS OLAP panel. In the WebFOCUS OLAP panel, you can perform extensive data analysis functionality on the organization. For example, you can slice and dice data in many ways to answer vital business questions such as, "What is selling and at what price point?" or "What products in what geographical location account for the largest percentage of my profits?"

WebFOCUS financial reports, with drill downs intact, can be automatically transferred to and displayed in Excel®. This conversion greatly reduces the possibility for errors. Report elements, including field names, headers, subtotals, sort breaks, and drill downs, are preserved within Excel. Clicking on embedded hyperlinks in Excel produces the same detailed reports as those created in a WebFOCUS financial report. With report information transposed into Excel, users can take advantage of Pivot Tables and other features. In addition, WebFOCUS supports a direct save of report data to Pivot Tables, which provide a powerful data-mining capability. By starting with a relational request to get the scope of desired data into a single report, and then automatically generating a Pivot Table for iterative exploration, this feature supports a workflow common to many organizations. Automatic generation of Pivot Tables is a WebFOCUS exclusive.

WebFOCUS forecasting features can help executives identify trends in numeric data and predict values beyond the range of values stored in the data source, enabling business management to anticipate future conditions and plan ahead.
Distribution

After financial reports are generated, organizations can use their existing technology and data-communication infrastructures to deliver reports across the enterprise in a number of ways, such as email, the web, and wireless devices. Financial reports can be distributed:

- On a scheduled or event-driven basis.
- Using automatic bursting, delivering reports to each user based on his or her security profile, access permissions, or other credentials. This more efficiently makes use of network resources.
- By setting up portal-like interfaces from which users can access financial reports.
- By incorporating financial reports into leading third-party portals such as Microsoft® SharePoint®, IBM® WebSphere®, PeopleSoft®, SAP®, and Oracle® WebLogic®.

The reports can be distributed:

- In a variety of output formats, including a PDF file in Adobe® Acrobat® format, an email attachment, simple HTML text, or direct input into an Excel spreadsheet. For example, a financial analyst can receive a monthly P&L report in an Excel spreadsheet, then analyze it offline instead of rekeying data from a paper printout into the spreadsheet.
- In multiple sections, with each section sent to separate recipients. For example, a single sales report can be sent to multiple regions, with the manager of each region receiving only the information pertaining to their territory.
- To many different people in multiple versions. Unlike most other business intelligence providers, WebFOCUS needs to run a report only once to achieve multiple report delivery to multiple people.

Archiving

The WebFOCUS Report Library is an archiving facility that provides organizations with the ability to store and manage financial reports for future retrieval and use. Because businesses generate large volumes of financial information on a regular basis, the ability to archive reports in a secure central storage and efficiently restore them is crucial for efficient compliance with information retention guidelines such as those of the IRS, which requires businesses to retain records for auditing purposes.

Among key features:

- **Versioning.** The ability to save multiple versions of the same report or document.
- **Expiration.** The ability to set an expiration date to a file or limit versions to a specified number.
- **Categorization.** The ability to logically organize documents by categories and domains, so they are easy to locate and retrieve.

- **Security.** The ability to keep sensitive information protected from unauthorized users by defining which users can view stored content.

- **E-mail Notification.** The ability to send an email automatically to users to inform them when a new file is added to the archive and provide them with a URL link to the information.

**WebFOCUS Financial Report Painter**

Just as financial information is a critical cornerstone of business, the WebFOCUS Financial Report Painter is a critical cornerstone of the WebFOCUS Financial Reporting Platform.

Fully integrated with the WebFOCUS Financial Reporting Platform, the robust Financial Report Painter, with its intuitive interface and matrix workpad, makes it easy for developers to create powerful, tailored financial reports that can:

- Perform calculations between row and column elements to produce reports in a row-by-row, column-by-column format.

- Post reports to an external file and pick them up at a later time for analysis. This is a powerful feature to develop and consolidate intermediate reports, then use the results again in a later report.

- Produce recursive reports, where the results from the end of one time period or column become the starting balance in the next. This is an effective way for organizations to project cash flow or other forward-looking reports.

- Generate profit and loss statements, consolidations, actuals, budgets, and government-mandated financial reports easily by simple point-and-click actions.

- Dynamically create summary financial statements from detailed financial data.

- Build financial reports from a wide range of data sources and structures, including Charts of Accounts and other hierarchies.

- Generate accurate financial reports directly from centralized, secure, and professionally maintained data sources. Organizations performing financial reporting through manual spreadsheets and other static, limited processes are at a distinct disadvantage in this highly scrutinized fast-paced business climate.

- Provide a wide range of report styling and customizing options to fit the needs of different audiences.
The Financial Report Painter is driven by the powerful financial modeling language within WebFOCUS, which is capable of handling large volumes of data, volatile data relationships, and custom Charts of Accounts. These Charts of Accounts can be quickly displayed in hierarchical trees that can be compressed or expanded with a simple click of the mouse, making it easy for a wide range of users to:

- View the contents and structure of financial data.
- Drag an account from any level and drop it into a report.
- Produce rows with different levels of aggregation.
- Minimize report maintenance as a result of Charts of Accounts changes. When accounts are added, deleted or realigned, the report procedure will not need to change to reflect the restructuring.
- Dynamically consolidate data in Charts of Accounts and generate real-time financial reports.

**Advantage Over SQL-Based Financial Reporting Tools**

Although many SQL-based reporting tools claim to be well-suited for financial reporting, evidence shows that these tools are actually limited in real-world situations. For example:

- Financial reports often require account data to be grouped and calculated by rows. SQL-based tools can only retrieve, group, and calculate data by columns.
- Financial reports often require items to be listed by liquidity or some other corporate or government-defined rule. SQL-based tools can only sort data alphabetically or numerically.
- Charts of Accounts hierarchies often change. SQL-based tools refer to account data explicitly, so financial reports built with SQL-based tools require constant maintenance.
- Organizations demand that financial reporting needs get filled fast and SQL-based tools usually require involvement of IT staff. However, IT personnel typically have limited understanding of financial and accounting operations, so reports must often be modified and revised several times before they fully meet the needs of an organization. This can slow down valuable decision-making time and place a burden on organization resources. Compliance with new reporting guidelines will place additional pressure on IT staff to build reports faster and more often.

Unlike SQL-based financial reporting tools, the WebFOCUS Financial Report Painter meets and exceeds these critical business demands rapidly and efficiently by:

- Retrieving, grouping, and calculating data by columns.
- Grouping and calculating account data by rows.
Sorting data alphabetically or numerically.

Dynamically maintaining reports as an organization Charts of Accounts change.

**Advantages for Data Access**

Another advantage of the WebFOCUS Financial Report Painter is its ability to access more than 100 data sources on some 35 platforms, including:

- Multi-dimensional cubes such as Oracle Essbase®.
- Analytical data structures, including DB2®, Oracle, SQL Server®, and Teradata®.
- Data resident on legacy systems, such as IMS™, VSAM, and IDMS on the mainframe, or VAX, UNIX®, Tandem, AS/400®, and Windows®, among others.

This ability for organizations to leverage all their data, from legacy to data warehouse, provides a powerful edge in enterprise financial reporting that very few financial reporting solutions can provide.

**Learning More About the WebFOCUS Financial Report Painter**

While this chapter provides an overview of WebFOCUS enterprise financial reporting, which is the most complete financial reporting solution your organization needs, this is just the beginning.

The rest of this book will introduce you to the WebFOCUS Financial Report Painter, which is the powerful, yet easy-to-use, financial report building tool created by Information Builders.

As you read about the WebFOCUS Financial Report Painter, you will begin to see more of its values, benefits, and advantages. Once you start to use it, you will learn how the full WebFOCUS financial reporting solution can fulfill the financial reporting requirements and improve the financial situation of your organization.

To learn how to use the WebFOCUS Financial Report Painter and find out more about its many features, see *Creating Reports With the Financial Report Painter* on page 25.


You can also refer to *Creating Financial Reports With Financial Modeling Language (FML)* on page 177, to discover the powerful language that drives the Financial Report Painter.

For more information about reporting against hierarchical data structures, see *Describing Data for an FML Hierarchy* on page 263.
This chapter provides information on how to use the Financial Report Painter.

In this chapter:

- Financial Report Painter Overview
- Performing Inter-Row Calculations
- Starting a Financial Report
- Inserting Rows of Text
- Designing Your Financial Report
- Suppressing the Display of Rows
- Running a Financial Report
- Saving and Retrieving Intermediate Report Results
- Creating a Financial Report Using a Cube Data Source
- Formatting Financial Reports
- Retrieving FOR Field Values From a Data Source
- Adding, Inserting, and Deleting Rows
- Reporting Dynamically From a Hierarchy
- Editing Row Types and Properties
- Supplying Data Directly
- Adding and Deleting Columns
Financial Report Painter Overview

The WebFOCUS Financial Report Painter, and its underlying Financial Modeling Language (FML), are designed for the special needs associated with creating, calculating, and presenting financially oriented data, such as balance sheets, consolidations, profit and loss statements, budgets, and certain government mandated financial reports. These reports are distinguished from other WebFOCUS reports because calculations are inter-row, as well as inter-column. Each row or line represents a unique entry or series of entries that can be aggregated directly from the input data or calculated as some function of the data.

Using an intuitive matrix design, the Financial Report Painter enables you to model and generate real-time financial reports that handle large volumes of data, volatile data relationships, and custom charts of accounts.

The Financial Report Painter generates FML code, beginning with the FOR command that initiates the row-based orientation of financial modeling, and the OVER commands that stack the rows in the report.

For details about FML syntax, see Creating Financial Reports With Financial Modeling Language (FML) on page 177.

Starting a Financial Report

**Procedure:** How to Start a Financial Report

In Report Painter:

1. Add the fields you want to include in the report, including the field that you intend to designate as your FOR field. The FOR field will allow the report to be structured on a row-by-row basis in the Financial Report Painter.

   You can include vertical (BY) and/or horizontal (Across) sort fields, however, these fields are not required. They are frequently omitted from financial reports where sorting is controlled by the placement of FOR field values in the matrix. Nevertheless, you can include both BY phrases and Across phrases in the request.

2. You may add any of the following report elements as required: headings and footings, totals, subtotals, calculated values, and images.

3. Select the field that you want to use as the FOR field, and click the *For* button on the Report Painter Columns toolbar. A report request can only contain one FOR field.

4. Click the *Matrix* tab at the bottom of the window.


**Tip:** If you want to return to Report Painter, click the *Report* tab below the matrix or choose *Show Report Painter* from the View menu.

To reopen the Financial Report Painter from Report Painter, click the *Matrix* tab.
Example: Report Painter With the FOR Field Selected

In this example, an Income Statement is open in Report Painter. A heading is defined and the report columns are specified. 

Ledger Account is designated as the FOR field and the For button is selected on the Report Painter Columns toolbar.

The next step is to click the Matrix tab below the Report Painter window to open the report in the Financial Report Painter. For an illustration, see Designing Your Financial Report on page 28.

Designing Your Financial Report

Most of the work you will do in the Financial Report Painter occurs in the Design matrix, where you can add:

- Tags that represent values of the FOR field in the data source.
Data that is not derived from a data source, including intermediate results from other reports that are posted to a file and picked up as data.

- Free text and blank lines.
- Underlines to set off the results of computations.
- Formulas for inter-row and intra-row calculations.

You can also format rows, columns, and cells in the Design matrix.

The following matrix contains a simple income statement that tracks revenues, costs, and expenses.

![Image of a financial statement matrix]

**Note:** Like all Financial Report Painter requests, this one originated in Report Painter. See *Report Painter With the FOR Field Selected* on page 28.

**Reference:**  **Financial Report Painter Design Matrix**

The Financial Report Painter enables you to define the elements of your report request. It consists of the matrix itself, the panel containing FOR field values from the data source, a set of toolbars, and a set of Properties check boxes.
Below the matrix are four tabs that enable you to switch quickly between the Matrix view, the Source code, the Report as it appears in Report Painter, and the images that may have been embedded in the report.

**Matrix elements:**

**Row labels (R1, R2, and so on)**

Are supplied by default in the first column of the matrix, where they provide unique identifiers for each report row. They become important as you place various report elements in the rows of the matrix. These labels enable you to identify rows to be used in calculations. You can also use the labels to identify rows and cells for formatting.

You can replace the default labels with more meaningful labels of your own.

**Row Type**

Identifies the type of information included in each row of the report. This column appears by default, but initially the values are blank. The Financial Report Painter adds the row type as you define each element in the report. Row types are: TAG, DATA, RECAP, TEXT, BAR, and PICKUP.

**FOR Field**

Contains the field you designated in Report Painter. FOR is the underlying command that enables row-by-row control in a WebFOCUS financial report. A report can only contain one FOR field. The data values of the FOR field, or calculations based on those values, are stacked over each other in the matrix.

**Title**

Lists any titles you define for rows. For TAG rows added through the TAG dialog box, and for DATA and RECAP rows, the title is blank unless provided. For TAG rows dragged onto the matrix from the FOR field values, the tag value appears as the default row titles.

You can assign multiple data values in a single row and then assign an identifying title to the group.

If you are working with hierarchical data for which captions have been defined in the Master File, the captions appear in the Title column. For details, see *Reporting Dynamically From a Hierarchy* on page 47.

**Report fields**

Each field you specify in Report Painter (including the FOR field) appears as a column title in the matrix. If you include any sort (BY) fields, they appear before the others in the report.
Column widths are initially adjusted to fit the longest item in each column. As you type into the matrix or enter values in dialog boxes, you can adjust the column widths as required by dragging or double-clicking the boundary lines.

You can include additional columns in your report without returning to Report Painter. For details, see *Adding and Deleting Columns* on page 113.

**FOR field values panel**

This panel displays the values of the FOR field in a resizable, scrollable window to the right of the matrix. You can add values to the Design matrix by dragging or double-clicking.

If you are reporting against a hierarchy, the values in the panel reflect the hierarchy tree. For details, see *Reporting Dynamically From a Hierarchy* on page 47.

**Design toolbars**

Financial Report Painter toolbar icons, which appear above the matrix, enable you to perform the following tasks:

- Add tag values from a data source, supplementary data values, free text, underlines, and Recap formulas to the matrix.
- Post data to a file and pick up the data that you have posted.
- Add, insert, and delete rows.
- Run financial reports.
- Change font style, size, and color, and justification for cells in the matrix.
- Perform standard operations, such as Save, Cut, Copy, Paste, and Undo/Redo.
The toolbar icons are:

<table>
<thead>
<tr>
<th>Icon</th>
<th>ToolTip</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Report Options</td>
<td>Opens the Report Painter Report Options dialog box, which control all report-wide properties.</td>
</tr>
<tr>
<td></td>
<td>Run</td>
<td>Runs the report.</td>
</tr>
<tr>
<td></td>
<td>Design mode</td>
<td>Toggles between the full Design matrix and a subset of the matrix columns for preview purposes.</td>
</tr>
<tr>
<td></td>
<td>Show properties</td>
<td>Shows the properties of the selected row in the dialog box in which those properties were defined.</td>
</tr>
<tr>
<td></td>
<td>Tag</td>
<td>Opens the TAG dialog box where you can specify values and properties for the TAG row.</td>
</tr>
<tr>
<td></td>
<td>RECAP</td>
<td>Opens the RECAP dialog box where you can create an expression that derives the value of the RECAP row and assign row properties.</td>
</tr>
<tr>
<td></td>
<td>Text</td>
<td>Opens the TEXT dialog box where you can enter a row of text, or a blank row.</td>
</tr>
<tr>
<td></td>
<td>Bar</td>
<td>Opens the BAR dialog box where you can specify an underline character.</td>
</tr>
<tr>
<td></td>
<td>Data</td>
<td>Opens the DATA dialog box where you can assign properties, such as title and label, to a row whose data you supply directly in the matrix (rather than retrieving it from a data source).</td>
</tr>
<tr>
<td></td>
<td>Pickup</td>
<td>Opens the PICKUP dialog box where you can assign row properties, such as title and label, and indicate the name of a file from which to pick up a row of previously posted data.</td>
</tr>
<tr>
<td></td>
<td>Add a row</td>
<td>Adds a blank row at the bottom of the matrix.</td>
</tr>
<tr>
<td>Icon</td>
<td>ToolTip</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td><img src="image1" alt="Insert a row icon" /></td>
<td>Insert a row</td>
<td>Inserts a blank row between existing rows in the matrix. Also opens the Insert Row dialog box where you can select a row type (for example, Tag, Recap, Data, Text, Bar, or Pickup). You can also specify values and properties, such as title and label.</td>
</tr>
<tr>
<td><img src="image2" alt="Delete a row icon" /></td>
<td>Delete a row</td>
<td>Removes the selected row from the matrix.</td>
</tr>
<tr>
<td><img src="image3" alt="Make RECAP (Rows) icon" /></td>
<td>Make RECAP (Rows)</td>
<td>Creates a RECAP row based on the sum of other values in the matrix. (This option applies when no expression is required.)</td>
</tr>
<tr>
<td><img src="image4" alt="Make RECAP (Cells) icon" /></td>
<td>Make RECAP (Cells)</td>
<td>Creates a RECAP cell based on the sum of other values in the matrix. (This option applies when no expression is required.)</td>
</tr>
<tr>
<td><img src="image5" alt="Increase/Decrease Size icon" /></td>
<td>Increase/Decrease Size</td>
<td>Increases or decreases font size by one measurement at a time.</td>
</tr>
<tr>
<td><img src="image6" alt="Bold, Italic, Underline icon" /></td>
<td>Bold, Italic, Underline</td>
<td>Adds emphasis using the selected style.</td>
</tr>
<tr>
<td><img src="image7" alt="Justify icon" /></td>
<td>Justify</td>
<td>Justifies (Left, Right, or Center) the characters in the selected cell, or applies the default justification.</td>
</tr>
<tr>
<td><img src="image8" alt="Font Color icon" /></td>
<td>Font Color</td>
<td>Applies the color chosen from the palette to the characters in the selected cell.</td>
</tr>
<tr>
<td>Icon</td>
<td>ToolTip</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td><img src="image1.png" alt="Icon" /></td>
<td>Match All Styles</td>
<td>Provides a drop-down list with options that are primarily used to copy format characteristics from one column and apply the characteristics to another column. To activate the button, you must multi-select two or more columns. When you click one of the options from the drop-down list, the Report Painter copies the style from the first column selected and applies the style to all other selected columns. Depending upon your selection, certain toolbar buttons may be inactive or unavailable. The styling options are: Match All Styles, Match Font, Match Grid/Border, Match Background Color, Match Conditional Styling, Match Justification, and Match Width Attributes.</td>
</tr>
</tbody>
</table>

**Properties Area**

The Properties area above the matrix is a resizable, scrollable region in which you can assign appropriate properties to the report or to specific row types.

If an editable cell is selected in the matrix, its content appears in the Properties box below the check boxes. To enter values in a selected cell, type in the Properties box and press the Enter key.

**Row Properties** check boxes affect individual rows:

**Invisible**

Suppresses the display of a row.

**When Exists**

Suppresses the display of a row when no data exists.

**Post to**

Enables you to enter the ddname for a file to which you want to post a row of output for later pickup and use.
**FML Properties** check boxes affect the entire report:

**Use Multiple Values**

Enables you to add the same data value to multiple rows in the Design matrix. For example, the same value can exist as a single value in one row, as part of a range in another row, and in a calculation in a third row.

For data that has not been structured as a hierarchy, values continue to be listed in the FOR field values panel after being used in a row. Without this option, values are removed from the list after being used.

In a hierarchy, a used value remains in the FOR field values panel, where it appears in red to indicate that it has already been added to the matrix.

In both instances, you can reuse the value as required.

**First Instance**

Applies when you are reporting against a data source with shared members, such as ESSBASE, in which the same data can be defined multiple times with different hierarchy field values.

Unless this option is selected, data shared by two different parents will be counted twice in an aggregation operation. When this check box is selected, the Financial Report Painter applies the .FST prefix operator to all summed numeric columns in the report, ensuring that a shared value is only counted once.

**Viewing Source Code**

If you are familiar with the underlying FML syntax, as you work in the graphical tool you may wish to view the source code generated by the Financial Report Painter.

**Procedure:** How to View Source Code

To toggle from a report request in the matrix to the corresponding code, click the Source tab below the Financial Report Painter.

**Example:** Viewing FML Source Code

To open the source code window, click the Source tab below the matrix.

```c
TABLE FILE CENTGL
SUM
  CENTGL.ACCOUNTS.GL_ACCOUNT_CAPTION
  CENTGL.ACCOUNTS.GL_ROLEUP_OP
  CENTGL.ACCOUNTS.GL_ACCOUNT_PARENT
  CENTGL.ACCOUNTS.GL_ACCOUNT_TYPE
FOR
  CENTGL.ACCOUNTS.GL_ACCOUNT
'2210' AS 'Retail - Television' LABEL R1
ON TABLE SET PAGE-NUM NOLOAD
ON TABLE NOTOTAL
ON TABLE PCHOLD FORMAT HTML
ON TABLE SET BLKINDENT ON
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLE *
    INCLUDE = endifit.
{
ENDSTYLE
END
```

**Tip:** While you can edit the code that appears in this window, it is recommended that you not do so. Changes to Painter-generated code may cause problems when you try to reopen the procedure in the tool.

### Previewing Your Report

You will generally work in Design mode as you build your request, but you may periodically wish to toggle to Preview mode to see the columns of the matrix that will actually be represented in the report output. For an illustration of Preview mode, see *Financial Report Painter Matrix in Preview Mode* on page 37.

**Procedure:  How to Toggle Between Design and Preview Modes**

Click the *Design mode* icon on the Financial Report Painter toolbar. The matrix display switches to Preview mode, which shows a subset of the information in the Design matrix.
Financial Report Painter Matrix in Preview Mode

In Preview mode, the Financial Report Painter displays the following:

- In the first column, a label assigned to each row in the report, such as R1 and R2 (or an explicit row label if you have supplied one).
- In the second column, default titles, or titles you assigned to rows in the report, representing TAG values of the FOR field and other row types.
- In subsequent columns, any other fields you specified for the report.

Running a Financial Report

You can run a report directly from the Financial Report Painter at any point in the Design cycle. You can also run the report from Report Painter or from the Developer Studio Explorer.

Procedure: How to Run a Financial Report

Click the Run icon on either the Financial Report Painter toolbar or the Report Painter toolbar. The report appears in the browser.
**Example: Running a Financial Report**

The following income statement started in Report Painter, and then completed in the Financial Report Painter Design matrix. You can run the statement by clicking the Run icon on the toolbar. If you want to review the matrix from which this report was generated, see *Designing Your Financial Report* on page 28.

<table>
<thead>
<tr>
<th>Caption</th>
<th>Op</th>
<th>Parent</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retail - Television</td>
<td>+</td>
<td>2200</td>
<td>R</td>
</tr>
<tr>
<td>Retail - Video Player</td>
<td>+</td>
<td>2200</td>
<td>R</td>
</tr>
</tbody>
</table>

**Creating a Financial Report Using a Cube Data Source**

You can create a financial report using a cube data source. To do this, you must add the Member Unique Name field of the hierarchy into Report Painter, and change the field type to FOR. The Member Unique Name field is typically the first field listed in the hierarchy when the Show Advanced Fields option is enabled. When you switch to the Matrix view tab, the values in that field will display hierarchically. If you do not use the Member Unique Name field, the values will not be listed hierarchically.

**Procedure: How to Create a Financial Report Using a Cube Data Source**

1. Create a new report using a cube data source.
2. In Report Painter, use the Object Inspector shortcut menu and click Show Advanced Fields. All fields within the Dimensions folder are now displayed.
3. Using the Object Inspector, add the Member Unique Name field of the hierarchy to your report.
4. Select the field and click For. The field is changed to a FOR field.
5. Click the Matrix view tab.
You can now create your financial report using data in the cube data source, as shown in the following image.

### Retrieving FOR Field Values From a Data Source

Tags identify the data values of the FOR field in your report. A report row can be associated with a tag that represents:

- A single data value of the FOR field.
- The aggregate of two or more data values of the FOR field.
- The aggregate of a range of data values of the FOR field.
- The aggregate of a related group of data values of the FOR field.

You can also identify Tag values that are organized in a parent/child hierarchy up to 99 levels deep and extending over many rows in the matrix. For details, see *Reporting Dynamically From a Hierarchy* on page 47.

The Tag button on the Financial Report Painter toolbar opens the TAG dialog box, in which you can define all of these variations.

**Procedure:** How to Specify a Tag for a Single Data Value

In the Design matrix:

1. Select the row in which you want to place a tag value.
To be able to use the same value in more than one row of the matrix, click the Use Multiple Values check box in the FML Report Properties area above the matrix.

   
or
   Right-click anywhere in the row (except on the label) and select Change Type to, and then Tag from the context menu.
   
The TAG dialog box opens at the Options tab.

3. Enter a data value, or click the Browse button and select one from the list. Your entry is reflected in the Tags box.

4. Click OK on the Options tab or select the General tab to continue.
   
The Row label (R1, and so on) appears automatically in the Label box.

5. To supply an explicit row label to replace the default label (R1, and so on) on the matrix, type it into the Label input box. The default label is retained internally.

6. Enter a row title that you want to display on the report (optional). If you do not supply a title, the title cell will be blank on the matrix and in the output.

7. Optionally, select formatting check boxes: Invisible, When Exists, Post to. For details, see Suppressing the Display of Rows on page 83 and Saving and Retrieving Intermediate Report Results on page 84.

8. Click OK.
   
The matrix shows TAG in the Row Type column and if you entered them, the explicit label in the Label column and the title in the Title column.

   Tip: You can also drag (or double-click) a tag from the FOR field values panel to a desired location on the matrix. The tag value appears in the For field cell and repeated in the title cell. Right-click in the TAG row and select Row Properties to open the TAG dialog box where you can enter a descriptive title for the tag value and/or an explicit row label.

Procedure: How to Specify a Tag for Multiple Data Values

   In the Design matrix:

   1. Select the row in which you want to place multiple tag values.
Note: To be able to use the same value in more than one row of the matrix, click the Use Multiple Values check box in the FML Report Properties area above the matrix.


or

Right-click anywhere in the row (except on the label) and select Change Type to, and then Tag from the context menu.

The TAG dialog box opens at the Options tab.

3. Enter a data value, or click the Browse button and select one from the list. Your entry is reflected in the Values box.

4. Click the Add button to add the value to the Tags box.

5. Enter or select another data value and click the Add button to add the second value to the Tags box.

If you are familiar with FML syntax, note that this entry adds the OR phrase to your request.

6. Repeat steps 4-5 for as many data values as necessary.

7. Click OK on the Options tab or select the General tab to continue.

8. Notice that the Row label (R1, and so on) appears automatically in the Label box.

To supply an explicit row label to replace the default label (R1, and so on) on the matrix, type it into the Label input box. The default label is retained internally.

9. Enter a row title that you want to display on the report (optional). If you do not supply a title, the title cell will be blank on the matrix and in the output.

10. Optionally, select a formatting check box: Invisible, When Exists, Post to. For details, see Suppressing the Display of Rows on page 83 and Saving and Retrieving Intermediate Report Results on page 84.

11. Click OK.

The matrix shows TAG in the Row Type column and, if you entered them, the explicit label in the Label column and the title in the Title column.
Tip: You can also drag (or double-click) tags from the FOR field values panel to a desired location on the matrix. Multiple tag values may be contiguous or non-contiguous. They appear on the matrix connected by the FML keyword OR. The tag values are repeated in the title cell connected by the keyword AND. Right-click in the TAG row and select Row Properties to open the TAG dialog box where you can enter a descriptive title for the combined values and/or an explicit row label.

**Procedure:** How to Specify a Tag for a Range of Data Values

In the Design matrix:

1. Select the row in which you want to place a range of tag values.

   **Note:** To be able to use the same value in more than one row of the matrix, click the Use Multiple Values check box in the FML Report Properties area above the matrix.


   or

   Right-click anywhere in the row (except on the label) and select Change Type to, and then Tag from the context menu.

   The TAG dialog box opens at the Options tab.

3. In the Value box, enter the first data value in the range, or click the Browse button and select one from the list.

4. In the To box, enter the last data value in the range, or click the Browse button and select one from the list.

5. Click the Add button to add the beginning and ending range values to the Tags box.

   If you are familiar with FML syntax, note that this entry adds the TO phrase to your request.

6. Click OK or select the General tab to continue.

7. Notice that the row label (R1, and so on) appears automatically in the Label box.

   To supply an explicit row label to replace the default label (R1, and so on) in the matrix, type it into the Label input box. The default label is retained internally.

8. Enter a row title that you want to display on the report (optional). If you do not supply a title, the title cell will be blank on the matrix and in the output.
9. Optionally, select a formatting check box: Invisible, When Exists, Post to. For details, see *Suppressing the Display of Rows* on page 83 and *Saving and Retrieving Intermediate Report Results* on page 84.

10. Click OK.

The matrix shows TAG in the Row Type column and, if you entered them, the explicit label in the Label column and the title in the Title column.

**Procedure: How to Specify a Tag for a Related Group of Data Values**

You can use the masking character ($) to identify a group of data values that are identified by common characters (for example, a group of accounts beginning with the number 10). This is useful for specifying a whole group of tag values without having to name each one.

1. Select the row in which you want to place a group of tag values.

   **Note:** To be able to use the same value in more than one row of the matrix, click the *Use Multiple Values* check box in the FML Report Properties area above the matrix.


   or

   Right-click anywhere in the row (except on the label) and select *Change Type to*, and then *Tag* from the context menu.

   The TAG dialog box opens at the Options tab.

3. In the Value box, enter the common characters for the group of tags, followed by one or more masking characters ($$). For example, 10$$ to represent 1010, 1011, 1020, 1021, 1030, 1031.

4. Click the Add button to add the group of tag values to the Tags box.

5. Click OK or select the General tab to continue.

6. Notice that the row label (R1, and so on) appears automatically in the Label box.

   To supply an explicit row label to replace the default label (R1, and so on) in the matrix, type it into the Label input box. The default label is retained internally.

7. Enter a row title that you want to display on the report (optional). If you do not supply a title, the title cell will be blank on the matrix and in the output.

8. Optionally, you can select a formatting check box: Invisible, When Exists, Post to. For details, see *Suppressing the Display of Rows* on page 83 and *Saving and Retrieving Intermediate Report Results* on page 84.
9. Click OK.

The matrix shows TAG in the Row Type column and, if you entered them, the explicit label in the Label column and the title in the Title column. If you do not specify a title, the tag value and masking characters appear.

Reference: TAG Dialog Box

Options Tab

Value

Identifies one or more data values for the TAG row.

If you are identifying a range of values, this field identifies the first data value in the range.

If you are using masking characters to specify a group of tag values with common identifying characters, enter the common and masking characters ($$) in the Values box.
To (optional)
Identifies the last data value in the range.

Tags
Displays the value(s) you have selected for the current TAG row.

Children
The following options appear on the TAG dialog box Options tab only when a parent/child relationship has been defined in the Master File being used for a report. In addition, an optional attribute in the Master File enables you to specify another field, which contains a descriptive caption for display in place of the hierarchy field values on the report. For related information, see Reporting Dynamically From a Hierarchy on page 47.

When you write a financial report against a hierarchy defined in the Master File, you can control data retrieval from the following options on a drop-down list:

Show selected item
Retrieves the value for the specified tag in the hierarchy.

Show selected item consolidated
Displays a single summary row for the numeric data values of all children.

Show only children
Displays the hierarchy starting with the first child of the specified parent, up to 99 levels deep. Each child instance appears over the next child instance. The parent level is not included in the output.

Successive levels of the hierarchy field are indented two spaces from the previous level. You can change the spacing of these indents. See How to Adjust Indents for Captions in a Hierarchy on page 51.

Show only children to level... Level: n
Displays children up to the level you indicate. The default value is 1. Therefore, if n is omitted, direct children appear. Level 2 retrieves direct children and grandchildren. The parent level is not included in the output.

Show with all children
Displays the hierarchy starting with the specified parent. (Level 99 is equivalent to ALL.) The parent level is included in the output.
Show with children to level... Level: n

Displays the values for the specified parent tag and its children, up to 99 levels deep. The default value is 1. Therefore, if n is omitted, direct children appear. The parent level is included in the output.

You can refine your display further by choosing the following check box:

Consolidate

You can display a summary row for children up to the level indicated, displayed below the detail rows for the children being summed. Consolidate is supported with the Show children and Show with children options.

Display children’s caption

You can display descriptive text defined in the Master File as a caption, in place of the FOR field values.

General Tab

Title

Enables you to specify a title for the TAG row. The title will appear on the report. This entry is optional. (Without it, the title cell will be blank in the matrix and on the report.)

For hierarchical data, the caption defined in the Master File is used for the title. The caption supplies descriptive text that can be used in place of the hierarchy field FOR values.

- If you add a parent to the matrix and specify Show selected item, the caption is applied. This value appears in the Title input line, where you can edit it.
- If you add a parent to the matrix and specify Show selected item, the caption is applied. This value appears in the Title input line, where you can edit it.
- If you add a parent and specify an option that also retrieves its children, captions for the children appear in the Title column. However, the cells are gray, indicating that you cannot change these titles. (This happens whether the Display caption for children is on or off.)
- If no caption is specified in the Master File, and you drag a tag value into the matrix, the tag value appears as the title. However, you can provide a descriptive title for the row in the Title column on the matrix or in the Title input box on the General tab.
If you do not want the caption to appear, uncheck the Display children’s captions box.

**Label**

Lists the default identification label (R1, R2, and so on) for the TAG row. You can replace the default row identification label with an explicit row label. (The default label is retained internally.)

In a hierarchy, each generated row is labeled with the specified label text. Children have the same label as their parents.

**Formatting Options:**

The following options apply to individual rows in the report.

**Invisible**

Creates a TAG row for use in a calculation, but does not display the row on the report. For details, see *How to Suppress Rows* on page 83.

**When Exists**

Displays a row only when data exists for the tag value. By default, a TAG row appears even if no data is found for the tag values. The default character used to represent the missing data is a period (.). The When Exists check box enables you to override this convention. For details, see *How to Suppress Rows With No Data* on page 84.

**Post to**

Posts the output of a TAG row to a work file. This row can then be used as if were provided in a DATA row. For details, see *Saving and Retrieving Intermediate Report Results* on page 84.

**Tip:** You can also apply these formatting options from the Row Properties check boxes above the matrix.

**Reporting Dynamically From a Hierarchy**

Hierarchical relationships between fields can be defined in a Master File and automatically displayed using the Financial Report Painter. The parent and child fields must share data values and their relationship should be hierarchical. For example, suppose that:

- Employee and manager IDs are contained within an employee data source.

  or
A general ledger data source contains both an account number field and an account parent field.

By examining these fields, it is possible to construct the entire organization chart or charts of accounts structure. However, to display the chart in a traditional row-based financial report, you would have to list the employee IDs or account numbers in the request in the order in which they should appear on the report. If an employee or account is added, removed, or transferred, you would have to change the report request to reflect this change in organizational structure.

With FML hierarchies, you can define the hierarchical relationship between two fields in the Master File and load this information into memory. The report request can then *dynamically* construct the rows that represent this relationship and display them in the report, starting at any point in the hierarchy.

**Tip:** You can construct the Master File using a text editor or the Synonym Editor. For related information, see *Requirements for FML Hierarchies* on page 217 and *Describing Data for an FML Hierarchy* on page 263.

In the Financial Report Painter, the hierarchy defined in the Master File is reflected in the FOR field values panel to the right of the Design matrix, as shown in the following illustration. If the hierarchy fields are defined with captions in the Master File, the resulting report indents the captions proportionate to their levels in the hierarchy.
For details about how the parent/child hierarchy is represented in the Master File, see *Describing Data for an FML Hierarchy* on page 263.

**Note:** The ability to define a hierarchy in a Master File is particularly useful when you are working with a cube data structure, such as ESSBASE. However, it can be employed with other data source types.

**Procedure:**  **How to Specify Tags for Data Values in a Hierarchy**

In the Design matrix:

1. Select the row in which you want to place tag values that have been organized in a parent/child hierarchy. The hierarchy is reflected in the FOR field values panel to the right of the matrix.

   **Note:** To be able to use the same value in more than one row of the matrix, click the *Use Multiple Values* check box in the FML Report Properties area above the matrix.

2. Click the *Tag* icon on the Financial Report Painter toolbar.

   or

   Right-click anywhere in the row (except on the label) and select *Change Type to*, and then *Tag* from the context menu.

   The TAG dialog box opens at the Options tab. (Note that when a hierarchy has been defined in the Master File against which you are reporting, a group of specialized options, described in steps 6-8, are included in the TAG dialog box.)

3. In the Value box, enter a value from the hierarchy tree, or click the *Browse* button and select one from the list.

4. Click the *Add* button to add the selected value to the Tags box.

5. Select the value in the Tags box to activate the hierarchy fields.

6. In the Children box, indicate what level of data you want to retrieve or consolidate in the current row of the matrix:

   **Show selected item consolidated**

   Displays a single summary row for the numeric data values of all children.
**Show only children**

Displays the hierarchy starting with the first child of the specified parent, up to 99 levels deep. Each child instance appears over the next child instance. The parent level is not included in the output.

Successive levels of the hierarchy field are indented two spaces from the previous level. You can change the spacing of these indents. For more information, see *How to Adjust Indents for Captions in a Hierarchy* on page 51.

**Show only children to level... Level: n**

Displays children up to the level you indicate. The default value is 1. Therefore, if \( n \) is omitted, direct children appear. Level 2 retrieves direct children and grandchildren. The parent level is not included in the output.

**Show with all children**

Displays the hierarchy starting with the specified parent. (Level 99 is equivalent to ALL.) The parent level is included in the output.

**Show with children to level... Level: n**

Displays the values for the specified parent tag and its children, up to 99 levels deep. The default value is 1. Therefore, if \( n \) is omitted, direct children appear. The parent level is included in the output.

7. By default, the *Display children's caption* box is checked. This enables you to display descriptive text, defined in the Master File, in place of the hierarchy field FOR values. If you do not want the caption to appear, uncheck the *Display children's captions* box.

- If you add a parent to the matrix and specify *Show selected item*, the caption is applied. This value appears in the Title input line, where you can edit it.

- If you add a parent and specify an option that also retrieves its children, captions for the children appear in the Title column. However, the cells are gray, indicating that you cannot change these titles. (This happens whether *Display children's caption* is on or off.)

- If no caption is specified in the Master File, the tag value appears as the title. However, you can provide a descriptive title for the row directly in the Title column on the matrix or in the Title input box on the General tab.

8. If you chose an option that retrieves children in step 6, you can check the *Consolidate* box on the Options tabs to display a summary row for children up to the level indicated, immediately following the detail rows for the children being summed.

9. Click *OK* or select the *General* tab to continue.
10. Row label (R1, and so on) will automatically appear in the Label box.

To supply an explicit row label to replace the default label (R1, and so on) in the matrix, type it into the Label input box. (The default label is retained internally.)

The same label applies to the parent level and all child levels.

11. Optionally, you can select a formatting check box from the General tab: The formatting check box options are:

- Invisible
- When Exists
- Post to

For details, see *Suppressing the Display of Rows* on page 83 and *Saving and Retrieving Intermediate Report Results* on page 84.

12. Click **OK**.

The matrix shows TAG in the Row Type column for the parent values. Each parent value for children that has been retrieved is preceded by a plus sign (+) that you can click to display the children. The plus sign (+) becomes a minus sign (-) that you can click to roll up the child values.

If you entered an explicit label in the Label input box, this element will appear in the matrix.

If you selected the *Use Multiple Values* check box in the FML Report Properties area above the matrix, the used value remains available in the FOR field values panel. It appears in red to indicate that it has already been added to the matrix.

**Tip:** You can also drag (or double-click) a tag from the FOR field values panel to a desired location on the matrix, then right-click in the TAG row and select *Row Properties* to open the TAG dialog box.

**Procedure:** **How to Adjust Indents for Captions in a Hierarchy**

To clarify relationships within a hierarchy, the captions (titles) of values are indented at each level. To adjust the indents:

1. Choose **Indent** from the Edit menu. The Indent dialog box opens.

   If a hierarchy has not been defined in the Master File for the data you are reporting against, the Indent option does not appear.
2. Choose one of the following option buttons:

- **Use default value of indent.** Specifies how many spaces to indent child entries.
- **Turn indent off.** This option left-justifies titles for values at all levels of the hierarchy.
- **Select a value of indent,** then enter an explicit measurement to represent the number of spaces to be indented based on the number of blank spaces preceding the caption text in the Master File and the unit of measurement defined in the Report Options Output tab (inches, centimeters, or points). For example, suppose that the caption text is preceded by two blanks and the base measurement is inches. If you enter .2, each level of values in the hierarchy will be indented .2 inches for each leading space (.2” x 2) from the previous level.

**Displaying an FML Hierarchy**

When reporting from an FML hierarchy, you can dynamically retrieve and display hierarchical data using two variations:

- **Show only children** displays only the children, not the parent value for those children.
- **Show with children** displays the parent and then the children.

In either case, you can show all children for the specified parent, or children to a specified level in the hierarchy, up to 99 levels deep. The default hierarchy depth is one level. To highlight these relationships visually, successive levels of the hierarchy field are indented two spaces from the previous level. (You can change the indentations. See *How to Adjust Indents for Captions in a Hierarchy* on page 51.)

The hierarchy appears sorted by the parent field and, within parent, sorted by the hierarchy field.

When displaying a hierarchy, you can show either detail or summary data for fields other than the For field in the request. You can also replace the For field value with a descriptive caption.

**Example:** **Displaying an FML Hierarchy With Captions**

The following example displays two levels of a charts of accounts hierarchy and shows descriptive captions defined in the Master File, instead of the account numbers, in the report.

1. Create a report procedure in Developer Studio and name the file `fmhiercaps`.

2. When prompted, choose `Report Painter` as your reporting tool (it is the default) and choose `CENTGL` as your data source. To examine this data, see *Sample Data for FML Hierarchy Examples* on page 68.

   Report Painter opens with CENTGL fields listed in the Object Inspector panel.
3. Drag the field GL_ACCOUNT_PARENT into the Report Painter window. The column title associated with this field is Parent. Parent is a Detail field by default.

4. Drag GL_ACCOUNT into the Report Painter window. The column title associated with GL_ACCOUNT is Ledger Account. Select Ledger Account and click the For button. (This will become the controlling field in your financial report.)

5. Click the Matrix tab (located toward the bottom of the Report Painter window) to open the Design matrix. Note that GL_ACCOUNT (the For field) is the title of the second column and its values appear in the For field values panel at the right of the matrix. You will be populating the matrix with these values.

6. Begin by dragging the tag 1000 from the FOR field values panel into row R1 of the matrix. The 1000 account tag appears in the GL_ACCOUNT column.

7. In row R1, right-click 1000 and select Row Properties from the menu. The TAG dialog box opens, with 1000 listed in the Tags box.

8. In the Children box, choose Show with children to level from the drop-down list, then select 2 in the Level box to display two levels of the hierarchy, with account 1000 as the starting point (or parent level). The Display children’s caption check box is selected by default. This will show the descriptive titles for the children, rather than their tag values (1000, 2000), on the report.
The TAG dialog box appears, as shown in the following image.

9. Click the plus (+) sign next to 1000 to expand the hierarchy one level.
The matrix appears, as shown in the following image.

If you want to expand the hierarchy another level, repeat the process.

10. To add some quick styling, click Report Options from the toolbar above the matrix. The Report Options dialog box opens.

- On the Style tab, click the Style File Selection button. The StyleSheet Selection dialog box opens.

- Click the Add new item button and select deflt1.sty from the StyleSheet File Selection dialog box.

- Click OK. The selected StyleSheet is added to the Include StyleSheet File section.

- Click Finish to close the StyleSheet Selection dialog box and click OK to close the Report Options dialog box.

The StyleSheet is applied to the report.
11. Click the Run icon on the toolbar to see the report, which lists the account numbers associated with the levels of the parent/child hierarchy. The indents for the hierarchy levels are set by default.
Tip: If you want to see the children in the hierarchy without the parent, choose *Show only children to level 2* in step 8, rather than *Show with children to level 2*. Without the parent line, the report would look like the following image.

<table>
<thead>
<tr>
<th>Gross Margin</th>
<th>Parent</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1000</td>
</tr>
<tr>
<td>Sales Revenue</td>
<td>2000</td>
</tr>
<tr>
<td>Retail Sales</td>
<td>2100</td>
</tr>
<tr>
<td>Retail · Television</td>
<td>2200</td>
</tr>
<tr>
<td>Retail · Stereo</td>
<td>2200</td>
</tr>
<tr>
<td>Retail · Video Player</td>
<td>2200</td>
</tr>
<tr>
<td>Retail · Computer</td>
<td>2200</td>
</tr>
<tr>
<td>Retail · Video Camera</td>
<td>2200</td>
</tr>
<tr>
<td>Mail Order Sales</td>
<td>2100</td>
</tr>
<tr>
<td>Mail Order · Television</td>
<td>2300</td>
</tr>
<tr>
<td>Mail Order · Stereo</td>
<td>2300</td>
</tr>
<tr>
<td>Mail Order · Video Player</td>
<td>2300</td>
</tr>
<tr>
<td>Mail Order · Computer</td>
<td>2300</td>
</tr>
<tr>
<td>Mail Order · Video Camera</td>
<td>2300</td>
</tr>
<tr>
<td>Internet Sales</td>
<td>2100</td>
</tr>
<tr>
<td>Internet · Television</td>
<td>2400</td>
</tr>
<tr>
<td>Internet · Stereo</td>
<td>2400</td>
</tr>
<tr>
<td>Internet · Video Player</td>
<td>2400</td>
</tr>
<tr>
<td>Internet · Computer</td>
<td>2400</td>
</tr>
<tr>
<td>Internet · Video Camera</td>
<td>2400</td>
</tr>
<tr>
<td>Cost Of Goods Sold</td>
<td>2000</td>
</tr>
<tr>
<td>Variable Material Costs</td>
<td>2500</td>
</tr>
<tr>
<td>Television COGS</td>
<td>2600</td>
</tr>
<tr>
<td>Stereo COGS</td>
<td>2600</td>
</tr>
<tr>
<td>Video COGS</td>
<td>2600</td>
</tr>
<tr>
<td>Computer COGS</td>
<td>2600</td>
</tr>
<tr>
<td>Video Camera COGS</td>
<td>2600</td>
</tr>
</tbody>
</table>

**Consolidating an FML Hierarchy**

The Consolidate option consolidates multiple levels of the hierarchy on one line of the report output. Consolidate can be used alone or in conjunction with the Show only children or Show with children options. Consolidation is designed to work with requests that use the Sum option for fields other than the FOR field. It is also designed to be used with detail level financial data, not data that is already consolidated.

- When used alone, the Consolidate option aggregates the parent and children on one line of the report output, summing the numeric data values included on the line.
When used in conjunction with Show only children, the Consolidate option displays one line for each child of the specified parent value. Each line is a summation of that child and all of its children. You can specify the number of levels of children to display (which determines the number of lines generated on the report output) and the depth of summation under each child. By default, only direct children will have a line in the report output and the summary for each child will include all of its children.

When used in conjunction with Show with children, the Consolidate option first displays a line in the report output that consists of the summation of the parent value and all of its children. Then it displays additional lines identical to those displayed by Show only children plus Consolidate.

In order to use a data record in more than one line of a financial report (for example, to display both detail and summary lines or to consolidate detail data at multiple levels), select the option Use Multiple Values before you begin to populate the matrix.

**Example:** Displaying One Summary Line for an FML Hierarchy

**Data Detour:** For this example, you will use two data sources: CENTGL and CENTSYSF. CENTSYSF contains detail level financial data. CENTGL defines the account hierarchy. To use the financial data with the account hierarchy, you must join the two data sources.

Before you complete this example, examine the data you will be using. While you can follow the flow of this example without looking at the data, doing so will help you understand the process and the output more fully. See *Sample Data for FML Hierarchy Examples* on page 68.

1. In Developer Studio, create a procedure and name the file *fmlhiersumrow*.
2. When prompted, choose Report Painter as your reporting tool (it is the default) and choose CENTGL as your host data source. The component connector window opens briefly, followed by the Report Painter, where CENTGL fields are listed in the Object Inspector panel.
3. Click the Join button from the Setup toolbar. The Join tool opens, listing the fields from CENTGL.MAS in a resizable window. Click the Add File (+) icon on the toolbar. When prompted, specify CENTSYSF as the cross-reference file. The two data sources are automatically joined at their respective SYS_ACCOUNT fields.
4. Run the Join and close the tool. The fields from CENTGL and CENTSYSF are listed in the Object Inspector.
5. Drag GL_ACCOUNT (Ledger Account), NAT_AMOUNT (Month Actual), and NAT_YTDAMT (YTD Actual) into the Report Painter window.
6. Select Month Actual and click the Sum button. YTD Actual automatically becomes a Sum field. Select Ledger Account and click the For button. (The For field will become the controlling field in your financial report.)
7. Click the Matrix tab below Report Painter to open the Design matrix. Note that GL_ACCOUNT (the FOR field) is the title of the second column and its values appear in the For field values panel at the right of the matrix. You will be populating the matrix with these values.

8. First, click the Use Multiple Values check box above the matrix. This will enable you to use the same For field value more than once, to present both the detail data and the consolidation.


   The TAG dialog box opens, with 3100 in the Tags list.

10. In the Children box, select Show with all children and accept the default to Display children’s caption.

    This will generate detail data for the parent value (3100) and all of its children, represented on the report by their descriptive captions rather than their account numbers.

![TAG dialog box]

11. To add an underline below the detailed numeric data, right-click in row R2, select Change Row type to, and click the option button for Bar. The BAR dialog box opens. Accept the default underline character.
12. Drag 3100 into the matrix again. Right-click and choose Row Properties. The TAG dialog box opens. This time, in the Children box, select *Show selected item consolidated* to generate one summary line for Selling Expenses and all of its children.
The matrix appears as shown in the following image.

13. Click the plus (+) sign next to 3100 in row R1 to expand the hierarchy.
Notice that Advertising has its own children, while the other children of Selling Expenses do not. Notice also that the consolidated row, R3, has no plus sign (+) or hierarchy to expand.

14. To add some quick styling, select **Styling** from the Report menu. The Report Options dialog box opens at the Style tab.

- Click the **Style File Selection** button. The StyleSheet Selection dialog box opens.
- Click the **Add new item** button and select *deflt1.sty* from the StyleSheet File Selection dialog box.
- Click **Ok**.

The selected StyleSheet is added to the Include StyleSheet File section.

- Click **Finish** to close the StyleSheet Selection dialog box and click **OK** to close the Report Options dialog box.

The StyleSheet is applied to the report.

15. Click the **Run** icon in the toolbar above the matrix to generate the following output.

![Matrix Output](image)

Note that only accounts with no children are populated in the detail level data source (CENTSYF), therefore, no values appear for Selling Expenses and Advertising. The consolidation row accurately sums the detailed data.
Tip: To display the sum of just the children, you must display the parent row, display the summary row, and use a RECAP to subtract the parent row from the sum. For example:

\[ \text{RECAP CHILDSUM} = R2-R1; \]

**Procedure: How to Consolidate FML Hierarchy Data**

The following procedure is an example of how to consolidate FML hierarchy data.

1. Create a procedure called `fmlconsolidate` in Developer Studio, following steps 1-8 in the example, *Displaying One Summary Line for an FML Hierarchy* on page 58. Note that this example uses the data described in *Sample Data for FML Hierarchy Examples* on page 68.

   **Tip:** Rather than starting from scratch, you can open the matrix created in *Displaying One Summary Line for an FML Hierarchy* on page 58, delete the content of the matrix, and then complete the steps that follow.

2. You should be looking at an empty matrix, with *Use Multiple Value* checked.

   Drag 3100 onto the matrix in row R1. Right-click on 3100 and select *Row Properties*. The TAG dialog box opens, with 3100 in the Tags list.

3. In the Children box, select *Show with all children* and accept the default to *Display children’s captions*. This will generate detail data for the hierarchy, starting with account 3100. Descriptive captions, rather than account numbers, will display in the output.

4. To separate the detailed output in the top section of the report from the consolidated output in the bottom section, you will add a blank row. Right-click in row R2 and select *Change Row Type to Text*. The TEXT dialog box opens. Leave the input box blank to specify a row with no content and click *OK*.

5. To underline the numeric columns, right-click in row R3, and select *Change Row Type to Bar*. Accept the default underline character and click *OK*.

6. Repeat step 4 to make row R4 a blank line.
The matrix appears, as shown in the following image.

7. To create a consolidated line for the parent account (3100) and each direct child, drag account 3100 into row R5, right-click 3100, and select Row Properties. In the TAG dialog box, select Show with all children and click the Consolidate check box.
The matrix appears, as shown in the following image.

8. Before you run the report, limit the data to be retrieved to the period from 2002-2003. To do this, click the Report menu at the top of the screen and choose the *Where/If* tab. The Expression Builder opens. Using any method you choose, create the following expression:

\[
\text{PERIOD EQ '2002/03'}
\]

9. Click *OK*.

10. Click the *Run* icon.
The output looks like the following image.

<table>
<thead>
<tr>
<th></th>
<th>Month Actual</th>
<th>YTD Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selling Expenses</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Advertising</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TV/Radio</td>
<td>1,049,146</td>
<td>2,954,342</td>
</tr>
<tr>
<td>Print Media</td>
<td>244,589</td>
<td>721,448</td>
</tr>
<tr>
<td>Internet Advertising</td>
<td>9,542</td>
<td>29,578</td>
</tr>
<tr>
<td>Promotional Expenses</td>
<td>53,719</td>
<td>151,732</td>
</tr>
<tr>
<td>Joint Marketing</td>
<td>97,135</td>
<td>289,799</td>
</tr>
<tr>
<td>Bonuses/Commissions</td>
<td>100,188</td>
<td>304,199</td>
</tr>
</tbody>
</table>

The top portion shows the detail level data.

The bottom portion shows the consolidated data. In the consolidated portion of the report:

- There is one line for the parent that is the sum of itself, plus all of its children to all levels.

- There is one line for each direct child of account 3100 (Selling Expenses): Advertising, Promotional Expenses, Joint Marketing, and Bonuses/Commissions.

- The line for Advertising is the sum of itself plus all of its children. If it had multiple levels of children, they would all have been added into the sum. The other direct children of 3100 did not themselves have children, so the sum on each of those lines consists of only the parent value.
Tip: If you wanted to see the children in the hierarchy without the parent, you could choose *Show only children*, rather than *Show with children*. Without the parent line, the report would output as shown in the following image.

<table>
<thead>
<tr>
<th></th>
<th>Month Actual</th>
<th>YTD Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selling Expenses</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Advertising</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TV/Radio</td>
<td>1,049,146.</td>
<td>2,954,342.</td>
</tr>
<tr>
<td>Print Media</td>
<td>244,589.</td>
<td>721,448.</td>
</tr>
<tr>
<td>Internet Advertising</td>
<td>9,542.</td>
<td>29,578.</td>
</tr>
<tr>
<td>Promotional Expenses</td>
<td>53,719.</td>
<td>151,732.</td>
</tr>
<tr>
<td>Joint Marketing</td>
<td>97,135.</td>
<td>289,799.</td>
</tr>
<tr>
<td>Bonuses/Commissions</td>
<td>100,188.</td>
<td>304,199.</td>
</tr>
</tbody>
</table>

What makes this reporting dynamic?

Suppose that you run this report weekly. If you ran the report today, and tomorrow another account were added as a child of 3100, the report for next week would automatically reflect the change in the hierarchy, as well as changes to the data, based on the versatile parent/child declaration in the Master File. No adjustment would be needed to keep either the Master File or the report request up to date.
**Reference:** Sample Data for FML Hierarchy Examples

The CENTGL Master File contains a charts of accounts hierarchy. The field GL_ACCOUNT_PARENT is the parent field in the hierarchy. The field GL_ACCOUNT is the hierarchy field. The field GL_ACCOUNT_CAPTION can be used as the descriptive caption for the hierarchy field.

```plaintext
FILE=CENTGL      , SUFFIX=FOC
SEGNAME=ACCOUNTS , SEGTYPE=S01
FIELDNAME=GL_ACCOUNT, ALIAS=GLACCT, FORMAT=A7,
            TITLE='Ledger,Account', FIELDTYPE=I, $
FIELDNAME=GL_ACCOUNT_PARENT, ALIAS=GLPAR, FORMAT=A7,
            TITLE=Parent,
            PROPERTY=PARENT_OF, REFERENCE=GL_ACCOUNT, $
FIELDNAME=GL_ACCOUNT_TYPE, ALIAS=GLTYPE, FORMAT=A1,
            TITLE=Type,$
FIELDNAME=GL_ROLLUP_OP, ALIAS=GLROLL, FORMAT=A1,
            TITLE=Op, $
FIELDNAME=GL_ACCOUNT_LEVEL, ALIAS=GLLEVEL, FORMAT=I3,
            TITLE=Lev, $
FIELDNAME=GL_ACCOUNT_CAPTION, ALIAS=GLCAP, FORMAT=A30,
            TITLE=Caption,
            PROPERTY=CAPTION, REFERENCE=GL_ACCOUNT, $
FIELDNAME=SYS_ACCOUNT, ALIAS=ALINE, FORMAT=A6,
            TITLE='System,Account,Line', MISSING=ON, $
```

The CENTSYSF data source contains detail-level financial data. This is unconsolidated financial data for a fictional company, CenturyCorp. It is designed to be separate from the CENTGL database as if it came from an external accounting system. It uses a different account line system (SYS_ACCOUNT) which can be joined to the SYS_ACCOUNT field in CENTGL. Data uses natural signs (expenses are positive, revenue is negative).

```plaintext
FILE=CENTSYSF     , SUFFIX=FOC
SEGNAME=RAWDATA   , SEGTYPE=S2
FIELDNAME = SYS_ACCOUNT , , A6       , FIELDTYPE=I,
            TITLE='System,Account,Line', $
FIELDNAME = PERIOD , , YYM      , FIELDTYPE=I,$
FIELDNAME = NAT_AMOUNT , , D10.0    , TITLE='Month,Actual', $
FIELDNAME = NAT_BUDGET , , D10.0    , TITLE='Month,Budget', $
FIELDNAME = NAT_YTDAMT , , D12.0    , TITLE='YTD,Actual', $
FIELDNAME = NAT_YTDBUD , , D12.0    , TITLE='YTD,Budget', $
```

You can create an FML hierarchy in a Master File using a text editor or the Synonym Editor. For details, see *Describing Data for an FML Hierarchy* on page 263.
Supplying Data Directly

In certain cases, you may need to include some additional constants (for example, exchange rates, inflation rates) in your model. Not all data values for the model have to be retrieved from the data source. Using the Financial Report Painter, you can supply data directly in the request. For related information, see *Saving and Retrieving Intermediate Report Results* on page 84.

Procedure: How to Add Data Values Directly to the Report

   
or
   Right-click anywhere in the row (except on the label) and select *Data* from the context menu. The DATA dialog box opens at the General tab.

2. In the General tab, you can:
   - Type a row title.
   - Supply an explicit row label to replace the default label on the matrix. (The default label is retained internally.)
   - Select formatting options: Invisible and/or Post to. See *DATA Dialog Box* on page 70. These entries are optional.

3. Click OK to record your entries.

4. DATA appears in the Row Type column. Type the data values (constants) in the appropriate cells in the matrix.
Reference:  DATA Dialog Box

General Tab

Title

Enables you to specify a title for the DATA row in which the title will appear on the report. This entry is optional.

Label

Lists the default label for the DATA row. You can replace the default row identification label with an explicit row label. The default label is retained internally.

Formatting Options:

The following options apply to individual rows in the report.

Invisible

 Creates a DATA row for use in a calculation, but does not display the row on the report. For details, see Suppressing the Display of Rows on page 83.

Post To

Posts the output of a DATA row to a work file. This data can then be used as though it were provided in a DATA row. For details, see Saving and Retrieving Intermediate Report Results on page 84.
Performing Inter-Row Calculations

The Recap operation in the Financial Report Painter enables you to perform calculations on data in the rows of a report to produce new rows. Since these calculations are performed on rows, each row referenced in the calculation must be uniquely identified either by the default label assigned by Financial Report Painter or a label you assign to the row.

To initiate a Recap operation, you must supply the format of the value that will receive the result of the calculation, and an expression that defines the calculation you wish to perform. Initially, the default row label (R1, and so on) serves as the identifying label for the calculated value. However, it is good practice to provide a more descriptive label for reference in other calculations.

In the Financial Report Painter, you can type an expression directly into the RECAP dialog box or use a tabbed dialog box to formulate your expression.

Note: If you want to create a Recap row that summarizes the data in existing rows and does not require an expression, you can use a simple alternative procedure. For details, see How to Create a RECAP Summary Row on page 81.

Procedure: How to Create a Recap Expression in Financial Report Painter

In the Design matrix:

1. Select the row in which you want to perform the calculation.

   
or

   Right-click anywhere in a blank row (except on the label) and select Change Type to, and then Recap from the context menu.

   The RECAP dialog box opens at the Options tab.

3. You can perform a Recap for a single column, a range of columns, columns at specified intervals, or for all columns in a row. To perform a Recap calculation and display output for:

   - A single column. Select a column number from the Columns drop-down list.
     
   - A range of columns. Select the first column number from the Columns drop-down list and the number appears in the From Column box. Enter the last column number in the To Column box.
Performing Inter-Row Calculations

- **Columns with an incremental interval.** Select the first column number from the Columns drop-down list and the number appears in the From Column box. Enter the last column number in the To Column box, then enter an increment number. For example, enter 2 for every other column, 3 for every third column, and so on.

- **All columns in a row.** Click *Apply to row.* (The Columns box is disabled.)

4. Assign a format to the Recap output field. You can either enter the format in the Format box or click the *Format* button to open the Format dialog box. For details, see *Format Dialog Box* on page 78.

5. Type a Recap expression in the input box, and go to step 10.

   or

   Click the *Advanced* button to expand the Recap Options tab, exposing features that will assist you in building an expression that may be composed of numbers and operators (numeric, alphanumeric, Boolean, and conditional), built-in functions, sort fields, and labels. The expression you create appears on the Options tab box as you make your selections. This process is described in steps 6 through 9.

6. Use the Recap calculator to compose the expression that WebFOCUS will evaluate. Click numbers plus the operators you need to create numeric, alphanumeric, date, logical, and conditional expressions.

   For details about writing expressions, see the *Creating Reporting Applications With Developer Studio* manual.

   You can also add parenthesis, single quotation marks, and designate uppercase and lowercase as required by clicking the buttons below the Expression window.

7. To include a built-in function in the expression, select the function from the drop-down list. The function is added to the Expression window, along with placeholders for the required arguments. Type appropriate values over the argument placeholders, or use the tools provided to select the values you need.

   For details, see the *Using Functions* manual.

8. To include a row label in your expression, click the label in the Labels drop-down list. (Note that the list shows default row labels unless you have defined explicit labels.)

9. To include a vertical sort (BY) field in your expression, click the field in the By Fields drop-down list.
Tip: You can use a sort field value in a RECAP command to allow the model to take different actions within each major sort break. For example, in a request sorted by REGION, the following calculation would compute a non-zero value only for the EAST region:

```
IF REGION EQ 'EAST' THEN .25*CASH ELSE 0;
```

10. Click the General tab and type the title for the Recap calculation that you want to display on the report. Though optional, this entry is generally included in financial reports.

11. Notice that the row label (R1, and so on.) appears automatically in the Label box of the General tab. This label is used to reference the result of the Recap calculation. To supply an explicit row label to replace the default label on the matrix, type it into the Label box. (The default label is retained internally.) Though optional, this entry is generally included in financial reports.

12. Click OK to complete the expression.

RECAP appears in the Row Type column of the matrix, along with the expression, label, and title you define.

Reference: RECAP Dialog Box
Options Tab

Several options in this dialog box enable you to specify the column or columns to which you want to apply a Recap formula:

Column

Identifies a single column that will contain the Recap formula. This is also the column in which the Recap output will appear on the report. You can choose the column from the drop-down list.

Apply to row

Indicates that the same Recap formula is to be used for all columns in the row. The Columns box is disabled when you select Apply to row.

From Column/To Column

Defines a range of columns to which the Recap formula is to be applied.

Increment

Indicates an interval of columns in the range that should have the Recap formula applied. For example, a value of 2 will apply the formula to every other column.

Note: Although not directly supported by options in the RECAP dialog box, there are a number of other ways in which you can identify the columns to which you want to apply the Recap calculation: column address, column value, and cell notation. To use these features, you must type the required syntax directly into the expression box in the RECAP dialog box.

For details, see Creating Financial Reports With Financial Modeling Language (FML) on page 177.

Format box

Identifies the field type, field length, and display options that will be used for the Recap output. The field type can be alphanumeric, numeric, or date.

Format button

Opens the Format dialog box, which is designed to help you choose the format for your Recap output. See Format Dialog Box on page 78.

Expression box

Displays the expression used to generate the Recap output.
Type the expression, or click the Advanced button to expand the Options tab, exposing features that will help you create the expression.

**Advanced button**

Expands the Options tab to make it easy to add functions, numbers, operators, labels, and sort fields to your expression. Your selections are reflected in the Expression box.

**Calculator**

Provides numbers and operators that you can use to create numeric, alphanumeric, Boolean, and conditional expressions. Click the desired number or operator to add it to the formula in the Expression box.

Four additional buttons below the Expression box enable you to:

- Enclose a value in parentheses. Click the ( ) button to add parenthesis.
  
  Parentheses affect the order in which WebFOCUS performs the specified operations.
  
  For information on when to use parentheses in expressions, see the Creating Reporting Applications With Developer Studio manual.

- Enclose alphanumeric or date literals in single quotation marks. Click the ‘ ’ button to add quotation marks.

- Convert entries in the expression box to upper or lowercase. Click the Upper or Lower button to specify case. Note that field names are case-sensitive.

**Labels**

Lists the labels (default or explicitly defined) that you can use in your expression. Select the label you want to add from the drop-down list. Your selection is added to the formula.
By Fields

Lists the vertical sort (BY) fields that you can use in your expression. Select the fields you want to add from the drop-down list. Your selection is added to the formula.

Functions

Lists all available built-in functions that you can add to your expression. A function is a program that returns a value.

Select a function from the drop-down list. Then, in the Expression box, highlight each argument and substitute the value or field name you wish to use by typing or by selecting items from other drop-down lists. For details, see the Using Functions manual.

General Tab

Title

Enables you to specify a title for the Recap calculation in which this title will appear on the report.

Label

Lists the default label for the Recap row. You can replace the default row identification label with an explicit row label that you want to reference on the left hand side of an expression. You can think of this label as a calculated value or as a variable that holds the result of the Recap calculation. It can also be used as part of the expression in subsequent Recap formulas.
Formatting Options

The following options apply to individual rows in the report. You can also apply these formatting options from the Row Properties check boxes above the matrix.

Invisible

Creates a Recap expression for use in a calculation, but does not display its results on the report.

Post To

Posts the output of a Recap line to a work file. This line can then be used as though it were provided in a DATA row. For details, see Saving and Retrieving Intermediate Report Results on page 84.
Reference: Format Dialog Box

The Format dialog box enables you to define the format of the output to be generated by a calculation.

Note: The Format button is available in the Compute, Define, and RECAP dialog boxes to define field formats for temporary fields and subtotal calculations. Click the Format button to open the Format dialog box.

![Format Dialog Box](image-url)
The Format dialog box has the following fields and options:

Format Types

Specifies the format of the current field. The field type can be alphanumeric, numeric, or date/time.

Date/Time

Opens the Date and Time Formats dialog box, where you can assign date and time formats, and apply date and time display options.

For more information, see the Creating Reporting Applications With Developer Studio manual.

Length

Specifies the length, in characters, of a field. Enter a number in the Length box, or click the arrow buttons to specify a number.

<table>
<thead>
<tr>
<th>Field Format</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alphanumeric</td>
<td>1-4096 (default, 20)</td>
</tr>
<tr>
<td>Floating Point</td>
<td>1-9 (default, 7.2)</td>
</tr>
<tr>
<td>Integer</td>
<td>1-11 (default, 5)</td>
</tr>
<tr>
<td>VarChar</td>
<td>1-4096 (default, 20)</td>
</tr>
<tr>
<td>Decimal</td>
<td>20.18 (default, 12.2)</td>
</tr>
<tr>
<td>Packed</td>
<td>33.31 (default, 12.2)</td>
</tr>
<tr>
<td>Dynamic</td>
<td>Select a Format Field.</td>
</tr>
</tbody>
</table>

Note: For numeric fields, include the decimal place in the length.

Decimal

Specifies the number of decimal places to the right of the decimal point in a Decimal, Packed Decimal, or Floating Point field.

Add minus sign if value is negative

Check this option to display a minus sign to the right of negative numeric data.
**Edit Options**

Adds display options to numeric field formats to control how the field will appear on reports.

**Example:** Calculating RECAP Expressions in the Financial Report Painter

This example shows two RECAP expressions, with the second using values derived in the first.

The first RECAP expression calculates TOTCASH (as TOTAL CASH) by adding the values in rows 1 to 3. The results of the calculation will appear in all columns. This is designated using the Apply to Row check box on the RECAP Options tab. See **RECAP Dialog Box** on page 73.

The second RECAP expression calculates GROCASH, as CASH GROWTH (%):

- TOTCASH(1) refers to total cash in column 1.
- TOTCASH(2) refers to total cash in column 2.
- The resulting calculation will appear in column 2 (LAST_YR) of the row labeled CASH GROWTH(%). This is designated in the Columns box on the RECAP Options tab.
The output is:

<table>
<thead>
<tr>
<th></th>
<th>CURRENT YEAR</th>
<th>LAST YEAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASH ON HAND</td>
<td>8,784</td>
<td>7,216</td>
</tr>
<tr>
<td>DEMAND DEPOSITS</td>
<td>4,494</td>
<td>3,483</td>
</tr>
<tr>
<td>TIME DEPOSITS</td>
<td>7,961</td>
<td>6,499</td>
</tr>
<tr>
<td>TOTAL CASH</td>
<td>21,239</td>
<td>17,198</td>
</tr>
</tbody>
</table>

**Procedure: How to Create a RECAP Summary Row**

To create a row in which existing data in other rows is summarized:

1. Place your cursor in a column that contains numeric data, then highlight the values you want to sum. The values to be added may be contiguous or non-contiguous.

2. Click the Make RECAP (Rows) \(\sum\) icon on the Design toolbar.

3. A new RECAP row appears in the Row Type column, below the last row in the matrix. The formula representing the sum of highlighted values (for example, \(R1 + R2 + R3\)) appears in the new row, in the column that contained the values.

**Tip:** Use this technique when the calculation you wish to perform is a simple addition of values that does not require an expression. For an illustration of this technique, see *Emphasizing a Row Using Border Lines* on page 98.

**Creating RECAP Expressions Using a Drag-and-Drop Operation**

RECAP expressions may be created in an FML matrix by dragging individual values from their original source cells and dropping them into a target cell in the same matrix. A valid source cell is any cell that contains numeric data, and a valid target cell is any cell that is either empty or contains only RECAP expressions. To create a RECAP expression, you can drag a value from any valid source cell, located in any row or column in an FML matrix, to any valid target cell in the same matrix. You should position a target cell in a section of your report, where data cells end and empty cells begin, at the end the rows or columns where the source cells are found.
Procedure: How to Create RECAP Expressions Using a Drag-and-Drop Operation

To create a drag-and-drop RECAP expression:

1. Position the pointer in a valid source cell.
2. Click the source cell, hold the left mouse button down, and press the Shift key.
3. Drag the source cell contents to a valid target cell and release both the left mouse button and the Shift key.

While dragging the source cell contents, you should observe that a symbol, which appears as a circle with a slash through it, is displayed when your mouse pointer is hovering above cells where you are not allowed to drop the source cell value. When you reach a valid source cell in the matrix, the symbol will no longer appear.

By default, using a drag-and-drop operation to create a RECAP expression places a plus sign (+) between all values dragged into a RECAP cell.

Inserting Rows of Text

You can add ad hoc text in a row of a financial report. This includes the ability to add blank rows (designated as text) to improve the appearance of the report. Your entry is identified as TEXT in the Row Type column.

Procedure: How to Add Text to a Financial Report

In the Design matrix:

1. Select a row in which you want to add text or create a blank row in the report.
   or
   Right-click anywhere in the row (except on the label) and select Change Type to. Select Text from the context menu.

   The Text dialog box opens.

3. Type up to 4096 characters of text in the text box, or leave the box blank.
4. Click OK.

TEXT appears in the Row Type column. If you entered text, its display starts in the Title column. If you left the entry box blank, an empty TEXT row appears.
**Reference:** TEXT Dialog Box

**Options Tab**

**Text box**

Enter up to 4096 characters of text, which will appear on the matrix starting in the Title column.

**Suppressing the Display of Rows**

Sometimes you may want to retrieve TAG, RECAP, or PICKUP rows or add values in DATA rows solely for use in a calculation, without wishing to display those rows on the report. The Financial Report Painter enables you to mark rows as invisible for this purpose.

You may also wish to suppress the display of a TAG row if no data is found for the values. The default character used to represent the missing data is a period (.). You can override this convention, making the display of a row dependent upon the existence of data.

**Procedure:** How to Suppress Rows

1. Select the TAG, RECAP, PICKUP, or DATA row you want to suppress.
2. Click the *Invisible* check box under Row Properties, above the matrix.

3. Right-click anywhere in the row (except on the label) and select *Row Properties* from the context menu. The appropriate dialog box for the row type opens.

4. On the General tab, select the *Invisible* check box.

**Procedure: How to Suppress Rows With No Data**

To suppress the display of a TAG row with missing data:

1. Select the row you want to suppress.

2. Click the *When Exists* check box under Row Properties, above the matrix.

3. Right-click anywhere in the row (except on the label) and select *Row Properties* from the context menu. The appropriate dialog box for the row type opens.

4. Select the *General* tab and choose *When Exists*.

**Saving and Retrieving Intermediate Report Results**

Many reports require results developed in prior reports. This technique requires that intermediate results be stored for reuse. An example is the need to compute net profit in an Income Statement prior to calculating equity in a Balance Sheet. In the Financial Report Painter, you can save selected rows from one or more models by posting them to a work file. These rows can then be recaptured by picking them up from the work file.

The default work file is FOCPOST. Note that this is the common WebFOCUS comma-delimited file format, so you can report from it directly if a FOCPOST Master File is available. Note also that you must create a FILEDEF command to allocate the file to a logical name (ddname) prior to executing the report. The Allocation Wizard enables you to perform a FILEDEF command before running the report that posts the data, and again before running the report that picks up the data. For details on Allocation operations, see the *Developing Reporting Applications* manual.

Note that you cannot prepare a financial report entirely from data that you enter directly in the matrix (for example, on a DATA row). However, you can prepare a report entirely from data that is stored in a comma-delimited work file.

**Posting Data**

You can save any TAG, RECAP, or DATA row by posting the output to a file. These rows can then be used as though they were provided in a DATA row.

The row will be processed in the usual manner in the report, depending on its other options, and then posted. The label of the row is written first, followed by the numeric values of the columns, each comma-separated, and ending with the terminator character of a dollar sign ($).
For example:

```
AR , 18829, 15954, $
INV , 27307, 23329, $
```

**Procedure: How to Post Data to a Work File**

1. Select a TAG, RECAP, or DATA row in the Design matrix.
2. Enter the ddname of the work file in the Post To input box for Row Properties, above the matrix.
3. Right-click anywhere in the row (except on the label) and select *Row Properties* from the context menu. The selected Row Type dialog box opens.
4. Click the *General* tab and enter a ddname in the Post To input box under Formatting Options.

The selected row is saved in the designated work file (in comma-delimited format) from which you can then pick it up for use in another financial report.

Repeat the process for each row you wish to save to a work file (either the same file or a different one).

**Note:** Remember that you must create a FILEDEF command to assign a logical (ddname) to the work file before you execute the request.

**Procedure: How to Retrieve Posted Data**

You can retrieve posted rows from a work file and use those rows as if they were provided in a TAG row.

Remember that you must create a FILEDEF command to assign a logical (ddname) to the work file before you execute the request.

From the matrix in which you want to pick up data:

1. Select the row that will contain the picked up data. (If necessary, add or insert a blank row (see *Formatting Financial Reports* on page 88).
2. Click the *Pickup* icon on the toolbar.

   or

   Right-click anywhere in the row (except on the label) and select *Pickup* from the context menu. The PICKUP dialog box opens at the General tab.

3. Enter a title for the PICKUP row in the Title box. This title will appear in the report.

4. To supply an explicit row label to replace the default label on the matrix, type it into the Label box. (The default label is retained internally.)

5. In the Pickup From box, enter the ddname for the work file that contains the posted data, followed by a space and the label of the data row you want to pick up.

6. To suppress the display of the PICKUP row, select the *Invisible* check box.

7. Click *OK*.

PICKUP appears in the Row Type column on the matrix, along with the label and title you have defined and the data values retrieved from the designated work file.

Repeat this process to include additional PICKUP rows in the report.

*Reference:* PICKUP Dialog Box
General Tab

Title

Specifies a title for the picked up data. This title will appear on the report.

Label

Lists the default label for the PICKUP row. You can replace the default row identification label with an explicit row label that you want to reference in an expression.

Formatting Options:

The following options apply to individual rows in the report.

Invisible

Picks up data for use in a RECAP calculation, but does not display the retrieved data.

Pickup From

Is the ddname of the work file that contains the data you wish to retrieve, followed by a space and the label of the PICKUP row.

Reference:  Design Matrix With Pickup Rows

In the following matrix, two rows are designated as PICKUP.

Notice that the labels for these rows are referenced in the RECAP formula.
You can see the underlying source code, including the name of the work file to which the rows of data were posted, in the following screen.

```
TABLE FILE CENTGL
SUM
  JO.RAWDATA.NAT_AMOUNT
  JO.RAWDATA.NAT_YTDAMT
FOR
  CENTGL.ACCOUNTS.GL_ACCOUNT
  '1000' AS 'Profit Before Tax' LABEL R1 OVER
DATA PICKUP R2
  AS '' LABEL R2 OVER
DATA PICKUP R3
  AS '' LABEL R3 OVER
BAR AS '' LABEL OVER
RECAP R5-PICKUP;
  AS 'RECAP'
WHERE PERIOD EQ '2002/03';
ON TABLE SET PAGE-NUM NOLKA
ON TABLE SET FORMULTIPLE ON
ON TABLE NOTOTAL
ON TABLE PCOHOLE FORMAT HTML
ON TABLE SET BLANKIDENT ON
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLE *
  INCLUDE = endif,
$
ENDSTYLE
END
```

**Formatting Financial Reports**

You can add underlines to a financial report to set off and clarify calculations and use a variety of formatting techniques to draw attention to individual columns, rows, and cells in a financial report. As in other WebFOCUS reports, certain formatting features can be triggered by conditions that you define.

You can also drill down to another procedure or a URL from a cell or column in a financial report.

For details about conditional formatting and drill-down procedures, see the *Creating Reporting Applications With Developer Studio* manual.

**Adding Underlines**

Reports that display columns of numbers frequently need to use underlines before calculations. You can specify a single or double underscore character in a BAR row.
**Procedure:** How to Add Underlines

In the Design matrix:

1. Select the row in which you want to add an underline.
2. Click the **Bar** icon on the toolbar.
   
or
   Right-click anywhere in the row (except on the label) and select *Change Type to*, and then **Bar**.

   The BAR dialog box opens.

3. The default BAR character is a single dash (-). To change the dash character, choose the double line (=) from the drop-down list on the Properties bar.

   **Note:** In HTML reports, the double line appears as a single thick line.

4. Click **OK**.

   BAR appears in the Row Type column. The selected symbol appears under columns that contain numeric data.

**Reference:** BAR Dialog Box
Options tab

Underscore characters

You can choose a single or double line. Single is the default.

Note: The double line is not clearly distinguishable in an HTML report and appears as a thicker single line.

Formatting Columns, Rows, and Cells

You can apply a wide range of formatting options to individual columns, rows, and cells in a financial report using options on the Field Properties Style tab.

You can further refine formatting:

- For columns, by identifying the column title and column data as separate objects for styling.
- For rows, by identifying the row and title as separate objects for styling.

If you style a row and then a cell within that row, the cell styling takes precedence for the cell.

If you style a column and then a cell with that column, the cell styling takes precedence for the cell.

If you style a cell, then style a row or column that the cell is in, the cell style will remain the same.

You can select an entire row by clicking the row label (R1, R2, R3, and so on) and then style that row using the font toolbar. For example, clicking R1 will highlight all entries in that row. While the row is highlighted, changing the font color to orange will result in every entry in that row having its font color changed to orange.

Row and cell styling options are specific to financial reports created in the Financial Report Painter. Column styling options are identical whether applied from the Financial Report Painter or Report Painter.

Procedure: How to Format a Column in the Design Matrix

1. Right-click a column and choose Options from the context menu. The Field Properties dialog box opens at the Style tab.

2. Select Column Title and Data as the active object, or select Column Data or Column Title to style them separately.

   If you have already styled either the title or data for a column and wish to quickly apply the same formatting to the other element, click the appropriate button: Copy Title Style to Data or Copy Data Style to Title.
3. Under Column Layout, you can change column width and justification.

**Width** options are:

**Maximum**

Sets the width according to the length defined in the field format.

**Minimum**

Sets the width of the column according to the widest value or heading in the field. This is selected by default.

**Truncate**

Enables you to specify where to truncate the column width based on the specified units (inches or centimeters). To indicate that a field value has been truncated in the browser, WebFOCUS places an exclamation point (!) after every alphanumeric and text field value and an asterisk (*) after every numeric field value.

**Wrap**

Enables you to specify where to wrap data based on the specified units (inches or centimeters).

**Justification** options are: left, right, center, or return to the default positioning.

**Active Technologies Reports Present Hidden**

Enables you to select the present hidden option. This determines the columns that will be hidden from view in the report output, when using the HTML active report output format.
4. Under Graphical options, select the font characteristics, border or grid characteristics, and/or background colors that you wish to apply to the column:

   a. **For borders**, click the Select Borders button. The Borders dialog box opens. Select width, style, and/or color options from the drop-down lists.

      You can apply the same specifications to all border lines or vary specifications for top, bottom, right, and/or left borders.

      **Note:** To set borders in an HTML report, Cascading Style Sheets must be on. Click Features from the Report menu. The Report Options dialog box opens. Ensure that an HTML styled report format is selected and verify that Cascading Style Sheets is checked.

      When Borders is selected, Grids is disabled.

   b. **For grids**, click the Select Grids button. Select a line style and indicate whether to display horizontal lines, vertical lines, or both. This option applies to columns in PDF reports but does not apply to columns in HTML reports.

      When Grids is selected, Borders is disabled.

   c. **For fonts**, click the Select Fonts button. The Fonts dialog box opens. Select font name, font style, font size, and/or color.

   d. **For background color**, click the Single Color option button under Background Coloring, and choose a color from the palette.

      If you identify the active object as column data, the Alternating Background Colors button is activated. You can use this feature to assign colors to alternating rows in one or more columns.

      Click OK to return to the Style tab.

5. Under Applying to condition in the Style tab, you can define or edit a condition that controls when specified formatting options are applied to one or more columns.

6. Click OK to return to the matrix where many styling changes will be reflected.
7. Click the Run icon on the toolbar to see the column formatting options applied in the report output.

**Tip:** To affect other column features, click the:

- **Drill Down tab** to drill down to another procedure, a URL, or another supported option.
- **General tab** to change the column title or field format, to make the column visible or invisible in the output, or to activate other features.

For details about conditional formatting and drill-down procedures, see the *Creating Reporting Applications With Developer Studio* manual.

**Example:** Formatting Columns With Data Visualization Graphs and Conditional Styling

This example uses the joined data source files, CENTGL and CENTSYSF (see *Sample Data for FML Hierarchy Examples* on page 68), and takes as its starting point the request created in *Displaying One Summary Line for an FML Hierarchy* on page 58. However, instead of applying a predefined stylesheet, as you did in the earlier example, you will be formatting the report yourself. You will apply bold formatting to each column title, data visualization bar graphs to a numeric column to help you quickly visualize trends and relationships in your data, and conditional styling to the data in a second numeric column.

1. If you completed the referenced example, reopen the Design matrix. Click the **Report Option** button on the toolbar, and in the Style tab, click the **Style File Selection** button to open the StyleSheet File Selector. Click the **Add new item** button, select *deflt.sty*, and click **OK** from the StyleSheet File Selection dialog box. Click **Finish** and **OK** to apply the selected StyleSheet.

If you did not complete the referenced example, follow those instructions up to step 13, then return here.

2. Click the **Report Option** button on the toolbar and select the **Output** tab. Make sure that **HTML** is the selected Display format and that **Cascading Style Sheets** is turned on. You will need this feature of HTML to display the data visualization graphs. Click **OK** to return to the matrix.

3. Right-click the **Month Actual** column and select **Options** from the menu. The Field Properties dialog box opens at the Style tab.

4. Make **Column Title** the active object.

5. Click the **Select Font** button and choose **Bold** from the Font Style box.

6. Click **OK**. The column title, Month Actual, becomes bold in the matrix.

7. Repeat steps 3-6 for YTD Actual, then click the **Field Properties General** tab.

8. Click the **Visualize** button. The Data Visualization dialog box opens.
9. Click the **Visualize** check box and select **RED** from the Color drop-down list.

10. Click **OK**. YTD Actual becomes bold in the matrix. You will not see the effect of the data visualization graphics until you run the report.

11. Click the **Run** icon on the toolbar.

The output is:

<table>
<thead>
<tr>
<th></th>
<th>Month Actual</th>
<th>YTD Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Selling Expenses</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Advertising</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TV/Radio</td>
<td>1,049,146</td>
<td>2,954,342</td>
</tr>
<tr>
<td>Print Media</td>
<td>244,589</td>
<td>721,448</td>
</tr>
<tr>
<td>Internet Advertising</td>
<td>9,542</td>
<td>29,578</td>
</tr>
<tr>
<td>Promotional Expenses</td>
<td>53,719</td>
<td>151,732</td>
</tr>
<tr>
<td>Joint Marketing</td>
<td>97,135</td>
<td>289,799</td>
</tr>
<tr>
<td>Bonuses/Commisions</td>
<td>100,188</td>
<td>304,199</td>
</tr>
<tr>
<td><strong>Selling Expenses</strong></td>
<td>1,554,319</td>
<td>4,451,098</td>
</tr>
</tbody>
</table>

Suppose that you want to highlight Monthly Actual Values that are less than 100,000.

1. In the Design matrix, right-click the column title **Month Actual** and choose **Options** from the menu. The Field Properties dialog box opens at the Style tab. The active item is Column Data. You will now define a condition to control the styling of this data.

2. Click the **Edit Condition** button. The Condition List dialog box opens.

3. Click **New**. The Edit Condition dialog box opens.
4. Accept the default condition name, `COND0001`. Select `NAT_AMOUNT` from the Fields box, *is less than* from the Relation drop-down list, and enter `100000` in the Values box.

5. Click OK. The Condition List dialog box shows your entries.
6. If they are correct, click OK again to return to the Field Properties dialog box at the Style tab, where you will define the style you want to apply when the condition is met.

7. Click the Font button. The Font dialog box opens.

8. Choose Bold from the Font Style list and click OK. Notice that your selection is reflected in the Sample box.

9. Click the Run icon.
The values below 100,000 appear in boldface type:

<table>
<thead>
<tr>
<th>Selling Expenses</th>
<th>Month Actual</th>
<th>YTD Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advertising</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TV/Radio</td>
<td>1,049,146</td>
<td>2,954,342</td>
</tr>
<tr>
<td>Print Media</td>
<td>244,589</td>
<td>721,448</td>
</tr>
<tr>
<td>Internet Advertising</td>
<td>9,542.</td>
<td>29,578.</td>
</tr>
<tr>
<td>Promotional Expenses</td>
<td>53,719.</td>
<td>151,732.</td>
</tr>
<tr>
<td>Joint Marketing</td>
<td>97,135</td>
<td>289,799</td>
</tr>
<tr>
<td>Bonuses/Commisions</td>
<td>100,188.</td>
<td>304,199.</td>
</tr>
</tbody>
</table>

| Selling Expenses | 1,554,319. | 4,451,098. |

**Procedure:** How to Format a Row in the Design Grid

1. Right-click a row label (either a default label like R1 or an explicit row label that you have assigned) and select Options from the context menu. The Field Properties dialog box opens at the Style tab, with Row identified as the active object.

2. Under Graphical, at the right of the Style tab, select the font characteristics, border characteristics, and/or background colors that you wish to apply to the row.

   a. **For borders**, click the Select Borders button. The Borders dialog box opens. Select width, style, and/or color options from the drop-down menus.

      You can apply the same specifications to all border lines or vary specifications for top, bottom, right, and/or left borders. Click OK.

   
   Note: To set borders in an HTML report, Cascading Style Sheets must be on. Click Features from the Report menu. The Report Options dialog box opens. Ensure that an HTML styled report format is selected and verify that Cascading Style Sheets is checked.

   b. **For fonts**, click the Select Fonts button. The Fonts dialog box opens. Select font name, font style, font size, and color. Click OK.

   c. **For background color**, click the Single Color option button under Background Coloring and choose a color from the palette. Click OK.

3. Click OK to return to the Design matrix where many styling changes will be reflected.
4. Click the Run icon on the toolbar to see the row formatting options applied to the report output.

**Reference:** Design Matrix Row Formatting Examples

The following are examples of formatting rows using the design matrix.

**Example:** Emphasizing a Row Using Border Lines

The following example places a thick dashed border around the RECAP row identified by the label TOTCASH.

1. Create the report using the sample data source Ledger. In Report Painter, designate Account as the For field and Amount as a Sum field.
2. Click the Matrix tab.
3. Above the matrix, select the Use Multiple Values check box to provide optimal flexibility in reusing tag values (although in this illustration you will use each value only once).
4. Drag the tag values 1010, 1020, and 1030 onto the matrix. Right-click each value and choose Row Properties. The TAG dialog box opens. Click the General tab and assign the labels CASH, DD, and TD to rows R1, R2, and R3, respectively. Click OK after each entry.
5. Create a RECAP row that totals the values of rows CASH (R1), DD (R2), and TD (R3).
One way to quickly create a row for this simple type of calculation is to place your cursor in a column that contains numeric data and highlight the values you want to add. Then click the **Make RECAP (Rows)** icon on the Design toolbar to create the RECAP row. A formula representing the sum of the highlighted values (identified by their row labels) appears in the new row, in the column that contained the values. For information about creating more complex RECAP expressions, see *Performing Inter-Row Calculations* on page 71.

<table>
<thead>
<tr>
<th>Label</th>
<th>Row Type</th>
<th>ACCOUNT</th>
<th>Title</th>
<th>1</th>
<th>AMOUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASH</td>
<td>TA0</td>
<td>1010</td>
<td>CASH ON HAND</td>
<td></td>
<td>11111</td>
</tr>
<tr>
<td>DD</td>
<td>TA0</td>
<td>1020</td>
<td>DEMAND DEPOSITS</td>
<td></td>
<td>11111</td>
</tr>
<tr>
<td>TD</td>
<td>TA0</td>
<td>1030</td>
<td>TIME DEPOSITS</td>
<td></td>
<td>11111</td>
</tr>
<tr>
<td>TOTCASH</td>
<td>RECAP</td>
<td></td>
<td>TOTAL CASH</td>
<td></td>
<td>R1 + R2 + R3</td>
</tr>
<tr>
<td>R5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6. Select **Styling** from the Report menu. The Report Options dialog box opens at the Style tab. Click the **Style File Selection** button. Click the **Add new item** button, select *deflt.sty*, and click **OK** from the StyleSheet File Selection dialog box. Click **Finish**, then click **OK** to apply the selected StyleSheet. This will add some basic styling to your report.

7. Right-click the row label **TOTCASH** and select **Options** from the context menu.
8. Click the Select Borders button under Graphical options. The Borders dialog box opens.

![Borders dialog box]

Note: To set borders in an HTML report, Cascading Style Sheets must be on. Click Features from the Report menu. The Report Options dialog box opens. Ensure that an HTML styled report format is selected and verify that Cascading Style Sheets is checked.

9. To apply the same specification to the top, bottom, right, and left borders, leave the Make all borders the same check box selected.

10. In the Top Border box:
   a. Select Medium from the Style drop-down list.
   b. Select Dashed from the Style drop-down list.

11. Click OK to return to the Style tab and OK again to return to the Design matrix.

12. Click the Run icon.
The output is:

<table>
<thead>
<tr>
<th>AMOUNT</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CASH ON HAND</td>
<td>8,784</td>
</tr>
<tr>
<td>DEMAND DEPOSITS</td>
<td>4,494</td>
</tr>
<tr>
<td>TIME DEPOSITS</td>
<td>7,961</td>
</tr>
<tr>
<td>TOTAL CASH</td>
<td>21,239</td>
</tr>
</tbody>
</table>

Notice that the dashed border has been placed around the entire row. The gray background is applied by a default style template that you selected in step 6.

In the following variation, specifications are applied separately to the top, bottom, left, and right border lines. A heavy black border line is placed above and below the RECAP row. A thinner dotted line is placed to the left and right of each column in the row.

1. Once again, right-click the row label TOTCASH and select Options from the context menu.
2. Click the Select Borders button under Graphical options. The Borders dialog box opens.
3. This time, deselect the Make all borders the same check box. In the Top Border box:
   a. Select Heavy from the Width drop-down menu.
   b. Select Solid from the Style drop-down menu.
4. In the Bottom border box, click the Same as Top Border check box.
5. In both the Left and Right Border boxes, specify Width as Medium and Style as Solid.

The output is:

<table>
<thead>
<tr>
<th>AMOUNT</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CASH ON HAND</td>
<td>8,784</td>
</tr>
<tr>
<td>DEMAND DEPOSITS</td>
<td>4,494</td>
</tr>
<tr>
<td>TIME DEPOSITS</td>
<td>7,961</td>
</tr>
<tr>
<td>TOTAL CASH</td>
<td>21,239</td>
</tr>
</tbody>
</table>

**Procedure:** How to Apply Boldface to a Free Text Row

The following procedure is an example of applying boldface to a free text row.

1. Create the report using the sample data source Ledger. In Report Painter, designate Account as the For field, and Amount as a Sum field.
2. Click the Matrix tab.

3. In row R1, right-click in the Row Type column. Choose Change type to from the context menu, and select Text. The TEXT dialog box opens.

4. Type —CASH ACCOUNTS— in the input box, then right-click in the Label column of row R1 and select Options. The Field Properties dialog box opens at the Style tab.

5. Under Graphical, click the Select Font button. The Fonts dialog box opens. Select Bold in the Font Style list and click OK. Click OK again to return to the matrix.

6. Drag the tag values 1010, 1020, and 1030 onto the matrix.

7. Right-click in the Row Type column for tag 1010 and select Row Properties. When the TAG dialog box opens, click the General tab and enter the title CASH ON HAND in the Title input box.

8. Repeat steps 6 and 7 for tags 1020 and 1030, using the titles DEMAND DEPOSITS and TIME DEPOSITS, respectively.

9. In row R5, right-click in the Row Type column, choose Change type to from the context menu, and select Text again. This time leave the input area in the Text dialog box blank to create an empty text row. It will appear as a skipped line in the report output.

10. In row R6, create a third text row and type —OTHER CURRENT ASSETS— in the input area of the Text dialog box, then right-click in the Label column of row R1 and select Options. The Field Properties dialog box opens at the Style tab.

11. To apply boldface to this text row, follow the instructions in step 5.

12. To complete the report, drag the tag values 1100 and 1200 into rows R7 and R8 on the matrix and enter the following row titles for tags 1100 and 1200, ACCOUNTS RECEIVABLE and INVENTORY, respectively, in the TAG dialog box.
The matrix should look as follows:

<table>
<thead>
<tr>
<th>Label</th>
<th>Row Type</th>
<th>ACCOUNT</th>
<th>Title</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>TEXT</td>
<td></td>
<td>---CASH ACCOUNTS---</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R2</td>
<td>TAG</td>
<td>1010</td>
<td>CASH ON HAND</td>
<td>1111</td>
<td></td>
</tr>
<tr>
<td>R3</td>
<td>TAG</td>
<td>1020</td>
<td>DEMAND DEPOSITS</td>
<td></td>
<td>1111</td>
</tr>
<tr>
<td>R4</td>
<td>TAG</td>
<td>1030</td>
<td>TIME DEPOSITS</td>
<td></td>
<td>1111</td>
</tr>
<tr>
<td>R5</td>
<td>TEXT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R6</td>
<td>TEXT</td>
<td></td>
<td>---OTHER CURRENT ASSETS---</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R7</td>
<td>TAG</td>
<td>1100</td>
<td>ACCOUNTS RECEIVABLE</td>
<td>1111</td>
<td></td>
</tr>
<tr>
<td>R8</td>
<td>TAG</td>
<td>1200</td>
<td>INVENTORY</td>
<td>1111</td>
<td></td>
</tr>
</tbody>
</table>

13. Click the Run icon.
The output is:

<table>
<thead>
<tr>
<th>AMOUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>---CASH ACCOUNTS---</td>
</tr>
<tr>
<td>CASH ON HAND</td>
</tr>
<tr>
<td>DEMAND DEPOSITS</td>
</tr>
<tr>
<td>TIME DEPOSITS</td>
</tr>
</tbody>
</table>

| ---OTHER CURRENT ASSETS--- |
| ACCOUNTS RECEIVABLE | 18,829 |
| INVENTORY           | 27,307 |

**Note:** If you were to look at the source code for this procedure, you would see the two free text rows specified using the following code:

```
TYPE=REPORT,
   LABEL=R1,,
   STYLE=BOLD,, $
TYPE=REPORT,
   LABEL=R6,,
   STYLE=BOLD,, $
```


**Procedure:** **How to Format a Cell in the Design Matrix**

The following procedure describes how to format a cell in the Design Matrix.

1. Right-click a cell and choose **Options** from the context menu. The Field Properties dialog box opens at the Style tab, with Cell identified as the active object.

2. Under **Graphical**, select the font characteristics, border characteristics, and/or background colors that you wish to apply to the cell:

   a. For borders, click the **Select Borders** button. The Borders dialog box opens. Select width, style, and/or color options from the drop-down menus. Click **OK**.

   You can apply the same specifications to all border lines or vary specifications for top, bottom, right, and/or left borders.
Note: To set borders in an HTML report, Cascading Style Sheets must be on. Click Features from the Report menu. The Report Options dialog box opens. Ensure that an HTML styled report format is selected and verify that Cascading Style Sheets is checked.

b. For fonts, click the Select Fonts button under Graphical options. The Fonts dialog box opens. Select font name, font style, font size, and color. Click OK.

c. For background color, click the Single Color option button under Background Coloring, and choose a color from the palette. Click OK.

3. Under Applying to condition, you can define or edit a condition that controls when specified formatting options are applied to a cell.

4. Click OK to return to the Design matrix, where most formatting changes will be reflected.

5. Click the Run icon on the toolbar to see the cell formatting options applied in the report output.

Note: To drill down from the selected cell to another procedure or to a URL, click the Drill Down tab. For details, see Formatting Columns, Rows, and Cells on page 90.

Although certain options on the General tab are enabled when cell is the active object, these options will be applied to the entire column, not only to the selected cell. The Field Properties General tab options are available for the following fields in the Design Matrix.

- Set Title and Output options are available for the FOR field in the Design Matrix.
- All options (except Table of Contents) are available for SUM and DETAIL fields in the Design Matrix.

Example: Highlighting a Cell With Border Lines and Boldface Type

The following example places a solid line of medium thickness around a cell in the RECAP row identified by the label TOTCASH.

1. Create the report using the sample data source Ledger. In the Report Painter, designate Account as the For field and Amount as a Sum field.

2. Click the Matrix tab.

3. Above the matrix, click the For Multiple button to provide optimal flexibility in using tag values (although in this illustration you will use each value only once).
4. Drag the tag values 1010, 1020, and 1030 onto the matrix. Notice that the tag values appear in the Title column. To provide more meaningful titles, either type the following entries into the cells on the matrix or open the TAG dialog box and enter the new titles in the General tab. Replace 1010 with CASH on HAND, 1020 with DEMAND DEPOSIT, and 1030 with TIME DEPOSIT.

5. In row R4, right-click on the Row Type column, choose Change Type to, and then Bar. The BAR dialog box opens. Click OK to confirm the default underline character. The underline is displayed in the Amounts column, below the values to be summed.

6. In row R5, create a RECAP row that totals the values of rows R1, R2, and R3. Click OK.

   One way to quickly create a row for this simple type of calculation is to place your cursor in a column that contains numeric data and highlight the values you want to add. Then click the Recap icon on the Design toolbar to create the RECAP row. A formula representing the sum of the highlighted values (identified by their row labels) is displayed in the new row, in the column that contained the values. For information about creating more complex RECAP expressions, see Performing Inter-Row Calculations on page 71.

7. Change the Title for the RECAP row to TOTAL CASH.

   The matrix appears as follows:

8. Right-click the cell that contains the RECAP formula and select Options from the menu. The Field Properties dialog box opens at the Style tab, with Cell identified as the active object.

9. Click the Select Borders button under Graphical options.
The Borders dialog box opens.

**Note:** To set borders in an HTML report, Cascading Style Sheets must be on. Click Features from the Report menu. The Report Options dialog box opens. Ensure that an HTML styled report format is selected and verify that Cascading Style Sheets is checked.

10. To apply the same specification to the top, bottom, right, and left borders, leave the *Make all borders the same* check box selected.

11. In the Top Border box:
   a. Select *Medium* from the Width drop-down list.
   b. Select *Solid* from the Style drop-down list.

12. Click OK to return to the Style tab and OK again to return to the Design matrix.

13. Right-click the same cell and choose *Options* again. Click the *Select Font* button and choose *Bold* from the Font Style list.

14. Click OK to return to the Style tab and OK again to return to the Design matrix.

15. Click the *Run* icon.
The output is:

<table>
<thead>
<tr>
<th>AMOUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASH ON HAND</td>
</tr>
<tr>
<td>DEMAND DEPOSIT</td>
</tr>
<tr>
<td>TIME DEPOSIT</td>
</tr>
<tr>
<td>TOTAL CASH</td>
</tr>
</tbody>
</table>

The bold font style and solid border draw attention to the Total in the Amount column.

**Procedure:** How to Apply Conditional Formatting to a Cell

To create a condition:

1. Right-click the cell in the Design matrix and select Options from the context menu.
   The Field Properties dialog box opens at the Style tab.
2. Click the Edit Conditions button.
   The Condition List dialog box opens.
3. Click New.
   The Edit Condition dialog box opens.
4. Type a name for the condition or accept the default.
5. Select the field for which you want to define a condition from the Field box.
6. Select a relation from the Relations list.
7. Specify a value by doing one of the following:
   - Type a literal value in the Value box.
   - Click the Values button to display existing database values and select a value.
   - Click Another Field to display other fields in the report and select a field.
8. Click OK.
   The condition is added to the Condition List in the Condition List dialog box, from which it can be applied to one or more cells, as well as to columns or rows.

**Tip:** To create another condition, repeat steps 4 to 10.
9. Click OK again to return to the Style tab on the Field Properties dialog box.

The default condition name (for example, COND0001) appears in the Applying to Condition field box.

The condition you created appears in the Condition Rule box.

10. You are now ready to associate the condition with a style. Select a defined condition from the Applying to Condition dialog box in the Style tab.

11. In the Graphical area of the Style tab, choose one or combination of the following:

- Click the Select Font button to specify conditions such as font, font style and size, color and effects.
- Click the Select Grid button to select a line style (only for PDF reports).
- Click the Select Borders button to select line style, width, and/or color options from the drop-down menus.

**Note:** To set borders in an HTML report, Cascading Style Sheets must be on. Click Features from the Report menu. The Report Options dialog box opens. Ensure that an HTML styled report format is selected and verify that Cascading Style Sheets is checked.

- In the Background Coloring area, click the Single Color option button to select a background color.

**Drilling Down From a Column, Cell, or Row**

The drill down feature enables you to add one or more layers of detail to a report by embedding procedures into the report. A drill down procedure can be:

- Any type of executable object, such as a report or graph.
- A link to a URL from an HTML or PDF report, or to a JavaScript from an HTML report.
- A link to a Maintain case (function).
- A link to a Maintain procedure.

After the base (summary) report appears in the browser, you can drill down to selected data, a URL, or script (or procedure). For example, if you run a report on the country and models of all cars imported into the United States, and you have embedded the appropriate procedures in the report and defined the correct parameters, you can drill down on:

- **ENGLAND,** to view a more detailed report on the sales of all English cars.
Formatting Financial Reports

- JAGUAR, to see specific sales information on the Jaguar.
- A summary report row, to view the values behind each field in the row.

From either Report Painter or the Financial Report Painter, you can drill down from a column title and its data, or from the title only, or the data only.

In addition, from the Financial Report Painter you can drill down from a single cell in the matrix or from a row or row title.

These behaviors are invoked from the Field Properties Drill Down tab. The active object box indicates the matrix element from which the drill down will be implemented. The Drill-Down Definition indicates what you want to drill down on: a URL, a procedure, or one of the other supported options.

Notice that you can also define conditions that determine when to run an embedded drill-down procedure. For example, suppose you have embedded a procedure in the country column of the summary report, but you want your drill-down report to appear only for England. You can define a condition that enables the embedded drill down only when country equals England.
The mechanics of establishing a drill-down relationship are the same, regardless of the active object you choose.

For details about this process, see the *Creating Reporting Applications With Developer Studio* manual.

**Procedure: How to Access the Drill Down Tab**

- For a column, right-click the column title.
  - If you select Options from a column title, Column and Title is the active object. However, you can change the active object to column title or column data.

- For a cell, right-click the cell, then choose *Options* from the context menu. The Field Properties dialog box opens. Click the *Drill-Down* tab.
  - If you select Options from a cell, Cell is the active object. However, you can change the active object to column and title, column title, column data, row title, or row data.

**Adding, Inserting, and Deleting Rows**

You can add a new row at the bottom of the matrix, insert a row between existing rows, or delete a row from the matrix. The Insert Row and Add Rows dialog boxes provide the additional tabs you need to define a row of the selected type.

You can also insert a new tag row above or below existing tag rows in a hierarchy.

**Procedure: How to Add or Insert a Row in the Matrix**

1. Click the *Add* or *Insert* icon on the toolbar.
   
   or

   Right-click anywhere in the row and select *Add Row* or *Insert Row* from the context menu.

   The Add or Insert Row dialog box opens.

2. Choose a row type: Tag, Recap, Data, Text, Bar, or Pickup.

   The Options and General tabs adjust to provide the appropriate options for the selected row type. (See topics on specific row types for additional information.)

3. Complete your entries in the Options and General tab.

4. Click *OK* to add the new row to the Row Type column in the matrix.
Procedure: How to Add or Insert a Hierarchy Tag Value in the Matrix

In a hierarchy, you can insert a tag row above or below an existing tag row or add a tag value to an existing row.

1. In the For field values panel, press the Shift key and simultaneously click the tag value you wish to add, then drag the value onto a tag row in the matrix.

2. Choose one of the following options from a menu that opens.
   - Insert before current row or Insert after current row.
     When you choose either of these options, you can select a Children option from a secondary menu rather than having to open the TAG dialog box to make this selection. For details about these options, see TAG Dialog Box on page 44.
   - Add to current row.
     When you choose this option, the keyword OR is added to the row and values of the referenced tags are summed. For related information, see How to Specify a Tag for Multiple Data Values on page 40.

Procedure: How to Delete a Row

Click the Delete icon on the toolbar.

or

Right-click anywhere in a row and select Delete Row from the context menu.

Editing Row Types and Properties

The Financial Report Painter provides a context menu as an alternative to the toolbar. You can use the context menu to add rows and to change the type and properties of a row.

Procedure: How to Change Row Types

1. Right-click anywhere in a row (except on the row label).
2. Select Change Type to from the context menu.
3. You are asked to verify that you want to change the row type. Click Yes.

The Financial Report Painter displays the dialog box you need to complete your row definition. For details, see topics on each row type.
**Procedure:** How to View or Change Row Properties

1. Right-click anywhere in a row (except on the row label).
2. Select *Row Properties* from the context menu.

The dialog box for the selected row type opens.

**Procedure:** How to Cut, Copy, and Paste Rows

To use the standard Windows Clipboard options:

1. Right-click anywhere in the row.
2. Select *Cut Row* or *Copy Row* from the context menu.
3. Right-click a new location on the matrix and choose *Paste Row* from the context menu.

**Note:** If you Cut and Paste, the row and its label are placed in the new location. If you Copy and Paste, the row contents are copied, but a default label is applied, since each row requires a unique label.

Adding and Deleting Columns

Although you initially define the columns in your report in Report Painter, as you refine your financial report, you can include additional columns or delete columns directly from the Financial Report Painter.

You can include a field from the data source as a new column or create a column as a calculated value.

**Procedure:** How to Add a Column From the Data Source in a Financial Report

1. Right-click a column title or number and select *Add Column* or *Insert Column* from context menu. The Add Columns or Insert Columns dialog box opens.
2. Select a field name from the *Select columns to insert* box and click *OK*. The field is either added as the last column in the report or inserted before the column in which you click.

**Procedure:** How to Add a Calculated Value in a Financial Report

To add a calculated value as a new column:

1. Select *Computes* from the Reports menu, or click the Computes icon from the Setup toolbar.

The Report Options dialog box opens at the Computes tab.

2. Create an expression from which the calculated value will be derived.
3. Click **Apply** to add the calculated value to the Object Inspector, then click **OK** to close the Computes tab and return to the matrix. The new column is displayed as the last column in the matrix.

**Tip:** You can drag this column to a different position in the report.

For details about calculated values, see *Creating Temporary Fields* in the *Creating Reporting Applications With Developer Studio* manual.

**Procedure:** **How to Delete a Column**

Right-click a column title or number and select **Delete column** from context menu.

The column is removed from the matrix.
The following tutorial is an example of creating a financial report. It gives a basic understanding of the functionality of the Financial Report Painter.

**In this chapter:**

- Create a Project
- Create a Procedure Within the Project
- Create the Base Report in Report Painter
- Create the Income Statement in the Financial Report Painter

### Create a Project

As an application developer for Century Corporation, you have been assigned the task of creating an income statement that examines the profit components of the company, including revenues, cost of goods sold, and expenses.

First, you will create a project called FINANCIAL in the local projects area of the Developer Studio Explorer. This is where you access and create files. You will build your financial report within this project.

1. Launch Developer Studio. The WebFOCUS Reporting Server is activated at the same time.
2. Right-click **Projects on localhost** in the Explorer and choose **New Project**.

![Project creation in Developer Studio](image)

The Create a Project wizard opens at Step 1.
3. Name the project FINANCIAL and accept the default directory for the new project. Notice that the project name is registered as FINANCIAL.GFA (Graphical FOCUS Application).

**Note:** Projects directories are created in lowercase.

4. Click Next.

The following message appears:

*Directory financial does not exist. Do you want to create it?*

5. Click Yes to confirm the creation of the physical directory for the new project.

The Create a Project wizard opens at Step 2.

6. Specify a directory path to the data source from which the project needs to retrieve information.

7. Click Add.
The Browse For Folder dialog box opens.

8. Choose ibinccen. This is the directory in which the sample files for the fictional Century Corporation are stored. You will be using some of these files to create your income statement.

9. Click OK.

11. Click the plus sign (+) to expand the FINANCIAL folder. You will see several subfolders: HTML Files, Image Files, Maintain Files, Master Files, Procedures, and Other.

**Tip:** These are virtual folders, categorized in this way for your convenience. They are called virtual folders because they apply a logical structure to the project but do not actually exist as physical directories. The files listed in these folders, along with the GFA file of a project, are actually stored on a WebFOCUS Reporting Server and a web or application server. Nevertheless, you can identify and access the files associated with a project from a folder in the local projects development area of Developer Studio, without concern for where the files really reside.
To complete the tutorial, you will use the Master Files and Procedures folders.

**Add Master Files to Your Project**

1. Expand the Master Files folder under the FINANCIAL project. You will see a list of all the Master Files in the path you specified when you created the project:

   `ibi\localhost\apps\ibinccen`

   Notice that the icons are grayed out. This indicates that the files are available, but not active for your project. Associate the Master File you will need for the tutorial with the FINANCIAL project.

2. Select `CENTSTMT` and click the Add icon on the toolbar (or right-click and select Add to Project).

   The icon for this Master File is active for your project.

3. Click the binoculars icon to limit the list to the active Master File.

   ![Developer Studio: Exploring - Developer Studio Desktop/Projects on localhost/TRANSACTION/Master Files](image)

   **Tip:** You can toggle back to the full list at any time.

**Data Detour: Where Is the Data?**

Detour briefly to see where the corresponding data is stored.

1. Expand `WebFOCUS Environments`. The WebFOCUS Environments area of the Explorer shows the actual files that reside on the WebFOCUS Reporting Server and the web or application server.

2. Double-click `localhost` under WebFOCUS Environments. This is the Reporting Server installed on your local machine.

3. Open `Data Servers`, followed by `EDASERVE`, then `Applications`.
4. Expand the folder labeled ibinccen (this is the directory you cited earlier as the location in which the data for your project is stored).

5. Organize the list by Type. Double-click the Master Files folder. Scroll down to see the full list of Master Files available on the path you identified, including the one you added to the project, CENTSTMT.MAS. Then scroll back up to see the corresponding data source, CENTSTMT.FOC.

The Master File is visible in the virtual folder in the local Projects development area, but the data source file is only visible on the server itself.

When you run a report later in this exercise, WebFOCUS will identify, locate, and read the active Master File (.MAS) for the data source (.FOC) named in the request. It will then interpret the contents of the data source based on the information in the Master File. You do not have to make the data source files active.

Notice that the ibinccen folder also contains some sample procedures in the Procedures folder (.FEX files). You will not be using these procedures. (They have been created for use in demos and other exercises.) For this tutorial you will create your own financial reporting procedure.

6. Close WebFOCUS Environments since you will create the procedure in the local projects development area.

**Create a Procedure Within the Project**

Create and name the procedure and select the tool you will use to create the report.

1. If it is not already open, expand the FINANCIAL project folder.

2. Click the Procedures folder. Since you will not be using any of the sample procedures in this project, click the binoculars icon to empty the active box.

3. Right-click the Procedures folder and choose New, and then Procedure.

The Add Procedure dialog box opens.
4. Name the procedure **INCOME1** and choose **Report Painter** from the Create with drop-down list.

5. Click **Open**.

Before Report Painter opens, the Open dialog box asks you to specify the data source you will be using to create your report.

6. Select **CENTSTMT.MAS**.

7. Click **Open**.

Report Painter opens. The Object Inspector displays the fields in the CENTSTMT data source.

Data Detour: What Is the Nature of the Data You Will Be Using?

The CENTSTMT Master File used in this tutorial has two segments:

- **SEGNAME=ACCOUNTS** defines a chart of accounts hierarchy. The field GL_ACCOUNT_PARENT is the parent field in the hierarchy. The field GL_ACCOUNT is the hierarchy field. The field GL_ACCOUNT_CAPTION can be used as the descriptive caption for the hierarchy field.

  The hierarchy will come into play in the second part of this tutorial, when you begin to structure the income statement in the Financial Report Painter.

- **SEGNAME=CONSOL** contains financial data structured as a data mart in which detailed data has already been rolled up to optimize reporting.

```plaintext
FILE=CENTSTMT, SUFFIX=FOC
SEGNAME=ACCOUNTS, SEGTYPE=S1
FIELD=GL_ACCOUNT, ALIAS=GLACCT, FORMAT=A7,
      TITLE='Ledger,Account', FIELDTYPE=I,$
FIELD=GL_ACCOUNT_PARENT, ALIAS=GLPAR, FORMAT=A7,
      TITLE=Parent,
      PROPERTY=PARENT_OF, REFERENCE=GL_ACCOUNT,$
FIELD=GL_ACCOUNT_TYPE, ALIAS=GLTYPE, FORMAT=A1,
      TITLE=Type,$
FIELD=GL_ROLLUP_OP, ALIAS=GLROLL, FORMAT=A1,
      TITLE=Op,$
FIELD=GL_ACCOUNT_LEVEL, ALIAS=GLLEVEL, FORMAT=I3,
      TITLE=Lev,$
FIELD=GL_ACCOUNT_CAPTION, ALIAS=GLCAP, FORMAT=A30,
      TITLE=Caption,
      PROPERTY=CAPTION, REFERENCE=GL_ACCOUNT,$
SEGNAME=CONSOL,
SEGTYPE=S1, PARENT=ACCOUNTS,$
FIELD=PERIOD, ALIAS=MONTH, FORMAT=YYM,$
FIELD=ACTUAL_AMT, ALIAS=AA, FORMAT=D10.0, MISSING=ON,
      TITLE='Actual',$
FIELD=BUDGET_AMT, ALIAS=BA, FORMAT=D10.0, MISSING=ON,
      TITLE='Budget',$
FIELD=ACTUAL_YTD, ALIAS=AYTD, FORMAT=D12.0, MISSING=ON,
      TITLE='YTD,Actual',$
FIELD=BUDGET_YTD, ALIAS=BYTD, FORMAT=D12.0, MISSING=ON,
      TITLE='YTD,Budget',$
```

Create a Procedure Within the Project
Tip: If you want to examine the Master File in Developer Studio, you can expand the Master Files folder in the FINANCIAL project, right-click CENTSTMT.MAS, and choose Edit in Text Editor to see the representation shown here. Alternatively, you can double-click CENTSTMT.MAS to open the file in the Synonym Editor, where you can select each field to see how its attributes have been defined. For details, see Describing Data for an FML Hierarchy on page 263.

Create the Base Report in Report Painter


Before the detour, you had specified CENTSTMT as the data source you would be using and selected Report Painter as your tool. Report Painter should be open on your screen.

1. Begin by scrolling down in the Object Inspector to verify that all of the fields from CENTSTMT are available for your report.

2. Double-click the following fields to bring them into the Report Painter window: GL_ACCOUNT, ACTUAL_AMT, and BUDGET_AMT. (Since descriptive names are assigned to each of these fields in the Master File, those names appear as column titles.) They are all Detail fields by default.
3. Change the title from *Budget* to *Planned*.
   a. Right-click the column title and choose *Column Title* from the menu.
      The Title dialog box opens with *Budget* displayed.
   b. Type *Planned*, as shown below, and click OK.

```
Title

Planned
```

4. Select *Ledger Account* and click the *By* button, then select *Actual* and click the *Sum* button.
   Notice that Planned automatically changes to a summary column as well. These changes will enable you to sort Actual and Planned data by account.

5. Insert a title for the report.
   a. Click *Insert* in the toolbar menu and select *Report Heading*. In the Report Heading object, type

   
   **Century Corporation**
   **2010 Income Statement**

   followed by a blank line to leave space between the heading and the report.
   b. Highlight the heading text, right-click and choose *Justify*, and select *Center*. 
Report Painter appears, as shown in the following image.
c. Click the Save icon, then click Run to see what you have accomplished so far.

As expected, the output shows Actual and Planned data for each account in the Chart of Accounts. Accounts are listed in low to high order. In the full report, accounts run through 5200.

Next, you will create a column to reflect the difference between Actual and Planned amounts. This information will help Century Corporation determine which products the Sales force should be encouraged to push and where efforts should be made to reduce spending.

6. Place the cursor after the last column in the Report. Click the Options button, then select the Computes tab from the Report menu.

The Report Options dialog box opens.
7. Name the field you are computing Variance and change the decimal format to D12.2B. (B will enclose any negative numbers in parentheses in the output.)

8. Click the Fields button (bottom-right) to open the fields list. Double-click ACTUAL_AMT, click the minus sign (-) on the calculator, then double-click BUDGET_AMT. Click Close to close the Insert Field window. Your expression is reflected in the input window.

9. Click Apply to make the calculated value available in the fields list. Click OK to add the column to Report Painter.

10. Click the Variance column and make sure that the Sum button is selected.

11. To give the new column a more descriptive title, right-click Variance and choose Column Title. In the Title dialog box, type:

   Increase/
   (Decrease)
12. Save your work, then click the Run icon.

The output shows the increase or decrease of each Actual value in relation to the corresponding Planned amount.

13. Although shown in parentheses, the negative values are still a little hard to see. Make them bold and red. The value of this information will be more evident when you have completed the income statement, where red, bold values will quickly draw attention to less than expected sales on the revenue side, but also to less than expected spending on the expense side.

First, you will define the condition you want to highlight and then the style you want to apply when that condition is met.

a. Right-click Increase/Decrease and select Options.
The Field Properties dialog box opens at the Style tab.

b. Confirm that the active object field is set to Column Data.

c. Click the Edit Conditions button (lower-left).

The Condition List dialog box opens.

d. Click New.

The Edit Condition dialog box opens.

e. In the Condition input box, assign the name DECREASE. Then create the following expression to reflect the negative values.
f. Choose Variance from the Field list, is less than from the Relation list, and type 0 in the Value input box.

![Edit Condition dialog box](image)

**Edit Condition**

<table>
<thead>
<tr>
<th>Field</th>
<th>Relation</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CENTSMT.ACCU</td>
<td>is less than</td>
<td>0</td>
</tr>
<tr>
<td>CENTSMT.CONS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Variance</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The Condition List dialog box opens, displaying the completed expression.

g. Click OK.

The Condition List dialog box opens, displaying the completed expression.

h. Click OK to return to the Style tab where you will define a style to apply when this condition is met.

i. With DECREASE selected in the Apply to Condition list, click the Select Font button under Graphical.

The Font dialog box opens.

j. Click the Color button, select RED from the color palette and click OK.

k. Then select Bold from the Font Style list and click OK.

l. Click OK again to return to Report Painter.

m. Click the Save icon followed by the Run icon.
Summary and Next Steps

You have created a project and within it, you have completed the column-based setup for a report that examines Actual and Planned data for a chart of accounts.

But your goal is to create an Income Statement that tracks revenue, cost of goods sold, and expenditures.

You will do that in the next exercise.

Begin with a quick review of the differences between a standard report, like the one you just created, and the financial report you will turn it into.
In a standard tabular report:

- Report rows are displayed in sort order (high-to-low or low-to-high).
- Rows are displayed only for values retrieved from the data source.
- Text rows can only be inserted at sort breaks.
- Inter-row calculations can only be performed at sort breaks.

In a financial report:

- Specific rows can be placed anywhere in the report.
- Rows can be displayed for selected values in the data source, or values calculated from that data, as well as for values that are inserted directly or picked up from another file.
- Text and blank rows can be inserted at any point in the report.
- Inter-row calculations can be performed at any point in the report.

The income statement you will create demonstrates many of these features.

Create the Income Statement in the Financial Report Painter

You are now ready to begin row-by-row modeling of the income statement in the Financial Report Painter.

An income statement is made up of Revenue, Cost of Goods Sold, and Expenses.
Typically, it also calculates Gross Margin and Profit:

- **Gross Margin** = Revenue - Cost of Goods Sold
- **Profit** = Gross Margin - Expenses

The following is the completed income statement you will be building. You may find it useful to refer to this output as you begin to replicate it in the following steps.
Open the Financial Report Painter

1. Select the Ledger Account field in Report Painter and click the For button.

This changes the sorting mechanism for the ledger accounts field from one that sorts data by accounts in numeric order to one in which you can exercise row-by-row control of the account values. You will see the impact of this change in the rest of the tutorial.
2. Click the Matrix tab from Report Painter.


Matrix. At the left of the Financial Report Painter is an area similar to a spreadsheet that displays the columns you specified in Report Painter, beginning with a column that will contain selected values of the For field, GL_ACCOUNT, followed by a column for the corresponding account titles. Although the report has no content at this point, each row is already identified by a default label (R1, R2 ...). You will populate these labeled rows. As you build the report, a row type will be associated with each row label.

For field values panel. At the right is a panel that displays the Chart of Accounts values for the GL_ACCOUNT field. These values are shown as a hierarchical tree, based on the parent/child relationships defined in the ACCOUNTS segment of the CENTSTM Master File. (See Data Detour: What Is the Nature of the Data You Will Be Using? on page 122). The plus (+) signs indicate that you can open the hierarchy values further.

In this exercise, you will add selected data values of GL_ACCOUNT as rows in the income statement.

You will also add free text rows, blank rows, and rows that contain underlines to set off calculations.
Build the Income Statement

1. You are going to create a report that itemizes revenues of various types, so start by adding a Text row to the matrix, titled Revenue.
   
a. In row R1, right-click in the Row Type column and select Change Type to, and then Text.
      The TEXT dialog box opens.
   
b. Under the Options tab, type the following:
      Revenue:
   
c. Click OK.
      Revenue appears in the Title column on the matrix.

2. Make the text row bold.
   
a. Right-click Revenue and select Options.
      The Field Properties dialog box opens at the Style tab.
   
b. Make Row the active object (upper-left).
      Notice that in the Financial Report Painter you can style rows and cells, as well as columns, whereas in Report Painter, you can only style columns.
   
c. Click Select Font under Graphical (upper-right).
      The Font dialog box opens.
   
d. Choose Bold as the Font Style and click OK.
e. Click OK on the Style tab to return to the matrix.

![Field Properties for Field CENTSTMATL.ACCOUNTS.GL_ACCOUNT]

The matrix appears as follows. Notice that row R1 is identified as a TEXT row type, and Revenue is bold, as requested.

![The matrix appears as follows. Notice that row R1 is identified as a TEXT row type, and Revenue is bold, as requested.]

You are now going to begin to add data values from the Chart of Accounts. The first step is expanding a section of the hierarchy to expose the levels of values.

3. In the For field values panel, expand account 2000 (Gross Margin), then expand 2100 (Sales Revenue), then 2200 (Retail Sales).
4. To complete the Revenue section of the report, you will add detailed information for each revenue category, followed by a summary row for each category, and finally a pre-consolidated row for all sales revenue.

a. Start by dragging the account 2200 (Retail Sales) from the Chart of Accounts to row R2 in the GL_ACCOUNT column on the matrix.

This value becomes red in the For field values panel to indicate that it has been used in the matrix.

Notice that TAG is entered in the Row Type column. Tags represent the data values of the For field.

b. Right-click account 2200 and select Row Properties.

The TAG dialog box opens.

c. Under Children, select Show only children from the drop-down list.

Notice that Display children’s caption is checked by default.

These selections indicate that you want to see values for the children of Retail Sales, (Retail - Television, Retail - Stereo, and so on), but not the value of Retail Sales itself.

In addition, the captions (descriptive titles) for the children will appear in the output in place of their tag values.

d. Click OK to return to the matrix.
Notice that a plus (+) sign precedes 2200, indicating that it contains subordinate values that you can expand if you wish.

5. To ensure that you can use the 2200 account and other values in more than one row of the matrix, click the Use Multiple Values check box in the FML Report Properties area above the matrix.

6. Place an underscore under these values.
   a. In the Row Type column in row R3, right-click and choose Change Type to, and then Bar. The BAR dialog box opens.
   b. Click OK to accept the default underline character.

Remember that the ability to add an underline or a text row at any point in a report is an important distinguishing feature of a WebFOCUS financial report.

7. Drag account 2200 into row R4 on the matrix.
   a. Once again, right-click 2200 and select Row Properties to open the TAG dialog box.
b. This time, accept the default option under Children, *Show selected item*.

![Image of a matrix with option for selecting children]

This matrix appears as follows, with the row R2 expanded to show its children.

![Image of a matrix with expanded row R2]

The gray indicates that these rows are part of their parent and cannot be edited separately. Notice that they all have the same row label: R2.

c. Click *OK*.
8. Save your work, then click Run to see the first part of the income statement.

<table>
<thead>
<tr>
<th>Revenue:</th>
<th>Actual</th>
<th>Planned</th>
<th>Increase/(Decrease)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retail - Television</td>
<td>7,971,992.</td>
<td>7,904,000.</td>
<td>67,992.00</td>
</tr>
<tr>
<td>Retail - Stereo</td>
<td>3,063,729.</td>
<td>3,040,000.</td>
<td>23,729.00</td>
</tr>
<tr>
<td>Retail - Video Player</td>
<td>6,332,185.</td>
<td>6,080,000.</td>
<td>252,185.00</td>
</tr>
<tr>
<td>Retail - Computer</td>
<td>1,204,955.</td>
<td>1,216,000.</td>
<td><em>(11,045.00)</em></td>
</tr>
<tr>
<td>Retail - Video Camera</td>
<td>30,782,323.</td>
<td>30,400,000.</td>
<td>382,323.00</td>
</tr>
<tr>
<td>Retail Sales</td>
<td>49,355,184.</td>
<td>48,640,000.</td>
<td>715,184.00</td>
</tr>
</tbody>
</table>

Notice that the descriptive captions are displayed rather than the account numbers. Notice also that retail revenue for computers is negative, showing lower than expected sales.

9. You are going to repeat this process for Mail Order Sales and Internet Sales.
   a. Start by adding a blank row between Retail Sales and Mail Order Sales.
   b. Right-click in row R5 and choose Change Type to, and then Text.
      The TEXT dialog box opens.
   c. You will use this dialog box to add a blank row, so simply click OK without typing any text.

10. For Mail Order Sales:
   a. Drag the account 2300 from the Chart of Accounts into row R6 in the GL_ACCOUNT column on the matrix.
   b. Right-click 2300 and select Row Properties.
      The TAG dialog box opens.
   c. Under Children, select Show only children from the drop-down list.
   d. Leave Display children’s caption checked and click OK.
   e. To add an underline, right-click in row R7 and choose Change Type to, and then Bar.
   f. When the BAR dialog box opens, click OK.
   g. Drag account 2300 into row R8 to create the summary row. Since the selection you want is the default (Show selected item), this is all you need to do. (You can also right-click 2300 to open the TAG dialog box and click OK to confirm the default.)
11. For Internet Sales:
   a. Drag account 2400 from the Chart of Accounts into row R10. This is a quick way to create a blank row. Notice that TEXT appears in the Row Type column for row R9.
   b. Right-click 2400 and select *Row Properties*.
      The TAG dialog box opens.
   c. Under Children, select *Show only children* from the drop-down list.
   d. Leave Display children’s caption checked and click *OK*.
   e. To add an underline, right-click in row R11 and choose *Change Type to*, and then *Bar*.
   f. When the BAR dialog box opens, click *OK*.
   g. Drag account 2400 into the matrix in row R12 to create the summary row. Since the selection you want is the default (Show selected item), this is all you need to do.

12. Add a consolidated row for total Sales Revenue. Drag account 2100 (Sales Revenue) into row R14. Since you want to show the selected item, there is no need to open the TAG dialog box.

Once again, a blank TEXT row is automatically added in row R13.

The Financial Report Painter appears, as shown in the following image.
13. Save your work, then click *Run* to see the output.

### Century Corporation
#### 2010 Income Statement

<table>
<thead>
<tr>
<th>Revenue:</th>
<th>Actual</th>
<th>Planned</th>
<th>Increase/(Decrease)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retail - Television</td>
<td>7,971,992</td>
<td>7,904,000</td>
<td>67,992.00</td>
</tr>
<tr>
<td>Retail - Stereo</td>
<td>3,063,729</td>
<td>3,040,000</td>
<td>23,729.00</td>
</tr>
<tr>
<td>Retail - Video Player</td>
<td>6,332,185</td>
<td>6,080,000</td>
<td>252,185.00</td>
</tr>
<tr>
<td>Retail - Computer</td>
<td>1,204,955</td>
<td>1,216,000</td>
<td>(11,045.00)</td>
</tr>
<tr>
<td>Retail - Video Camera</td>
<td>30,782,323</td>
<td>30,400,000</td>
<td>382,323.00</td>
</tr>
<tr>
<td>Retail Sales</td>
<td>49,355,184</td>
<td>48,540,000</td>
<td>715,184.00</td>
</tr>
<tr>
<td>Mail Order - Television</td>
<td>30,892</td>
<td>0</td>
<td>30,892.00</td>
</tr>
<tr>
<td>Mail Order - Stereo</td>
<td>122,418</td>
<td>121,600</td>
<td>818.00</td>
</tr>
<tr>
<td>Mail Order - Video Player</td>
<td>122,454</td>
<td>121,600</td>
<td>854.00</td>
</tr>
<tr>
<td>Mail Order - Computer</td>
<td>2,427,475</td>
<td>2,432,000</td>
<td>(4,522.00)</td>
</tr>
<tr>
<td>Mail Order - Video Camera</td>
<td>4,196,174</td>
<td>4,012,800</td>
<td>183,374.00</td>
</tr>
<tr>
<td>Mail Order Sales</td>
<td>6,899,646</td>
<td>6,688,000</td>
<td>211,646.00</td>
</tr>
<tr>
<td>Internet - Television</td>
<td>30,513</td>
<td>0</td>
<td>30,513.00</td>
</tr>
<tr>
<td>Internet - Stereo</td>
<td>298,051</td>
<td>304,000</td>
<td>(5,949.00)</td>
</tr>
<tr>
<td>Internet - Video Player</td>
<td>310,499</td>
<td>304,000</td>
<td>6,499.00</td>
</tr>
<tr>
<td>Internet - Computer</td>
<td>1,204,687</td>
<td>1,216,000</td>
<td>(11,313.00)</td>
</tr>
<tr>
<td>Internet - Video Camera</td>
<td>4,264,110</td>
<td>4,256,000</td>
<td>8,140.00</td>
</tr>
<tr>
<td>Internet Sales</td>
<td>5,107,890</td>
<td>6,080,000</td>
<td>27,110.00</td>
</tr>
<tr>
<td>Sales Revenue</td>
<td>62,362,490</td>
<td>61,408,000</td>
<td>954,490.00</td>
</tr>
</tbody>
</table>

Notice that computer sales are below expectations in all revenue categories. You have completed the Revenue rows in the income statement.

14. You will now subtract Cost of Goods Sold from Sales Revenue to determine Gross Margin. Since these exist as consolidated values in the data mart, no calculation is required. You simply drag the accounts into the matrix.

   a. Start by dragging account 2500 (Cost of Goods Sold) into row R15.
   b. Make row R16 a bar row.
c. Drag account 2000 (Gross Margin) into row R17. The new section of the matrix appears as shown in the following image.
15. Save your work, then run the income statement to see the effect.

<table>
<thead>
<tr>
<th>Revenue:</th>
<th>Actual</th>
<th>Planned</th>
<th>Increase/(Decrease)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retail - Television</td>
<td>7,971,992.</td>
<td>7,904,000.</td>
<td>67,992.00</td>
</tr>
<tr>
<td>Retail - Stereo</td>
<td>3,063,729.</td>
<td>3,040,000.</td>
<td>23,729.00</td>
</tr>
<tr>
<td>Retail - Video Player</td>
<td>6,322,185.</td>
<td>6,080,000.</td>
<td>252,185.00</td>
</tr>
<tr>
<td>Retail - Computer</td>
<td>1,204,955.</td>
<td>1,216,000.</td>
<td>(11,045.00)</td>
</tr>
<tr>
<td>Retail - Video Camera</td>
<td>30,782,323.</td>
<td>30,400,000.</td>
<td>382,323.00</td>
</tr>
</tbody>
</table>

| Retail Sales                    | 49,355,184.  | 48,640,000. | 715,184.00          |
| Mail Order - Television         | 30,892.      | 0.          | 30,892.00           |
| Mail Order - Stereo             | 122,418.     | 121,600.    | 818.00              |
| Mail Order - Video Player       | 122,454.     | 121,600.    | 854.00              |
| Mail Order - Computer           | 2,417,408.   | 2,432,000.  | (4,522.00)          |
| Mail Order - Video Camera       | 4,051,474.   | 4,012,800.  | 38,674.00           |

| Mail Order Sales                | 6,899,416.   | 6,688,000.  | 211,416.00          |
| Internet - Television           | 30,513.      | 0.          | 30,513.00           |
| Internet - Stereo               | 298,051.     | 304,000.    | (5,949.00)          |
| Internet - Video Player         | 310,479.     | 304,000.    | 6,499.00            |
| Internet - Computer             | 1,204,687.   | 1,216,000.  | (11,313.00)         |
| Internet - Video Camera         | 4,764,140.   | 4,756,000.  | 8,140.00            |

| Internet Sales                  | 6,107,890.   | 6,080,000.  | 27,890.00           |
| Sales Revenue                   | 62,362,490.  | 61,408,000. | 954,490.00          |
| Cost Of Goods Sold              | 36,723,257.  | 36,480,000. | 243,257.00          |
| Gross Margin                    | 25,639,233.  | 24,928,000. | 711,233.00          |

16. All that is left is to include Expenses and calculate Profits.
   a. Scroll down in the For field values panel to see Total Operating Expenses.
   b. Expand accounts 3000 (General and Admin Expenses) and 5000 (TotalR + DCosts) to see what comprises each one. (To make the list of For field values more manageable, you can collapse the revenue accounts.)
   c. Although expenses are broken down into several categories, you are really interested in total expenses so drag account 3000 into row R19. (R18 becomes a TEXT row.)
   d. Drag account 5000 into the same cell. Notice that both values appear in the GL_ACCOUNT column connected by the keyword OR, which, in Boolean logic, indicates that the two tag values will be summed.

17. Modify the title so that it reflects the combined data.
a. Instead of opening the TEXT dialog box, double-click the Title cell in row R19 to select it, then type the following directly into the input box above the matrix:

**Operating Expenses and R + D Costs**

b. Click outside the cell to confirm the change.

18. Make R20 a bar row. This time, select the double line (==) in the BAR dialog box. (In HTML, it will appear as a broader single line.)


20. Finish the report with another double underline and save your work. The last section of the matrix looks like the following image:

21. Run the report one last time.
The 2010 Income Statement for Century Corporation appears as follows:

<table>
<thead>
<tr>
<th>Century Corporation</th>
<th>2010 Income Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Revenue:</strong></td>
<td></td>
</tr>
<tr>
<td>Retail - Televisions</td>
<td>2,975,901</td>
</tr>
<tr>
<td>Retail - Stoves</td>
<td>3,065,797</td>
</tr>
<tr>
<td>Retail - Video Games</td>
<td>6,332,185</td>
</tr>
<tr>
<td>Retail - Computers</td>
<td>1,204,955</td>
</tr>
<tr>
<td>Retail - Video Cameras</td>
<td>86,793,595</td>
</tr>
<tr>
<td><strong>Total Sales:</strong></td>
<td>95,205,184</td>
</tr>
<tr>
<td><strong>Cost of Goods Sold:</strong></td>
<td>63,640,060</td>
</tr>
<tr>
<td><strong>Gross Margin:</strong></td>
<td>31,565,124</td>
</tr>
<tr>
<td><strong>Operating Expenses and R+D Costs:</strong></td>
<td>21,900,070</td>
</tr>
<tr>
<td><strong>Profit Before Tax:</strong></td>
<td>3,793,953</td>
</tr>
</tbody>
</table>

From the final income statement, Century Corp can attribute profits to sales success in various areas, as well as to success in keeping costs below budget in 2010. Now you can claim success in completing the Income Statement tutorial.
Create the Income Statement in the Financial Report Painter
Before you start this tutorial, be sure to complete *Tutorial: Creating a Financial Report Using the Financial Report Painter* on page 115, which teaches some basic concepts and skills that are applicable in this tutorial. It also sets up a project called FINANCIAL that you will use in this exercise.

**In this chapter:**

- Tutorial Overview
- Create a Procedure Within the Project
- Join the Data Sources
- Create the Base Report in Report Painter
- Create the Income Statement in the Financial Report Painter

**Tutorial Overview**

Our fictional company, Century Corporation, needs another income statement. This one will resemble the one you created in the last exercise, but with three fundamental differences based on the structure of the data you will be using.

- In the first tutorial, both the hierarchy and the financial data were contained in the same data source. In this exercise, the hierarchy is defined in one Master File and the financial data in another. You will have to join the two to create the report.

- In the first tutorial, the data was structured as a preconsolidated data mart with rollups built into the data. In this exercise, the data is unconsolidated. Do the consolidations using options in the Financial Report Painter.

- In the first tutorial, the data mart resolved differences between data storage conventions in a typical General Ledger system and user expectations for how those values appear in output. In this exercise, you will resolve those differences yourself.

Once again, you will work in the local projects area of Developer Studio and run reports against the WebFOCUS Reporting Server on your local machine. Be sure to select the option that installs the local Reporting Server with Developer Studio.
Create a Procedure Within the Project

Create and name the new procedure and select the tool you will use to create the report.

1. Launch Developer Studio. The WebFOCUS Reporting Server is activated at the same time.
2. Expand the FINANCIAL project folder under Projects on localhost.
3. Open the Master Files folder and click the binoculars icon to expose the list of available data sources.
   a. Scroll down and locate CENTGL.MAS, then right-click and choose Add to project.
   b. Repeat this process for CENTSTMT.MAS.
   c. Click the binoculars icon again to toggle back to the list of Master Files that are active for the FINANCIAL project. You will see only the selected master files.
4. Click the Procedures folder. You will see the INCOME1 report you created in the first tutorial.
5. To start a new procedure, right-click the Procedures folder and choose New, and then Procedure. The Add Procedure dialog box opens.
6. Name the procedure INCOME2 and choose Report Painter from the Create with drop-down list.

![Add Procedure dialog box]

7. Click Open.
Before the Painter opens, the Open dialog box asks you to specify the data source you will be using to create your report.

8. Select centgl.mas and click Open.

Report Painter opens, displaying the fields of the CENTGL data source in the Object Inspector.

CENTGL is one of two data sources you will need for your report. Your next task is to join the two data sources so that the fields from both will be available to you.

**Join the Data Sources**

Join CENTGL, CENTSTMT, and CENTSYSF to make all of their fields available for reporting.

- CENTGL.MAS contains a chart of accounts hierarchy.
- CENTSTMT.MAS and CENTSYSF.MAS contain detail-level financial data.

You must specify the chart of accounts as the first file to be joined. (This is sometimes referred to as the host file.)
1. Click the *Join* button from the Setup toolbar. The Join tool opens, listing the fields from `CENTGL.MAS` in a resizable window.

![Image of Join window with fields listed]

2. Click the *Add File (+)* icon on the toolbar. Select the `CENTSTMT` and `CENTSYSF` Master Files and click *Open*.

Notice that the data sources are automatically joined at their respective `GL_ACCOUNT` and `SYS_ACCOUNT` fields. (If you do not see the join arrow, resize each window.)

![Image of joined data sources]

3. Run the Join and close the tool. When prompted, update the procedure. The fields from `CENTGL`, `CENTSTML`, and `CENTSYSF` are listed in the Object Inspector.
Before you start to create the report, examine the data you will be using, along with some accounting concepts that will influence your work. Proceed to Data Detour: Working With Positive and Negative Values in Financial Data on page 153.

**Data Detour: Working With Positive and Negative Values in Financial Data**

The CENTSYSF Master File, which you joined to CENTGL (described in Join the Data Sources on page 151), describes account transactions that contains unconsolidated detail-level financial data.

In most General Ledger systems, data is maintained internally as follows:

- Revenues are represented as negative (-) values.
- Expenses are represented as positive (+) values.

CENTSYSF.FOC (the data source described by CENTSYSF. MAS) complies with these conventions. Therefore, if you were to report directly against this raw data, you would see the negative values for revenue accounts in your report, as illustrated in the example Displaying Revenues and Expenses Without Adjusting Positive and Negative Values on page 154.

This is contrary to the expectations of accountants and other financial analysts since expenses are generally thought of as negative values (money going out) and revenues as positive values (money coming in).

In this exercise, you will learn a simple technique that uses a function called FMLINFO, which converts data to conform to user expectations in the report output, while complying with internal General Ledger system conventions.

For the income statement in this exercise, you will be using fields from CENTGL and CENTSYSF: GL_ACCOUNT, the For field that will give you row-by-row control of account values, and NAT_AMOUNT (Actual) and NAT_BUDGET (Planned), each of which will require manipulation of the negative values to give you the results you need in your output.
The CENTGL Master File describes the Chart of Accounts hierarchy. The field GL_ACCOUNT_PARENT is the parent field in the hierarchy. The field GL_ACCOUNT is the hierarchy field. The field GL_ACCOUNT_CAPTION can be used as the descriptive caption for the hierarchy field. (The hierarchy will come into play when you begin to structure the income statement in the Financial Report Painter.)

FILE=CENTGL,SUFFIX=FOC
SEGNAME=ACCOUNTS,SEGTYPE=S01
FIELDNAME=GL_ACCOUNT, ALIAS=GLACCT, FORMAT=A7,
   TITLE='Ledger, Account', FIELDTYPE=I, $
FIELDNAME=GL_ACCOUNT_PARENT, ALIAS=GLPAR, FORMAT=A7,
   TITLE=Parent, PROPERTY=PARENT_OF, REFERENCE=GL_ACCOUNT, $
FIELDNAME=GL_ACCOUNT_TYPE, ALIAS=GLTYPE, FORMAT=A1, TITLE=Type,$
FIELDNAME=GL_ROLLUP_OP, ALIAS=GLROLL, FORMAT=A1, TITLE=Op, $
FIELDNAME=GL_ACCOUNT_LEVEL, ALIAS=GLLEVEL, FORMAT=I3, TITLE=Lev, $
FIELDNAME=GL_ACCOUNT_CAPTION, ALIAS=GLCAP, FORMAT=A30,
   TITLE=Caption, PROPERTY=CAPTION, REFERENCE=GL_ACCOUNT, $
FIELDNAME=SYS_ACCOUNT, ALIAS=ALINE, FORMAT=A6,
   TITLE='System, Account, Line', MISSING=ON, $

Example: Displaying Revenues and Expenses Without Adjusting Positive and Negative Values

This example is not part of the tutorial. It is included to illustrate the problem you are about to solve.

The following report was produced from the joined data sources, CENTGL, CENTSTMT, and CENTSYSF.

You added three fields in Report Painter, GL_ACCOUNT, NAT_AMOUNT, and NAT_BUDGET.
In the Financial Report Painter, you added detail and summary rows to the matrix for one Revenue account, 2200 (Retail Sales), and one Expense account, 3100 (Selling Expenses).

When you ran the report, Revenue (both the detail and summary rows) were preceded by minus (-) signs, while Expenses (both detail and summary rows) are shown as positive values.
Create the Base Report in Report Painter

Resume the tutorial by creating the base report.

You will be choosing fields from the first ACCOUNTS folder in the Object Inspector. This is where the financial data resides. (The fields in CENTGL structure the hierarchy in the Financial Report Painter.)

1. Drag GL_ACCOUNT from the first ACCOUNTS folder into the Report Painter window. (Since the name Ledger Account is defined in the Master File, it appears as the column title.)

2. To assign a report title, type the following in the Page Heading input area, followed by a blank line to leave space between the heading and the report.

   Century Corporation Income Statement
   For Period Ended: June 30, 2002

3. Highlight the heading text, right-click and choose Justify, and then Center.

4. Select Ledger Account and click the For button.

   That is all you are going to do in Report Painter. You can do everything else from the Financial Report Painter.

5. Save your work and click the Matrix tab to open the Financial Report Painter.


Create the Income Statement in the Financial Report Painter

The For field, GL_ACCOUNT, is the only column listed in your report. You will be adding other columns to this report in the Financial Report Painter.
Initially, the hierarchy is collapsed in the For field values panel. You will expand various sections of the hierarchy as you begin to populate the rows of the matrix.

In the rest of this exercise, you will match revenue and expenses in the income statement, first showing all the revenue earned and then deducting from the revenue all the expenses (the costs of doing business) that you incurred in producing that revenue.

**Add Calculated Columns to the Income Statement**

Your first task is to add three new columns to the report.

- **Actual** and **Planned**. Rather than including these fields directly from the data source, you will calculate their values using a specialized function called FMLINFO, which will reverse the negative values for revenue in the report output.

- **Variance**. You will reflect the difference between the calculated Actual and Planned amounts in the previous two columns. This information will help Century Corporation plan sales initiatives and spending reductions, as needed.

You will calculate all three columns from the Computes tab of the Report Options dialog box, which you can access directly from the Financial Report Painter.

1. Place the cursor in the column labeled 1 on the matrix, then select *Computes* from the Report menu.

   The Report Options dialog box opens at the Computes tab.
a. In the Fields input box, name the field *Actual*, then click the *Format* button and choose *B* from the Edit Options list, to display parentheses around any negative values in the output. Click OK to close the Format dialog box.

You will now create an expression that calculates Actual as a new field in which negative revenue values are converted to positive values.

b. Formulate the expression in the input box using a combination of keys on the Compute calculator, fields from the Field list, which you can access from the *Fields* button, and type characters:

\[
\text{IF FMLINFO('FORVALUE', 'A7') LT '2500' THEN NAT_AMOUNT * (-1) ELSE NAT_AMOUNT}
\]

**Tip:** You must type the following part of the expression:

\[
\text{FMLINFO('FORVALUE', 'A7')}
\]

where:

**FORVALUE**

Is the General Ledger account, in this case, GL_ACCOUNT, that you are using as the For field in the report.

**A7**

Is the format of that field, as defined in the Master File.
Note that the FMLINFO function is only available to calculate values from the Compute tab and is not included on the generic list of functions that you access from the Functions button.

c. Click Apply to add the calculated value to the Fields list. Then click OK to add Actual as a column on the matrix.

d. Click the Save icon.

Before you repeat this process for Planned, reflect on what FMLINFO does. For accounts less than 2500, FMLINFO multiplies the Actual value by -1, thereby converting the negative value into a positive value for those accounts.

2. In the FOR field values panel, expand all the accounts with number less than 2500 in For field values panel.

You will see that accounts 2100 through 2450 comprise the Revenue accounts in the hierarchy. Accounts 1000 (Profit Before Tax) and 2000 (Gross Margin) are "net" calculations based on the revenue accounts and, therefore, subject to the same sign conventions. Without the adjustment provided by FMLINFO, all of these accounts would generate negative numbers in the Actual and Planned columns, as illustrated in Displaying Revenues and Expenses Without Adjusting Positive and Negative Values on page 154.
So, you need to understand the structure of the accounts hierarchy with which you are working to apply the FMLINFO calculation to the correct accounts.

3. Place the cursor in the column labeled 2 and choose Computes from the Report menu.
   The Computes tab shows your previous expression. Since you will need to use a nearly identical expression to generate the Planned field, select the entire expression and press Ctrl + C (Copy).
   a. Click the New button. The expression box is cleared.
      Name the field Planned in the Fields box. Accept the default field format, D12.2.
   b. Click in the expression box and press Ctrl + V (Paste). The previous expression is pasted into the input box. Change $NAT\_AMOUNT$ to $NAT\_BUDGET$ in both places.
   c. Click Apply to add the calculated value to the Fields list and OK to add Planned as a column on the matrix.
   d. Click the Save icon.
      Create the third column, Variance, and rename it Increase/(Decrease).

4. Select the Computes icon from the Setup toolbar. The Report Options dialog box opens at the Computes tab.
   a. Click the New button to clear the input box.
b. Name the field *Variance* in the Fields input box, then click the *Format* button and choose *B* from the Edit Options list, to display parentheses around any negative values in the output. Click *OK* to close the Format dialog box.

c. Click the *Fields* button to open the Fields list. Click the *Computed Fields* folder to see the two calculated values, Actual and Planned.

d. Double-click *Actual* in the Fields list to enter it in the expression box, type a minus sign or click minus (-) on the Compute calculator, then double-click *Planned* in the Fields list. Click *Close* to close the Insert Field dialog box.

e. Click *Apply* to add the calculated value to the Fields list and *OK* to add the Variance as a column in the matrix.

f. To change the title of the Variance column to Increase/(Decrease), right-click *Variance* and choose *Options*. The Field Properties dialog box opens.

g. Click the *General* tab, then click the *Set Title* button.

The Title dialog box opens.

h. Type the following in the Title dialog box.

*Increase/ (Decrease)*

i. Click *OK* twice, then click the *Save* icon.
The matrix includes the three calculated columns and appears as follows:

![Matrix](image)

**Tip:** If you would like to add these formatting touches, you can right-click Increase/Decrease and select Options. The Field Properties dialog box will open at the Style tab. Then follow the instructions beginning with step 11 from *Create the Base Report in Report Painter* on page 156. When you are done, you will return to the Financial Report Painter.

You have finished structuring the columns in the report and will start adding rows to the matrix.

**Add Revenue Rows to the Income Statement**

Complete the Revenue section of the income statement.

1. Your report will display information about several revenue accounts so start by adding a Text row to the matrix, titled Revenue.
   a. In row R1, right-click in the Row Type column and select *Change Row Type to*, and then *Text*.
      The TEXT dialog box opens.
   b. Type the following:
      
      Revenue:
    
   c. Click OK.
      
      Revenue appears in the Title column on the matrix.

2. Make the text row bold.
   a. Right-click Revenue and select *Options*.
      The Field Properties dialog box opens at the Style tab.
b. Make Row the active object (upper-left).

c. Click the Select Font button under Graphical (upper-right).

    The Font dialog box opens.

d. Choose Bold as the Font Style and click OK.

e. Click OK on the Style tab to return to the matrix.

    Row R1 is identified as a TEXT row type, and Revenue is bold.

    You are now going to begin to add data values from the Chart of Accounts. The first step is expanding a section of the hierarchy to expose the levels of values.

3. In the For field values panel, expand account 2000 (Gross Margin), then expand 2100 (Sales Revenue), then 2200 (Retail Sales).

4. To complete the Revenue section of the report, you will add detailed information for the children in each revenue category, followed by a consolidated row for each category, and finally a consolidated row for all sales revenue.

    **Tip:** Some of these steps will seem very similar to those in the previous tutorial. Be on the alert for places where they are different. Remember that you will be explicitly consolidating data that was pre-consolidated in the data mart.

    a. Start by dragging the account 2200 (Retail Sales) from the Chart of Accounts to row R2 in the GL_ACCOUNT column on the matrix.

        Notice that TAG is entered in the Row Type column.

    b. Right-click account 2200 and select Row Properties.

        The TAG dialog box opens.

    c. Under Children, select Show only children from the drop-down list.
Notice that *Display children's caption* is checked by default.

These selections indicate that you want to see values for the children of Retail Sales, (Retail - Television, Retail - Stereo, and so on), but not the value of Retail Sales itself.

In addition, the captions (descriptive titles) for the children will appear in the output in place of their tag values.

d. Click **OK** to return to the matrix.

Notice that a plus (+) sign precedes 2200, indicating that it contains subordinate values that you can expand if you wish.

5. To ensure that you can use the 2200 account and other values in more than one row of the matrix, click the *Use Multiple Values* check box in the FML Report Properties area above the matrix.

6. Place an underscore under these values.

   a. Right-click in the *Row Type* column in row R3 and choose *Change row type to*, and then *Bar*.

   The BAR dialog box opens.
b. Click OK to accept the default underline character.

7. Drag account 2200 into row R4 on the matrix.
   a. Right-click 2200 and select Row Properties to open the TAG dialog box.
   b. Under Children, choose Show selected item consolidated.
This will roll up the data for the children of Retail Sales and display the total in a single row.

![TAG dialog box](image)

c. Click OK.

**Tip:** In the data mart tutorial, you selected *Show selected item* at this point, since the data you were using was pre-consolidated.

You are going to repeat this process for Mail Order Sales and Internet Sales.

8. For Mail Order Sales:

   a. Drag the account 2300 from the Chart of Accounts into row R6 in the GL_ACCOUNT column on the matrix, making row R5 a blank TEXT row.

   b. Right-click 2300 and select *Row Properties*.

      The TAG dialog box opens.

   c. Under Children, select *Show only children* from the drop-down list.

   d. Leave *Display children's caption* checked and click *OK*.

   e. To add an underline, right-click in row R7 and choose *Change row type* to, and then *Bar*.

   f. When the BAR dialog box opens, click *OK*. 
g. Drag account 2300 into row R8 to create the summary row.

h. Right-click 2300 and select Row Properties to open the TAG dialog box.

i. Under Children, choose Show selected item consolidated and click OK.

9. For Internet Sales:

   a. Drag account 2400 from the Chart of Accounts into row R10, making row R9 a blank TEXT row.

   b. Right-click 2400 and select Row Properties.

      The TAG dialog box opens.

   c. Under Children, select Show only children.

   d. Leave Display children's caption checked and click OK.

   e. To add an underline, right-click in row R11 and choose Change row type to, and then Bar.

   f. When the BAR dialog box opens, click OK.

   g. Drag account 2400 into row R12 to create the summary row.

   h. Right-click 2400 and select Row Properties to open the TAG dialog box.

   i. Under Children, choose Show selected item consolidated and click OK.

10. Add a row that rolls up all of the Sales Revenue data.

    a. Drag account 2100 (Sales Revenue) into row R14, making R13 a blank text row.

    b. Right-click 2100 and select Row Properties.

       The TAG dialog box opens.

    c. Under Children, choose Show selected item consolidated.

    d. Leave Display children's caption checked and click OK.
The Financial Report Painter appears as follows.

The plus signs (+) indicate the presence of children, which you can expose if you wish. The sigma signs indicate consolidated data.
11. Save your work and run the report. The output is:

**Century Corporation Income Statement**  
**For Period Ended: June 30, 2010**

<table>
<thead>
<tr>
<th>Revenue:</th>
<th>Actual</th>
<th>Planned</th>
<th>Increased/ (Decrease)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retail - Television</td>
<td>7,971,992.00</td>
<td>7,904,000.00</td>
<td>67,992.00</td>
</tr>
<tr>
<td>Retail - Stereo</td>
<td>3,063,729.00</td>
<td>3,040,000.00</td>
<td>23,729.00</td>
</tr>
<tr>
<td>Retail - Video Player</td>
<td>6,332,185.00</td>
<td>6,080,000.00</td>
<td>252,185.00</td>
</tr>
<tr>
<td>Retail - Computer</td>
<td>1,204,955.00</td>
<td>1,216,000.00</td>
<td>(11,045.00)</td>
</tr>
<tr>
<td>Retail - Video Camera</td>
<td>30,782,323.00</td>
<td>30,400,000.00</td>
<td>382,323.00</td>
</tr>
<tr>
<td>Retail Sales</td>
<td>49,355,184.00</td>
<td>48,640,000.00</td>
<td>715,184.00</td>
</tr>
<tr>
<td>Mail Order - Television</td>
<td>30,892.00</td>
<td>0.00</td>
<td>30,892.00</td>
</tr>
<tr>
<td>Mail Order - Stereo</td>
<td>122,418.00</td>
<td>121,600.00</td>
<td>818.00</td>
</tr>
<tr>
<td>Mail Order - Video Player</td>
<td>122,454.00</td>
<td>121,600.00</td>
<td>854.00</td>
</tr>
<tr>
<td>Mail Order - Computer</td>
<td>2,427,478.00</td>
<td>2,412,000.00</td>
<td>(4,522.00)</td>
</tr>
<tr>
<td>Mail Order - Video Camera</td>
<td>4,196,174.00</td>
<td>4,012,800.00</td>
<td>183,374.00</td>
</tr>
<tr>
<td>Mail Order Sales</td>
<td>6,899,416.00</td>
<td>6,688,000.00</td>
<td>211,416.00</td>
</tr>
<tr>
<td>Internet - Television</td>
<td>30,513.00</td>
<td>0.00</td>
<td>30,513.00</td>
</tr>
<tr>
<td>Internet - Stereo</td>
<td>298,051.00</td>
<td>304,000.00</td>
<td>(5,949.00)</td>
</tr>
<tr>
<td>Internet - Video Player</td>
<td>310,499.00</td>
<td>304,000.00</td>
<td>6,499.00</td>
</tr>
<tr>
<td>Internet - Computer</td>
<td>1,204,687.00</td>
<td>1,216,000.00</td>
<td>(11,313.00)</td>
</tr>
<tr>
<td>Internet - Video Camera</td>
<td>4,264,140.00</td>
<td>4,256,000.00</td>
<td>8,140.00</td>
</tr>
</tbody>
</table>

You have completed the Revenue section of the income statement.

Notice that revenues are represented as positive values in both the Actual and Planned columns.

Your next task is to add rows for Cost of Goods Sold and Gross Margin.
**Add Cost of Goods Sold and Gross Margin Rows to the Income Statement**

You are now going to add a row for Cost of Goods Sold and one for Gross Margin, which subtracts total Cost of Goods Sold from total Revenues.

1. Collapse the hierarchy for accounts 2000-2400 and expand the hierarchy for account 2500 (Cost of Goods Sold).

2. Make row R15 a blank TEXT row.

3. Make row R16 a TEXT row, but this time, instead of opening the TEXT dialog box, type the following text in the Title cell:
   
   **Cost of Goods Sold:**

4. Make the text row bold.
   
   a. Right-click *Cost of Goods Sold* and select *Options*.
   
      The Field Properties dialog box opens at the Style tab.
   
   b. Make *Row* the active object (upper-left).
   
   c. Click the *Select Font* button under Graphical (upper-right).
   
      The Font dialog box opens.
   
   d. Choose *Bold* as the Font Style and click *OK*.
   
   e. Click *OK* on the Style tab to return to the matrix.

5. Drag account 2500 (Cost of Goods Sold) into row R17 on the matrix.

   Notice that Variable Material Costs, one of the children of Cost of Goods Sold, has children of its own.

   a. Right-click 2500 and select *Row Properties*.
   
      The TAG dialog box opens.
   
   b. Under Children, select *Show only children to level* and specify Level 1.
   
      This indicates that you want to display only one level of the hierarchy. The children of Variable Material Costs will *not* be shown.
   
   c. Click the *Consolidate* check box.
   
      This ensures that although the children of Variable Material Costs will not be explicitly broken out in the report, the value on the Variable Material Costs line will reflect a roll up of the values of the children.
d. Leave *Display children's caption* checked and click *OK*.

![TAG dialog box](image)


7. Drag account 2500 (Cost of Goods Sold) into the matrix in row R19.
   a. Right-click 2500 and select *Row Properties*.
      The TAG dialog box opens.
   b. Under Children, choose *Show selected item consolidated*.
      This will consolidate the Cost of Goods Sold data for all children (including those of Variable Material Costs) in a single row.
   c. Leave *Display children's caption* checked and click *OK*.

8. Make row R20 a Bar row.

   a. Right-click 2000 and select *Row Properties*.
      The TAG dialog box opens.
   b. Under Children, choose *Show selected item consolidated*. 
c. Leave *Display children’s caption* checked and click OK.

d. On the matrix, click the plus sign (+) to expose the direct children of Cost of Goods Sold.

The matrix looks like the following image.
10. Save your work, run the report, and examine the output, which follows.

![Century Corporation Income Statement](image)

Notice that Century Corp has exceeded expectations in terms of keeping costs down and can report a healthy Gross Margin for the reporting period.
Add Expense and Profit Rows to the Income Statement

The last section of the income statement defines the expenses of Century Corp during this period and calculates profit before taxes (Profit= Gross Margin - Expenses).

1. Begin by making row R22 a blank TEXT row.

2. Make row R23 a TEXT row and type the following text in the Title cell:

   Expenses:

3. Make the text bold.
   a. Right-click Expenses and select Options.
      The Field Properties dialog box opens at the Style tab.
   b. Make Row the active object (upper-left).
   c. Click the Select Font button under Graphical (upper-right).
      The Font dialog box opens.
   d. Choose Bold as the Font Style and click OK.
   e. Click OK on the Style tab to return to the matrix.

4. Drag account 3000 (Total Operating Expenses) into row R24.
   a. Right-click 3000 and select Row Properties.
      The TAG dialog box opens.
   b. Under Children, choose Show selected item consolidated.
      Profit will be calculated based on total expenses so you need to display consolidated data for each expense category.

5. Drag account 5000 (Total R + D Costs) into row R26, making row R25 a blank TEXT row.
   a. Right-click 5000 and select Row Properties.
      The TAG dialog box opens.
   b. Once again, under children, choose Show selected item consolidated.

6. Separate Expenses from the Profit calculation coming next with a heavier underscore.
   Create a Bar in row R27, but this time select ===== in the BAR dialog box.

7. Drag account 1000 (Profit Before Taxes) into row R28.
   a. Right-click 1000 and select Row Properties.

Create the Income Statement in the Financial Report Painter
The TAG dialog box opens.

b. Under Children, choose *Show selected item consolidated*.

8. Finish the report with another double underline and save your work.

The lower section of the matrix looks like the following image.

9. Save your work and run the report.
The consolidated Century Corporation Income Statement shows an Actual Profit that significantly exceeds expectations.

### Century Corporation Income Statement
For Period Ended: June 30, 2010

<table>
<thead>
<tr>
<th>Revenue:</th>
<th>Actual</th>
<th>Planned</th>
<th>Increased/Decrease</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retail - Television</td>
<td>7,971,992.00</td>
<td>7,904,000.00</td>
<td>67,992.00</td>
</tr>
<tr>
<td>Retail - Stereo</td>
<td>3,063,729.00</td>
<td>3,040,000.00</td>
<td>23,729.00</td>
</tr>
<tr>
<td>Retail - Video Player</td>
<td>6,322,185.00</td>
<td>6,080,000.00</td>
<td>252,185.00</td>
</tr>
<tr>
<td>Retail - Computer</td>
<td>1,204,958.00</td>
<td>1,216,000.00</td>
<td>(11,042.00)</td>
</tr>
<tr>
<td>Retail - Video Camera</td>
<td>30,782,323.00</td>
<td>30,400,000.00</td>
<td>382,323.00</td>
</tr>
<tr>
<td>Total Retail Sales</td>
<td>49,355,184.00</td>
<td>48,640,000.00</td>
<td>715,184.00</td>
</tr>
<tr>
<td>Mail Order - Television</td>
<td>30,892.00</td>
<td>.00</td>
<td>30,892.00</td>
</tr>
<tr>
<td>Mail Order - Stereo</td>
<td>122,418.00</td>
<td>121,600.00</td>
<td>818.00</td>
</tr>
<tr>
<td>Mail Order - Video Player</td>
<td>122,454.00</td>
<td>121,600.00</td>
<td>854.00</td>
</tr>
<tr>
<td>Mail Order - Computer</td>
<td>2,427,478.00</td>
<td>2,432,000.00</td>
<td>(4,522.00)</td>
</tr>
<tr>
<td>Mail Order - Video Camera</td>
<td>4,196,174.00</td>
<td>4,012,800.00</td>
<td>183,374.00</td>
</tr>
<tr>
<td>Mail Order Sales</td>
<td>6,899,416.00</td>
<td>6,688,000.00</td>
<td>211,416.00</td>
</tr>
<tr>
<td>Internet - Television</td>
<td>30,513.00</td>
<td>.00</td>
<td>30,513.00</td>
</tr>
<tr>
<td>Internet - Stereo</td>
<td>298,051.00</td>
<td>304,000.00</td>
<td>(5,949.00)</td>
</tr>
<tr>
<td>Internet - Video Player</td>
<td>310,499.00</td>
<td>304,000.00</td>
<td>6,499.00</td>
</tr>
<tr>
<td>Internet - Computer</td>
<td>1,204,687.00</td>
<td>1,216,000.00</td>
<td>(11,313.00)</td>
</tr>
<tr>
<td>Internet - Video Camera</td>
<td>4,264,140.00</td>
<td>4,256,000.00</td>
<td>8,140.00</td>
</tr>
<tr>
<td>Internet Sales</td>
<td>6,107,890.00</td>
<td>6,080,000.00</td>
<td>27,890.00</td>
</tr>
<tr>
<td>Sales Revenue</td>
<td>62,362,490.00</td>
<td>61,408,000.00</td>
<td>954,490.00</td>
</tr>
</tbody>
</table>

### Costs of Goods Sold

<table>
<thead>
<tr>
<th>Costs of Goods Sold</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable Material Costs</td>
<td>27,438,625.00</td>
<td>27,360,000.00</td>
<td>78,625.00</td>
</tr>
<tr>
<td>Direct Labor</td>
<td>6,176,900.00</td>
<td>6,080,000.00</td>
<td>96,900.00</td>
</tr>
<tr>
<td>Fixed Costs</td>
<td>3,107,742.00</td>
<td>3,040,000.00</td>
<td>67,742.00</td>
</tr>
<tr>
<td>Total Cost of Goods Sold</td>
<td>36,723,267.00</td>
<td>36,480,000.00</td>
<td>243,267.00</td>
</tr>
</tbody>
</table>

### Gross Margin

| Gross Margin                   | 25,639,223.00 | 24,928,000.00  | 711,223.00       |

### Expenses:

| Total Operating Expenses       | 20,032,931.00 | 20,216,000.00 | (183,069.00)     |
| Total R&D Costs                | 1,867,139.00  | 1,824,000.00  | 43,139.00        |
| Profit Before Tax              | 3,739,153.00  | 2,888,000.00  | 851,153.00       |
The Financial Modeling Language (FML) is designed for the special needs associated with creating, calculating, and presenting financially oriented data, such as balance sheets, consolidations, or budgets. These reports are distinguished from other reports because calculations are inter-row, as well as inter-column, and each row or line represents a unique entry or series of entries that can be aggregated directly from the input data or calculated as a function of the data.

In this appendix:

- Reporting With FML
- Creating Rows From Data
- Supplying Data Directly in a Request
- Performing Inter-Row Calculations
- Referring to Rows in Calculations
- Referring to Columns in Calculations
- Referring to Rows and Columns in Calculations
- Referring to Cells in Calculations
- Using Functions in RECAP Calculations
- Inserting Rows of Free Text
- Adding a Column to an FML Report
- Creating a Recursive Model
- Reporting Dynamically From a Hierarchy
- Customizing a Row Title
- Formatting an FML Report
- Suppressing the Display of Rows
- Saving and Retrieving Intermediate Report Results
- Creating HOLD Files From FML Reports
Reporting With FML

FML is an integrated extension of the TABLE command. By adding the FOR phrase and the RECAP command, you can handle an expanded range of applications.

**Note:** MORE is not supported in FML requests.

In conjunction with Dialogue Manager, FML can evaluate "what if" scenarios and develop complete decision support systems. These systems can take advantage of business intelligence features, such as statistical analysis and graphics, in addition to standard financial statements.

Procedures using FML are not hard-wired to the data. As in any other report request, they can easily be changed. FML includes the following facilities:

- **Row and column formatting.** You can specify results in a row-by-row, column-by-column fashion. For more information, see *Performing Inter-Row Calculations* on page 193.

- **Intermediate results.** You can post FML results to an external file and pick them up at a later time for analysis. This is useful when intermediate results are developed and a final procedure consolidates the results later. For more information, see *Saving and Retrieving Intermediate Report Results* on page 258.

- **Inline data entry.** FML enables you to specify constants from within the procedure, in addition to the data values retrieved from your data source. For more information, see *Supplying Data Directly in a Request* on page 192.

- **Recursive reporting.** You can produce reports where the results from the end of one time period or column become the starting balance in the next. For example, you can use recursive reports to produce a cash flow projection. For more information, see *Creating a Recursive Model* on page 215.

- **Dynamic reporting from a chart of accounts or a similar hierarchy of information.** You can create a report that changes as the organization of information changes, ensuring that you automatically retrieve information that reflects the latest structure and its values. There is no need to alter either the Master File or the report request. For more information, see *Reporting Dynamically From a Hierarchy* on page 216.
Example: Sample FML Request

This example produces a simple asset sheet, contrasting the results of two years. It illustrates many key features of the Financial Modeling Language (FML). Numbers to the left of the procedure lines correspond to explanations that follow the request.

```
TABLE FILE FINANCE
HEADING CENTER
"COMPARATIVE ASSET SHEET </2"
SUM AMOUNT ACROSS HIGHEST YEAR
WHERE YEAR EQ '1983' OR '1982'
1. FOR ACCOUNT
2. 1000 AS 'UTILITY PLANT' LABEL UTP OVER
2. 1010 TO 1050 AS 'LESS ACCUMULATED DEPRECIATION' LABEL UTPAD OVER
3. BAR OVER
4. RECAP UTPNET = UTP-UTPAD; AS 'TOTAL PLANT-NET' OVER
   BAR OVER
   2000 TO 3999 AS 'INVESTMENTS' LABEL INV OVER
5. "CURRENT ASSETS"
   4000 AS 'CASH' LABEL CASH OVER
   5000 TO 5999 AS 'ACCOUNTS RECEIVABLE-NET' LABEL ACR OVER
   6000 AS 'INTEREST RECEIVABLE' LABEL ACI OVER
   6500 AS 'FUEL INVENTORY' LABEL FUEL OVER
   6600 AS 'MATERIALS AND SUPPLIES' LABEL MAT OVER
   6900 AS 'OTHER' LABEL MISC OVER
   BAR OVER
   RECAP TOTCAS=CASH+ACR+ACI+FUEL+MAT+MISC; AS 'TOTAL CURRENT ASSETS' OVER
   BAR OVER
   7000 AS 'DEFERRED DEBITS' LABEL DEFDB OVER
```
1. FOR and OVER are FML phrases that enable you to structure the report on a row-by-row basis.

2. LABEL assigns a variable name to a row item for use in a RECAP calculation.
   1000 and 1010 TO 1050 are tags that identify the data values of the FOR field, ACCOUNT in the FINANCE data source. A report row can be associated with a tag that represents a single data value (like 1000), multiple data values, or a range of values (like 1010 TO 1050).

3. BAR enables you to underline a column of numbers before performing a RECAP calculation.

4. The RECAP command creates a new value based on values already identified in the report with LABEL. In this case, the value UTPNET is derived from UTP and UTPAD and is renamed TOTAL PLANT-NET with an AS phrase to provide it with greater meaning in the report.

5. Free text can be incorporated at any point in an FML report, similar to underlines.

6. Notice that this RECAP command derives a total (TOTAL ASSETS) from values retrieved directly from the data source, and from values derived from previous RECAP computations (UTPNET and TOTCAS).
The output is shown as follows.

**COMPARATIVE ASSET SHEET**

<table>
<thead>
<tr>
<th></th>
<th>1983</th>
<th>1982</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>UTILITY PLANT</strong></td>
<td>1,430,903</td>
<td>1,294,611</td>
</tr>
<tr>
<td>LESS ACCUMULATED DEPRECIATION</td>
<td>249,504</td>
<td>213,225</td>
</tr>
<tr>
<td><strong>TOTAL PLANT-NET</strong></td>
<td>1,181,399</td>
<td>1,081,386</td>
</tr>
<tr>
<td><strong>INVESTMENTS</strong></td>
<td>818</td>
<td>5,639</td>
</tr>
<tr>
<td><strong>CURRENT ASSETS</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CASH</td>
<td>4,938</td>
<td>4,200</td>
</tr>
<tr>
<td>ACCOUNTS RECEIVABLE-NET</td>
<td>28,052</td>
<td>23,758</td>
</tr>
<tr>
<td>INTEREST RECEIVABLE</td>
<td>15,945</td>
<td>10,206</td>
</tr>
<tr>
<td>FUEL INVENTORY</td>
<td>35,158</td>
<td>45,643</td>
</tr>
<tr>
<td>MATERIALS AND SUPPLIES</td>
<td>16,099</td>
<td>12,909</td>
</tr>
<tr>
<td>OTHER</td>
<td>1,264</td>
<td>1,743</td>
</tr>
<tr>
<td><strong>TOTAL CURRENT ASSETS</strong></td>
<td>101,456</td>
<td>98,459</td>
</tr>
<tr>
<td><strong>DEFERRED DEBITS</strong></td>
<td>30,294</td>
<td>17,459</td>
</tr>
<tr>
<td><strong>TOTAL ASSETS</strong></td>
<td>1,313,967</td>
<td>1,202,943</td>
</tr>
</tbody>
</table>

***PRELIMINARY ASSET SHEET BASED ON UNAUDITED FIGURES***

**Creating Rows From Data**

A normal TABLE request sorts rows of a report according to the BY phrase you use. The data retrieved is sorted from either low-to-high or high-to-low, as requested. The rows may be limited by a screening phrase to a specific subset, but:

- They appear in a sort order.
- Rows appear only for values that are retrieved from the file.
You can only insert free text between rows when a sort field changes value, such as:

`ON DIVISION SUBFOOT`

You can only insert calculations between rows when a sort field changes value, such as:

`ON DIVISION RECAP`

In contrast, the FML FOR phrase creates a matrix in which you can structure your report row-by-row. This organization gives you greater control over the data that is incorporated into a report, and its presentation. You can:

- Report on specific data values for a field in a data source and combine particular data values under a common label, for use in calculations.
- Type data directly into the request to supplement data retrieved from the data source.
- Include text, underlines, and calculations at points in the report that are not related to sort breaks.
- Perform recursive processing, in which the result of an interim calculation is saved and then used as the starting point for a subsequent calculation.
- Suppress the display of rows for which no data is retrieved.
- Identify rows by labels and columns by numbers, addresses, and values so that you can point to the individual cells formed at each intersection (as on a spreadsheet).

**Syntax:**

How to Retrieve FOR Field Values From a Data Source

The syntax for specifying rows is:

```
FOR fieldname [AS 'coltitle'] value [OR value OR...] [AS 'text']
[LABEL label] OVER
.
.
.
[value [OR value ...]] [AS 'text'] [LABEL label]
END
```

where:

- `fieldname`
  
  Is the FOR field for the FML report.

- `coltitle`
  
  Is the column title for the FOR field on the report output.
value
Is the value (also known as a tag value) describing the data that is retrieved for this row of the report.

AS 'text'
Enables you to assign a name to a tag value, which replaces the tag value in the output. Enclose the text in single quotation marks.

label
Assigns a label to the row for reference in a RECAP expression. The label can be up to 66 characters and cannot have blanks or special characters. Each explicit label you assign must be unique.

Even if you assign an explicit label, the positional label (R1, R2, and so on) is retained internally.

By default, a tag value for a FOR field (like 1010) may be added only once to the FML matrix. However, if you wish to add the same value of a FOR field to the matrix more than once, you can turn on the FORMULTIPLE parameter (the default setting is OFF). For more information, see How to Use the Same FOR Field Value in Multiple Rows on page 188.

For more information about the FMLFOR, FMLLIST, and FMLINFO functions that return the tag values used in an FML request, see the Using Functions manual.
**Example:** Creating Rows From Values in a Data Source

Assume you have a simple data source with financial data for each corporate account, as follows:

<table>
<thead>
<tr>
<th>ACCOUNT</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1010</td>
<td>CASH ON HAND</td>
</tr>
<tr>
<td>1020</td>
<td>DEMAND DEPOSITS</td>
</tr>
<tr>
<td>1030</td>
<td>TIME DEPOSITS</td>
</tr>
<tr>
<td>1100</td>
<td>ACCOUNTS RECEIVABLE</td>
</tr>
<tr>
<td>1200</td>
<td>INVENTORY</td>
</tr>
</tbody>
</table>

Using the **FOR** phrase in FML, you can issue the following **TABLE** request in which each value of **ACCOUNT** is represented by a tag (1010, 1020, and so on), and displays as a separate row:

```
TABLE FILE LEDGER
SUM AMOUNT
FOR ACCOUNT
1010 OVER
1020 OVER
1030 OVER
1100 OVER
1200 END
```

The output is shown as follows.

<table>
<thead>
<tr>
<th>AMOUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1010  8,784</td>
</tr>
<tr>
<td>1020  4,494</td>
</tr>
<tr>
<td>1030  7,961</td>
</tr>
<tr>
<td>1100 18,829</td>
</tr>
<tr>
<td>1200 27,307</td>
</tr>
</tbody>
</table>

**Creating Rows From Multiple Records**

There are different ways to combine multiple values from your data sources into an FML report row. You can use:

- The **OR** phrase to sum the values of two or more tags in a single expression. For more information, see *How to Sum Values in Rows With the OR Phrase* on page 185.

- The **TO** phrase to identify a range of tag values on which to report. For more information, see *How to Identify a Range of Values With the TO Phrase* on page 186.
A mask to specify a group of tag values without having to name each one. For more information, see *How to Use Masking Characters to Retrieve Tag Values* on page 187.

By default, a FOR field value can only be included in a single row of an FML matrix. However, by turning on the FORMULTIPLE parameter, you can include the same data value in multiple rows in the FML matrix. For example, the same value can exist as a solitary value in one row, be part of a range in another row, and be used in a calculation in a third row. For more information, see *How to Use the Same FOR Field Value in Multiple Rows* on page 188.

In addition to these methods, you can extract multiple tags for a row from an external file.

**Syntax:**  
**How to Sum Values in Rows With the OR Phrase**

To sum the values of two or more tags in a single report row, use the OR phrase in the FOR phrase. The syntax is:

```
FOR fieldname
    value1 OR value2 [OR valuen...][AS 'text'] [LABEL label] [OVER]
```

where:

*fieldname*
   Is a field name in the data source.

*value1, value2, valuen*
   Are the tag values to be retrieved and summed.

*AS 'text'*
   Assigns a title to the combined tag values. Enclose the text in single quotation marks ('').

*label*
   Assigns a label to the row for reference in a RECAP expression. The label can be up to 66 characters and cannot have blanks or special characters. Each explicit label you assign must be unique.

Even if you assign an explicit label, the positional label (R1, R2, and so on) is retained internally.
Example:  
**Summing Values in Rows**

The following model sums the values of three tags (1010, 1020, 1030) as CASH.

```
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 OR 1020 OR 1030 AS 'CASH' OVER
1100 AS 'ACCOUNTS RECEIVABLE' OVER
1200 AS 'INVENTORY'
END
```

The output is shown as follows.

<table>
<thead>
<tr>
<th>AMOUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASH</td>
</tr>
<tr>
<td>ACCOUNTS RECEIVABLE</td>
</tr>
<tr>
<td>INVENTORY</td>
</tr>
</tbody>
</table>

**Syntax:**  
**How to Identify a Range of Values With the TO Phrase**

To sum the values of a range of tags in a single report row, use the TO phrase in the FOR phrase. The syntax is:

```
FOR fieldname
value1 TO value2 [AS 'text'] [LABEL label] [OVER]
```

where:

- `fieldname` is a field name in the data source.
- `value1` is the tag value at the lower limit of the range.
- `TO` is the required phrase.
- `value2` is the tag value at the upper limit of the range.
- `AS 'text'` assigns a title to the combined tag values. Enclose the text in single quotation marks ('').
- `label` assigns a label to the row for reference in a RECAP expression. The label can be up to 66 characters and cannot have blanks or special characters. Each explicit label you assign must be unique.
Even if you assign an explicit label, the positional label (R1, R2, and so on) is retained internally.

**Example:**  **Identifying a Range of Values**

Since CASH accounts in the LEDGER system are identified by the tags 1010, 1020, and 1030, you can specify the range 1010 to 1030:

```plaintext
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 TO 1030 AS 'CASH'
END
```

**Syntax:**  **How to Use Masking Characters to Retrieve Tag Values**

If the tag field has a character (alphanumeric) format, you can perform a masked match. Use the dollar sign character ($) as the mask. For instance,

```
A$$D
```

matches any four-character value beginning with A and ending with D. The two middle places can be any character. This is useful for specifying a whole group of tag values without having to name each one.

**Example:**  **Using Masking Characters to Match a Group of Tags**

In this example, the amounts associated with all four-character accounts that begin with 10, expressed with a mask as 10$$, are used to produce the CASH row of the report.

```plaintext
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
10$$ AS 'CASH' OVER
1100 AS 'ACCOUNTS RECEIVABLE' OVER
1200 AS 'INVENTORY'
END
```

The output is shown as follows.

<table>
<thead>
<tr>
<th></th>
<th>AMOUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASH</td>
<td>21,239</td>
</tr>
<tr>
<td>ACCOUNTS RECEIVABLE</td>
<td>18,829</td>
</tr>
<tr>
<td>INVENTORY</td>
<td>27,307</td>
</tr>
</tbody>
</table>
**Syntax: How to Use the Same FOR Field Value in Multiple Rows**

You can use the same value of a FOR field in many separate rows (whether alone, as part of a range, or in a calculation) by including the following syntax before or within an FML request.

```
SET FORMULTIPLE={ON|OFF}
```

or

```
ON TABLE SET FORMULTIPLE {ON|OFF}
```

where:

**ON**

Enables you to reference the same value of a FOR field in more than one row in an FML request.

With FORMULTIPLE set to ON, a value retrieved from the data source is included on every line in the report output for which it matches the tag references.

**OFF**

Does not enable you to include the same value in multiple rows. OFF is the default value.

With FORMULTIPLE set to OFF, multiple tags referenced in any of these ways (OR, TO, *) are evaluated first for an exact reference or for the end points of a range, then for a mask, and finally within a range. For example, if a value is specified as an exact reference and then as part of a range, the exact reference is displayed. Note that the result is unpredictable if a value fits into more than one row whose tags have the same priority (for example, an exact reference and the end point of a range).

For more information, see *Reporting Dynamically From a Hierarchy* on page 216.
Example: Referencing the Same Value in More Than One Row

This request retrieves the tag values for accounts 1010, 1020, and 1030, and lists corresponding values individually. It then aggregates the same values and displays the sum as TOTAL CASH. Similarly, the tag values for accounts 1100 and 1200 displays as detail items, and then summarized as TOTAL NON-CASH ASSETS.

```
SET FORMULTE=ON
TABLE FILE LEDGER
SUM AMOUNT
FOR ACCOUNT
1010 AS 'CASH ON HAND'    OVER
1020 AS 'DEMAND DEPOSITS' OVER
1030 AS 'TIME DEPOSITS'   OVER
BAR                  OVER
1010 OR 1020 OR 1030 AS 'TOTAL CASH'  OVER
" "                  OVER
1100 AS 'ACCOUNTS RECEIVABLE' OVER
1200 AS 'INVENTORY'       OVER
BAR                  OVER
1100 TO 1200 AS 'TOTAL NON-CASH ASSETS'
END
```

The output is shown as follows.

```
AMOUNT
-----
CASH ON HAND       8,784
DEMAND DEPOSITS    4,494
TIME DEPOSITS      7,961
TOTAL CASH         21,239
ACCOUNTS RECEIVABLE 18,829
INVENTORY          27,307
TOTAL NON-CASH ASSETS 46,136
```
Example: Using Tags From External Files

In this example, the values for a row of the FML report come from an external file called CASHSTUF, which contains the following tags.

1010
1020
1030

The following TABLE request uses the tag values from the external file, summing the amounts in accounts 1010, 1020, and 1030 into the CASH row of the FML report.

```
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
(CASHSTUF) AS 'CASH' OVER
1100 AS 'ACCOUNTS RECEIVABLE'
END
```

Notice that the file name must be enclosed in parentheses.

Using the BY Phrase in FML Requests

Only one FOR phrase is permitted in a TABLE request. It substitutes in part for a BY phrase, which controls the sort sequence. However, the request can also include up to 32 BY phrases. In general, BY phrases specify the major (outer) sort fields in FML reports, and the FOR phrase specifies the minor (inner) sort field. Note that the BY ROWS OVER phrase is not supported in a request that uses the FOR phrase.
Combining BY and FOR Phrases in an FML Request

In this example, the report results for ACCOUNT (the inner sort field) are sorted by REGION (the outer sort field).

```
DEFINE FILE REGION
CUR_YR=E_ACTUAL;
LAST_YR=.831*CUR_YR;
REGION/A4=IF E_ACTUAL NE 0 OR E_BUDGET NE 0 THEN 'EAST' ELSE 'WEST';
END

TABLE FILE REGION
HEADING CENTER
"CURRENT ASSETS FOR REGION <REGION"
" "
SUM CUR_YR LAST_YR
BY REGION NOPRINT
FOR ACCOUNT
10$$ AS 'CASH' OVER
1100 AS 'ACCOUNTS RECEIVABLE' OVER
1200 AS 'INVENTORY' OVER
BAR OVER
RECAP CUR_ASSET/I5C = R1 + R2 + R3;
END
```

The output is shown as follows.

```
CURRENT ASSETS FOR REGION EAST

<table>
<thead>
<tr>
<th></th>
<th>CUR_YR</th>
<th>LAST_YR</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASH</td>
<td>9,511.00</td>
<td>7,903.64</td>
</tr>
<tr>
<td>ACCOUNTS RECEIVABLE</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>INVENTORY</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>CUR_ASSET</td>
<td>9,511</td>
<td>7,903</td>
</tr>
</tbody>
</table>
```
A sort field value can be used in a RECAP command to allow the model to take different actions within each major sort break. For instance, the following calculation computes a non-zero value only for the EAST region.

RECAP X=IF REGION EQ 'EAST' THEN .25*CASH ELSE 0;
AS 'AVAILABLE FOR DIVIDENDS'

For more information, see *Performing Inter-Row Calculations* on page 193.

### Supplying Data Directly in a Request

In certain cases, you may need to include additional constants (such as exchange rates or inflation rates) in your model. Not all data values for the model have to be retrieved from the data source. Using FML, you can supply data directly in the request.

**Syntax:**

How to Supply Data Directly in a Request

```FML
DATA value,[..., value],$ [AS 'text'] [LABEL label] OVER
```

where:

- **value**
  - Specifies the values that you are supplying. Values in a list must be separated by commas. The list must end with a comma and a dollar sign (,$).

- **AS 'text'**
  - Enables you to assign a title to the data row. Enclose the text in single quotation marks.
  - Without this entry, the row title is blank on the report.

- **label**
  - Assigns a name to the data row for use in RECAP calculations. The label can be up to 66 characters and cannot have blanks or special characters. Each explicit label you assign must be unique.
**Example:**  **Supplying Data Directly in a Request**

In this example, two values (.87 and 1.67) are provided for the exchange rates of euros and pounds, respectively.

```
DEFINE FILE LEDGER
EUROS/I5C=AMOUNT;
POUNDS/I5C=3.2*AMOUNT;
END

TABLE FILE LEDGER
SUM EUROS AS 'EUROPE,DIVISION'
POUNDS AS 'ENGLISH,DIVISION'
FOR ACCOUNT
1010 AS 'CASH--LOCAL CURRENCY' LABEL CASH          OVER
DATA .87, 1.67 ,$ AS 'EXCHANGE RATE' LABEL EXCH       OVER
RECAP US_DOLLARS/I5C = CASH * EXCH;
END
```

The values supplied are taken one column at a time for as many columns as the report originally specified.

The output is shown in the following image.

<table>
<thead>
<tr>
<th></th>
<th>EUROPE</th>
<th>ENGLISH</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASH--LOCAL CURRENCY</td>
<td>8,784</td>
<td>28,108</td>
</tr>
<tr>
<td>EXCHANGE RATE</td>
<td>.87</td>
<td>1.67</td>
</tr>
<tr>
<td>US_DOLLARS</td>
<td>7,642</td>
<td>46,940</td>
</tr>
</tbody>
</table>

**Performing Inter-Row Calculations**

The RECAP command enables you to perform calculations on data in the rows of the report to produce new rows. You must supply the name and format of the value that results from the calculation, and an expression that defines the calculation you wish to perform. Since RECAP calculations are performed among rows, each row in the calculation must be uniquely identified. FML supplies default row labels for this purpose (R1, R2, and so on). However, you may assign more meaningful labels. For more information, see *Referring to Rows in Calculations* on page 195.
Syntax: How to Define Inter-Row Calculations

RECAP calcname[/format]=expression; [AS 'text']

where:

RECAP

Is the required command name. It should begin on a line by itself.

calcname

Is the name you assign to the calculated value. The name can be up to 66 characters long, and must start with an alphabetic character. This name also serves as an explicit label. For more information, see Referring to Rows in Calculations on page 195.

format

Is the USAGE format of the calculated value. It cannot exceed the column width. The default is the format of the column in which the calculated value is displayed.

expression

Can be any calculation available with the DEFINE command (including IF ... THEN ... ELSE syntax, functions, excluding DECODE and EDIT, and fields in date format). The expression may extend to as many lines as it requires. A semicolon is required at the end of the expression. For more information, see Using Functions in RECAP Calculations on page 208 and the Using Functions manual.

The expression can include references to specific rows using the default FML positional labels (R1, R2, and so on), or it can refer to rows, columns, and cells using a variety of flexible notation techniques. Note that Rn references can only be used for rows previously evaluated within the model. For more information, see Referring to Rows in Calculations on page 195, Referring to Columns in Calculations on page 198, and Referring to Cells in Calculations on page 207.

AS 'text'

Changes the default title of the row. By default, the name of the RECAP value is displayed as the row title in output. The AS phrase replaces the default. Enclose the text in single quotation marks.
Reference: Usage Notes for RECAP

- RECAP expressions refer to other rows in the model by their labels (either explicit or default). Labels referred to in a RECAP expression must also be specified in the report request.

- The format specified for the RECAP result overrides the format of the column. In the following example,

```
RECAP TOTVAL/D6.2S=IF R1 GT R4 THEN R4 ELSE R1;
AS 'REDUCED VALUE'
```

TOTVAL/D6.2S displays the result as six positions with two decimal places (and displays blanks if the value was zero) in each column of the report, regardless of the format of the data in the column. This feature can be used to display percentages in a column of whole numbers.

- Subtotals are not supported in FML.

- In environments that support the RETYPE command, note that RETYPE does not recognize labels in FML with field format redefinition.

- Rn references (default positional row labels) can only be used for rows previously evaluated within the model.

### Referring to Rows in Calculations

FML assigns a default positional label to each TAG, DATA, RECAP, and PICKUP row. These positional labels are automatically prefixed with the letter R, so that the first such row in the model is R1, the second is R2, and so on. You can use these labels to refer to rows in RECAP expressions.

**Note:** Default labels are not assigned to rows that contain underlines, blank lines, or free text, since these row types need not be referenced in expressions.

When you refer to rows in a RECAP expression, you can:

- Use the positional row label assigned by FML.
- Create an explicit row label of your own.

**Note:** You should not create an explicit label with a name of the form Rn, as that type of name is used for default positional row labels assigned by FML and may cause problems with subsequent RECAPs.

- Mix positional and explicit row labels.
If you assign an explicit label, the positional label (R1, R2, and so on) is retained internally.

Note that an explicit label is not needed for a RECAP row, because the name of the calculated value on the left of the equal sign can be used as a label.

In addition to their role in RECAP calculations, you can use labels to format rows in an FML report. For more information, see Formatting an FML Report on page 234.

**Syntax:** How to Assign an Explicit Row Label

```
rowtype [AS 'text'] LABEL label [OVER]
```

where:

- **rowtype**
  - Can be a TAG, DATA, or PICKUP row.

- **AS 'text'**
  - Assigns a different name to the row for the report. Enclose the text in single quotation marks (').

- **label**
  - Assigns a label to a row for reference in a RECAP expression or a StyleSheet declaration. The label can be up to 66 characters and cannot have blanks or special characters. Each explicit label you assign must be unique.

**Note:** You should not create an explicit label with a name of the form Rn, as that type of name is used for default positional row labels assigned by FML and may cause problems with subsequent RECAPs.

Even if you assign an explicit label, the positional label (R1, R2, and so on) is retained internally.
**Example:** Referring to Default Row Labels in RECAP Expressions

In this example, FML assigns account 1010 the implicit label R1, account 1020, the implicit label R2, and account 1030, the implicit label R3. Since no label is assigned to a BAR row, the RECAP row is assigned the implicit label R4.

```plaintext
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 AS 'CASH ON HAND' OVER
1020 AS 'DEMAND DEPOSITS' OVER
1030 AS 'TIME DEPOSITS' OVER
BAR OVER
RECAP TOTCASH = R1 + R2 + R3; AS 'TOTAL CASH'
END
```

The output is shown as follows.

```
AMOUNT
------
CASH ON HAND      8,784
DEMAND DEPOSITS   4,494
TIME DEPOSITS     7,961
------
TOTAL CASH       21,239
```

**Example:** Referring to Explicit Row Labels in RECAP Expressions

The following request assigns the labels CA, AR, and INV to three tag rows, which are referenced in the RECAP expression.

```plaintext
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
10$ AS 'CASH' LABEL CA OVER
1100 AS 'ACCOUNTS RECEIVABLE' LABEL AR OVER
1200 AS 'INVENTORY' LABEL INV OVER
BAR OVER
RECAP CURASST/I5C = CA + AR + INV;
END
```

The output is shown as follows.

```
AMOUNT
------
CASH                21,239
ACCOUNTS RECEIVABLE 18,829
INVENTORY           27,307
------
CURASST             67,375
```

Note that the RECAP value could subsequently be referred to by the name CURASST, which functions as an explicit label.
Using Labels to Repeat Rows

In certain cases, you may wish to repeat an entire row later in your report. For example, the CASH account can appear in the Asset statement and Cash Flow statement of a financial analysis, as shown below.

```
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
"ASSETS"       OVER
10$$ AS 'CASH' LABEL TOTCASH OVER
.
.
"CASH FLOW"     OVER
RECAP SAMECASH/I5C = TOTCASH; AS 'CASH'
END
```

When you refer to the CASH row the second time, you can use a RECAP calculation (with a new name) and refer to the label, either explicitly (TOTCASH) or implicitly (R1), in the row where CASH was first used.

**Tip:** If you set the FORMULTIPLE parameter ON, you can repeat the row without giving it another name. For more information, see *Creating Rows From Multiple Records* on page 184.

Referring to Columns in Calculations

An FML report can refer to explicit columns, as well as explicit rows. You can refer to columns using:

- Column numbers.
- Contiguous column notation in RECAP expressions. For example, (2,5) represents columns 2 through 5.
- Column addressing.
- A factor to represent every other column, or every third column, and so on.
- Column notation to control the creation of column reference numbers.
- Column values.
Example: Applying Column Declarations in RECAP Expressions

The following request generates an FML matrix with four rows and three columns of data.

```
DEFINE FILE LEDGER
CUR_YR/I5C=AMOUNT;
LAST_YR/I5C=.87*CUR_YR - 142;
END

TABLE FILE LEDGER
  SUM CUR_YR AS 'CURRENT,YEAR'
  LAST_YR AS 'LAST,YEAR'
  COMPUTE CHANGE/I5C = CUR_YR - LAST_YR;
FOR ACCOUNT
  1010 AS 'CASH ON HAND' OVER
  1020 AS 'DEMAND DEPOSITS' OVER
  1030 AS 'TIME DEPOSITS' OVER
  BAR OVER
RECAP TOTCASH/I5C = R1 + R2 + R3; AS 'TOTAL CASH'
END
```

Both the columns of the report, as well as the cells of the matrix, can be referenced in another FML report.

The output is shown in the following image.

```
<table>
<thead>
<tr>
<th>CURRENT YEAR</th>
<th>LAST YEAR</th>
<th>CHANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASH ON HAND</td>
<td>8,784</td>
<td>7,216</td>
</tr>
<tr>
<td>DEMAND DEPOSITS</td>
<td>4,494</td>
<td>3,483</td>
</tr>
<tr>
<td>TIME DEPOSITS</td>
<td>7,961</td>
<td>6,499</td>
</tr>
<tr>
<td>TOTAL CASH</td>
<td>21,239</td>
<td>17,198</td>
</tr>
</tbody>
</table>
```

For example, you could use the value 6,499 in another FML report by referring to column 2, row 3. For more information, see Referring to Cells in Calculations on page 207.
Referring to Columns in Calculations

You can perform a calculation for one column or for a specific set of columns. To identify the columns, place the column number in parentheses after the label name.

Example: Referring to Column Numbers in a RECAP Expression

```plaintext
DEFINE FILE LEDGER
CUR_YR/I5C=AMOUNT;
LAST_YR/I5C=.87*CUR_YR - 142;
END

TABLE FILE LEDGER
SUM CUR_YR AS 'CURRENT,YEAR'
LAST_YR AS 'LAST,YEAR'
FOR ACCOUNT
1010 AS 'CASH ON HAND' OVER
1020 AS 'DEMAND DEPOSITS' OVER
1030 AS 'TIME DEPOSITS' OVER
BAR OVER
RECAP TOTCASH/I5C = R1 + R2 + R3; AS 'TOTAL CASH' OVER
" " OVER
RECAP GROCASH(2)/F5.2 = 100*TOTCASH(1)/TOTCASH(2) - 100;
AS 'CASH GROWTH(%)'
END
```

In the second RECAP expression, note that:

- TOTCASH(1) refers to total cash in column 1.
- TOTCASH(2) refers to total cash in column 2.
- The resulting calculation is displayed in column 2 of the row labeled CASH GROWTH(%).

The RECAP value is only calculated for the column specified.

The output is shown in the following image.

```
    CURRENT   LAST
      YEAR     YEAR
CASH ON HAND     8,784     7,216
DEMAND DEPOSITS  4,494     3,483
TIME DEPOSITS    7,961     6,499

TOTAL CASH       21,239   17,198
CASH GROWTH(%)   23.50
```

After data retrieval is completed, a single column is calculated all at once, and multiple columns one by one.
Referring to Contiguous Columns in Calculations

When a set of contiguous columns is needed within a RECAP, you can separate the first and last column numbers with commas. For example, DIFFERENCE (2,5) indicates that you want to compute the results for columns 2 through 5.

Example:  Recapping Over Contiguous Columns

In this example, the RECAP calculation for ATOT occurs only for columns 2 and 3, as specified in the request. No calculation is performed for column 1.

DEFINE FILE LEDGER
CUR_YR/I5C=AMOUNT;
LAST_YR/I5C=.87*CUR_YR - 142;
NEXT_YR/I5C=1.13*CUR_YR + 222;
END

TABLE FILE LEDGER
SUM NEXT_YR CUR_YR LAST_YR
FOR ACCOUNT
10$$ AS 'CASH' OVER
1100 AS 'ACCOUNTS RECEIVABLE' OVER
1200 AS 'INVENTORY' OVER
BAR OVER
RECAP ATOT(2,3)/I5C = R1 + R2 + R3;
AS 'ASSETS--ACTUAL'
END

The output is shown in the following image.

<table>
<thead>
<tr>
<th></th>
<th>NEXT_YR</th>
<th>CUR_YR</th>
<th>LAST_YR</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASH</td>
<td>25,992</td>
<td>21,239</td>
<td>17,198</td>
</tr>
<tr>
<td>ACCOUNTS RECEIVABLE</td>
<td>21,941</td>
<td>18,829</td>
<td>15,954</td>
</tr>
<tr>
<td>INVENTORY</td>
<td>31,522</td>
<td>27,307</td>
<td>23,329</td>
</tr>
<tr>
<td>ASSETS--ACTUAL</td>
<td></td>
<td>67,375</td>
<td>56,481</td>
</tr>
</tbody>
</table>
Referring to Columns in Calculations

When you need a calculation for every other or every third column instead of every column, you can supply a factor, or column address, to do this. Column addressing is useful when several data fields are displayed within each value of a column sort.

**Syntax:** How to Use Column Addressing in a RECAP Expression

The left-hand side of the expression has the form:

\[
\text{value}(s,e,i)[/format]=
\]

where:

- **value**
  - Is the name you assign to the result of the RECAP calculation.

- **s**
  - Is the starting column.

- **e**
  - Is the ending column (it may be * to denote all columns).

- **i**
  - Is the increment factor.

- **format**
  - Is the USAGE format of the calculated value. The default value is the format of the original column.

**Example:** Applying Column Addressing in a RECAP Expression

In the following statement, there are two columns for each month:

*SUM ACTUAL AND FORECAST ACROSS MONTH*

If you want to perform a calculation only for the ACTUAL data, control the placement of the results with a RECAP in the form:

\[
\text{RECAP } \text{calcname}(1,*,2)=\text{expression};
\]

The asterisk means to continue the RECAP for all odd-numbered columns (beginning in column 1, with an increment of 2, for all columns).
Referring to Relative Column Addresses in Calculations

A calculation can use a specific column as a base, and refer to all other columns by their displacement from that column. The column to the left of the base column has a displacement of -1 relative to the base column. The column to the right has a displacement of +1. For example,

\[ \text{COMP} = \text{FIX}(*) - \text{FIX}(*-1); \]

can refer to the change in fixed assets from one period to the next. The reference to COMP=FIX(*) is equivalent to COMP=FIX.

When referring to a prior column, the column must already have been retrieved, or its value is zero.

Applying Relative Column Addressing in a RECAP Expression

This example computes the change in cash (CHGCASH) for columns 1 and 2.

```sql
DEFINE FILE LEDGER
CUR_YR/I5C=AMOUNT;
LAST_YR/I5C=.87*CUR_YR - 142;
NEXT_YR/I5C=1.13*CUR_YR + 222;
END

TABLE FILE LEDGER
SUM NEXT_YR CUR_YR LAST_YR
FOR ACCOUNT
10$$ AS 'TOTAL CASH' LABEL TOTCASH OVER
" " OVER
RECAP CHGCASH(1,2)/I5SC = TOTCASH(*) - TOTCASH(*+1); AS 'CHANGE IN CASH'
END
```

The output is shown in the following image.

<table>
<thead>
<tr>
<th>TOTAL CASH</th>
<th>NEXT_YR</th>
<th>CUR_YR</th>
<th>LAST_YR</th>
</tr>
</thead>
<tbody>
<tr>
<td>25,992</td>
<td>17,198</td>
<td>21,239</td>
<td>17,198</td>
</tr>
</tbody>
</table>

| CHANGE IN CASH | 4,753 | 4,041 |

Controlling the Creation of Column Reference Numbers

Column notation assigns a sequential column number to each column in the internal matrix created for a report request. If you want to control the creation of column reference numbers for the columns that are used in your report, use the CNOTATION column notation command.
Because column numbers refer to columns in the internal matrix, they are assigned after retrieval and aggregation of data are completed. Columns created and displayed in a report are stored in the internal matrix, and columns that are not displayed in a report may also be generated and stored in the internal matrix. Columns stored in the internal matrix include calculated values, reformatted field values, BY fields, fields with the NOPRINT option, and certain RECAP calculations such as FORECAST and REGRESS. Every other column in the internal matrix is assigned a column number by default, which means you have to account for all internally generated columns, if you want to refer to the appropriate column value in your request.

You can change the default assignment of column reference numbers by using the SET CNOTATION=PRINTONLY command which assigns column numbers only to columns that display in the report output. You can use column notation in COMPUTE and RECAP commands to refer to these columns in your request.

**Syntax: How to Control the Creation of Column Reference Numbers**

```plaintext
SET CNOTATION={ALL | PRINTONLY | EXPLICIT}
```

where:

- **ALL**
  - Assigns column reference numbers to every column in the internal matrix. ALL is the default value.

- **PRINTONLY**
  - Assigns column reference numbers only to columns that display in the report output.

- **EXPLICIT**
  - Assigns column reference numbers to all fields referenced in the request, whether displayed or not.

**Note:** CNOTATION is not supported in an ON TABLE phrase.
Referring to Column Values in Calculations

When a report is sorted using the ACROSS phrase, all of the retrieved values are aligned under their appropriate columns. Each column has a title consisting of one value of the ACROSS field. The entire column of data can be addressed by this value in a RECAP calculation.

**Example:** Referring to a Column by Its Value in a RECAP Expression

The following request uses a factor that depends on the value of the ACROSS field (YEAR) to calculate the inventory cost for each year. It then calculates the profit by summing the assets and subtracting the inventory cost for each year.

```
TABLE FILE LEDGER
SUM AMOUNT ACROSS YEAR
FOR ACCOUNT
10$ AS 'CASH' LABEL CASH OVER
1100 AS 'ACCOUNTS RECEIVABLE' LABEL RECEIVE OVER
BAR OVER
1200 AS 'INVENTORY VALUE' LABEL INVENT OVER
RECAP INVENTORY_FACTOR/F5.2 = IF YEAR LT '1986'
    THEN 1.1 ELSE 1.25; AS 'INVENTORY COST FACTOR' OVER
RECAP INVENTORY_COST = INVENTORY_FACTOR * INVENT;
    AS 'INVENTORY COST' OVER
BAR OVER
RECAP PROFIT = CASH + RECEIVE - INVENTORY_COST;
END
```

The output is shown in the following image.

<table>
<thead>
<tr>
<th>YEAR</th>
<th>1985</th>
<th>1986</th>
<th>1987</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASH</td>
<td>5,663</td>
<td>7,001</td>
<td>8,575</td>
</tr>
<tr>
<td>ACCOUNTS RECEIVABLE</td>
<td>5,295</td>
<td>6,250</td>
<td>7,284</td>
</tr>
<tr>
<td>INVENTORY VALUE</td>
<td>7,754</td>
<td>9,076</td>
<td>10,477</td>
</tr>
<tr>
<td>INVENTORY COST FACTOR</td>
<td>1.10</td>
<td>1.25</td>
<td>1.25</td>
</tr>
<tr>
<td>INVENTORY COST</td>
<td>8,529</td>
<td>11,345</td>
<td>13,096</td>
</tr>
<tr>
<td>PROFIT</td>
<td>2,429</td>
<td>1,906</td>
<td>2,763</td>
</tr>
</tbody>
</table>
Referring to Rows and Columns in Calculations

You can style multiple RECAP commands in a matrix when the RECAP statements are placed after the last ACROSS value.

**Example:** Styling Multiple RECAP Statements in a Matrix

```plaintext
TABLE FILE GGSALES
SUM UNITS
BY PRODUCT
ACROSS REGION
RECAP
TTL1/I8 = C1+C2+C3+C4;
TTL2/D12.2 = TTL1*1.25; ON TABLE SET STYLE *
TYPE=DATA, COLUMN=TTL1 (*), COLOR=BLUE, BACKCOLOR=SILVER, STYLE=BOLD, $
TYPE=DATA, COLUMN=TTL2 (*), COLOR=RED, BACKCOLOR=AQUA, STYLE=BOLD, $
ENDSTYLE
END

The output is shown in the following image.
```

<table>
<thead>
<tr>
<th>Region</th>
<th>Midwest</th>
<th>Northeast</th>
<th>Southeast</th>
<th>West</th>
<th>TTL1</th>
<th>TTL2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Biscotti</td>
<td>86105</td>
<td>145242</td>
<td>119594</td>
<td>70436</td>
<td><strong>421377</strong></td>
<td><strong>526,721.25</strong></td>
</tr>
<tr>
<td>Capuccino</td>
<td>44785</td>
<td>73264</td>
<td>71168</td>
<td></td>
<td><strong>189217</strong></td>
<td><strong>236,521.25</strong></td>
</tr>
<tr>
<td>Coffee Grinder</td>
<td>50393</td>
<td>40977</td>
<td>47083</td>
<td>48081</td>
<td><strong>186534</strong></td>
<td><strong>233,157.50</strong></td>
</tr>
<tr>
<td>Coffee Pot</td>
<td>47156</td>
<td>46185</td>
<td>49922</td>
<td>47422</td>
<td><strong>190695</strong></td>
<td><strong>239,368.75</strong></td>
</tr>
<tr>
<td>Croissant</td>
<td>139182</td>
<td>137394</td>
<td>156456</td>
<td>197022</td>
<td><strong>630054</strong></td>
<td><strong>787,567.50</strong></td>
</tr>
<tr>
<td>Espresso</td>
<td>101154</td>
<td>68127</td>
<td>68030</td>
<td>71675</td>
<td><strong>308986</strong></td>
<td><strong>386,232.50</strong></td>
</tr>
<tr>
<td>Latte</td>
<td>231623</td>
<td>222866</td>
<td>209654</td>
<td>213920</td>
<td><strong>878063</strong></td>
<td><strong>1,097,578.75</strong></td>
</tr>
<tr>
<td>Mug</td>
<td>86718</td>
<td>91497</td>
<td>88474</td>
<td>93881</td>
<td><strong>360570</strong></td>
<td><strong>450,712.50</strong></td>
</tr>
<tr>
<td>Scone</td>
<td>116127</td>
<td>70732</td>
<td>73779</td>
<td>72776</td>
<td><strong>333414</strong></td>
<td><strong>416,767.50</strong></td>
</tr>
<tr>
<td>Thermos</td>
<td>46587</td>
<td>48870</td>
<td>48976</td>
<td>45648</td>
<td><strong>190081</strong></td>
<td><strong>237,601.25</strong></td>
</tr>
</tbody>
</table>

206 WebFOCUS
Referring to Cells in Calculations

You can refer to columns and rows using a form of cell notation that identifies the intersection of a row and a column as \((r, c)\).

**Syntax:** How to Use Cell Notation for Rows and Columns in a RECAP Expression

A row and column can be addressed in an expression by the notation:

\[ E(r, c) \]

where:

- \(E\) is a required constant.
- \(r\) is the row number.
- \(c\) is the column number. Use an asterisk (*) to indicate the current column.

**Example:** Referring to Columns Using Cell Notation in a RECAP Expression

In this request, two RECAP expressions derive VARIANCEs (EVAR and WVAR) by subtracting values in four columns \((1, 2, 3, 4)\) in row three (PROFIT). These values are identified using cell notation \((r,c)\).

```
TABLE FILE REGION
SUM E_ACTUAL E_BUDGET W_ACTUAL W_BUDGET
FOR ACCOUNT
3000 AS 'SALES' OVER
3100 AS 'COST' OVER
BAR OVER
RECAP PROFIT/I5C = R1 - R2;
" " OVER
RECAP EVAR(1)/I5C = E(3,1) - E(3,2);
AS 'EAST--VARIANCE' OVER
RECAP WVAR(3)/I5C = E(3,3) - E(3,4);
AS 'WEST--VARIANCE'
END
```
The output is shown as follows.

<table>
<thead>
<tr>
<th></th>
<th>E_ACTUAL</th>
<th>E_BUDGET</th>
<th>W_ACTUAL</th>
<th>W_BUDGET</th>
</tr>
</thead>
<tbody>
<tr>
<td>SALES</td>
<td>6,000</td>
<td>4,934</td>
<td>7,222</td>
<td>7,056</td>
</tr>
<tr>
<td>COST</td>
<td>4,650</td>
<td>3,760</td>
<td>5,697</td>
<td>5,410</td>
</tr>
<tr>
<td></td>
<td>1,350</td>
<td>1,174</td>
<td>1,525</td>
<td>1,646</td>
</tr>
<tr>
<td>EAST--VARIANCE</td>
<td>176</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WEST--VARIANCE</td>
<td></td>
<td></td>
<td></td>
<td>-121</td>
</tr>
</tbody>
</table>

**Note:** In addition to illustrating cell notation, this example demonstrates the use of column numbering. Notice that the display of the EAST and WEST VARIANCEs in columns 1 and 3, respectively, are controlled by the numbers in parentheses in the request: EVAR (1) and WVAR (3).

### Using Functions in RECAP Calculations

You may provide your own calculation routines in RECAP rows to perform special-purpose calculations, a useful feature when these calculations are mathematically complex or require extensive look-up tables.

User-written functions are coded as subroutines in any language that supports a call process, such as FORTRAN, COBOL, PL/1, and BAL. For information about creating your own functions, see the *Using Functions* manual.

**Syntax:** How to Call a Function in a RECAP Command

```
RECAP calcname[(s,e,i)][/format]=function(input1,...,inputn,'format2');
```

where:

- **calcname**
  - Is the name you assign to the calculated value.

- **(s,e,i)**
  - Specify a start (s), end (e), and increment (i) value for the column where you want the value displayed. If omitted, the value appears in all columns.

- **format**
  - The format for the calculation is optional. The default is the format of the column. If the calculation consists of only the subroutine, make sure that the format of the subroutine output value (format2) agrees with the calculation format. If the calculation format is larger than the column width, the value displays in that column as asterisks (*).
**Example: Calling a Function in a RECAP Command**

Suppose you have a function named INVEST in your private collection of functions (INVEST is not available in the supplied library), and it calculates an amount on the basis of cash on hand, total assets, and the current date. In order to create a report that prints an account of company assets and calculates how much money the company has available to invest, you must create a report request that invokes the INVEST function.

The current date is obtained from the &YMD system variable. The NOPRINT option beside it prevents the date from appearing in the report. The date is solely used as input for the next RECAP statement.
The request is:

```sql
TABLE FILE LEDGER
HEADING CENTER
"ASSETS AND MONEY AVAILABLE FOR INVESTMENT </2"
SUM AMOUNT ACROSS HIGHEST YEAR
IF YEAR EQ 1985 OR 1986
FOR ACCOUNT
  1010 AS 'CASH'                  LABEL CASH      OVER
  1020 AS 'ACCOUNTS RECEIVABLE'   LABEL ACR       OVER
  1030 AS 'INTEREST RECEIVABLE'   LABEL ACI       OVER
  1100 AS 'FUEL INVENTORY'        LABEL FUEL      OVER
  1200 AS 'MATERIALS AND SUPPLIES' LABEL MAT       OVER
BAR
RECAP TOTCAS = CASH+ACR+ACI+FUEL+MAT; AS 'TOTAL ASSETS' OVER
BAR
RECAP THISDATE/A8 = &YMD; NOPRINT
RECAP INVAIL = INVEST(CASH,TOTCAS,THISDATE,'D12.2'); AS
               'AVAIL. FOR INVESTMENT'
BAR AS '='
END
```

The output is shown in the following image.

<table>
<thead>
<tr>
<th>ASSETS AND MONEY AVAILABLE FOR INVESTMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>YEAR</td>
</tr>
<tr>
<td>1986</td>
</tr>
<tr>
<td>CASH</td>
</tr>
<tr>
<td>ACCOUNTS RECEIVABLE</td>
</tr>
<tr>
<td>INTEREST RECEIVABLE</td>
</tr>
<tr>
<td>FUEL INVENTORY</td>
</tr>
<tr>
<td>MATERIALS AND SUPPLIES</td>
</tr>
<tr>
<td>TOTAL ASSETS</td>
</tr>
<tr>
<td>AVAIL FOR INVESTMENT</td>
</tr>
</tbody>
</table>

**Inserting Rows of Free Text**

Insert text anywhere in your FML report by typing it on a line by itself and enclosing it within double quotation marks. You can also add blank lines, designated as text, to improve the appearance of the report.

In addition, you can include data developed in your FML report in a row of free text by including the label for the data variable in the text row.
**Example: Inserting Free Text**

In this example, three rows of free text are inserted, one blank and two text rows.

TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
" --- CASH ACCOUNTS ---" OVER
1010 AS 'CASH ON HAND' OVER
1020 AS 'DEMAND DEPOSITS' OVER
1030 AS 'TIME DEPOSITS' OVER
" " OVER
" --- OTHER CURRENT ASSETS ---" OVER
1100 AS 'ACCOUNTS RECEIVABLE' OVER
1200 AS 'INVENTORY'
END

The output is shown as follows.

<table>
<thead>
<tr>
<th>AMOUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>------</td>
</tr>
<tr>
<td>--- CASH ACCOUNTS ---</td>
</tr>
<tr>
<td>CASH ON HAND</td>
</tr>
<tr>
<td>DEMAND DEPOSITS</td>
</tr>
<tr>
<td>TIME DEPOSITS</td>
</tr>
<tr>
<td>--- OTHER CURRENT ASSETS ---</td>
</tr>
<tr>
<td>ACCOUNTS RECEIVABLE</td>
</tr>
<tr>
<td>INVENTORY</td>
</tr>
</tbody>
</table>

Notice that the blank row was created by enclosing a blank within double quotation marks on a separate line of the report request.

**Syntax: How to Insert Data Variables in Text Rows**

"text <label{(c)}[]>"

where:

<

Is a required left caret to bracket the label.

label

Is the explicit or implicit row label. (In a RECAP, the calculated value functions as the label.)

c

Is an optional cell identifier that indicates the column number of the cell. However, this identifier is required whenever there is more than one column in the report. If you use it, enclose it in parentheses.
Adding a Column to an FML Report

The request controls the number of columns in any report. For instance, if a request contains the display command SUM AMOUNT AND FORECAST, the report contains two columns: AMOUNT and FORECAST.

Add columns in an FML request, just as in a TABLE request, using the COMPUTE command to calculate a value or simply to allocate the space, column title, and format for a column.
**Example: Adding a Column to an FML Report**

This example uses a COMPUTE command to generate the calculated value CHANGE and display it as a new column in the FML report. The following request generates an FML matrix with four rows and three columns of data.

```
DEFINE FILE LEDGER
CUR_YR/I5C=AMOUNT;
LAST_YR/I5C=.87*CUR_YR - 142;
END

TABLE FILE LEDGER
SUM CUR_YR AS 'CURRENT,YEAR'
   LAST_YR AS 'LAST,YEAR'
COMPUTE CHANGE/I5C = CUR_YR - LAST_YR;
FOR ACCOUNT
  1010 AS 'CASH ON HAND'               OVER
  1020 AS 'DEMAND DEPOSITS'            OVER
  1030 AS 'TIME DEPOSITS'              OVER
  BAR                                 OVER
RECAP TOTCASH/I5C = R1 + R2 + R3; AS 'TOTAL CASH'
END
```

The output is shown in the following image.

![Output Image](image.png)

**Note:** The designated calculation is performed on each tag or RECAP row of the report. The RECAP rows, however, may change the calculation.
Adding a New Time Period as a Column

The following request adds a future time period to a report.

DEFINE FILE LEDGER
CUR_YR/P5C=AMOUNT;
LAST_YR/P5C=.87*AMOUNT - 142;
END

TABLE FILE LEDGER
SUM AMOUNT
ACROSS YEAR AND COMPUTE 1999/P5C = 2.5*AMOUNT;
FOR ACCOUNT
1010 AS 'CASH ON HAND' OVER
1020 AS 'DEMAND DEPOSITS' OVER
1030 AS 'TIME DEPOSITS' OVER
BAR OVER
RECAP TOTCASH/P5C = R1 + R2 + R3; AS 'TOTAL CASH' OVER
RECAP CHANGE(2,* ) = TOTCASH(*) - TOTCASH(*-1);
END

The output is shown as follows.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CASH ON HAND</td>
<td>1,684</td>
<td>2,100</td>
<td>5,000</td>
<td>4,210</td>
</tr>
<tr>
<td>DEMAND DEPOSITS</td>
<td>619</td>
<td>875</td>
<td>3,000</td>
<td>1,548</td>
</tr>
<tr>
<td>TIME DEPOSITS</td>
<td>3,360</td>
<td>4,026</td>
<td>575</td>
<td>8,400</td>
</tr>
<tr>
<td>TOTAL CASH</td>
<td>5,663</td>
<td>7,001</td>
<td>8,575</td>
<td>14,158</td>
</tr>
<tr>
<td>CHANGE</td>
<td>1,332</td>
<td>1,574</td>
<td>5,383</td>
<td></td>
</tr>
</tbody>
</table>
Creating a Recursive Model

Models involving different time periods often require using the ending value of one time period as the starting value for the next. The calculations describing these situations have two characteristics:

- The labels on one or more RECAP rows are duplicates of other rows. They are used repeatedly to recompute certain values.

- A calculation may refer to a label not yet described, but provided later in the model. If, at the end of the model, a label that is needed is missing, a message is displayed.

Recursive models require that the columns are produced in sequential order, one by one. In nonrecursive models, all of the columns can be produced simultaneously. Schematically, these patterns are shown as follows.

FML automatically switches to sequential order as soon as either of the two modeling conditions requiring the switch is recognized (either reuse of labels by different rows, or forward reference to a label in a calculation).
Example: Creating a Recursive Model

The following example illustrates recursive models. Note that one year of ENDCASH becomes the next year of STARTING CASH.

```
DEFINE FILE REGION
CUR_YR=E_ACTUAL;
LAST_YR=.831*CUR_YR;
NEXT_YR=1.2297*CUR_YR;
END

TABLE FILE REGION
SUM LAST_YR CUR_YR NEXT_YR
FOR ACCOUNT
10$$ AS 'STARTING CASH' LABEL STCASH OVER
RECAP STCASH(2,*') = ENDCASH(*-1); OVER
" " OVER
3000 AS 'SALES' LABEL SLS OVER
3100 AS 'COST' LABEL COST OVER
BAR OVER
RECAP PROFIT/I5C = SLS - COST; OVER
" " OVER
RECAP ENDCASH/I5C = STCASH + PROFIT;
END
```

The output is shown as follows.

<table>
<thead>
<tr>
<th></th>
<th>LAST_YR</th>
<th>CUR_YR</th>
<th>NEXT_YR</th>
</tr>
</thead>
<tbody>
<tr>
<td>STARTING CASH</td>
<td>7,903.64</td>
<td>9,024.00</td>
<td>10,374.00</td>
</tr>
<tr>
<td>SALES</td>
<td>4,986.00</td>
<td>6,000.00</td>
<td>7,378.20</td>
</tr>
<tr>
<td>COST</td>
<td>3,864.15</td>
<td>4,650.00</td>
<td>5,718.11</td>
</tr>
<tr>
<td>PROFIT</td>
<td>1,121</td>
<td>1,350</td>
<td>1,660</td>
</tr>
<tr>
<td>ENDCASH</td>
<td>9,024</td>
<td>10,374</td>
<td>12,034</td>
</tr>
</tbody>
</table>

Reporting Dynamically From a Hierarchy

Hierarchical relationships between fields can be defined in a Master File, and automatically displayed using the Financial Modeling Language (FML). The parent and child fields must share data values, and their relationship should be hierarchical. The formats of the parent and child fields must both be either numeric or alphanumeric.
For example, suppose that:

- Employee and manager IDs are contained within an employee data source.
- A general ledger data source contains both an account number field and an account parent field.

By examining these fields, it is possible to construct the entire organization chart or chart of accounts structure. However, to print the chart in a traditional FML report, you need to list the employee IDs or account numbers in the request syntax in the order in which they should appear on the report. If an employee or account is added, removed, or transferred, you have to change the report request to reflect this change in organizational structure. For example:

```
TABLE FILE EMPLOYEE
PRINT DEPARTMENT CURR_JOBCODE
FOR EMP_ID
999999999   OVER
222222222   OVER
...
```

In contrast, with FML hierarchies you can define the hierarchical relationship between two fields in the Master File and load this information into memory. The FML request can then dynamically construct the rows that represent this relationship and display them in the report, starting at any point in the hierarchy. In the example shown, EMP_ID is called the hierarchy field.

**Requirements for FML Hierarchies**

1. In the Master File, use the PROPERTY=PARENT_OF and REFERENCE=\*hierarchyfld* attributes to define the hierarchical relationship between two fields. For more information, see the "Describing Data With WebFOCUS Language" manual.

   The hierarchy must be loaded into memory. This loaded hierarchy is called a chart. If the hierarchy is defined in the Master File and referenced by the FML request, it is loaded automatically. If you want to use a hierarchy defined in a Master File that is not either referenced in the FML request or joined to the Master File referenced in the FML request, issue the LOAD CHART command before issuing the FML request.

   The number of charts that can be loaded is 16. Charts are automatically unloaded when the session ends.
2. In the FOR phrase of the FML request. Use the GET/WITH CHILDREN or ADD phrase to retrieve the hierarchical data starting at a specific point in the hierarchy.

To use FML hierarchies, the FOR field must either be:

- The hierarchy field.

  or

- Used as the join field to a unique segment that has the hierarchy field. In this case, the hierarchy field must be the join field. Note that the condition that the join be unique only applies if the hierarchy is defined in the cross-referenced segment.

In other words, the FOR field must be in a parent-child hierarchy, or linked to one. The latter case allows transaction data that contains the hierarchy field to be joined to a separate data source that contains the hierarchy definition.

As with any FML request, a tagged row is displayed even if no data is found in the file for the tag values, with a period (.) representing the missing data. You can override this convention by adding the phrase WHEN EXISTS to the definition of a tagged row. This makes displaying a row dependent upon the existence of data for the tag.

**Example: Defining a Hierarchy in a Master File**

The CENTGL Master File contains a charts of accounts hierarchy. The field GL_ACCOUNT_PARENT is the parent field in the hierarchy. The field GL_ACCOUNT is the hierarchy field. The field GL_ACCOUNT_CAPTION can be used as the descriptive caption for the hierarchy field.

```plaintext
FILE=CENTGL  , SUFFIX=FOC
SEGNAME=ACCOUNTS, SEGTYPE=S01
FIELDNAME=GL_ACCOUNT, ALIAS=GLACCT, FORMAT=A7,
          TITLE='Ledger, Account', FIELDTYPE=I, $
FIELDNAME=GL_ACCOUNT_PARENT, ALIAS=GLPAR, FORMAT=A7,
          TITLE=Parent,
          PROPERTY=PARENT_OF, REFERENCE=GL_ACCOUNT, $
FIELDNAME=GL_ACCOUNT_TYPE, ALIAS=GLTYPE, FORMAT=A1,
          TITLE=Type, $
FIELDNAME=GL_ROLLUP_OP, ALIAS=GLROLL, FORMAT=A1,
          TITLE=Op, $
FIELDNAME=GL_ACCOUNT_LEVEL, ALIAS=GLLEVEL, FORMAT=I3,
          TITLE=Lev, $
FIELDNAME=GL_ACCOUNT_CAPTION, ALIAS=GLCAP, FORMAT=A30,
          TITLE=Caption,
          PROPERTY=CAPTION, REFERENCE=GL_ACCOUNT, $
FIELDNAME=SYS_ACCOUNT, ALIAS=ALINE, FORMAT=A6,
          TITLE='System, Account, Line', MISSING=ON, $
```
The CENTSYSF data source contains detail-level financial data. This is unconsolidated financial data for a fictional corporation, CenturyCorp. It is designed to be separate from the CENTGL database as if it came from an external accounting system. It uses a different account line system (SYS_ACCOUNT) which can be joined to the SYS_ACCOUNT field in CENTGL. Data uses \textit{natural} signs (expenses are positive, revenue negative).

\begin{verbatim}
FILE=CENTSYSF     ,SUFFIX=FOC
SEGNAME=RAWDATA   ,SEGTYPE=S2
FIELDNAME=SYS_ACCOUNT   ,  ,A6       , FIELDTYPE=I, 
               TITLE='System,Account,Line', $
FIELDNAME=PERIOD        ,  ,YYM      , FIELDTYPE=I, $
FIELDNAME=NAT_AMOUNT    ,  ,D10.0    , TITLE='Month,Actual', $
FIELDNAME=NAT_BUDGET    ,  ,D10.0    , TITLE='Month,Budget', $
FIELDNAME=NAT_YTDAMT    ,  ,D12.0    , TITLE='YTD,Actual', $
\end{verbatim}

**Displaying an FML Hierarchy**

The GET CHILDREN and WITH CHILDREN commands dynamically retrieve and display hierarchical data on the FML report. GET CHILDREN displays only the children, not the parent value referenced in the command. WITH CHILDREN displays the parent and then the children.

**Syntax:** \textbf{How to Display an FML Hierarchy}

\begin{verbatim}
TABLE FILE filename{PRINT|SUM} ...
FOR hierarchyfld
parentvalue {GET|WITH} CHILD[REN] [n|ALL]
   [AS CAPTION|'text'] [LABEL label]
...
END
\end{verbatim}

where:

\textit{filename}

Is the name of the file to be used in the FML request. If the hierarchy for this request cannot be loaded automatically, it must have been loaded previously by issuing the LOAD CHART command.

\textit{hierarchyfld}

Is the hierarchy field name. If the request references a joined structure, the name must be the field name from the host file. The alias name is not supported.

\textit{parentvalue}

Is the parent value for which the children are to be retrieved.
**GET CHILDREN**

Displays the hierarchy starting from the first child of the specified `parentvalue`. It does not include the parent in the display. (This corresponds to the FML syntax `CHILD1 OVER CHILD2 OVER ...`.)

**WITH CHILDREN**

Displays the hierarchy starting from the specified `parentvalue`. It includes the parent in the display. (This corresponds to the FML syntax `parentvalue OVER CHILD1 OVER CHILD2 OVER ...`).

**n|ALL**

Is a positive integer from 1 to 99, specifying the number of levels of the hierarchy to display. If a number greater than 99 is specified, a warning message is displayed and `n` is set to 99. The default value is 1. Therefore, if `n` is omitted, only direct children are displayed. GET or WITH CHILDREN 2 displays direct children and grandchildren. GET or WITH CHILDREN 99 displays children to 99 levels. ALL is a synonym for 99. Each child instance is printed over the one that follows. Successive levels of the hierarchy field are indented two spaces from the previous level.

**CAPTION**

Indicates that the caption values to display should be taken from the field defined as the CAPTION in the Master File.

Note that the AS CAPTION phrase is supported for tagged rows, including those that do not use the GET/WITH CHILDREN or ADD syntax. However, the hierarchy must be defined (by specifying the PARENT_OF attribute) in order to load and display the caption values. If the hierarchy is not defined, the AS CAPTION phrase is ignored.

`'text'`

Is a text string to use as the row title for the hierarchy field values. The CAPTION field defined in the Master File is not used as the caption on the report output.

**label**

Is an explicit row label. Each generated row is labeled with the specified label text.

**Note:** The hierarchy is displayed sorted by the parent field and, within parent, sorted by the hierarchy field.

For information about the FMLFOR, FMLLIST, FMLCAP, and FMLINFO functions that return the tag values and captions used in an FML request, see the *Using Functions* manual.
Example: Displaying an FML Hierarchy

The following request displays two levels of account numbers, starting from account 3000:

```
SET BLANKINDENT=ON
TABLE FILE CENTGL
PRINT GL_ACCOUNT_PARENT
FOR GL_ACCOUNT
  3000 WITH CHILDREN 2
END
```

The output is shown as follows.

```
Parent
-------
3000   1000
3100   3000
  3110  3100
  3120  3100
  3130  3100
  3140  3100
  3200  3000
  3300  3200
  3400  3200
  3500  3200
  3600  3200
  3700  3200
  3800  3200
  3900  3200
```

Note: If the request specifies GET CHILDREN instead of WITH CHILDREN, the line for the parent value (3000) does not display on the report output.
Example: Displaying an FML Hierarchy With Captions

The following request displays two levels of a chart of accounts hierarchy, starting with account 1000 (the top of the hierarchy), and displays the caption field values instead of the account numbers.

```sql
SET BLANKINDENT=ON
TABLE FILE CENTGL
PRINT GL_ACCOUNT_PARENT
FOR GL_ACCOUNT
1000 WITH CHILDREN 2 AS CAPTION
END
```

The output is shown as follows.

<table>
<thead>
<tr>
<th>Parent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profit Before Tax</td>
</tr>
<tr>
<td>Gross Margin</td>
</tr>
<tr>
<td>Sales Revenue</td>
</tr>
<tr>
<td>Cost Of Goods Sold</td>
</tr>
<tr>
<td>Total Operating Expenses</td>
</tr>
<tr>
<td>Selling Expenses</td>
</tr>
<tr>
<td>General + Admin Expenses</td>
</tr>
<tr>
<td>Total R+D Costs</td>
</tr>
<tr>
<td>Salaries</td>
</tr>
<tr>
<td>Misc. Equipment</td>
</tr>
</tbody>
</table>

1000  2000  2000  1000  3000  3000  1000  5000  5000

Note: If the request specifies GET CHILDREN instead of WITH CHILDREN, the line for the parent value (1000, Profit Before Tax) does not display on the report output.

Consolidating an FML Hierarchy

The ADD command consolidates multiple levels of the hierarchy on one line of the FML report output. You can use ADD alone or in conjunction with GET CHILDREN or WITH CHILDREN. Note that ADD is designed to work with requests that use the SUM command. It is also designed to be used with detail-level data, not data that is consolidated.

When used alone, ADD aggregates the parent and children on one line of the report output, summing the numeric data values included on the line. This corresponds to the FML syntax `parentvalue OR CHILD1 OR CHILD2 OR ...`

When used in conjunction with GET CHILDREN, ADD displays one line for each child of the specified parent value. Each line is a summation of that child and all of its children. You can specify the number of levels of children to display (which determines the number of lines generated on the report output) and the depth of summation under each child. By default, only direct children have a line in the report output, and the summary for each child includes all of its children.
When used in conjunction with WITH CHILDREN, ADD first displays a line in the report output that consists of the summation of the parent value and all of its children. Then it displays additional lines identical to those displayed by GET CHILDREN ADD.

In order to use a data record in more than one line of an FML report (for example, to display both detail and summary lines or to consolidate detail data at multiple levels), the following setting is required:

```
SET FORMULTIPLE=ON
```

**Syntax:** How to Create One Summary Row for an FML Hierarchy

```
TABLE FILE filenameSUM ... 
FOR hierarchyfld
parentvalue ADD [n|ALL]
  [AS CAPTION|'text'] [LABEL label]
  ...
END
```

where:

- **filename**
  - Is the name of the file to be used in the FML request. If the hierarchy for this request cannot be loaded automatically, it must have been loaded previously by issuing the LOAD CHART command.

- **hierarchyfld**
  - Is the hierarchy field name. If the request references a joined structure, the name must be the field name from the host file. The alias name is not supported.

- **parentvalue**
  - Is the parent value that determines the starting point in the hierarchy for the aggregation.

- **ADD**
  - Displays the parent and \( n \) levels of its children on one row, summing the numeric data values displayed on the row. This corresponds to the FML syntax `parentvalue` or `CHILD1 OR CHILD2 OR CHILD3 and more, if applicable.`
To display the sum of just the children, you must display the parent row, display the summary row, and use a RECAP to subtract the parent row from the sum. For example:

```
FOR ...
  parentvalue  OVER
  parentvalue ADD 1  OVER
RECAP CHILDSUM = R2-R1;
```

\( n \mid ALL \)

Is a positive integer from 1 to 99, specifying the number of levels of the hierarchy to aggregate. ALL is the default value. Therefore, if \( n \) is omitted, all children are included in the sum. If \( n \) is 1, only direct children are included. If \( n \) is 2, direct children and grandchildren are included. ADD 99 includes up to 99 levels of children. ALL is a synonym for 99.

**CAPTION**

Indicates that the caption of the parent value displays for the total row.

Note that the AS CAPTION phrase is supported for tagged rows, including those that do not use the GET CHILDREN or ADD syntax. However, the hierarchy must be defined (by specifying the PARENT_OF attribute) in order to load and display the caption values. If the hierarchy is not defined, the AS CAPTION phrase is ignored.

\'(text)\'

Is a text string to use as the row title for the aggregate row. The CAPTION field defined in the Master File is not used as the caption on the report output.

**label**

Is an explicit row label. Each generated row is labeled with the specified label text.
**Example:** Displaying One Summary Line for an FML Hierarchy

The CENTSYSF data source contains detail-level financial data. To use the account hierarchy in the CENTGL data source with this financial data, the two data sources are joined. The data in CENTSYSF is stored with natural signs, which means, in financial terms, that revenues and liabilities are stored as negative numbers. The portion of the hierarchy used in this request contains only positive data.

Note that the join is not required to be unique, because the hierarchy is defined in the host segment.

First the WITH CHILDREN command displays the lines of the hierarchy starting with account Selling Expenses (3100). Note that only accounts with no children are populated in this detail-level data source. The ADD command then creates one line that is the sum of account 3100 and all of its children.

```
SET BLANKINDENT=ON
SET FORMULTIPLE=ON
JOIN SYS_ACCOUNT IN CENTGL TO ALL SYS_ACCOUNT IN CENTSYSF
TABLE FILE CENTGL
SUM NAT_AMOUNT/D10.0 NAT_YTDAMT/D10.0
FOR GL_ACCOUNT
3100 WITH CHILDREN ALL AS CAPTION OVER
BAR
3100 ADD AS CAPTION
IF PERIOD EQ '2002/03'
END
```

The output is shown as follows.

<table>
<thead>
<tr>
<th>Selling Expenses</th>
<th>Month</th>
<th>YTD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Actual</td>
<td>Actual</td>
</tr>
<tr>
<td>Advertising</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TV/Radio</td>
<td>1,049,146</td>
<td>2,954,342</td>
</tr>
<tr>
<td>Print Media</td>
<td>244,589</td>
<td>721,448</td>
</tr>
<tr>
<td>Internet Advertising</td>
<td>9,542</td>
<td>29,578</td>
</tr>
<tr>
<td>Promotional Expenses</td>
<td>53,719</td>
<td>151,732</td>
</tr>
<tr>
<td>Joint Marketing</td>
<td>97,135</td>
<td>289,799</td>
</tr>
<tr>
<td>Bonuses/Commisions</td>
<td>100,188</td>
<td>304,199</td>
</tr>
<tr>
<td>Selling Expenses</td>
<td>1,554,319</td>
<td>4,451,098</td>
</tr>
</tbody>
</table>
**Syntax:**  How to Consolidate FML Hierarchy Data to Any Level and Depth

```
TABLE FILE filename
SUM ...
FOR hierarchyfld
parentvalue {GET|WITH} CHILDREN [n|ALL] ADD [m|ALL]
[AS CAPTION|'text'] [LABEL label]
  .
  .
END
```

where:

- **filename**
  Is the name of the file used in the FML request. If the hierarchy for this request cannot load automatically, it previously loaded by issuing the LOAD CHART command.

- **hierarchyfld**
  Is the hierarchy field name. If the request references a joined structure, the name must be the field name from the host file. The alias name is not supported.

- **parentvalue**
  Is the parent value that determines the starting point in the hierarchy for the aggregation.

- **GET|WITH**
  GET specifies that the first line generated on the report is the consolidated line for the first child of the parent value. WITH specifies that the first line generated on the report is the consolidated line for the parent value, followed by the consolidated lines for each of its children, to the level specified by \( n \).

- **n|ALL**
  Is a positive integer from 1 to 99, specifying the number of levels of children to display. The line of output for each child has the sum of that child and its children to the depth specified for the ADD option. The default value is 1. Therefore, if \( n \) is omitted, each direct child has a line on the report. If \( n \) is 2, direct children and grandchildren each have a line on the report output. ALL is a synonym for 99.

- **ADD**
  Sums the hierarchy to the depth specified by \( m \) for each line generated by the GET or WITH CHILDREN command.
\( m | \text{ALL} \)

Is a positive integer from 1 to 99, specifying the number of levels of children to consolidate on each line of the report output. If a number greater than 99 is specified, a warning message is displayed and \( m \) is set to 99. The default value is \( \text{ALL} \). Therefore, if \( m \) is omitted, the consolidated line sums all children. If \( m \) is 2, only direct children and grandchildren are consolidated for each line on the report output. \( \text{ADD} \ 99 \) aggregates children to 99 levels. \( \text{ALL} \) is a synonym for 99.

**CAPTION**

Indicates that the caption of the parent value displays for the total row.

Note that the \text{AS CAPTION} phrase is supported for tagged rows, including those that do not use the \text{GET CHILDREN} or \text{ADD} syntax. However, the hierarchy must be defined (by specifying the \text{PARENT_OF} attribute) in order to load and display the caption values. If the hierarchy is not defined, the \text{AS CAPTION} phrase is ignored.

'\text{text}'

Is a text string to use as the row title for the aggregate row. The CAPTION field defined in the Master File is not used as the caption on the report output.

\text{label}

Is an explicit row label. Each generated row is labeled with the specified label text.

---

**Example:**  **Consolidating FML Hierarchy Data**

In the following request, the first WITH CHILD command displays the detail data for the hierarchy starting with account 3100. The next WITH CHILD command creates a consolidated line for the parent account (3100) and each direct child.

```
SET BLANKINDENT=ON
SET FORMULTIPLE=ON
JOIN SYS_ACCOUNT IN CENTGL TO ALL SYS_ACCOUNT IN CENTSYSF
TABLE FILE CENTGL
SUM NAT_AMOUNT/D10.0 NAT_YTDAMT/D10.0
FOR GL_ACCOUNT
3100 WITH CHILDREN ALL AS CAPTION      OVER
" "                                    OVER
BAR AS =                               OVER
" "                                    OVER
3100 WITH CHILDREN ADD AS CAPTION
IF PERIOD EQ '2002/03'
END
```

Note that the join is not required to be unique, because the hierarchy is defined in the host segment.
In the following output, the top portion shows the detail-level data. The bottom portion shows the consolidated data. In the consolidated portion of the report:

- There is one line for the parent that is the sum of itself plus all of its children to all levels.
- There is one line for each direct child of account Selling Expenses (3100): Advertising, Promotional Expenses, Joint Marketing, and Bonuses/Commisions.
- The line for Advertising is the sum of itself plus all of its children. If it has multiple levels of children, they are all added into the sum. The other direct children of 3100 do not themselves have children, so the sum on each of those lines consists of only the parent value.

<table>
<thead>
<tr>
<th></th>
<th>Month Actual</th>
<th>YTD Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selling Expenses</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Advertising</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TV/Radio</td>
<td>1,049,146</td>
<td>2,954,342</td>
</tr>
<tr>
<td>Print Media</td>
<td>244,589</td>
<td>721,448</td>
</tr>
<tr>
<td>Internet Advertising</td>
<td>9,542</td>
<td>29,578</td>
</tr>
<tr>
<td>Promotional Expenses</td>
<td>53,719</td>
<td>151,732</td>
</tr>
<tr>
<td>Joint Marketing</td>
<td>97,135</td>
<td>289,799</td>
</tr>
<tr>
<td>Bonuses/Commisions</td>
<td>100,188</td>
<td>304,199</td>
</tr>
</tbody>
</table>

| Selling Expenses     |            |            |
| Advertising          |            |            |
| TV/Radio             | 1,303,277  | 3,705,368  |
| Promotional Expenses | 53,719     | 151,732    |
| Joint Marketing      | 97,135     | 289,799    |
| Bonuses/Commisions   | 100,188    | 304,199    |
Using GET CHILDREN instead of WITH CHILDREN eliminates the top line from each portion of the output. The remaining lines are the same.

<table>
<thead>
<tr>
<th>Advertising</th>
<th>Month</th>
<th>YTD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Actual</td>
<td>Actual</td>
</tr>
<tr>
<td>TV/Radio</td>
<td>1,049,146</td>
<td>2,954,342</td>
</tr>
<tr>
<td>Print Media</td>
<td>244,589</td>
<td>721,448</td>
</tr>
<tr>
<td>Internet Advertising</td>
<td>9,542</td>
<td>29,578</td>
</tr>
<tr>
<td>Promotional Expenses</td>
<td>53,719</td>
<td>151,732</td>
</tr>
<tr>
<td>Joint Marketing</td>
<td>97,135</td>
<td>289,799</td>
</tr>
<tr>
<td>Bonuses/Commissions</td>
<td>100,188</td>
<td>304,199</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Advertising</th>
<th>Month</th>
<th>YTD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Actual</td>
<td>Actual</td>
</tr>
<tr>
<td>Advertising</td>
<td>1,303,277</td>
<td>3,705,368</td>
</tr>
<tr>
<td>Promotional Expenses</td>
<td>53,719</td>
<td>151,732</td>
</tr>
<tr>
<td>Joint Marketing</td>
<td>97,135</td>
<td>289,799</td>
</tr>
<tr>
<td>Bonuses/Commissions</td>
<td>100,188</td>
<td>304,199</td>
</tr>
</tbody>
</table>
The following request displays a consolidated line for account 2000 and each of its direct children and grandchildren.

```sql
SET FORMULTIPLE=ON
JOIN SYS_ACCOUNT IN CENTGL TO ALL SYS_ACCOUNT IN CENTSYSF
TABLE FILE CENTGL
SUM NAT_AMOUNT/D10.0 NAT_YTDAMT/D10.0
FOR GL_ACCOUNT
2000 WITH CHILDREN 2 ADD AS CAPTION
IF PERIOD EQ '2002/03'
END
```

The output is shown as follows.

![Chart Image]

### Loading a Hierarchy Manually

In most cases, a hierarchy loads automatically as a result of the request syntax. However, if you need to use a hierarchy defined in one Master File against a data source that is not joined to the hierarchy file (but that contains the same hierarchy field), you can manually load the hierarchy data using the LOAD CHART command.

The number of charts that can load is limited by available memory. Charts automatically unload when the session ends.

The chart loads by running a TABLE request that produces a list of parent values and their associated children.

```sql
TABLE FILE chartfile
BY parentfield BY hierarchyfield
[SUM captionfield]
END
```
The resulting chart contains the following information. It may also contain the associated captions, depending on whether the AS CAPTION phrase was used in the request.

<table>
<thead>
<tr>
<th>parentfield</th>
<th>hierarchyfield</th>
</tr>
</thead>
<tbody>
<tr>
<td>parentvalue1</td>
<td>child1</td>
</tr>
<tr>
<td>parentvalue1</td>
<td>child2</td>
</tr>
<tr>
<td>parentvalue2</td>
<td>child3</td>
</tr>
</tbody>
</table>

**Syntax:** How to Load a Hierarchy From One Master File for Use With a Separate Master File

You can manually load the hierarchy data, if you need to use a hierarchy defined in one Master File, against a data source that is not joined to the hierarchy file but that contains the same hierarchy field.

Available memory dictates the number of charts that can load. Charts automatically unload when WebFOCUS terminates.

`LOAD CHART chartfile[.sega].hierarchyfld`  
`[FOR requestfile[.segb].fieldb]`

where:

- **chartfile**
  Is the name of the Master File that contains the hierarchy information.

- **sega**
  Is the name of the segment that contains the hierarchy field. The segment name is only required if a field in another segment in the structure has the same field name as the hierarchy field.

- **hierarchyfld**
  Is the hierarchy field. It is required because a Master File can define multiple hierarchies.

- **FOR**
  Loads a hierarchy defined in a Master File that is not used in the FML report request. For example, if Master File B contains the hierarchy information but Master File A is used in the request (without a join between Master Files A and B), issue the following LOAD CHART command prior to the FML request:

`LOAD CHART B.FLDB FOR A.FLDA`  
`TABLE FILE A ...`
**requestfile**

Is the name of the Master File used in the FML request.

**segb**

Is the name of the segment that contains the hierarchy field values in the Master File used in the FML request. It is not required if it has the same name as `sega`.

**fieldb**

Is the field in the Master File specified in the FML request that contains the values of the hierarchy field. It is not required if it has the same name as the hierarchy field.

**Note:**

- If you issue the LOAD CHART command multiple times for the same hierarchy, the new hierarchy overlays the previous version in memory.
- If you issue the LOAD CHART command for a data source that is dynamically joined to the hierarchy file, you must issue the JOIN command prior to issuing the LOAD CHART command.

**Reference:**  Usage Notes for FML Hierarchies

- PROPERTY and REFERENCE are propagated to HOLD Master Files when HOLDATTR is set to ON.
- The following setting is required in order to use a data record in more than one row of an FML request (for example, both a detail and summary row):
  
  ```sql
  SET FORMULTIPLE=ON
  ```
- When reporting against a rolled-up data source such as ESSBASE, the data values stored for the parent instance are an aggregate of all of its children. Do not use the ADD feature on consolidated data.
- When reporting against a data source with shared members (such as ESSBASE), in which the same data can be defined multiple times with different hierarchy field values, data shared by two different parents is counted twice in an aggregation operation. To avoid this double aggregation, use the FST operator in the SUM command for the shared fields.

**Customizing a Row Title**

You can customize a row title in an FML report for accurate data identification. Using the AS phrase, you can provide new titles for TAG, DATA, RECAP, and PICKUP rows.
**Syntax:** How to Customize a Row Title in FML

For a TAG row, use the syntax:

```
value AS {'title'|CAPTION}
```

For a DATA or PICKUP row, use the syntax:

```
value AS 'title'
```

For a RECAP row, use the syntax:

```
RECAP calcname[/format]=expression; AS 'title'
```

where:

- **value**
  - Is the value on which you are reporting, whether retrieved from a data source or external file (represented by a tag), supplied directly by a user in the request, or picked up from a work file.

- **title**
  - Is the customized row title, enclosed in single quotation marks if it contains embedded blanks.

In a TAG, DATA, or PICKUP row, the default row title is `value`.

In a RECAP row, the default title is `calcname`.

**CAPTION**

In the Master File of a hierarchical data source, CAPTION identifies a TAG row using a caption field. Note that the hierarchy in the Master File defines the PARENT-OF the FOR field.

- **calcname**
  - Is the value that is derived by the RECAP calculation.

**Example:** Changing the Titles of Tag Rows

In the following example, the row titles CASH ON HAND and DEMAND DEPOSITS provide meaningful identifications for the corresponding tags.

```
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 AS 'CASH ON HAND' OVER
1020 AS 'DEMAND DEPOSITS'
END
```

Note that single quotation marks are necessary since the row title being assigned has embedded blanks.
The output is shown as follows.

<table>
<thead>
<tr>
<th>AMOUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASH ON HAND      8,784</td>
</tr>
<tr>
<td>DEMAND DEPOSITS   4,494</td>
</tr>
</tbody>
</table>

If no AS phrase is included, the tag values are displayed in the report.

**Example:** Customizing a Row Title for a RECAP Value

This request creates the title TOTAL CASH for the RECAP value TOTCASH.

```
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 AS 'CASH ON HAND' OVER
1020 AS 'DEMAND DEPOSITS' OVER
1030 AS 'TIME DEPOSITS' OVER
RECAP TOTCASH = R1 + R2 + R3; AS 'TOTAL CASH'
END
```

The output is shown as follows.

<table>
<thead>
<tr>
<th>AMOUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASH ON HAND      8,784</td>
</tr>
<tr>
<td>DEMAND DEPOSITS   4,494</td>
</tr>
<tr>
<td>TIME DEPOSITS     7,961</td>
</tr>
<tr>
<td>TOTAL CASH       21,239</td>
</tr>
</tbody>
</table>

If no AS phrases are included, the name of the RECAP value (TOTCASH) displays in the report.

**Formatting an FML Report**

Improve the readability and presentation of your FML report by:

- **Underlining numeric columns.** Reports with columns of numbers frequently need to display underlines before some RECAP calculations. You can specify an underline character, introduced by the word BAR, in place of the tag value.

- **Adding page breaks.** You can request a new page at any point in a report by placing the word PAGE-BREAK in place of the tag value.

- **Formatting rows, columns, and cells.** You can apply StyleSheet attributes, such as FONT, SIZE, STYLE, and COLOR, to individual rows and columns, or to cells within those rows.

- **Adding borders around rows, columns, and cells.** You can use BORDER attributes in a StyleSheet to specify the weight, style, and color of border lines around a row or cell. You can specify formatting variations for the top, bottom, left, and right borders.
Indenting text or numbers. You can indent a tag value, label text, or caption text a specified number of spaces for an FML tag row, hierarchy, or RECAP row. If you apply the indent to rows in an FML hierarchy, the parent line of the hierarchy is indented the number of spaces specified as the indent.

Note: For an HTML, PDF, or PostScript report, you can use the BLANKINDENT setting to specify an indentation between levels of an FML hierarchy. See Indenting Row Titles in an FML Hierarchy on page 253.

Syntax: How to Add an Underline Character for Columns

BAR [AS 'character'] OVER

where:

character

Is either the hyphen character (-) or the equal character (=). Enclose the character in single quotation marks. The default character is the hyphen (-).

Example: Underlining Columns

This example uses the default underscore character (_).

TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 AS 'CASH ON HAND' OVER
1020 AS 'DEMAND DEPOSITS' OVER
1030 AS 'TIME DEPOSITS' OVER
BAR OVER
RECAP TOTCASH = R1 + R2 + R3;
END

The output is shown as follows.

AMOUNT
-----
CASH ON HAND 8,784
DEMAND DEPOSITS 4,494
TIME DEPOSITS 7,961
-----
TOTCASH 21,239

Notice that the BAR ... OVER phrase underlines only the column containing the display field.
Syntax:  **How to Specify a Page Break in an FML Report**

Include the following syntax in the FML request in place of a tag value.

```
PAGE-BREAK OVER
```

**Example:**  **Specifying a Page Break in an FML Report**

In this example, a page break is inserted after the first two RECAP commands to highlight each calculation.

```
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 AS 'CASH ON HAND' OVER
1020 AS 'DEMAND DEPOSITS' OVER
1030 AS 'TIME DEPOSITS' OVER
BAR OVER
RECAP TOTCASH = R1 + R2 + R3; AS 'TOTAL CASH' OVER
PAGE-BREAK OVER
1100 AS 'ACCOUNTS RECEIVABLE' LABEL RECEIVE OVER
1200 AS 'INVENTORY' LABEL INVENT OVER
BAR OVER
RECAP TOTASSET = RECEIVE + INVENT; AS 'TOTAL ASSETS' OVER
PAGE-BREAK OVER
RECAP TOTAL = TOTCASH + TOTASSET;
END
```
The output is shown as follows.

```
<table>
<thead>
<tr>
<th></th>
<th>AMOUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASH ON HAND</td>
<td>8,784</td>
</tr>
<tr>
<td>DEMAND DEPOSITS</td>
<td>4,494</td>
</tr>
<tr>
<td>TIME DEPOSITS</td>
<td>7,961</td>
</tr>
</tbody>
</table>

TOTAL CASH             | 21,239 |
```

```
<table>
<thead>
<tr>
<th></th>
<th>AMOUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCOUNTS RECEivable</td>
<td>18,829</td>
</tr>
<tr>
<td>INVENTORY</td>
<td>27,307</td>
</tr>
</tbody>
</table>

TOTAL ASSETS                | 46,136 |
```

```
<table>
<thead>
<tr>
<th></th>
<th>AMOUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOTAL</td>
<td>67,375</td>
</tr>
</tbody>
</table>
```

**Syntax:** How to Format a Row, Column, or Cell in an FML Report

```
TYPE=type, [COLUMN=column] [LABEL={Rn|label}], format_def, $
```

where:

- **type**
  - Identifies the component you wish to format. Valid values are:
    - **REPORT** denotes a row with the specified label.
    - **DATA** denotes a row with the specified label, which contains user-supplied data values.
    - **FREETEXT** denotes a free text or a blank row with the specified label.
UNDERLINE denotes underlines generated by BAR. Formatting of an underline is supported for PDF and PS, but not for HTML reports.

**column**

Identifies a specific column. You can identify the column by its name or position in a row.

**LABEL**

Is the controlling factor in identifying and formatting an FML row.

Note that the label is used to identify a row for calculation or formatting. The label for a TAG or DATA row never appears in the report output. It is used only to identify rows within the FML code. For a RECAP row, the name of the calculated value serves as a label. It appears in the report unless an alternate title is specified.

**label** is an explicit row label that you can assign to identify a row more clearly.

**format_def**

Is the formatting definition, such as FONT, SIZE, STYLE, and COLOR. See *Formatting an FML Report* on page 234.

**Note:** To format a cell, identify the cell as the intersection of a column and a row using COLUMN= ... plus LABEL= ... in the same StyleSheet declaration.

**Example:**  Formatting Rows in an FML Report

The following illustrates how to identify and format an entire FML row, consisting of the row label and the row data. The LABEL attribute in the StyleSheet identifies the three TAG rows, which are styled here as italic.

```FML
SET PAGE-NUM=OFF
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 AS 'CASH ON HAND' LABEL COH OVER
1020 AS 'DEMAND DEPOSITS' LABEL DD OVER
1030 AS 'TIME DEPOSITS' LABEL TD OVER
BAR OVER
RECAP TOTCASH = R1 + R2 + R3; AS 'TOTAL CASH'
ON TABLE SET STYLESHEET *
TYPE = REPORT, GRID = OFF, $
TYPE = REPORT, LABEL = COH, STYLE = ITALIC, $
TYPE = REPORT, LABEL = DD,  STYLE = ITALIC, $
TYPE = REPORT, LABEL = TD,  STYLE = ITALIC, $
ENDSTYLE
END
```

238 WebFOCUS
Applying Boldface to a TAG Row in an FML Report

Example: Applying Boldface to a TAG Row in an FML Report

This request applies boldface to the customized row title, CASH, and to the related data in the AMOUNT column. The StyleSheet uses the explicit label CA to identify the component to format.

```
SET PAGE-NUM=OFF
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
10$ AS 'CASH' LABEL CA OVER
1100 AS 'ACCOUNTS RECEIVABLE' LABEL AR OVER
1200 AS 'INVENTORY' LABEL INV OVER
RECAP CURASST/I5C = CA + AR + INV;
ON TABLE SET STYLESHEET *
  TYPE = REPORT, GRID = OFF, $
  TYPE = REPORT, LABEL = CA, STYLE = BOLD, $
ENDSTYLE
END
```

The output is shown in the following image.

<table>
<thead>
<tr>
<th>Amount</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CASH</td>
<td>21,239</td>
</tr>
<tr>
<td></td>
<td>ACCOUNTS RECEIVABLE</td>
<td>18,829</td>
</tr>
<tr>
<td></td>
<td>INVENTORY</td>
<td>27,307</td>
</tr>
<tr>
<td></td>
<td>CURASST</td>
<td>67,375</td>
</tr>
</tbody>
</table>
**Example:** Applying Boldface to a Cell in an FML Matrix

This request generates a report in which the data value for AMOUNT is bold in the row titled CASH. However, the row title CASH is not bold. This is accomplished by pinpointing the cell in the StyleSheet declaration. In this case, the column (N2) within the row (CA).

```FML
SET PAGE-NUM=OFF
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
10$$ AS 'CASH' LABEL CA OVER
1100 AS 'ACCOUNTS RECEIVABLE' LABEL AR OVER
1200 AS 'INVENTORY' LABEL INV OVER
RECAP CURASST/I5C = CA + AR + INV;
ON TABLE SET STYLESHEET *
TYPE = REPORT, GRID = OFF, $
TYPE = REPORT, COLUMN = N2, LABEL = CA, STYLE = BOLD, $
ENDSTYLE
END
```

The output is shown in the following image.

<table>
<thead>
<tr>
<th>AMOUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASH</td>
</tr>
<tr>
<td>ACCOUNTS RECEIVABLE</td>
</tr>
<tr>
<td>INVENTORY</td>
</tr>
<tr>
<td>CURASST</td>
</tr>
</tbody>
</table>
### Example: Applying Boldface to a Column in an FML Report

This request identifies the AMOUNT column by name and formats its title and data in bold. The same result is achieved if the column is identified as N2.

```
SET PAGE-NUM=OFF
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
"---CASH ACCOUNTS---" OVER
1010 AS 'CASH ON HAND' OVER
1020 AS 'DEMAND DEPOSITS' OVER
1030 AS 'TIME DEPOSITS' OVER
" " OVER
"---OTHER CURRENT ASSETS---" OVER
1100 AS 'ACCOUNTS RECEIVABLE' OVER
1200 AS 'INVENTORY'
ON TABLE SET STYLESHEET *
TYPE = REPORT, GRID = OFF, $
TYPE = REPORT, COLUMN = AMOUNT, STYLE = BOLD, $
ENDSTYLE
END
```

The output is shown in the following image.

<table>
<thead>
<tr>
<th>AMOUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>---CASH ACCOUNTS---</td>
</tr>
<tr>
<td>CASH ON HAND</td>
</tr>
<tr>
<td>DEMAND DEPOSITS</td>
</tr>
<tr>
<td>TIME DEPOSITS</td>
</tr>
<tr>
<td>---OTHER CURRENT ASSETS---</td>
</tr>
<tr>
<td>ACCOUNTS RECEIVABLE</td>
</tr>
<tr>
<td>INVENTORY</td>
</tr>
</tbody>
</table>
**Example:** Applying Boldface to a Free Text Row

This request styles the free text as bold. Since in this example the same styling applies to both free text rows, labels are not required to distinguish between them.

```
SET PAGE-NUM=OFF
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
"---CASH ACCOUNTS---" LABEL CA OVER
1010 AS 'CASH ON HAND' OVER
1020 AS 'DEMAND DEPOSITS' OVER
1030 AS 'TIME DEPOSITS' OVER
" " OVER
"---OTHER CURRENT ASSETS---" LABEL OCA OVER
1100 AS 'ACCOUNTS RECEIVABLE' OVER
1200 AS 'INVENTORY'
ON TABLE SET STYLESHEET *
TYPE = REPORT, GRID = OFF, $
TYPE = FREETEXT, STYLE = BOLD, $
ENDSTYLE
END
```

The output is shown in the following image.

<table>
<thead>
<tr>
<th>AMOUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>---CASH ACCOUNTS---</td>
</tr>
<tr>
<td>CASH ON HAND</td>
</tr>
<tr>
<td>DEMAND DEPOSITS</td>
</tr>
<tr>
<td>TIME DEPOSITS</td>
</tr>
<tr>
<td>---OTHER CURRENT ASSETS---</td>
</tr>
<tr>
<td>ACCOUNTS RECEIVABLE</td>
</tr>
<tr>
<td>INVENTORY</td>
</tr>
</tbody>
</table>
**Example:** Formatting Free Text Rows Separately in an FML Report

This request uses the SIZE attribute to distinguish two lines of free text: CASH ACCOUNTS and OTHER CURRENT ASSETS. The labels CA and OCA are used to identify and format each row separately.

```fml
SET PAGE-NUM=OFF
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
" --- CASH ACCOUNTS ---" LABEL CA OVER
1010 AS 'CASH ON HAND' OVER
1020 AS 'DEMAND DEPOSITS' OVER
1030 AS 'TIME DEPOSITS' OVER
" " OVER
" --- OTHER CURRENT ASSETS ---" LABEL OCA OVER
1100 AS 'ACCOUNTS RECEIVABLE' OVER
1200 AS 'INVENTORY'
ON TABLE SET STYLESHEET *
TYPE = REPORT, GRID = OFF, $
TYPE = FREETEXT, LABEL = CA, STYLE = BOLD, SIZE = 12, $
TYPE = FREETEXT, LABEL = OCA, STYLE = BOLD, SIZE = 10, $
ENDSTYLE
END
```

The output is shown in the following image.

<table>
<thead>
<tr>
<th>AMOUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>---CASH ACCOUNTS---</td>
</tr>
<tr>
<td>CASH ON HAND</td>
</tr>
<tr>
<td>DEMAND DEPOSITS</td>
</tr>
<tr>
<td>TIME DEPOSITS</td>
</tr>
<tr>
<td>---OTHER CURRENT ASSETS---</td>
</tr>
<tr>
<td>ACCOUNTS RECEIVABLE</td>
</tr>
<tr>
<td>INVENTORY</td>
</tr>
</tbody>
</table>
**Example:** Styling Text and a Variable in a Free Text Row

In this example, the text and variable components of the free text row are styled separately. The text, Current Assets, is italic and the value derived from the RECAP calculation is bold.

```
SET PAGE-NUM=OFF
TABLE FILE LEDGER
SUM AMOUNT AS 'Amount' FOR ACCOUNT
10$$ AS 'Cash' LABEL CA OVER
1100 AS 'Accounts Receivable' LABEL AR OVER
1200 AS 'Inventory' LABEL INV OVER
RECAP CURASST/I5C = CA + AR + INV; NOPRINT OVER
"Current Assets: <CURASST"
ON TABLE SET STYLESHEET *
TYPE = REPORT, GRID=OFF, $ TYPE = FREETEXT, OBJECT = TEXT, ITEM = 1, SIZE = 12, STYLE = ITALIC, $ TYPE = FREETEXT, OBJECT = FIELD, ITEM = 1, STYLE = BOLD, $ ENDSTYLE
END
```

The output is shown in the following image.

```
<table>
<thead>
<tr>
<th>AMOUNT</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CASH ON HAND</td>
<td>8,784</td>
</tr>
<tr>
<td>DEMAND DEPOSITS</td>
<td>4,494</td>
</tr>
<tr>
<td>TIME DEPOSITS</td>
<td>7,961</td>
</tr>
<tr>
<td>TOTAL CASH</td>
<td>21,239</td>
</tr>
</tbody>
</table>
```
**Example:** Applying Boldface to an FML RECAP Row

This request applies boldface to the row title and calculated value in a RECAP row. Notice that the RECAP label in the StyleSheet is TOTCASH. In a RECAP, the name assigned to the calculated value serves as the explicit label.

```plaintext
SET PAGE-NUM=OFF
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 AS 'CASH ON HAND' LABEL CASH OVER
1020 AS 'DEMAND DEPOSITS' LABEL DD OVER
1030 AS 'TIME DEPOSITS' LABEL TD OVER
RECAP TOTCASH = R1 + R2 + R3; AS 'TOTAL CASH'
ENDSTYLE

The output is shown in the following image.

<table>
<thead>
<tr>
<th>AMOUNT</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CASH ON HAND</td>
<td>8,784</td>
</tr>
<tr>
<td>DEMAND DEPOSITS</td>
<td>4,494</td>
</tr>
<tr>
<td>TIME DEPOSITS</td>
<td>7,961</td>
</tr>
<tr>
<td>TOTAL CASH</td>
<td>21,239</td>
</tr>
</tbody>
</table>
**Syntax:**

**How to Add and Format Row and Cell Borders**

To request a uniform border around a row or cell, use this syntax:

```
TYPE=REPORT, LABEL=row_label, [COLUMN=column,] BORDER=option,
[BORDER-STYLE=line_style,] [BORDER-COLOR={color|RGB(r g b)},] $
```

To specify different characteristics for the top, bottom, left, and/or right borders, use this syntax:

```
TYPE=REPORT, LABEL=row_label, [COLUMN=column,] BORDER-position=option,
[BORDER-position-STYLE=line_style,]
[BORDER-position-COLOR={color|RGB(r g b)},] $
```

To specify different characteristics for the top, bottom, left, and/or right borders, use this syntax:

```
TYPE=REPORT, LABEL=row_label, [COLUMN=column,] BORDER-position=option,
[BORDER-position-STYLE=line_style,]
[BORDER-position-COLOR={color|RGB(r g b)},] $ 
```

where:

- **row_label**
  
  Is the row to which the specified border characteristics are applied.

- **column**
  
  Used in conjunction with row label. Designates a cell (at the point of intersection of the row and the column) to which the specified border characteristics are applied.

- **option**
  
  Can be one of the following values:

  - **ON** turns borders on for the entire heading or footing. **ON** generates the same line as **MEDIUM**.
  
  - **OFF** turns borders off for the entire heading or footing. **OFF** is the default.
  
  - **LIGHT** specifies a thin line. You can specify a light line for the entire heading or footing, or for one or more border positions.
  
  - **MEDIUM** identifies a medium line (ON sets the line as MEDIUM). You can specify a light line for the entire heading or footing, or for one or more border positions. Note that the medium line setting ensures consistency with lines created with GRID attributes.
  
  - **HEAVY** identifies a thick line. You can specify a heavy line for the entire heading or footing, or for one or more border positions.

  - **width** specifies the line width in points (where 72 pts=1 inch). You can specify a line width in points for the entire heading or footing or for one or more border positions. Line width specified in points is displayed differently in HTML and PDF output. For uniform appearance, regardless of display format, use **LIGHT**, **MEDIUM**, or **HEAVY**.
**position**

Specifies which border line to format. Valid values are TOP, BOTTOM, LEFT, RIGHT.

You can specify a position qualifier for any of the BORDER keywords. This enables you to format line width, line style, and line color individually, for any side of the border.

**line_style**

Sets the style of the border line. WebFOCUS StyleSheets support all of the standard Cascading Style Sheets line styles. Several 3-dimensional styles are only available in HTML, as noted by asterisks. Valid values are:

- NONE
- SOLID
- DOTTED
- DASHED
- DOUBLE*
- GROOVE*
- RIDGE*
- INSET*
- OUTSET*

**color**

Is one of the preset color values. The default value is BLACK.

If the display or output device does not support colors, it substitutes shades of gray.

**RGB**

Specifies the font color using a mixture of red, green, and blue.

(r g b)

Is the desired intensity of red, green, and blue, respectively. The values are on a scale of 0 to 255, where 0 is the least intense and 255 is the most intense. Note that using the three color components in equal intensities results in shades of gray.

**Note:** For HTML reports, the BORDERS feature requires that cascading style sheets be turned ON. This code is not required for PDF and PS reports.
**Example:** Emphasizing a Row Using Uniform Border Lines

This example places a dashed border of medium thickness around the RECAP row identified by the label TOTCASH. For HTML reports, the BORDERS feature requires that cascading style sheets be turned ON.

```
SET PAGE-NUM=OFF
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 AS 'CASH ON HAND' LABEL CASH OVER
1020 AS 'DEMAND DEPOSITS' LABEL DD OVER
1030 AS 'TIME DEPOSITS' LABEL TD OVER
RECAP TOTCASH = R1 + R2 + R3; AS 'TOTAL CASH'
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLESHEET *
  TYPE = REPORT, GRID = OFF, $
  TYPE = REPORT, LABEL = TOTCASH, BORDER = MEDIUM,
    BORDER-STYLE = DASHED, $
ENDSTYLE
END
```

The output is shown in the following image.

```
AMOUNT
CASH ON HAND  8,784
DEMAND DEPOSITS  4,494
TIME DEPOSITS  7,961
*TOTAL CASH : 21,239
```
Example: Emphasizing a Row Using Different Top/Bottom and Left/Right Borders

This example places a heavy black border line above and below the RECAP row identified by the label TOTCASH, and a thin silver dotted line to the left and right of each column in the row.

For HTML reports, the BORDERS feature requires that cascading style sheets be turned ON.

```
SET PAGE-NUM=OFF
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 AS 'CASH ON HAND' LABEL CASH OVER
1020 AS 'DEMAND DEPOSITS' LABEL DD OVER
1030 AS 'TIME DEPOSITS' LABEL TD OVER
RECAP TOTCASH = R1 + R2 + R3; AS 'TOTAL CASH'
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLESHEET *
TYPE = REPORT, GRID = OFF,$
TYPE = REPORT, LABEL = TOTCASH,
  BORDER-TOP = HEAVY,
  BORDER-BOTTOM = HEAVY,
  BORDER-LEFT = LIGHT,
  BORDER-RIGHT = LIGHT,
  BORDER-TOP-STYLE = SOLID,
  BORDER-BOTTOM-STYLE = SOLID,
  BORDER-LEFT-STYLE = DOTTED,
  BORDER-RIGHT-STYLE = DOTTED,
  BORDER-LEFT-COLOR = 'SILVER',
  BORDER-RIGHT-COLOR = 'SILVER', $
ENDSTYLE
END
```

The output is shown in the following image.

<table>
<thead>
<tr>
<th>Amount</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CASH ON HAND</td>
<td>8,784</td>
</tr>
<tr>
<td>DEMAND DEPOSITS</td>
<td>4,494</td>
</tr>
<tr>
<td>TIME DEPOSITS</td>
<td>7,961</td>
</tr>
<tr>
<td>TOTAL CASH</td>
<td>21,239</td>
</tr>
</tbody>
</table>
**Example:** Adding Uniform Border Lines Around a Cell

This example places a border of medium thickness around the cell in the second column of the row identified by the label TOTCASH. The combined LABEL and COLUMN specifications are identified in the cell. The BORDERS feature requires that cascading style sheets be turned ON.

```
SET PAGE-NUM=OFF
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 AS 'CASH ON HAND' LABEL CASH OVER
1020 AS 'DEMAND DEPOSITS' LABEL DD OVER
1030 AS 'TIME DEPOSITS' LABEL TD OVER
RECAP TOTCASH = R1 + R2 + R3; AS 'TOTAL CASH'
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLESHEET *
TYPE = REPORT, GRID = OFF,$
TYPE = REPORT, LABEL = TOTCASH, COLUMN = N2, BORDER = MEDIUM, $
ENDSTYLE
END
```

The output is shown in the following image.

```
<table>
<thead>
<tr>
<th>AMOUNT</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CASH ON HAND</td>
<td>8,784</td>
</tr>
<tr>
<td>DEMAND DEPOSITS</td>
<td>4,494</td>
</tr>
<tr>
<td>TIME DEPOSITS</td>
<td>7,961</td>
</tr>
<tr>
<td>TOTAL CASH</td>
<td>21,239</td>
</tr>
</tbody>
</table>
```

**Syntax:** How to Specify an Indent for an FML Label, Tag, or Caption

```
FOR forfield [IN k]
tag [[GET CHILDREN|WITH CHILDREN] n] INDENT m [AS ['text'|CAPTION]] [OVER]

or

RECAP fieldname[/format]=expression; INDENT m [AS 'text']
```

where:

`forfield`

Is a field in the data source whose values are included in the report.
\( k \)
Is the starting column for the FOR value in an FML report.

\( \text{tag} \)
Is a value of \textit{forfield} to be displayed on a row of the FML report.

\( n \)
Is the number of levels of an FML hierarchy to display on the FML report.

\( m \)
Is a positive integer (zero is not supported) specifying the number of spaces to indent the tag value, label, or caption of an FML row or hierarchy. The indentation starts from column one if there is no IN phrase specified in the FOR command. If there is an IN phrase, indentation starts from the column specified in the IN phrase. The maximum indentation is the same as the maximum length of an AS name.

If you indent an FML hierarchy, the parent line of the hierarchy is indented the number of spaces specified as the indent. The hierarchy levels are indented two spaces from each other. If the GET CHILDREN phrase is used, the first line of the hierarchy is indented an additional two spaces because the hierarchy output begins with the first child rather than the parent. For more information about the use of GET CHILDREN, see Displaying an FML Hierarchy on page 219.

\( \text{'text'} \)
Is a label to be displayed on a row of the FML report.

\( \text{CAPTION} \)
Indicates that a caption field has been defined in the Master File.

\( \text{OVER} \)
Indicates that this row is not the last row to be displayed.

\( \text{fieldname} \)
Is a name you assign to the value calculated by the RECAP command.

\( \text{format} \)
Is the USAGE format for RECAP field. It cannot exceed the column width. The default is the format of the column in which the calculated value is displayed.

\( \text{expression} \)
Is the expression that describes how to calculate the field value for RECAP.
Example: Indenting a Tag Row in an FML Hierarchy

In the following request, the label of the second row for tag value 3000 is indented five spaces. Because the GET CHILDREN phrase is used, the first line of the FML hierarchy, in the third row for tag value 3000, is indented seven spaces (five + two).

```
SET FORMULTIPLE=ON
TABLE FILE CENTGL
PRINT GL_ACCOUNT_PARENT
FOR GL_ACCOUNT
  3000 AS 'Not Indented' OVER
  3000 INDENT 5 AS 'Indented 5' OVER
  3000 GET CHILDREN 2 INDENT 5 AS 'Hierarchy Indented 5'
END
```

The output is shown as follows.

```
Parent
------
Not Indented                     3000
   Indented 5                    3000
      Hierarchy Indented 5       3000
         Hierarchy Indented 5   3100
         Hierarchy Indented 5   3100
         Hierarchy Indented 5   3100
         Hierarchy Indented 5   3100
         Hierarchy Indented 5   3100
         Hierarchy Indented 5   3100
         Hierarchy Indented 5   3000
         Hierarchy Indented 5   3200
         Hierarchy Indented 5   3200
         Hierarchy Indented 5   3200
         Hierarchy Indented 5   3200
         Hierarchy Indented 5   3200
         Hierarchy Indented 5   3200
         Hierarchy Indented 5   3200
         Hierarchy Indented 5   3200
         Hierarchy Indented 5   3200
         Hierarchy Indented 5   3200
```
**Example:** Indenting FML RECAP Rows

The following request sums price, cost, and quantity in stock for digital and analog product types. The first RECAP command calculates the total for each column, and indents the label five spaces. The second RECAP command calculates the profit, and indents the label 10 spaces.

```
SET FORMULTIPLE=ON
TABLE FILE CENTINV
SUM PRICE COST QTY_IN_STOCK
FOR PRODTYPE
Digital
Analog
BAR
RECAP TOTAL = R1 + R2; INDENT 5  AS 'Total:' OVER
BAR
RECAP PROFIT(2) = TOTAL(1) - TOTAL(2); AS 'Profit:' INDENT 10
END
```

The output is shown as follows.

<table>
<thead>
<tr>
<th>Price:</th>
<th>Cost:</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital</td>
<td>4,080.00</td>
<td>3,052.00</td>
</tr>
<tr>
<td>Analog</td>
<td>1,883.00</td>
<td>1,371.00</td>
</tr>
<tr>
<td>Total:</td>
<td>5,963.00</td>
<td>4,423.00</td>
</tr>
<tr>
<td>Profit:</td>
<td>1,540.00</td>
<td></td>
</tr>
</tbody>
</table>

**Indenting Row Titles in an FML Hierarchy**

To clarify relationships within an FML hierarchy, the captions (titles) of values are indented at each level. Use the BLANKINDENT parameter in an HTML, PDF, or PostScript report to specify the indentation between each level in the hierarchy. You can use the default indentation for each hierarchy level or choose your own indentation value. To print indented captions in an HTML report, you must set the BLANKINDENT parameter to ON or to a number.

SET BLANKINDENT does not redefine the width of the indented column, if it is not wide enough to accommodate the indented fields. Columns in table-based formats will automatically size themselves as needed, while columns in position-based formats, such as PDF, PostScript, or PPTX, shift out of alignment. You can use StyleSheet syntax to make the column wide enough for the indented values, and move the columns that follow it. Change the width of a column using the StyleSheet SQUEEZE = \( n \) attribute to supply the required space.

A related feature enables you to change the number of blank spaces before the parent line of a hierarchy or before any FML tag or RECAP row in any FML request. For more information, see *Formatting an FML Report* on page 234.
**Syntax:** How to Indent FML Hierarchy Captions in an HTML Report

```plaintext
SET BLANKINDENT={ON|OFF|n}
ON TABLE SET BLANKINDENT {ON|OFF|n}
```

where:

**ON**
- Indents FML hierarchy captions 0.125 units for each space that normally displays before the caption. For child levels in an FML hierarchy, it indents 0.125 units for each space that normally displays between this line and the line above it.

**OFF**
- Turns off indentations for FML hierarchy captions in an HTML report. OFF is the default value.
  - For other formats, uses the default indentation of two spaces.

**n**
- Is an explicit measurement in the unit of measurement defined by the UNITS parameter. This measurement is multiplied by the number of spaces that normally displays before the caption. For child levels in an FML hierarchy, it indents \( n \) units for each space that normally displays between this line and the line above it. The default number of spaces is two. Zero (0) produces the same report output as OFF. Negative values for \( n \) are not supported.

**Example:** Using the Default Indentation for FML Hierarchy Captions

The following request creates an HTML report with the default indentation:

```plaintext
SET PAGE-NUM=NOPAGE
SET BLANKINDENT=ON
SET FORMULTIPLE=ON
TABLE FILE CENTGL
PRINT GL_ACCOUNT_PARENT
FOR GL_ACCOUNT
3000                  AS CAPTION      OVER
3000 GET  CHILDREN 2  AS CAPTION
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLE *
TYPE = REPORT, GRID = OFF, $
ENDSTYLE
END
```
Specifying an Indentation Value for FML Hierarchy Captions

The following request specifies an indentation of .25 for each level of an FML hierarchy. This number is expressed in the default unit of measurement, which is inches:

```
SET PAGE-NUM=NOPAGE
SET BLANKINDENT=.25
SET FORMULTIPLE=ON
TABLE FILE CENTGL
PRINT GL_ACCOUNT_PARENT
FOR GL_ACCOUNT
  3000                 AS CAPTION          OVER
  3000 GET  CHILDREN 2 AS CAPTIONON TABLE SET STYLE *
  TYPE = REPORT, GRID = OFF, $
ENDSTYLE
END
```

**Example:** Specifying an Indentation Value for FML Hierarchy Captions

The following request specifies an indentation of .25 for each level of an FML hierarchy. This number is expressed in the default unit of measurement, which is inches:

```
SET PAGE-NUM=NOPAGE
SET BLANKINDENT=.25
SET FORMULTIPLE=ON
TABLE FILE CENTGL
PRINT GL_ACCOUNT_PARENT
FOR GL_ACCOUNT
  3000                 AS CAPTION          OVER
  3000 GET  CHILDREN 2 AS CAPTIONON TABLE SET STYLE *
  TYPE = REPORT, GRID = OFF, $
ENDSTYLE
END
```
Suppressing the Display of Rows

The output is shown in the following image.

<table>
<thead>
<tr>
<th></th>
<th>Parent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Operating Expenses</td>
<td>1000</td>
</tr>
<tr>
<td>Selling Expenses</td>
<td>3000</td>
</tr>
<tr>
<td>Advertising</td>
<td>3100</td>
</tr>
<tr>
<td>Promotional Expenses</td>
<td>3100</td>
</tr>
<tr>
<td>Joint Marketing</td>
<td>3100</td>
</tr>
<tr>
<td>Bonuses/Commisions</td>
<td>3100</td>
</tr>
<tr>
<td>General + Admin Expenses</td>
<td>3000</td>
</tr>
<tr>
<td>Salaries-Corporate</td>
<td>3200</td>
</tr>
<tr>
<td>Company Benefits</td>
<td>3200</td>
</tr>
<tr>
<td>Depreciation Expenses</td>
<td>3200</td>
</tr>
<tr>
<td>Gain/(Loss) Sale of Equipment</td>
<td>3200</td>
</tr>
<tr>
<td>Leasehold Expenses</td>
<td>3200</td>
</tr>
<tr>
<td>Interest Expenses</td>
<td>3200</td>
</tr>
<tr>
<td>Utilities</td>
<td>3200</td>
</tr>
</tbody>
</table>

**Suppressing the Display of Rows**

You may sometimes wish to retrieve data in a TAG row solely for use in a calculation, without displaying the row in a report. To suppress the display of a tag row, add the word NOPRINT to the row declaration, as in a TABLE request.

You may also wish to suppress the display of a TAG row if no data is found for the values. For more information, see *Suppressing Rows With No Data* on page 257.

In addition, you can suppress the display of RECAP rows by adding the word NOPRINT to the RECAP command, following the semicolon. This technique is useful to suppress the display of an intermediate RECAP value, which is intended for use as input to other calculations.
Example: Suppressing the Display of a TAG Row

This example uses the value of COST in its computation, but does not display COST as a row in the report.

```
DEFINE FILE REGION
AMOUNT/I5C=E_ACTUAL;
END

TABLE FILE REGION
SUM AMOUNT FOR ACCOUNT
3000 AS 'SALES' LABEL SLS OVER
3100 AS 'COST' LABEL COST NOPRINT OVER
RECAP PROFIT/I5C = SLS - COST; OVER
" "
RECAP ROS/F6.2 = 100*PROFIT/SLS;
AS 'RETURN ON SALES'
END
```

The output is shown in the following image.

<table>
<thead>
<tr>
<th>AMOUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>SALES</td>
</tr>
<tr>
<td>PROFIT</td>
</tr>
<tr>
<td>RETURN ON SALES</td>
</tr>
</tbody>
</table>

|       | 6,000 | 1,350 | 22.50 |

Suppressing Rows With No Data

The text for a tag row is displayed even if no data is found in the file for the tag values, with a period (.) representing the missing data. You can override this convention by adding the phrase WHEN EXISTS to the definition of a TAG row. This makes displaying a row dependent upon the existence of data for the tag. This feature is useful, for example, when the same model is applied to different divisions in a company.
Example: Suppressing Rows With No Data

In this example, assume that the variable DIVISION contains Division 1, a real estate syndicate, and Division 2, a bank. The following request describes their balance sheets in one FML report. Rows that are irrelevant for each division are not displayed.

```FML
TABLE FILE LEDGER
HEADING CENTER
"BALANCE SHEET FOR DIVISION <DIVISION"
""
SUM AMOUNT
BY DIVISION NOPRINT
ON DIVISION PAGE-BREAK
FOR ACCOUNT
2000 AS 'LAND' WHEN EXISTS LABEL LD OVER
2100 AS 'CAR LOANS' WHEN EXISTS LABEL LOAN OVER
.
.
.
```

Saving and Retrieving Intermediate Report Results

Many reports require results developed in prior reports. This can be accomplished only if a place is provided for storing intermediate values. An example is the need to compute net profit in an Income Statement prior to calculating equity in a Balance Sheet. FML can save selected rows from one or more models by posting them to a work file. The posted rows can then be picked up from the work file and reused.

The default work file is FOCPOST. This is a comma-delimited file from which you can report directly if a FOCPOST Master File is available. In order to use the work file in a request, you must assign a physical name to the FOCPOST ddname before running the report that posts the data, and again before running the report that picks up the data.

You can assign the physical name to the file by issuing a FILEDEF command on Windows and UNIX, or a TSO ALLOCATE or DYNAM ALLOCATE command on z/OS, before the request is run. You may create a FILEDEF command by using the Allocation Wizard.

While you cannot prepare an FML report entirely from data that you supply directly in your request, you can prepare a report entirely from data that is stored in a comma-delimited work file.

Posting Data

You can save any TAG, RECAP, or DATA row by posting the output to a file. You can use these rows as though they were provided in a DATA row.
The row is processed in the usual manner in the report, depending on its other options, and then posted. The label of the row is written first, followed by the numeric values of the columns, each comma-separated, and ending with the terminator character ($). For more information, see Posting Rows to a Work File on page 259.

**Note:** Only fields that are actually displayed on the report output are posted. Fields that are not printed (for example, fields specified with the NOPRINT option, extra fields that are created when you reformat fields in the request, or fields implied by use in a calculation) are not posted.

**Syntax:** How to Post Data to a File

The syntax for saving any TAG, RECAP, or DATA row is:

```plaintext
POST [TO ddname]
```

where:

`ddname`

Is the logical name you assign to the work file in which you are posting data.

Add this syntax to any row you wish to post to the work file.

**Example:** Posting Rows to a Work File

The following request creates an FML report, and posts two tag rows to the LEDGEOUT work file.

```plaintext
FILEDEF LEDGEOUT DISK [APP]\LEDGEOUT.DAT
DEFINE FILE LEDGER
CUR_YR/I5C=AMOUNT;
LAST_YR/I5C=.87*CUR_YR - 142;
END
TABLE FILE LEDGER
SUM CUR_YR LAST_YR
FOR ACCOUNT
  1100 LABEL AR POST TO LEDGEOUT OVER
  1200 LABEL INV POST TO LEDGEOUT
END
```

The output is shown in the following image.

<table>
<thead>
<tr>
<th>CUR_YR</th>
<th>LAST_YR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1100</td>
<td>18,829</td>
</tr>
<tr>
<td>1200</td>
<td>27,307</td>
</tr>
<tr>
<td>1100</td>
<td>15,954</td>
</tr>
<tr>
<td>1200</td>
<td>23,329</td>
</tr>
</tbody>
</table>
**Syntax:** How to Pick Up Data From a Work File

You can retrieve posted rows from any work file and use them as if they were provided in a DATA row by adding the following phrase to an FML request.

\[
\text{DATA PICKUP [FROM } ddname [OR id1 \ldots] [LABEL label] [AS 'text']
\]

where:

- **ddname**
  - Is the logical name of the work file from which you are retrieving data.
- **id**
  - Is the label that was assigned in the work file to the posted row of data that is now being picked up.
- **label**
  - Is the label you wish to assign to the data you are picking up.

The label you assign to the picked data can, but is not required to, match the label (id) of the posted data.

You can include LABEL and AS phrases, but WHEN EXISTS is not supported.

**Note:** The retrieved fields are mapped to all fields (printed or not) in the memory repository (internal matrix) of the report. If the matrix contains columns that do not correspond to the fields in the posted file, the retrieved values may be misaligned. For example, if you reformat a field in the PICKUP request, that field will be represented by two columns in the internal matrix. However, the posted file will have only one value representing that field, and the retrieved values will not be mapped properly to the associated columns in the matrix.

**Example:** Picking Up Data From a Work File

In the following example, the data in the LEDGER data source and in the LEDGEOUT work file are used in the RECAP calculation. To see how this file was created, refer to Posting Rows to a Work File on page 259.
Tip: You must assign a logical name to the file by issuing a FILEDEF command on Windows and UNIX, or a DYNAM ALLOCATE command on z/OS, before the request is run. You may create a FILEDEF command by using the Allocation Wizard.

```
DEFINE FILE LEDGER
CUR_YR/I5C=AMOUNT;
LAST_YR/I5C=.87*CUR_YR - 142;
END

TABLE FILE LEDGER
SUM CUR_YR LAST_YR
FOR ACCOUNT
1010 TO 1030 AS 'CASH' LABEL CASH OVER
DATA PICKUP FROM LEDGEOUT AR AS 'ACCOUNTS RECEIVABLE' LABEL AR OVER
DATA PICKUP FROM LEDGEOUT INV AS 'INVENTORY' LABEL INV OVER
BAR OVER
RECAP CUR_ASSET/I5C = CASH + AR + INV;
END
```

The output is shown in the following image.

<table>
<thead>
<tr>
<th></th>
<th>CUR YR</th>
<th>LAST YR</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASH</td>
<td>21,239</td>
<td>17,198</td>
</tr>
<tr>
<td>ACCOUNTS RECEIVABLE</td>
<td>18,829</td>
<td>15,954</td>
</tr>
<tr>
<td>INVENTORY</td>
<td>27,307</td>
<td>23,329</td>
</tr>
<tr>
<td>CUR_ASSET</td>
<td>67,375</td>
<td>56,481</td>
</tr>
</tbody>
</table>

The following line can be used to pick up the sum of the two accounts from LEDGEOUT.

```
DATA PICKUP FROM LEDGEOUT AR OR INV
AS 'ACCTS REC AND INVENTORY'
```

Note: Since the rows in a PICKUP file are stored in standard comma-delimited format, they can be provided either from a prior posting, or directly by a user.

Creating HOLD Files From FML Reports

A report created with FML can be extracted to a HOLD file in the same way as all other reports created with the TABLE language.

In this case, you identify the set of tag values specified for each row by the description field (the AS text supplied in the model). When no text is given for a row, the first tag value is used automatically. Therefore, in simple models with only one tag per row and no text, the lines in the HOLD file contain the single tag value. The rows derived from the RECAP calculation form part of the HOLD file. Pure text rows (including BAR rows) are omitted.
For HOLD to be supported with RECAP, the format of the RECAP field must be the same as the format of the original column.

This feature enables you to create new rows in the HOLD file that are the result of calculations. The augmented HOLD file may then be used in a variety of TABLE requests.

**Note:** You cannot reformat RECAP rows when creating HOLD files.

**Example: Creating a Hold File From an FML Report**

The following request creates a HOLD file that contains records for CASH, ACCOUNTS RECEIVABLE, INVENTORY, and the RECAP row CURRENT ASSETS.

```
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 TO 1030 AS 'CASH' OVER
1100 AS 'ACCOUNTS RECEIVABLE' OVER
1200 AS 'INVENTORY' OVER
RECAP CA = R1 + R2 + R3; AS 'CURRENT ASSETS'
ON TABLE HOLD
END
```

Query the HOLD file:

```
> ? hold
```

**DEFINITION OF HOLD FILE: HOLD**

<table>
<thead>
<tr>
<th>FIELDNAME</th>
<th>ALIAS</th>
<th>FORMAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMOUNT</td>
<td>E01</td>
<td>A 19</td>
</tr>
<tr>
<td>AMOUNT</td>
<td>E02</td>
<td>I5C</td>
</tr>
</tbody>
</table>

Then report from the HOLD file as:

```
TABLE FILE HOLD
PRINT E01 E02
END
```

The output is shown in the following image.

<table>
<thead>
<tr>
<th>AMOUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASH</td>
</tr>
<tr>
<td>ACCOUNTS RECEIVABLE</td>
</tr>
<tr>
<td>INVENTORY</td>
</tr>
<tr>
<td>CURRENT ASSETS</td>
</tr>
<tr>
<td>21,239</td>
</tr>
<tr>
<td>18,829</td>
</tr>
<tr>
<td>27,307</td>
</tr>
<tr>
<td>67,375</td>
</tr>
</tbody>
</table>
Describing Data for an FML Hierarchy

This manual focuses on WebFOCUS financial reporting with the Financial Reporting Painter, supported by the underlying Financial Modeling Language (FML).

It includes detailed information about reporting against hierarchical data structures.

For your convenience, this appendix provides the information you need to understand how these hierarchical relationships between fields are described in a Master File.

As with other Master File attributes, you can define the hierarchy using either a text editor or the Synonym Editor.

For detailed information about all aspects of the data description language, see the Describing Data With WebFOCUS Language manual and/or the Describing Data With Graphical Tools manual.

In this appendix:

- Data Requirements for FML Hierarchies
- Coding an FML Hierarchy in a Text Editor
- Defining an FML Hierarchy in the Synonym Editor

Data Requirements for FML Hierarchies

In the Master File, use the PROPERTY=PARENT_OF and REFERENCE=hierarchyfld attributes to define the hierarchical relationship between two fields.

To use FML hierarchies, the FOR field must either be:

- The hierarchy field.

  or

- Used as the join field to a unique segment that has the hierarchy field. In this case, the hierarchy field must be the join field. Note that the condition that the join be unique only applies if the hierarchy is defined in the cross-referenced segment.

In other words, the FOR field must be in a parent-child hierarchy or it must be linked to a parent-child hierarchy. The latter case allows transaction data that contains the hierarchy field to be joined to a separate data source that contains the hierarchy definition.
Coding an FML Hierarchy in a Text Editor

You can manually code the relationship between parent and child fields in a Master File, and, optionally, provide descriptive captions to display in reports in place of the specified hierarchy field values.

**Syntax:** How to Specify a Hierarchy Between Fields in a Master File

```
FIELD=parentfield,...,PROPERTY=PARENT_OF, REFERENCE=[seg.]hierarchyfld, $
DEFINE name/fmt=expression;,PROPERTY=PARENT_OF,REFERENCE=hierarchyfld, $
```

where:

- **parentfield**
  - Is the parent field in the hierarchy.

- **PROPERTY=PARENT_OF**
  - Identifies this field as the parent of the referenced field in a hierarchy.

  You can specify attributes in every field. Therefore, you can define multiple hierarchies in one Master File. However, an individual field can have only one parent. If multiple fields have PARENT_OF attributes for the same hierarchy field, the first parent found by traversing the structure in top-down, left-to-right order is used as the parent.

- **seg**
  - Is the segment location of the hierarchy field. This is required if more than one segment has a field named *hierarchyfield*.

- **hierarchyfld**
  - Is the child field in the hierarchy.

PARENT_OF is also allowed on a virtual field in the Master File:

```
DEFINE name/fmt=expression;,PROPERTY=PARENT_OF,REFERENCE=hierarchyfld, $
```

**Syntax:** How to Assign Descriptive Captions for Hierarchy Field Values

The following attributes specify a caption for a hierarchy field in a Master File:

```
FIELD=captionfield,...,PROPERTY=CAPTION, REFERENCE=[seg.]hierarchyfld,$
```

For related information, see *Requirements for FML Hierarchies* on page 217.
where:

*captionfield*

Is the name of the field that contains the descriptive text for the hierarchy field. For example, if the employee ID is the hierarchy field, the last name may be the descriptive text to be displayed on the report in place of the ID.

**PROPERTY=CAPTION**

Signifies that this field contains a descriptive caption to be displayed in place of the hierarchy field values.

A caption can be specified for every field, but an individual field can have only one caption. If multiple fields have CAPTION attributes for the same hierarchy field, the first parent found by traversing the structure in top-down, left-to-right order will be used as the caption.

*seg*

Is the segment location of the caption field. This is required if more than one segment has a field named *captionfield*.

*hierarchyfld*

Is the hierarchy field.

CAPTION is also allowed on a virtual field in the Master File:

```
DEFINE name/format=expression;,PROPERTY=CAPTION,REFERENCE=hierarchyfld ,$
```
**Example:** Define a Hierarchy in a Master File

The CENTGL Master File contains a chart of accounts hierarchy. The field GL_ACCOUNT_PARENT is the parent field in the hierarchy. The field GL_ACCOUNT is the hierarchy field. The field GL_ACCOUNT_CAPTION can be used as the descriptive caption for the hierarchy field.

FILE=CENTGL , SUFFIX=FOC
SEGNAME=ACCOUNTS , SEGTYPE=S01
FIELDNAME=GL_ACCOUNT, ALIAS=GLACCT, FORMAT=A7, TITLE='Ledger,Account', FIELDTYPE=I, $
FIELDNAME=GL_ACCOUNT_PARENT, ALIAS=GLPAR, FORMAT=A7, TITLE=Parent, PROPERTY=PARENT_OF, REFERENCE=GL_ACCOUNT, $
FIELDNAME=GL_ACCOUNT_TYPE, ALIAS=GLTYPE, FORMAT=A1, TITLE=Type, $
FIELDNAME=GL_ROLLUP_OP, ALIAS=GLROLL, FORMAT=A1, TITLE=Op, $
FIELDNAME=GL_ACCOUNT_LEVEL, ALIAS=GLLEVEL, FORMAT=I3, TITLE=Lev, $
FIELDNAME=GL_ACCOUNT_CAPTION, ALIAS=GLCAP, FORMAT=A30, TITLE=Caption, PROPERTY=CAPTION, REFERENCE=GL_ACCOUNT, $
FIELDNAME=SYS_ACCOUNT, ALIAS=ALINE, FORMAT=A6, TITLE='System,Account,Line', MISSING=ON, $

The CENTSYSF data source contains detail-level financial data. This is unconsolidated financial data for a fictional corporation, CenturyCorp. It is designed to be separate from the CENTGL database as if it came from an external accounting system. It uses a different account line system (SYS_ACCOUNT) which can be joined to the SYS_ACCOUNT field in CENTGL. Data uses natural signs (expenses are positive, and revenue are negative).

FILE=CENTSYSF , SUFFIX=FOC
SEGNAME=RAWDATA , SEGTYPE=S2
FIELDNAME=SYS_ACCOUNT, A6, FIELDTYPE=I, TITLE='System,Account,Line', $
FIELDNAME=PERIOD, YYM, FIELDTYPE=I, $
FIELDNAME=NAT_AMOUNT, D10.0, TITLE='Month,Actual', $
FIELDNAME=NAT_BUDGET, D10.0, TITLE='Month,Budget', $
FIELDNAME=NAT_YTDAMT, D12.0, TITLE='YTD,Actual', $
FIELDNAME=NAT_YTDBUD, D12.0, TITLE='YTD,Budget', $

Defining an FML Hierarchy in the Synonym Editor

As an alternative to manually coding the hierarchy attributes in a text editor, you can define them using the Synonym Editor. The controlling attributes are PROPERTY and REFERENCE in the Synonym Editor.
Reference: Synonym Editor: Column and Field Attributes

The Synonym Editor opens to the Tree View tab, which shows a hierarchy of segments and columns on the left, with the attributes and values of the selected field on the right.

Columns in a synonym can have the following attributes:

**General**

**FIELDNAME**

Is the name of the column.

**ALIAS**

Assigns an alternative name for a column, or the real column name for a DBMS synonym.

**MISSING**

Controls how null data is handled, that is, if no transaction value is supplied.

**TITLE**

Supplies a title to replace the column name that is normally used in reports and enables you to specify multiple language titles for the column or field.

**USAGE**

Describes the data type and format for the column for usage or display.
Note: Additional attributes, DEFCENT and YRTHRESH, are available if the Usage field is set to Date, Time, or DateTime (Timestamp) format. Use these attributes to enter the century and year threshold values for the column or field.

**Miscellaneous**

**DESCRIPTION**

Contains a description or comments about the column or field.

**ACCEPT**

Specifies criteria for validating data.

**OR**

Enables you to specify an acceptable value.

**FROM-TO**

Enables you to specify a range of acceptable value fields.

**FIND**

Enables you to supply file and field names to instruct WebFOCUS where to search for a data source and for a list of acceptable values. You supply the field name of the data field for which the validation criteria are being assigned, the file name of the target FOCUS data source where the field can be found, and the field name of the target data field that contains the validation criteria.

Note: Find is only available for FOCUS data sources and does not apply to OLAP-enabled Master Files. Note also that, in the Maintain environment, Find is not supported when developing a Master File.

**WITHIN**

Contains the name of a field to be included in a dimension.

**PROPERTY**

Sets the property for the field.

**REFERENCE**

Enables you to reference another data source.

**FIELDTYPE**

Identifies an indexed column.
**Note:** FIELDTYPE=R indicates a read-only column, which will not be updated by DataMigrator flows. This setting is useful for columns are that automatically assigned a value by the RDBMS, such as a Sybase or Microsoft SQL Server Identity or Timestamp column.

**ACCESS_PROPERTY**

Specifies access options for data in the column.

**INTERNAL**

Defines a column that does not appear in sample data or in the list of available columns. restricts the field from showing in any of the Field Lists in the reporting tools.

**NEED_VALUE**

Defines a column that requires a value to access the data.

**Select By**

Defines a column by value, range, or multivalues.

**AUTHRESP**

Defines a column that describes the result of an authentication operation. Correct response values must be provided in the ACCEPT attribute (using the OR predicate if more than one value is acceptable).

**AUTHTOKEN**

Defines a column that contains a response token to be passed as an input value to the operation to be executed.

**HELPMESSAGE**

Appends a help message to a column.

**Note:** The attributes available depend on the type of synonym.
<p>| <strong>ACROSS field</strong> | Horizontally sorts and groups data in a report by the values in the selected field. You can sort a report by more than one field. |
| <strong>BAR</strong> | A row type in Financial Report Painter. You can place an underline beneath a column of numbers in a financial report. |
| <strong>BY field</strong> | Vertically sorts and groups data in a report by the values in the selected field. You can sort a report by more than one field. |
| <strong>cell</strong> | A box formed by the intersection of a row and a column. This is an individual container where you enter information. |
| <strong>children</strong> | Values that are subordinate to a parent value in a hierarchy. |
| <strong>column</strong> | Vertical orientation of cells defined by a common category. |
| <strong>conditional formatting</strong> | The process where a condition is applied to one or more cells in a financial report. |
| <strong>DATA</strong> | A row type in Financial Report Painter. You can supply data (such as exchange rates or inflation rates) directly to a row for calculation purposes in a financial report. |
| <strong>data source</strong> | See Master File. |
| <strong>data value</strong> | A value that can either be supplied by a Master File or entered directly into a financial report through the DATA dialog box. |
| <strong>Design matrix</strong> | The environment where you create financial reports. |
| <strong>DETAIL field</strong> | Displays data in a report without sorting or grouping the values. |
| <strong>hierarchy</strong> | The parent/child relationship between fields. The relationship is defined in the Master File. |
| <strong>HOLD file</strong> | The output of a report request stored in a file that you can use as input to another WebFOCUS procedure. |
| <strong>label</strong> | The identification for a row. You can replace the default row identification label with an explicit row label that you want to reference in an expression. |
| <strong>Master File</strong> | Describes the data source from where you are reporting. The Master File is a map of the data source and all of the fields. By looking at the Master File, you can determine what fields are in the data source, what they are named, and how they are formatted. You can also determine how the fields in the data source relate to each other. |</p>
<table>
<thead>
<tr>
<th><strong>Matrix</strong></th>
<th>The tab located at the bottom of the Financial Report Painter. You can access your report information and build a financial report in a spreadsheet environment.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OR</strong></td>
<td>Sums the value of two or more tags in a single report row in FML. Used in the FOR phrase.</td>
</tr>
<tr>
<td><strong>parent</strong></td>
<td>A value that supersedes other values attributed to it in a hierarchy.</td>
</tr>
<tr>
<td><strong>PICKUP</strong></td>
<td>A row type in Financial Report Painter. You can retrieve posted rows from a work file and use them as if they were provided in a TAG row.</td>
</tr>
<tr>
<td><strong>RECAP</strong></td>
<td>A row type in Financial Report Painter. You can place an expression composed of numbers and operators (for example, alphanumeric, Boolean, and conditional) in a financial report.</td>
</tr>
<tr>
<td><strong>row</strong></td>
<td>A horizontal orientation of cells defined by a common category.</td>
</tr>
<tr>
<td><strong>SUM field</strong></td>
<td>Provides a total of all values for the selected numeric field when the report is executed.</td>
</tr>
<tr>
<td><strong>TAG</strong></td>
<td>A row type in Financial Report Painter. You can insert parent and/or child values from a hierarchy, which are tagged (for example, 1100, 1200, 2100), into a financial report.</td>
</tr>
<tr>
<td><strong>TEXT</strong></td>
<td>A row type in Financial Report Painter. You can insert free text or a blank row into a financial report.</td>
</tr>
<tr>
<td><strong>title</strong></td>
<td>The definition of a row. Definitions can be supplied by a Master File or customized in Financial Report Painter.</td>
</tr>
</tbody>
</table>
Sums the value of a range of tags in a single report row in FML. Used in the FOR phrase.
Index

A
accessing Financial Report Painter 26
accessing project data 119
WebFOCUS Environments and 119
accounting conventions for financial data 153, 154
accounting data 17
ADD parameter 223
adding blank rows 210
adding calculated values in financial reports 113, 127, 157, 213
adding columns to financial reports 113, 212, 213
adding cost of goods sold 170
adding expenses 174
adding financial data to reports 192
adding gross margin 170
adding profit 174
adding rows to financial reports 181, 184
adding tag rows from a hierarchy 112
adding tag rows to financial reports 39, 40, 42, 43, 49, 162, 181, 182, 219
adding text rows to financial reports 82, 162, 210
adding underlines to financial reports 88, 89, 162, 165, 235
addressing columns 202
aggregating children 58
aggregating financial data 18
archiving financial reports 20
AS CAPTION phrase 219, 220
assigning field formats 78
attributes 234

B
BAR
command 235
dialog box 88, 90, 165
rows 88, 235
blank rows 82, 162, 166, 167, 210
adding 82, 162, 166, 167
BORDER attribute 234
borders 90, 97, 104, 246, 248, 249, 250
adding for cells 246, 250
adding for rows 248, 249
budgets 15
bursting reports 20
BY fields 74, 124
BY phrase 190, 191
with financial reports 190, 191

C
calculated values 113, 127, 128, 157, 161, 212
adding in financial reports 113
changing titles 127, 161
formatting 128
calculating a value 113, 123
calculations 71, 74, 208
calculations and functions 208, 209
calling functions 208, 209
CAPTION parameter 52, 218, 223, 230, 231, 264
captions 51, 52, 218, 222, 230, 231, 264, 266
in a hierarchy 51, 52, 222, 264
in Master Files 218, 230, 231, 264, 266
indenting 51
specifying for hierarchy fields 264
adding borders 104

Creating Financial Reports 275
Index

cells (continued)
  identifying 71, 90
  notation 207
CENTGL.MAS (Chart of Accounts) 151, 266
CENTSTMT.MAS (Data Mart) 115, 122
  hierarchy in 122
CENTSTMT.MAS (Detail-level financial data) 151
CENTSYSF.MAS (General ledger accounts) 266
changing +/- signs using FMLINFO 157
changing column titles 123
changing row titles 29, 44, 70, 74, 87, 232, 233, 234, 260
  DATA rows 70
  PICKUP rows 29, 44, 74, 87, 260
  RECAP rows 74, 234
  TAG rows 44, 233
Chart of Accounts hierarchy 18, 21, 47, 68, 115, 156, 162, 216, 218, 266
  reports 21, 47
charts of accounts 219
choosing fields in Object Inspector 27, 120, 150
CNOTATION SET parameter 203, 204
column
  addresses 71, 202
COLUMN subtype 238
columns 27, 28, 71, 74, 83, 88, 90, 93, 97, 98, 101, 113, 114, 123, 128, 157, 181, 200, 202, 203, 204, 205, 237
  adding in Report Painter 27
  adding to financial reports 27, 113
  addresses 74, 202, 203
  deleting from financial reports 113, 114
  formatting 28, 88, 90, 93, 97, 98, 101, 128, 237
  hiding 83
  identifying 71, 74, 90
  in financial reports 181
  notation 203, 204
  numbers 200
  reference numbers 203
  values 205
  combining multiple values 184
combining records 184, 185
combining values 39, 40, 42, 43, 44, 184, 185, 186, 187
Computes tab 113, 123, 127, 157
conditional formatting 109, 128
Conditional Styling dialog box 109
consolidated financial data 58, 115, 166, 167
  adding 167
consolidating cost of goods sold 170
consolidating financial data 18, 44, 49, 57, 58, 63, 219, 222, 223, 225, 226
  in multiple rows 226
  multiple rows 63
  single row 58, 223, 225
constants 69, 192, 193
  in Financial Modeling Language (FML) 192
  in financial reports 192
  in FML (Financial Modeling Language) 192
  in FML requests 192, 193
contiguous columns 201
contiguous columns in financial reports 201
controlling column reference numbers 203
cost of goods sold 132, 170
  defining 170
creating a base report in Report Painter 27, 123, 156
creating a calculated value 113, 127, 157
creating a procedure 120, 150
creating financial reports 19, 26, 52, 58, 179, 184
  for hierarchies 52, 58
creating rows 181, 182, 184
  from multiple records 184
  in financial reports 181
creating tag rows 39, 182, 184

D

data 15, 39, 69, 84, 85, 115, 170, 181, 260
  adding as tag rows 39
data (continued)
adding to financial reports 69
financial 15
mart 115, 170
posting 84, 85
retrieval 85, 181, 260
values 39
DATA dialog box 69, 70
data source structure 122, 153
data sources 119, 122, 150, 151, 153
  joining 151, 153
data values
  retrieving 184
defining output formats 78
deleting columns from financial reports 114
deleting rows from financial reports 111, 112
delivering financial data 16
design matrix 28, 29
Design mode 36
Design/Preview icon 36
designing financial reports 28
Developer Studio
  creating a project 115
Dialogue Manager 178
differences between tabular/financial reports 131
displaying captions 44, 219
displaying children 49, 52, 162, 164, 219, 221, 222
displaying hierarchy values as captions 222
displaying parents and children 49, 52, 219, 221, 222
displaying total revenue 167
dragging tag values 39, 40, 162
drill down capability 109
dynamic reporting 47, 67, 216
accessing 26
adding rows 39, 111
adding tags 39, 40
adding text 82
adding underlines 88, 89
advantages for data access 23
advantages over SQL tools 22
creating Recap expressions 71
creating RECAP expressions 71
defining output format 71
Design matrix 156, 162
Design mode 36
editing rows 112, 113
formatting options 90
Preview mode 36, 37
Report Painter and 26
retrieving posted data 85, 87
saving intermediate results 84, 85
suppressing row display 83, 84
toolbar 28
viewing source code 35, 36
financial reporting 16, 17
government regulations for 17
adding columns to 113
adding data 39, 69, 181, 192
analytic 19
archiving 20
charts of accounts 21, 47, 49, 216, 219, 221, 222
consolidating data 18
constants in 69
creating 19, 21
designing 28
distributing 20
drilling down 109
external files 190
financial reports (continued)
for external use 17
for internal use 17
formatting 28, 39, 44, 69, 70, 83, 87, 88, 90, 234, 235, 237, 240
formatting options 39, 44, 69, 70, 83, 87, 234, 237
formatting options RECAP rows 71, 74
hierarchies 47, 52, 58, 219, 223, 230, 231, 232
HOLD files 261, 262
inserting text rows 82, 210
inserting variables in text rows 211, 212
inter-row calculations 71, 194, 195
international currencies 19
Master File for Chart of Accounts 68, 266
Master File for hierarchies 266
Master Files for 266
posting data in 84, 85
records in multiple rows 189
recursive models 215, 216
repeating rows 198
retrieving posted data 85, 87
running 37, 38
saving intermediate results 84, 85, 258
sorting 26, 27
sorting with BY 190
sorting with FOR 190
specifying ranges in 39, 42, 43, 49
supplying data as constants 192
suppressing rows in 83
types 21
FML (Financial Modeling Language) 26, 178, 179
and Dialogue Manager 178
FML hierarchies 47, 49, 52, 68, 216, 218, 219, 221, 222, 230, 231, 235, 253, 254, 263, 264, 266
coding in text editor 264
displaying 52, 216, 219
fields 264, 266
indenting captions 254
indenting row titles 253
indenting text or numbers 235
loading into memory 230, 231
Master Files for 68, 266
requirements 263
FML Painter 26, 27, 28, 29, 35, 36, 37, 39, 40, 42, 43, 49, 71, 78, 82, 83, 84, 85, 87, 88, 89, 112, 113
accessing 27
adding tags 39, 40, 42, 43, 49
adding text 82
adding underlines 88, 89
creating Recap expressions 71
creating RECAP expressions 71
defining output format 71, 78
Design mode 36
ingoing rows 112, 113
Preview mode 36, 37
retrieving posted data 85, 87
saving intermediate results 84, 85
suppressing row display 83, 84
toolbar 28
viewing source code 35, 36
FMLINFO function 157
calculated values and 157
FOCPOST files 84, 85, 258
FOCUS StyleSheets 234
For button 28
FOR field 26, 27, 28, 29, 39, 188
reusing values 29, 188
values panel 28, 29
FOR phrase 178, 179, 182, 188, 190, 191
reusing values 188
forecasting trends 19
Format dialog box 71, 78
formatting a calculated value 123
formatting cells 104, 108, 234, 237, 240
formatting cells and labels 240
formatting columns 90, 93, 234, 237, 241
formatting rows 97, 98, 101, 234, 237, 238, 239, 242
formatting rows and labels 237, 238, 239
formatting text 244
formatting text rows 243
FORMULTIPLE parameter 188, 189
free text 101, 210, 211, 237, 242, 243, 244
formatting 101, 237, 242, 243, 244
FREETEXT component 243, 244
functions 72, 208

G
GET CHILDREN parameter 52, 219, 223
GFA file 115
government mandated financial reports 17
government regulations for financial reporting 17
Graphical FOCUS Application (GFA) 115
gross margin 132, 170
groups of values 40, 42, 43, 187
identifying 187

H
headings 123, 156
hiding rows 83, 84, 256, 257, 258
hierarchies 49, 51, 52, 122, 153, 156, 162, 216, 223, 231, 232, 264, 266
Chart of Accounts 156, 162
displaying 52, 216
displaying values as captions 49, 52
field captions 264, 266
field properties 264, 266
indenting levels 51
reporting from 162
HOLD files for financial reports 261, 262

I
identifying cells 74, 207, 237
identifying columns 74, 90, 198, 199, 200, 201, 202, 203, 205, 237
by address 202
identifying columns (continued)
  by number 200
  by relative address 203
  by value 205
  contiguous columns 74, 201
  in financial reports 198, 199, 200
identifying data for a procedure 120, 150
identifying ranges of multiple values 186
identifying rows 74, 97, 195, 196, 197, 237, 238, 239
  in financial reports 196, 197
  in WebFOCUS StyleSheets 237, 238, 239
income statement 115, 132, 149, 156, 162
  creating 156, 162
  elements 132
indentation
  specifying between levels 253
indenting captions 222
indenting hierarchy levels 51
inserting rows in financial reports 111
inserting tag rows from a hierarchy 112
inserting text in financial reports 82, 210, 211
inserting variables in free text 211, 212
inter-row calculations 71, 193, 194
internal matrixes 203
international currencies in reports 19
invisible check box 83
invisible columns 83

J
joining data sources 151, 153

L
LABEL attribute 237, 238, 239, 240
LABEL subtype 238, 246
labels 29, 44, 70, 74, 87, 195, 196, 197, 243
  default 29
  explicit 29, 44, 70, 74, 87
  for rows 196, 197
  formatting rows 243
LOAD CHART command 230, 231
loading a hierarchy into memory 230, 231

M
making columns invisible 83
masking characters 43, 44, 187
  retrieving multiple values with 187
  retrieving values with 187
Master Files 68, 115, 122, 149, 150, 218, 230, 231, 263, 264, 266
  add to a project 150
  add to project 150
  CENTGLL.MAS 150
  CENTSTMT.MAS 115
  for financial reports 218
  for FML hierarchies 218, 230, 231
  for hierarchies 68, 218, 266
  hierarchies in 230, 231, 263
  specifying hierarchy fields 264, 266
  unconsolidated data and 149
  viewing as text 122
  viewing in graphical tool 122
matrix 29
Matrix button 28
Matrix tab 27, 28
multiple records 184
multiple values 39, 40, 42, 43, 44, 184
  combining 40
  identifying ranges of 42
  retrieving with masking characters 43, 44

N
negative values in financial data 153, 154
non-recursive models 215
NOPRINT command 256, 257

O
Object Inspector 27, 120, 123, 150, 151, 156
adding fields from 27, 123, 156
joining files from 151
OLAP analysis 19
OR operator 40, 185, 186

P
PAGE BREAK command 236
page breaks in financial reports 236
PARENT_OF attribute 264
PICKUP dialog box 84, 85, 87
positional labels 195, 196, 197
positive values in financial data 153, 154
POST command 259
posted data
retrieving 84, 85, 87, 260
posting data 84, 85, 258, 259
in financial reports 258
posting financial data 258
posting intermediate results to a file 84, 85
pre-consolidated data 170
Preview mode in Financial Report Painter 36, 37
Procedures folder 120, 150
profit 132, 170, 174
Project Wizard 115
projects 115, 119, 150
adding a Master File 150
data and 119
GFA file and 115
virtual folders 115
wizard 115

Properties bar 28
properties for hierarchy fields 264, 266

Q
Quick Style toolbar 31

R
range of records
combining 186
range of values 42, 186, 187
combining 186, 187
ranges 39, 42, 43, 49, 187
specifying in financial reports 42, 43, 49, 187
RECAP command 178, 179, 193, 194, 195
and FML reports 193
RECAP dialog box 71, 74
RECAP expressions 71, 193, 195
creating 193
creating with Financial Report Painter 71
RECAP operations 71, 80
RECAP rows 71, 81, 193, 245
formatting 245
summarizing 81
records 184, 188, 189
in multiple rows 188, 189
reusing 188, 189
recursive models 215, 216
recursive reports 21
REFERENCE attribute 264
relative column addresses 203
repeating rows 198
report bursting 20
report distribution 20
report fields in Financial Report Painter 28, 29
Report Options Computes tab 113
Index

Report Painter 26, 31, 120, 123, 150, 156
   changing column titles 123
   choosing 120, 150
   Financial Report Painter and 26
toolbars 31
ReportCaster 20
reporting against consolidated financial data 115
reporting against hierarchies 47, 49, 52, 58, 216,
   219, 221, 222, 223, 232
reporting against unconsolidated financial data 149
reporting solution 15
revenue 132, 162
row formatting 234, 237
row labels 70, 74, 87, 195, 197
   assigning to DATA rows 70
   assigning to PICKUP rows 87
   assigning to RECAP rows 74
   assigning to TAG rows 70, 74, 87
row labels in Financial Report Painter 28, 29
row titles 28, 29, 44, 70, 74, 87, 232, 233, 234,
   260
   DATA rows 70
   PICKUP dialog box 87
   PICKUP rows 87, 260
   RECAP rows 74, 234
   TAG rows 44, 233
row types in Financial Report Painter 28, 29
row-based reports 21
rows 28, 29, 71, 74, 83, 84, 88, 97, 98, 101, 111,
   112, 113, 182
   assigning titles to 29
   calculations and 71, 74
   deleting from financial reports 111, 112
   editing 112, 113
   formatting 28, 88, 97, 98, 101
   identifying 71, 74, 97
   retrieving values for 182
   suppressing display in financial reports 83, 84
rows and calculations 193, 194, 195
rows in financial reports 181
running financial reports 37, 38
S
saving intermediate report results 84, 85, 258
saving rows 84, 258, 259
scheduling report distribution 20
SET BLANKINDENT parameter 253
SET CNOTATION parameter 203, 204
SET FORMULTIPLE parameter 188, 189
SET parameters 254
setting field formats 127
setting formats for calculated values 157
sort fields 27, 190
sort sequence 190
sorting reports 27
sorting with BY 27
sorting with FOR 27
source code 35, 36
   viewing FML code 35, 36
Source tab 35, 36
storing project data 119
style sheets 234
StyleSheet attributes 234
styling
   specifying conditions for 128
styling a calculated values 128
styling a column conditionally 123
subroutines 72, 208
subtypes 238
summary RECAP rows 81
supplying data directly in FML 69, 192, 193
suppressing display in financial reports 257, 258
suppressing rows 83, 256, 257, 258
suppressing rows in financial reports 256, 257, 258

tabular reports 131
TAG dialog box 39, 44, 162, 164
Tag icon 39
TAG rows 40, 42, 43, 44, 49, 112, 256, 257, 258
adding 40, 42, 43, 44, 49
adding from a hierarchy 112
suppressing display in financial reports 256, 257, 258
tag values 29, 39, 182, 188, 189, 223
reusing 29, 182, 188, 189, 223
TEXT dialog box 82, 83, 162
text rows 82, 101, 210, 211, 242
formatting 101, 242
titles in FML Painter 28, 29
TO phrase 42, 186, 187
toolbars 31
Report Painter 31
total revenue 167

unconsolidated data 149
underlines in financial reports 89, 235
underlining calculations 165
using unconsolidated data 149

values 29, 39, 40, 42, 43, 44, 185, 188, 189, 205
combining 40, 42, 43, 44
combining range of 42
for columns 205
identifying groups of 40
in multiple rows 188, 189
retrieving for rows 39
retrieving with masking characters 43
reusing 29, 189

virtual folders 115

WebFOCUS Environments 119
WebFOCUS Financial Reporting Platform 18
When Exists check box 84
WHEN EXISTS phrase 257, 258
WITH CHILDREN parameter 219
Customer Connections

Where users partner with Information Builders to provide feedback and create quality documentation.

Partner With Us

Information Builders Technical Content Management team is comprised of many talented individuals who work together to design and deliver quality technical documentation products. Your feedback supports our ongoing efforts!

You can also join a usability study to get an early look at new documentation products and services. Your participation helps us create documentation products that meet your needs.

To send us feedback or make a customer connection, contact Michele Brady, Service Director, Technical Content Management at Michele_Brady@ibi.com.

Request Permission to Use Copyright Material

To request permission to repurpose copyrighted material, please contact Frances Gambino, Vice President, Technical Content Management at Frances_Gambino@ibi.com.

Visit the Information Builders Technical Content Library

To visit the Technical Content Library, go to http://documentation.informationbuilders.com/.