



QAD Enterprise Applications

Training Guide Reporting Framework

70-3015A
QAD Enterprise Applications 2010
Standard Edition and Enterprise Edition
March 2010

This document contains proprietary information that is protected by copyright and other intellectual property laws. No part of this document may be reproduced, translated, or modified without the prior written consent of QAD Inc. The information contained in this document is subject to change without notice.

QAD Inc. provides this material as is and makes no warranty of any kind, expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. QAD Inc. shall not be liable for errors contained herein or for incidental or consequential damages (including lost profits) in connection with the furnishing, performance, or use of this material whether based on warranty, contract, or other legal theory.

QAD and MFG/PRO are registered trademarks of QAD Inc. The QAD logo is a trademark of QAD Inc.

Designations used by other companies to distinguish their products are often claimed as trademarks. In this document, the product names appear in initial capital or all capital letters. Contact the appropriate companies for more information regarding trademarks and registration.

Copyright ©2010 by QAD Inc.

ReportingFramework_TG_v2010SE_EE.pdf/czs/sti

QAD Inc.

100 Innovation Place
Santa Barbara, California 93108
Phone (805) 566-6000
<http://www.qad.com>

Contents

About This Course	1
Chapter 1 Introduction	3
Terminology	4
Report Viewer Demo & Features	6
Filter Screen	7
Using Filter Screen	8
Search Conditions	9
Settings	10
Output Document Types	11
Scheduled Reports	12
Viewer Screen	13
Reporting Framework Architecture	15
Reporting Framework Architecture	17
Reporting Framework Architecture: Report Render Engine	18
Report Render Engine	19
Reporting Framework Architecture: Data Source	20
Report Data Sources	21
Reporting Framework Architecture: Layout Definition	22
Report Layout Definition	23
Reporting Framework Architecture: Report	24
Report Output	25
Reporting Framework Architecture: Report Generation	26
Reporting Server Architecture: Report Batch Processor	27
Scheduled Reports	28
Installation and Deployment	29
Report Resource Maintenance	30
Using Report Resource Maintenance	31
Report Resource Designer	33
Using Report Resource Designer	34
Adding Reports to Menu	35
Template Designer	36
Using Template Designer	37
Report Import/Export and Development Process 1	39
Report Import/Export and Development Process 2	40

Browse-to-Report Feature	41
Browse-to-Report Feature: Customization	42
Using Browse-to-Report Feature	43
Browse-to-Report Feature: Output	44
Exercise	45
Summary	46

Chapter 2 Development Using Designer47

Session Objectives	48
Report Designer Program	49
Using Report Resource Designer	50
Toolbar Functions	51
Using Report Wizard	53
Report Wizard: Select Report Resource	54
Report Wizard: Select Template	55
Report Wizard: Select Data Source	56
Report Wizard: Select Layout Step	57
Report Wizard: Select Fields (Column Layout)	58
Report Wizard: Select Fields (Column Layout, Groups)	59
Report Wizard: Select Fields (Label Layout)	60
Report Wizard: Summary	61
Manually Working With Layout	62
General Layout Concepts	63
Primary Tool Bar	64
Secondary Tool Bar	66
Reports Tab	68
Data Tab	69
ControlsTab	70
Properties Window	71
Using Properties Window	72
Translated Labels	73
Number-to-Word Translator	75
Exercise	76
Sub-Reports	77
Creating Sub-Reports	78
Using Sub-Reports: Tips	79
Sub-Reports: Exercise	80
VBScript Logic	81
VBScript Examples	83
VBScript Exercise	84
Report Templates	85
Template Behavior	87
Template Designer	88

Using Template Designer	89
Applying Templates to Report Designs	91
Class Property	92
Overriding Template Attributes	93
Template Process Development Guidelines	94
Special Template for Browse Reports	96
Templates: Exercise	97
Summary	98
Chapter 3 Progress Data Source Program	99
Session Objectives	100
Reporting Framework Architecture: Data Source	101
Developing a Progress Data Source Program	102
Program Call Structure	103
Progress Data Source Program Calls	104
Overall Code Structure - Page 1	105
Overall Code Structure - Page 2	107
1. Data Set Definition Block - Example	108
2. Table Emptying Code - Example	109
ReportHelper.p Program	110
3. Metadata Definition Code Block	111
Metadata: Using CreateBufferHeader	112
Metadata: Using CreateField	113
Metadata: CreateField	114
Metadata: CreateFieldForDBField	116
Metadata: CreateFieldLikeDBField	117
Metadata: CreateField - Example	118
Metadata: CreateFieldForDBField - Example	119
Metadata: CreateFieldLikeDBField - Example	120
Metadata: Lookup Syntax	121
Metadata: Value Lists	122
Metadata: Value Lists - Example	123
Metadata: Translations	124
4. Data Retrieval Logic Code Block	125
Data Retrieval - Defining the Query	126
FillQueryStringVariable Procedure	127
Static where-clause Conditions	129
GetFilterValue Function	130
GetFilterValue - Example	131
Data Retrieval - Query Iteration	132
Data Retrieval - Query Iteration Example	133
Progress Data Source Deployment	134
Progress Data Source Deployment: Propath Example	135

Exercise 136
Summary 137

Chapter 4 Reporting Framework Administration.....139

Session Objectives 140
System Configuration: client-session.xml 141
Report Data Source Provider Settings 142
Data Source Provider Settings - Change For Financials 143
rptAdmin and rptDsgn Roles / Groups 144
Program Authorization Matrix 145
Adding a Report to the Menu 146
Exercise 147
Scheduled Reports 148
Scheduled Reports: Installation and Deployment 149
Reporting Framework: Rendering Architecture 150
Scheduled Report Server - Architecture 151
Batch Processor - Internal Design 152
Scheduled Report Server: Batch Processor Design 153
Scheduled Reports: Multiple Report Server Processes 154
Scheduled Reports Status 155
Scheduled Report Administration: Batch ID Maintenance 156
Scheduled Report Administration: Configuring Windows Task Scheduler ... 157
Printer Setup 158
Using Printer Setup Maintenance 159
Email SMTP Settings 160
Email Template File 161
Scheduled Report Administration 162
End User Report Scheduling 163
Scheduled Report Browse 164
Scheduled Report History Browse 165
Scheduled Report Maintenance 166
Scheduled Reports: Exercise 167
Summary 168

About This Course

Course Description

This course provides training on Reporting Framework in QAD Enterprise Applications.

- Certification Preparation
- Other QAD Documentation
- Online Help
- QAD Website
- Conventions

Course Objectives

By the end of this class, students will understand how to:

- Create new reports and add them to the QAD menu
- Work with tools and techniques for page layout design, including use of templates
- Write the Progress code for a custom data source
- Administer the Reporting Framework, including report servers for scheduled batch reports

Audience

- Implementation consultants
- System administrators
- Key users

Prerequisites

Basic knowledge of QAD Enterprise Applications.

Course Credit and Scheduling

This course is valid for 12 credit hours and is typically taught in two days.

QAD Web Resources

The QAD website provides product and company overviews. The Print Solution option on the opening page provides a means of compiling desired content into a document specialized to your industry, business implementation, and needs.


<http://www.qad.com/>

From QAD's main site, you can access QAD's Learning or Support sites.

Chapter 1


Introduction

Terminology

QAD
Our Passion. Your Advantage.

Terminology

- ▲ Report
- ▲ Report Resource Object (RRO)
- ▲ Report Definition (a.k.a. Page Layout Definition)
- ▲ Report Template



QAD Proprietary

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.

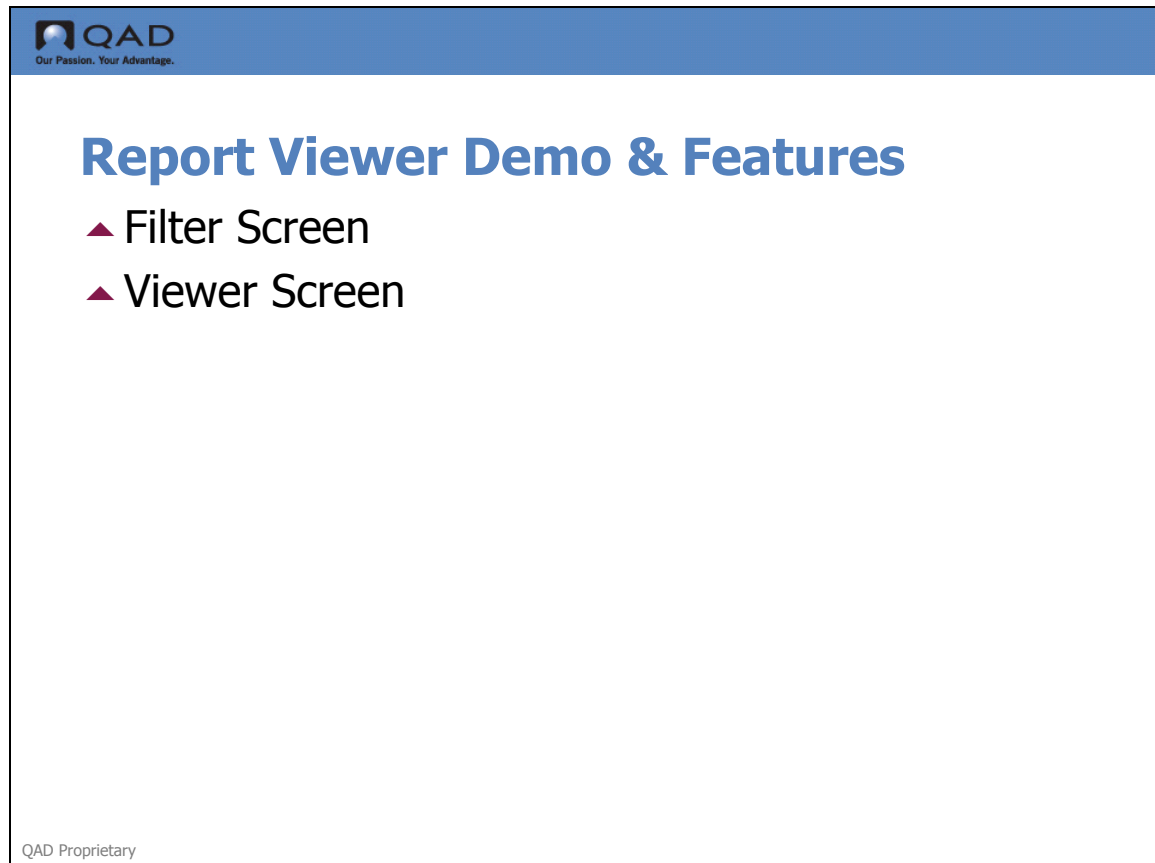
Report resources provide the data you want to display in the report while report definitions specify how to display the data in the report:

A report resource object (RRO) 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 that data binding, layout, and customized formatting of a report. It is saved as an XML file that can be edited in the Report Designer.

A report template is special kind of report definition that cannot be rendered directly by itself, but instead can be used to control certain aspects of the rendering of other reports. When designing a report, a template can be specified (optionally) in which case the report can inherit many kinds of attributes from the template, such as field colors and fonts. If at a later time these attributes are changed in the template, those changes will be seen in every report that is using that template. Any given report can inherit from at most one template, but a given template may be used to control any number of reports. Thus templates enable report developers to making changes in a single place (the template) which will have a mass effect on many reports. This is a powerful tool that can

assist the report development process in many ways, such as reducing initial development time, enforcing common standards across reports, and quickly implementing future changes to these standards.

Report Viewer Demo & Features



Report Viewer Demo & Features

- ▲ Filter Screen
- ▲ Viewer Screen

QAD Proprietary

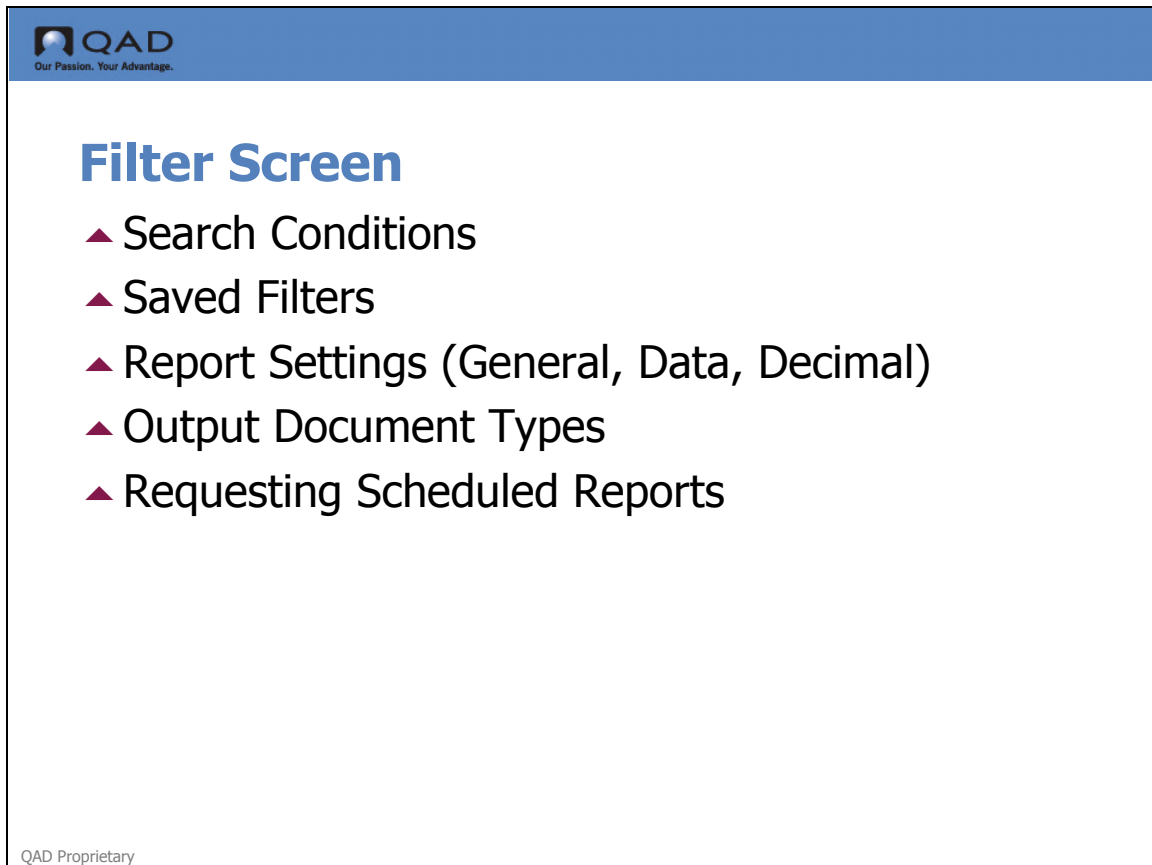
The Report Viewer includes two tabs: Filter for the filter screen and Viewer for the viewer screen.

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

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.

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.

Filter Screen



The screenshot shows the QAD logo and tagline 'Our Passion. Your Advantage.' in the top left corner. Below the logo, the title 'Filter Screen' is displayed in a large, bold, blue font. Underneath the title, there is a list of five items, each preceded by a blue triangle icon: 'Search Conditions', 'Saved Filters', 'Report Settings (General, Data, Decimal)', 'Output Document Types', and 'Requesting Scheduled Reports'. At the bottom left of the screenshot area, the text 'QAD Proprietary' is visible.

You can access the Filter screen by clicking on the Report Viewer's Filter tab. On the Filter screen, you can manage search conditions to filter report results, save filters, specify report settings, and specify output document types. Finally, you can schedule reports.

Using Filter Screen

The screenshot shows the QAD Filter screen for the report 'QAD_PrintInvoice/QAD_Println...'. The interface includes a menu bar with options like 'New Filter', 'Open', 'Save', 'Save As', 'Delete', 'Settings', 'Document', 'Run', and 'Schedule'. Below the menu is a 'Search Conditions' table with the following entries:


Field	Operator	Value	Buttons
Sales Order	contains	SO	+ x
Sold-To	equals		+ x
Ship Date	equals		+ x
Language	equals		+ x
Bill To	equals		+ x
Invoice Date	equals	3/4/2010	+ x
Update	equals	No	+ x
Show Comments	equals	Yes	+ x
Print Options	equals	No	+ x
Print Only Lines to In	equals	yes	+ x
Company	equals	10000000	+ x

QAD Proprietary

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.

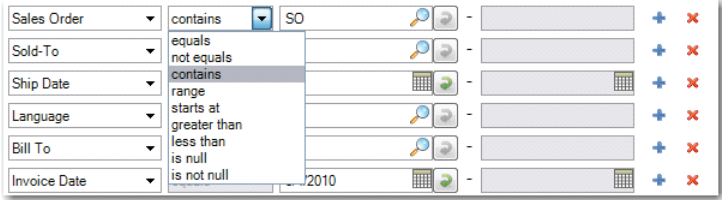
Use the Filter screen to set the Search Conditions that specify the filter criteria, the Settings option to specify General, Data, and Decimal preferences, and the output type pull-down to specify output type as Document, Excel, or PDF.

Search Conditions


Our Passion. Your Advantage.

Search Conditions

- ▲ Search Operators include:
 - equals
 - not equals
 - contains
 - range
 - starts at
 - greater than
 - less than
 - is null
 - is not null
- ▲ Adding search conditions
- ▲ Saving search conditions



QAD Proprietary


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.

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

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.

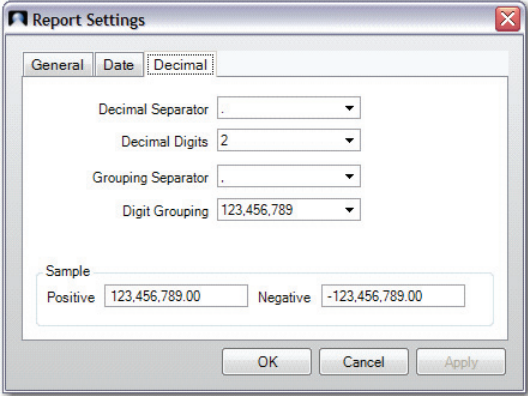
A filter is a personalized set of search conditions and settings, which means that the filters you created can only be accessed and managed by you and the administrator, and no one else.

Settings


Our Passion. Your Advantage.

Settings

- ▲ General
- ▲ Data
- ▲ Decimal




QAD Proprietary

Use Settings (Report Settings window) 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. 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. 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.

Output Document Types

 **QAD**
Our Passion. Your Advantage.

Output Document Types

- ▲ Select output type from pull-down
- ▲ Output types include:
 - Document
 - Excel
 - PDF



The screenshot shows the 'Filter Viewer' window. It has a menu bar with 'New Filter', 'Open', 'Save', 'Save As', 'Delete', 'Settings', 'Document', 'Run', and 'Schedule'. Below the menu is a search bar with 'Sales Order' in a dropdown, 'contains' as the operator, and 'SO' as the value. A dropdown menu is open under 'Document', showing 'Document', 'Pdf', 'Excel', and 'Document' (highlighted). There are also '+' and 'x' icons at the bottom right of the menu.

QAD Proprietary

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.

Scheduled Reports



Scheduled Reports

- ▲ To schedule reports, choose Schedule > New
- ▲ Specify options, including:
 - Batch ID
 - Printer
 - E-Mail
 - Save Report Output
 - Run Once
- ▲ To view scheduled reports, use Scheduled Report Browse (choose Schedule > View Schedule)
- ▲ To maintain a scheduled report, use Scheduled Report Maintenance
- ▲ To view scheduled report history, use Scheduled Report History

QAD Proprietary

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.

Viewer Screen

The screenshot displays the QAD Viewer interface. At the top left is the QAD logo with the tagline "Our Passion. Your Advantage." Below this is the title "Viewer Screen". A toolbar contains icons for Print, Save, and Refresh. Below the toolbar are navigation controls including "Page 1", "100%", and zoom in/out icons. The main content area shows an invoice report for Quality Products, Inc. The report includes the QAD logo, company name, address, and contact information. It also features a table with invoice details and "Bill To" and "Sold To" information. The word "Invoice" is prominently displayed in blue, with a "Duplicate" link below it. The QAD logo is also present in the bottom left corner of the report area.

Viewer Screen

- ▲ Toolbar (Print, Save, Refresh)
- ▲ Navigation and View Controls

QAD Proprietary

The Viewer screen includes the following options:

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 Viewer screen.

One Page. Display the report in a one-page view in the Viewer screen.

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.

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.

14 Reporting Framework — Training Guide

Zoom Out. Decrease the report review size.

Cancel Rendering. Cancel rendering the report.

Reporting Framework Architecture



Reporting Framework Architecture

- ▲ High Level Overview
- ▲ Report Render Engine
- ▲ Data Source
- ▲ Layout Definition
 - Pre-Render Engine
- ▲ Report Output
- ▲ Report Server and Scheduled Reports
- ▲ Installation and Deployment
 - Client
 - Server

QAD Proprietary

The QAD Reporting Framework contains five key components: report render engine, report layout definition, data source, pre-render engine, and 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.

- 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.

- Report layout definition

Report layout definition defines what gets displayed on the report, and where. You use Report Designer in the QAD .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.

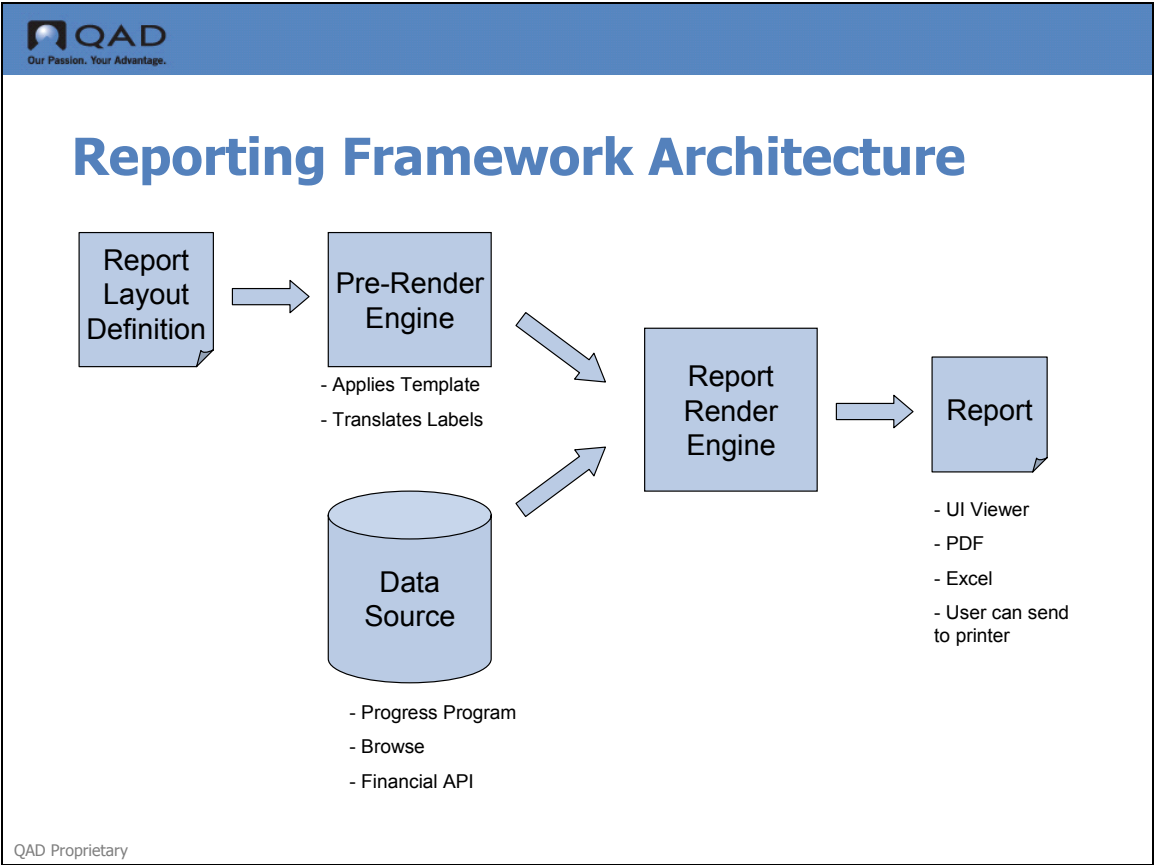
- Pre-render engine

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.

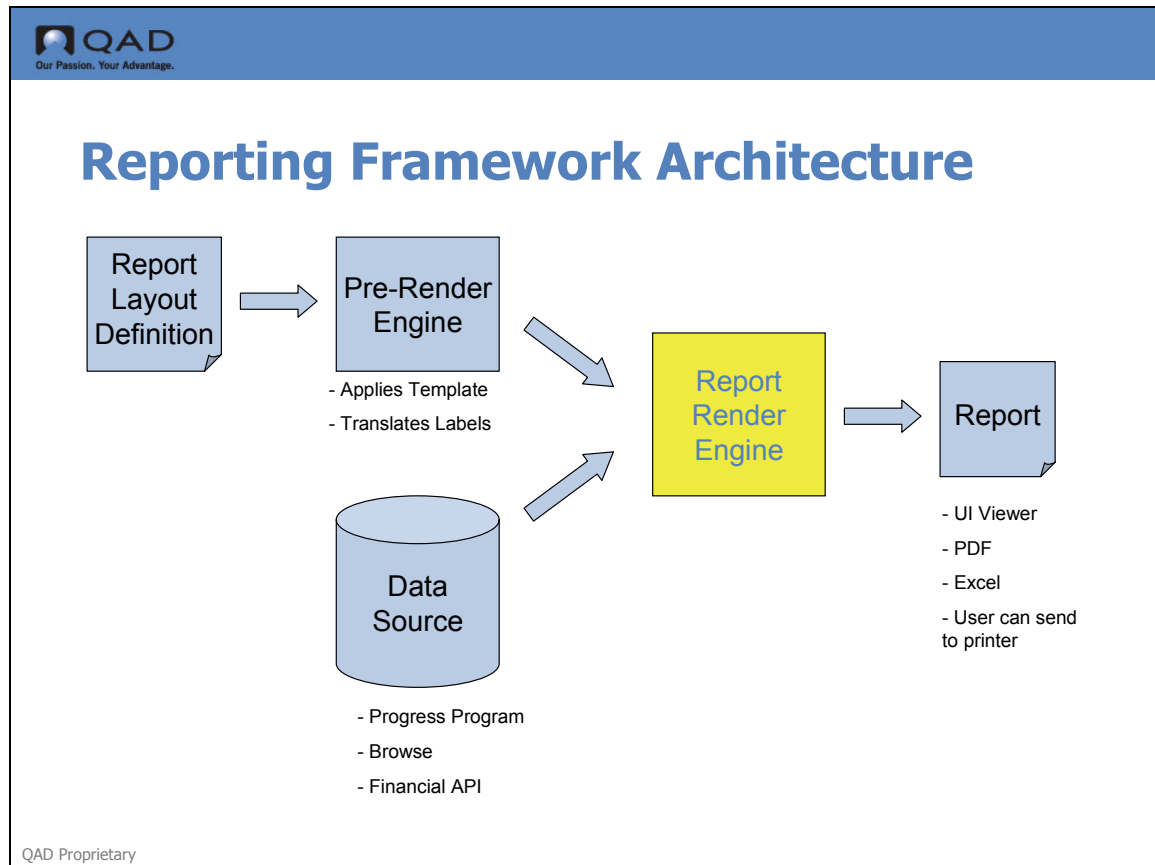
- Report output

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.

Reporting Framework Architecture



Reporting Framework Architecture: Report Render Engine



Report Render Engine



Report Render Engine

- ▲ Third party tool (Component One)
 - No end-user license fees to QAD customers
 - <http://www.componentone.com/>
 - <http://helpcentral.componentone.com/CS/forums/>
- ▲ .NET (Windows only) solution
- ▲ Inputs:
 - Data set (queried from data source)
 - Report layout definition (page layout)
- ▲ Output:
 - Rendered report

QAD Proprietary

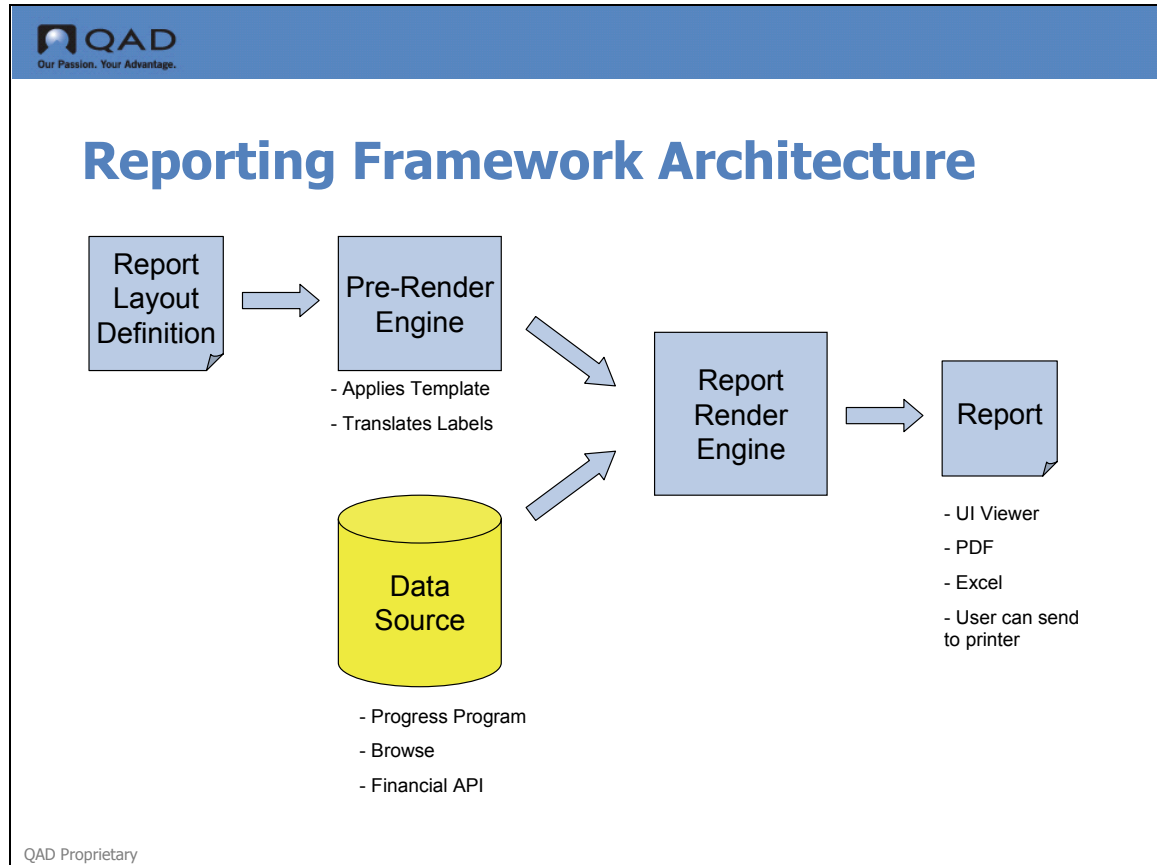
The report render engine takes in data set and report layout definition as inputs, then renders and produces the report output.

The report render engine is based on Component One.

The report render engine 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.

Reporting Framework Architecture: Data Source



Report Data Sources



Report Data Sources

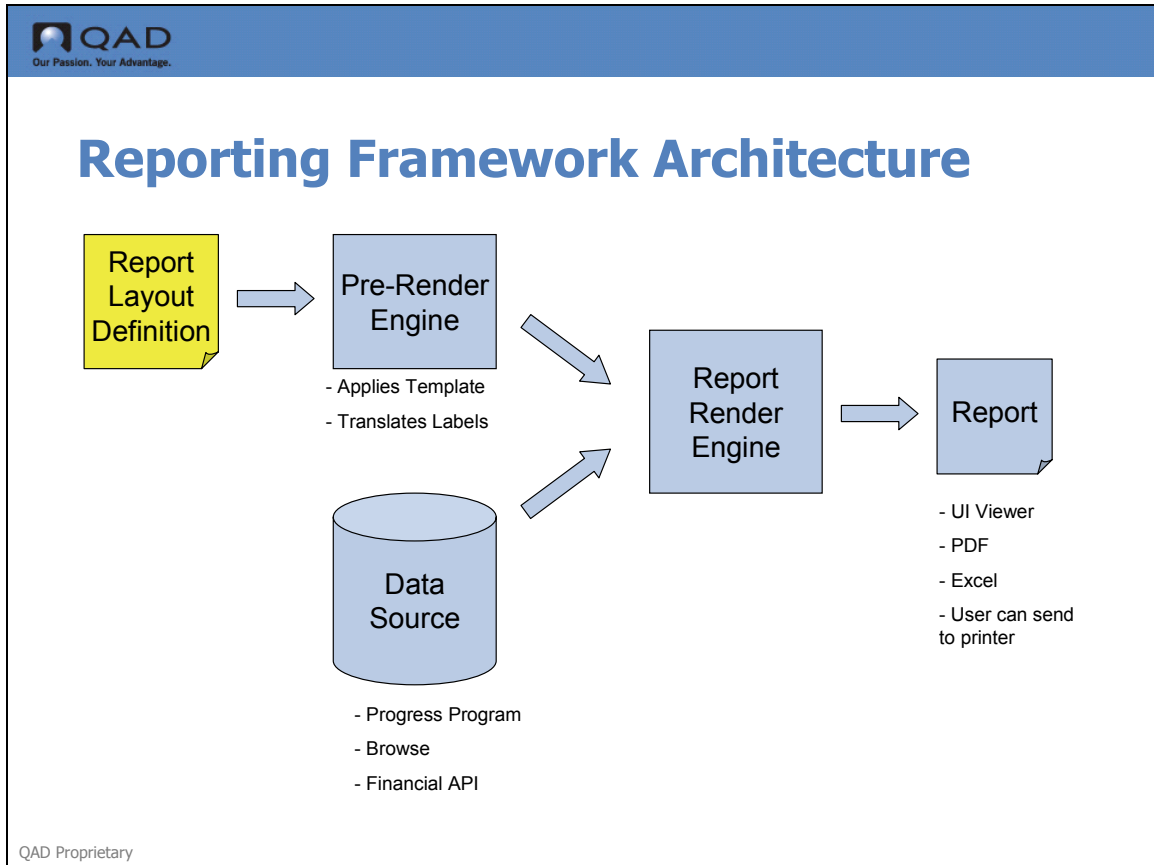
- ▲ Progress program ("Proxy")
 - Most capable and flexible option
 - Requires coding
- ▲ Browse
 - Created with Browse Maintenance program
 - No coding required
- ▲ Financial Query API
 - Created with CBF tool

QAD Proprietary

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.

Note It is even possible to create a new data source type to connect to other stores of data. This would require writing a .NET implementation of the `IDataSourceProvider` interface.

Reporting Framework Architecture: Layout Definition



Report Layout Definition



Report Layout Definition

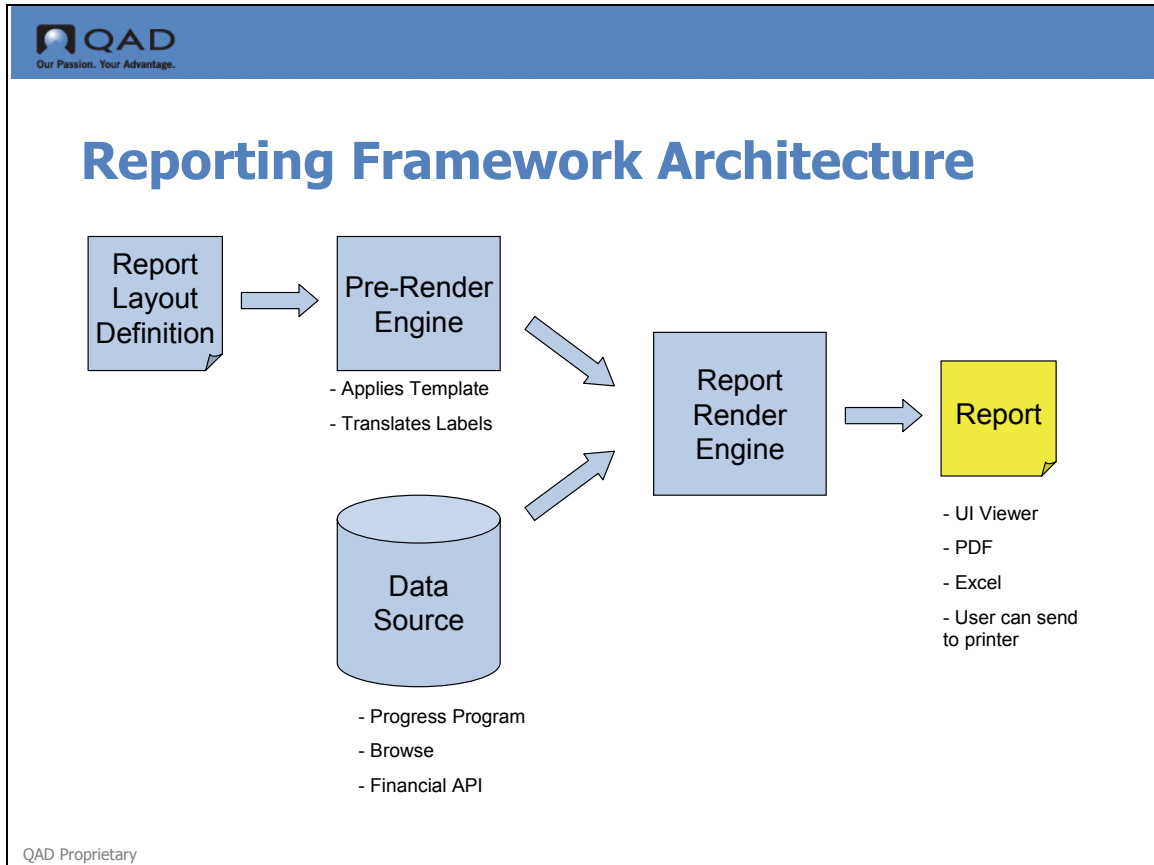
- ▲ Defines what gets displayed where on the page
- ▲ Created by report designer
- ▲ Report Resource Designer program
 - WYSIWYG editor available in .NET UI
 - Banded layout model (like Crystal, Jasper)
- ▲ Import/Export tool
 - Allows reports to be controlled as XML files
 - Useful for version control, deployment

QAD Proprietary


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 the Report Designer, either visually or in the text editor mode.

Use the Report Resource Designer program to create report definitions.

Reporting Framework Architecture: Report



Report Output



Report Output

- ▲ Reports can be rendered in various formats:
 - .NET UI viewer form (“Document” format)
 - PDF file
 - Excel file

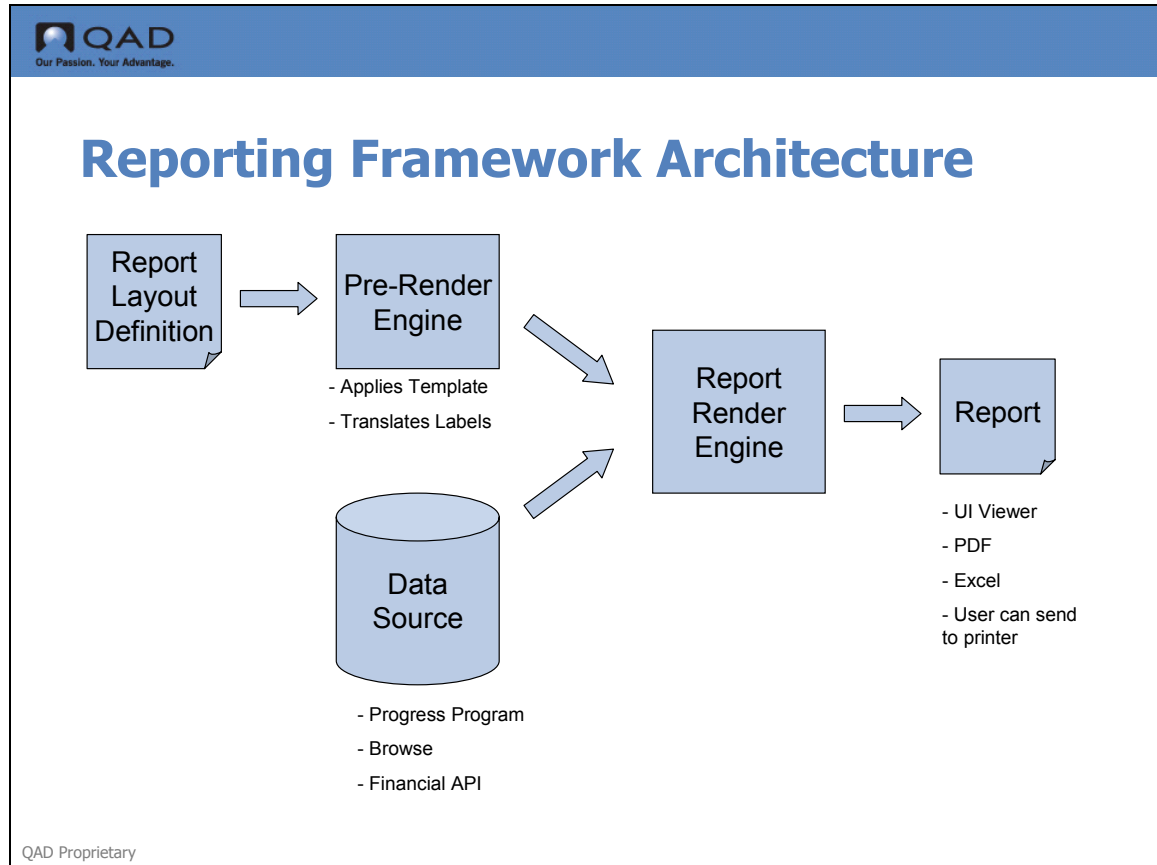
QAD Proprietary

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.

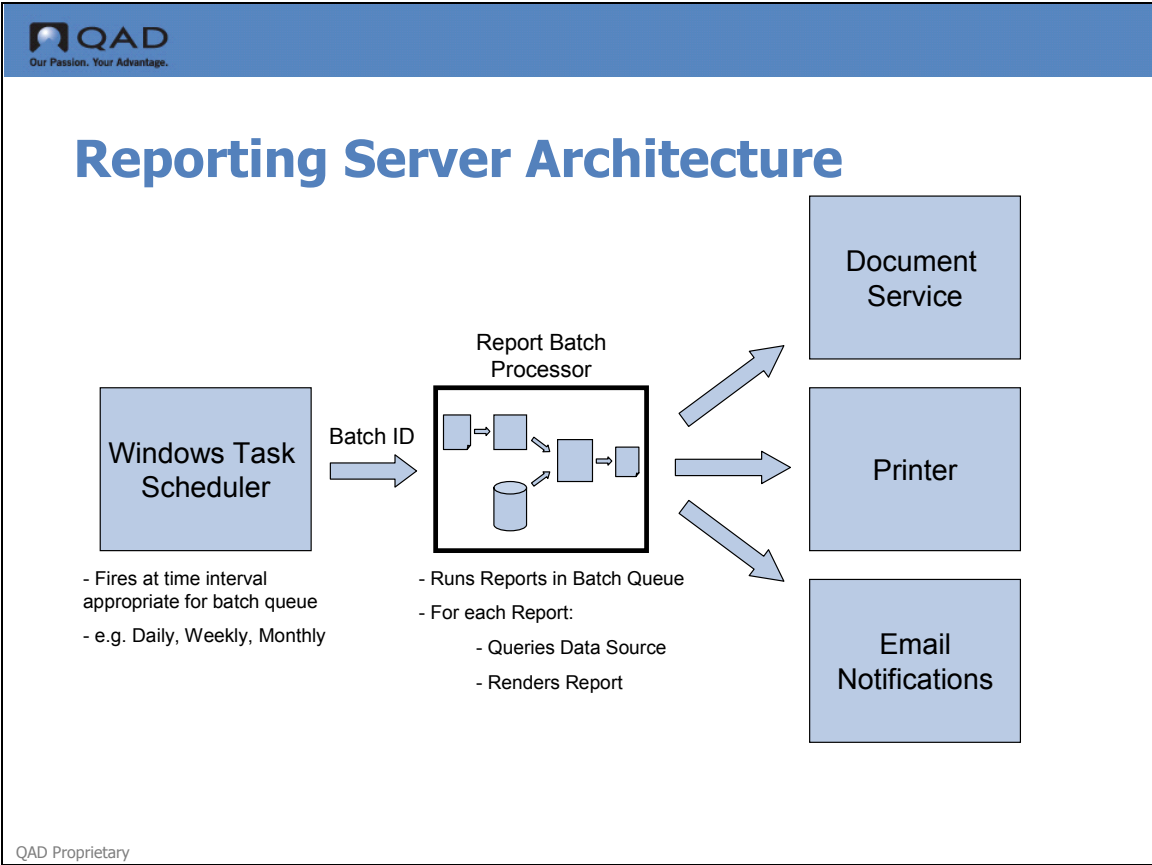
You can choose from three output formats when the report is run:

- 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.

Reporting Framework Architecture: Report Generation



Reporting Server Architecture: Report Batch Processor



Scheduled Reports



Scheduled Reports

- ▲ Report Server(s) run queues of reports (batches)
- ▲ Server processes
 - Non-GUI .NET UI processes
 - Standard installation of .NET UI
 - Launched from command line
- ▲ Windows Task Scheduler used to periodically launch batches (e.g. daily, monthly)
- ▲ Output to printer and/or file on web server
- ▲ Optional notifications to email, .NET UI Inbox

QAD Proprietary

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.

Multiple report servers can be set up for increased throughput and failover. The different servers can be configured to process different batches, or can even jointly process reports in the same batch. The Report Batch Processor coordinates the processing of scheduled reports with different priorities in the correct sequence across multiple report servers.

Installation and Deployment



Installation and Deployment

- ▲ Report Framework seamlessly integrated into .NET UI
- ▲ No special installation needed
- ▲ No license fees to customers
- ▲ Report Server Installation
 - Windows OS required
 - Install QAD .NET UI client in standard fashion
 - Run the server using command line (no GUI)
 - Can deploy multiple servers
 - Failover
 - Increased throughput

QAD Proprietary

Report Resource Maintenance



Report Resource Maintenance

- ▲ Specify *report code* of your choice
 - Must be unique
 - QAD-shipped reports use “QAD_” prefix
- ▲ Specify *report type* = “Report”
- ▲ Specify desired *data source type*
 - Browse, Proxy, or FinancialAPI (EE only)
- ▲ Specify *data source reference*
 - e.g. browse ID (if using Browse data source)
 - e.g. .p program name (if using Proxy data source)
 - e.g. financial report query object (if FinancialAPI)

QAD Proprietary

Use Report Resource Maintenance to create a report resource.

Using Report Resource Maintenance



Open Report Resource Maintenance and 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.

Report 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.

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.

Report Resource Designer



Report Resource Designer

- ▲ Used to create / modify report page layout designs (a.k.a. "report layout definitions")
- ▲ Report Code must already exist (RR Maint.)
- ▲ A report can have one or more definitions
- ▲ WYSIWYG drag+drop report layout editor
- ▲ Organized into sections (or "bands"), similar to Crystal, Jasper
- ▲ A deep dive into the designer will be done later in the course

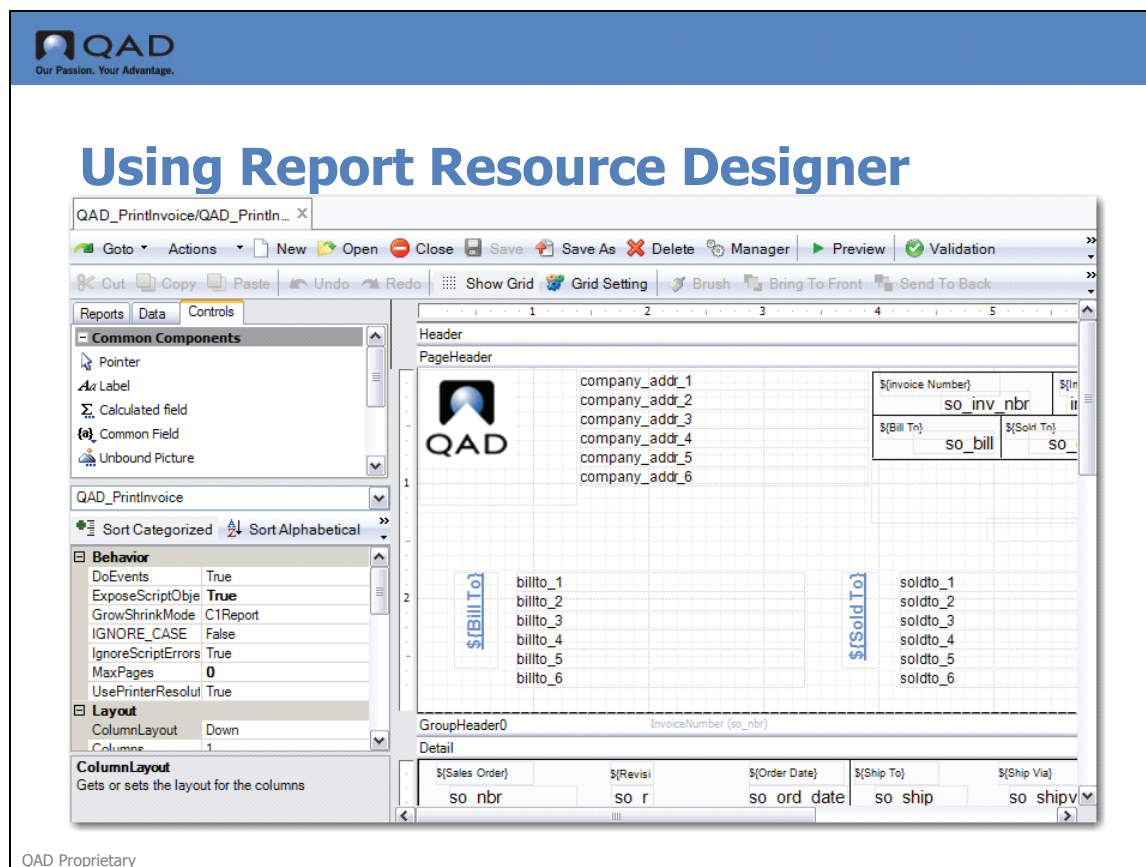
QAD Proprietary

Use the Report Resource Designer to create 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.

Select the report resource you previously created and click Next.

Using Report Resource Designer



To use the Report Resource Designer:

- 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. Select the report resource you previously created and click Next.
- 3 Select a report template or select None to use the default built-in report template. Click Next.
- 4 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.
- 5 To view all the fields in a table, click the plus sign next to the table to expand the tree.
- 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.
- 7 Select Fields into the report.
- 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.

Adding Reports to Menu



Adding Reports to Menu

- ▲ Report code must exist with at least one layout definition
 - The *default definition* will be executed when the menu item is run
- ▲ Use Menu System Maintenance to create a standard menu item
 - *Exec Procedure* must be a URN of the form:
 `urn:qad-report:c1:ReportCode`
- ▲ Grant menu item authorization using standard menu security tools
 - SE: Menu Security Maintenance
 - EE: Role Permissions Maintain

QAD Proprietary

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.

In the Exec Procedure field, 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.

Template Designer



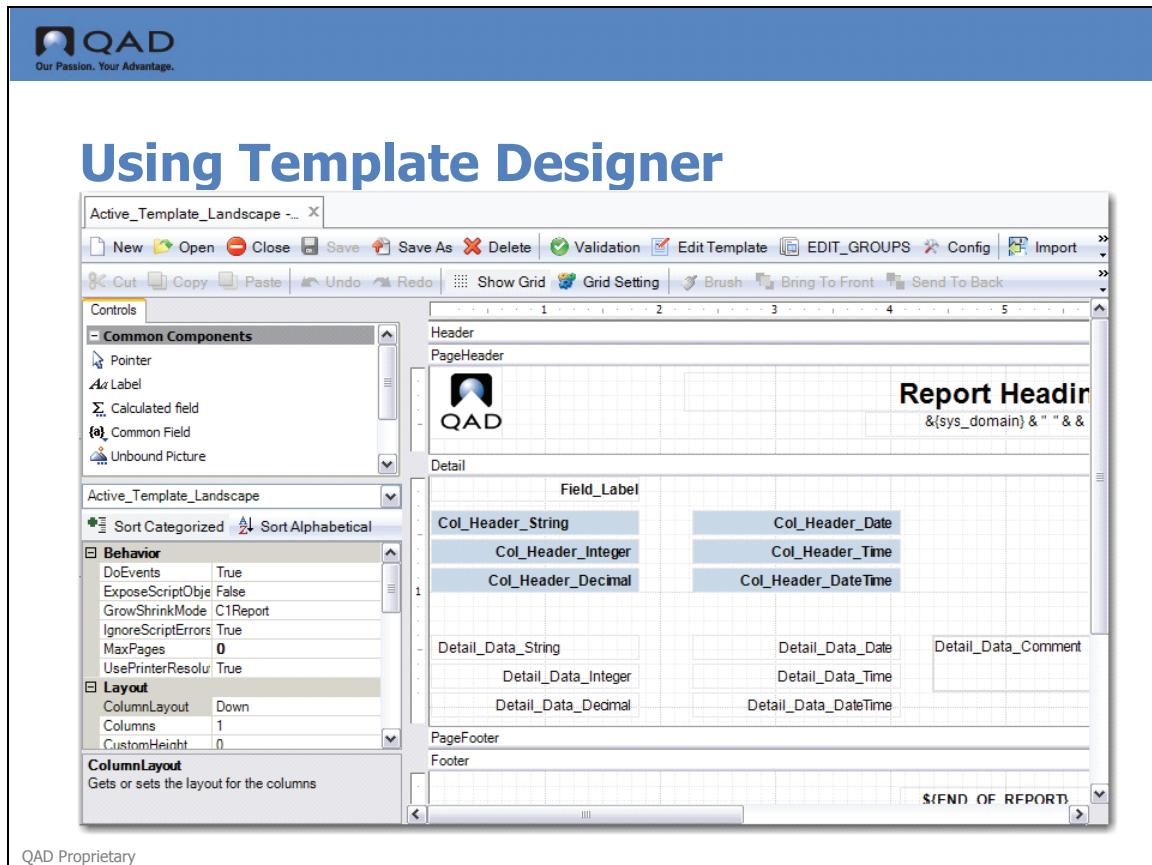
Template Designer

- ▲ Used to create / modify report templates
- ▲ A *report template* is a mechanism by which attributes can be inherited by one or more reports
 - Similar to CSS for web page design
 - Inherited attributes include properties such as fonts, colors, and alignment
 - Fields can be inherited (e.g. standard page header content, logos)
- ▲ Allows for mass changes to many reports by changing just a template common to them
- ▲ Useful mechanism for standardization

QAD Proprietary

You design report templates in Template Designer, which is similar to the Report Resource Designer. The toolbar contains the following buttons that are specific to Template Designer: Edit Template, Add Group, Config, Import, and Export.

Using Template Designer



The Template Designer works similarly to the Report Resource Designer, but there are a few important differences.

Fields placed in template sections (except for the Detail section, to be discussed next) behave much like fields in Report Resource Designer: the fields themselves will get put onto any reports whose sections inherit from the template sections. For example, the logo, title, and other fields in the PageHeader section of the template will automatically appear as fields in all reports that have sections pointing to the template's PageHeader section.

Fields in the Detail section of a template have a completely different behavior: they do not actually appear on any reports, but instead can be used to define properties (e.g. font, fore color) that can be inherited by actual fields in reports. Report fields can specify which template field to inherit properties from.

To use the Template Designer:

- 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.
- 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.
 - 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.
 - c** Under the Field Configuration tab, for each data type, select a section in the template and specify a class defined within that section.
 - d** Click OK.
- 8** Back in the Template Designer main screen, click the Save button on the toolbar to save the template.

Report Import/Export and Development Process 1



Report Import/Export and Development Process

- ▲ Report Resource Export
- ▲ Report Resource Import
- ▲ Development Process Guidelines

QAD Proprietary

When a report developer saves entries in Report Resource Maintenance, or saves report layout definitions in Report Resource Designer or Template Designer, the information is saved into tables in the qadadmin database. The export and import programs provide a way to conveniently get this data into and out of the database by allowing the data to be serialized to and from local XML files.

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

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.

Report Import/Export and Development Process 2



Report Import/Export and Development Process

- ▲ Report Resource Import / Export programs
 - Import report definitions from XML file(s) to DB records
 - Export report definitions from DB records to XML file(s)
- ▲ Allows reports to be controlled as XML files
- ▲ Useful for backup, version control, deployment
- ▲ Granularity: One XML file per RRO
- ▲ XML contains the following:
 - All information about the RRO
 - All report layout definitions for the RRO
 - Contains reference to data source, but not the data source itself
 - e.g. Proxy .p data source programs need separate file control

QAD Proprietary

It is strongly recommended that report developers regularly export the reports they are creating to XML. Besides providing data backup and providing a textual artifact that can be version controlled, exported reports can also be used to resolve problems resulting from multiple report developers inadvertently working on the same report (which should generally be avoided).

Browse-to-Report Feature



Browse-to-Report Feature

- ▲ Available on all QAD browses
- ▲ Requires QAD 2009.1 or higher (.NET UI 2.8.2)
- ▲ Creates a report from any running Browse
 - Alternate visualization of the browse data
- ▲ Run from browse's Action -> Report menu
 - Report is dynamically auto-generated
 - Report is opened in a new tab and auto-run
 - All records matching the browse search conditions will be queried for the report

QAD Proprietary

Browse-to-Report Feature: Customization



Browse-to-Report Feature: Customization

- ▲ User can customize the report layout using browse settings
 - For example:
 - Column size, order and visibility
 - Summaries
 - Groups
 - Sort column
 - Limitation:
 - Chart in browse is not put into report
- ▲ Saving browse to Favorites menu saves settings
- ▲ Empowers any end user to perform report layout customization

QAD Proprietary

Using Browse-to-Report Feature

The screenshot shows the QAD Sales Order Browse interface. At the top left is the QAD logo with the tagline "Our Passion. Your Advantage." Below the logo is a blue header with the text "Using Browse-to-Report Feature". The main window is titled "Sales Order Browse" and contains a menu bar with options: Actions, Print, Add to Favorites, Chart, Chart Designer, Refresh, New, and Edit. A dropdown menu is open under "Actions", listing: EXPORT_TO_CSV, Export to Excel, Export to PDF, Workflow, Email, and Report. The "Report" option is highlighted. Below the menu is a search bar with "Records per page: All" and navigation buttons. The main area displays a table of sales order items. The table has columns: Sales Order, Revision, Status, Line, Item Number, Unit of Measure, Quantity Ordered, and Quantity Open. The data is grouped into sections: "Sales Order" (2 items), "Sold-To:1011K (11 items)" (11 items), and "Sold-To:4000 (993 items)" (1 item visible).

Sales Order	Revision	Status	Line	Item Number	Unit of Measure	Quantity Ordered	Quantity Open
dzv8	0		1			0.0	0.0
dzv8	0		2	112@3#4\$5%6^7&8*	EA	1.0	1.0
- Sold-To:1011K (11 items)							
AD1S123	2		1	1-BB	EA	1.0	1.0
AD1S123	2		2	2-BB	EA	0.0	0.0
AD1S123	2		3			0.0	0.0
AD1S123	2		4	␣:À1A&A***,<.>/	EA	12.0	12.0
AD1S123	2		5	␣:À3A?A?,<.>/	EA	25.0	25.0
AD1S123	2		6	1234,FDSA	FX	12.0	12.0
AD1S123	2		7	112@3#4\$5%6^7&8*	EA	1.1	1.1
dzv4	0		1	␣:À1A&A***,<.>/	EA	0.0	0.0
pam1	1		1	1234,FDSA	EA	60.0	60.0
pam1	1		2	5-BB	EA	50.0	50.0
pam1	1		3	6-BB	EA	67.0	67.0
- Sold-To:4000 (993 items)							
csz_1	0		1	1-BB	EA	2.0	2.0

QAD Proprietary

Browse-to-Report Feature: Output

QAD
Our Passion. Your Advantage.

Browse-to-Report Feature: Output

Sales Order Browse x Report - Sales Order Browse x

Filter Viewer

Print Save Refresh

Page 1 100%

Sold-To
00010000

Sales Order	Revision	Status	Line	Item Number	Unit of Measure	Quantity Ordered	Quantity
d725058d	2		1	kitds	EA		-1.0
d725058d	2		2	D-01	EA		1.0
d725058d	2		3	D-01	EA		4.0
d725058f	2		1	kitds	EA		1.0
d725058x	2		1	kit2ds	EA		1.0
d725058x	2		2	1-BB	EA		13.0
d725058y	2		1	kit2ds	EA		-1.0
dm101	0		1	1-BB	EA		10.0
dzv009	1		1	V1.	EA		12.0
dzv2	0		1	ⓀⓀⓀ:;Ä 1ÄÄÄ™™, <.>/	EA		21.0
dzv3	0		1	ⓀⓀⓀ:;Ä 1ÄÄÄ™™, <.>/	EA		123.0
dzv3	0		2	ⓀⓀⓀ:;Ä 3ÄÄÄ™™, <.>/	EA		5,431.0
dzv5	0		1	ⓀⓀⓀ:;Ä 1ÄÄÄ™™, <.>/	EA		3.0
dzv5	0		2	ⓀⓀⓀ:;Ä 3ÄÄÄ™™, <.>/	EA		1,235,234.0

QAD Proprietary

Exercise



Exercise

- 1) Create a new Sales Order report based on a Browse data source (use browse so009)
 - Report Resource Maintenance
 - Report Resource Designer
- 2) Test run from Report Resource Designer program
- 3) Export the report to XML
 - Report Resource Export
- 4) Add the report to the menu
 - Menu System Maintenance
 - Role Permissions Maintain (EE)
 - Run it from the menu

QAD Proprietary

Summary



Summary

- ▲ Report Viewer
- ▲ Report Render Engine
- ▲ Report Data Sources
- ▲ Report Layout Definition
- ▲ Report Resource Maintenance
- ▲ Template Designer
- ▲ Report Export/Import and Development Process
- ▲ Browse-to-Report Feature

QAD Proprietary

This section has covered the Report Viewer, Report Render Engine, Report Data Sources, Report Layout Definition, Report Resource Maintenance, Template Designer, Report Export/Import and development process, and the Browse-to-Report feature.

Chapter 2

Development Using Designer

Session Objectives




Session Objectives

In this section you will learn how to:

- ▲ Create and modify report page layouts using the QAD Reporting Framework's Designer program
- ▲ Use sub-reports to work with multi-table data sources
- ▲ Add VBScript logic for dynamic layout modification
- ▲ Use and create templates, allowing standardization and mass changes across reports

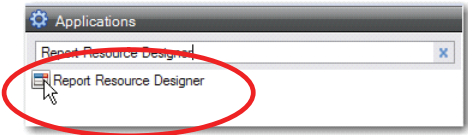
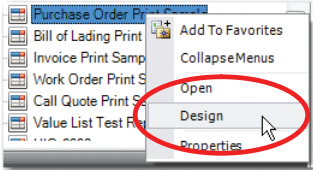
QAD Proprietary

Report Designer Program



Report Designer Program

- ▲ WYSIWYG editor program for editing report page layout definitions
- ▲ Only available to users belonging to the rptDsgn or rptAdmin roles (groups)
- ▲ Two ways to invoke the program:
 - *Report Resource Designer* menu item
 - Right-click on any report menu item and click *Design*

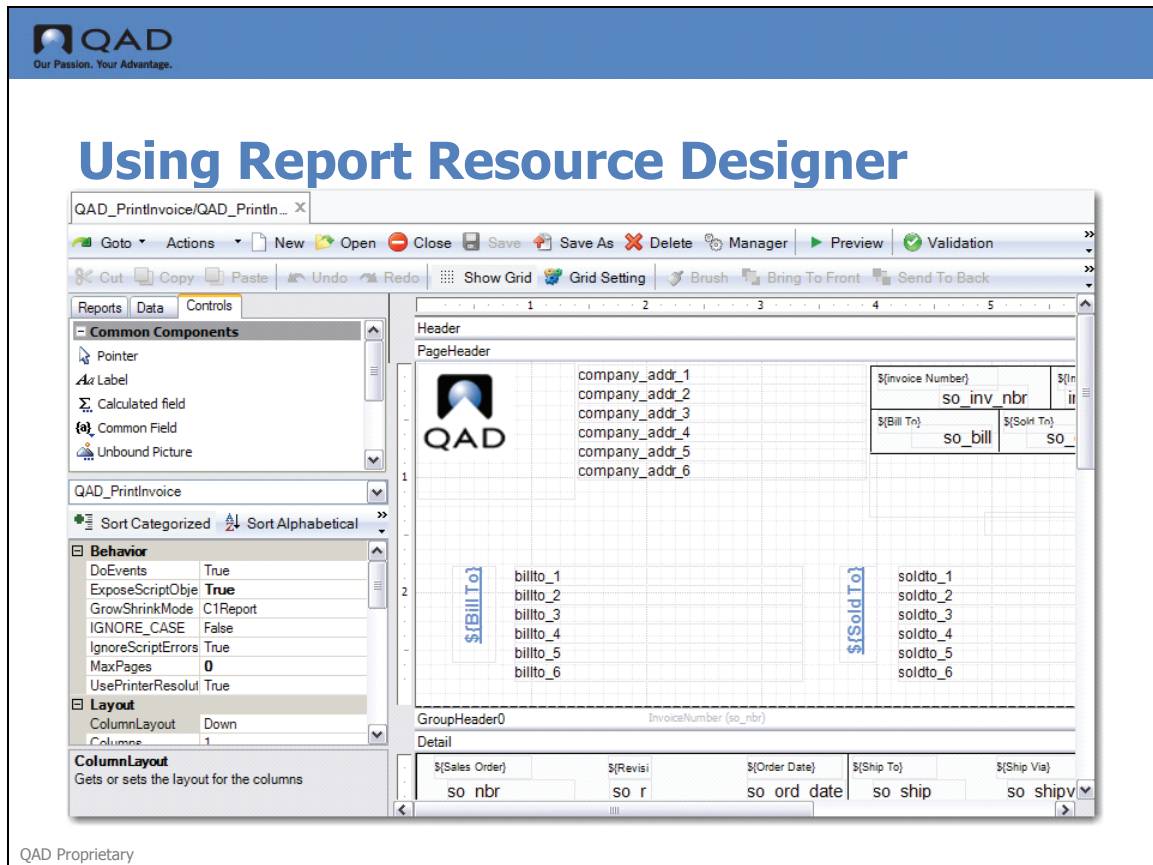
QAD Proprietary

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.

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

Using Report Resource Designer




In the 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.

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.

Toolbar Functions



Toolbar Functions

- ▲ New, Open, Close, Save, Save As, Delete
- ▲ Manager Form
- ▲ Preview
- ▲ Edit Report Definition Form and Validation Tool
- ▲ About Form, with discussion of designer version issues
- ▲ Actions Tab
 - Metadata export
 - Data export/import

QAD Proprietary

New: Launch Report Wizard to create a new report definition.

Open. Open an existing definition.

Close. Close the current report definition.

Save. Save current report definition.

Save As. Save current report definition under a different name.

Manager Form (Report Definition Manager Form). Click the Manager button on the toolbar...

To delete an existing report definition: In the Manager 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.

Click Preview to display the current report definition in preview mode. In the preview mode, you can only navigate through the generated report.

You can export report metadata to an XML file for review and debug a report design. On the toolbar, choose Actions and click Export Metadata.

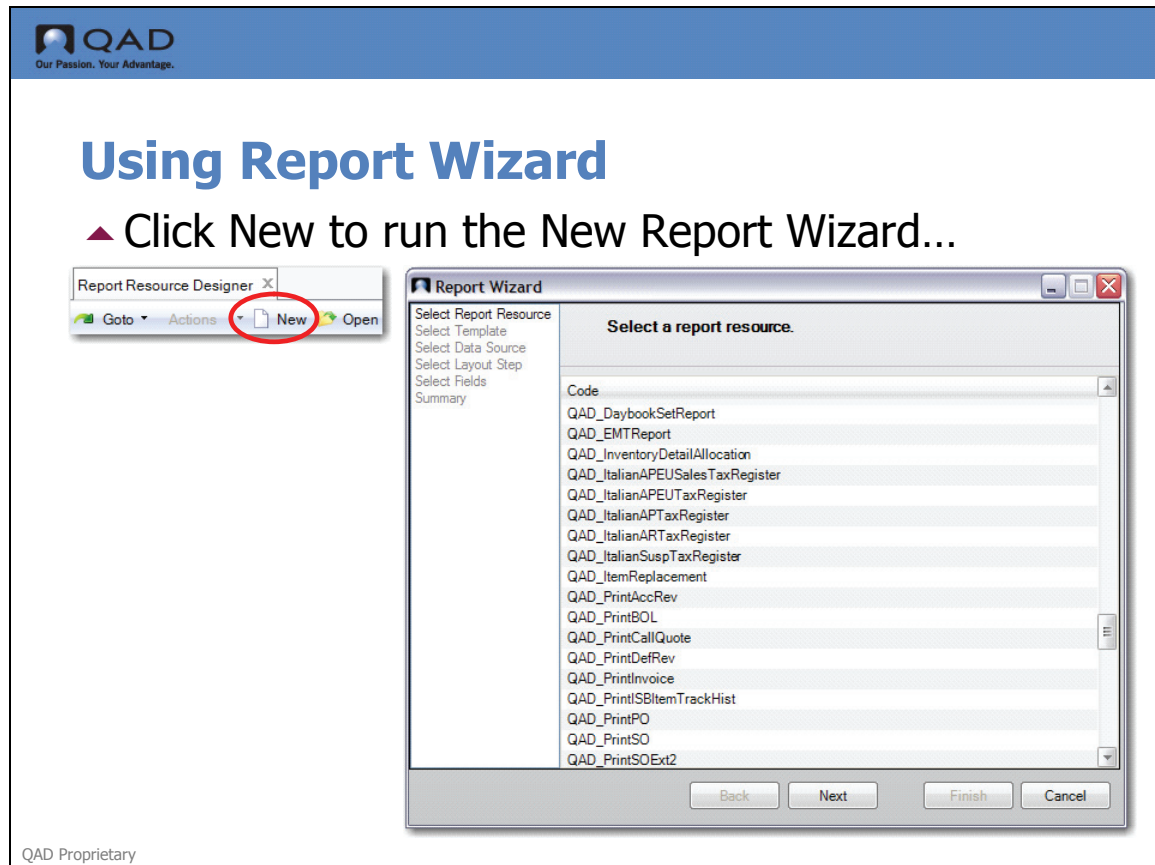
In the Save As dialog box, specify the name of the XML file and where you want to save the file. By default, the file is named after the current report with `_meta.xml` appended.

You can export the data for a report to an XML file for review, testing, and debugging purposes. You can then import an XML file containing report data and run it. The exporting and importing actions will not take place until a preview of the report is run from the Report Designer.

Importing report data from an XML file is a useful tool for scenarios such as:

- Repeated testing of a report whose data source takes a long time to run.
- Manually changing data values for the purpose of quickly testing report design logic, without having to change them in the live production database.

Using Report Wizard



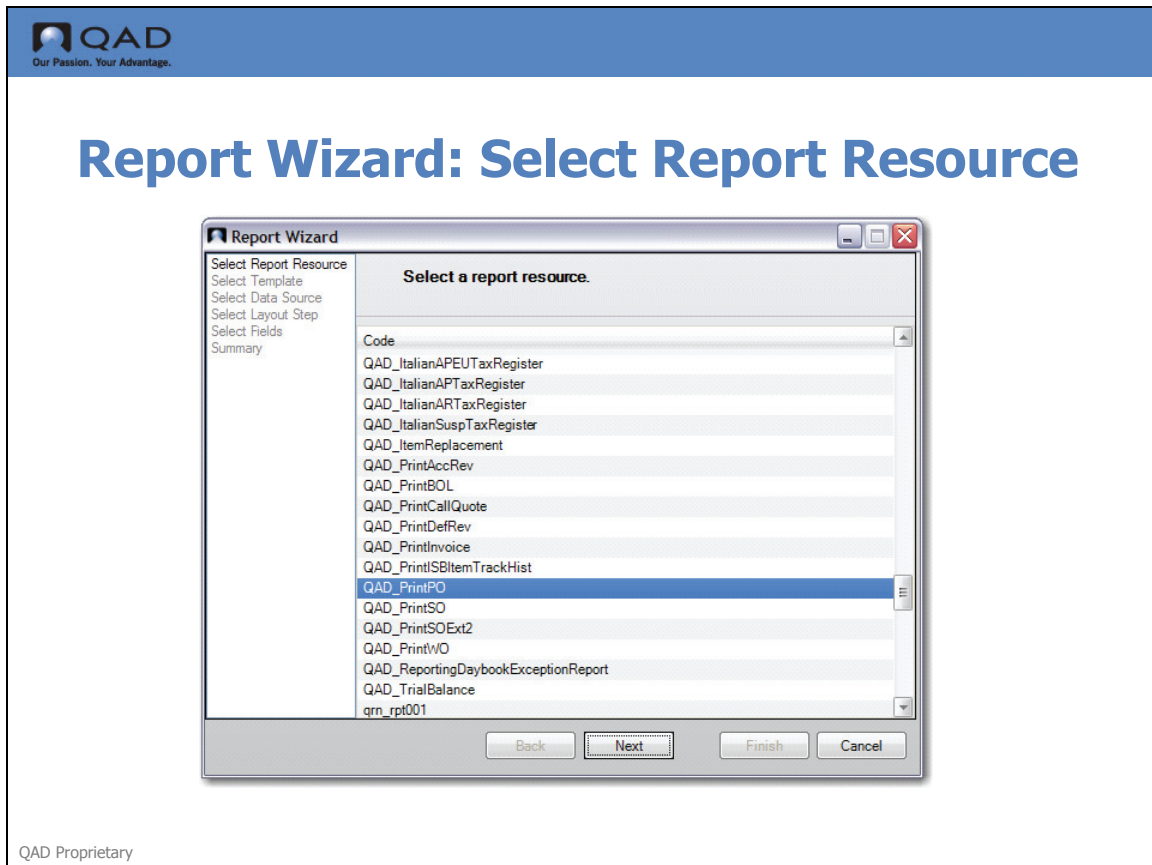
The screenshot shows the QAD Report Wizard interface. On the left, the 'Report Resource Designer' window has a menu bar with 'Goto', 'Actions', 'New', and 'Open'. The 'New' button is circled in red. On the right, the 'Report Wizard' dialog box is open, displaying a list of report resources under the heading 'Select a report resource.' The list includes various report codes such as QAD_DaybookSetReport, QAD_EMTRReport, and QAD_PrintSOExt2. The dialog box has a 'Back' button, a 'Next' button, a 'Finish' button, and a 'Cancel' button.

QAD Proprietary

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.

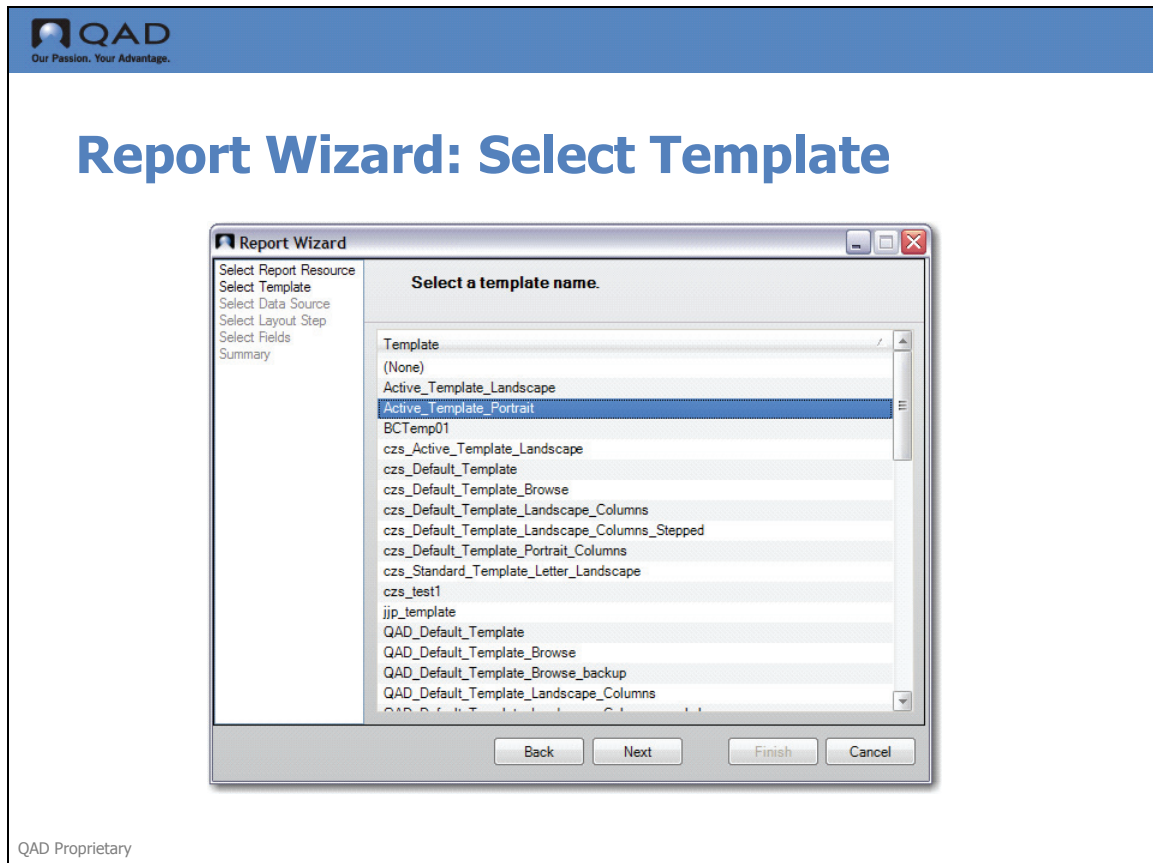
First, click New to launch the Report Wizard.

Report Wizard: Select Report Resource



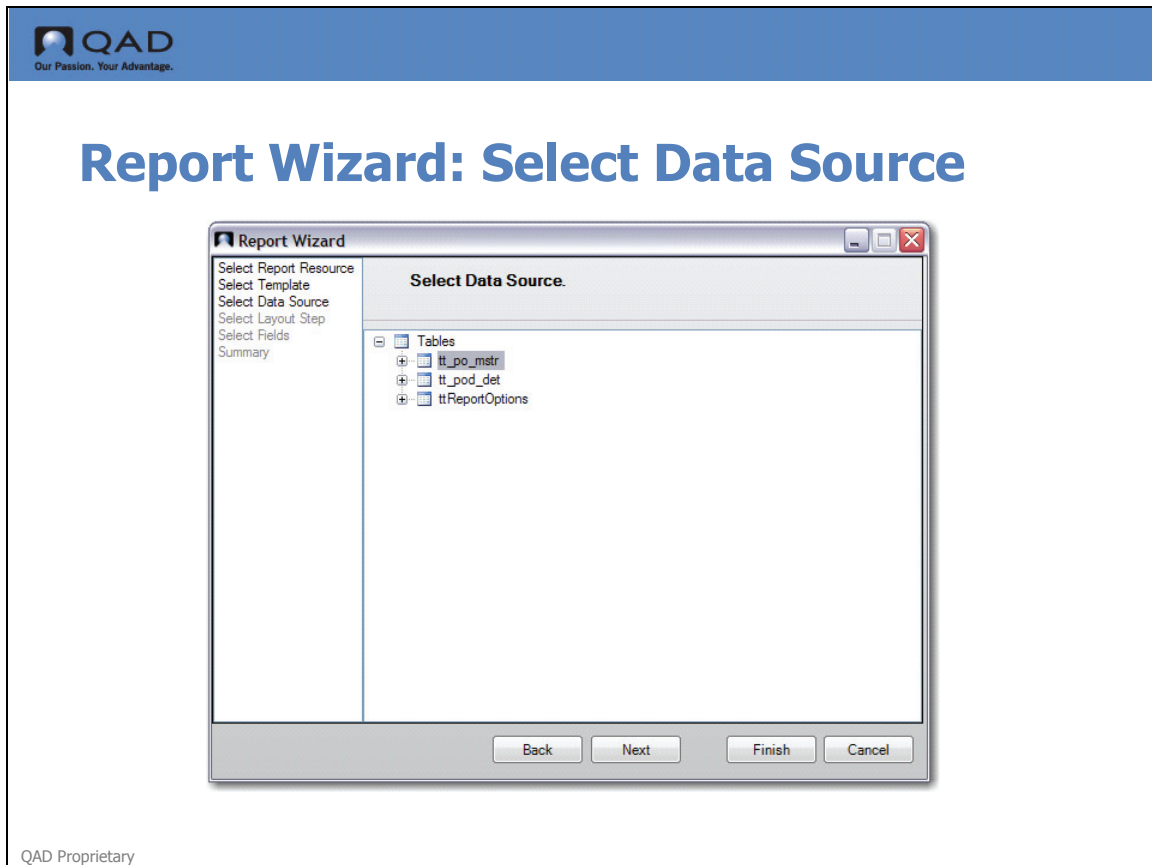
Select a report resource and click Next.

Report Wizard: Select Template



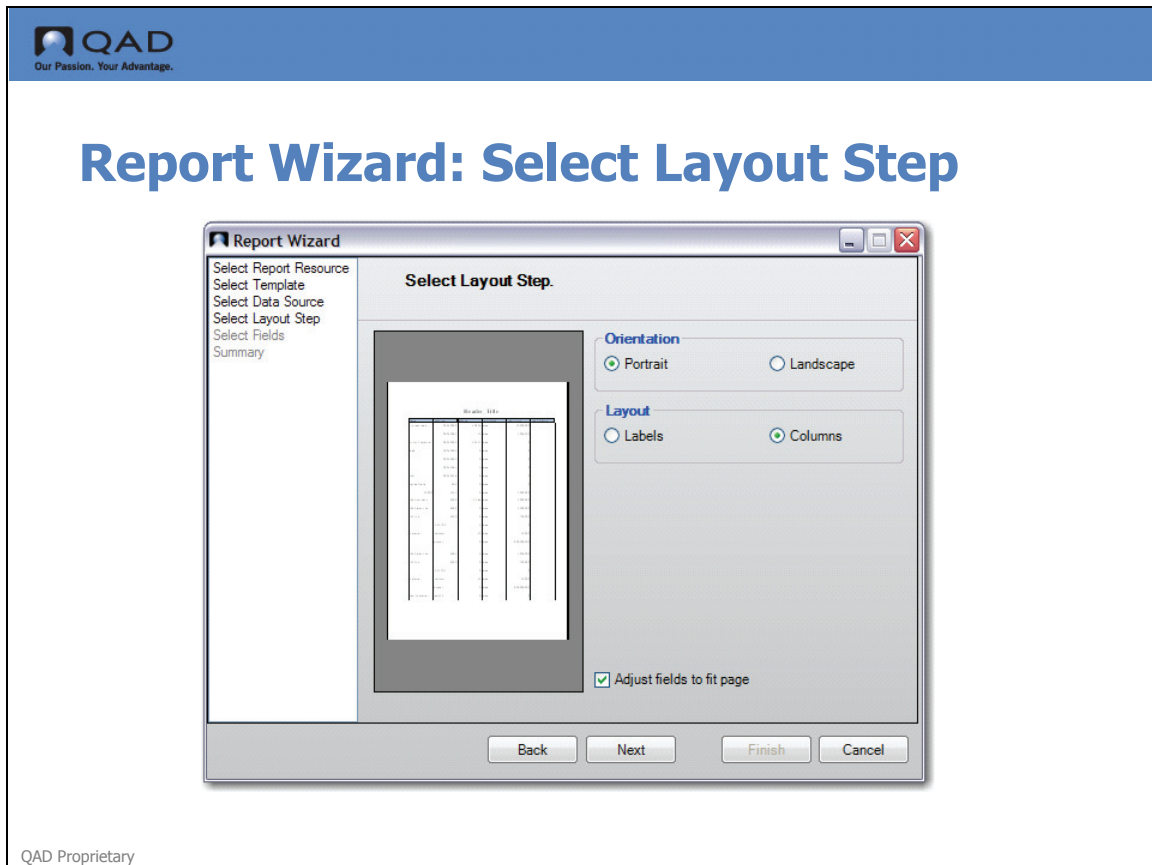
Select a template name such as `Active_Template_Portrait` and click Next.

Report Wizard: Select Data Source



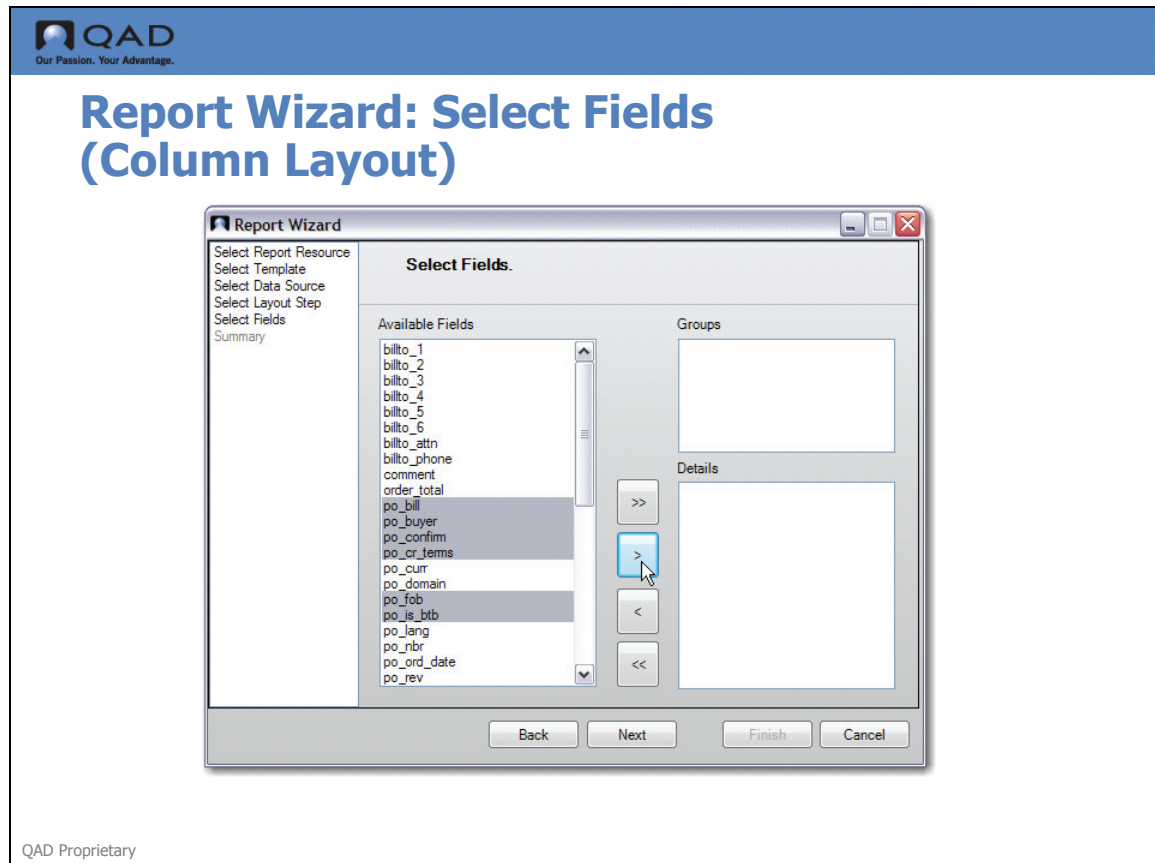
Select a data source and click Next.

Report Wizard: Select Layout Step



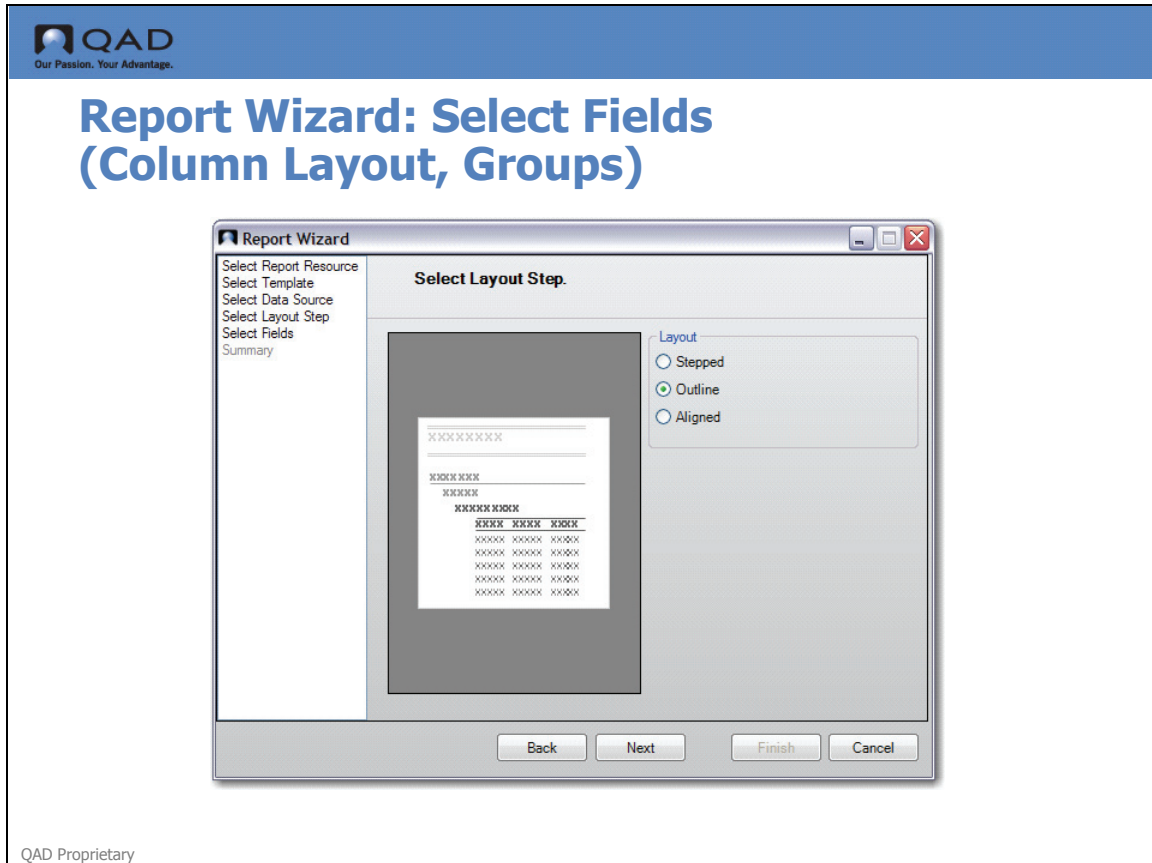
In the Select Layout Step, you can choose your report orientation as either Portrait or Landscape. Additionally, you can choose the report layout as Labels or Columns. After you click Next, the next screen will vary depending on whether you have chosen Labels or Columns.

Report Wizard: Select Fields (Column Layout)



For Column layout, you select the fields you want to use from the available fields of your data source. Optionally, you can organize the columns into groupings. This feature is similar to the “group by” display feature in browses.

Report Wizard: Select Fields (Column Layout, Groups)



If you have chosen column layout and specified groups, you next select the type of layout you would like as Stepped, Outline, or Aligned.

Report Wizard: Select Fields (Label Layout)

QAD
Our Passion. Your Advantage.

Report Wizard: Select Fields (Label Layout)

Report Wizard

Select Report Resource
Select Template
Select Data Source
Select Layout Step
Select Fields
Summary

Select Fields.

Number of columns: 2

Available Fields

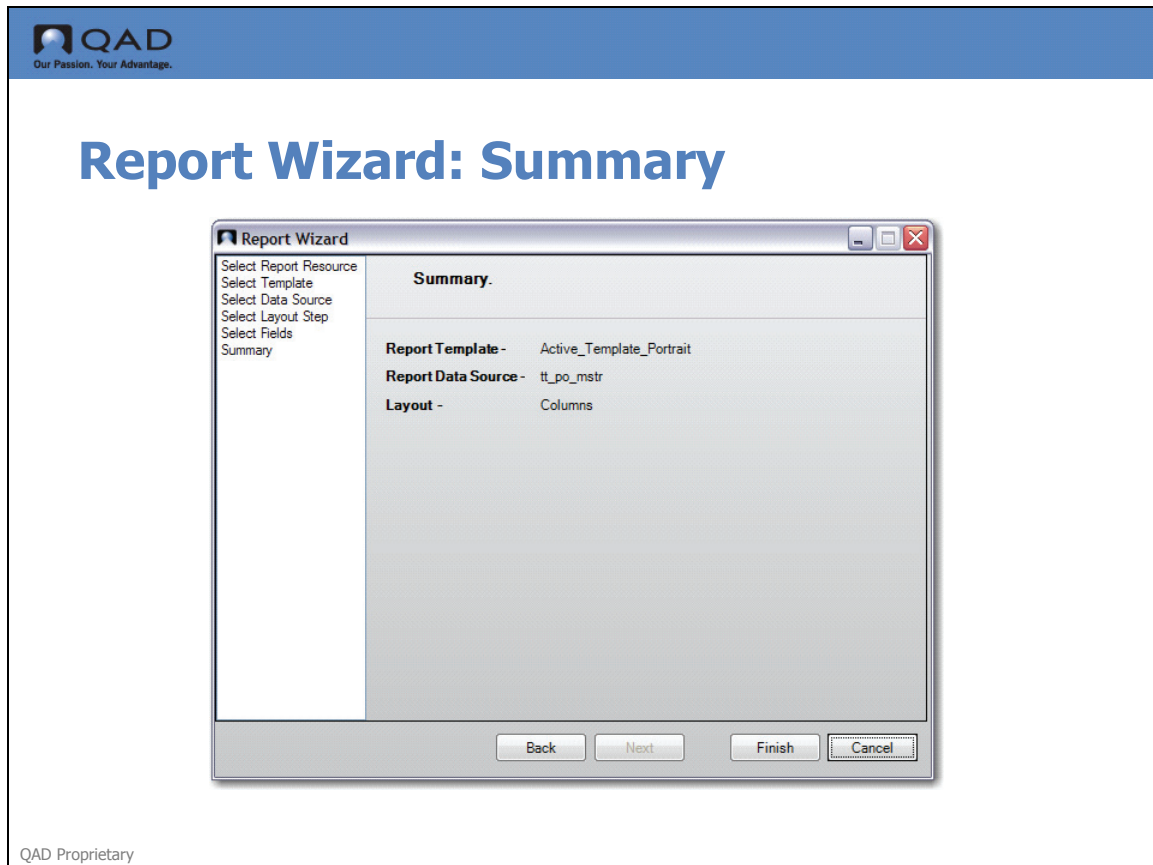
- billto_1
- billto_2
- billto_3
- billto_4
- billto_5
- billto_6
- billto_attn
- billto_phone
- comment
- order_total
- po_bill
- po_buyer
- po_confirm
- po_cr_terms
- po_curr
- po_domain
- po_fob

Columns

Back Next Finish Cancel


QAD Proprietary

Report Wizard: Summary



Upon completion, the Report Wizard's summary screen displays information about the template, data source, and layout. Finally, Click Finish.

Manually Working With Layout




Manually Working With Layout

- ▲ General Concepts
 - Sections
 - Fields
- ▲ Secondary Tool Bar
 - [sub-sections for each tool]
- ▲ Report Tab
 - Sorting and Grouping
 - Multi-Level Sorting
- ▲ Data Tab
 - Fields
 - Parameters
 - Settings
- ▲ Controls Tab
 - [sub-sections for each type of control]

QAD Proprietary

(This slide gives an overview of the topics covered in more detail in subsequent slides.)

General Layout Concepts



General Layout Concepts

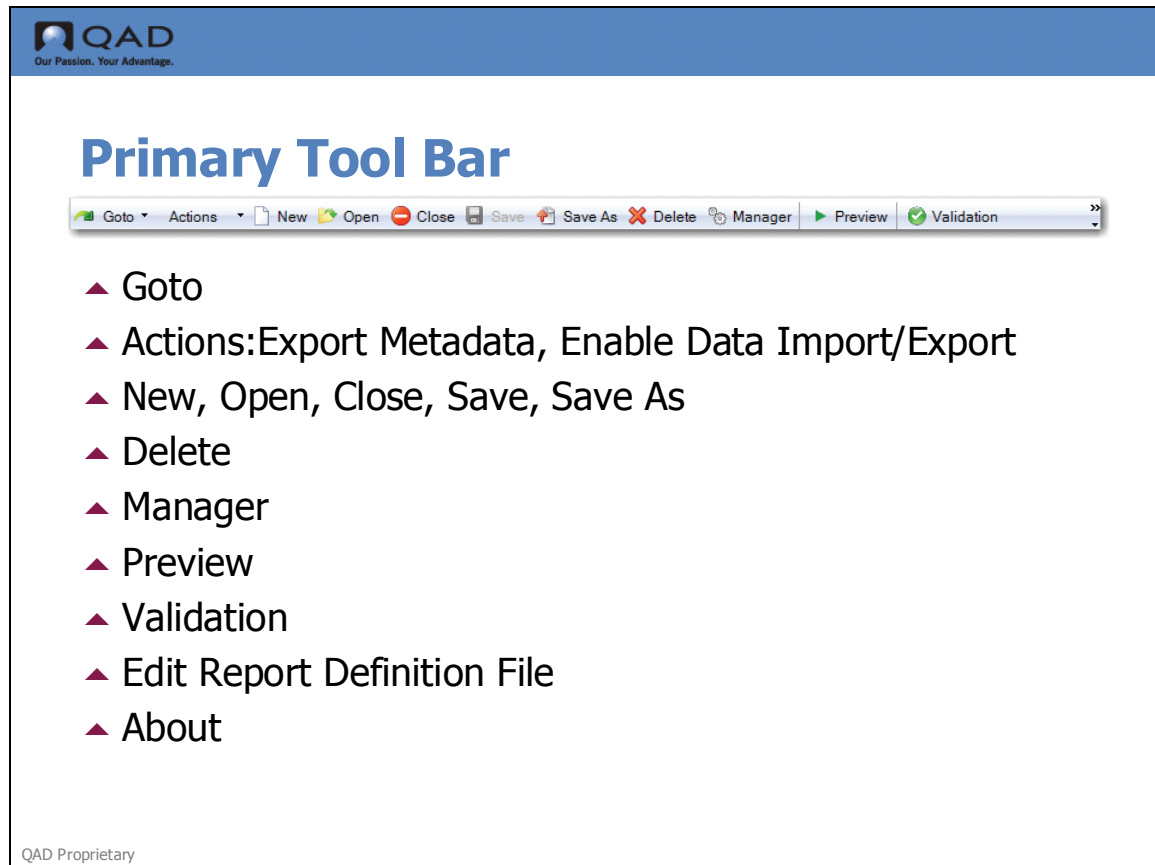
- ▲ Sections
 - Header: Rendered once at beginning of report
 - Page Header: Rendered at beginning of each page
 - Group Header: Rendered at start of each record group
 - Detail: Rendered once per data record
 - Group Footer: Rendered at end of each record group
 - PageFooter: Rendered at end of each page
 - Footer: Rendered once at end of report
- ▲ Fields

QAD Proprietary

Fields: Report fields can be added to any desired section. Fields can be used for many different types of information to display:

- Can be tied to data fields in the underlying data source
- Can be tied to filter parameters or other report settings
- Can be used to display labels or other static text
- Can be used for calculated expressions
- Can be used to display images, or build simple graphics (lines, rectangles)
- Can be used to create charts
- Can be used for bar codes

Primary Tool Bar



The screenshot shows the QAD Primary Tool Bar with the following icons and actions:

- ▲ Goto
- ▲ Actions: Export Metadata, Enable Data Import/Export
- ▲ New, Open, Close, Save, Save As
- ▲ Delete
- ▲ Manager
- ▲ Preview
- ▲ Validation
- ▲ Edit Report Definition File
- ▲ About

QAD Proprietary

Goto. Go to Report Resource Maintenance.

Actions. Menu of options for exporting and importing metadata and data for testing and debugging purposes.

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.

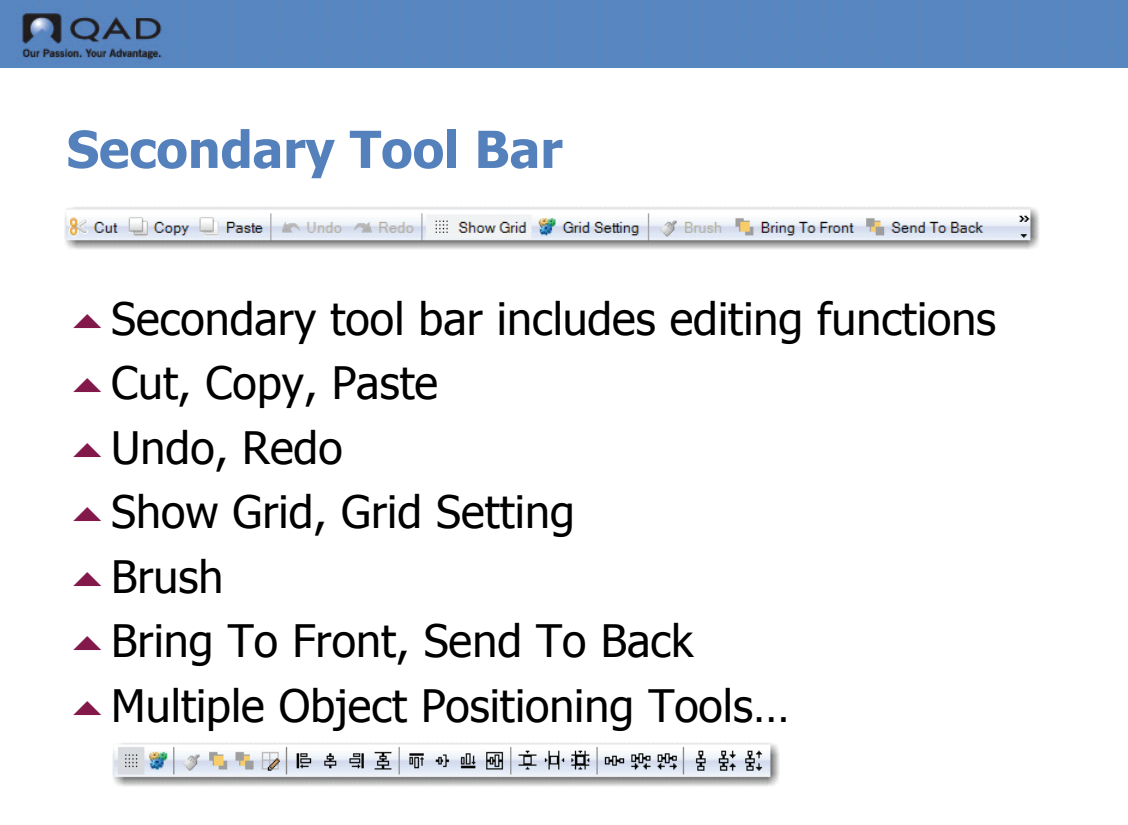
Preview. Display the current report definition in preview mode. In the preview mode, you can only navigate through the generated report.

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.

Secondary Tool Bar



The screenshot shows the QAD logo at the top left with the tagline "Our Passion. Your Advantage." Below it, the title "Secondary Tool Bar" is displayed in a large blue font. The tool bar itself contains the following icons from left to right: Cut, Copy, Paste, Undo, Redo, Show Grid, Grid Setting, Brush, Bring To Front, and Send To Back. Below the tool bar, a list of functions is provided with red triangle bullet points. At the bottom of the screenshot, there is a secondary toolbar with Korean text labels: "복사", "붙여넣기", "되돌리기", "다시 실행", "표시줄", "정렬", "앞면으로", "뒤면으로", "다중 객체 위치 지정 도구".

- ▲ Secondary tool bar includes editing functions
- ▲ Cut, Copy, Paste
- ▲ Undo, Redo
- ▲ Show Grid, Grid Setting
- ▲ Brush
- ▲ Bring To Front, Send To Back
- ▲ Multiple Object Positioning Tools...

QAD Proprietary

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.

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 “Multiple Object Positioning Tools” only apply to multiple objects on the report and will be grayed if only one object is selected:

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.

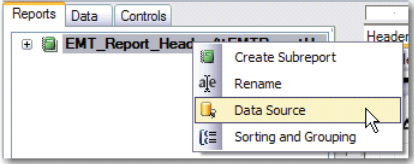
Decrease Vertical Spacing. Vertically reduce spacing between multiple selected objects.

Increase Vertical Spacing. Vertically increment spacing between multiple selected objects.

Reports Tab

 QAD
Our Passion. Your Advantage.

Reports Tab




- ▲ View / Edit name of report
- ▲ Data Source
- ▲ Sorting and Grouping
- ▲ Create Sub reports (more details later)

QAD Proprietary

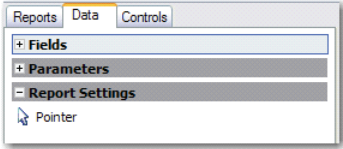
The Reports tab displays the report definition you are designing as well as all subordinate sub-reports embedded in the current report.

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


Our Passion. Your Advantage.

Data Tab



- ▲ Used to choose type of data field to create:
 - Fields: List of available fields from data source
 - Parameters: List of filter parameter values
 - Report Settings: List of system-supplied report settings
- ▲ Usage
 - First, click on desired data field type in the tab list
 - Next, click and drag the mouse on the desired section to create a rectangle defining the field position

QAD Proprietary


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.

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

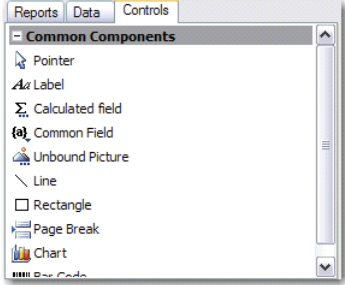
The Parameters group contains all the available search condition parameters under the Filter tab in Report Viewer.

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

ControlsTab


Our Passion. Your Advantage.

Controls Tab



- ▲ Used to choose type of field control to create
- ▲ Usage
 - First, click on desired field control type in the tab list
 - Next, click and drag the mouse on the desired section to create a rectangle defining the field position

QAD Proprietary

Pointer. Cancels any previous field selection to get back to the pointer tool.

Label. Static text (with provisions for linking to translated labels).

Calculated Field. Contains a VBScript expression.

Common Field. Several types of pre-defined fields.

Unbound Picture. Used to display an image from a file.

Note Image bits get embedded into report design XML.


Line, Rectangle. Simple geometric shapes.

Page Break. Inserts page break for every N rows of data.

Chart. Create simple charts tied to data source.

Bar Code. Create bar code tied to any data field.

Properties Window



Properties Window

- ▲ Displays properties for selected object
 - Can select by highlighting in WYSIWYG panel (report, section, field)
 - Can also select using drop-down box above the grid
- ▲ Report-Level Properties
 - e.g. Paper Size, Margins, Width and the Ruler
 - TIP: Check / set these before working on report !!
 - In QAD 2010 release, can be inherited from templates
- ▲ Section-Level Properties
 - Section sizing in WYSISYG panel
- ▲ Field-Level Properties

QAD Proprietary

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.

Using Properties Window

QAD Proprietary

The various Section, Report, and Field properties are described in detail in the *Reporting Framework User Guide*.

Translated Labels



Translated Labels

- ▲ Original implementation - long labels only
 - 2009.1 release (2.8.2 .NET UI) and lower
 - Triggered by special Text in a label-type report field:
 - `${TERM_KEY}` – will display long label
- ▲ New implementation – smart label chooser
 - 2010 release (2.9 .NET UI) and higher
 - Triggered by special Text in a label-type report field:
 - `${TERM_KEY}` – displays longest label that fits
 - `${TERM_KEY}S` – displays short label
 - `${TERM_KEY}M` – displays medium label
 - `${TERM_KEY}L` – displays long label
 - `${TERM_KEY}!` – displays stacked label

QAD Proprietary

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. This table contains translated labels that can be specified with a string that specifies the desired label (the label term key, for example “SALES_ORDER”). A report design can use these label term keys so that when the report is run, the system will retrieve the translated label in the appropriate language for the user.

If you attempt to translate a label 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.

For Label report fields, the value of the Text property will typically be literally displayed in the field when the report is rendered. However, the system can be instructed to automatically select translated labels when the report is rendered. This mode of operation is triggered when the Text property contains a value with the special `${ LABEL_TERM}` format, which instructs the system to look up the translated label corresponding to the LABEL_TERM used.

System labels can have short, medium, and long versions, of which the system intelligently selects the largest that will fit into the report field, taking into account factors such as field width and font size. For example, if the field width is not large enough to display the large label for the given text in the given font, the medium label will be displayed automatically if it fits.

Note that the Reporting Framework is flexible about the format of the label terms. For example, you can enter the term for Sales Order as `${SALES_ORDER}` or `${Sales Order}`. The system will replace spaces with underscores and perform case-sensitive matching at run-time.

There are also some variations on this format that allow you to explicitly specify which label should be used:

- `${TERM}S` — specifies the short label
- `${TERM}M` — specifies the medium label
- `${TERM}L` — specifies the long label
- `${TERM}!` — specifies a stacked label

Note that in the case of stacked labels, the height of the field must be large enough to fit all the levels in the stacked label or the display of the label will be truncated.

For Calculated fields, the value of the Text property is evaluated as a VBScript expression and the result will be displayed in the field when the report is rendered. One special case is when the expression consists of the name of a report data field, in which case the value of that data field will be displayed when the report is rendered.

For Calculated fields, you can also specify to have integers converted into the words for the number. For instance, you can have “12” displayed as “twelve.” The format is as follows:

- `${TERM}N` — specifies integer to words conversion

Number-to-Word Translator



Number-to-Word Translator

- ▲ In 2010 release (2.9 .NET UI) and higher
 - Triggered by special Text in a calculated report field:
 - `${DataFieldName}N`
 - Data field must be an integer
 - Example:
 - Data field contains value of 273
 - Report output will display “two hundred seventy three”
 - Uses language of the .NET UI user
 - Limitations: no support for Chinese, Polish or Korean. Japanese does not use actual symbols (words are spelled out in English letters)

QAD Proprietary

Exercise



Exercise

- 1) Create a second layout definition for the Sales Order report created in the previous session
 - Group the data by sales order number
 - Structure the layout to have a master-detail page appearance
 - Header fields listed in the Group Header
 - Detail (line) fields indented below header fields
- 2) Make the new layout the default (so that it executes from the menu)
 - Manager tool from Designer program
- 3) Export the report to XML
 - Report Resource Export

QAD Proprietary

Exercises include the following:

- 1 Create a second layout definition for the Sales Order report created in the previous session.
- 2 Make the new layout the default (so that it executes from the menu).
- 3 Export the report to XML.

Sub-Reports



Sub-Reports

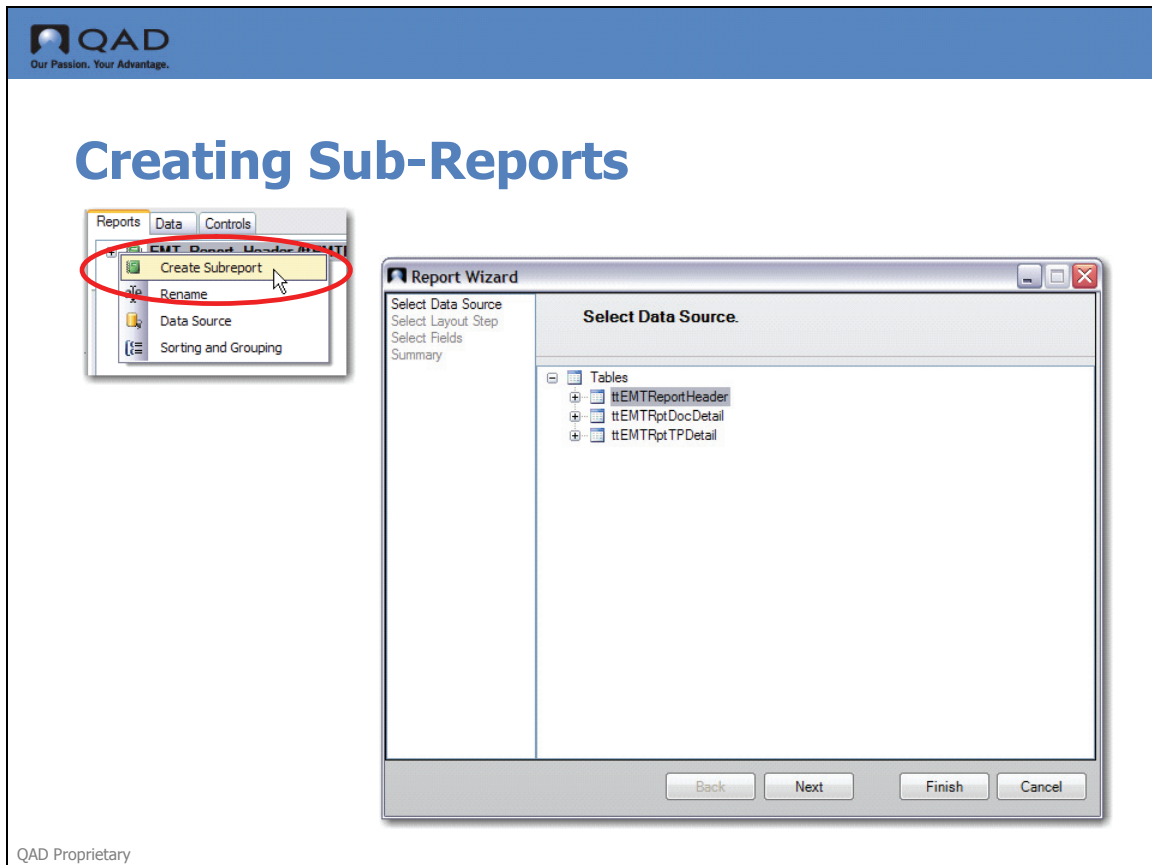
- ▲ Data Source Considerations
 - One report can only access one data table
 - If there are multiple data tables in the data source, then multiple reports are required – use Sub-Reports
- ▲ How to Create
 - Right-click on report in Reports tab – select “Create Subreport”
 - New report Wizard launches
 - After Wizard, click and drag mouse on the desired section to create the sub-report field
 - Link the data fields of parent and child tables
 - Right-click the sub-report field and choose “Link Subreport”
 - Edit sub-report as desired
 - Right-click the sub-report field and choose “Edit Subreport”
 - Or, double-click the sub-report in the Reports tab

QAD Proprietary

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 Sub-Reports



In the Reports tab, choose Create Subreport and then from the Report Wizard, select the data source.

Using Sub-Reports: Tips



Using Sub-Reports: Tips

- ▲ Usually placed in the detail section of parent report
 - The sub-report will then run once per record
 - The sub-report data rows will be the subset of rows in the sub-report's data table, filtered according to the link condition
- ▲ Should generally avoid use of Page Header/PageFooter sections in sub-report
- ▲ VBScript variables defined in parent report can be visible in child sub-reports
 - only if the parent's ExposeScriptObjects property is set to *true*
- ▲ Nesting is allowed
 - Sub-reports can contain sub-reports
 - A report can contain many sub-reports
 - No limit to the size of this tree of reports
- ▲ Common error: forgetting to *link* the sub-report
 - This will result in a large number of records output in the report (every record in the sub-report table will be displayed for each record in the parent table – Cartesian product)

QAD Proprietary

Sub-Reports: Exercise



Sub-Reports: Exercise


- 1) Create a master-detail Sales Order report, using a sub-report with a 2-table data source
 - Report Resource Maintenance
 - Report Code: SalesOrderSubReport
 - Data source: Proxy- UserGuideSampleReport.p
 - Need to first install it into the system
 - Report Resource Designer
 - Use sub-report for detail layout
- 2) Export the report to XML
 - Report Resource Export

QAD Proprietary

Exercises include the following:

- 1 Create a master-detail Sales Order report using a sub-report with a two-table data source.
- 2 Export the report to XML.

VBScript Logic



VBScript Logic

- ▲ **Event Model**
 - VBScript code blocks can be placed into event hooks at the Report and Section levels.
 - VBScript code blocks will fire when the corresponding event occurs during report rendering
 - Can access field and section properties
 - e.g. `header.visible = false`
 - e.g. `soldto_attnLbl.ForeColor = rgb(255,0,0)`
- ▲ **Use in Calculated Fields**
 - Linking to data source fields
 - Built-in function finder, variables (Page, Pages,
- ▲ **Design Philosophy**
 - General rule: Use as little VBScript as needed
 - VBScript makes reports harder to maintain, and harder to convert to other rendering technologies
 - Dynamic Layout Options
 - VBScript to select sub-reports vs. VBScript to control fine-grained fields

QAD Proprietary

Section-level Script settings:

OnFormat. Enter a Visual Basic script that will be executed during the formatting stage of report rendering.

OnPrint. Enter a Visual Basic script that will be executed in the final stage of rendering, after all report field values have been filled.

Report-level Script settings:

OnClose. Enter a Visual Basic script that will be executed when the report finishes rendering.

OnError. Enter a Visual Basic script that will be executed when an error occurs.

OnNoData. Enter a Visual Basic script that will be executed when the report has no data.

OnOpen. Enter a Visual Basic script that will be executed when the report starts rendering.

OnPage. Enter a Visual Basic script that will be executed each time a new page is created.

Design Philosophy for using VBScript:

Less is more! Try to use as little as possible, since it can add to the cost of maintaining the report in the future. VBScript used too much can add a lot of complexity to the report layout design.

VBScript is often necessary for achieving certain dynamic layout goals, for example changing a file's color or visibility based on its data value.

Some reports may need to change the properties of many fields at a time based on data values, even moving their location dynamically. Instead of the brute-force approach of using conditional statements with lots of fine-grained field property setting VBScript statements, it is often preferable to instead use two or more sub-reports, each containing one of the different layout options, and using a single VBScript statement to conditionally determine which sub-report to make visible. This is a more maintainable approach that leads to report designs that are easier to read, understand, and enhance in the future.

VBScript Examples



VBScript Examples

▲ Filtering out data row based on some data value:

- Put VBScript block in Detail section's OnFormat event:

```
if ucase(pt_mstr_pt_part_type) = "FINGOOD" then
    Detail.Visible = false
else
    Detail.Visible = true
endif
```

▲ Changing a field's color based on its data value:

- Put VBScript block in Detail section's OnPrint event:

```
if (pt_price >= 100) then
    'color the field text RED
    [pt_priceCtl].ForeColor = "#FF0000"
else
    'color the field text BLACK
    [pt_priceCtl].ForeColor = "#000000"
endif
```

QAD Proprietary

VBScript Exercise



VBScript: Exercise

- ▲ Perform the techniques for the two VBScript examples given previously.
 - Filtering out data row based on some data value
 - Changing a field's color based on its data value
- ▲ Add summary calculation fields to sum / count field values across groups.
 - Use the grouped Sales Order report
 - sum the quantity ordered for each order
 - count the number of lines for each order
 - count the number of lines across ALL orders
 - Use Calculated Field type in the appropriate section
 - Use aggregate script functions (e.g. Sum()) on the appropriate data field

QAD Proprietary

Exercises include the following:

- 1 Perform the techniques for the two VBScript examples given previously.
- 2 Add summary calculation fields to sum / count field values across groups.

Report Templates



Report Templates

- ▲ A template is a special kind of report that cannot be directly rendered
- ▲ Allows report attributes to be defined in one place (the template) and applied to many reports
- ▲ Changes made to one template will affect all reports that use that template
- ▲ A report can reference at most one template
- ▲ QAD-shipped templates have a "QAD_" prefix and can get overwritten in upgrades. They should not be modified, but copies can be made and safely modified.

QAD Proprietary

A report template is special kind of report definition that cannot be rendered directly by itself, but instead can be used to control certain aspects of the rendering of other reports. When designing a report, a template can be specified (optionally) in which case the report can inherit many kinds of attributes from the template, such as field colors and fonts. If at a later time these attributes are changed in the template, those changes will be seen in every report that is using that template.

Any given report can inherit from at most one template, but a given template may be used to control any number of reports. Thus templates enable report developers to making changes in a single place (the template) which will have a mass effect on many reports. This is a powerful tool that can assist the report development process in many ways, such as reducing initial development time, enforcing common standards across reports, and quickly implementing future changes to these standards.

There are three general types of report properties that can be governed by templates:

- Top-level report properties (e.g. paper size, margins)
- Section properties (e.g. The back color of the PageHeader section)
- Field properties (e.g. The font and ForeColor of fields)

In addition to inheriting properties, reports can also inherit fields from a template. For instance, a template might contain fields in the page header that display date, time, domain, and a corporate logo. These fields will be automatically added to all reports using that template.

Templates can be edited in the Template Designer, which is very similar to the Report Designer program.

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.

Template Behavior




Template Behavior

- ▲ Templates define “Classes” that can be applied to objects in a report definition.
- ▲ Section Classes: sections in the report will inherit:
 - Most properties of the section class
 - BackColor, Visible, CanGrow, CanShrink, ForcePageBreak, KeepTogether, Repeat, OnFormat, OnPrint
 - All fields in the section class
 - only for Header, PageHeader, PageFooter, and Footer sections
- ▲ Field Classes: fields in the report will inherit:
 - Most properties of the field class
 - which properties depends on field type, e.g. Font, BackColor, ForeColor, Height, ...
- ▲ New in QAD 2010 (.NET UI 2.9) release: report-level properties can be inherited from templates

QAD Proprietary

Template Designer



Template Designer

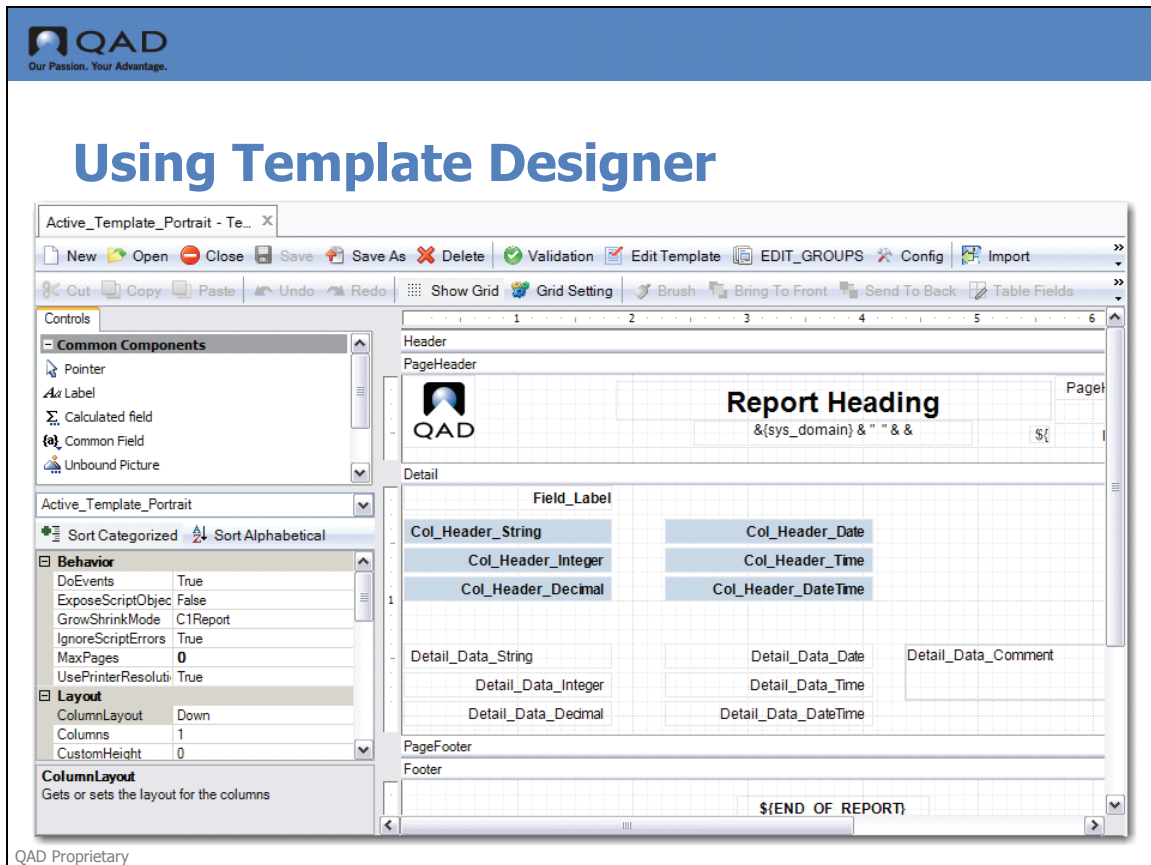
- ▲ Tool to edit templates
 - Accessible in QAD .NET UI interface
 - Requires user membership in 'rptAdmin' or 'rptDsgn' roles/groups
- ▲ Similar to Report Designer, but has some differences
 - Detail Section Behavior
 - Fields in detail section do not get passed to reports
 - Detail fields are only used to define field classes
- ▲ Template Import/Export
 - Similar to report import/export
 - Templates can be serialized as XML

QAD Proprietary

The 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: Edit Template, Add Group, Config, Import, Export.

Using Template Designer



- 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.
- 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.
 - 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.
 - c** Under the Field Configuration tab, for each data type, select a section in the template and specify a class defined within that section.
 - d** Click OK.
- 8** Back in the Template Designer main screen, click the Save button on the toolbar to save the template.

Applying Templates to Report Designs



Applying Templates to Report Designs

- ▲ When designing a report, there are two ways to specify a template:
 - Can assign a template when the Wizard is run to create a new report definition (the “New” tool button runs the Wizard)
 - Can assign a template to an already-created report definition using the “Manager” tool button
- ▲ Class Property
 - Use at section and field levels to bind that section/field to the desired template section/field
 - New report Wizard will auto-assign many classes
 - Report-level properties are automatically linked to template (2010 release and higher)

QAD Proprietary

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.

Class Property

QAD
Our Passion. Your Advantage.

Class Property

The screenshot displays the QAD Reporting Framework interface. The main window shows a report preview with a grid. The report content includes a QAD logo, a page header with the text `$(SALES_ORDI` and `&{sys_domain} & " " &`, a group header with `$(Sales Order)` and `$(Ship-To)`, a detail row with columns `so_mstr_so_nbr`, `so_mstr_so_ship`, `sod_det_sod_due_date`, and `so`, a group footer with `$(END_OF_REPORT)`, and a footer with `$(END_OF_REPORT)`. The left sidebar shows the 'Class Property' dialog for the 'Col_Header_String' class. The 'Template' property is highlighted with a red circle and is set to 'Col_Header_String'. Other properties listed include 'Layout' (Anchor: Top, Bounds: 0, 0, 2910, 300, CanGrow: False, CanShrink: False, ForcePageBreak: None, KeepTogether: False, ZOrder: 0) and 'Class' (Gets or sets the template).

QAD Proprietary

Overriding Template Attributes



Overriding Template Attributes

- ▲ Properties inherited from the template can be manually modified
 - The modified property will be completely decoupled from the template
 - One decoupled, changes made to the template will not affect the de-coupled property
 - To re-couple the template property:
 - Click on the property value in the property grid
 - Click on the “Use Default Value” button at the top of the property grid
 - the value will be refreshed from the template
 - the property will be re-coupled to the template
- ▲ Fields inherited from the template section classes can be modified
 - The modified field will be completely decoupled from the template
 - One decoupled, changes made to the template will not affect the de-coupled field
 - To re-couple all template fields (and properties) for the section:
 - Change the section’s template class to None
 - Then change it again to the original class
 - all template fields will be refreshed from the template.

QAD Proprietary

Template Process Development Guidelines



Template Process Development Guidelines

- ▲ When editing templates, be careful when saving since the changes can affect many reports across the system
- ▲ QAD Active templates (2010 release and higher)

QAD Proprietary

Starting with the QAD Enterprise Applications 2010 release, the QAD standard reports included with the product use one of the following templates:

- `Active_Template_Landscape` —the template typically used by QAD standard reports with landscape orientation.
- `Active_Template_Portrait` — the template typically used by QAD standard reports with portrait orientation.

Additionally, four QAD standard templates are included:

- `QAD_Standard_Template_A4_Landscape` — the QAD standard template for the A4 (210 x 297 mm, or 8.27 x 11.69 inches) paper size with landscape orientation.
- `QAD_Standard_Template_A4_Portrait` — the QAD standard template for the A4 paper size with portrait orientation.
- `QAD_Standard_Template_Letter_Landscape` — the QAD standard template for the Letter (8.5 x 11 inches) paper size with landscape orientation.
- `QAD_Standard_Template_Letter_Portrait` — the QAD standard template for the Letter paper size with portrait orientation.

As shipped, the contents of Active_Template_Landscape are identical to QAD_Standard_Template_Letter_Landscape, and the contents of Active_Template_Portrait are identical to QAD_Standard_Template_Letter_Portrait. If template customizations are desired, they can be done in the Active templates (which the QAD reports reference), and the original QAD_ templates should be kept unmodified in case any changes might need to be rolled back.

Careful management of these templates is highly recommended.

Special Template for Browse Reports



Special Template for Browse Reports

- ▲ Template name: *QAD_Default_Template_Browse*
- ▲ Used for reports created on-the-fly from running .NET UI Browsers
 - Any end user can invoke such a report from a browse's Actions -> Report menu item.
 - Feature available in the QAD 2009.1 release
- ▲ Customers can modify this template to tailor the cosmetic appearances of these browse reports
 - Be sure to back up any changes using the export tool since this template will get overwritten in upgrades

QAD Proprietary

Templates: Exercise



Templates: Exercise

- 1) Modify (or create) a report definition that does not have a template specified, using *Report Resource Designer*
 - Use the Manager tool to specify a template
 - Assign classes to sections and fields
 - Override some field properties for a data field
 - Override some field properties from a template-inherited field
- 2) Create a new report definition, specifying a template in the New Report Wizard
 - Examine the class assignments made by the Wizard, changing if desired
- 3) Modify the template used in #1 above, using *Template Designer*
 - Modify some field colors / fonts, including for the overridden fields. Re-run the report and verify that the changes only take effect for non-overridden fields.
 - Re-couple the overridden fields to the template, and re-run the report.
 - Add a field to the PageHeader section, and change the BackColor of that section
 - Open the report in the Designer and run it; is it as expected?
 - Remove the template class for the PageHeader section and then re-assign it. Does the report run differently now?

QAD Proprietary

Exercises include the following:

- 1 Modify (or create) a report definition that does not have a template specified, using Report Resource Designer
- 2 Create a new report definition, specifying a template in the New Report Wizard
- 3 Modify the template used in #1 above, using Template Designer

Summary



Summary

In this section you have learned how to:

- ▲ Create and modify report page layouts using the QAD Reporting Framework's Designer program
- ▲ Use sub-reports to work with multi-table data sources
- ▲ Add VBScript logic for dynamic layout modification
- ▲ Use and create templates, allowing standardization and mass changes across reports

QAD Proprietary

In this section you have learned how to:

- Create and modify report page layouts using the QAD Reporting Framework's Designer program
- Use sub-reports to work with multi-table data sources
- Add VBScript logic for dynamic layout modification
- Use and create templates, allowing standardization and mass changes across reports

Chapter 3

Progress Data Source Program

Session Objectives



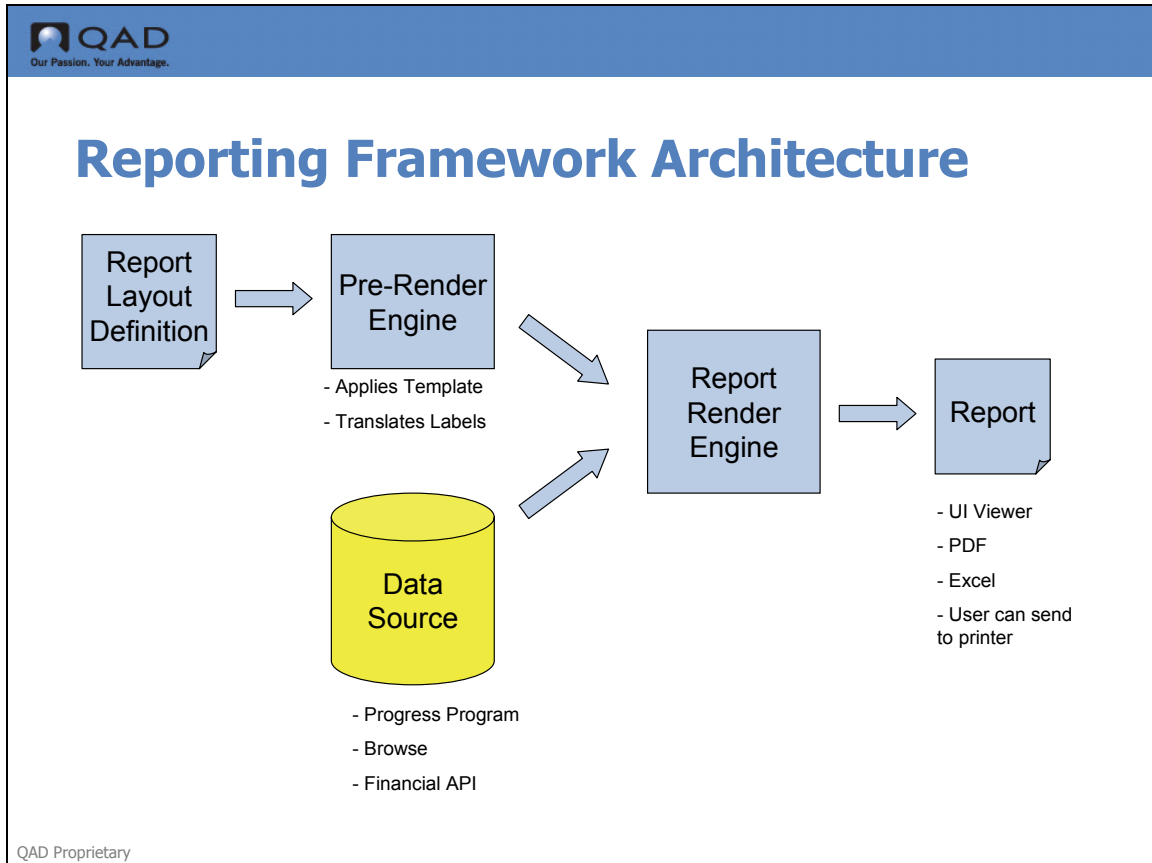
Session Objectives

In this session you will learn how to:

- ▲ Develop a Progress “Proxy” Data Source Program
- ▲ Deploy the data source program into the run-time environment

QAD Proprietary

Reporting Framework Architecture: Data Source



There are three types of data sources that a report can use: Browse, FinancialAPI, and Progress Program. Our focus in this session is on developing and deploying a Progress Program – a proxy data source program.

Developing a Progress Data Source Program

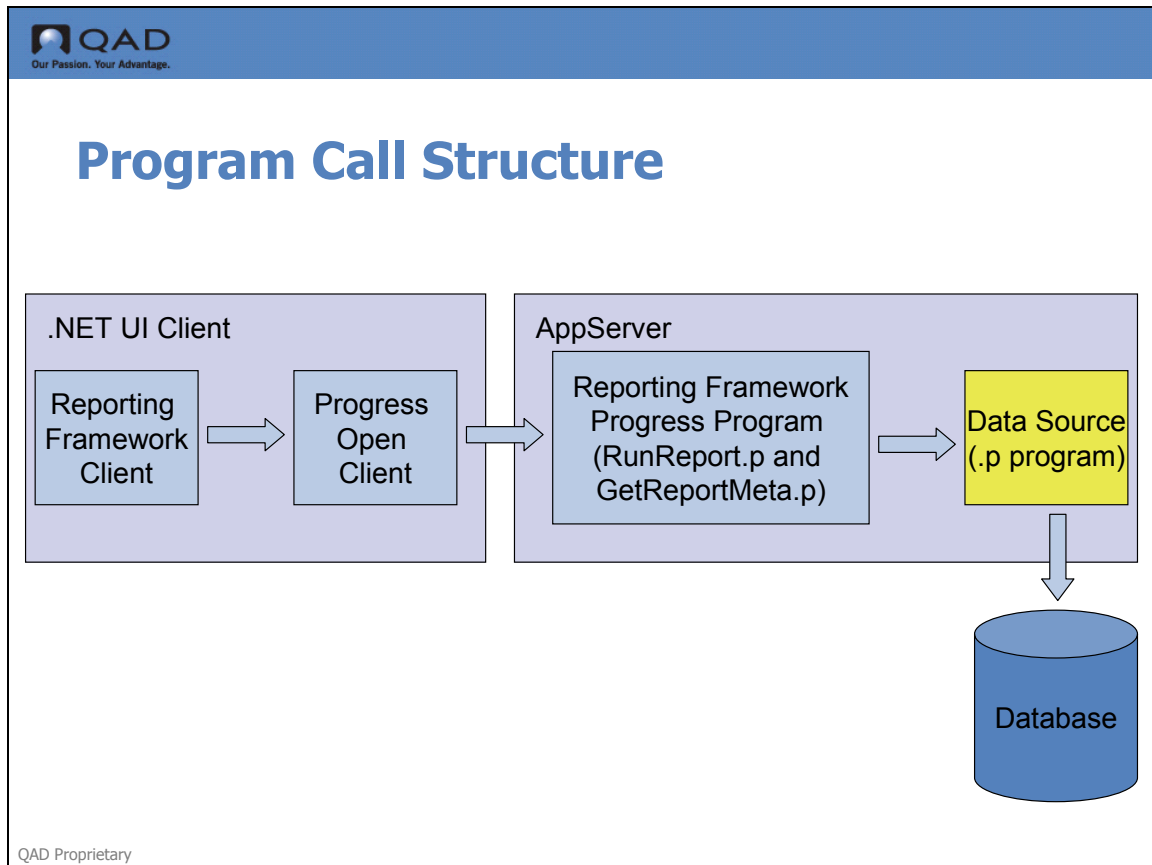


Developing a Progress Data Source Program

- ▲ Program runs on the .NET UI Progress Application Server tier
- ▲ Two types of client requests must be supported
 - Metadata request
 - Information about data set tables and fields
 - Field attributes drive client behavior on prompt page
 - Data request
 - Query business data, restricting by filter conditions
- ▲ Generic entry programs: RunReport.p, GetReportMeta.p
 - All client requests go through GetReportMeta.p for metadata
 - All client requests go through RunReport.p for data query
 - RunReport.p and GetReportMeta.p delegate the call to the specified data source program

QAD Proprietary

Program Call Structure



This diagram illustrates the context in which a Progress data source program is run.

Progress Data Source Program Calls



Progress Data Source Program Calls

- ▲ A data source program requires four blocks of code to be written:
 - 1. A data set definition
 - Consists of one or more temp-tables that define the data set structure for the report
 - Tables can be related (e.g. master-detail)
 - 2. Statements to empty the temp-table(s) in the data set
 - 3. Metadata definition that defines attributes for the fields in the data set
 - 4. Data retrieval logic that populates the data set according to filter conditions

QAD Proprietary

A data source program comprises four blocks of code:

- A data set definition (consisting of one or more temp-tables) that defines the data set structure for the report
- Statements to empty the temp-table(s) in the data set
- Metadata definition that defines attributes for the fields in the data set
- Data retrieval logic that populates the data set

Overall Code Structure - Page 1



Overall Code Structure – Page 1

```

{mfdeclre.i}
{gplabel.i}

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

/* Report data set definition block */

/* TODO Insert your data set code here */

/* 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 block */

/* TODO empty the temp-table(s) */

for first ttReportRequest no-lock:
  run FillMetaData.

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

```

Continued ...

QAD Proprietary

In order for the program to be invoked properly by RunReport.p, it must have a specific structure. The following sample code illustrates this structure, showing where the four blocks of code should be inserted:

```

{mfdeclre.i}
{gplabel.i}

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

/* Report data set definition block */

/* TODO Insert your data set code here */

/* 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 the temp-table(s) */

for first ttReportRequest no-lock:

  run FillMetaData.

  if runReport then do:

```

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

end.

/* Metadata definition block */

procedure FillMetaData:

    /* TODO Insert your metadata code here */

end procedure.

/* Data retrieval logic block */
procedure RunReport:

    define output parameter dataset-handle phReportResults.

    /* TODO Insert your data retrieval code here */

    phReportResults = dataset dsReportResults:handle.

end procedure.
```

The code contains several comments starting with `TODO`. These comments indicate where the code blocks should be inserted. The following sections will discuss how to write the code blocks to be inserted into the above structure.

Overall Code Structure - Page 2



Overall Code Structure – Page 2

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

                    /* TODO Insert your metadata code here */

                end procedure.

                /* Data retrieval logic block */
                procedure RunReport:

                    define output parameter dataset-handle
                    phReportResults.


                    /* TODO Insert your data retrieval code
                    here */

                    phReportResults = dataset
                    dsReportResults:handle.

                end procedure.
```

QAD Proprietary

1. Data Set Definition Block - Example



1. Data Set Definition Block - Example

```

/* Temp-table definition block */
define temp-table ttSalesHeader
  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_slspn1 like so_mstr.so_slspn[1]
  field sales_order_slspn2 like so_mstr.so_slspn[2]
  field sales_order_slspn3 like so_mstr.so_slspn[3]
  field sales_order_slspn4 like so_mstr.so_slspn[4]

  index SalesHeaderIdx is primary so_nbr
.

define temp-table ttSoLine
  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)

```

QAD Proprietary

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.

Note Indexes specified for the temp tables can be used to define unique constraints and the default sort order of the records.

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

2. Table Emptying Code - Example



2. Table Emptying Code - Example

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

QAD Proprietary

Each temp-table in the report data set should be emptied before the main program logic is executed. This is necessary since temp-table contents could still be cached on the Application Server from previous client requests.

ReportHelper.p Program



ReportHelper.p Program

- ▲ Performs tasks that are common to all data source programs
 - Initializes Metadata data set
- ▲ Provides utility functions / procedures
 - Makes specific data source program code more succinct
 - e.g. functions to assist in creating metadata
 - e.g. functions to assist with building dynamic filter condition strings for 'where' clause.
- ▲ Run as a persistent procedure
 - Handle is passed into data source program as an input parameter

QAD Proprietary

3. Metadata Definition Code Block



3. Metadata Definition Code Block

- ▲ Needs to fill the metadata output data set
 - Define Buffer Header (for each table)
 - Use `CreateBufferHeader` from `ReportHelper.p`
 - Define Field Metadata
 - Use `CreateField` from `ReportHelper.p` (and similar procedures)
 - Specify Lookups
 - Translation Considerations
 - Field Labels
 - Value Lists
 - Logical-Typed `fieldFormat`

QAD Proprietary

Metadata: Using CreateBufferHeader



Metadata: Using CreateBufferHeader

- ▲ **CreateBufferHeader procedure**
 - Creates the metadata that describes a table
 - Must be run once for each temp-table in the report data set
- ▲ **Parameters:**
 - tableName
 - tableLabel
- ▲ **Example:**

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

QAD Proprietary

Table 3.1
CreateBufferHeader Parameters

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

Metadata: Using CreateField



Metadata: Using CreateField

- ▲ **CreateField procedures**
 - Create the metadata that describes a field
 - Three variants:
 - **CreateField** - allows the programmer to explicitly pass in the values of each metadata parameter.
 - **CreateFieldForDBField** - can be used if the field is similar to one in the business database
 - Some metadata parameters will be determined by the system based on the database field.
 - e.g. field label, data type, field format, and lookup name
 - **CreateFieldLikeDBField** - similar to **CreateFieldForDBField**
 - Allows the field name of the report field to be different from that of the database field.

QAD Proprietary

Either of three predefined procedures can be used to create field metadata. The procedures are similar and have the same effect of creating a metadata record for one field. A description of each is given below.


CreateField — this variant allows the programmer to explicitly pass in the values of each metadata parameter.

CreateFieldForDBField — this variant can be used if the field is similar to one in the business database, in which case some of the metadata parameters will be determined by the system based on the database field. For example, the field label, data type, field format, and lookup name will be driven by the database field.

CreateFieldLikeDBField — this variant is almost identical to **CreateFieldForDBField** except that it allows the field name of the report field to be different from that of the database field.

(The following sections list the input parameters for each of these three procedures.)

Metadata: CreateField



Metadata: CreateField

- ▲ CreateField input parameters include:
 - bufferName
 - fieldName
 - fieldLabel
 - dataType
 - fieldFormat
 - lookupName
 - isSearchField
 - isReadOnlySearch
 - isVisible
 - isSingleEntry
 - isOperatorChangeable
 - isRequiredCondition
 - isEditable
 - defaultValue
 - defaultOperator
 - defaultValueType
 - defaultValue2
 - defaultValueType2


QAD Proprietary

Table 3.2
CreateField Parameters

Name	Input/Output	Data Type	Description
bufferName	Input	Character	Temp-table name
fieldName	Input	Character	Report field name
fieldLabel	Input	Character	User-facing label for this field, as it will appear on the prompt page of the report viewer
dataType	Input	Character	Progress datatype of the field
fieldFormat	Input	Character	Progress format for the field. In the case of logical-typed fields, the field format string is used to specify the user-facing labels for the true and false values the field can hold; the syntax is: <code><trueLabel>/<>falseLabel></code> . For example, "Debit/Credit". Note: in a multi-language system, these values should not be hard coded but instead dynamically translated using the <code>getLabel</code> function. The format strings will be used in search condition values on the prompt page, as well as data values in the report output.
lookupName	Input	Character	Program name of the lookup program (if any) that will be invoked from the lookup icon for this field on the prompt page of the report viewer. For example, "gplu340.p".

Name	Input/Output	Data Type	Description
isSearchField	Input	Logical	Whether this field should be a search field on the prompt page
isReadOnlySearch	Input	Logical	Whether this field is read-only on the prompt page
isVisible	Input	Logical	Whether this field is visible in Report Designer
isSingleEntry	Input	Logical	Always set this to False
isOperatorChangeable	Input	Logical	Whether the operator can be changed on the prompt page
isRequiredCondition	Input	Logical	Whether the field is mandatory on the prompt page
isEditable	Input	Logical	Whether the field can be edited on the prompt page
defaultValue	Input	Character	Default value of the first search field
defaultOperator	Input	Character	Default operator of the first search field
defaultValueType	Input	Character	Default value type of the first search field
defaultValue2	Input	Character	Default value of the second search field
defaultValueType2	Input	Character	Default value type of the second search field

Metadata: CreateFieldForDBField



Metadata: CreateFieldForDBField

▲ CreateFieldForDBField parameters include:

- bufferName
- tableName
- fieldName
- isSearchField
- isReadOnlySearch
- isVisible
- isSingleEntry
- isOperatorChangeable
- isRequiredCondition
- isEditable
- defaultValue
- defaultOperator
- defaultValueType
- defaultValue2
- defaultValueType2

QAD Proprietary

Table 3.3
CreateFieldForDBField Parameters

Name	Input/Output	Data Type	Description
bufferName	Input	Character	Temp-table name
tableName	Input	Character	QAD ERP database table name
fieldName	Input	Character	QAD ERP database field name
isSearchField	Input	Logical	Whether this field should be a search field on the prompt page
isReadOnlySearch	Input	Logical	Whether this field is read-only on the prompt page
isVisible	Input	Logical	Whether this field is visible in Report Designer
isSingleEntry	Input	Logical	Always set this to False
isOperatorChangeable	Input	Logical	Whether the operator can be changed on the prompt page
isRequiredCondition	Input	Logical	Whether the field is mandatory on the prompt page
isEditable	Input	Logical	Whether the field can be edited on the prompt page
defaultValue	Input	Character	Default value of the first search field
defaultOperator	Input	Character	Default operator of the first search field
defaultValueType	Input	Character	Default value type of the first search field
defaultValue2	Input	Character	Default value of the second search field
defaultValueType2	Input	Character	Default value type of the second search field

Metadata: CreateFieldLikeDBField



Metadata: CreateFieldLikeDBField

- ▲ CreateFieldLikeDBField parameters include:
- bufferName
 - fName
 - tableName
 - fieldName
 - isSearchField
 - isReadOnlySearch
 - isVisible
 - isSingleEntry
 - isOperatorChangeable
 - isRequiredCondition
 - isEditable
 - defaultValue
 - defaultOperator
 - defaultValueType
 - defaultValue2
 - defaultValueType2

QAD Proprietary

Table 3.4
CreateFieldLikeDBField Parameters

Name	Input/Output	Data Type	Description
bufferName	Input	Character	Temp-table name
fName	Input	Character	Field name in the temp-table (report field name)
tableName	Input	Character	QAD ERP database table name
fieldName	Input	Character	QAD ERP database field name
isSearchField	Input	Logical	Whether this field should be a search field on the prompt page
isReadOnlySearch	Input	Logical	Whether this field is a read-only on the prompt page
isVisible	Input	Logical	Whether this field is visible in Report Designer
isSingleEntry	Input	Logical	Always set this to False
isOperatorChangeable	Input	Logical	Whether the operator can be changed on the prompt page
isRequiredCondition	Input	Logical	Whether the field is mandatory on the prompt page
isEditable	Input	Logical	Whether the field can be edited on the prompt page
defaultValue	Input	Character	Default value of the first search field
defaultOperator	Input	Character	Default operator of the first search field
defaultValueType	Input	Character	Default value type of the first search field
defaultValue2	Input	Character	Default value of the second search field
defaultValueType2	Input	Character	Default value type of the second search field

Metadata: CreateField - Example



Metadata: CreateField - Example

```
/* The so_nbr field is set as a search field, so the seventh  
parameter is True. */
```

```
run CreateField in reportHandle  
  (bufferName,  
   "so_nbr",  
   getLabel("SALES_ORDER"),  
   "character",  
   "x(8)",  
   "gplu239.p",  
   true,  
   false,  
   true,  
   false,  
   true,  
   false,  
   true,  
   "",  
   {&ParameterOperator_Equals},  
   {&ParameterValue_Type_Constant},  
   "",  
   {&ParameterValue_Type_Constant}).
```

QAD Proprietary

Metadata: CreateFieldForDBField - Example



Metadata: CreateFieldForDBField - Example

```
/* The following field will have some of its metadata
attributes */
/* driven from the so_cust database field. */

run CreateFieldForDBField in reportHandle
  (bufferName,
   "so_mstr",
   "so_cust",
   true,
   false,
   true,
   false,
   true,
   false,
   true,
   "",
   {&ParameterOperator_Equals},
   {&ParameterValue_Type_Constant},
   "",
   {&ParameterValue_Type_Constant}).
```

QAD Proprietary

Metadata: CreateFieldLikeDBField - Example



Metadata: CreateFieldLikeDBField - Example

```
/* The following field will be called order_date in the report
even though some */
/* of its metadata attributes will be driven from the
so_ord_date database field. */
```

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

QAD Proprietary

Metadata: Lookup Syntax



Metadata: Lookup Syntax

- ▲ Short-hand Notation:
 - Can be used if the desired lookup is a browse defined by Browse Maintenance
 - e.g. "gplu239.p" – denotes browse gp239 as lookup
- ▲ Complete Notation:
 - Must be used if the browse is from a different server implementation (e.g. EE Financial query)
 - *<lookup provider type>:<lookupID>:<lookup return field>:<lookup filter field>*
 - e.g.
`BaseLibrary.Lookup.BLFLookupProvider:BJournalSAO.SelectJournal:tcJournalCode:tJournal.JournalCode`

Metadata: Value Lists



Metadata: Value Lists

- ▲ Allows values for a *character*-typed field to be restricted to a discrete set of values
- ▲ Each *value* can be accompanied with a user-facing *label* that can appear in two places:
 - List of values in filter condition (viewer prompt page)
 - Report output document
 - The *value* is what is stored in the DB
- ▲ Use the `valueList` field metadata attribute
- ▲ Syntax:
 - `<Label 1>,<DB value 1>,<Label 2>,<DB value 2>, ...`

QAD Proprietary

Metadata: Value Lists - Example



Metadata: Value Lists - Example

```
define variable statusValueList as
character no-undo.

statusValueList = "New,1,In
Process,2,Completed,3".

run SetFieldMetaParam in ReportHandle(
    bufferName,
    "status",
    "valueList",
    statusValueList)
```

- ▲ **Note:** SetFieldMetaParam is a procedure in ReportHelper.p that can be used to set field metadata attributes
 - One of the CreateField procedures must first have been called to create the record

Metadata: Translations



Metadata: Translations

- ▲ Most translations for reports are done in the page layout design (Report Resource Designer)
- ▲ There are also some areas that may require translated text in the metadata returned by a data source program:
 - Field Labels - `fieldLabel` metadata attribute
 - Value Lists - `valueList` metadata attribute
 - Logical-Typed value labels - `fieldFormat` metadata attribute
- ▲ Use `global_user_lang` shared variable to determine language to use for getting translated labels
- ▲ New in QAD 2010 (.NET UI 2.9) release: `getLabel()` utility function in `ReportHelper.p`
 - Input: term key string
 - Output: translated long label, using `global_user_lang`
 - Example: `getLabel("SALES_ORDER")`
 - Function definition:

```
FUNCTION getLabel returns character (
  input pTerm as character):
```

4. Data Retrieval Logic Code Block



4. Data Retrieval Logic Code Block

- ▲ Used to populate the report data set tables
- ▲ Must be implemented in procedure `RunReport`
- ▲ Need to Use Dynamic Query
 - To handle filter conditions from client request
 - `FillQueryStringVariable` helper procedure
- ▲ Query Iteration and Temp Table Population

QAD Proprietary

Data Retrieval - Defining the Query



Data Retrieval – Defining the Query

```
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 "order_date", input-output
queryString).

queryString = queryString + ":".

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

QAD Proprietary

FillQueryStringVariable Procedure



FillQueryStringVariable Procedure

- ▲ Helper procedure from ReportHelper.p
- ▲ Used to add dynamic filter conditions to the query string
- ▲ Will construct a where-clause segment and append it to the query string with a logical `and`
 - Inspects client request data set to read filter conditions
- ▲ Call signature:

```
PROCEDURE FillQueryStringVariable:
    define input parameter bufferName as character no-undo.
    define input parameter fieldName as character no-undo.
    define input-output parameter queryString as character
    no-undo.
```

- ▲ Input parameters include:
 - `bufferName` (temp-table name)
 - `fieldName` (field name in temp-table)
 - `queryString` (dynamic query string)

QAD Proprietary

The `FillQueryStringVariable` function will get the search conditions sent from the client request (each consisting of the search field, operator, and value entered by the user) and then construct the corresponding where clause fragment and add it to the query string dynamically at run time.

The following lists the input parameters for the `FillQueryStringVariable` function:

Name	Input/Output	Data Type	Description
<code>bufferName</code>	Input	Character	Temp-table name
<code>fieldName</code>	Input	Character	Field name in the temp-table
<code>queryString</code>	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.

Note If other static filter conditions are desired in the query string, they can be added in place of the “where true” part of the query string in the above example. For example, if we wanted this report to filter its query to only include orders whose `so_site` value is “SITE1”, we could use the following statement to begin the query string:

```
queryString = "for each so_mstr no-lock where so_site = " + QUOTER("SITE1").
```

The `QUOTER()` function returns the input string wrapped in quotes, and is useful when specifying text in a dynamic query string that must appear quoted in the final query string.

If static sorting is desired, a “by” clause can be added at the end of the dynamic query string. For example, the following change to our above example will cause records to be sorted primarily by `so_cust`:

```
queryString = queryString + " by so_cust:".
```

Static where-clause Conditions



Static where-clause Conditions

- ▲ Static filter conditions can be hard coded into the beginning of the query string

- Example: Filter query to only include orders for "SITE1"

```
queryString = "for each so_mstr no-lock
where so_site = "
+ QUOTER("SITE1").
```

- If no static conditions are needed, add a dummy one ("where true") so that and-ed dynamic conditions are appended properly

- ▲ For static sorting, add a 'by'-clause at the end of the query string

- Example: Sort records by customer

```
queryString = queryString + " by so_cust:".
```

QAD Proprietary

NOTE: The QUOTER() function returns the input string wrapped in quotes, and is useful when specifying text in a dynamic query string that must appear quoted in the final query string.

GetFilterValue Function



GetFilterValue Function

- ▲ Helper procedure from ReportHelper.p
- ▲ Used to retrieve filter values from input request
- ▲ Call signature:

```
FUNCTION GetFilterValue returns character  
  (input ipBufferName as character,  
   input ipFieldName as character,  
   input defaultValue as character):
```

- ▲ Input parameters include:
 - ipBufferName (temp-table name)
 - ipFieldName (field name in temp-table)
 - defaultValue (value to return if none found in request)
- ▲ Returns: Field value from filter conditions in request

QAD Proprietary

GetFilterValue - Example



GetFilterValue - Example

```
define var orderLineFilterValue as integer no-undo.  
  
orderLineFilterValue =  
integer(GetFilterValue("ttReportOptions", "sod_line",  
"1")).
```

QAD Proprietary

Data Retrieval - Query Iteration



Data Retrieval – Query Iteration

- ▲ Iterate over the query records and populate the temp tables of the report data set
- ▲ Example...

QAD Proprietary

Data Retrieval - Query Iteration Example



Data Retrieval – Query Iteration Example

```

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_slspn1 = so_mstr.so_slspn[1]
    ttSalesHeader.sales_order_slspn2 = so_mstr.so_slspn[2]
    ttSalesHeader.sales_order_slspn3 = so_mstr.so_slspn[3].
    ttSalesHeader.sales_order_slspn4 = so_mstr.so_slspn[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 */

```

QAD Proprietary

Progress Data Source Deployment



Progress Data Source Deployment

- ▲ Progress data source programs are deployed on the .NET UI Progress application server tier
 - `<desktop source code directory>/com/qad/shell/report/reports`
 - `<desktop source code directory>` usually is:
`/qad/web/server/docs/<ENVname>/ebdesktop2/<WEBAPPNAME>/`
- ▲ Compile the program and save
 - Must be connected to `qaddb` and `qadadm` DBs
 - Can use the `mkdt` program in `<desktop source code directory>`
 - `./mkdt compile`

QAD Proprietary

Note If you prefer to physically locate your Progress data source programs in another directory path, that is fine. Just be sure to realize that at run-time the reporting system will attempt to locate the program using the path `com/qad/shell/report/reports/<DataSourceRef>`, where `<DataSourceRef>` is the value in the Data Source Ref field in Report Resource Maintenance. Therefore, make sure that you name your file accordingly, put your alternate directory into the application server's propath, and place your data source program in a `com/qad/shell/report/reports` subdirectory under your alternate location.

Progress Data Source Deployment: Propath Example



Progress Data Source Deployment Propath Example

▲ Example propath for compilation:

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

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

```
propath = propath + "," +  
"/qad/web/server/docs/93/ebdesktop2/dev93ui/com/qad/shell/r  
eport".
```

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

Exercise



Exercise

- 1) Install and compile the sample .p data source programs given in the course materials
 - HelloWorldReport.p
 - SimpleItemReport.p
 - TimeDelayReport.p
 - UserGuideSampleReport.p
- 2) Import the XML layout definitions for the sample reports
 - Report Resource Import
- 3) Test run each from Report Resource Designer program
- 4) Modify the SimpleItemReport.p program to add a Unit of Measure field
 - Add field to .p to contain pt_mstr.pt_um
 - Add field to page layout in Designer
 - Add it to the SimpleItemReport_Grouped layout

QAD Proprietary

- 1 Install and compile the sample .p data source programs given in the course materials:
 - HelloWorldReport.p
 - SimpleItemReport.p
 - TimeDelayReport.p
 - UserGuideSampleReport.p
- 2 Import the XML layout definitions for the sample reports:
 - Report Resource Import
- 3 Test run each from Report Resource Designer program.
- 4 Modify the SimpleItemReport.p program to add a Unit of Measure field:
 - Add field to .p to contain pt_mstr.pt_um
 - Add field to page layout in Designer
 - Add it to the SimpleItemReport_Grouped layout

Summary



Summary

In this section, you have learned how to...

- Develop a Progress “Proxy” Data Source Program
- Deploy the data source program into the run-time environment

QAD Proprietary

In this section, you have learned how to:

- Develop a Progress “Proxy” Data Source Program.
- Deploy the data source program into the run-time environment.

Chapter 4

Reporting Framework Administration

Session Objectives



Session Objectives

In this session you will learn how to:

- ▲ Configure the QAD Reporting Framework
- ▲ Set up user authorization for report development and admin programs
- ▲ Manage reports on the system menu
- ▲ Administer scheduled report servers

QAD Proprietary

System Configuration: client-session.xml



System Configuration: client-session.xml

- ▲ Several reporting-related settings must exist in the client-session.xml configuration file on the .NET UI home server
- ▲ File location:
 - `<TomcatInstallDir>/webapps/qadhome/configurations/<config>/client-session.xml`
- ▲ Reporting-related entries:
 - ReportDataSourceProviders
 - Tells system how to load data source implementation classes
 - Allows custom implementations to be added
 - SMTP settings
 - For email notifications from scheduled report runs

QAD Proprietary

It is possible to write a new custom data source provider implementation. For example, you might have a non-QAD data store that you wish to report against. Write a .NET class that implements the `IDatasourceProvider` interface, deploy it in the .NET UI environment, and register it by adding a `<Provider>` section to `client-session.xml`. The `Name` is arbitrary (and will appear in the data source options in `ReportResourceMaintenance`), and the `Assembly` and `Class` name must match the DLL and class name (with full namespace) that you developed.

Report Data Source Provider Settings



Report Data Source Provider Settings

- ▲ Registers report data source provider implementations
- ▲ The following is the entry as currently shipped:

```
<ReportDataSourceProviders>
  <Provider>
    <Name>GenericProxy</Name>
    <Assembly>QAD.Plugin.Reports.ReportFramework</Assembly>
    <Class>QAD.Plugin.Reports.ReportFramework.GenericProxyDatasourceProvider</Class>
  </Provider>
  <Provider>
    <Name>Browse</Name>
    <Assembly>QAD.Browse</Assembly>
    <Class>QAD.Browse.BrowseDatasourceProvider</Class>
  </Provider>
  <Provider>
    <Name>FinancialAPI</Name>
    <Assembly>BaseLibrary</Assembly>
    <Class>BaseLibrary.Reports.BLFDatasourceProvider</Class>
  </Provider>
</ReportDataSourceProviders>
```

QAD Proprietary

It is possible to write a new custom data source provider implementation. For example, you might have a non-QAD data store that you wish to report against.

- 1 Write a .NET class that implements the `IDatasourceProvider` interface
- 2 Deploy it in a DLL in the .NET UI environment
- 3 Add a generalized code entry for it: `FieldName=rptres_datasource_type`, `Value=any integer` (safest to pick one >100 to avoid future QAD conflicts), `Comments:arbitrary`; will appear in the code lookup in RR Maint
- 4 Register it by adding a `<Provider>` section to `client-session.xml`. The `Name` should match the integer chosen for the code value in step 3, and the `Assembly` and `Class` name must match the DLL and class name (with full namespace) that you developed.

Data Source Provider Settings - Change For Financials



Data Source Provider Settings – Change For Financials

- ▲ The entry for the FinancialAPI changed between 2009 EE and 2009.1 EE releases

- Entry for 2009 EE:

```
<Provider>
  <Name>FinancialAPI</Name>
  <Assembly>BaseLibrary</Assembly>
  <Class>BaseLibrary.Reports.BLFDatasourceProvider</Class>
</Provider>
```

- Entry for 2009.1 EE:

```
<Provider>
  <Name>FinancialAPI</Name>
  <Assembly>BaseAdapters.Reporting</Assembly>
  <Class>BaseAdapters.Reporting.DatasourceProvider.BLFDatasourceProvider</Class>
</Provider>
```

rptAdmin and rptDsgn Roles / Groups



rptAdmin and rptDsgn Roles / Groups


- ▲ These roles are used to grant access to report development and admin programs
- ▲ rptAdmin membership allows access to all of the reporting programs
- ▲ rptDsgn membership allows access to a subset of the programs
 - Appropriate for report developers
- ▲ Admin Programs for User Roles (EE):
 - Role View, Role Create, User Role View, Role Membership Maintain
- ▲ Admin Programs for User Groups (SE):
 - User Group Maintenance

QAD Proprietary

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.

Program Authorization Matrix



Program Authorization Matrix

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

QAD Proprietary

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

Table 4.1
Program-Role Access Control Matrix

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

Adding a Report to the Menu



Adding a Report to the Menu

- ▲ Use standard Menu System Maintenance
 - Exec Procedure must be a URN of the format:
`urn:qad-report:c1:<ReportCode>`
- ▲ If the report has multiple layout definitions, the one marked as the *default* will execute from the menu.
- ▲ Menu security is done in standard fashion
 - EE: Role Permissions Maintain
 - SE: Menu Security Maintenance

QAD Proprietary

The ReportCode value in the URN must match the one entered into Report Resource Maintenance for the desired report.

Exercise



Exercise

- 1) Locate the client-session.xml file on your system
- 2) View the file (use `vi` or `more`)
 - Locate the `ReportDataSourceProviders` section
 - Locate the SMTP section
- 3) Launch *User Role View* and determine which users are in the `rptAdmin` and `rptDsgn` groups

QAD Proprietary

Exercises include the following:

- 1 Locate the client-session.xml file on your system.
- 2 View the file (use `vi` or `more`):
 - Locate the `ReportDataSourceProviders` section
 - Locate the SMTP section
- 3 Launch User Role View and determine which users are in the `rptAdmin` and `rptDsgn` groups.

Scheduled Reports



Scheduled Reports

- ▲ Report Server(s) run queues of reports (batches)
- ▲ Server processes
 - Non-GUI .NET UI processes
 - Standard installation of .NET UI
 - Launched from command line
- ▲ Windows Task Scheduler used to periodically launch batches (e.g. daily, monthly)
- ▲ Output to printer and/or file on web server
- ▲ Optional notifications to email, .NET UI Inbox

QAD Proprietary

NOTE: QRF scheduled report servers should not be confused with the financial report daemons in the EE product. The latter is separate from the QRF and is used in conjunction with the Crystal Reports for EE financials.

Scheduled Reports: Installation and Deployment



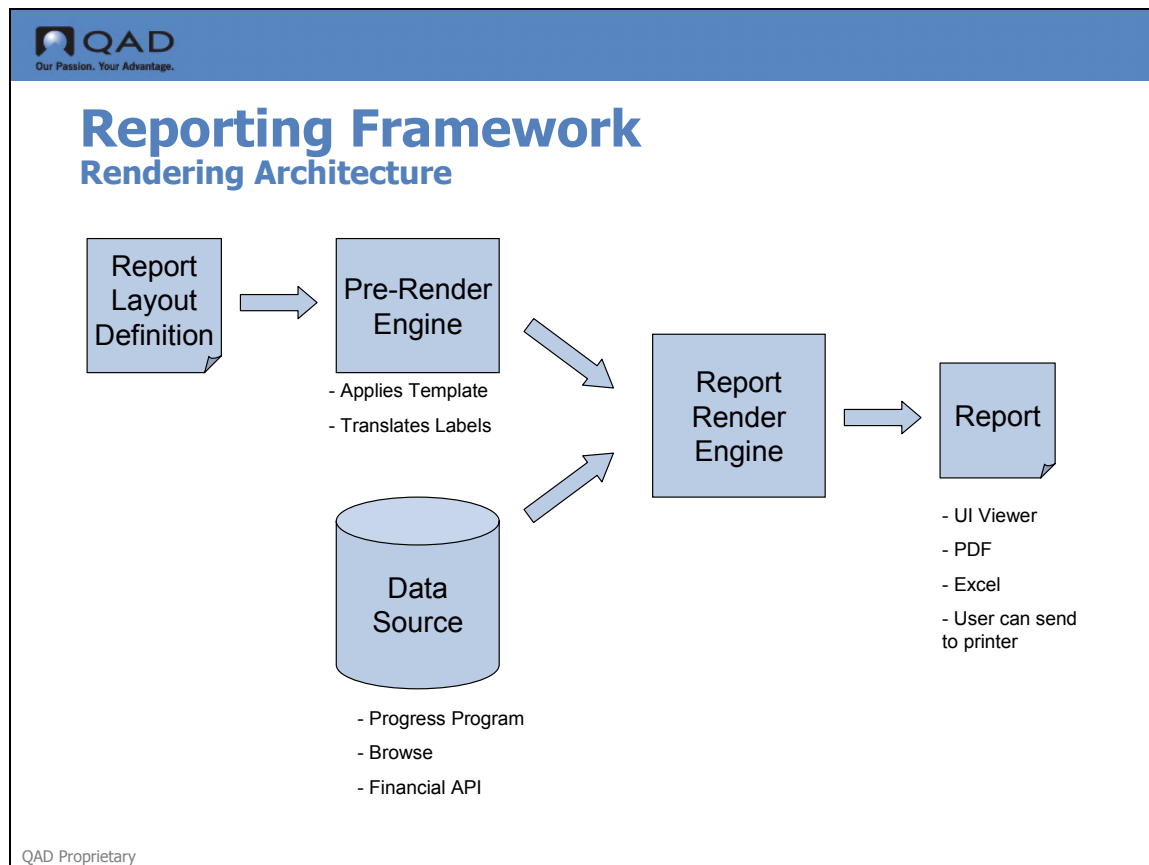
Scheduled Reports

Installation and Deployment

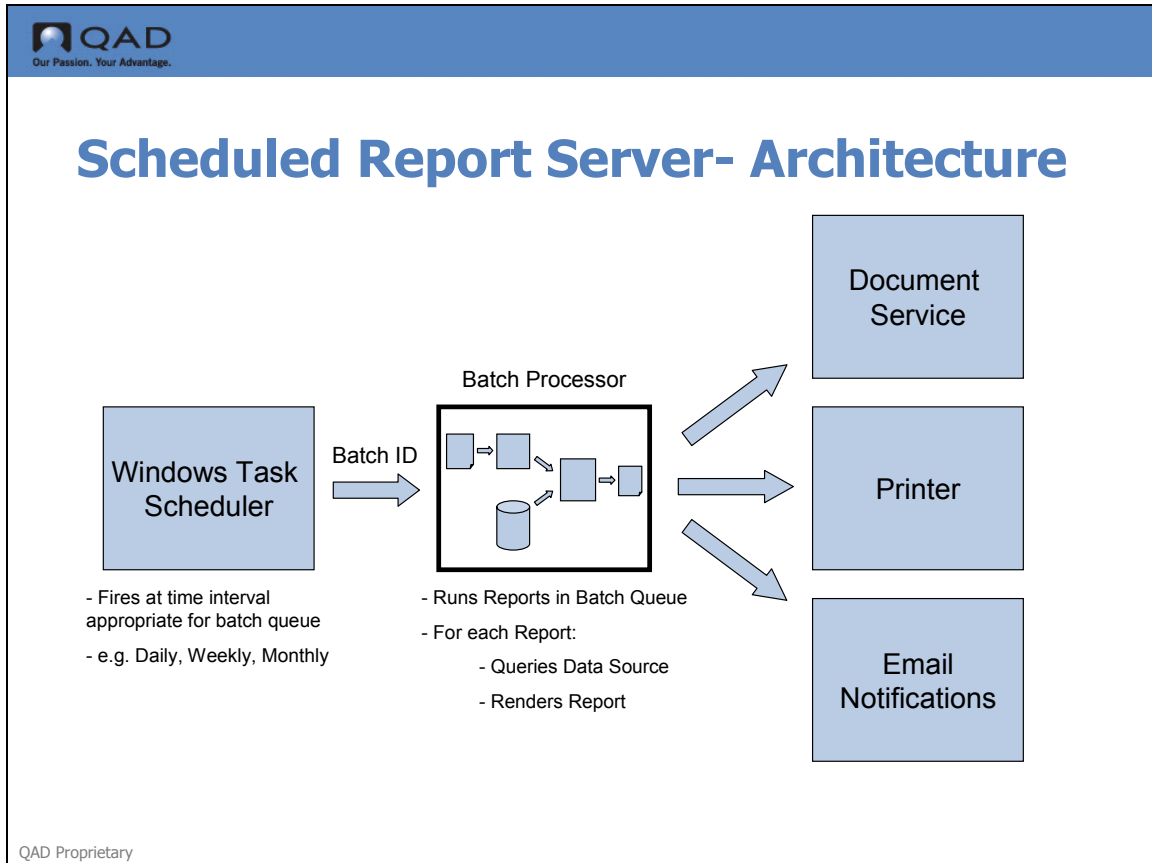
- ▲ Report Framework seamlessly integrated into .NET UI
- ▲ No special installation needed
- ▲ No license fees to customers
- ▲ Report Server Installation
 - Windows OS required
 - Install .NET UI client in standard fashion
 - Run the server using command line (no GUI)
 - Can deploy multiple servers
 - Failover
 - Increased throughput

QAD Proprietary

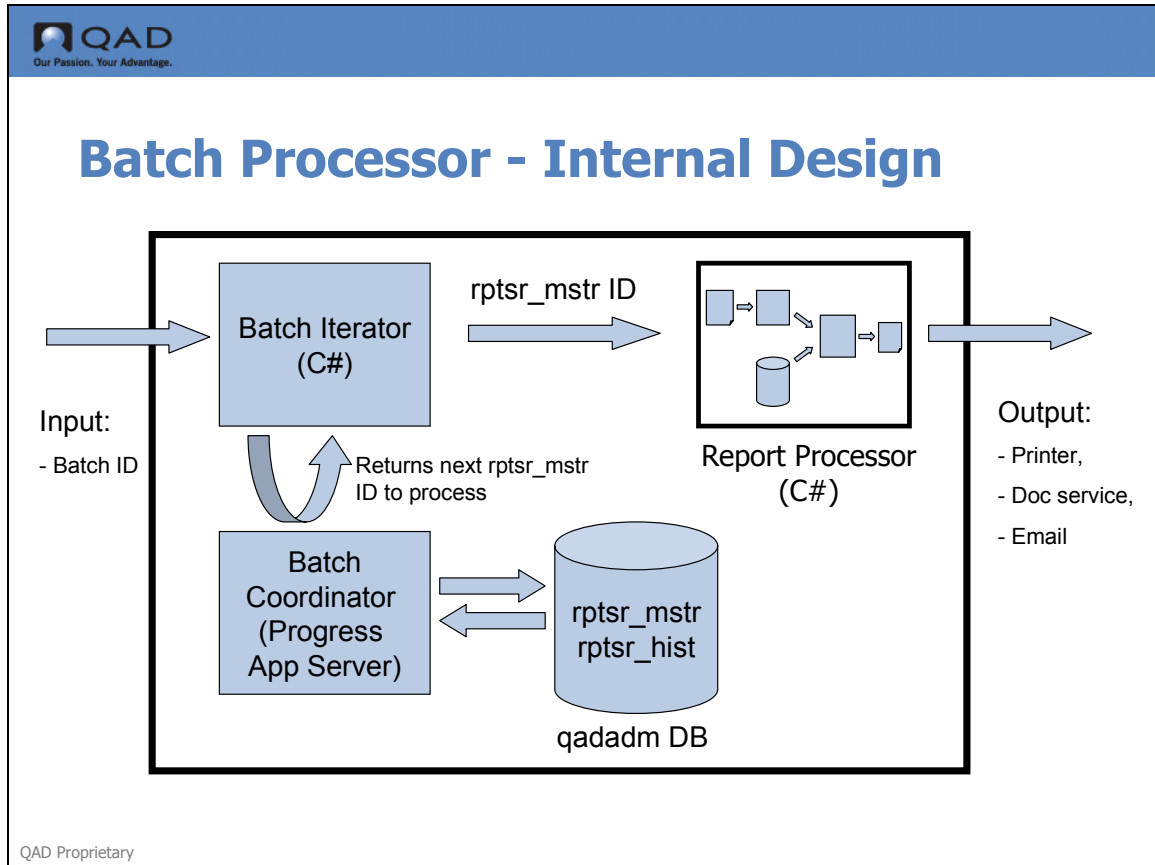
Reporting Framework: Rendering Architecture



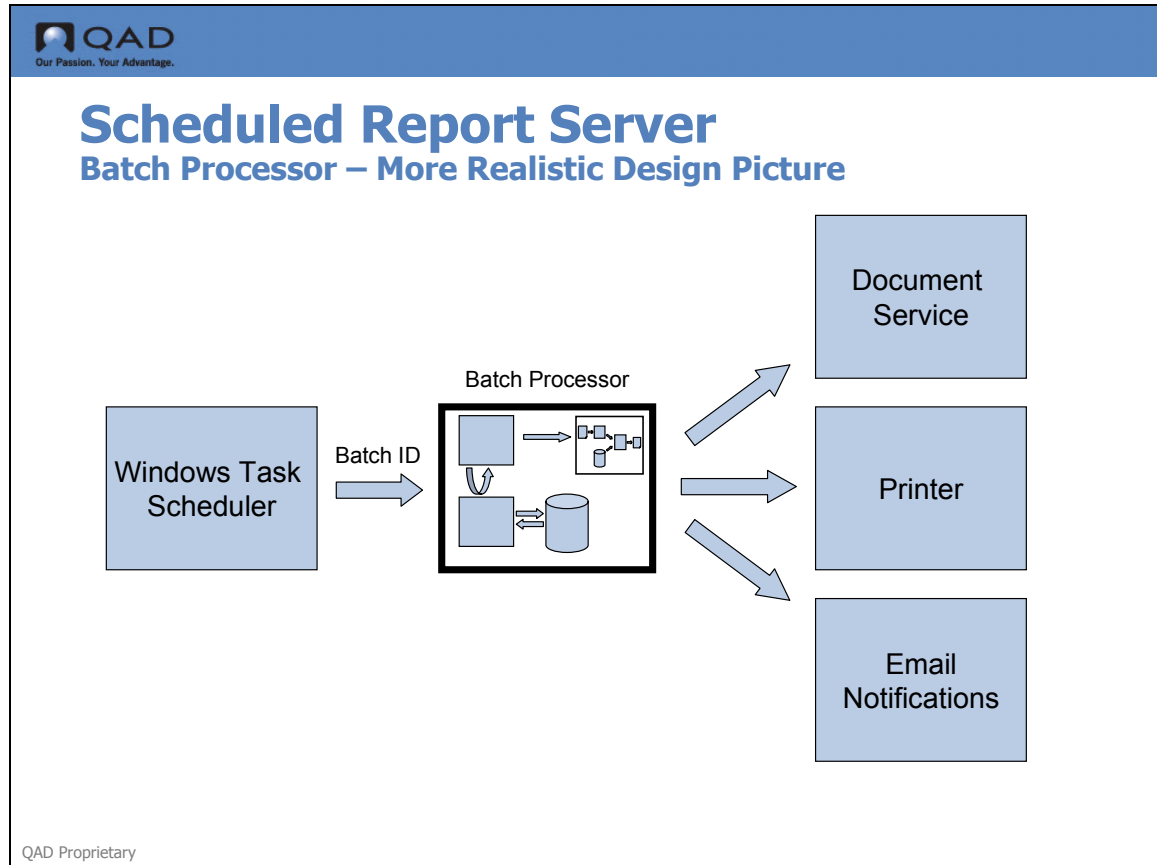
Scheduled Report Server - Architecture



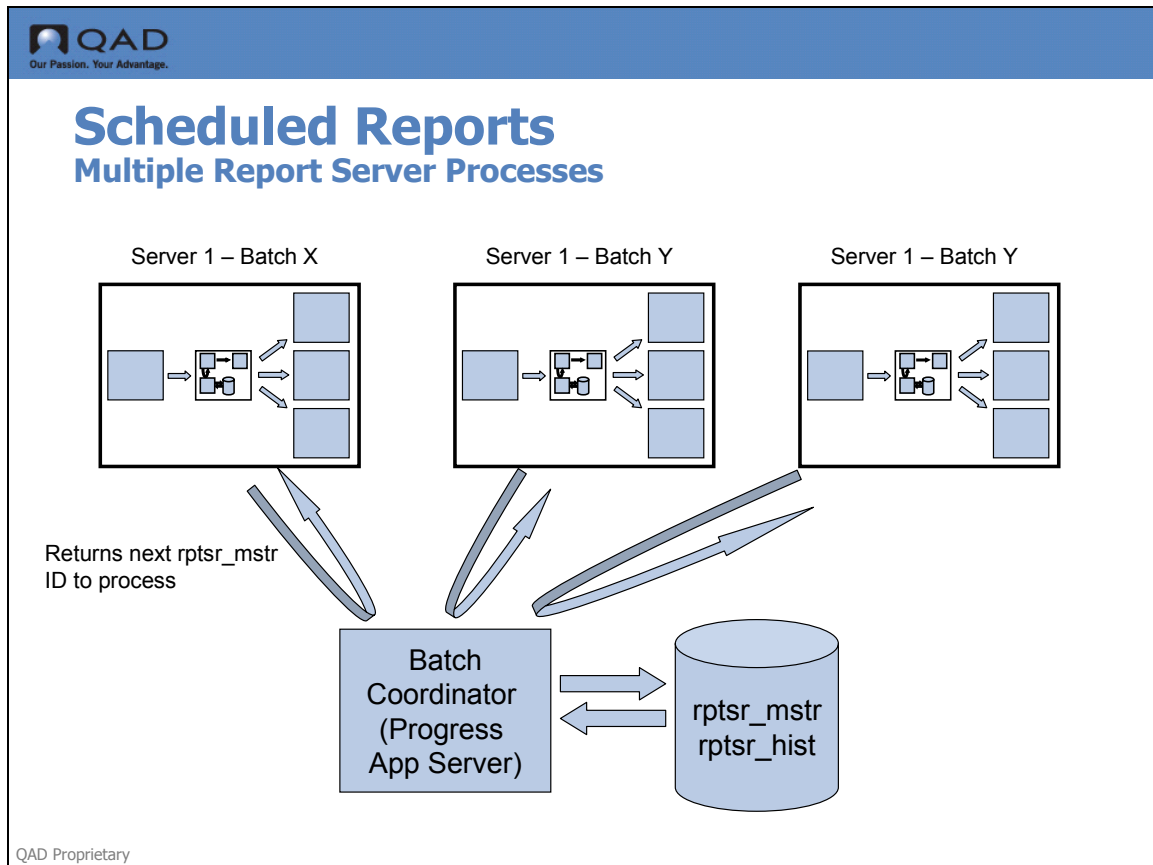
Batch Processor - Internal Design



Scheduled Report Server: Batch Processor Design



Scheduled Reports: Multiple Report Server Processes



When multiple report server processes are run, they can be either on the same machine or different machines.

The physical partitioning can be determined by examining batch frequency and number of reports. Report rendering is a CPU-intensive process.

Any one report server process is linked to a single Batch ID.

It is allowed to have multiple server processes handling the same Batch ID.

Scheduled Reports Status



Scheduled Report *Status*

- ▲ Scheduled report records in the batch queues have a `rptsr_mstr.rptsr_status` field containing current status
- ▲ Used by batch processor to coordinate the processing of the batch
- ▲ Status Values:
 - **NEW:** Newly created and has never run
 - **WAITING:** Initialized to run in current batch run
 - **RUNNING:** Currently running
 - **COMPLETED:** Finished running, with no errors
 - **ERROR:** Finished running, with errors

QAD Proprietary

Scheduled Report Administration: Batch ID Maintenance



Scheduled Report Administration Batch ID Maintenance

- ▲ Same batch DB table used (bc_mstr) as for legacy MFG/PRO reports
 - Different detail table: rptsr_mstr (one record per scheduled report)
 - Each batch ID can logically a queue of scheduled reports (as rptsr_mstr records)
- ▲ Same *Batch ID Maintenance* program as for legacy MFG/PRO reports

Batch ID Maintenance

Go To Actions Copy Print Preview Attach

Batch ID: breport1

Batch ID:

Permanent:

Priority:

QAD Proprietary

Scheduled Report Administration: Configuring Windows Task Scheduler



Scheduled Report Administration Configuring Windows Task Scheduler

- ▲ Every time the scheduler fires, a report server process will be launched for a specific Batch ID
 - e.g. Can have nightly, weekly, monthly batches
- ▲ Configure the Task Scheduler to use a command-line to launch the report server:
 - For local parameter file reference:
 - QAD.Client.exe -param.url:file:///c:/params.pf
 - For remote parameter file reference:
 - QAD.Client.exe -param.url:http://somehost/rpt/params.pf
 - Example params.pf file:

```
-silent
-config-name:test
-user:mfg
-password:(blank)
-workspace:Domain1.1000
-report-batch:nightly1
-enable:qad.plugin.services
-enable:qad.plugin.reports
-enable:qad.plugin.reportserver
-report-mode:batch
```

Printer Setup



Printer Setup

- ▲ Scheduled Report printers:
 - must be set up as Windows printers on the report server machine(s)
 - Allows the .NET report server processes to use them
 - Must be set up in QAD Applications
 - Allows the UI to use them (New Scheduled Report form printer lookup)
 - Use *Printer Setup Maintenance* (36.13.2)
 - e.g. Device Pathname=\\shnt07\XMPR03

QAD Proprietary

Using Printer Setup Maintenance

The screenshot displays the 'Printer Setup Maintenance' web application. At the top left is the QAD logo with the tagline 'Our Passion. Your Advantage.' The main heading is 'Using Printer Setup Maintenance'. Below this is a browser window titled 'Printer Setup Maintenance' with a menu bar containing 'Go To', 'Actions', 'Copy', 'Print', 'Preview', and 'Attach'. The window content is divided into two sections: 'Printer Definition' and 'Printer Control'. The 'Printer Definition' section includes fields for 'Output To: xmpr03', 'Description: \\shnt07\XMPR03', 'Max Pages: 0', and 'Device Pathname: \\shnt07\XMPR03'. To the right of these fields are 'Destination Type' (a dropdown menu), 'Printer Type', 'Lines / page: 66', 'Scroll Output: ', and 'Spooler: '. The 'Printer Control' section contains fields for 'Initialize Command:', 'Initialize Ctrl:', '80 Column Start:', '132 Column Start:', 'Reset Ctrl:', and 'Reset Command:'. The bottom left corner of the browser window shows 'QAD Propr.'.

Email SMTP Settings



Email SMTP Settings

- ▲ Configures the .NET UI system to tell it what SMTP server to use for sending email
- ▲ Use the following entry in client-session.xml:

Template:


```
<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>SMTPPassword</Smtp.Password>
<!-- SMTP use SSL -->
<Smtp.UseSSL>>false</Smtp.UseSSL>
...
</Configuration>
```

Example:

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

QAD Proprietary

Email Template File



Email Template File

- ▲ Allows administrator to customize the email notifications sent from report servers
- ▲ File located on home server:
report-email-template.txt
- ▲ File location:
<TomcatInstallDir>/webapps/qadhome/configurations/<config>/storage/reports/
- ▲ Example Contents:


```
[SUBJECT]
Scheduled Report Completed: {$RRO_DESC}
[BODY]
A scheduled report from QAD Enterprise Applications has completed:

Report:    {$RRO_DESC} ({$RRO_CODE})
Description:  {$SR_DESC}
Link to Report: {$REPORT_FILE_LINK}
```

QAD Proprietary

Available dynamic variables to use in the email template:

`{$RRO_CODE}` : Report Code (as entered in Report Resource Maintenance).

`{$RRO_DESC}` : Description (as entered in Report Resource Maintenance).

`{$SR_DESC}` : Scheduled Report Description (as entered in the New Schedule Report form).

`{$REPORT_FILE_LINK}` : URL Link to output PDF file (if file output was specified when scheduled report was requested).

Scheduled Report Administration



Scheduled Report Administration

- ▲ Any End User
 - Can create and monitor scheduled reports from the report viewer screen
 - Visibility restricted to the current report open in viewer
- ▲ Administrator
 - Scheduled Report Browse
 - Shows current status of all scheduled reports in each batch
 - Scheduled Report History Browse
 - Shows all past scheduled report runs
 - Scheduled Report Maintenance
 - Allows viewing and editing of scheduled report properties
 - e.g. changing priorities, deleting

QAD Proprietary

End User Report Scheduling



End User Report Scheduling

Scheduling and monitoring can be done from the viewer

The screenshot shows a software window titled 'Sales Order Print Sample - Vie...' with several tabs. The 'Filter: Viewer' tab is active. The menu bar includes 'New Filter', 'Open', 'Save', 'Save As', 'Delete', 'Settings', 'Document', 'Run', and 'Schedule'. The 'Schedule' menu is open, showing options: 'New', 'View Schedule', and 'View History'. Below the menu is a 'Search Conditions' table with the following data:

Field	Operator	Value	Buttons
Sales Order	equals		+ x
Sold-To	equals	002	+ x
Order Date	equals	Current Month	+ x
Language	equals		+ x
Update	equals	No	+ x
Increment Order Revi	equals	Yes	+ x
Show Comments	equals	Yes	+ x
Print Options	equals	Yes	+ x
Company	equals	10000000	+ x

QAD Proprietary

Scheduled Report Browse



Scheduled Report Browse

Shows current status of scheduled reports in each batch

The screenshot shows a web application window titled 'Scheduled Report Browse'. It features a search bar at the top with a 'Search' button and a 'Clear All' button. Below the search bar, there are navigation controls and a table of report batches. The table has the following columns: Domain, Batch ID, Priority, Create Date, Create Time, Status, Report Code, Active, Permanent, and Last Sta. The table contains 18 rows of data, with the first row highlighted in blue.

Domain	Batch ID	Priority	Create Date	Create Time	Status	Report Code	Active	Permanent	Last Sta
Domain1	ci01	0	7/12/2009	22:42:47	COMPLETE	hzyrpt004	Yes	Yes	
Domain1	ci01	0	7/12/2009	22:53:42	COMPLETE	hzyrpt004	Yes	Yes	
Domain1	ci01	0	7/12/2009	22:58:08	COMPLETE	hzyrpt004	Yes	Yes	
Domain1	ci01	0	7/12/2009	23:10:28	COMPLETE	hzyrpt004	Yes	Yes	
Domain1	ci01	0	7/12/2009	23:11:51	COMPLETE	hzyrpt004	Yes	Yes	
Domain1	ci01	0	7/12/2009	23:12:41	COMPLETE	hzyrpt004	Yes	Yes	
Domain1	ci01	0	7/12/2009	23:15:34	COMPLETE	hzyrpt004	Yes	Yes	
Domain1	ci01	0	7/12/2009	23:18:07	COMPLETE	hzyrpt004	Yes	Yes	
Domain1	ci01	0	7/12/2009	23:19:55	COMPLETE	hzyrpt004	Yes	Yes	
Domain1	ci01	0	7/12/2009	23:20:42	COMPLETE	hzyrpt004	Yes	Yes	
Domain1	ci01	0	7/12/2009	23:22:19	COMPLETE	hzyrpt004	Yes	Yes	
Domain1	ci01	0	7/12/2009	23:23:25	COMPLETE	hzyrpt004	Yes	Yes	
Domain1	ci01	0	7/12/2009	23:24:49	COMPLETE	hzyrpt004	Yes	Yes	
Domain1	ci01	0	7/13/2009	00:02:33	NEW	hzyrpt001	Yes	Yes	
Domain1	cn	0	7/16/2009	20:04:29	COMPLETE	ynh_rpt001	Yes	Yes	
Domain1	cn01	0	7/12/2009	23:43:22	COMPLETE	hzyrpt001	Yes	Yes	
Domain1	cn01	0	7/16/2009	17:36:34	COMPLETE	hzyrpt001	Yes	Yes	

QAD Proprietary

Scheduled Report History Browse



Scheduled Report History Browse

Shows past scheduled report runs

Domain	Batch ID	Start Date	Start Time	Status	Report Code	URL
Domain1	ci01	7/10/2009	00:42:26	ERROR	hzyrpt004	
Domain1	ci01	7/10/2009	00:55:44	ERROR	hzyrpt004	
Domain1	ci01	7/10/2009	01:07:38	COMPLETE	hzyrpt004	http://coli48.qad.com:8888/qadzinfandee/webdav/configurations/test/storag
Domain1	ci01	7/10/2009	01:13:53	COMPLETE	hzyrpt002	http://coli48.qad.com:8888/qadzinfandee/webdav/configurations/test/storag
Domain1	ci01	7/12/2009	23:48:44	COMPLETE	hzyrpt004	http://coli48.qad.com:8888/qadzinfandee/webdav/configurations/test/storag
Domain1	ci01	7/12/2009	23:49:08	COMPLETE	hzyrpt004	http://coli48.qad.com:8888/qadzinfandee/webdav/configurations/test/storag
Domain1	ci01	7/12/2009	23:49:18	COMPLETE	hzyrpt004	http://coli48.qad.com:8888/qadzinfandee/webdav/configurations/test/storag
Domain1	ci01	7/12/2009	23:49:28	COMPLETE	hzyrpt004	http://coli48.qad.com:8888/qadzinfandee/webdav/configurations/test/storag
Domain1	ci01	7/12/2009	23:49:39	COMPLETE	hzyrpt004	http://coli48.qad.com:8888/qadzinfandee/webdav/configurations/test/storag
Domain1	ci01	7/12/2009	23:49:49	COMPLETE	hzyrpt004	http://coli48.qad.com:8888/qadzinfandee/webdav/configurations/test/storag
Domain1	ci01	7/12/2009	23:49:59	COMPLETE	hzyrpt004	http://coli48.qad.com:8888/qadzinfandee/webdav/configurations/test/storag
Domain1	ci01	7/12/2009	23:50:09	COMPLETE	hzyrpt004	http://coli48.qad.com:8888/qadzinfandee/webdav/configurations/test/storag
Domain1	ci01	7/12/2009	23:50:20	COMPLETE	hzyrpt004	http://coli48.qad.com:8888/qadzinfandee/webdav/configurations/test/storag
Domain1	ci01	7/12/2009	23:50:29	COMPLETE	hzyrpt004	http://coli48.qad.com:8888/qadzinfandee/webdav/configurations/test/storag
Domain1	ci01	7/12/2009	23:50:40	COMPLETE	hzyrpt004	http://coli48.qad.com:8888/qadzinfandee/webdav/configurations/test/storag
Domain1	ci01	7/12/2009	23:50:51	COMPLETE	hzyrpt004	http://coli48.qad.com:8888/qadzinfandee/webdav/configurations/test/storag

QAD Proprietary

Scheduled Report Maintenance



Scheduled Report Maintenance

Allows viewing and editing of scheduled report properties

The screenshot shows a web browser window with several tabs: "Printer Setup Maintenance", "Scheduled Report Browse", "Scheduled Report History Bro...", and "Scheduled Report Maintenance". The active tab is "Scheduled Report Maintenance". The browser address bar shows "Batch ID:ci01". The page content includes:

- Batch ID: ci01
- Domain: Domain1
- Batch Details section with the following fields:
 - Schedule ID: 9
 - Report Code: hzyrpt004
 - Priority: 0
 - Alert:
 - Permanent:
 - Interval: 0
 - Status: COMPLETE
 - Timeout: 0
 - Description: ci001
- A search field for "Batch ID" containing "ci01" with a magnifying glass icon.
- Metadata fields:
 - Create Date: 07/12/09 22:42:47
 - Last Run Date: 07/12/09 23:48:44
 - Host Name: HZY-SH
 - Process ID: 6524
 - Version: 31

QAD Proprietary

Scheduled Reports: Exercise



Scheduled Reports: Exercise

- 1) Create a new batch ID
 - Batch ID Maintenance
- 2) Create several scheduled reports in the batch
 - Run report from menu, use Schedule button
 - Make sure some have RunOnce=false, SaveFile=true
 - Use View Schedule button to review
- 3) Configure and run a scheduled report server process
 - A) From DOS prompt
 - B) From Windows Task Scheduler
 - Start > Control Panel > Scheduled Tasks
 - Monitor batch runs using *Scheduled Report Browse* and *Scheduled Report History Browse*
- 4) Modify batch using Scheduled Report Maintenance
 - Delete one of the scheduled reports from the batch
 - Change some priority values to change order of running
 - Run batch again and see if the behavior is as expected

QAD Proprietary

- 1 Create a new batch ID
 - Batch ID Maintenance
- 2 Create several scheduled reports in the batch
 - Run report from menu, use Schedule button
 - Make sure some have RunOnce=false, SaveFile=true
 - Use View Schedule button to review
- 3 Configure and run a scheduled report server process
 - a A) From DOS prompt
 - b B) From Windows Task Scheduler
 - Start > Control Panel > Scheduled Tasks
 - Monitor batch runs using Scheduled Report Browse and Scheduled Report History Browse
- 4 Modify batch using Scheduled Report Maintenance
 - Delete one of the scheduled reports from the batch
 - Change some priority values to change order of running
 - Run batch again and see if the behavior is as expected

Summary



Summary

In this session you have learned how to:

- ▲ Configure the QAD Reporting Framework
- ▲ Set up user authorization for report development and admin programs
- ▲ Manage reports on the system menu
- ▲ Administer scheduled report servers

QAD Proprietary

In this session you have learned how to:

- Configure the QAD Reporting Framework
- Set up user authorization for report development and admin programs
- Manage reports on the system menu
- Administer scheduled report servers