QAD Enterprise Applications 2009
Enterprise Edition

User Guide

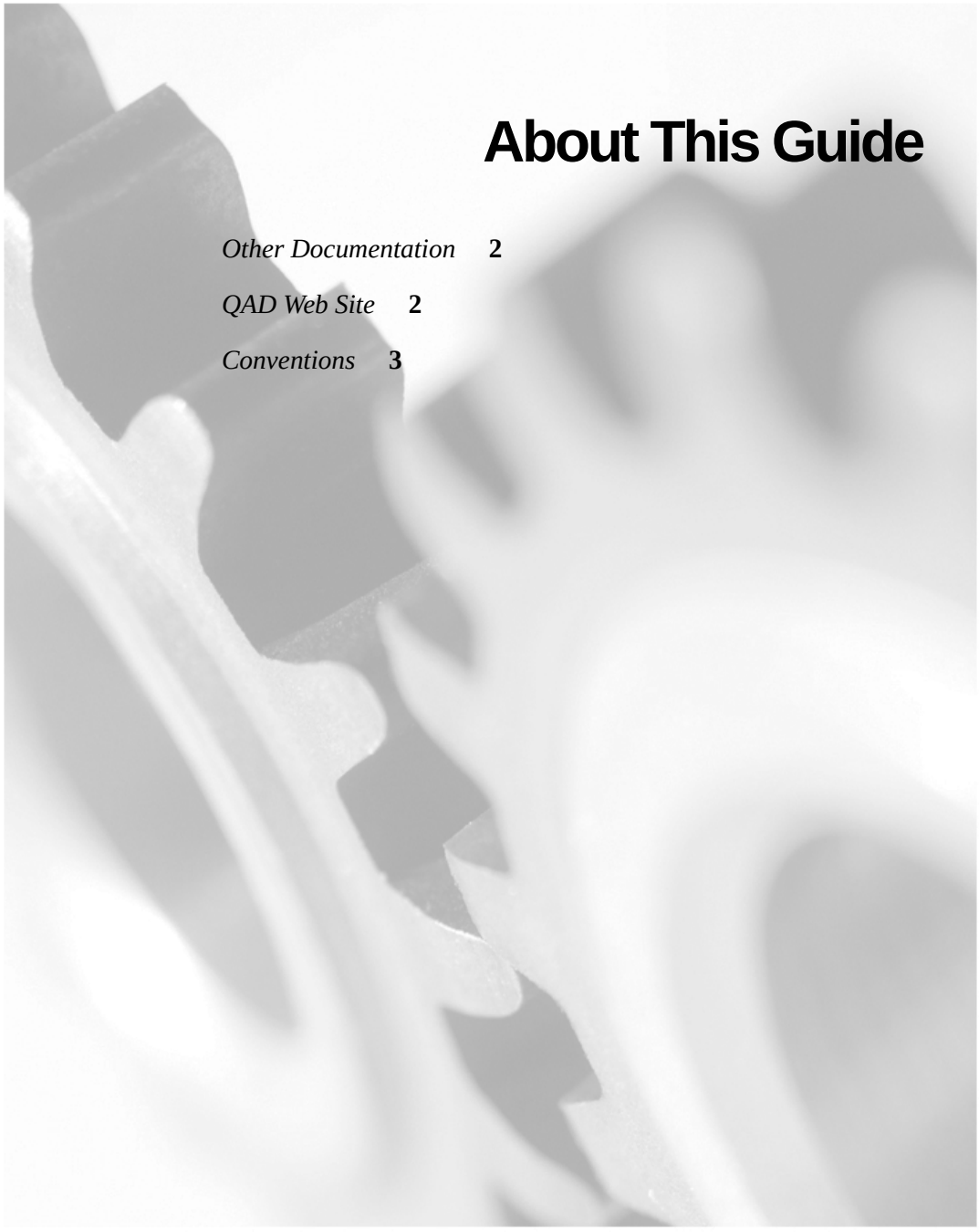
QAD Reporting Framework

Overview
Report Design
Report Administration
Using Reports

78-0799A
QAD 2009.1 Enterprise Edition
September 2009

QAD Inc.
100 Innovation Place
Santa Barbara, California 93108
Phone (805) 684-6614
Fax (805) 684-1890
http://www.qad.com

Contents



About This Guide

This guide is divided into three sections:

- The Overview section provides a general introduction to the key features of the QAD Reporting Framework and discusses the fundamental concepts of report design.
- The Report Design section covers the development and design of reports using the QAD Reporting Framework and is intended for the report developer/designer.
- The Report Administration section covers the setup and administration of the QAD Reporting Framework functions and is intended for the report administrator.

Other Documentation

- For QAD Enterprise Applications software installation instructions, refer to the appropriate installation guide for your system.
- For information on using QAD Enterprise Applications, refer to the *User Guides*.

QAD Web Site

QAD's Web site provides a wide variety of information about the company and its products. You can access the Web site at:

http://www.qad.com

For users with a QAD Web account, product documentation is available for viewing or downloading from the QAD Online Support Center at:

http://support.qad.com/

You can register for a QAD Web account at the QAD Online Support Center. Your customer ID number is required. Access to certain areas is dependent on the type of agreement you have with QAD.

Most user documentation is available in two formats:

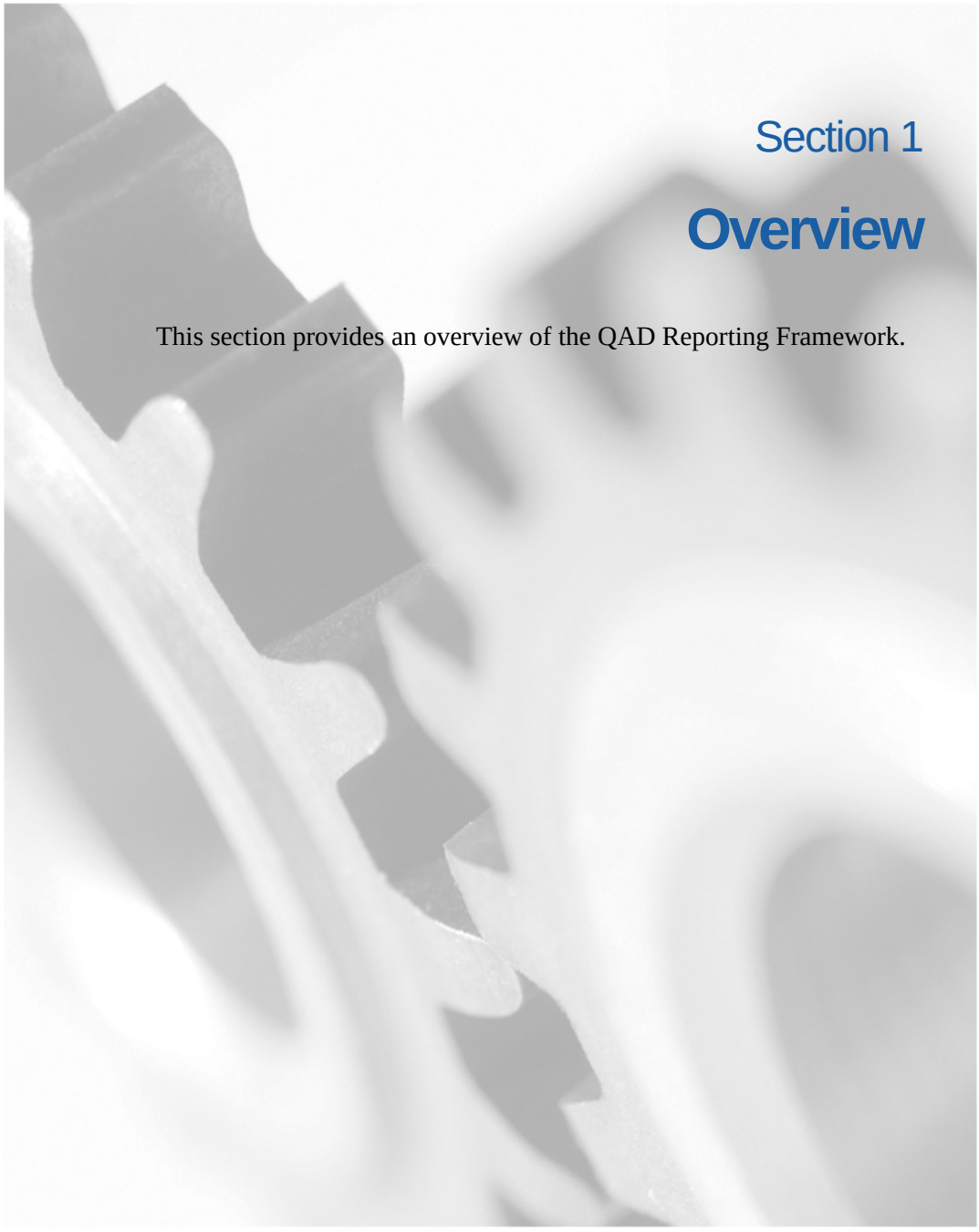
- Portable document format (PDF). PDF files can be downloaded from the QAD Web site to your computer. You can view them with the free Adobe Acrobat Reader.
- HTML. You can view user documentation through your Web browser. The documents include search tools for easily locating topics of interest.

Features also include an online solution database to help QAD Configurator users answer questions about setting up and using the product. Additionally, the QAD Web site has information about training classes and other services that can help you learn about QAD Configurator.

Conventions

This document uses the text or typographic conventions listed in the following table.

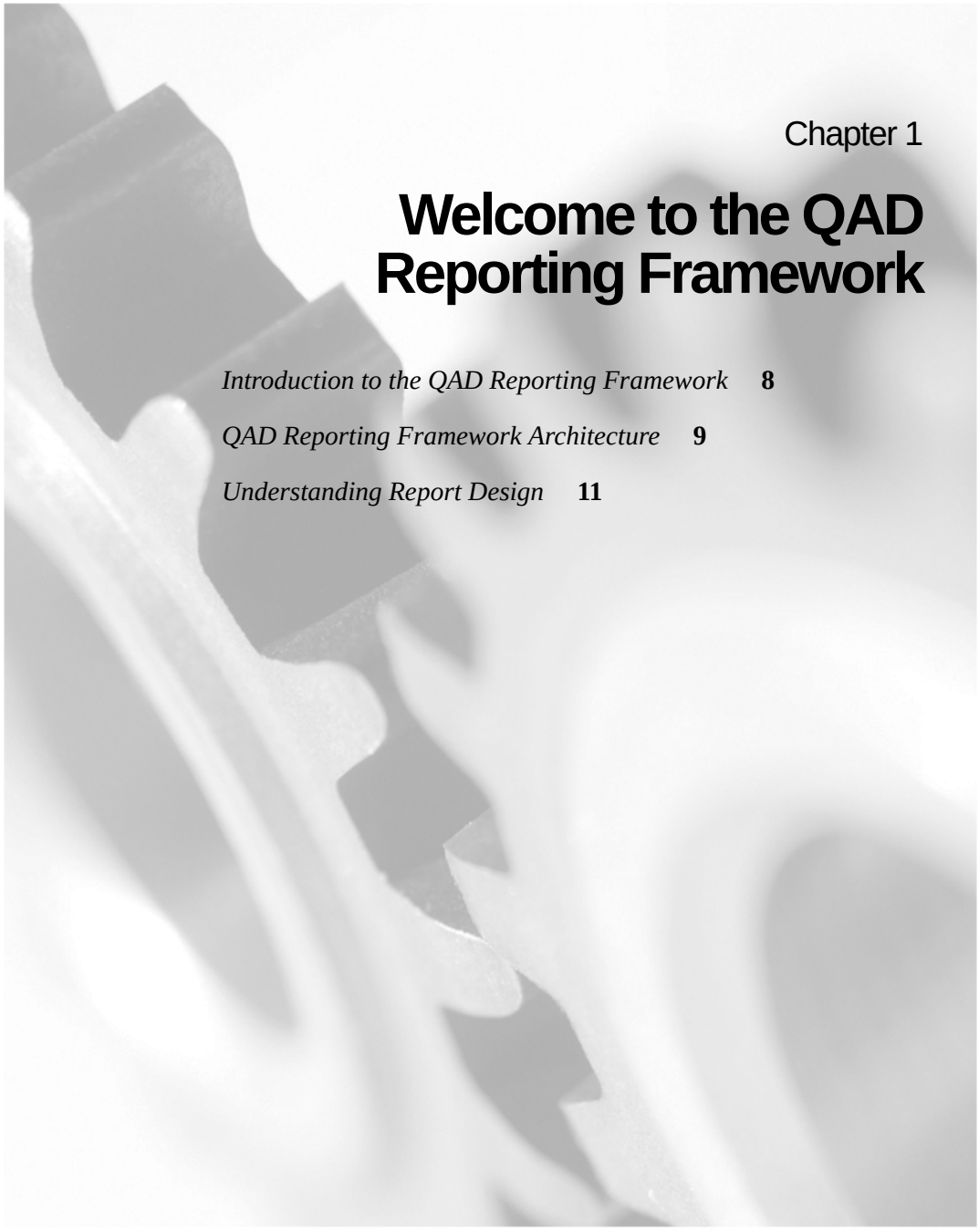
If you see:	It means:
`monospaced text`	A command or file name.
`italicized monospaced text`	A variable name for a value you enter as part of an operating system command; for example, *`YourCDROMDir`*.
`indented command line`	A long command that you enter as one line, although it appears in the text as two lines.
Note	Alerts the reader to exceptions or special conditions.
Important	Alerts the reader to critical information.
Warning	Used in situations where you can overwrite or corrupt data, unless you follow the instructions.



Section 1

Overview

This section provides an overview of the QAD Reporting Framework.



Chapter 1

Welcome to the QAD Reporting Framework

Introduction to the QAD Reporting Framework

Reports help you analyze and interpret important information. The QAD Reporting Framework makes it easy to create simple reports, and it also has the comprehensive tools you need to produce complex or specialized reports.

Multiple Data Sources

The QAD Reporting Framework is designed to produce the report you want from a range of data sources. You can extract data from Progress databases, browses, or through the QAD Financials API for reporting purposes.

Powerful and Flexible Report Authoring

The QAD Reporting Framework offers you both simplicity and flexibility in creating your reports.

The built-in Report Wizard guides you step by step through building basic reports and completing common reporting tasks. A rich set of report design tools lets you create more complex reports tailored to your specific requirements.

Columns, groups, calculated fields, subreports, and formatting help make sense of data and uncover important relationships that might otherwise be hidden.

Multiple Output Formats

The flexibility of the QAD Reporting Framework does not end with creating reports. Your reports can be published to a variety of outputs including printer, PDF, and Excel files.

Report Scheduling

You can schedule the system to run your reports automatically at a certain time or at a specified interval and send scheduled report outputs to your desired destination, such as a printer or the document service on the report server. You can also have the system notify you that your scheduled reports have run.

QAD Reporting Framework Architecture

The QAD Reporting Framework contains five key components: report render engine, report layout definition, data source, pre-render engine, and report. The following diagram illustrates the QAD Reporting Framework architecture at a high level.

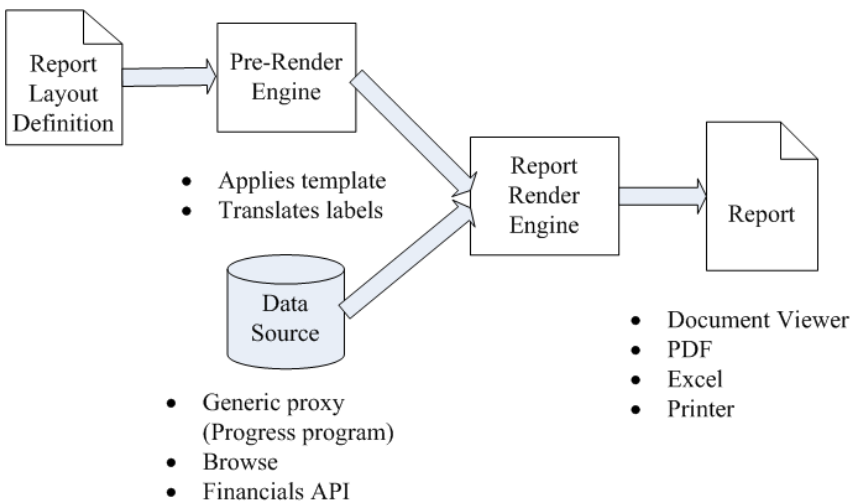


Fig. 1.1 QAD Reporting Framework Architecture

- Data source

  Data to be displayed on the report are queried from the underlying business system through the data source definition. The QAD Reporting Framework supports three types of data sources: generic proxy (Progress program), browse, and Financials API. For detailed information on the three types of data sources, see “Choosing a Report Data Source” on page 22.

- Report layout definition

Report layout definition defines what gets displayed on the report, and where. You use Report Designer in the .NET UI to define the report layout in WYSIWYG (What You See Is What You Get) fashion. You can also import and export report layout definitions as XML files, which makes it very easy for you to deploy reports or migrate them between systems, such as moving reports from the test environment to the production environment.

- Pre-render engine

  The pre-render engine pre-processes the report layout definition by applying a report template to it as well as performing label translations and produces a modified report layout definition. The resultant report layout definition along with the data source are then fed into the report render engine, which generates the actual report.

- Report render engine

  As the core of the solution, the report render engine takes in data set and report layout definition as inputs, then renders and produces the report output. Since the QAD Reporting Framework is a .NET solution, the report render engine can only run on the Windows operating system.

  The rendering process takes place on the computer that actually runs the report. If you run a report in the QAD .NET UI on your PC, then your PC's CPU power is consumed to render the report, which helps to distribute the processing load across client machines.

- Report

  The report output can be rendered in three different formats, depending on your preference. The default format is a document displayed on the screen, which you can view by paging and zooming. You can then send it to printer if you want a hard copy. Alternately, the report can be exported into the PDF or Excel format, which you can print or save as a file.

QAD Report Server Architecture

Reports can also be run on the report server. In this operational mode, the report render engine is run as part of a batch process that runs on the server. Reports in a batch are generated periodically during each batch

run, such as in a daily, weekly, or monthly batch. Typically, you use the Windows Task Scheduler on the server to periodically fire off the running of a batch through a command line.

Any report run from the server can be sent to the printer of your choice, or an output file, typically a PDF file, that is stored on the QAD .NET UI web server, which can be accessed from a URL. Optionally, you can have the system send you an e-mail notification every time a report is run.

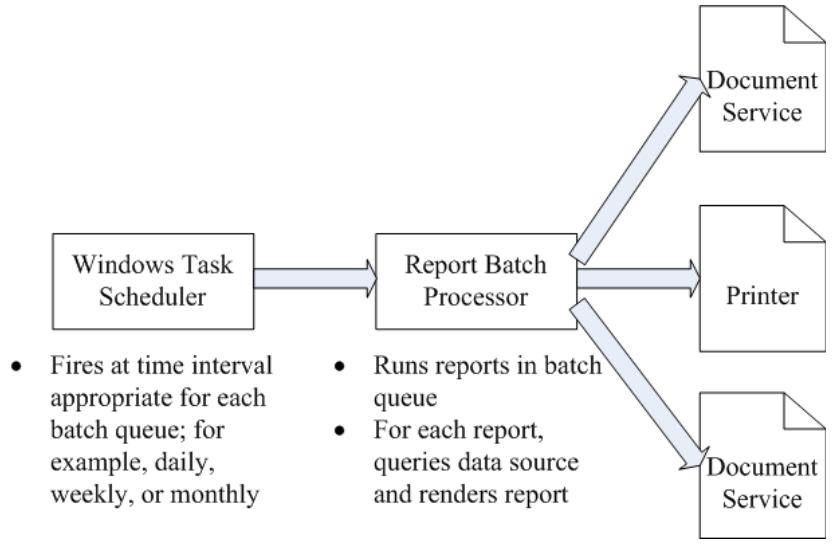


Fig. 1.2
QAD Reporting Framework Architecture

You can have any number of Windows report servers that can process report batches. If you have a batch with a lot of reports in it that runs frequently, you can have several servers pointing to the batch, which balances the processing load as well as serves as a fail-safe redundancy.

Understanding Report Design

It is advisable to take a structured approach to preparing a report. This approach includes the following elements:

- Deciding on the content of the report
- Developing a prototype on paper

This section is designed to provide a conceptual understanding of the reporting process.

Deciding on the Content of the Report

Before you do anything else, you should outline the information you want the report to provide. The following sections provide a guide to making that outline.

Stating the Purpose

What is the overall purpose of the report?

Reports are management tools. Their purpose is to help you quickly grasp the essential elements and relationships found in raw data, to help you make effective decisions. For a report to be effective, it has to present the correct data in a logical way. If it presents the wrong data, or if it presents the right data in a haphazard manner, the report may slow the decision-making process or may even lead to incorrect decisions.

A good starting place in the development of a report is to write out the purpose of the report in a sentence or two. The purpose statement helps you focus on your primary needs, and it gives the report both a starting point and a goal. Here are some examples of purpose statements:

- "The purpose of this report is to show monthly and year-to-date sales by sales representatives, compare this year's numbers to last year's, and flag representatives whose sales figures do not meet company standards."
- "The purpose of this report is to show sales activity for each item in inventory, and to suggest reorder quantities based on that activity."

Defining the purpose of the report before you start is a critical step in the overall process.

Who is going to read the report?

A single report is often used by many individuals. A detailed, company-wide sales report, for example, may be used by sales representatives, the regional sales manager, the national sales manager, and the Chief Operating Officer (COO).

These individuals will be interested in different aspects of the report:

- A sales representative will use the report to evaluate individual sales performance and compare this performance to that of other representatives in the region.
- The regional sales manager will use the report to evaluate regional representatives and compare the region's performance to that of other regions.
- The national sales manager will use the report to evaluate the performance of regional managers and compare overall sales to the current sales forecasts.
- The COO will use the report to evaluate the performance of the Vice President of Marketing and the sales department as a whole, and to project such things as manufacturing needs and warehouse locations.

Since each user of the report has different interests, it is important to plan the report so it includes the information each user is looking for.

Determining the Layout of the Report

What is the report title going to be?

Write out a working title for the report. You may decide to change it later, but at least you will have a title to use when creating the prototype report.

What identifying information is needed in the header and footer?

You may wish to include the print date, information on who prepared the report, a block of text to describe the purpose of the report, the range of data covered, or something similar. If you are going to include such information, write it down so you can use it in preparing your prototype. The information can come from a variety of sources, depending on the kind of information you plan to use.

- Information on who prepared the report might be drawn from individual data fields in the database tables used. If it is to be drawn from a database table, what table? Or, what combination of tables?
- A block of text can be created as a text object and placed anywhere on the report.

- The QAD Reporting Framework can generate information such as the print date or page numbers.

Finding the data

What data do you want to use in the report?

Do you know the type of database you are reporting from? Will you be reporting off a browse or a database table? Are you familiar enough with the data to find the necessary information? When looking for a Customer ship-to address, can the field be found in a database table? If not, seek help from someone who is familiar with the system database.

What specific data should appear in the body of the report?

The body should contain all the data needed to fulfill the statement of purpose you wrote for the report. It should also contain all of the data needed by the various users that you have identified.

This step requires you to look at the available database tables. The QAD Reporting Framework allows you to combine data from different databases when you create reports, so you have a great deal of flexibility in your work.

- Much of the data in a typical report is taken directly from data fields. Which data fields will be used, and where are they located?
- Other data will be calculated based on data fields. Which data fields will be used in the calculations?
- Still other data will be placed directly into the report using text objects (headings, notes, labels, and so on).

Does the data exist or does it need to be calculated?

Some report information can be drawn directly from data fields (sales information, for example); other information will have to be calculated based on data field values (for example, sales commission, based on the relationship of sales to quota). In your planning, it can be helpful to segregate or flag data that needs to be calculated from data that can be used directly.

What types of fields contain data?

You should take the time to get to know the data type for data fields that will be used in your calculations. Since formula functions and operators work with specific kinds of data, it is important to recognize the data type you are working with, before you start any calculations. For example, some functions require numeric data, while others work with only string fields.

Developing a Prototype on Paper

While a paper prototype is useful regardless of your level of expertise with the QAD Reporting Framework, it is particularly valuable when you are first learning the system. With the paper prototype in hand, you can put your full effort into learning and using the functions, rather than into trying to design and learn at the same time.

To design a paper prototype:

1. Get the same size paper you will be using for the finished report.
2. Position the title and other descriptive header information, using boxes or lines to represent report elements.
3. Position the footer information.
4. Review the page layout for balance.
5. Look at the information you intend to include in the body of the report:
   - Count the number of fields being used and estimate the appropriate spacing between fields.
   - Use rectangles to pencil in the fields within the estimated spacing.
   - Change the spacing if you need to.
   - Decide on a logical sequence for presenting the data in the body of the report.
   - Label the fields to indicate that sequence.
6. Use small boxes to indicate group values and totals.

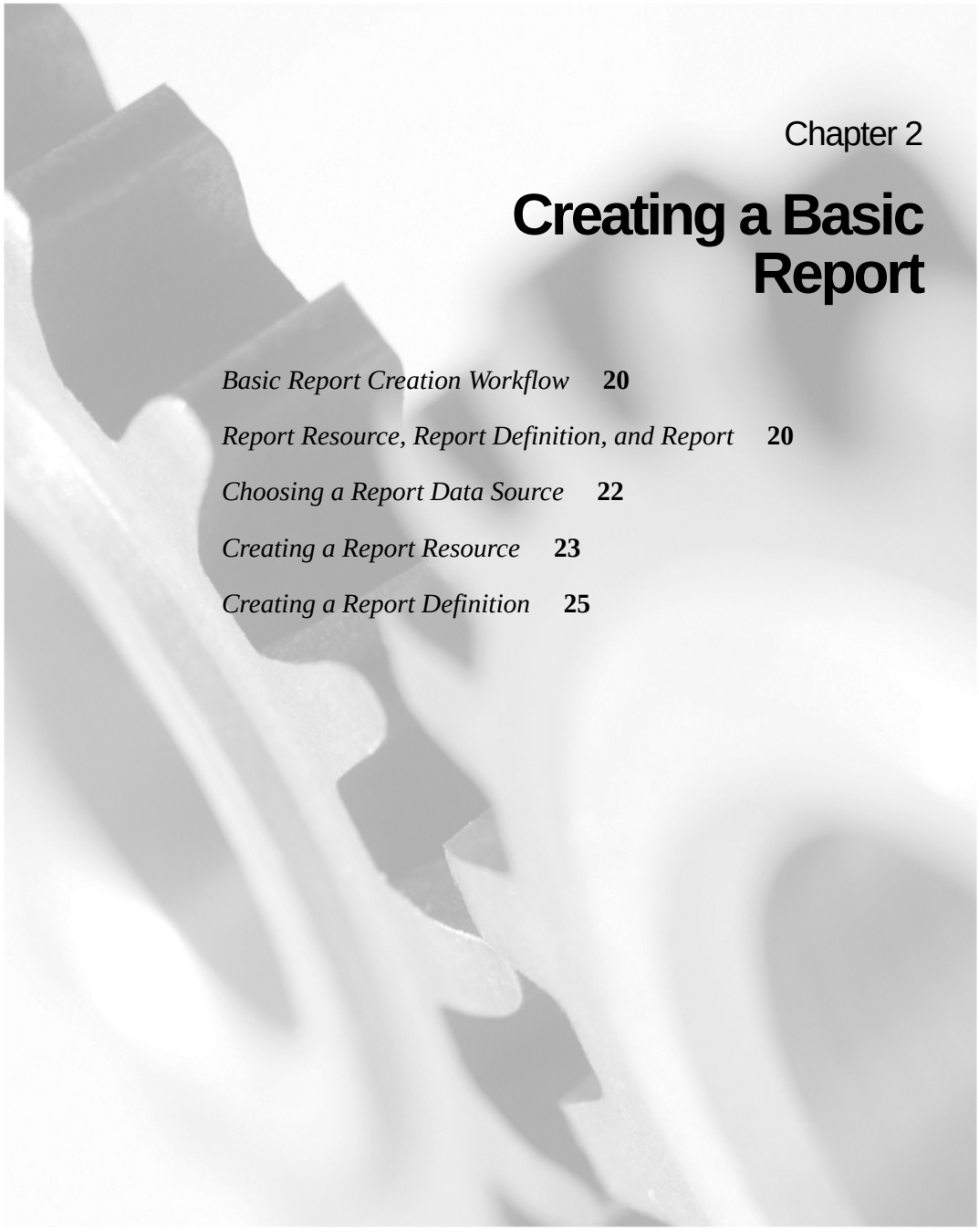
7 Darken any elements you want highlighted to make them stand out from the rest of the prototype.

8 Review the finished product for layout and balance, and make changes as needed.

Section 2

Report Design

This section covers development and design of reports using the QAD Reporting Framework and is intended for use by report developers/ designers.



Chapter 2

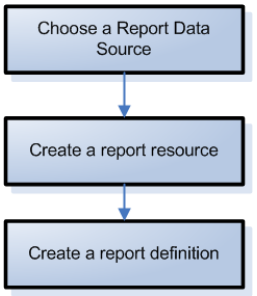
Creating a Basic Report

Basic Report Creation Workflow

The QAD Reporting Framework makes it very easy and straightforward to build a basic report. It takes just a matter of minutes to generate your first basic report. Report Wizard provides you with step-by-step instructions to build a basic report. Built-in report templates take care of most of the reporting layout and formatting for you.

Creating a basic report consists of three main steps.

Fig. 2.1
Basic Report Creation Workflow



Report Resource, Report Definition, and Report

Report resource, reporting definition, and report are a set of interrelated concepts in the QAD Reporting Framework.

A **report resource** represents a unique, cross-domain report object that contains report metadata, report definitions, report data source definitions, filter definitions, report parameters, and report settings.

A **report definition** contains all the information that defines the data binding, layout, and customized formatting of a report. It is saved as an XML file that can be edited in Report Designer, either visually or in the text editor mode.

For a report resource, you can create multiple report definitions that represent different layout and formats to address different requirements. For example, you can design a cash flow statement report definition to meet external financial requirements, and another with a different layout and formatting for internal reporting.

A **report** is a collection of your desired data, as defined in the report resource, organized in your desired format, as defined in the report definition. This is what the QAD Reporting Framework is all about.

To sum up, report resources are primarily concerned with what data to display and report definitions are mainly about how to display them in the generated reports. And since you may want to retrieve different data from the same data source and present it in different formats, you can create multiple report definitions within the same report resource to generate different reports.

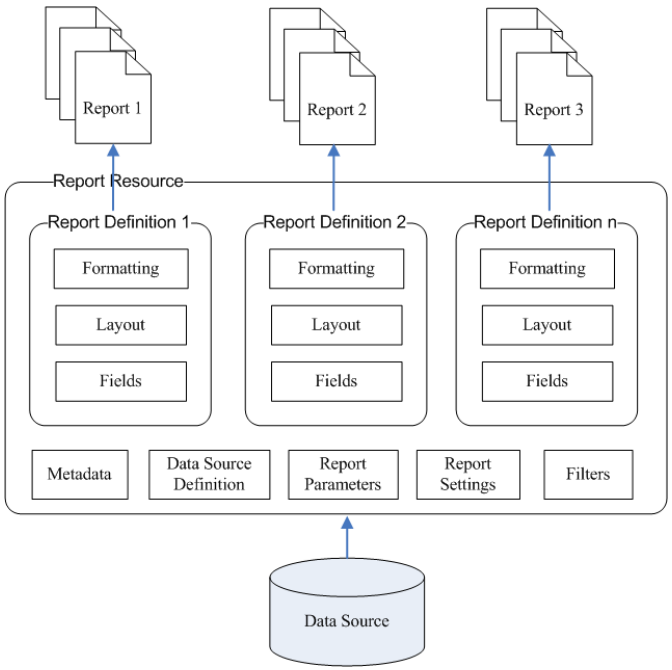


Fig. 2.2
Report Resource, Report Definition, and Report

Example You need to generate sales order by customer reports and unconfirmed sales order reports on a weekly basis. You create a report resource that retrieves data from the sales order-related tables and then create two report definitions for the two kinds of reports. You then either run these reports manually every week or schedule a weekly batch run for them to generate the reports you need.

Choosing a Report Data Source

Before you create a report, you need to determine where your report data comes from. The QAD Reporting Framework supports three types of data sources: generic proxy (Progress program), browse, and QAD Financials API. Depending on which type of data source you use, you may need to perform some additional implementation steps.

Generic Proxy (Progress Program) Data Source

Generic proxy is a built-in data source provider that calls a Progress program through the QAD .NET UI to get the required data. The Progress program implements all the logic of data queries, calculations, or even database updates when the report is run. This is the most powerful and flexible type of data source in that you can do anything that the Progress language could do to manipulate data in the report. However, it does require you to have Progress programming experience and knowledge of the underlying database schemas in order to write such Progress programs.

Generic data source implementation entails the following general steps:

1. Develop a generic proxy `.p` program. To develop proxy programs for the QAD Reporting Framework, you are required to be familiar with both Progress programming and QAD ERP database schema.

2. Deploy the generic proxy program into the QAD .NET UI AppServer tier in the following directory under the webapp root location:

   `<Web App Root>/WEB-INF/pro/com/qad/shell/report/reports`

   Note The proxy layer is generic and can call any data source `.p` program you deploy.

3. To improve performance, compile the program into a `.r` file by running `mkdt compile` from the `<Web App Root>/WEB-INF/pro/com/` directory.

For detailed information on implementing generic proxies, see “Implementing the Generic Proxy” on page 101.

Browse Data Source

The QAD Reporting Framework supports both classic QAD ERP (MFG/PRO) browses and Financials browses as data sources.

In QAD Enterprise Applications, browses display selected data in the form of a table. Column headings are field labels; rows are field values. The field values in a browse come from any table in the QAD Enterprise Applications (MFG/PRO) schema. A browse includes selected values from one table or several joined tables.

You can use an existing browse as the data source for your report by associating the browse ID with the report resource. Then, you can have all the fields in the browse at your disposal to design your report definition.

Classic QAD ERP (MFG/PRO) browses are maintained using Browse Maintenance (36.20.13) and View Maintenance (36.20.18). For information about defining browses, see *User Guide: QAD .NET User Interface*.

Financials browses are maintained using the Financials CBF tool.

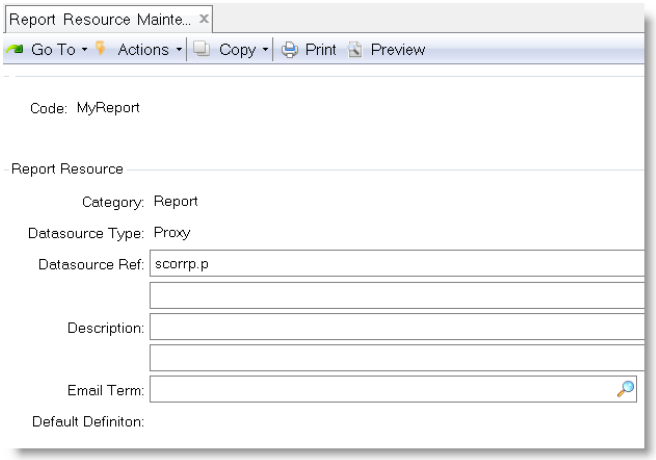
QAD Financials API Data Source

Any query defined in the QAD Financials API can be used as a report data source.

Creating a Report Resource

Use Report Resource Maintenance to create a report resource.

Fig. 2.3
Report Resource Maintenance



Enter the following fields. Click Next or press Enter to move to the next frame or field; click Back to return to the previous field.

Code. Specify a code that identifies a report resource.

Important QAD-provided built-in reports, report resources and templates all begin with “QAD_”. Do not create or modify reports, report resources, or templates with this prefix. Otherwise, your customized changes will get overwritten during system upgrades from QAD.

Category. Select one of the following report resource types for different report providers: Report for QAD .NET-based reporting and Dashboard for Cognos dashboard reports.

Note Cognos dashboard reports are currently not implemented.

Data Source Type. Specify a data source type that indicates how the report retrieves its data:

- Browse: The report uses browses as its data source. Both classic browses (created in Browse Maintenance) and Financials browses (created in the Financials CBF tool) are supported.
- Proxy: The report accesses the database through the generic proxy program.
- Financials API: The report retrieves data through the QAD Financials API.

Data Source Ref. Provide the reference information for retrieving data through the data source provider.

- For classic QAD ERP browses maintained in Browse Maintenance, enter `<BrowseServerType>:<QueryID>`; for example, `QAD.Browse.MfgProBrowseServer:so009`. `<BrowseServerType>` is optional. If it is not specified, the data source reference defaults to the QAD ERP browse server. In this case, `so009` is equivalent to `QAD.Browse.MfgProBrowseServer:so009`.

  For Financials browses created in the Financials CBF tool, enter the following:

  `BaseAdapters.CBAdapters.QadBrowseAdapter:<BusinessComponent>.<QueryMethod>`

  Here is an example using this naming syntax:

  `BaseAdapters.CBAdapters.QadBrowseAdapter:BJournalEntry.SelectPosting`
- For the generic proxy data source, specify a data source proxy program file name; for example, `myReport.p`.
- For the Financials API data source, specify a Financials reporting component name followed by a method name; for example, BGLReport.GLList, where BGLReport is a component name and GLList is a method name.

Description. Provide a description of the report resource.

E-Mail Term. Specify the default mail subject title to be used in scheduled report notification mails.

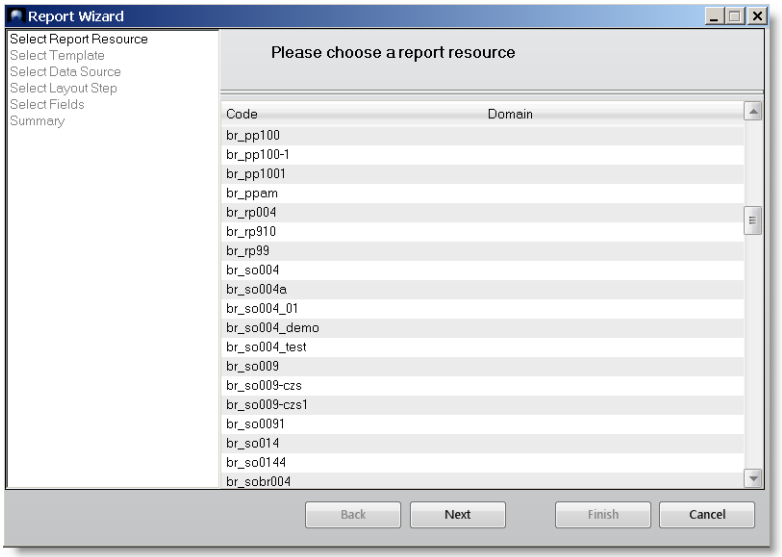
Default Definition. Specify the default report definition for the report resource. When you open a report resource in Report Viewer or Report Designer, this report definition is loaded by default.

Creating a Report Definition

1. Type Report Resource Designer in the menu search field and press Enter.
2. Click the New icon on the Report Designer Toolbar. The Report Wizard window appears.

3 Select the report resource you previously created and click Next.

Fig. 2.4
Report Wizard: Select a Report Resource



4 Select a report template or select None to use the default built-in report template. Click Next. For detailed information about report templates, see “About Report Templates” on page 65.

5 Select a table as the report data source and click Next.

- All the available tables you can select as data sources are listed in a tree.
- To view all the fields in a table, click the plus sign next to the table to expand the tree.

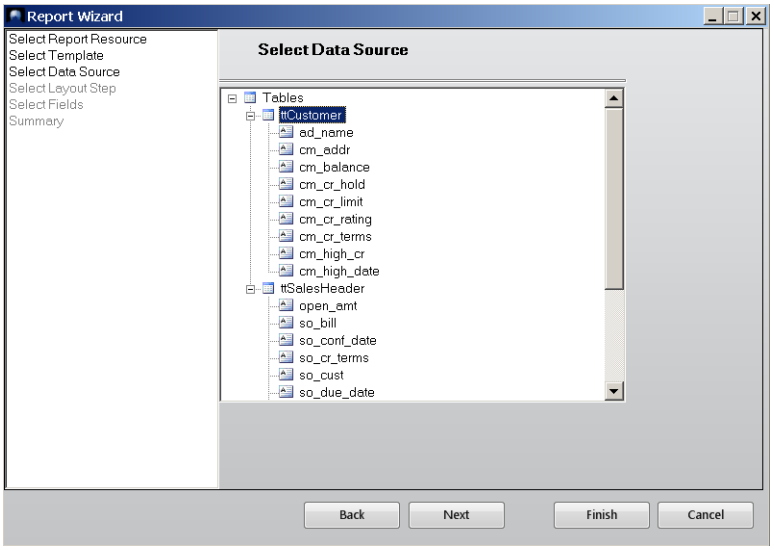


Fig. 2.5
Report Wizard: Select Data Source

6 This screen offers you several options to define how the data will be organized on the page. Select the layout that best approximates what you want the final report to look like.

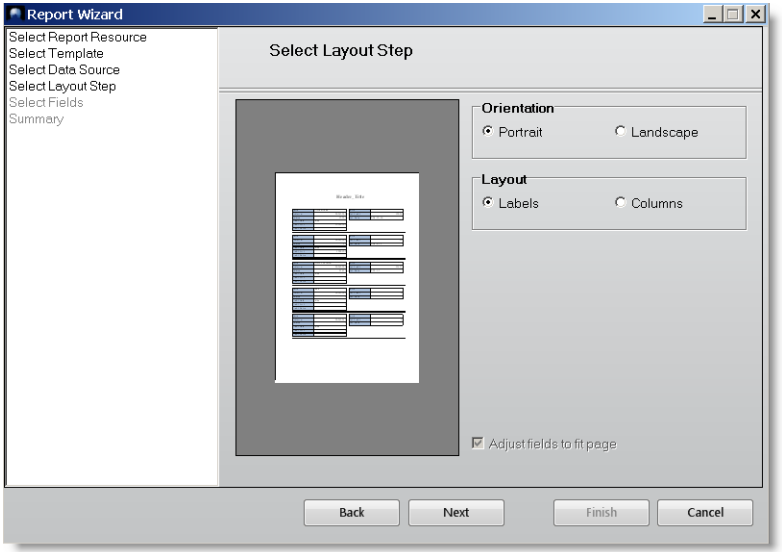


Fig. 2.6
Report Wizard: Select Layout

Orientation. Choose whether to design and render the report in Portrait or Landscape mode.

Layout. Specify how you want the field names and fields organized in the report.

- Labels: For each field in the report, the corresponding field name is placed to the left as a label.
- Columns: Field data is organized in a column or multiple columns with field names as column headers.

Adjust fields to fit page. Select this option to adjust fields to fit the page width; otherwise, clear this option.

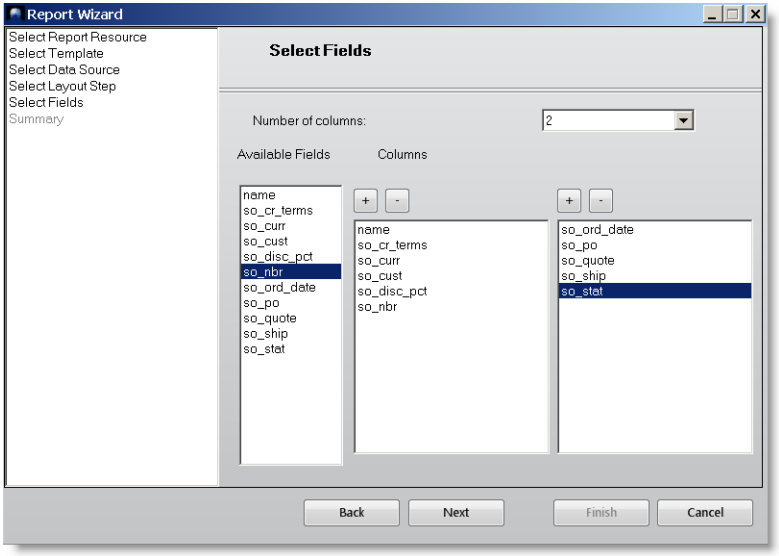
Click Next to continue.

7 Select Fields into the report.

This screen displays differently depending on whether you chose the label-style or column-style layout in the previous step.

- This screen displays when you chose the label-style layout in the previous step.

Fig. 2.7
Report Wizard: Select Fields



In this screen, choose the number of columns you want to place data in and specify the fields to display in each column.

To specify the number of columns in the report

Select a column number from the list. A report can have up to four columns. When you select a column number, the number of the Details boxes changes accordingly.

If you have selected fields into the Selected Fields boxes, they will be cleared when you select a new column number.

To select a field into a column

You can either drag the field into the Selected Fields box or select the field in the Available Fields box and then click the plus icon (+) above the Selected Fields box.

To select multiple fields all at once into a column

Hold down Shift and click to select a number of fields in a row or hold down Ctrl and click several discontinued fields and then perform the drag-and-drop action or use the plus icon.

To remove a field from a column

Select the field in the Selected Fields box and click the minus icon (–) above the Selected Fields box.

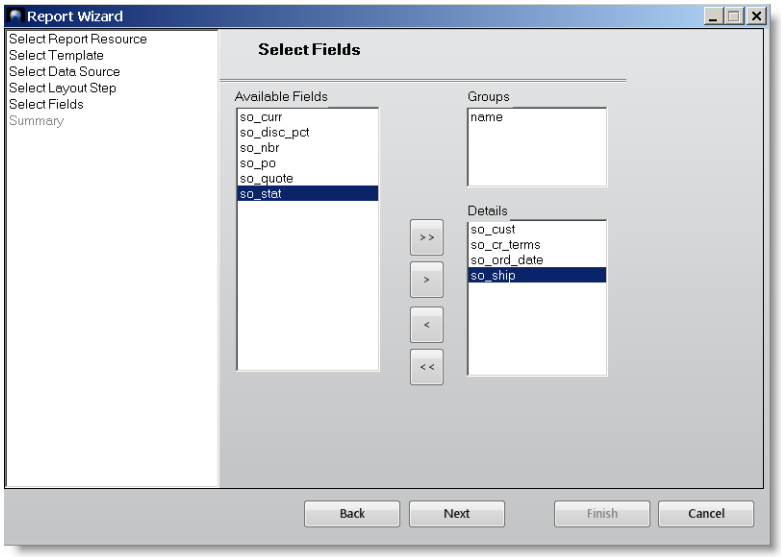
To remove multiple fields all at once from a column

Hold down Shift and click to select a number of fields in a row or hold down Ctrl and click several non-contiguous fields and then click the minus icon above the Selected Fields box.

Click Next to continue.

- This screen displays when you chose the column-style layout in the previous step.

Fig. 2.8
Report Wizard: Select Fields



In this screen, select the fields you want to display in the report and optionally specify fields to group data by in the report.

To select fields from the source table into the report

Select them in the Available Fields box on the left and drag them into the Details box.

To select fields by which to group data in the report

Select them in the Available Fields list box on the left and drag them into the Groups box on the right.

You can remove fields from the Groups and Details boxes through drag-and-drop in the opposite direction.

To move all the fields from the Available Fields box to the Details box and vice versa, use the >> and << buttons.

Click Next. Another layout option screen displays. Select a layout for the report and click Next.

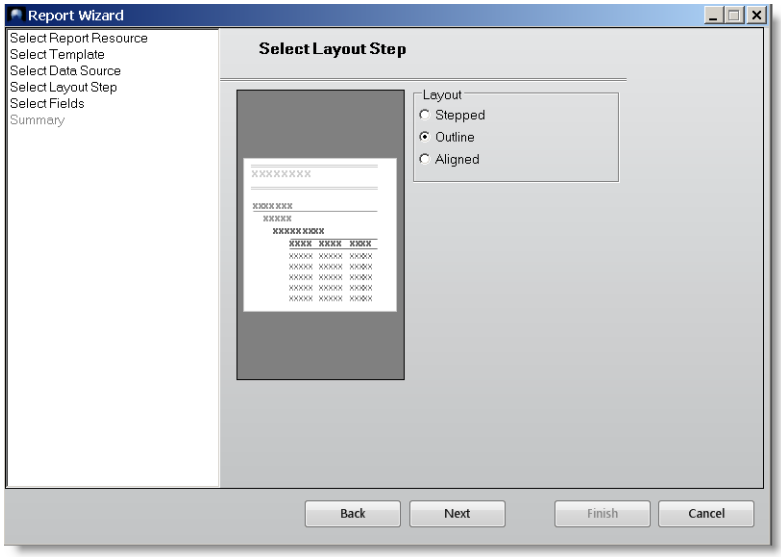
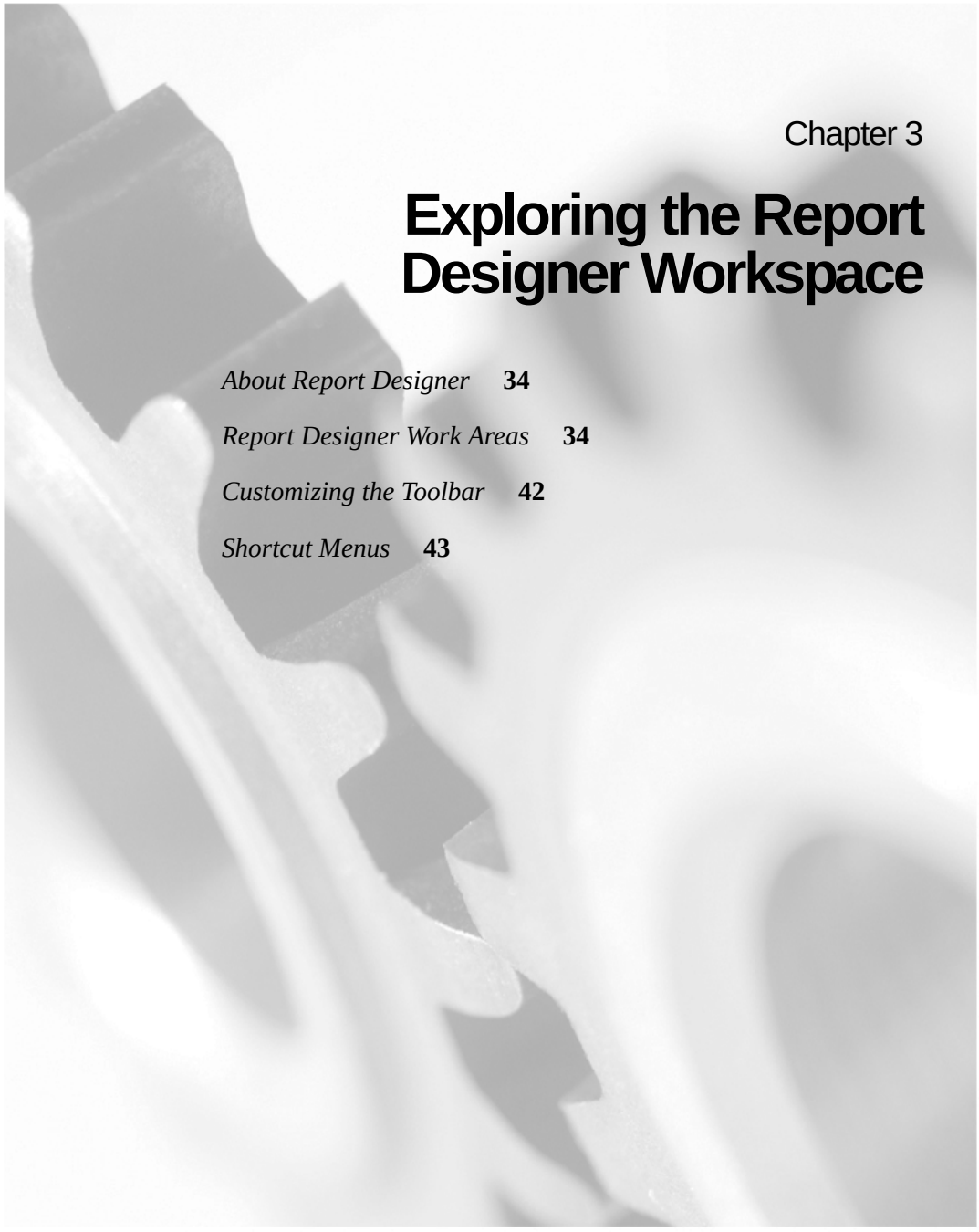


Fig. 2.9
Report Wizard: Select Column-Style Layout

8 The Summary screen recaps the information you have specified for the report definition. If you want to modify the settings, click Back to return to previous steps to edit them; otherwise, click Finish to complete the basic report setup and exit Report Wizard.

9 When you return to the Report Designer main screen, the report displays in the visual design mode in the Design pane based on the newly created report definition. Save the report as a new report definition. You can further customize it in Report Designer.



Chapter 3

Exploring the Report Designer Workspace

About Report Designer

Report Designer is a powerful, visual report design tool that lets you manage report definition files, define report layout, bind data to report fields, format reports, and enhance your reports using advanced components such as labels, charts, pictures, and drawings.

Report Designer features an intuitive, Microsoft Access-style user interface with a rich set of form components and formatting tools at your disposal that give you total control over the content, layout, and format of your reports. If you are familiar with the Microsoft Access report design view, you will find Report Designer very easy to use.

Report Designer provides a Report Wizard that guides you through the process of creating a basic report definition from scratch. You can then proceed from this starting point to further design specialized or more complex reports using Report Designer's comprehensive data-binding, layout design, and formatting tools.

Report Designer Work Areas

The main Report Designer window has the following components:

- Toolbar: Provides shortcuts to the most common design functions.
- Toolbox: Provides tools for creating report fields.
- Main Designer Pane: This is the main working area of the Designer. It shows the report's sections and fields and allows you to change the report definition.
- Properties Window: Allows you to edit properties for the objects that are selected in the Designer.

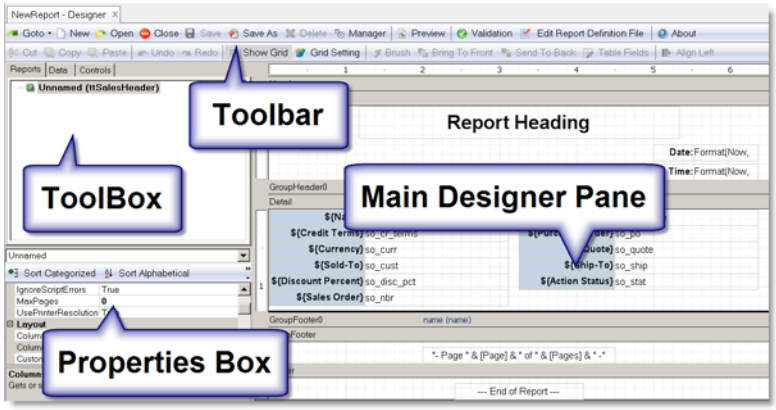


Fig. 3.1
Report Designer Workspace

Toolbar

The toolbar provides access to the following groups of functions:

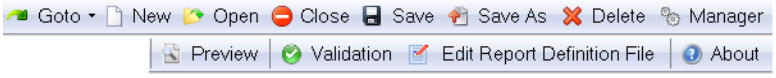


Table 3.1
Main Functions

Button	Name	Description
	Goto	Go to Report Resource Maintenance.
	New	Launch Report Wizard to create a new report definition.
	Open	Launch Report Definition Manager to open an existing report definition.
	Close	Close the current report definition.
	Save	Save the current report definition.
	Save As	Save the current report definition as another one.
	Delete	Delete the current report definition.
	Manager	Launch Report Definition Manager to delete existing report definitions, set the default report definition, and modify some of their attributes.

Button	Name	Description
	Preview	Display the current report definition in preview mode. In the preview mode, you can only navigate through the generated report.
	Validate	Check the validity of the current report definition file. If errors are found, error messages will be displayed.
	Edit Report Definition File	Open the current report definition file in code mode for editing.
	About	Display Report Designer version information.

Table 3.2
Edit Functions

Button	Name	Description
	Cut	Cut the selected objects on the report.
	Copy	Copy the selected objects on the report to the clipboard.
	Paste	Paste cut or copied objects from the clipboard to the currently selected area on the report.
	Undo	Undo any actions you have performed on the report.
	Redo	Redo the actions you have undone.

Table 3.3
Format Functions

Button	Name	Description
	Show Grid	Toggle background grid on and off.
	Grid Settings	Configure grid settings such as grid units and grid spacing.
	Brush	When multiple objects are selected, apply the format of the last selected object to all other selected objects.
	Bring to Front	Bring the selected object to the foreground.
	Send to Back	Send the selected object to the background.
The following functions only apply to multiple objects on the report and will be grayed if only one object is selected.		

Button	Name	Description
	Table Fields	Merge selected objects into table fields.
	Align Left	Align multiple selected objects to the left boundary of the last selected object.
	Align Center	Horizontally align multiple selected objects to the center of the last selected object.
	Align Right	Align multiple selected objects to the right boundary of the last selected object.
	Center Horizontally on Section	Horizontally position selected objects to the center of the section.
	Align Top	Align multiple selected objects to the top boundary of the last selected object.
	Align Middle	Vertically align multiple selected objects to the middle of the last selected object.
	Center Vertically on Section	Vertically position selected objects to the center of the section.
	Align Bottom	Align multiple selected objects to the bottom boundary of the last selected object.
	Equal Height	Resize multiple selected objects to the same height.
	Equal Width	Resize multiple selected objects to the same width.
	Equal Size	Resize multiple selected objects to the same height and width.
	Equal Horizontal Spacing	Reposition multiple selected objects so that they are equally spaced out horizontally.
	Decrease Horizontal Spacing	Horizontally reduce spacing between multiple selected objects.
	Increase Horizontal Spacing	Horizontally increment spacing between multiple selected objects.
	Equal Vertical Spacing	Reposition multiple selected objects so that they are equally spaced out vertically.

Button	Name	Description
	Decrease Vertical Spacing	Vertically reduce spacing between multiple selected objects.
	Increase Vertical Spacing	Vertically increment spacing between multiple selected objects.

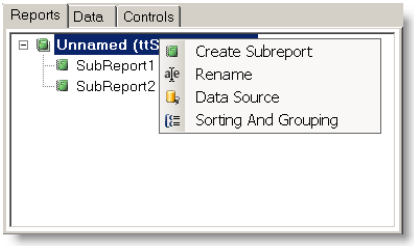
Toolbox

The Report Designer toolbox has three tabs: Reports, Data, and Controls.

Reports Tab

The Reports tab displays the report definition you are designing as well as all subordinate subreports embedded in the current report.

Fig. 3.2
Reports Tab



Right-clicking on the current report definition in this window brings up a shortcut menu that gives you access to a number of report design functions.

Data Tab

The Data tab contains three groups of data: fields, parameters, and report settings. Except for the Pointer button, which is used to deselect any currently selected object, each button under these groups creates a field on the report and initializes its properties.

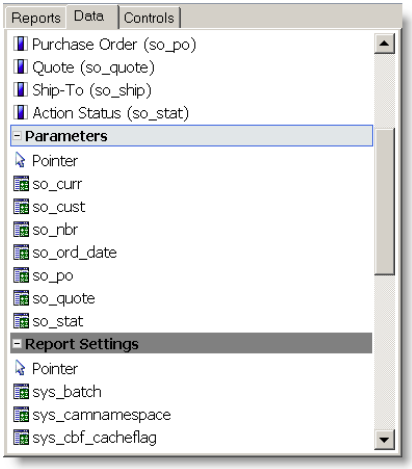


Fig. 3.3
Data Tab

The Fields group contains all the available fields that are bound to the source record set.

The Parameters group contains all the available search condition parameters under the Filter tab in Report Viewer.

The Report Settings group contains all the available setting variables under the Settings tab in Report Viewer.

Controls Tab

The Controls tab displays all the common controls and components that you can add to your reports.

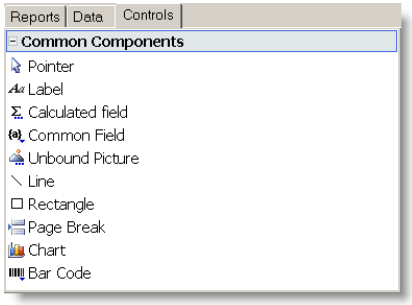


Fig. 3.4
Controls Tab

Table 3.4
Controls

Control	Name	Description
	Pointer	Deselects any field on the report and releases the mouse focus.
	Label	Creates a calculated field. When you click this button, the code editor dialog box appears so you can enter the VBScript expression whose value you want to display.
	Calculated Field	Creates a calculated field. When you click this button, the code editor dialog box appears so you can enter the VBScript expression whose value you want to display.
	Common Calculated Field	Creates a field with a commonly used expression. When you click this button, a menu appears and you can select expressions that render the date or time when the report was created or printed, the page number, page count, “page n of m,” or the report name.
	Unbound Picture	Creates a field that displays a static picture, such as a logo. When you click this button, a dialog box appears to prompt you for a picture file to insert in the report. A copy is made of the picture you select and placed in the same directory as the report file. You must distribute this file with the application unless you embed the report file in the application. When you embed a report file in your application, any unbound picture files are embedded, too.
	Line	Creates a line. Lines are often used as separators.
	Rectangle	Creates a rectangle. Rectangles are often used to highlight groups of fields or to create tables and grids.
	Page Break	Creates a field that inserts a page break.
	Chart	Creates a field that displays a chart. Unlike most bound fields, Chart fields display multiple values. To select the data you want to display, set the Chart field’s DataX and DataY properties. To format the values along the X and Y axis, set the FormatX and FormatY properties. You can use the ChartType property to specify the type of the chart: bar, column, pie, scatter, line, or area.
	Barcode	Creates a field that displays a barcode that is rendered from your designated data. To specify a barcode standard, select the standard you want to use from the Barcode property list in the Properties box. You can choose from the following barcode standards: Code39, Code 93, Code128, Code2of5, Codeabar, PostNet, Ean13, Ean8, UpcA.

Properties Window

The Properties window allows you to edit properties for the objects that are selected in the Designer.

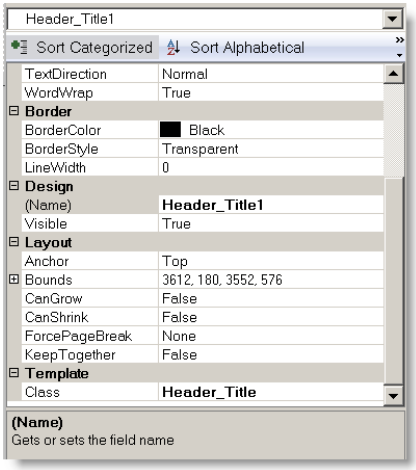


Fig. 3.5
Properties Window

The object drop-down list always displays the currently selected object on the report. If no report object is selected, the box displays the current report definition type.

To select an object, either click on it on the report in the design pane, or select the name of the object from the object drop-down list.

The Properties window's toolbar lets you change the way properties are displayed and revert a property's value to the default.

Table 3.5
Properties Window Toolbar

Button	Name	Description
	Sort by Category	Groups properties under categories, which vary according to the type of the selected object. You can expand or collapse a category to show or hide the properties under it. This is the default display mode.
	Sort Alphabetically	Sorts properties in alphabetical order.
	Use Default Value	Reverts a selected property value to the default. If the value of the selected property is already the default value, this button is grayed.

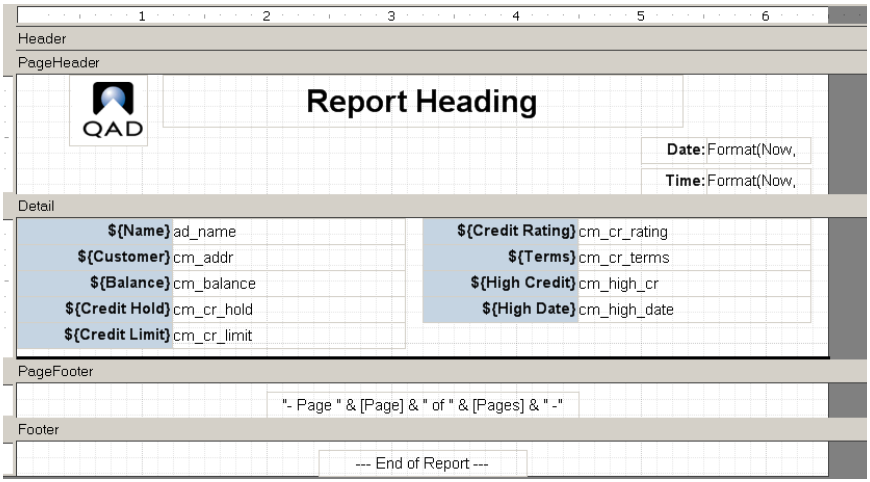
The main area of the Properties window displays all the properties of the selected object. They change as the selected object changes.

The property value displays to the right of the property name. To edit a property value, click on the value field and it turns into a text field, drop-down box, or combo box, depending on the type of the value; specify the value you want.

Design Pane

The Design pane is the main working area of the Designer. It shows the report's sections and fields and allows you to visually change the report definition through point-and-click. For details on how to design a report, see Chapter 4, "Designing Reports in Report Designer" on page 45.

Fig. 3.6
Design Pane



Customizing the Toolbar

Right-click a toolbar and choose from the shortcut menu to change the way the toolbar looks.

Fig. 3.7
Customize the Toolbar

Show Text: Select this option to display tooltips for the toolbar menus; otherwise, clear this option.

Use Large Icon: Select this option to display large menu icons; clear this option to show small icons.

Shortcut Menus

Report Designer provides context-sensitive shortcut menus that give you access to most commonly used functions.

To bring up shortcut menus:

- Right-click a report definition under the Reports tab in the toolbox.
- Right-click on a blank space in the Design pane.
- Right-click an object in the Design pane.
- Right-click a subreport in the Design pane.

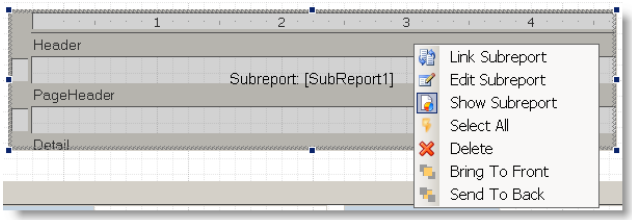


Fig. 3.8 Subreport Shortcut Menu

Table 3.6 Subreport Shortcut Commands

Button	Name	Description
	Link Subreport	Opens a dialog box that lets you define which records will be included in the subreport by specifying a master field in the main report and a child field in the subreport. This will set the Text property on the subreport field to an expression that will be used as a filter on the subreport data source.
	Edit Subreport	Open the subreport fully in the Design pane for editing.

Chapter 4

Designing Reports in Report Designer

The QAD Reporting Framework gives you not only the simplicity to build basic reports from scratch, but also the flexibility to tailor them to meet your specific requirements.

The basic report generated for you by the Report Wizard is a good starting point, but you will usually need to adjust and enhance it to get exactly what you want. You can do this with Report Designer.

Launching Report Designer

Use one of the following ways to access Report Designer:

- Type Report Resource Designer in the menu search field and press Enter.
- If you have created a menu item for your report, locate it in the Applications Pane and right-click on it; then choose Design from the shortcut menu. The Report Designer window appears.

Note Before you can access Report Designer to create a report definition, you must first create a valid report resource.

Managing Report Definition Files

In Report Designer, you can create, load, and delete report definition files, as well as edit them either in the WYSIWYG (What You See Is What You Get) or code edit mode.

Creating a New Report Definition

Click the New button on the Report Definition toolbar. Report Wizard takes you through the process of creating a basic report definition. Click the Save icon on the toolbar to save the definition as an XML file. For details on creating a basic report definition using Report Wizard, see “Creating a Report Definition” on page 25.

Loading an Existing Report Definition File

Click the Open button on the toolbar; then in the Select Report Definition window, double-click the report definition you want to load. The Select Report Definition window also let you enter search conditions to search for the report definition you want to load.

Using Report Definition Manager

You can use Report Definition Manager to delete existing report definitions as well as modify some of their attributes.

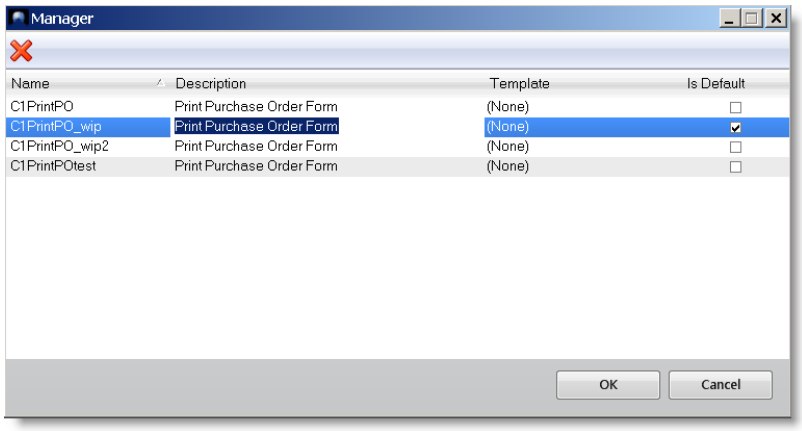


Fig. 4.1
Report Sections Example

To launch Report Definition Manager

Click the Manager button on the toolbar.

To delete an existing report definition

In the Report Definition window, click the report definition and then click the Delete button at the top. Confirm the deletion when prompted.

To select a different template for a report definition

Click the current template next to the report definition and select a different template from the list. The layout and formatting of the report definition will be changed after you assign a different template to it.

To set the default report definition for the report resource

Select the Is Default check box for the report definition. You must set one and only one default report definition.

When you open the report resource from the Applications menu tree, the default report definition is loaded.

Working with Report Sections

About Report Sections

A basic report is divided into five sections: header, page header, details, page footer, and footer. The sections contain fields that hold the labels, variables, and expressions that you want in the printed report. If you add groups to a report, the report will also contain a group header and a group footer section. For example, a report with 3 grouping levels will have 11 sections.

The sections of the report determine what each page, group, and the beginning and end of the report look like. The following table describes where each section appears in the report and what it is typically used for.

Table 4.1
Report Sections

Section	Appears	Typically Contains
Report Header	Once per report	The report title and summary information for the whole report.
Page Header	Once per page	Labels that describe detail fields, and/or page numbers.
Group Header	Once per group	Fields that identify the current group, and possibly aggregate values for the group; for example, total, percentage of the grand total.
Detail	Once per record	Fields containing data from the source record set.
Group Footer	Once per group	Aggregates values for the group.
Page Footer	Once per page	Page number, page count, date printed, report name.
Report Footer	Once per report	Summary information for the whole report.

In this example, the Header section contains a label with the report title. The Page Header section contains labels that display the current date and time. The Group Header section contains labels that identify the fields in

the Detail section, and the Page Footer section contains fields that show the page number and the total page count for the report. Data is grouped inside a group section marked by a group header and a group footer.

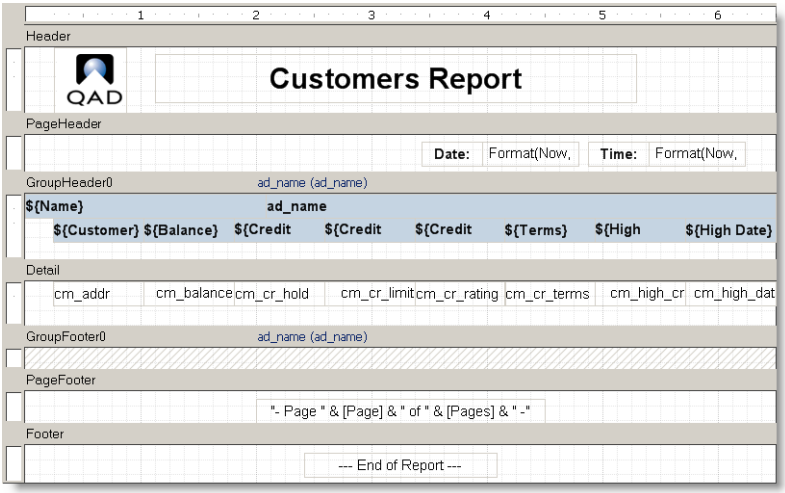


Fig. 4.2
Report Sections Example

Note that sections can be made invisible, but they cannot be added or removed, except by adding or removing groups.

The following diagram shows how each section is rendered on a typical report.

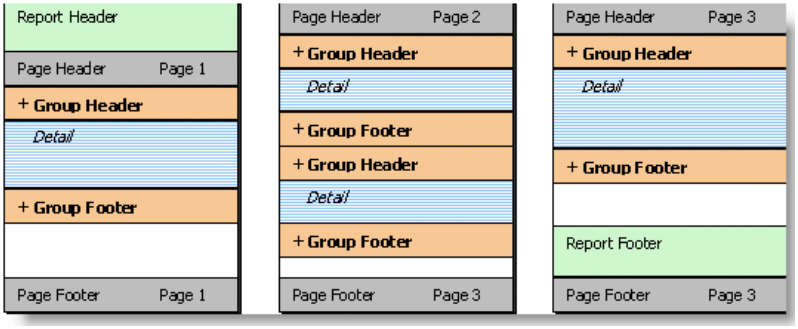


Fig. 4.3
Rendering a Report

- Report Header

  The first section rendered is the Report Header. This section usually contains information that identifies the report.

- Page Header

  After the Report Header comes the Page Header. If the report has no groups, this section usually contains labels that describe the fields in the Detail Section.

- Group Headers and Group Footers

  The next sections are the Group Headers, Detail, and Group Footers. These are the sections that contain the actual report data. Group Headers and Footers often contain aggregate functions such as group totals, percentages, maximum and minimum values, and so on. Group Headers and Footers are inserted whenever the value of the expression specified by the GroupBy property changes from one record to the next.

- Detail

  The Detail section contains data for each record. It is possible to hide this section by setting its Visible property to False, and display only Group Headers and Footers. This is a good way to create summary reports.

- Page Footer

  At the bottom of each page is the Page Footer Section. This section usually contains information such as the page number, total number of pages in the report, and/or the date the report was printed.

- Report Footer

  Finally, the Report Footer section is printed before the last page footer. This section is often used to display summary information about the entire report.

- Customized sections

  You can determine whether a section is visible by setting its Visible property to True or False. Group Headers can be repeated at the top of every page (whether or not it is the beginning of a group) by setting their Repeat property to True. Page Headers and Footers can be removed from pages that contain the Report Header and Footer sections by setting the PageHeader and PageFooter properties on the Layout object.

Resizing a Report Section

To resize a section, select its border and with your mouse pointer drag to the position where you want it. The rulers on the left and on top of the design window show the size of each section (excluding the page margins). Note that you cannot make the section smaller than the height and width required to contain the fields in it. To reduce the size of a section beyond that, move or resize the fields in it first, then resize the Section.

To see how this works, move the mouse to the area between the bottom of the Page Header section and the gray bar on top of the Detail Section. The mouse cursor changes to show that you are over the resizing area. Click the mouse and drag the line down until the section is about twice its original height.

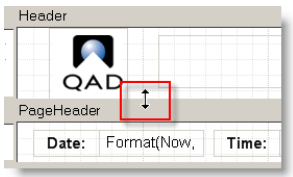


Fig. 4.4 Resizing a Report Section

Release the mouse button and the section is resized.

Hiding a Report Section

You can hide a report section so that it will not appear in the printed report. However, a hidden section is still visible in the design view.

To hide a report section, click a section to select it; then set its Visible property to False in the Properties box.

Grouping and Sorting

You can organize the data in your report by grouping and sorting data, using running sums, and creating aggregate expressions.

Grouping and Sorting Data

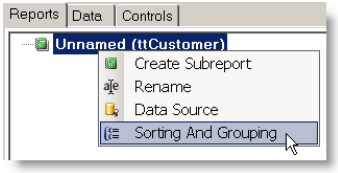
After designing the basic report layout, you may decide that grouping the records by certain fields or other criteria would make the report easier to read. Grouping allows you to separate groups of records visually and display introductory and summary data for each group. Groups are also used for sorting the data, even if you do not plan to show the Group Header and Footer sections.

You can also specify how each group should be sorted using the group's GroupBy and Sort properties.

To add or edit the groups and specify the sorting rule in the report

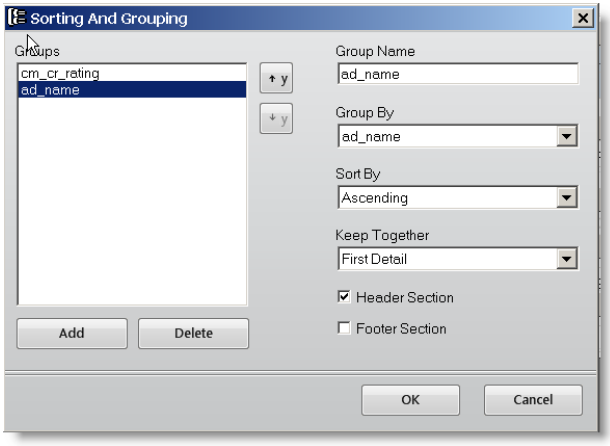
1 In Report Designer, right-click a report under the Reports tab in the toolbox; then select Sorting and Grouping from the shortcut menu.

Fig. 4.5
Select Sorting and Grouping



2 The Sorting and Grouping dialog box appears. Use this dialog box to create, edit, reorder, and delete groups.

Fig. 4.6
Sorting and Grouping Dialog Box



3 To create a group, click the Add button and set the properties for the new group. The Group By field defines how the records will be grouped in the report. For simple grouping, you can select fields directly from the drop-down list.

4 Next, select the type of sorting you want (Ascending in this example). You can also specify whether the new group will have visible Header and Footer sections, and whether the group should be rendered together on a page.

5 If you add more fields, you can change their order using the arrow buttons on the right of the Groups list. This automatically adjusts the position of the Group Header and Footer sections in the report. To delete a field, use the Delete button.

6 Once you are done arranging the fields, click OK to dismiss the dialog box and see the changes in the Designer. On the top of the new sections there are labels that contain the section name and the value of the group's GroupBy property.

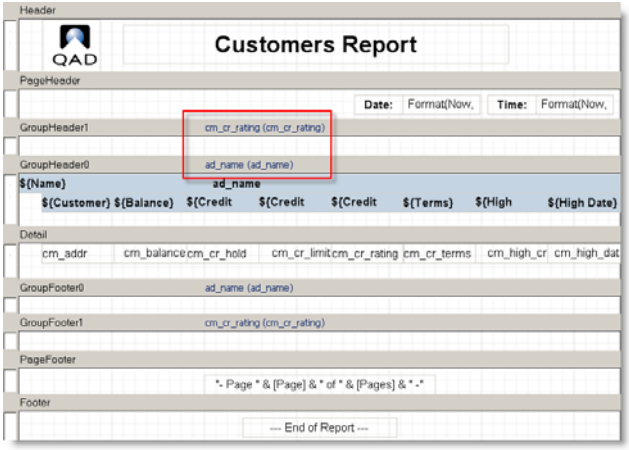


Fig. 4.7
New Group Section

Adding Running Sums

You can easily maintain running sums over groups or over the entire report.

To keep running sums over groups:

1. In Report Designer, open the report definition you want to design.
2. Switch to the Item tab in the toolbox and click Calculated Field under the Common Components group.
3. The VBScript Editor displays. Enter the following script:

   ```
   Sum(FieldToSumUp)
   ```

   Where *FieldToSumUp* is the name of the field you want to sum up.
4. Move the mouse pointer over to the GroupHeader section of the report and the pointer changes into a cross-hair. Click and drag to define the rectangle that the new field will occupy, and then release the button to create the new field.

Enhancing the Report with Fields

To enhance your report, you can add fields (for example, lines, rectangles, labels, pictures, charts, and so on) to any section. You can also modify the existing fields by changing their properties with the Properties window, or move and resize the fields with the mouse.

Creating Reporting Fields

With the report definition loaded into Report Designer and a data source defined, you can add and edit report fields.

You can bind three different types of data to the fields you add to a report:

- Fields: Values of fields that are bound to the source record set.
- Parameters: Values of search condition parameters under the Filter tab in Report Viewer.
- Report settings: Values of settings variables under the Settings tab in Report Viewer.

The Report Designer toolbox allows you to easily add fields to your report.

To add a field to your report

1. Click the Data tab in the Report Designer toolbox.
2. Click and expand a group to display all available data buttons.
3. Click the field you want to add to the report; then move the mouse pointer over the report and the pointer changes into a cross-hair.
4. Click and drag to define the rectangle that the new field will occupy, and then release the button to create the new field.

If you change your mind, press Ctrl+Z or click the Undo button to cancel the operation.

You can also add fields by copying and pasting existing fields, or by holding down the CTRL key and dragging a field or group of fields to a new position to create a copy.

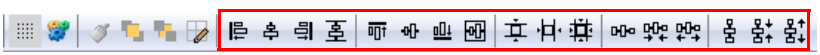
Manipulating and Formatting Fields

You can use the mouse to select fields in Report Designer as usual:

- Click a field to select it.
- Shift-click a field to toggle its selected state.
- Ctrl-drag creates a copy of the selected fields.
- Click the empty area and drag your mouse pointer to select multiple fields.
- With your mouse pointer, drag field corners to resize fields.
- Double-click right or bottom field corners to auto size the field.

To select fields that intersect vertical or horizontal regions of the report, click and drag the mouse on the rulers along the edges of the Designer. If fields are small or close together, it may be easier to select them by name. You can select fields and sections by picking them from the drop-down list above the Properties window.

Format Fields



When multiple fields are selected, you can use the buttons on the Format toolbar to align, resize, and space them. When you click any of these buttons, the last field in the selection is used as a reference and the settings are applied to the remaining fields in the selection.

Apply Styles

The Brush button applies the style of the reference field to the entire selection. The style of a field includes all font, color, line, alignment, and margin properties. You can use the Properties window to set the value of individual properties to the entire selection.

Determine Order For Overlapping Fields

If some fields overlap, you can control their z-order using the Bring to Front/Send to Back buttons. This determines which fields are rendered before (behind) the others.

Move Fields Using the Keyboard

You can also select and move fields using the keyboard:

- Use the TAB key to select the next field.
- Use SHIFT-TAB to select the previous field.
- Use the arrow keys to move the selection one pixel at a time (or shift arrow to move by five pixels).
- Use the DELETE key to delete the selected fields.
- When a single field is selected, you can type into it to set the Text property.

Changing Field, Section, and Report Properties

Once an object is selected, you can use the Properties window to edit its properties.

When one or more fields are selected, the Properties window shows property values that all fields have in common, and leaves the other properties blank. If no fields are selected and you click on a section (or on the bar above a section), the Section properties are displayed. If you click the gray area in the background, the Report properties are displayed.

To see how this works, click the label in the Header section and change its Font and ForeColor properties. You can also change a field's position and dimensions by typing new values for the Left, Top, Width, and Height properties.

The Properties window expresses all measurements in twips (the native unit used by Report Designer), but you can type in values in other units (in, cm, mm, pix, pt) and they will be automatically converted into twips. For example, if you set the field's Height property to 0.5 inches, the Properties window will convert it into 720 twips.

A twip (derived from TWentieth of an Imperial Point) is a typographical measurement, defined as 1/20 of a typographical point. One twip is 1/1440 inch or 17.639 µm when derived from the PostScript point at 72 to the inch, and 1/1445.4 inch or 17.573 µm based on the printer's point at 72.27 to the inch.

Changing the Data Source

1. In Report Designer, right-click a report under the Reports tab in the toolbox; then select Data Source from the shortcut menu.
2. The Select Data Source dialog box displays. Click to select the table you want to set as the new data source; then click OK.
3. The data source is changed. When you switch to the Data tab in the toolbox, you can see a new set of data-bound fields.

Note When you select a new data source, any field associated with the old data source becomes invalid and will not display data in the report.

Controlling the Maximum Number of Records Per Report Page

You can control the maximum number of records to display on each page of a report by using the page break control.

In Report Designer, place the page break control beneath the data fields on the report; then in the Properties pane, specify the value of its RecordsPerPage property. During report rendering, when the number of records on a page reaches the maximum number of records allowed, the remaining records go to subsequent pages.

Note The records-per-page property does not apply to reports in the Excel format. All records are displayed continuously in an Excel report.

Note A report contains subreports and when the records in a subreport reaches the maximum number per page allowed, the report page breaks even when the parent report has not reached the records per page limit.

Creating Localizable Labels

You can design reports that can be readily localized into multiple languages. When localized, the system dynamically reads the label master table to determine the appropriate labels to display on your reports.

To make a label localizable, write the label in this format: ${*Term*}. The term is the key that links labels to fields, allowing the system to determine which label to display. The system validates the language code and accesses the term to find the right label to display in the corresponding language. The term remains the same regardless of the language selected. Terms display in all uppercase with underscores; for example, SALES_ORDER is the term that translates into the string Sales Order when the language code is US.

Field labels bought in from report data sources can be found in the label master. If you place a label on the report that does not exist in the label master, you should create a record for it using Label Master Maintenance (36.4.17.24) so that the label can be translated and maintained.

Note Sample proxy programs currently provided by QAD and the browse data source provider use English labels instead of terms for generating labels on the report. This may cause bad labels to be generated by the Report Resource Designer program's Wizard when run in other language environments. To address this problem, you can either manually change the label text inside ${ } to appropriate label terms present in the label master, or modify the proxy programs to retrieve terms instead of labels to be displayed as labels on the report.

Creating a Master-Detail Report Using Subreports

About Subreports

Subreports are regular reports contained in a field in another report (the main report). Subreports are usually designed to display detail information based on a current value in the main report, in a master-detail scenario. You can create multiple subreports in a main report.

For example, the main report contains product categories and the subreport in the Detail section contains product details for the current category.

Creating a Master-Detail Report

1 Create a master report.

Create a basic report using Report Wizard and, optionally, further customize the report using the design tools provided in Report Designer. For details on creating a report definition using Report Wizard, see “Creating a Report Definition” on page 25.

2 Create a subreport.

a In Report Designer, right-click a report definition under the Reports tab in the toolbox and choose Create Subreport from the shortcut menu.

b The Report Wizard window displays. Follow the same steps you used when creating the basic report. See “Creating a Report Definition” on page 25.

c When you have created a basic detail report using Report Wizard and return to the Report Designer main screen, in the Detail section of your report, click and drag the mouse pointer to make the field for the subreport. The subreport is embedded in the master report.

d Optionally, right-click the subreport field and select Edit Subreport from the shortcut menu to fully open the subreport in the Design pane; then further customize the report using the design tools provided in Report Designer.

e Optionally, you can repeat the previous steps to add multiple subreports to the master report.

3 Link the subreport to the master report

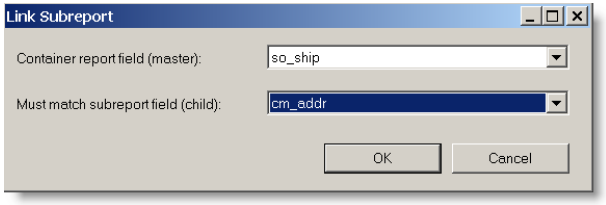
The master-detail relationship is controlled by the Text property of the subreport field. This property should contain an expression that evaluates into a search condition that can be applied to the subreport data source.

The Report Designer can build this expression automatically for you. Complete the following steps:

a Right-click the subreport field and select Link Subreport from the menu.

b A dialog box appears and lets you to select which fields should be linked. Once you make a selection and click OK, the Report Designer builds the link expression and assigns it to the Text property of the subreport field in the background.

Fig. 4.8
Link Subreports



4 The master-detail report is created.

Note If you do not link the subreport to the master report, a Descartes Accumulate of data will be displayed on the report.

Adding Unbound Images to the Report

Unbound images are static images such as logos and watermarks that are not stored in the database.

1. Click the Controls tab in the Report Designer toolbox.
2. Click and expand the Common Components tree to display all available components.
3. Click the Unbound Picture button.
4. A file browse dialog box appears. Locate and select the image file you want to add to the report.
5. Move the mouse pointer over the report and the cursor changes into a cross-hair. Click on your report where you would like to place the image, and then resize the field to show the image.

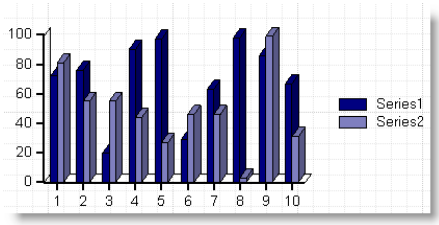
If you change your mind, press Ctrl+Z or click the Undo button to cancel the operation.

You can also add fields by copying and pasting existing images, or by holding down the CTRL key and dragging a field or group of fields to a new position to create a copy.

Adding Charts to the Report

1. Click the Controls tab in the Report Designer toolbox.
2. Click and expand the Common Components tree to display all available components.
3. Click the Chart button.
4. Move the mouse pointer over the report and the cursor changes into a cross-hair. Click on your report where you would like to place the chart, and then resize the field to show the image.

Fig. 4.9
Link Subreports



5 With the chart selected, modify the following required chart properties in the Properties window:

- DataX: Specify the data series to plot the X axis of the chart; for example, month(sod_date).
- DataY: Specify the data series to plot the Y axis of the chart, separating multiple data series using semicolons; for example, sod_qty_ord; sod_qty_pick.
- ChartType: Specify the type of the chart: bar, column, pie, scatter, line, or area.

6 Optionally modify other properties to tailor the chart to your specific needs.

Maintaining Customized Report Parameters

Report parameters serve two purposes:

- In Report Designer, report parameters can be found under the Data tab in the toolbox and you can bind them with fields to enhance your reports. See “Enhancing the Report with Fields” on page 54.
- Report parameters also appear in the Report Filter window as filter criteria that you can set to filter data in your reports. See “Running Reports” on page 93.

Report parameters that appear in Report Designer and Report Filter are predefined data hard-coded in the data source proxy program and can only be maintained by modifying the program code.

You can also create and maintain your own report parameters, but these customized parameters can only be used in Report Designer to enhance your reports. Although they also appear in Report Filter, they cannot be used to filter data when generating reports.

Use Report Parameter Maintenance to create, edit, and delete customized report parameters.

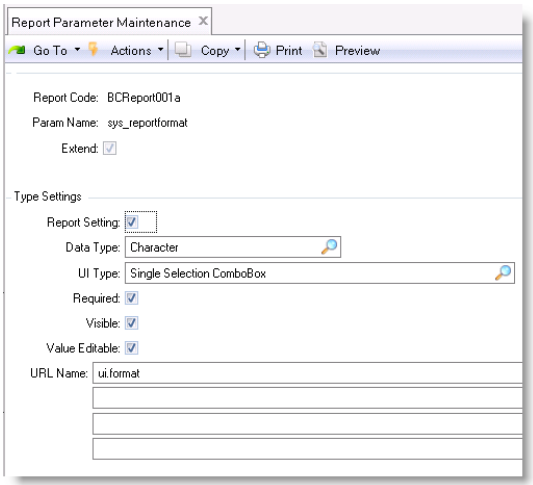


Fig. 4.10
Report Parameter Maintenance

Param Name. Enter a parameter name. If the parameter name does not exist, a new parameter will be created. Parameters with the sys prefix are built-in system parameters.

Extend. If you specified an existing built-in system parameter, the system makes a copy of the parameter for the report resource. You can choose whether or not to modify the inherited standard parameter settings.

Yes: Modify the parameter settings based on the inherited values. If you select this option, the parameter settings will display with inherited values and you must change at least one of the settings.

No: Accept all the inherited standard parameter settings without making any changes. If you choose No, the parameter settings will not display.

The default option is No.

For non-system parameters, this field is not applicable.

Report Setting. Specify whether or not this is a report setting parameter.

Data Type. Specify a data type for the parameter: character, date, datetime, decimal, integer, logical, and or time.

UI Type. Specify a UI type that determines how the parameter will be displayed in Report Viewer: calendar, datetime prompt, multiple selection combo box, single selection combo box, text box, or time prompt.

This field is currently not implemented.

Required. Specify whether this parameter is required in Report Viewer.

Visible. Specify whether the parameter will be displayed in Report Viewer.

Value Editable. Specify whether the user can edit the value of the parameter.

URL Name. This field is currently not implemented.

Value Type. Specify the type of the first parameter value in the filter criteria:

- Constant: A hard-coded string.
- Method: A system built-in method such as Today and Time.
- Session: A key/value pair retrieved from the current session that contains information such as the current username.

Value. Specify the default of the first parameter value. The value must be compatible with its value type. If the parameter data type is logical, the value must be true or false (both in lower case).

2nd Type. If the filter criteria contains two parameter values; for example, salesperson in (‘Hunter’, ‘Johnson’); specify the type of the second parameter value.

2nd Value. If the filter criteria contains two parameter values; for example, salesperson in (‘Hunter’, ‘Johnson’); specify the default of the second parameter value. If the parameter data type is logical, the value must be true or false (both in lower case).

Operator Allowed. Specify whether the parameter can be used with operators in the filter criteria.

Operator Editable. If Operator Allowed is Yes, specify whether the user can edit the operator used with the parameter in the filter criteria.

Operator. If Operator Allowed is Yes, specify the default operator used with the parameter in the filter criteria.

Working with Templates

When you create a basic report from scratch using Report Wizard, you supply the information required by Report Wizard and the system creates a report definition with consistent-looking controls neatly arranged in your desired layout, along with common report sections including report header, report footer, page header, and page footer. You do not need to realign the fields or unify font size or color of the labels because the report template takes care of most of the formatting and layout for you. Even if you did not select a template when creating the report definition, the new report uses a built-in report template by default, which determines the contents and layout of the report header, footer, page header, page footer, as well as the formatting of all the control elements in the report. You may be already working with the report template and enjoying the benefits it brings even before you realize it.

About Report Templates

A report template is a pre-designed XML file that contains formatting and layout information that can be applied to report definitions. It can be edited in Template Designer either visually or in the text editor mode.

Template are used for creating attributes of fields and other objects that can be inherited across many reports, such as certain fonts or colors for text objects that you might not want to specify for each report. You can inherit these globally through templates, and also make mass modifications across reports if you want to change them. For example, you can place a corporate logo in the header of a report template and it will appear on all the reports the template is applied to.

Using templates lets you:

- Create new report definitions quickly and efficiently.

- Update multiple element styles in report definitions on the fly.
- Easily standardize report definitions with consistent styles.
- Separate styles from data in designing report definitions.

Elements in the report template represent classes or styles that can be applied to corresponding elements in a report definition based on a class-mapping relationship.

A field defined in the report template represents a field class identified by a unique class name. When the field class is applied to a field in a report definition, most of its properties are carried over to the field so that the field takes on the same formatting and layout.

The header, page header, page footer, footer, or a group section defined in the report template represents a section class identified by a unique class name. When the section class is applied to a section in a report definition, it is virtually copied over to the report definition complete with all the elements in it.

About Template Designer

You design report templates in Template Designer. If you are familiar with Report Designer, there is no learning curve in using Template Designer.

Template Designer is almost identical to Report Designer, except for the following differences.

- The Toolbox only displays the Controls tab which contains a limited set of control components that can be used in the template: Label, Calculated Field, Common Field, Unbound Pictures, Line, Rectangle, and Chart.
- The toolbar contains the following buttons that are specific to Template Designer:

Table 4.2 Template Designer Specific Toolbar Buttons

Button	Name	Description
	Edit Template	Open the current report template file in code mode for editing.
	Add Group	Add groups to the template to create new section classes.
	Configure	Configure default fields and sections mapping.
	Import	Import report template files.
	Export	Export report template files.

Creating a New Report Template

1. Launch Template Designer. Type Template Designer in the menu search field and press Enter.
2. Click the New button on the toolbar.
3. In the Create Template dialog box, enter a unique template name and click OK.

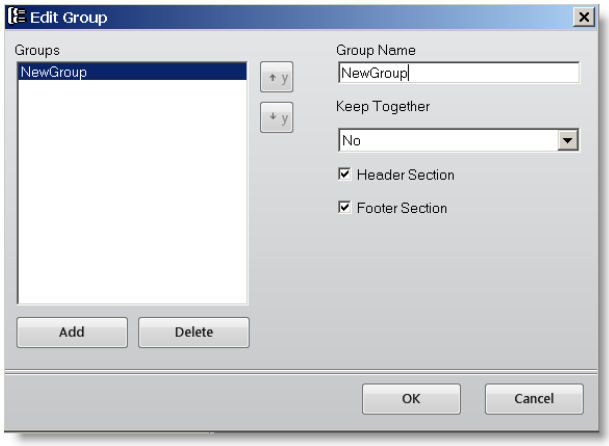
   Important QAD-provided built-in reports, report resources, and templates all begin with “QAD_”. Do not create or modify reports, report resources or templates with this prefix. Otherwise, your customized changes will get overwritten during system upgrades from QAD.
4. In the Design pane, create and format field classes in the same way as you work with fields when working with a report definition. Provide unique class names for the classes. See “Enhancing the Report with Fields” on page 54.
5. If you want to define a header, page header, page footer, and footer section class name, click the default section name and enter a new name in the (name) field in the Properties pane.

6 If you want to add new sections, use the following steps:

a Click the New Group button on the toolbar.

b In the Edit Group dialog box, click Add and specify the properties for the new group.

Fig. 4.11
Add Group



c Click OK.

7 Configure the default section and field class mapping to specify the default classes to be applied to the corresponding sections and fields in the report definition.

a Click the Configure button on the toolbar. The Class Configuration Form dialog box appears.

b Under the Section Configuration tab, specify a class name for each section type.

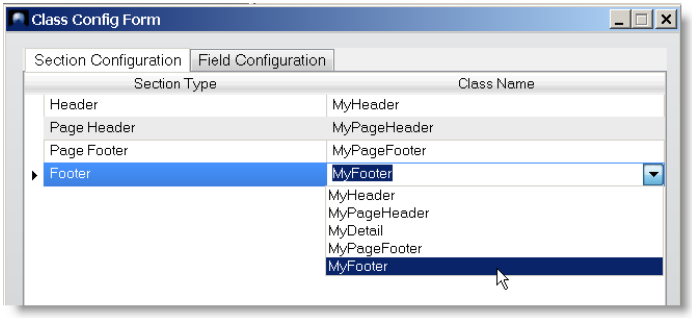


Fig. 4.12
Section Configuration

c Under the Field Configuration tab, for each data type, select a section in the template and specify a class defined within that section.

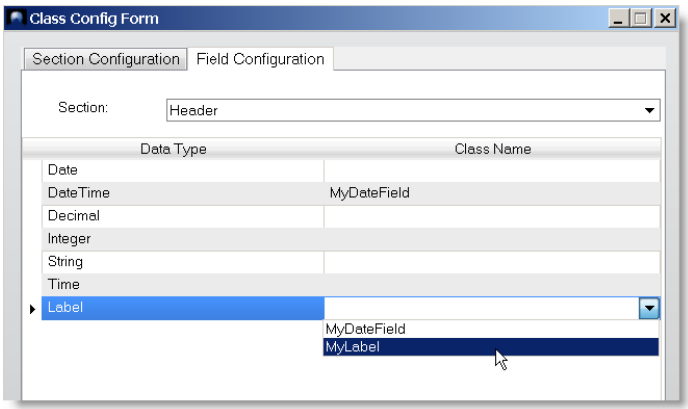


Fig. 4.13
Field Configuration

d Click OK.

8 Back in the Template Designer main screen, click the Save button on the toolbar to save the template.

Applying Report Templates

After you design a report template, you can apply it to multiple report definitions to enforce a consistent look and feel across all these reports.

Applying a report template to a report definition applies all the mapped classes defined in the template to the corresponding classes in the report definition based on a class-mapping relationship. This takes place on two levels:

- On the section level, when a section class—a class defined by the header, footer, page header, page footer, detail, or a group section in the report template—is applied to the mapped header, footer, page header, page footer section in the report definition, all the contents in the template section are copied over to the report definition section.

  Note This does not apply to the detail and group sections in the report definition.

- On the field level, when a field class—a class defined by the field in the report template—is applied to a mapped field in the report definition, a predefined set of properties are copied from the template field to the report definition field.

The class-mapping relationships are defined as a step in “Creating a New Report Template” on page 67.

To apply an existing report template to a report definition

Do one of the following:

- When “Creating a Report Definition” using Report Wizard, in the Select Template step of the Report Wizard, select the report template from the template list.
- In Report Designer, click the Manager button on the toolbar and assign the report template to the report definition in Report Definition Manager. For details, see “Managing Report Definition Files” on page 46.

Once the report template is applied to the report definition, the changes in layout and formatting immediately take effect in Report Designer.

Customizing Template-Based Report Definitions

Applying a report template to a report definition applies classes to affected sections and fields in the report definition across the board. You can still customize the report definition by applying a different class to individual report element or modifying their other properties in Report Designer.

However, when you manually change an element's properties including applied class in a template-based report definition and confirm the change, the element is disengaged from the report template in the report definition and will no longer be affected by the template.

Applying a Report Template Class to an Individual Element

1. In Report Designer, select the element by either clicking it in the Application area, or selecting it from the element list in the Properties window.
2. In the Properties window, expand Template, and click the Browse button next to the Class property.
3. The Select Class dialog box displays all the classes of the matching type (section or field) defined in the current report template. Select the class you want to apply and click OK.

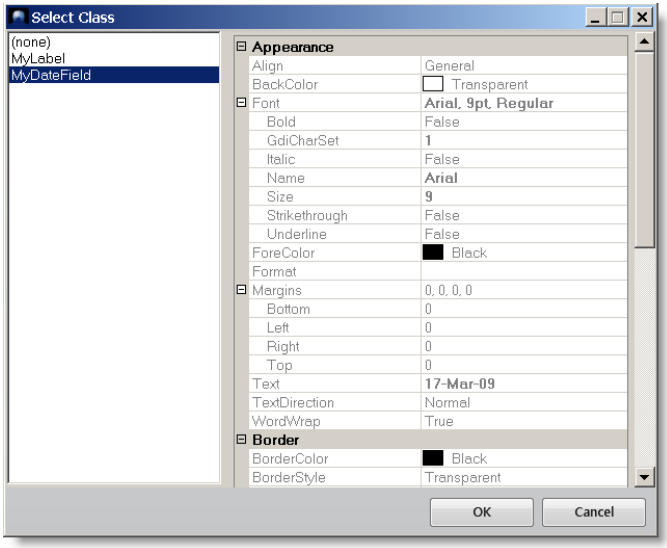


Fig. 4.14
Select Class

4. The class is applied to the element with immediate changes in layout and formatting.

Importing and Exporting Report Resources

Importing Report Resources

Use Report Resource Import (36.4.21.21) to import report resource data files into the system.

Import Directory. Enter the directory from which to import report resource data files. You must enter a valid directory in this field.

Include Subdirectories. Indicate whether to import resource data files in the subdirectories under the specified import directory.

Include Files. Enter the report resource data (`.rro`) files you want to import into the system using wildcards. Use period (.) to represent any single character and asterisk (*) to represent any number of characters in the file name to include multiple report resource data files. If you leave this field blank, all report resource data files in the specified directory will be included.

Exclude Files. Enter the report resource data (`.rro`) files you want to exclude for import into the system using wildcards. Use period (.) to represent any single character and asterisk (*) to represent any number of characters in the file name to exclude multiple report resource data files.

Schema File Location. Enter the full path name and filename of the report resource schema file (`QADReportResource.xsd`).

Update Existing Reports. Indicate whether you want to overwrite existing report resources in the system with the imported report resource data.

Update Existing Parameters. Indicate whether you want to overwrite existing report resource parameters in the system with the imported report resource data.

Exporting Report Resources

Use Report Resource Export to export data of specified report resources into `.xml` files. This function lets you back up report resources or create report resource files to be imported into another system.

Use Wildcard. Indicate whether you want to use wildcards to select report resources for export.

Yes: You can use period (.) to represent any single character and asterisk (*) to represent any number of characters in the report code to select multiple report resources.

No: You enter the first and last in a range of report codes for selecting multiple report resources for export.

Code. If Use Wildcard is Yes, enter the report code of the report resource you want to export. You can use period (.) to represent any single character and asterisk (*) to represent any number of characters in the report code to select multiple report resources.

If Use Wildcard is No, enter the first in a range of report codes for selecting multiple report resources for export.

To. Enter the last in a range of report codes for selecting multiple report resources for export. Leave blank to include all codes through the last.

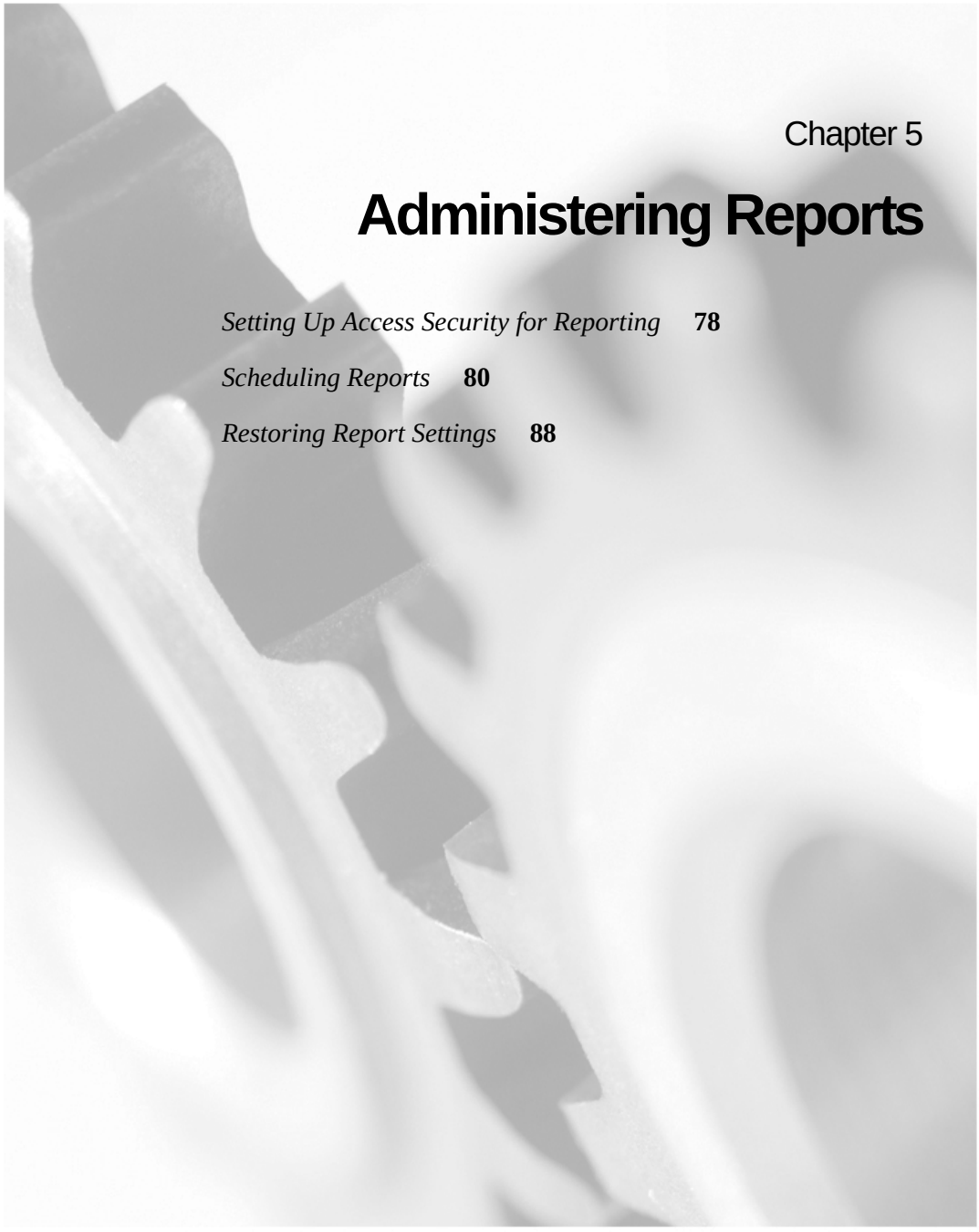
Overwrite Data File. Indicate whether to overwrite existing report resource data files in the export directory.

Export Directory. Enter your log-in directory to export the report resource data files to. Blank means the current directory you are in.

Section 3

Report Administration

This section covers setup and administration of reporting functions and is intended for use by the report administrator.



Chapter 5

Administering Reports

Setting Up Access Security for Reporting

Access must be controlled to reporting programs as well as report resources (associated with menu items) so that only authorized persons can gain access to and manipulate reporting data.

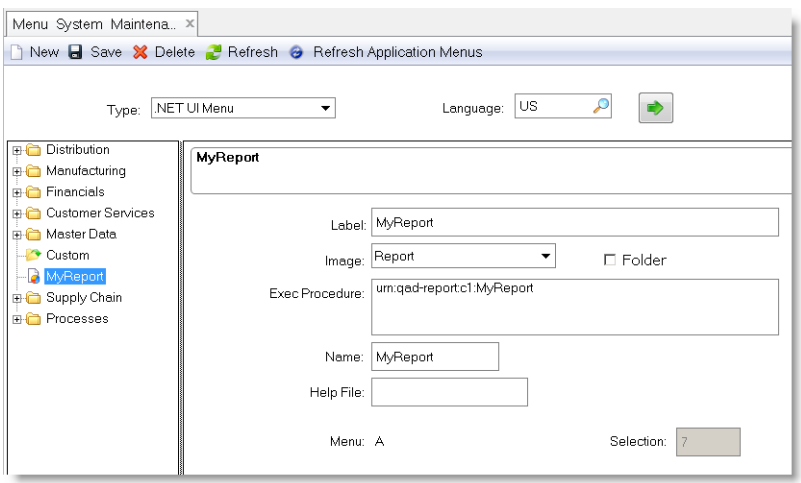
You use the system's role-based access security mechanism to control access to reporting resources in the same way as you do with other menu-level programs in QAD Enterprise Applications.

Setting Up Report Resource Menu Item and Security

Users open reports through report menu items that they have been given access to based on their role membership. Use Menu System Maintenance to create a menu item to provide access to a report resource.

1. Go to Menu System Maintenance (36.4.4).
2. Specify .NET UI Menu in the Type field and select US in the Language field; then press Enter.
3. In the menu structure frame, right-click on a menu item under which you want to create the report menu item and then choose New from the shortcut menu.
4. Enter the detailed information for the report menu item.

Fig. 5.1
Menu System Maintenance



Label. Enter a name for the report menu item. It does not have to be the same as the report code.

Image. Select Report from the list.

Exec Procedure. Enter a component-based activity specified in the form of a uniform resource name (URN):

```
urn:qad-report:c1:ReportCode
```

Where *ReportCode* is the report code of the report resource you are creating a menu item for.

Name. Optionally, enter a menu name.

5 Click Save on the toolbar to save your changes.

6 Click Refresh Application Menus on the toolbar. When refresh is complete, the new menu displays in the menu tree.

7 Set up security for the report menu item. For details on how to set up security, see *User Guide: QAD Security and Controls*.

Setting Up the rptAdmin and rptDsgn Roles

You must create two roles—rptAdmin and rptDsgn—in Role Create for the report administrator and report designer/developer respectively, and then assign them to the particular user IDs you would like to perform the associated activities. These roles add another layer of security that controls access to some activities within the programs. Since the activity-level controls are hard-coded in the reporting programs, you will not be able to perform certain activities within these programs if you create roles with other names for the report administrator and report designer/developer.

Note Make sure you use the correct capitalization for the roles.

You must grant the rptAdmin and rptDsgn roles access to the following programs based on this table:

Table 5.1 Program-Role Access Control Matrix

Reporting Program	rptAdmin	rptDsgn
Report Designer	Allow	Allow
Template Designer	Allow	Allow

Reporting Program	rptAdmin	rptDsgn
Report Resource Import	Allow	Allow
Report Resource Export	Allow	Allow
Scheduled Report Maintenance	Allow	
Report Resource Maintenance	Allow	Allow
Report Parameter Maintenance	Allow	Allow
Personal User Filter Maintenance	Allow	Allow
Admin User Filter Maintenance	Allow	
Report Settings Restore	Allow	

For details about common access security features, see *User Guide: QAD Security and Controls.*

Scheduling Reports

You can automate the process of generating routine reports by scheduling them to automatically run at specified times or intervals and have the reports sent to a specified destination, such as a printer or the document service on the report server.

To schedule reports to run at a specified time or interval, on the report server, you create a Windows scheduled task for a batch and group the reports in the batch.

The Windows Task Scheduler should be configured to launch the Report Batch Processor, which is a non-GUI instance of the QAD .NET UI launched from the command line, for a specific batch as scheduled and runs all the scheduled reports grouped in the batch. If already set up, the report outputs are sent to the QAD .NET UI document service and/or server-side printer as configured.

There can be multiple report servers, and one batch ID is also allowed to be configured on more than one report server. The Report Batch Processor coordinates the processing of scheduled reports with different priorities in the correct sequence across multiple report servers.

Scheduled reports have the following additional features compared to reports run in Report Viewer:

- The output file (PDF/Excel) of a scheduled report can be uploaded to the document service so that the user can view it in the QAD .NET UI later.
- E-mail notifications, including SMTP mails and inbox messages embedded in the QAD .NET UI. You should specify e-mail addresses and the inbox user IDs when creating scheduled reports.
- Server-side printing. You can specify the printer that the report server uses to print the output file.
- Alert—a special type of scheduled report. The difference between an alert and a normal scheduled report is that an alert will not have its report rendered if the data query returns no records.
- Scheduled reports are maintained by the administrator from the maintenance program. The administrator controls the running sequence of scheduled reports by modifying their priorities.

Set Up a Scheduled Batch

1. Create a batch in Batch ID Maintenance (36.14.1).
2. On the report server, create a scheduled task for the batch through Windows Task Scheduler.
   a. Create a parameter file to contain command line parameters with fixed values. Use the following `params.pf` file as an example:

```
-silent
-config-name:test
-user:mfg
-password:(blank)
-workspace:Domain1.1000
-report-batch:czs1
-enable:qad.plugin.services
-enable:qad.plugin.reports
-enable:qad.plugin.reportserver
-report-mode:batch
```

   You need to set your own desired values for these parameters:

   - -config-name: The name of the configuration that the report server should log on to. This is the same as the value chosen by an end user from the drop-down list in the login screen of the QAD .NET UI application when run in GUI mode.

- -user: User ID for logging on to the QAD .NET UI system
- -password: Password for logging on to the QAD .NET UI system
- -workspace: The workspace key of the desired workspace to run scheduled reports in. This is important, since any batch queue is specified by a unique combination of domain and batch ID, and the domain that the report server will use is the domain associated with the specified workspace key.
- -report-batch: The batch ID that will be used in conjunction with the domain associated with the specified workspace key to determine the batch of reports to run.

b In the launching script of the report server process, use the parameter file in place of the parameters:

```
QAD.Client.exe -param.url:file:///c:/params.pf
```

If the parameter file is referenced by a URL, you can choose to place the file on the local machine or a report server.

```
QAD.Client.exe -param.url:http://localhost/rpt/params.pf
```

Note The language that translations will be done in is the language of the server user ID specified in the command line arguments—not the language of the user who submits the report. You can use different batch IDs for different languages if necessary and use different report server user IDs for report servers processing different batches to deal with this limitation.

Setting Up E-Mail Notifications

To set up e-mail notifications, add the following entries to `client-session.xml` on the home server:

```
<Configuration>
...
<!-- SMTP server host name -->
<Smtp.Host>SMTPHostname</Smtp.Host>
<!-- SMTP port name -->
<Smtp.Port>SMTPPortNumber</Smtp.Port>
<!-- SMTP from email address -->
<Smtp.From>E-Mail</Smtp.From>
<!-- SMTP username -->
<Smtp.Username>SMTPUsername</Smtp.Username>
<!-- SMTP password -->
<Smtp.Password>SMTPassword</Smtp.Password>
<!-- SMTP use SSL -->
```

```
<Smtp.UseSSL>false</Smtp.UseSSL>
</Configuration>
```

Use the following settings as a reference:

```
<Smtp.Host>smtp.qad.com</Smtp.Host>
<Smtp.Port>25</Smtp.Port>
<Smtp.From>Report Server <joedoe@qad.com></Smtp.From>
<Smtp.Username>admin</Smtp.Username>
<Smtp.Password>123</Smtp.Password>
<Smtp.UseSSL>false</Smtp.UseSSL>
```

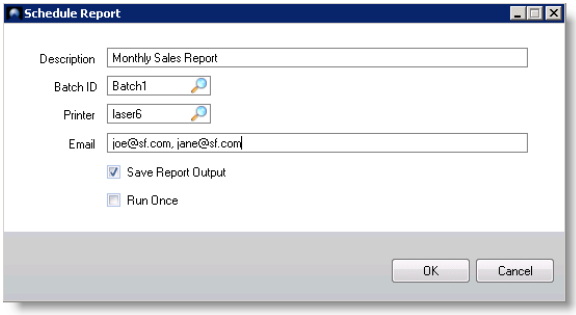
Setting Up a Printer

1. Set up a physical printer on the report server. From the Windows Start menu, select Control Panel|Printers and Faxes|Add a Printer to add a printer.
2. In QAD Enterprise Applications, use Printer Setup Maintenance (36.13.2) to add the printer as an output device so that it is available for use. For details on setting up printers, see *User Guide: QAD System Administration*.

Creating a Scheduled Report

1. Open a report by double-clicking the report menu item in the Applications menu tree or right-clicking it and choosing Open from the shortcut menu. The Report Filter window is displayed in the application area.
2. On the toolbar, click Schedule and then click New.
3. In the Schedule Report dialog box, enter the required information and click OK.

Fig. 5.2
Schedule Report



Batch ID. Specify the batch ID for the scheduled report. The batch ID is created by the administrator in Batch ID Maintenance (36.14.1) and determines when and how often the report will be run on the report server.

Printer. If you want to have the scheduled report printed, specify a printer to send the report to. Printers are set up for the report server by the administrator in Printer Setup Maintenance (36.13.2).

E-Mail. Enter e-mail addresses or Inbox user IDs you want to have scheduled report notifications sent to. Separate multiple entries with commas.

Save Report Output. Select this option if you want the report server to send the scheduled report output file to the document service.

Run Once. Select this option if you want to mark the scheduled report as non-permanent. A non-permanent scheduled report will run only once with the next batch run, regardless how many times the associated batch is scheduled to run. A permanent scheduled report will run every time the batch is run.

4 A confirmation message appears. The report is successfully scheduled.

Viewing Scheduled Reports

To view details of already scheduled reports, do one of the following:

- In Report Viewer, click Schedule on the toolbar and then click View Schedule.

- Type Scheduled Report Browse in the menu search field and press Enter.

The browse displays detailed information of scheduled reports. You can modify a scheduled report by right-clicking it and then clicking Scheduled Report Maintenance.

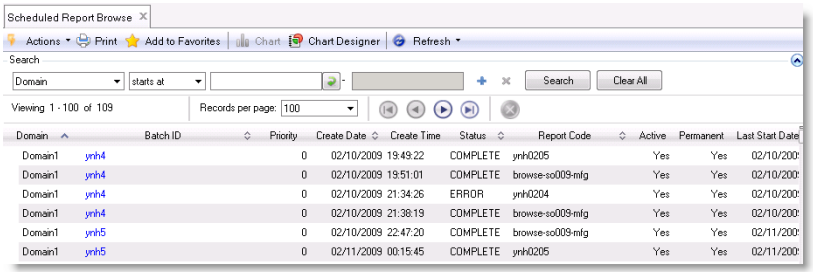


Fig. 5.3
Schedule Report Browse

Key Fields Descriptions

Create Date. The date on which the scheduled report was created.

Create Time. The time when the scheduled report was created.

Status. Specifies the status of the scheduled report.

- New: The scheduled report is newly created and has never run.
- Waiting: The scheduled report is ready to run in the next batch run.
- Running: The scheduled report is running.
- Completed: The scheduled report has run with no errors.
- Error: The scheduled report has run with errors.

Report Code. The report resource code associated with the current scheduled report.

Active. Indicates whether the scheduled report is currently active. Inactive reports will not be run.

Permanent. Indicates whether the scheduled report will run with every batch run or just once.

- Yes: The scheduled report will always run with every batch run.
- No: The report will only run once.

Last Start Date. The date on which the last run started.

Last Start Time. The time when the last run started.

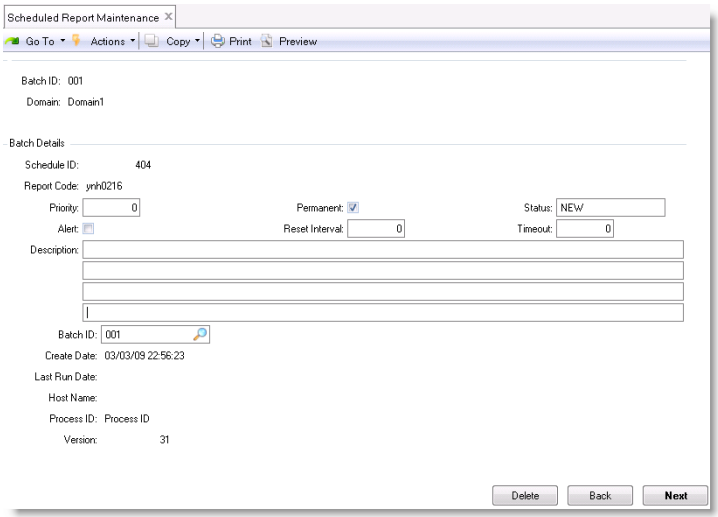
Last End Date. The date on which the last run ended.

Last End Time. The time when the last run ended.

Maintaining Scheduled Reports

To maintain a scheduled report, in Scheduled Report Browse, right-click the batch ID and then click Scheduled Report Maintenance.

Fig. 5.4
Schedule Report Maintenance



Key Fields Descriptions

Schedule ID. This is a system-assigned number that uniquely identifies a scheduled report within the current batch.

Report Code. The report resource code of the current scheduled report.

Priority. Specify an integer number that indicates the priority of the scheduled report. Scheduled reports with greater numbers have higher priorities and will run prior to lower-priority reports in the same batch.

Permanent. Specify whether the scheduled report will run with every batch run or just once.

- Yes: The scheduled report will always run with every batch run.
- No: The report will only run once.

Status. Displays the status of the current scheduled report.

- New: This is a newly created scheduled report.
- Waiting: The scheduled report is ready to run in the next batch run.
- Running: The scheduled report is currently running.
- Completed: The scheduled report has run with no errors.
- Error: The scheduled report has run with errors.

You can type in another status to manually change the current status of the scheduled report. This is useful on such occasions as when you have killed the batch run process on the report server but the status of scheduled report still shows Running.

Alert. This field is currently not implemented.

Reset Interval. This field is currently not implemented.

Timeout. This field is currently not implemented.

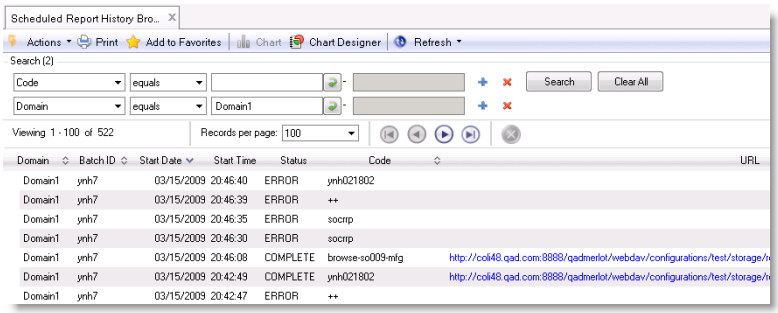
Batch ID. Displays the batch ID the current scheduled report pertains to. You can specify another batch ID if you want to move the scheduled report to that batch.

Viewing Scheduled Reports History

To view the scheduled reports history, do one of the following:

- In Report Viewer, click Schedule on the toolbar and then click View History.
- Type Scheduled Report History Browse in the menu search field and press Enter.

Fig. 5.5
Schedule Report History



Restoring Report Settings

After you make changes to report settings in Report Parameter Maintenance (36.4.21.3), you can change them back to the default settings. To do this, go to Report Settings Restore (36.4.21.23) and set Restore to Yes.

Section 4

Using Reports

This section covers day-to-day use of reporting functions and is intended for use by the report end user.

Chapter 6

Running Reports

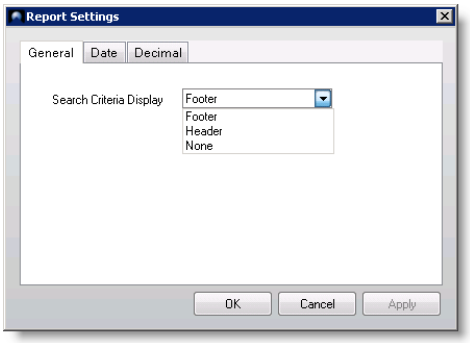
Configuring Report Settings

Use Report Settings to customize how certain elements of data will be displayed in the rendered report.

In the Filter screen, click Settings on the toolbar to bring up the Report Settings dialog box.

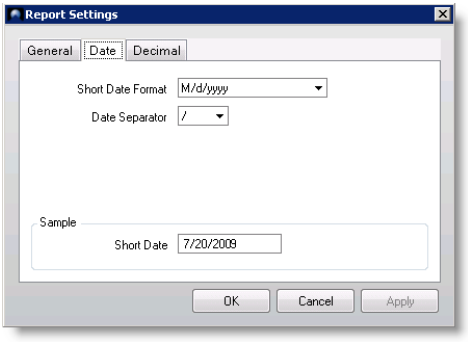
- Under the General tab, specify whether to display search criteria in the report, and if yes, whether to display this information in the report header or footer.

Fig. 6.1
Report Settings: General



- Under the Date tab, select a format for the dates to be displayed in the report and specify a date separator. You can see a sample of the date format you specify at the bottom of the dialog box.

Fig. 6.2
Report Settings: Date



- Under the Decimal tab, specify how numbers will be displayed in the report, including decimal separator, decimal digits, grouping separator, and grouping format. A sample number is displayed at the bottom of the dialog box.

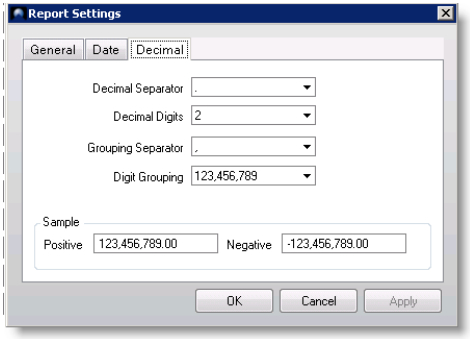


Fig. 6.3
Report Settings: Number

Note After you manually format a date or number field in a report in Report Designer, the reporting format settings will no longer apply to these fields.

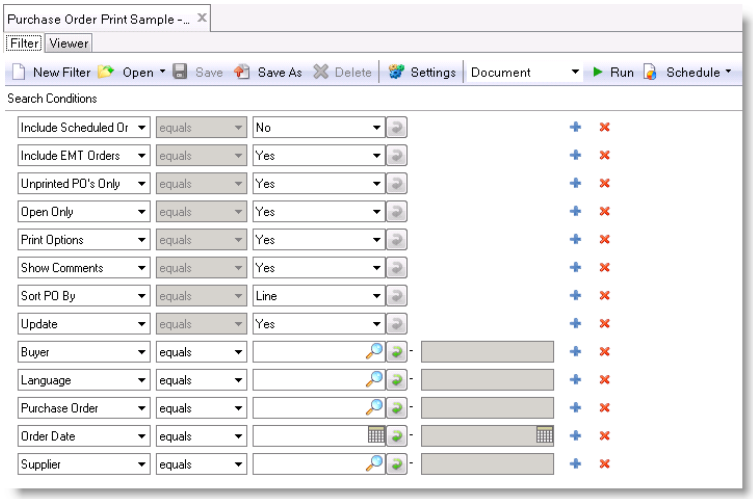
Running Reports

After a report is designed, you can set filter criteria to filter data in the report, run the report, and send it to different output destinations.

To run a report

1 Double-click the report menu item in the Applications menu tree or right-click it and choose Open from the shortcut menu. The Report Filter screen is displayed in the application area.

Fig. 6.4
Report Filter



2 By default, a report will display all the records available in the source data. However, you may want to retrieve just a certain range of records in the report; for example, sales records between last September and this March. You do this by setting search conditions to filter data in the report. You can also use filters to load existing search conditions. For information about using filters, see “Using Report Filters” on page 98.

Note The default report filter parameters are predefined in the data source proxy program and you can only change them by modifying the program code.

To set search conditions

The query constructor provides extensive, configurable filter capabilities that let you create both simple and complex queries. Choose a search operator from the drop-down list.

a The search operators include the following:

- equals

- not equals
- contains
- range
- starts at (the default)
- greater than
- less than
- is null
- is not null

b If you choose the Range operator, enter a beginning value of the range in the first search box. Optionally, enter an ending value of the range in the second search box.

c To refine your search further, click the plus (+) icon to add another search row. You can add as many rows as needed, each with different search values and operators. When you specify several criteria, note that multiple criteria for the same field are treated as a logical AND condition.

d To remove a search criteria row, click on the delete (X) icon.

e Optionally, save the new search conditions as a filter for future reuse. For information about working with filters, see “Using Report Filters” on page 98.

3 On the toolbar, select an output format from the list next to the Settings button. You can choose from three output formats when the report is run:

- Document: The report is displayed in the Report Viewer window.
- Excel: The report is generated in Microsoft Excel format. You can save the file and open it in the Report Viewer window.
- PDF: The report is generated in PDF format. You can save the file and open it in the Report Viewer window.

4 On the toolbar, click Run. A report generation progress bar appears. When report generation is complete, the report is displayed in the Report Viewer window directly or opened as a PDF or Excel file depending on which output format you selected.

Running Reports Directly From Browses

You can directly run reports from browses by selecting Report from the Action menu in the browse screen. The sorting, grouping, and search criteria in the browse are all carried over to the report, which uses the browse as its data source. You can further filter data in the report by defining new search criteria in the Filter screen. This is an alternative to generating a browse-based report by creating a report resource and manually specifying a browse as its data source. However, since you do not create a report resource, the browse-based report created this way cannot be assigned to a menu. A built-in template is used to render browse-based reports.

Note Unlike in the browse, the column header by which data is grouped is case-sensitive in the report. For example, while EA and ea are considered the same column header in the browse, they are treated as different field names with data separately grouped under them in the report.

Viewing, Exporting, and Printing a Report

In Report Viewer, use the toolbar buttons to navigate through the report and perform other functions such as saving and printing.

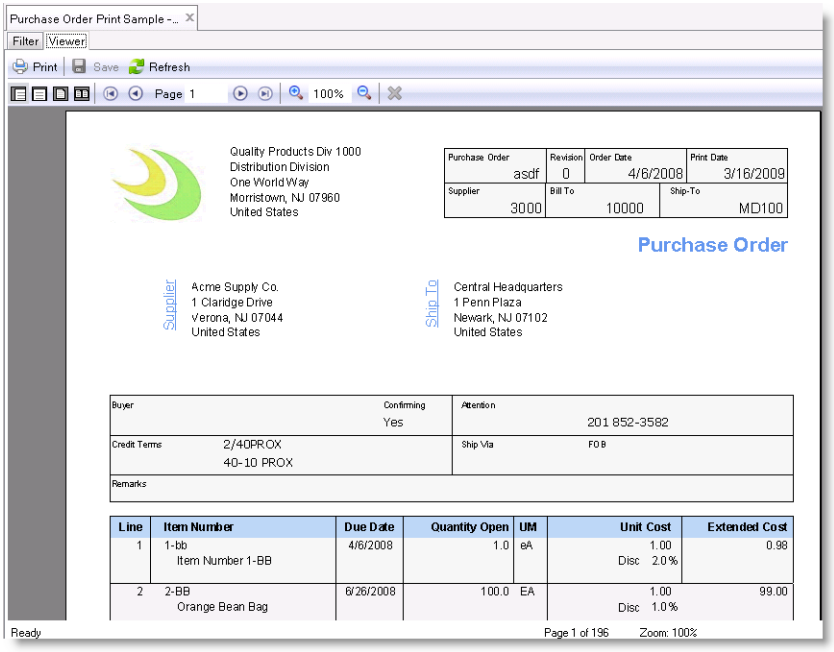


Fig. 6.5
Report Viewer

Button	Name	Description
	Print	Send the report to a printer.
	Save	Save the report to a specified location.
	Refresh	Regenerate the report using your last setting.
	Actual Size	Display the report in its actual size.
	Page Width	Fit the report to the width of the Report Viewer window.
	One Page	Display the report in a one-page view in the Report Viewer window.
	Two Pages	Display the report in a side-by-side two-page view in the Report Viewer window.
	First Page	Jump to the first page.
	Previous Page	Go to the previous page.

Button	Name	Description
	Next Page	Go to the next page.
	Last Page	Jump to the last page.
	Zoom In	Magnify the report preview size.
	Size	Specify the exact report preview size.
	Zoom Out	Decrease the report review size.
	Cancel Rending	Cancel rendering the report.

Using Report Filters

If a report always contains a certain range of data and is exported to a certain format, you do not have to define the filter criteria and output settings every time you generate the report. You can save the search conditions and output settings as a filter and open it to load the same set of configurations when you run the report later.

A filter is a personalized set of search conditions and settings, which means that the filters you created can only be accessed and managed by you and the administrator, and no one else.

Creating a New Filter

1. In the Filter window, click New Filter on the toolbar. All the search conditions are reset to the default values.
2. Change the filter criteria.
3. On the toolbar, click Save As.
4. In the Save As dialog box, enter a unique filter name, and optionally, a brief description; then click OK.
5. The filter is created. If you make further changes to the search conditions, click Save on the toolbar to save the changes.

Loading an Existing Filter

1. On the toolbar, click Open and then select an existing filter from the list.
2. If the list is too long, click More to choose the filter from a browse window.
3. After you select an existing filter, its search conditions and settings are loaded in the Filter window.
4. If you want to save any changes to the loaded filter, click Save on the toolbar.

Maintaining Your Own Filters

Filters are user-specific. Use Personal User Filter Maintenance (36.4.21.14) to maintain your own filters.

System. Specify whether or not the filter is system-defined.

User. View or enter the user for whom to define the filter. When the filter is system-defined, this field is disabled.

Filter. Enter a filter name.

Description. Enter the description of the filter.

Default. Indicates whether this is the default filter in Report Viewer.

Param Name. Enter a parameter name in the filter criteria. If the parameter name already exists, you can either create a new parameter with the same name or update the existing one.

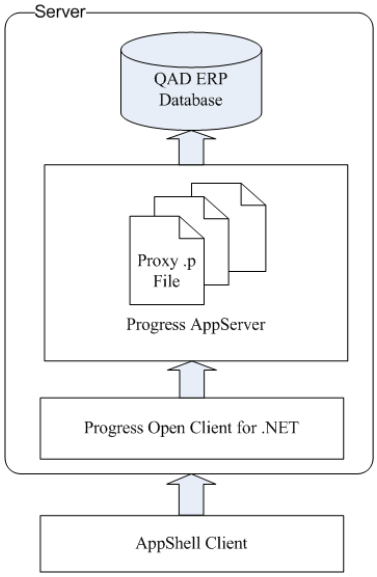
Appendix A

Implementing the Generic Proxy

Overview

A generic proxy is a Progress .p program on the report server that serves as the data source provider that retrieves data from the database, constructs datasets, and passes them to Report Designer or Report Viewer through Progress Open Client for .NET to generate reports.

Fig. 6.6
Generic Proxy



Implementing the generic proxy entails the following two steps:

1. Develop the generic proxy code.
2. Deploy the generic proxy on the report server.

Developing the Generic Proxy Code

A generic proxy file comprises three blocks of code:

- Temp-table definition that define the dataset structure
- Metadata definition that defines the metadata in Report Designer
- Data retrieving logic that populates data in temp-tables

Defining Dataset

Use the following sample program to develop code to define the temp-table for returning dataset.

```
{mfdeclre.i}
{gplabel.i}

{com/qad/shell/report/dsReportRequest.i}
{com/qad/shell/report/ReportConstants.i}

/* Temp-table definiation block */
/* TODO Report Table Definition */

/* Main Block */
define input parameter runReport as logical.
define input parameter reportHandle as handle.
define input parameter dataset for dsReportRequest.
define output parameter dataset-handle phReportResults.

{com/qad/shell/report/reporting.i}

define variable bufferName as character no-undo.

/* TODO empty temp-table */

for first ttReportRequest no-lock:

   run FillMetaData.

   if runReport then do:
      run RunReport
         (output dataset-handle phReportResults).
   end.

end.

/* Metadata definition block */
/* Metadata */
procedure FillMetaData:

/* TODO other procedures*/

end procedure.

/* Data retrieving logic block */
procedure RunReport:

define output parameter dataset-handle phReportResults.

/* TODO data retrieving logic */

phReportResults = dataset dsReportResults:handle.

end procedure.
```

Note You must use the same structure when writing your code—especially regarding input/output parameters.

Defining Temp-Tables for the Dataset

A dataset may contain multiple temp-tables. In the following example, we create two simple temp-tables with master-detail relationship.

```
define temp-table ttSalesHeader before-table ttSalesHeaderBefore
   field so_nbr like so_mstr.so_nbr
   field so_cust like so_mstr.so_cust
   field so_ord_date like so_mstr.so_ord_date
   field sales_order_slspsn1 like so_mstr.so_slspsn[1]
   field sales_order_slspsn2 like so_mstr.so_slspsn[2]
   field sales_order_slspsn3 like so_mstr.so_slspsn[3]
   field sales_order_slspsn4 like so_mstr.so_slspsn[4]
.

define temp-table ttSoLine before-table ttSoLineBefore
   field sales_order_number like so_mstr.so_nbr
   field sales_detail_line like sod_det.sod_line
   field sales_detail_item like sod_det.sod_part
   field sales_detail_unit_measure like sod_det.sod_um
   field sales_detail_due_date like sod_det.sod_due_date
```

Note If you want to use some fields in the temp-table as search fields, you must use the same field names in the temp-tables as those in the database.

Defining Indexes for Temp-Tables

Create indexes for temp-tables to improve performance.

Example ttSalesHeader is only referenced by so_mstr, so use so_nbr as the primary index; ttSoLine fields are from so_mstr and sod_det, so use so_nbr and sod_line as primary index.

Define temp-table relations in the data retrieving layer to lower system overhead. Since ttSalesHeader and ttSoLine have the master-detail relationship, we define the dataset relation as follows:

```
define dataset dsReportResults for ttSalesHeader, ttSoLine
   data-relation drLine for ttSalesHeader, ttSoLine
   relation-fields (so_nbr, sales_order_number)
```

Here is the complete code that we have written so far to define temp-tables and datasets:

```
/* Temp-table definition block */
```

```
define temp-table ttSalesHeader before-table ttSalesHeaderBefore
   field so_nbr like so_mstr.so_nbr
   field so_cust like so_mstr.so_cust
   field so_ord_date like so_mstr.so_ord_date
   field sales_order_slspsn1 like so_mstr.so_slspsn[1]
   field sales_order_slspsn2 like so_mstr.so_slspsn[2]
   field sales_order_slspsn3 like so_mstr.so_slspsn[3]
   field sales_order_slspsn4 like so_mstr.so_slspsn[4]
   index SalesHeaderIdx is primary so_nbr
.

define temp-table ttSoLine before-table ttSoLineBefore
   field sales_order_number like so_mstr.so_nbr
   field sales_detail_line like sod_det.sod_line
   field sales_detail_item like sod_det.sod_part
   field sales_detail_unit_measure like sod_det.sod_um
   field sales_detail_due_date like sod_det.sod_due_date
   index SoLineIdx is primary sales_order_number
sales_detail_line.
.

define dataset dsReportResults for ttSalesHeader, ttSoLine
   data-relation drLine for ttSalesHeader, ttSoLine
   relation-fields (so_nbr, sales_order_number)
```

Note There must be a dataset named dsReportResults defined for temp-tables for the proxy program to work, even if they have no relations.

Defining Metadata

You define metadata to specify which fields and tables the user can use to design the report. Every table in the metadata needs a buffer name, a buffer header, and fields. With temp-tables already defined, we just need to define metadata for each temp-table.

Defining Buffer Name and Creating BufferHeader

Use the exact temp-table name for the buffer name. Run the CreateBufferHeader procedure to create buffer header for each temp-table.

Table A.1 CreateBuffer Header Parameters

Seq.	Name	Input/ Output	Data Type	Description
1	tableName	Input	Character	Temp-table name
2	tableLabel	Input	Character	Label displayed in Report Designer

Creating Fields for Each Temp-Table

Three predefined procedures can be used to create field metadata: CreateFieldForDBField, CreateFieldLikeDBField, and CreateField.

CreateFieldForDBField

This procedure is used to create metadata for fields in the database using the same table name, field name, and format.

Table A.2 CreateFieldForDB Field Parameters

Seq.	Name	Input/ Output	Data Type	Description
1	bufferName	Input	Character	Temp-table name
2	tableName	Input	Character	QAD ERP database table name
3	fieldName	Input	Character	QAD ERP database field name
4	isSearchField	Input	Logical	Whether this field a search field
5	isReadOnlySearch	Input	Logical	Whether this field is read-only
6	isVisible	Input	Logical	Whether this field is visible in Report Designer
7	isSingleEntry	Input	Logical	Always set this to False
8	isOperatorChangeable	Input	Logical	Whether the operator can be changed
9	isRequiredCondition	Input	Logical	Whether the field is mandatory
10	isEditable	Input	Logical	Whether the field can be edited
11	defaultValue	Input	Character	Default value of the first search field
12	defaultOperator	Input	Character	Default operator of the first search field
13	defaultValueType	Input	Character	Default value type of the first search field
14	defaultValue2	Input	Character	Default value of the second search field
15	defaultValueType2	Input	Character	Default value type of the second search field

CreateFieldLikeDBField

This procedure is used to create metadata for temp-table fields used in temp-table definition.

Table A.3 CreateFieldLikeDB Field Parameters

Seq.	Name	Input/ Output	Data Type	Description
1	bufferName	Input	Character	Temp-table name
2	fName	Input	Character	Field name in the temp-table
3	tableName	Input	Character	QAD ERP database table name
4	fieldName	Input	Character	QAD ERP database field name
5	isSearchField	Input	Logical	Whether this field a search field
6	isReadOnlySearch	Input	Logical	Whether this field is read-only
7	isVisible	Input	Logical	Whether this field is visible in Report Designer
8	isSingleEntry	Input	Logical	Always set this to False
9	isOperatorChangeable	Input	Logical	Whether the operator can be changed
10	isRequiredCondition	Input	Logical	Whether the field is mandatory
11	isEditable	Input	Logical	Whether the field can be edited
12	defaultValue	Input	Character	Default value of the first search field
13	defaultOperator	Input	Character	Default operator of the first search field
14	defaultValueType	Input	Character	Default value type of the first search field
15	defaultValue2	Input	Character	Default value of the second search field
16	defaultValueType2	Input	Character	Default value type of the second search field

Example Create buffer for the ttSalesHeader temp-table:

```
bufferName = "ttSalesHeader".

   run CreateBufferHeader in reportHandle
   (bufferName, "Sales Orders").
```

Example Create metadata for the so_nbr field in the ttSalesHeader temp-table:

```
run CreateFieldLikeDBField in reportHandle
   (bufferName,
    "so_nbr",
    "so_mstr",
    "so_nbr",
    true,
    false,
```

```
 true,
 false,
 true,
 false,
 true,
 "",
 {&ParameterOperator_Equals},
 {&ParameterValueType_Constant},
 "",
 {&ParameterValueType_Constant}).
```

Note The so_nbr field is set as a search field, so the fifth parameter is True.

Create metadata for all the other fields in ttSalesHeader or ttSoLine:

```
/* Metadata definition block */
/* Metadata */
procedure FillMetaData:

   /* Create buffer header for ttSalesHeader */
   bufferName = "ttSalesHeader".

   run CreateBufferHeader in reportHandle
   (bufferName, "Sales Orders").

   /* Create field for ttSalesHeader */

  run CreateFieldLikeDBField in reportHandle
   (bufferName,
    "so_nbr",
    "so_mstr",
    "so_nbr",
    true,
    false,
    true,
    false,
    true,
    false,
    true,
    "",
    {&ParameterOperator_Equals},
    {&ParameterValueType_Constant},
    "",
    {&ParameterValueType_Constant}).

   run CreateFieldLikeDBField in reportHandle
   (bufferName,
    "so_cust",
    "so_mstr",
    "so_cust",
    true,
    false,
    true,
    false,
    true,
    false,
```

```
 true,
 "",
 {&ParameterOperator_Equals},
 {&ParameterValueType_Constant},
 "",
 {&ParameterValueType_Constant}).

run CreateFieldLikeDBField in reportHandle
(bufferName,
 "so_ord_date",
 "so_mstr",
 "so_ord_date",
 true,
 false,
 true,
 false,
 true,
 false,
 true,
 "",
 {&ParameterOperator_Equals},
 {&ParameterValueType_Constant},
 "",
 {&ParameterValueType_Constant}).

run CreateFieldLikeDBField in reportHandle
(bufferName,
 "sales_order_slspsn1",
 "so_mstr",
 "so_slspsn",
 false,
 false,
 true,
 false,
 true,
 false,
 true,
 "",
 {&ParameterOperator_Equals},
 {&ParameterValueType_Constant},
 "",
 {&ParameterValueType_Constant}).

run CreateFieldLikeDBField in reportHandle
(bufferName,
 "sales_order_slspsn2",
 "so_mstr",
 "so_slspsn",
 false,
 false,
 true,
 false,
 true,
 false,
 true,
 "",
 {&ParameterOperator_Equals},
 {&ParameterValueType_Constant},
```

```
 "",
 {&ParameterValueType_Constant}).

run CreateFieldLikeDBField in reportHandle
(bufferName,
 "sales_order_slspsn3",
 "so_mstr",
 "so_slspsn",
 false,
 false,
 true,
 false,
 true,
 false,
 true,
 "",
 {&ParameterOperator_Equals},
 {&ParameterValueType_Constant},
 "",
 {&ParameterValueType_Constant}).

run CreateFieldLikeDBField in reportHandle
(bufferName,
 "sales_order_slspsn4",
 "so_mstr",
 "so_slspsn",
 false,
 false,
 true,
 false,
 true,
 false,
 true,
 "",
 {&ParameterOperator_Equals},
 {&ParameterValueType_Constant},
 "",
 {&ParameterValueType_Constant}).


/* Create buffer header for ttSoLine */
bufferName = "ttSoLine".

run CreateBufferHeader in reportHandle
(bufferName, "Sales Order Lines").

/* Create field for ttSoLine */
run CreateFieldLikeDBField in reportHandle
(bufferName,
 "sales_order_number",
 "so_mstr",
 "so_nbr",
 false,
 false,
 true,
 false,
 true,
 false,
```

```
 true,
 "",
 {&ParameterOperator_Equals},
 {&ParameterValueType_Constant},
 "",
 {&ParameterValueType_Constant}).

run CreateFieldLikeDBField in reportHandle
(bufferName,
 "sales_detail_line",
 "sod_det",
 "sod_line",
 false,
 false,
 true,
 false,
 true,
 false,
 true,
 "",
 {&ParameterOperator_Equals},
 {&ParameterValueType_Constant},
 "",
 {&ParameterValueType_Constant}).

run CreateFieldLikeDBField in reportHandle
(bufferName,
 "sales_detail_item",
 "sod_det",
 "sod_part",
 false,
 false,
 true,
 false,
 true,
 false,
 true,
 "",
 {&ParameterOperator_Equals},
 {&ParameterValueType_Constant},
 "",
 {&ParameterValueType_Constant}).

run CreateFieldLikeDBField in reportHandle
(bufferName,
 "sales_detail_unit_measure",
 "sod_det",
 "sod_um",
 false,
 false,
 true,
 false,
 true,
 false,
 true,
 "",
 {&ParameterOperator_Equals},
 {&ParameterValueType_Constant},
```

```
        "",
        {&ParameterValueType_Constant}).

      run CreateFieldLikeDBField in reportHandle
      (bufferName,
       "sales_detail_due_date",
       "sod_det",
       "sod_due_date",
       false,
       false,
       true,
       false,
       true,
       false,
       true,
       "",
       {&ParameterOperator_Equals},
       {&ParameterValueType_Constant},
       "",
       {&ParameterValueType_Constant}).

  end procedure.
```

Writing Report Data Retrieving Logic

With data structure and metadata defined, we can now write a simple procedure to populate temp-tables with data. Use a dynamic query if there are any search fields. The data retrieving logic should be coded into the RunReport procedure.

Defining Dynamic Query and Search Field

Search fields all reside in the so_mstr table.

```
define variable queryString as character no-undo.
define variable hSOQuery as handle.
define query SOQuery for so_mstr.

hSOQuery = query SOQuery:handle.

queryString = "for each so_mstr no-lock where true ".

run FillQueryStringVariable in reportHandle (input
"ttSalesHeader", input "so_nbr", input-output queryString).
run FillQueryStringVariable in reportHandle (input
"ttSalesHeader", input "so_cust", input-output queryString).
run FillQueryStringVariable in reportHandle (input
"ttSalesHeader", input "so_ord_date", input-output queryString).

queryString = queryString + ":".

hSOQuery:query-prepare(queryString).
hSOQuery:query-open().
```

```
hSOQuery:get-next().
```

Note The FillQueryStringVariable function will get the parameters sent by AppShell, what search fields are used and what value is typed, and then construct the dynamic query string for search fields at run time.

Seq.	Name	Input/ Output	Data Type	Description
1	bufferName	Input	Character	Temp-table name
2	fieldName	Input	Character	Field name in the temp-table
3	queryString	Input-Output	Character	Dynamic query string

Note For the function to work, the temp-table fields defined as search fields in the metadata definition should use exactly the same names as those in the database.

Retrieving Data

Loop the so_mstr to create the ttSalesHeader temp-table and retrieve detailed information from each so_mstr. This is the master-detail data retrieving logic in its simplest form.

```
repeat while not hSOQuery:query-off-end:
      create ttSalesHeader.
      assign
         ttSalesHeader.so_nbr               = so_mstr.so_nbr
         ttSalesHeader.so_cust              = so_mstr.so_cust
        ttSalesHeader.so_ord_date          = so_mstr.so_ord_date
       ttSalesHeader.sales_order_slspsn1  = so_mstr.so_slspsn[1]
       ttSalesHeader.sales_order_slspsn2  = so_mstr.so_slspsn[2]
         ttSalesHeader.sales_order_slspsn3  =
so_mstr.so_slspsn[3].
         ttSalesHeader.sales_order_slspsn4  =
so_mstr.so_slspsn[4].

      for each sod_det no-lock
where sod_det.sod_nbr = so_mstr.so_nbr:
create ttSoLine.
assign
   ttSoLine.sales_order_number        = sod_det.sod_nbr
   ttSoLine.sales_detail_line         = sod_det.sod_line
   ttSoLine.sales_detail_item         = sod_det.sod_part
           ttSoLine.sales_detail_unit_measure = sod_det.sod_um
   ttSoLine.sales_detail_due_date     = sod_det.sod_due_date.
      end.

      hSOQuery:get-next().
   end. /* Repeat query */
```

Important Things to Note

Parameters

You should use the four parameters (three input parameters, one output parameter), and four parameters only; otherwise, the proxy will not work.

- Use runReport to specify whether to run the data retrieving logic.
- Use reportHandle to handle the `.p` file that contains all the predefined functions and procedures.
- dsReportRequest is the dataset passed by AppShell containing information we need to run the report.
- phReportResults is the handle of the dataset we defined within the proxy containing the real data that retrieved from database. This handle should be assigned within the main block.

Put the code in the main block:

```
define input parameter runReport as logical.
define input parameter reportHandle as handle.
define input parameter dataset for dsReportRequest.
define output parameter dataset-handle phReportResults.
```

Include Files

Always include these .i files in the proxy.

```
{mfdeclre.i}
{gplabel.i}

{com/qad/shell/report/dsReportRequest.i}
{com/qad/shell/report/ReportConstants.i}
{com/qad/shell/report/reporting.i}
```

Empty Temp-Tables

Always empty temp-tables before executing the data retrieving logic to ensure there is no residual data.

```
/* Empty temp-table */
empty temp-table ttSalesHeader no-error.
empty temp-table ttSoLine      no-error.
```

Get the Request Dataset

ttReportRequest is the temp-table name already defined in the reporting framework. It contains the information that AppShell passes to this `.p` file. So the `for first` clause is needed and the temp-table name should not be changed.

```
for first ttReportRequest no-lock:
   /* Define meta logic */
   /* Retrieve data logic */
end.
```

Use the Data Retrieving Logic as a Switch

The runReport parameter should be used as a switch. It is passed from AppShell to the proxy indicating whether or not to retrieve data in the proxy since sometimes AppShell only needs metadata. Use this switch to optimize performance.

```
if runReport then do:
      run RunReport
         (output dataset-handle phReportResults).
   end.
```

Specify Financials Lookups as the Proxy Data Source

When writing programs for the proxy data source, you can specify Financials lookups in the metadata as opposed to standard browse lookups. To do this, specify the following for the lookup name in the metadata: *<lookup provider type>:<lookupID>:<lookup return field>:<lookup filter field>*.

<lookup return field> and *<lookup filter field>* are optional. In the example above, the two fields are the same as the calling field in the browse whose lookup button is pressed.

Here is an example for the Financials lookup:

```
BaseLibrary.Lookup.BLFLookupProvider:BJournalSAO.Selec
tJournal:tcJournalCode:tJournal.JournalCode
```

Deploying the Generic Proxy

After the proxy program is developed, deploy it onto the report server. Compile the proxy code and put the `.r` or `.p` file in the specified directory so that Appserver can run this file.

Both the `.p` and `.r` files are needed when the server runs the code. Compile the `sosorp_Finished.p` file.

1. Connect to the QAD ERP production database (qaddb) and administration database (qadadm).
2. Compile the proxy program, adding the following to the Propath:
   - Proxy program directory: `<desktop source code directory>/com/qad/shell/report/reports`

     Where `<desktop source code directory>` usually is

     `/qad/web/server/docs/<ENVname>/ebdesktop2/<WEBAPPNAME>/`
   - QAD ERP installation directory.

Here is a Propath sample:

```
propath = propath + "," +
"/qad/web/server/docs/93/ebdesktop2/dev93ui/com/qad/shell/repor
t/reports".

propath = propath + "," +
"/qad/web/server/docs/93/ebdesktop2/dev93ui".

propath = propath + "," +
"/qad/web/server/docs/93/ebdesktop2/dev93ui/com/qad/shell/repor
t".

propath = propath + "," +
"/qad/mfgpro/93/stage".
```

Compile `com/qad/shell/report/reports/TestProxy.p` and save.

Complete Proxy Program Sample Code

```
{mfdeclre.i}
{gplabel.i}

{com/qad/shell/report/dsReportRequest.i}
{com/qad/shell/report/ReportConstants.i}

/*=============================================================
```

```
===============*/
/* ***************************  TEMP TABLES
****************************** */
/*=============================================================
===============*/

/* Temp-table definiation block */

define temp-table ttSalesHeader before-table ttSalesHeaderBefore
   field so_nbr like so_mstr.so_nbr
   field so_cust like so_mstr.so_cust
   field so_ord_date like so_mstr.so_ord_date
   field sales_order_slspsn1 like so_mstr.so_slspsn[1]
   field sales_order_slspsn2 like so_mstr.so_slspsn[2]
   field sales_order_slspsn3 like so_mstr.so_slspsn[3]
   field sales_order_slspsn4 like so_mstr.so_slspsn[4]
   index SalesHeaderIdx is primary so_nbr
.

define temp-table ttSoLine before-table ttSoLineBefore
   field sales_order_number like so_mstr.so_nbr
   field sales_detail_line like sod_det.sod_line
   field sales_detail_item like sod_det.sod_part
   field sales_detail_unit_measure like sod_det.sod_um
   field sales_detail_due_date like sod_det.sod_due_date
   index SoLineIdx is primary sales_order_number
sales_detail_line.
.

define dataset dsReportResults for ttSalesHeader, ttSoLine
   data-relation drLine for ttSalesHeader, ttSoLine
   relation-fields (so_nbr, sales_order_number)
.

/*=============================================================
===============*/
/* ***************************  MAIN BLOCK
****************************** */
/*=============================================================
===============*/
define input parameter runReport as logical.
define input parameter reportHandle as handle.
define input parameter dataset for dsReportRequest.
define output parameter dataset-handle phReportResults.

{com/qad/shell/report/reporting.i}

define variable bufferName as character no-undo.

empty temp-table ttSalesHeader no-error.
empty temp-table ttSoLine      no-error.

for first ttReportRequest no-lock:

   run FillMetaData.

   if runReport then do:
      run RunReport
```

```
            (output dataset-handle phReportResults).
      end.

   end.

   /* Metadata definition block */
   /* Metadata */
   procedure FillMetaData:

      /* Create buffer header for ttSalesHeader */
      bufferName = "ttSalesHeader".

      run CreateBufferHeader in reportHandle
      (bufferName, "Sales Orders").

      /* Create field for ttSalesHeader */

     run CreateFieldLikeDBField in reportHandle
      (bufferName,
       "so_nbr",
       "so_mstr",
       "so_nbr",
       true,
       false,
       true,
       false,
       true,
       false,
       true,
       "",
       {&ParameterOperator_Equals},
       {&ParameterValueType_Constant},
       "",
       {&ParameterValueType_Constant}).

      run CreateFieldLikeDBField in reportHandle
      (bufferName,
       "so_cust",
       "so_mstr",
       "so_cust",
       true,
       false,
       true,
       false,
       true,
       false,
       true,
       "",
       {&ParameterOperator_Equals},
       {&ParameterValueType_Constant},
       "",
       {&ParameterValueType_Constant}).

      run CreateFieldLikeDBField in reportHandle
      (bufferName,
       "so_ord_date",
       "so_mstr",
       "so_ord_date",
```

```
 true,
 false,
 true,
 false,
 true,
 false,
 true,
 "",
 {&ParameterOperator_Equals},
 {&ParameterValueType_Constant},
 "",
 {&ParameterValueType_Constant}).

run CreateFieldLikeDBField in reportHandle
(bufferName,
 "sales_order_slspsn1",
 "so_mstr",
 "so_slspsn",
 false,
 false,
 true,
 false,
 true,
 false,
 true,
 "",
 {&ParameterOperator_Equals},
 {&ParameterValueType_Constant},
 "",
 {&ParameterValueType_Constant}).

run CreateFieldLikeDBField in reportHandle
(bufferName,
 "sales_order_slspsn2",
 "so_mstr",
 "so_slspsn",
 false,
 false,
 true,
 false,
 true,
 false,
 true,
 "",
 {&ParameterOperator_Equals},
 {&ParameterValueType_Constant},
 "",
 {&ParameterValueType_Constant}).

run CreateFieldLikeDBField in reportHandle
(bufferName,
 "sales_order_slspsn3",
 "so_mstr",
 "so_slspsn",
 false,
 false,
 true,
 false,
```

```
 true,
 false,
 true,
 "",
 {&ParameterOperator_Equals},
 {&ParameterValueType_Constant},
 "",
 {&ParameterValueType_Constant}).

run CreateFieldLikeDBField in reportHandle
(bufferName,
 "sales_order_slspsn4",
 "so_mstr",
 "so_slspsn",
 false,
 false,
 true,
 false,
 true,
 false,
 true,
 "",
 {&ParameterOperator_Equals},
 {&ParameterValueType_Constant},
 "",
 {&ParameterValueType_Constant}).


/* Create buffer header for ttSoLine */
bufferName = "ttSoLine".

run CreateBufferHeader in reportHandle
(bufferName, "Sales Order Lines").

/* Create field for ttSoLine */
run CreateFieldLikeDBField in reportHandle
(bufferName,
 "sales_order_number",
 "so_mstr",
 "so_nbr",
 false,
 false,
 true,
 false,
 true,
 false,
 true,
 "",
 {&ParameterOperator_Equals},
 {&ParameterValueType_Constant},
 "",
 {&ParameterValueType_Constant}).

run CreateFieldLikeDBField in reportHandle
(bufferName,
 "sales_detail_line",
 "sod_det",
 "sod_line",
```

```
 false,
 false,
 true,
 false,
 true,
 false,
 true,
 "",
 {&ParameterOperator_Equals},
 {&ParameterValueType_Constant},
 "",
 {&ParameterValueType_Constant}).

run CreateFieldLikeDBField in reportHandle
(bufferName,
 "sales_detail_item",
 "sod_det",
 "sod_part",
 false,
 false,
 true,
 false,
 true,
 false,
 true,
 "",
 {&ParameterOperator_Equals},
 {&ParameterValueType_Constant},
 "",
 {&ParameterValueType_Constant}).

run CreateFieldLikeDBField in reportHandle
(bufferName,
 "sales_detail_unit_measure",
 "sod_det",
 "sod_um",
 false,
 false,
 true,
 false,
 true,
 false,
 true,
 "",
 {&ParameterOperator_Equals},
 {&ParameterValueType_Constant},
 "",
 {&ParameterValueType_Constant}).

run CreateFieldLikeDBField in reportHandle
(bufferName,
 "sales_detail_due_date",
 "sod_det",
 "sod_due_date",
 false,
 false,
 true,
 false,
```

```
    true,
    false,
    true,
    "",
    {&ParameterOperator_Equals},
    {&ParameterValueType_Constant},
    "",
    {&ParameterValueType_Constant}).

end procedure.

/* Data retrieving logic block */
procedure RunReport:

   define output parameter dataset-handle phReportResults.

   /* Retrieve the data from database */
   define variable queryString as character no-undo.
   define variable hSOQuery as handle.
   define query SOQuery for so_mstr.

   hSOQuery = query SOQuery:handle.

   queryString = "for each so_mstr no-lock where true ".

   run FillQueryStringVariable in reportHandle (input
"ttSalesHeader", input "so_nbr", input-output queryString).
   run FillQueryStringVariable in reportHandle (input
"ttSalesHeader", input "so_cust", input-output queryString).
   run FillQueryStringVariable in reportHandle (input
"ttSalesHeader", input "so_ord_date", input-output queryString).

   queryString = queryString + ":".

   hSOQuery:query-prepare(queryString).
   hSOQuery:query-open().
   hSOQuery:get-next().

   repeat while not hSOQuery:query-off-end:
      create ttSalesHeader.
      assign
         ttSalesHeader.so_nbr               = so_mstr.so_nbr
         ttSalesHeader.so_cust              = so_mstr.so_cust
        ttSalesHeader.so_ord_date          = so_mstr.so_ord_date
       ttSalesHeader.sales_order_slspsn1  = so_mstr.so_slspsn[1]
       ttSalesHeader.sales_order_slspsn2  = so_mstr.so_slspsn[2]
         ttSalesHeader.sales_order_slspsn3  =
so_mstr.so_slspsn[3].
         ttSalesHeader.sales_order_slspsn4  =
so_mstr.so_slspsn[4].

      for each sod_det no-lock
where sod_det.sod_nbr = so_mstr.so_nbr:
create ttSoLine.
assign
   ttSoLine.sales_order_number        = sod_det.sod_nbr
   ttSoLine.sales_detail_line         = sod_det.sod_line
   ttSoLine.sales_detail_item         = sod_det.sod_part
```

```
            ttSoLine.sales_detail_unit_measure = sod_det.sod_um
      ttSoLine.sales_detail_due_date     = sod_det.sod_due_date.
         end.

         hSOQuery:get-next().
      end. /* Repeat query */

phReportResults = dataset dsReportResults:handle.

end procedure.
```

Index