QAD Enterprise Applications

Training Guide

Reporting Framework

70-3229-2012SE_EE
QAD 2012 Standard & Enterprise Edition
March 2012

ReportingFramework_TG_v2012SE_EE.pdf/czs/sti

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

Contents

Change Summary

The following table summarizes significant differences between this document and the last published version.

Date/Version	Description	Reference
March 2012 EE	Rebranded for QAD 2012 EE	---
September 2011.1 EE	Rebranded for QAD 2011.1 EE	---

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.

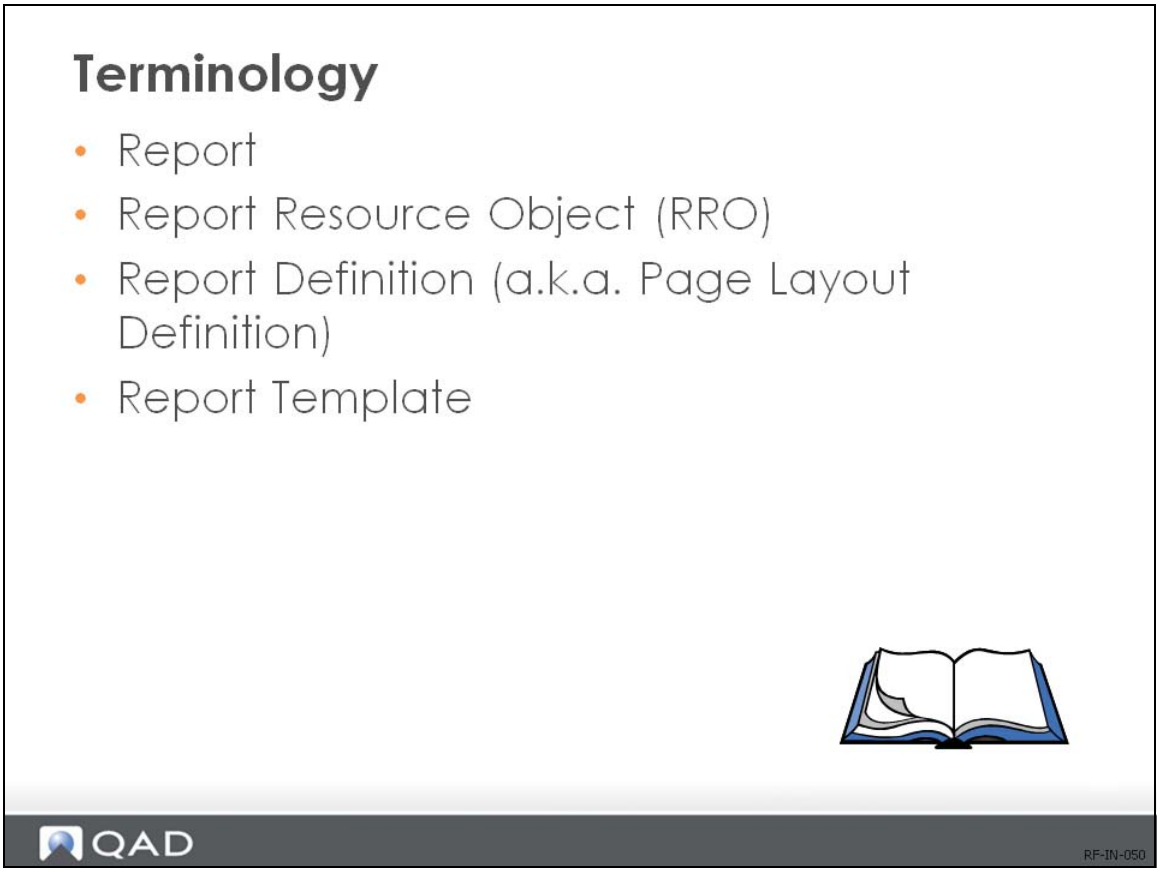
http://www.qad.com/

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

Chapter 1

Introduction

Terminology



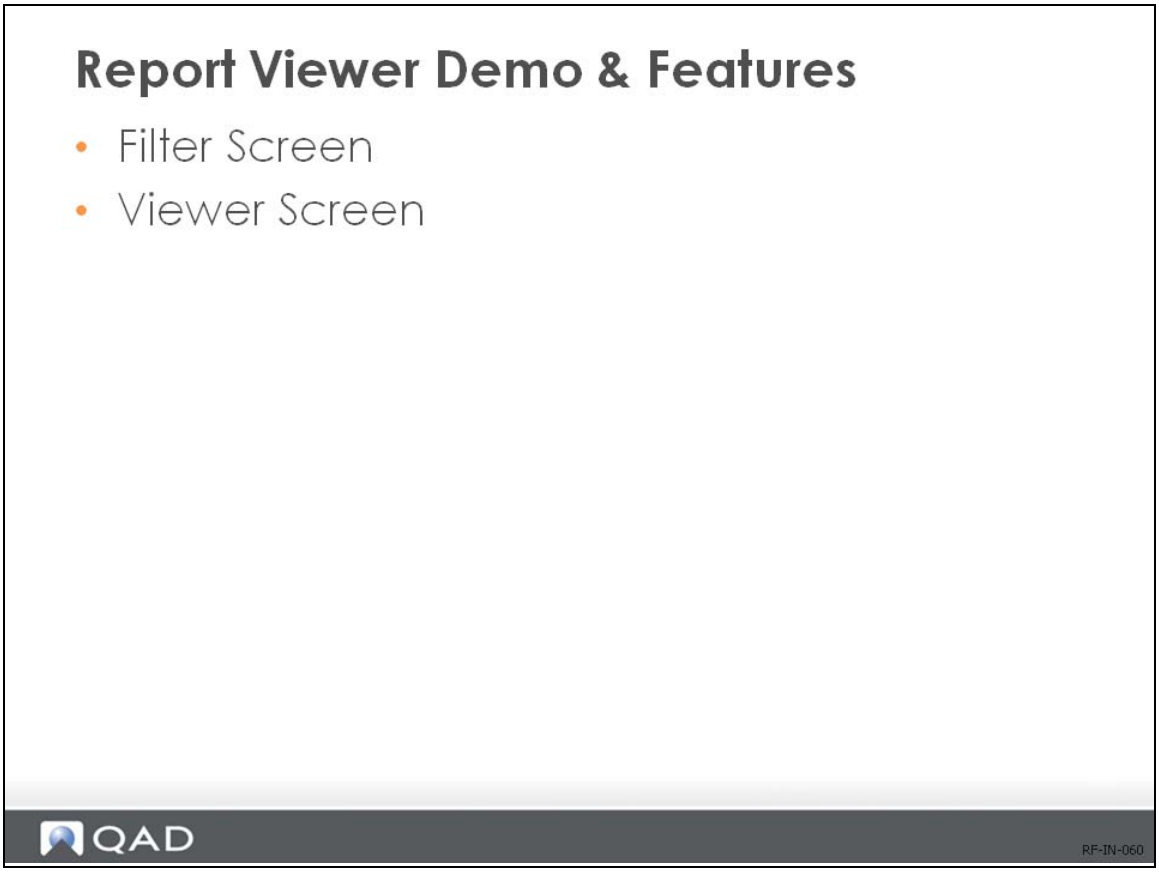
A report is a collection of your desired data, as defined in the report resource, organized in your desired format as defined in the report definition.

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



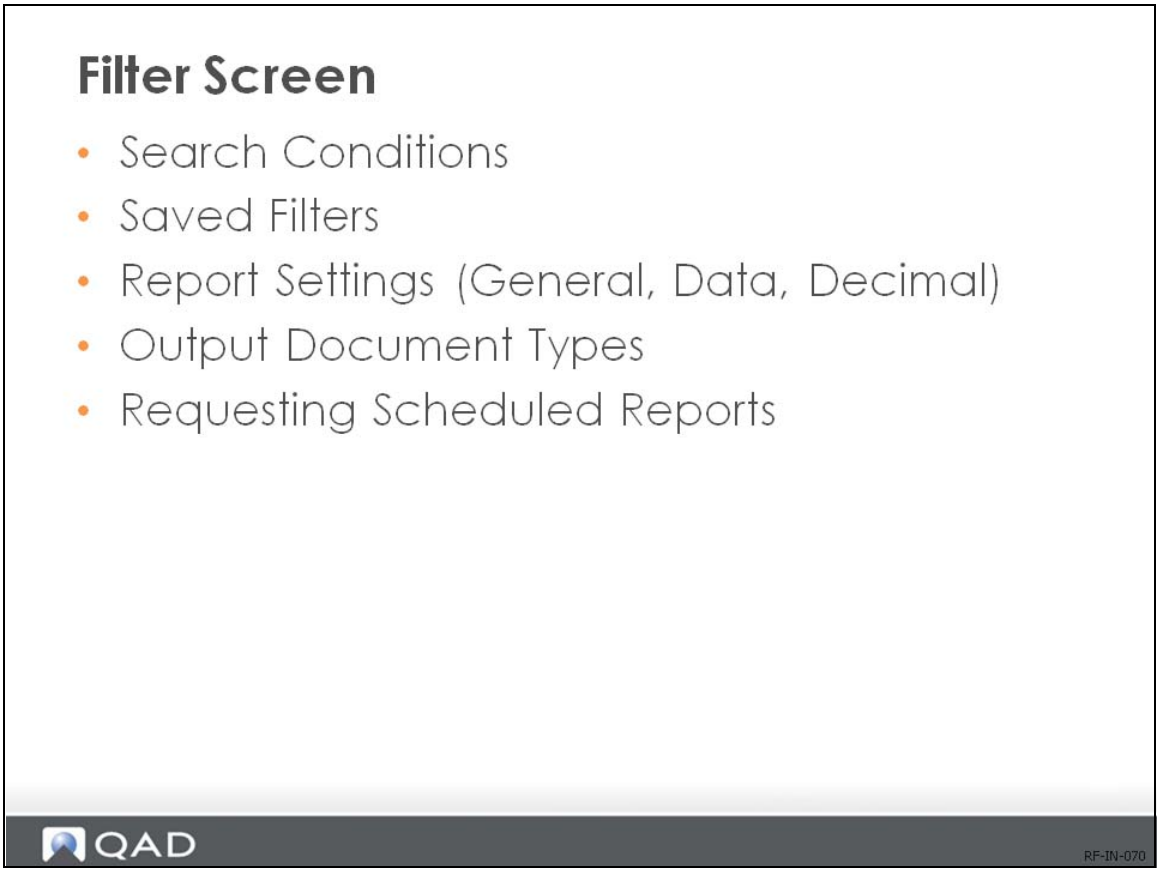
The Report Viewer includes two tabs: Filter for the filter screen and Viewer for the viewer screen.

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.

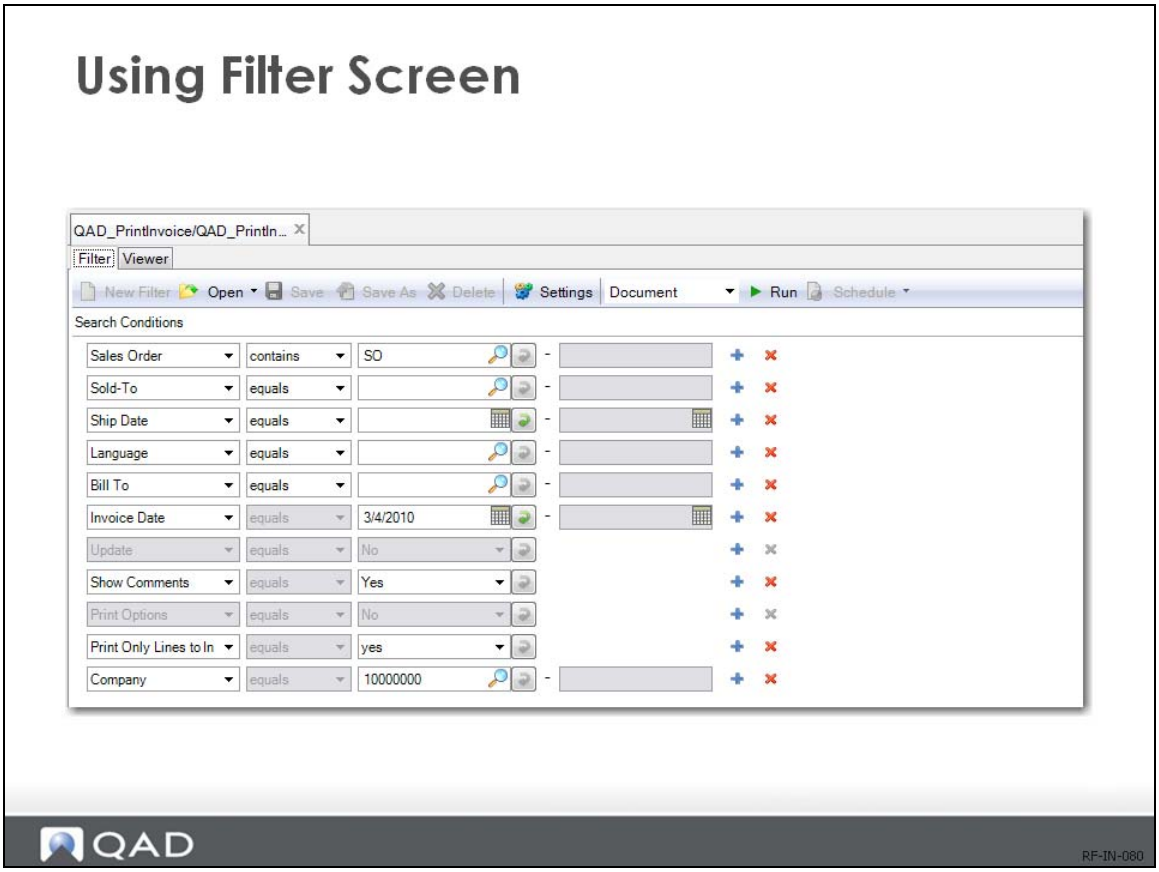
If a report always contains a certain range of data and is exported to a certain format, you do not have to define the filter criteria and output settings every time you generate the report. You can save the search conditions and output settings as a filter and open it to load the same set of configurations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

Filter Screen



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.

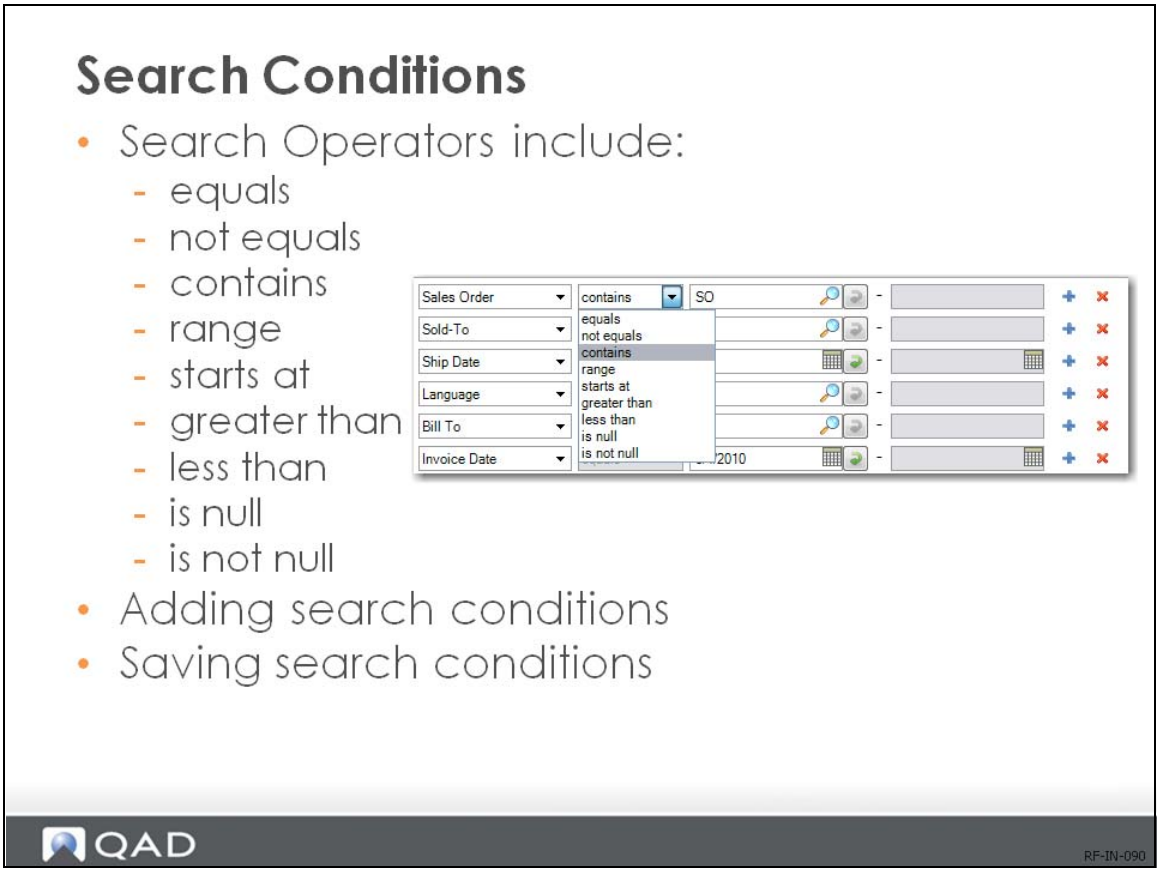
Using Filter Screen



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



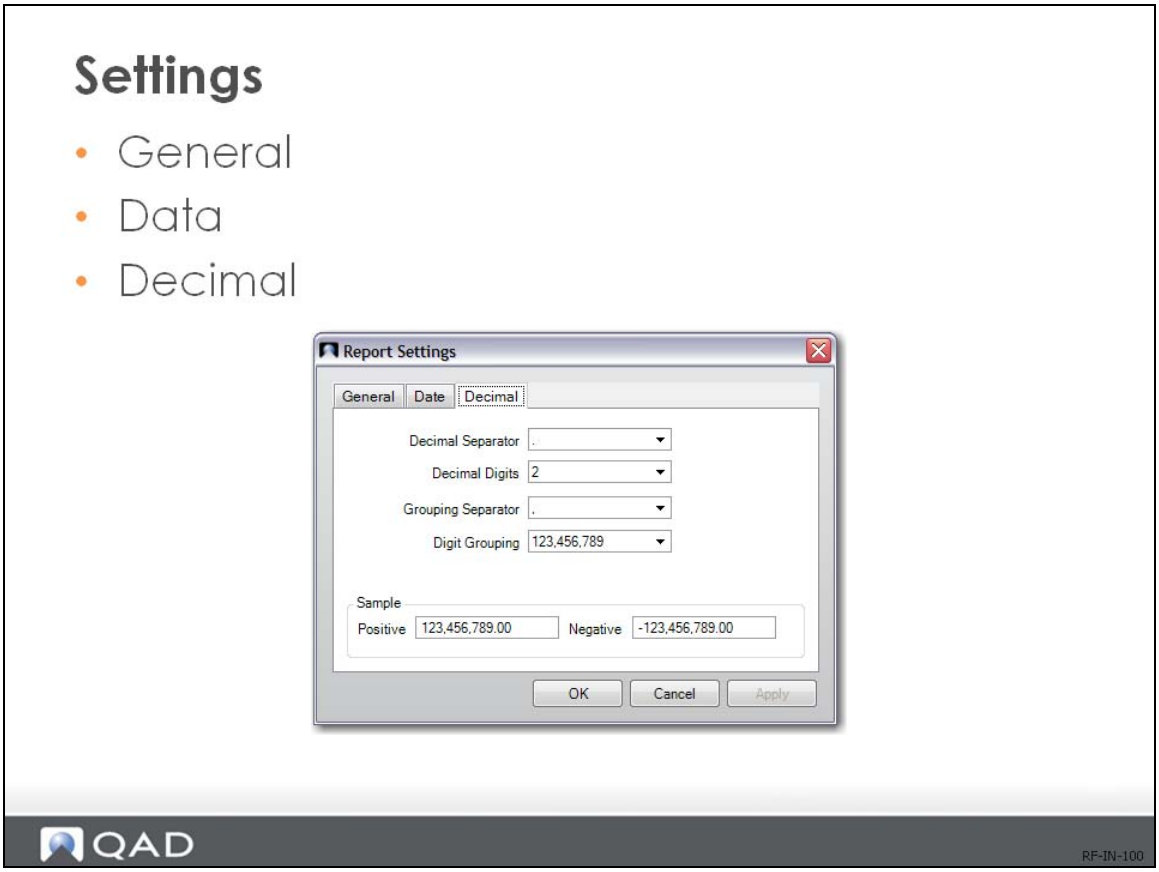
By default, a report will display all the records available in the source data. However, you may want to retrieve just a certain range of records in the report; for example, sales records between last September and this March. You do this by setting search conditions to filter data in the report. You 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.

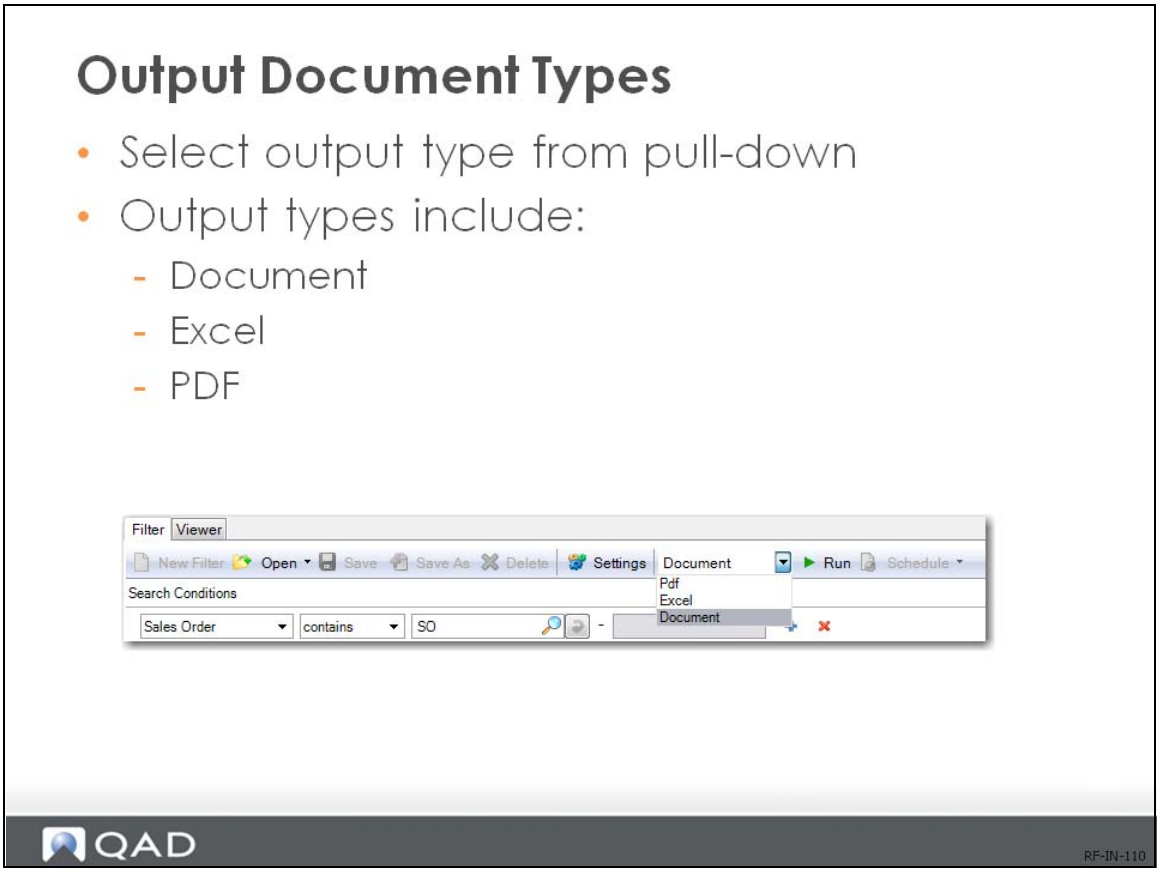
Settings



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

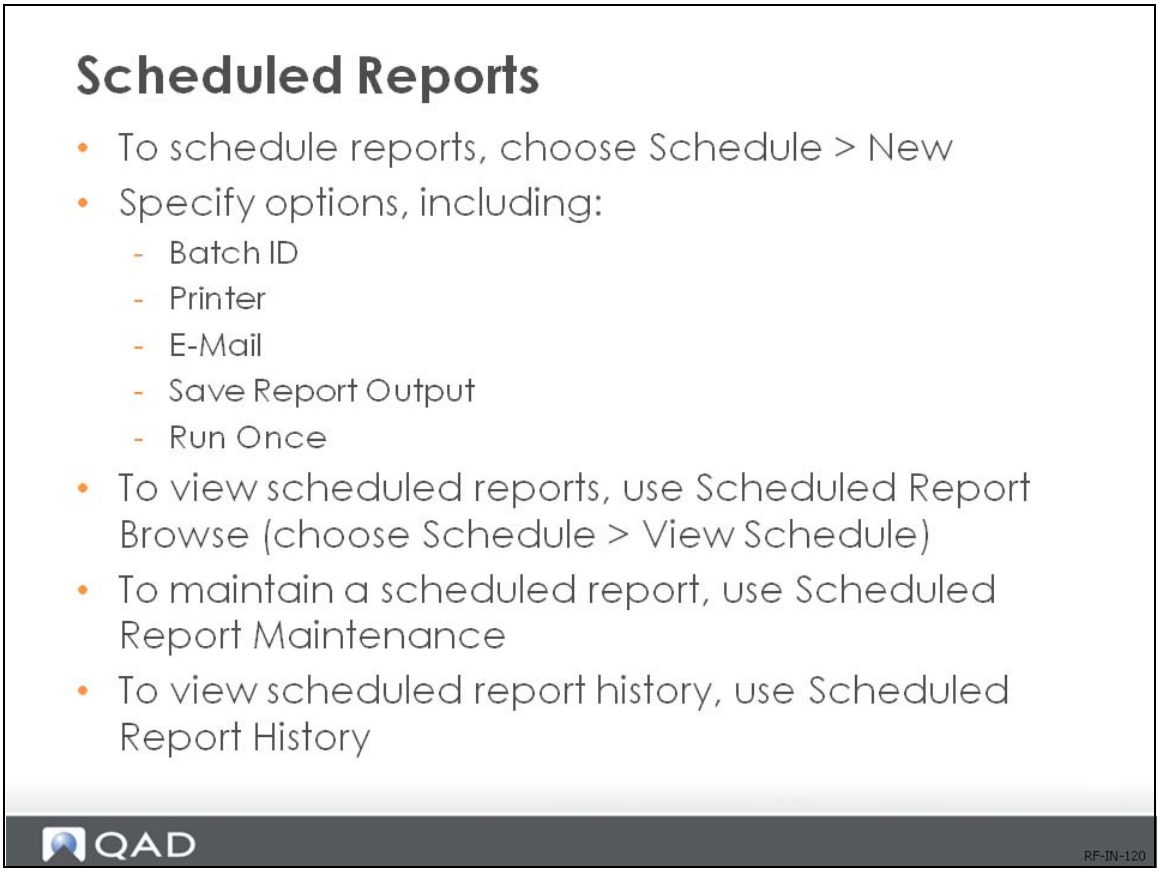


Document. The report is displayed in the Report Viewer window.

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



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.

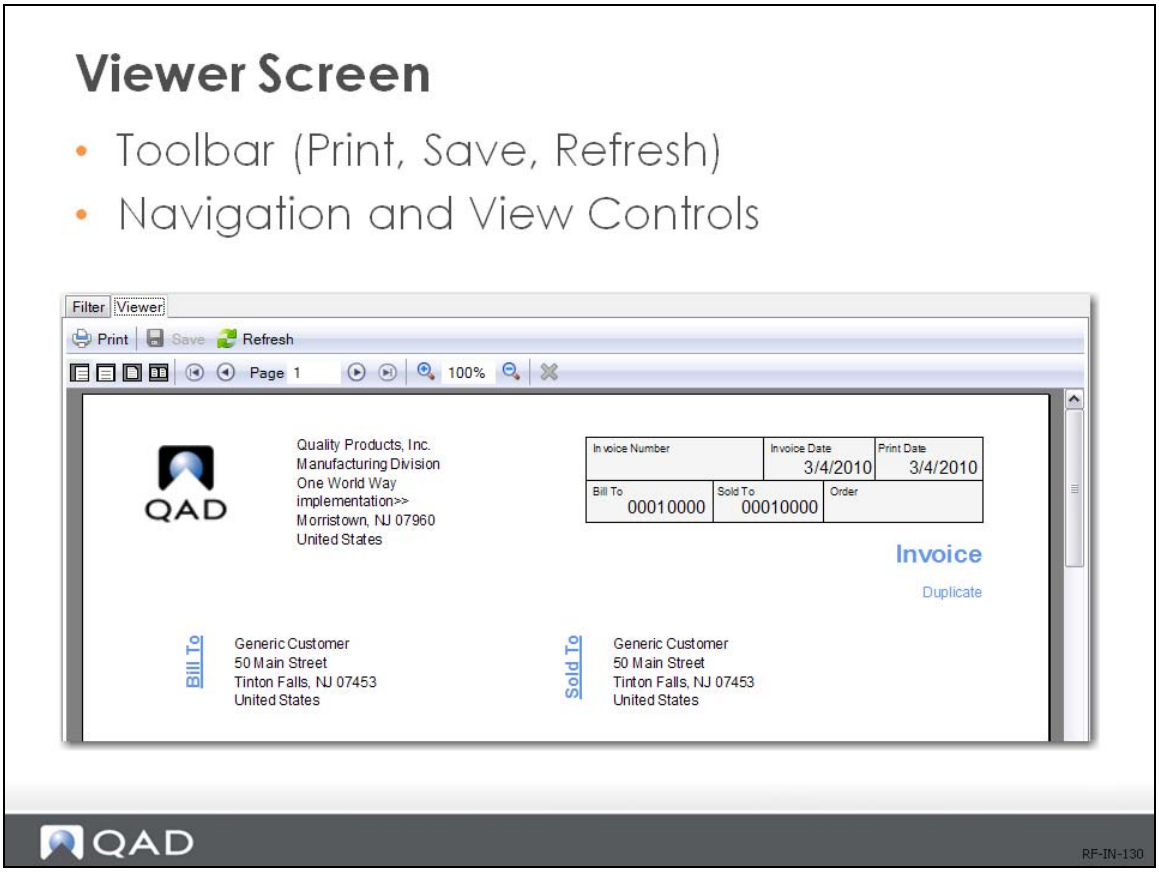
Printer. If you want to have the scheduled report printed, specify a printer to send the report to. Printers are set up for the report server by the administrator in Printer Setup Maintenance (36.13.2).

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 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.

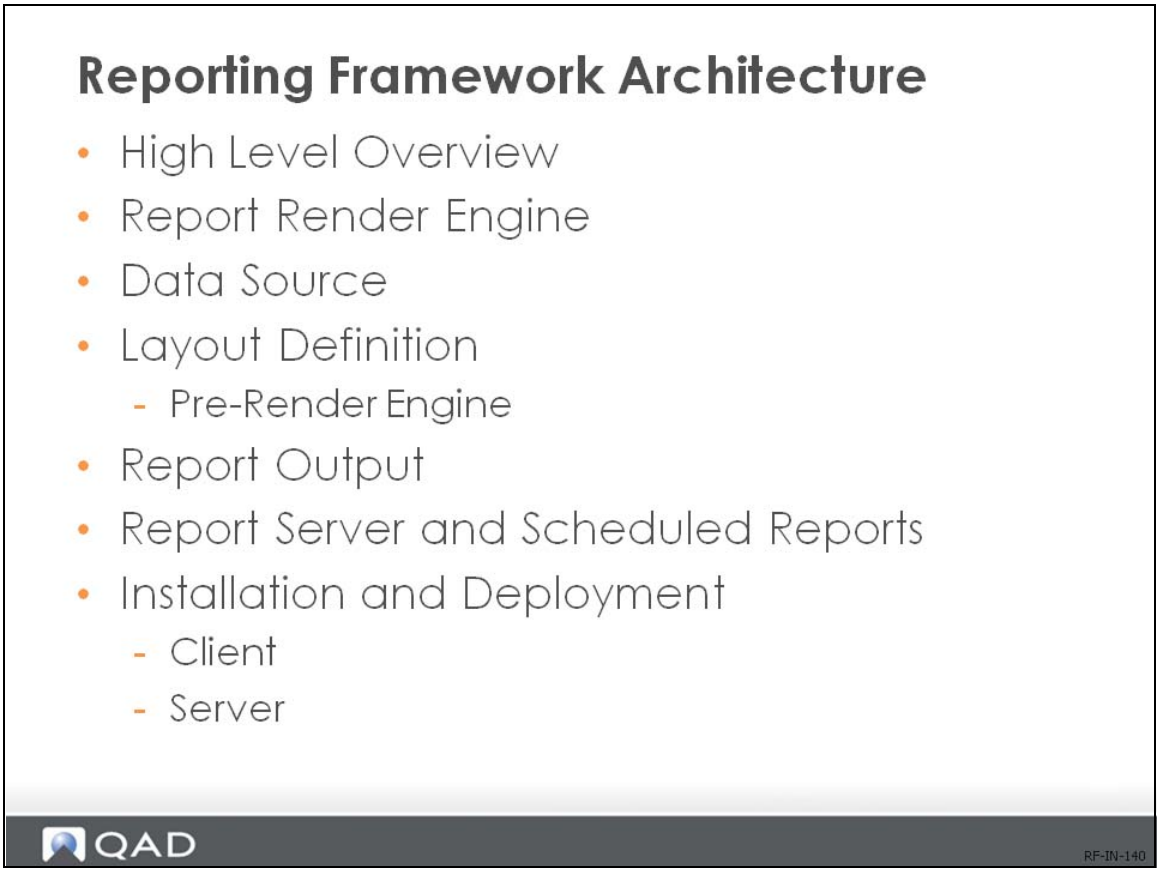
Zoom In. Magnify the report preview size.

Size. Specify the exact report preview size.

Zoom Out. Decrease the report review size.

Cancel Rendering. Cancel rendering the report.

Reporting Framework Architecture



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.

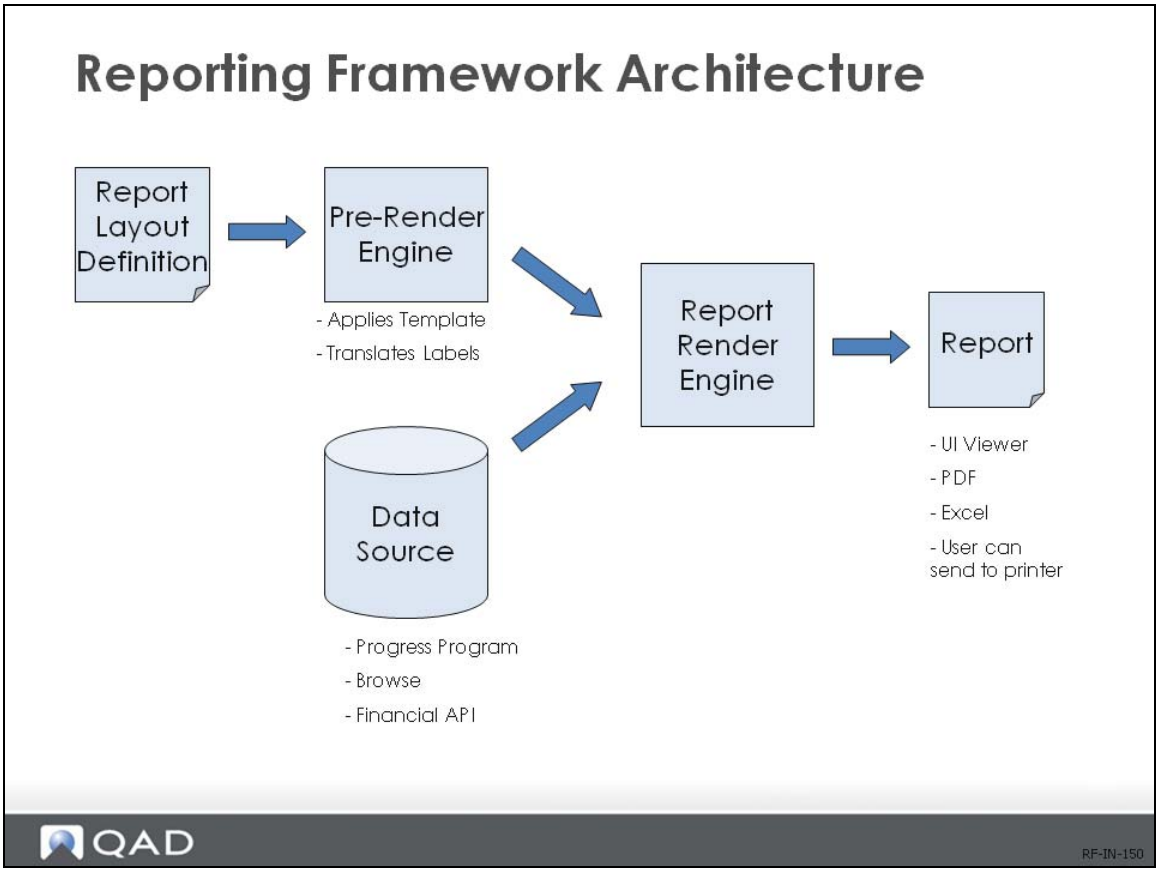
- Pre-render engine

  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.

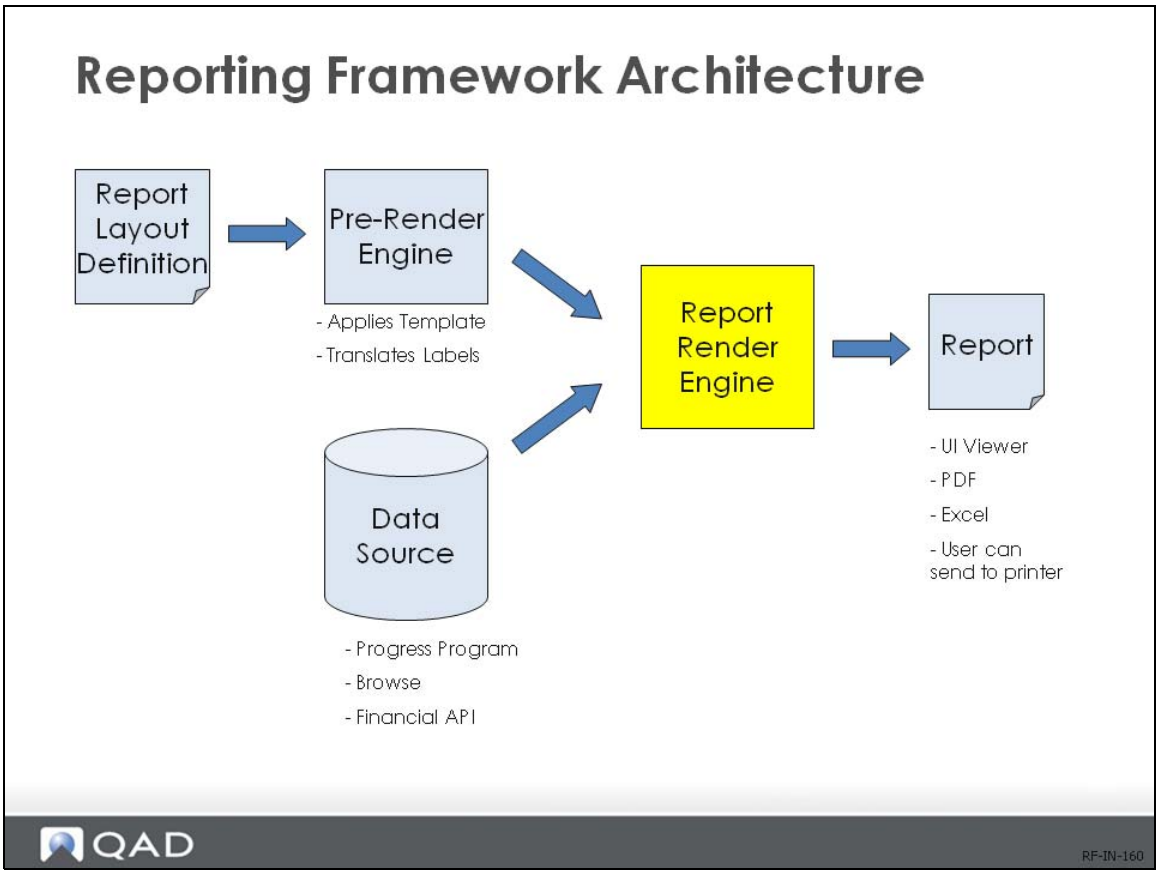
- Report output

  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.

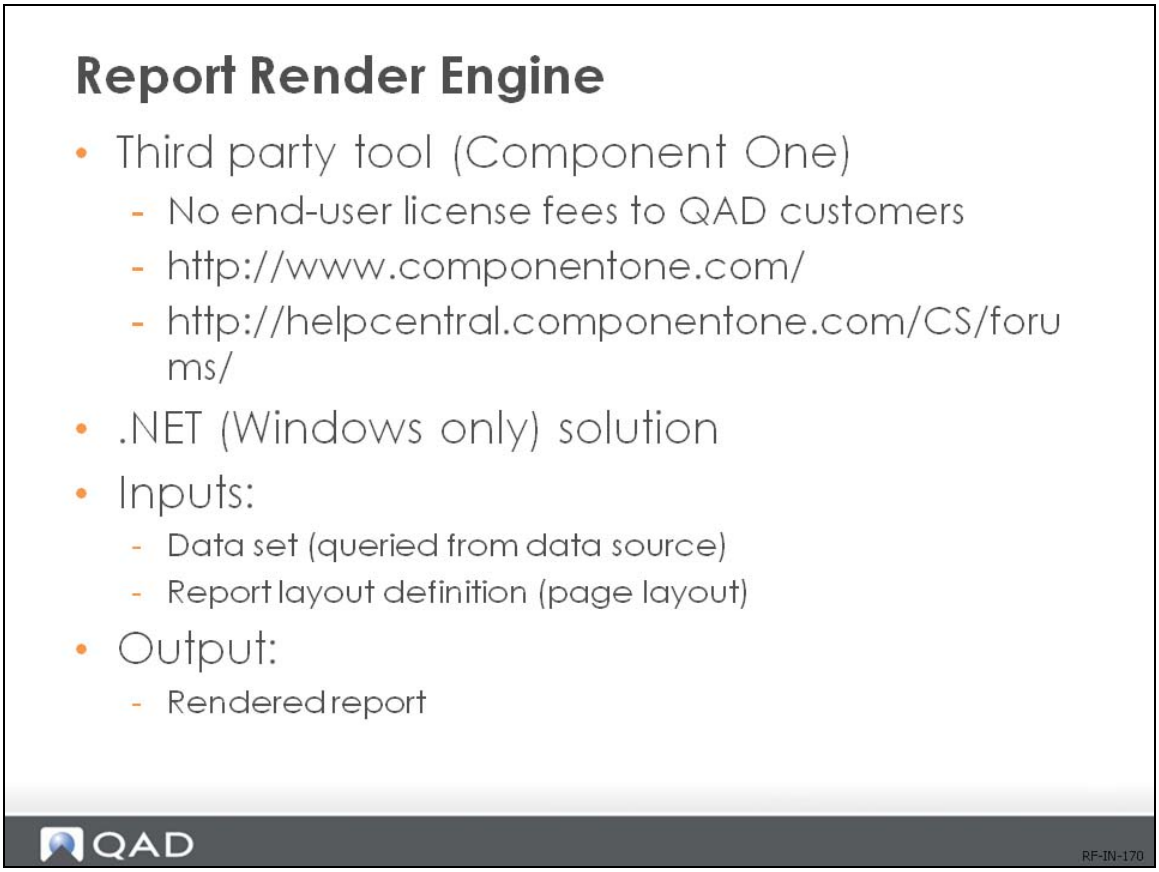
Reporting Framework Architecture



Reporting Framework Architecture: Report Render Engine



Report Render Engine



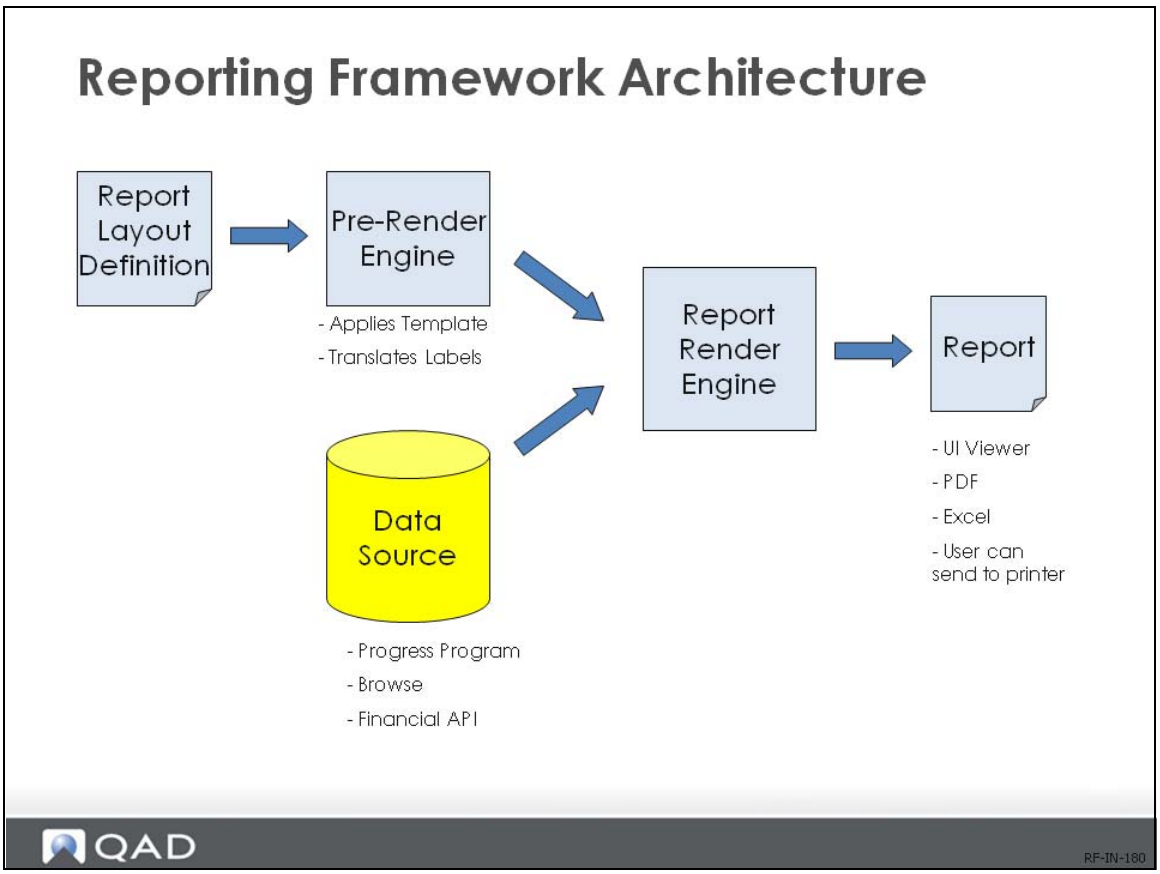
The report render engine takes in data set and report layout definition as inputs, then renders and produces the report output.

The report render engine is based on Component One.

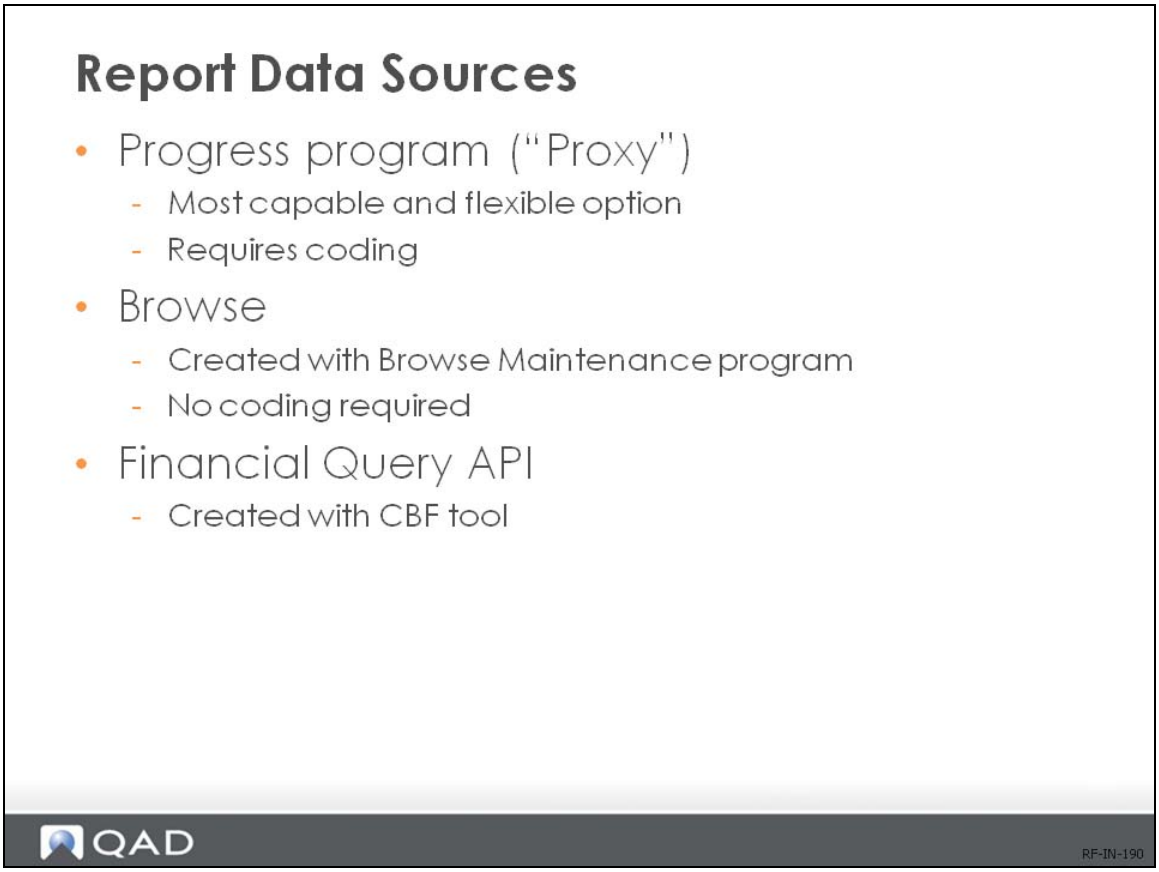
The report render engine is a .NET solution: the report render engine can only run on the Windows operating system.

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



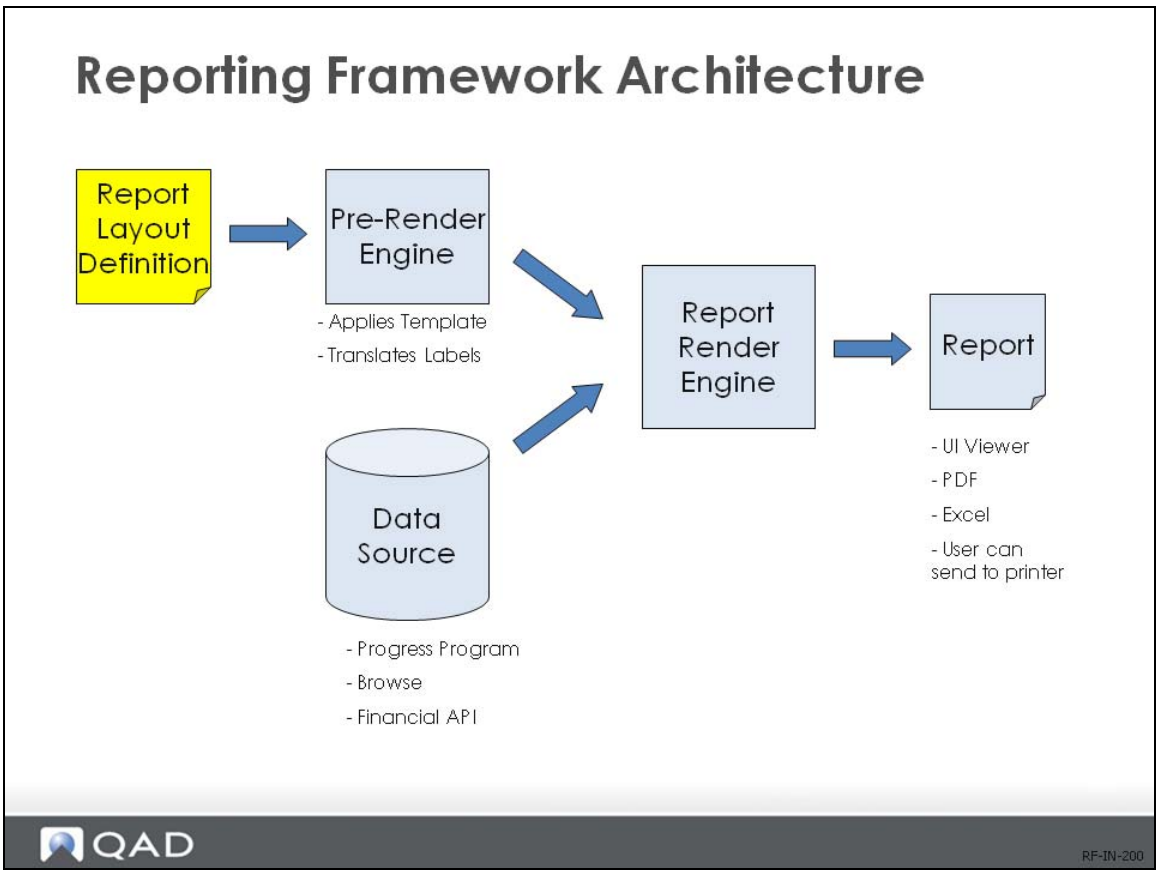
Report Data Sources



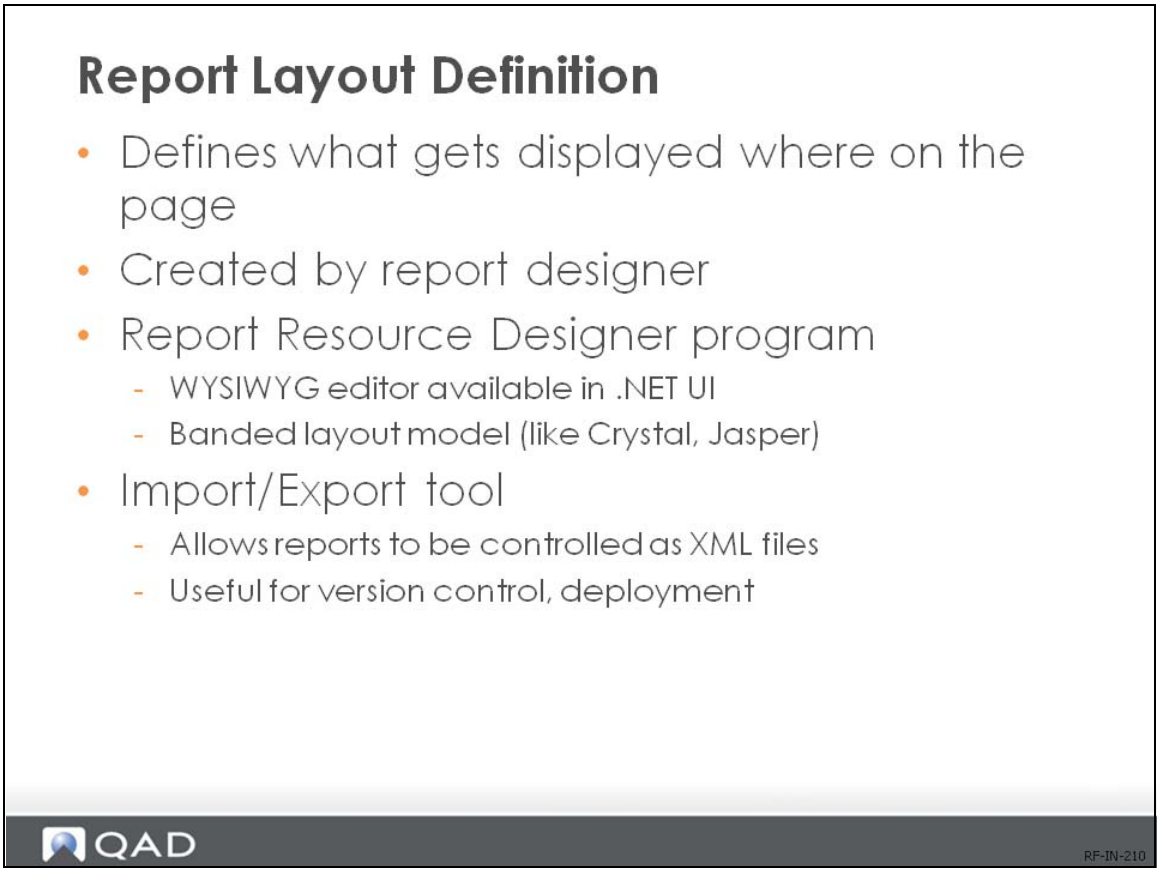
Before you create a report, you need to determine where your report data comes from. The QAD Reporting Framework supports three types of data sources: generic proxy (Progress program), browse, and QAD Financials API. Depending on which type of data source you use, you may need to perform some additional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



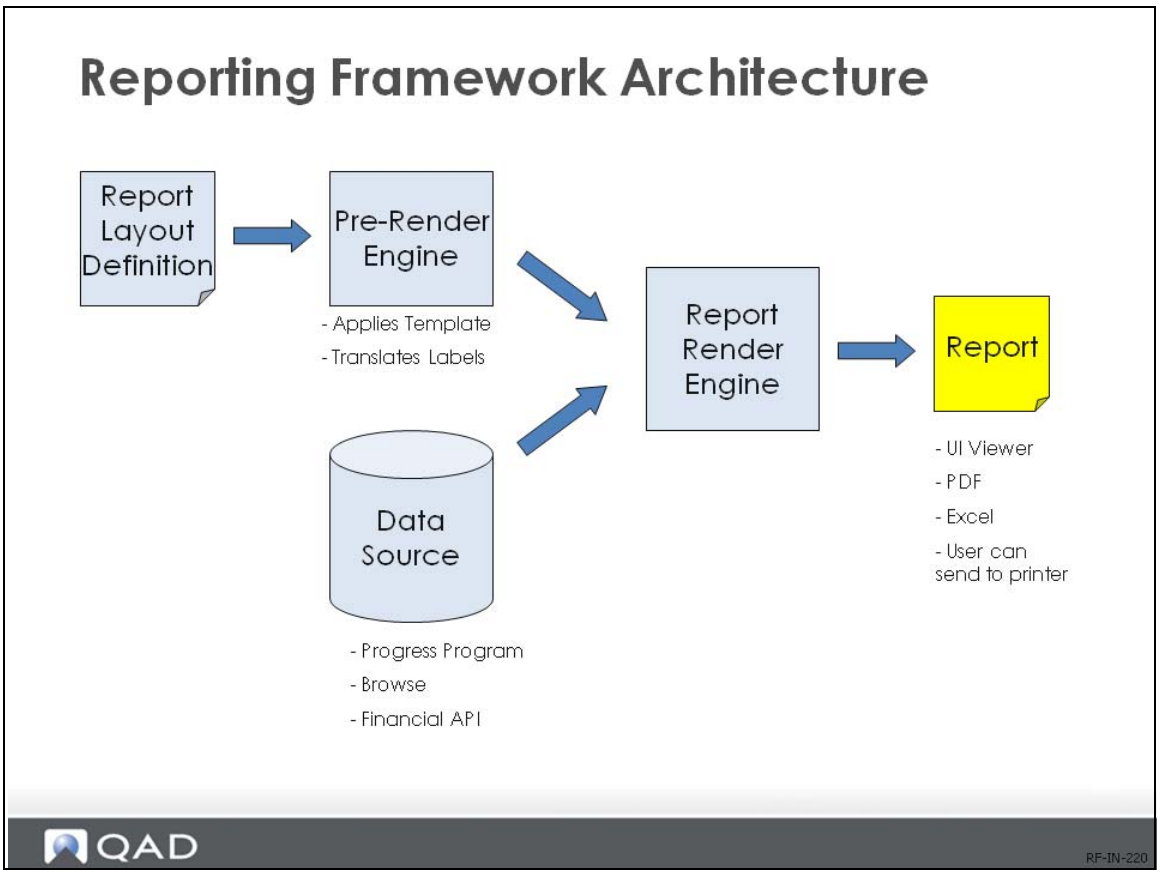
Report Layout Definition



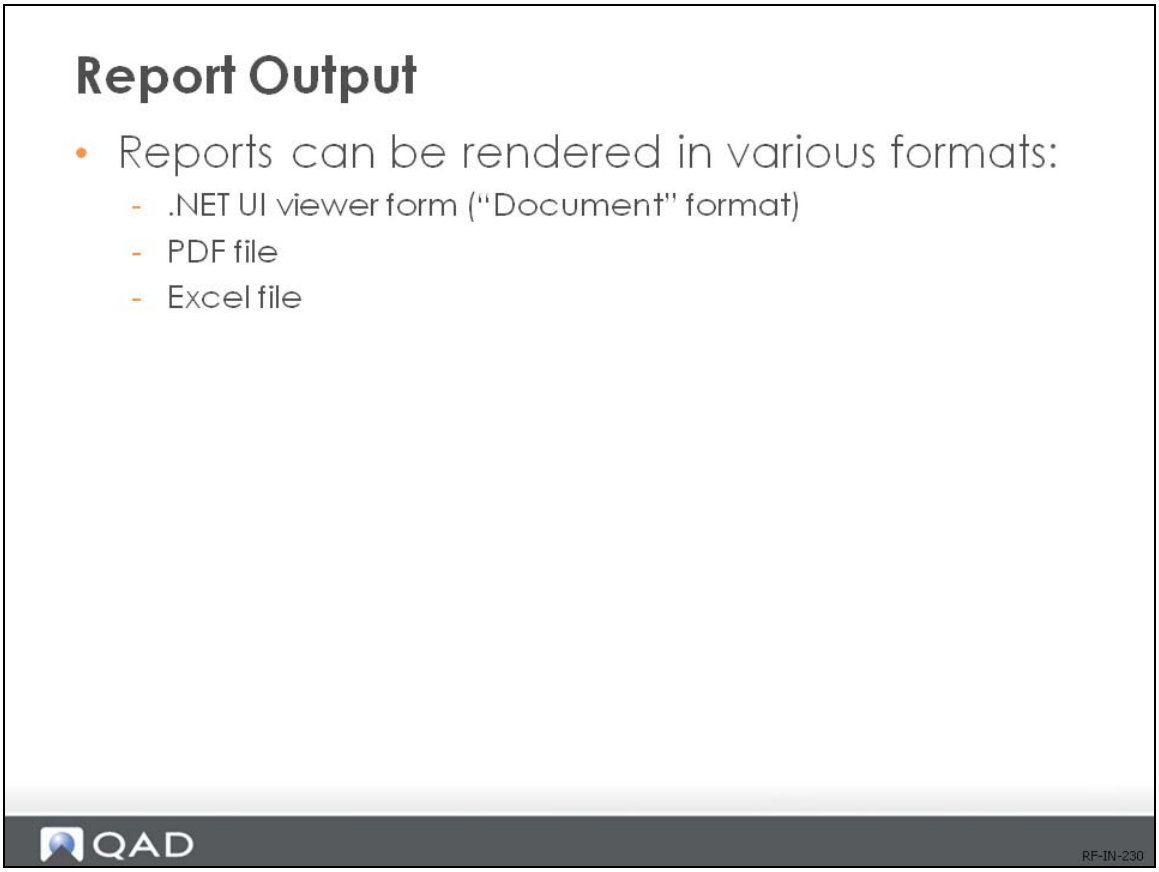
A report definition contains all the information that defines the data binding, layout, and customized formatting of a report. It is saved as an XML file that can be edited in the Report Designer, either visually or in the text editor mode.

Use the Report Resource Designer program to create report definitions.

Reporting Framework Architecture: Report



Report Output

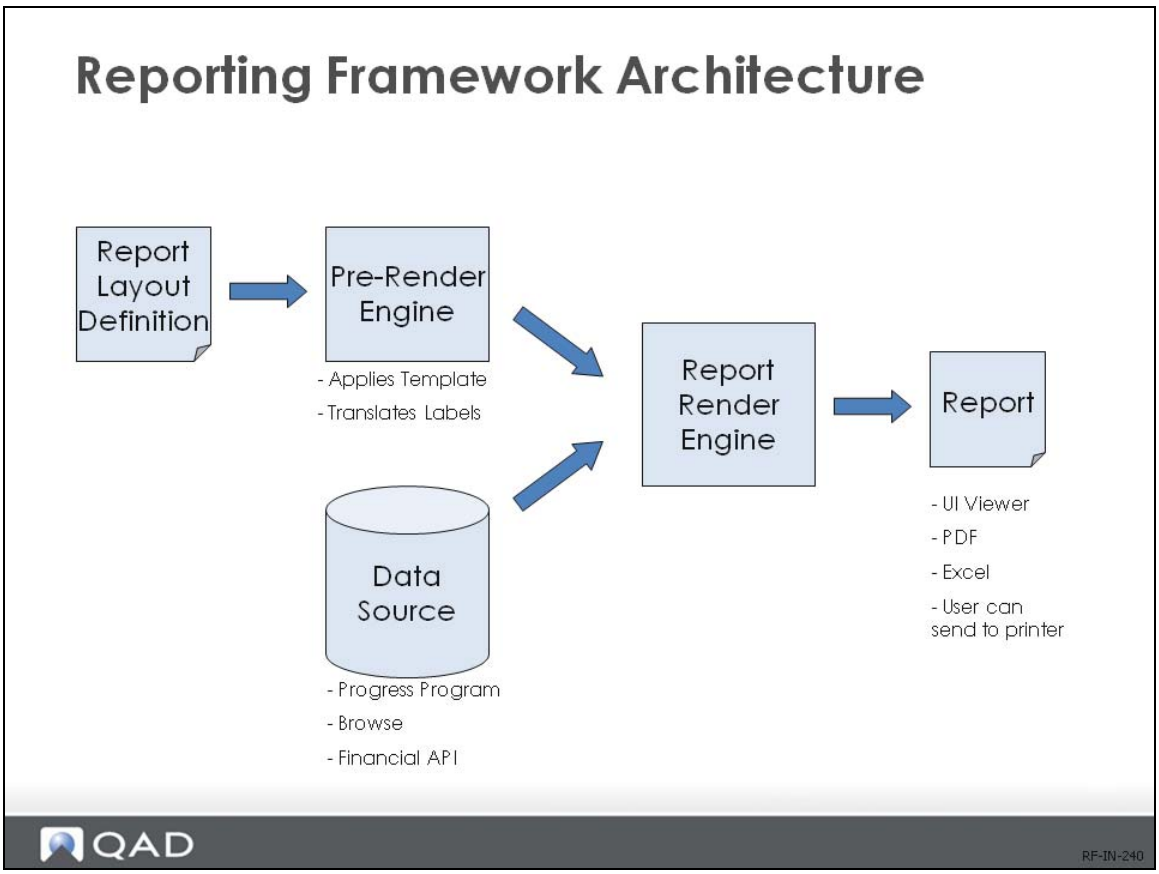


After a report is designed, you can set filter criteria to filter data in the report, run the report, and send it to different output destinations.

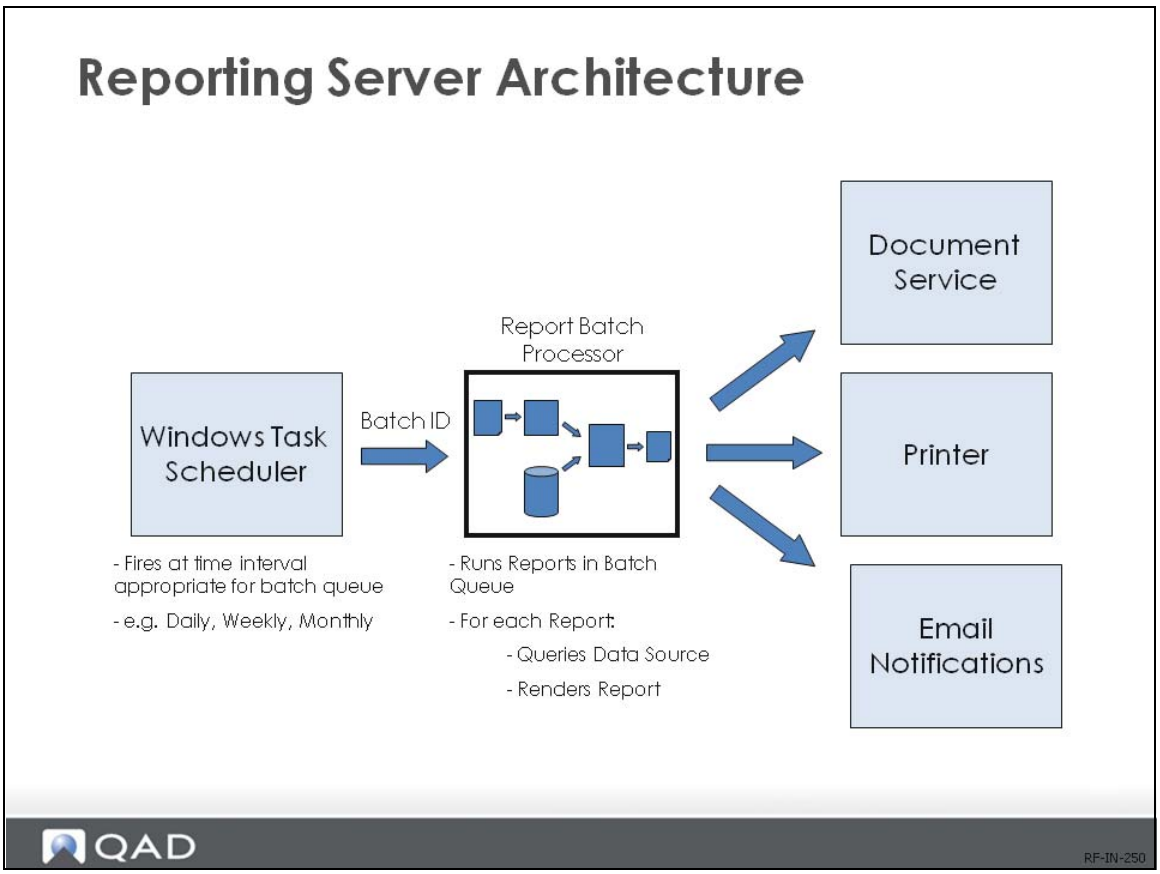
You can choose from three output formats when the report is run:

- 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.

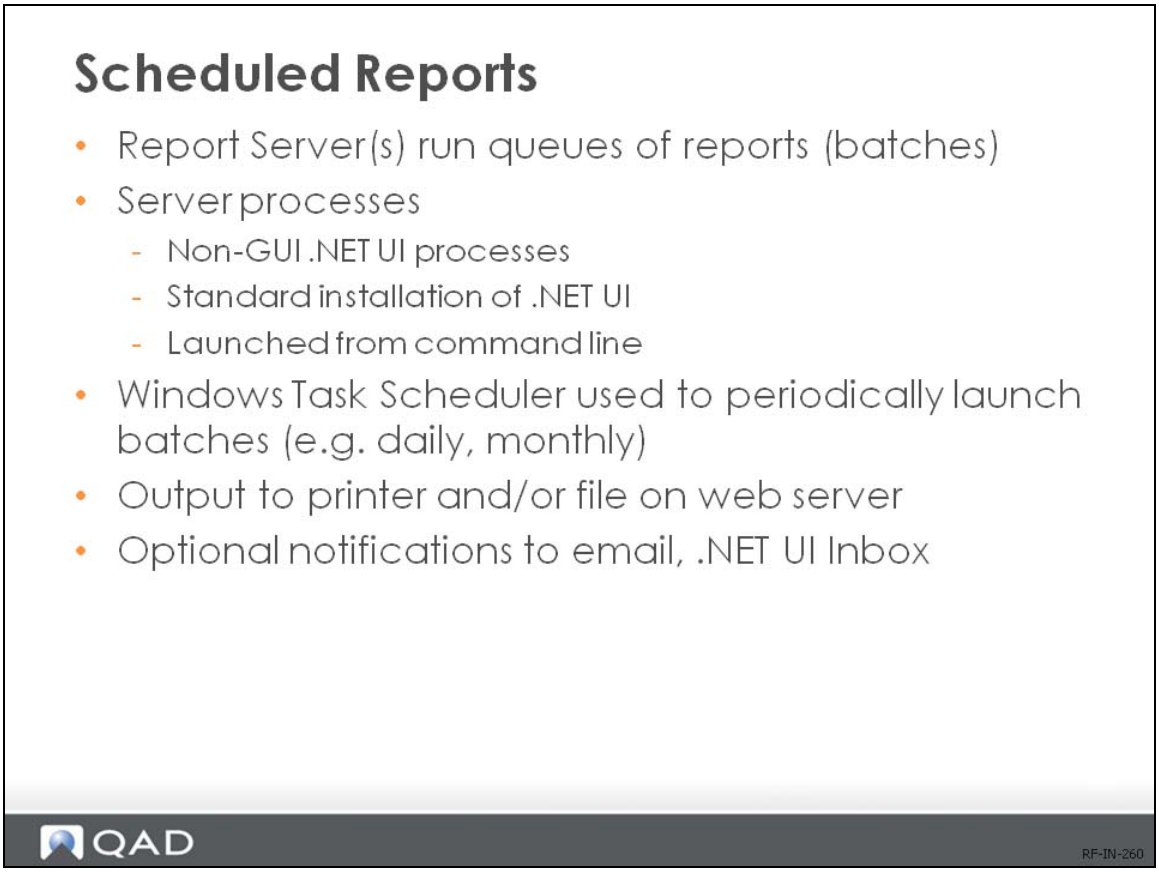
Reporting Framework Architecture: Report Generation



Reporting Server Architecture: Report Batch Processer



Scheduled Reports



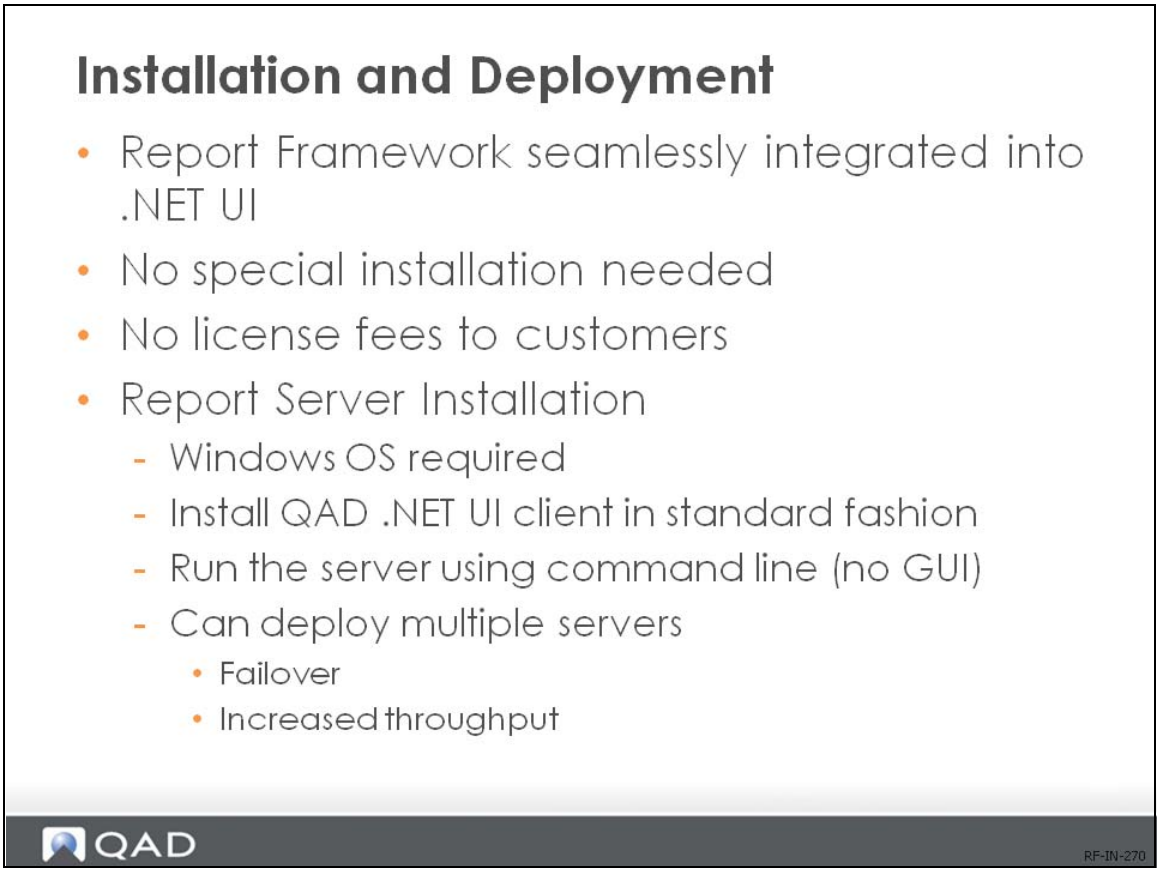
You can automate the process of generating routine reports by scheduling them to automatically run at specified times or intervals and have the reports sent to a specified destination, such as a printer or the document service on the report server.

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

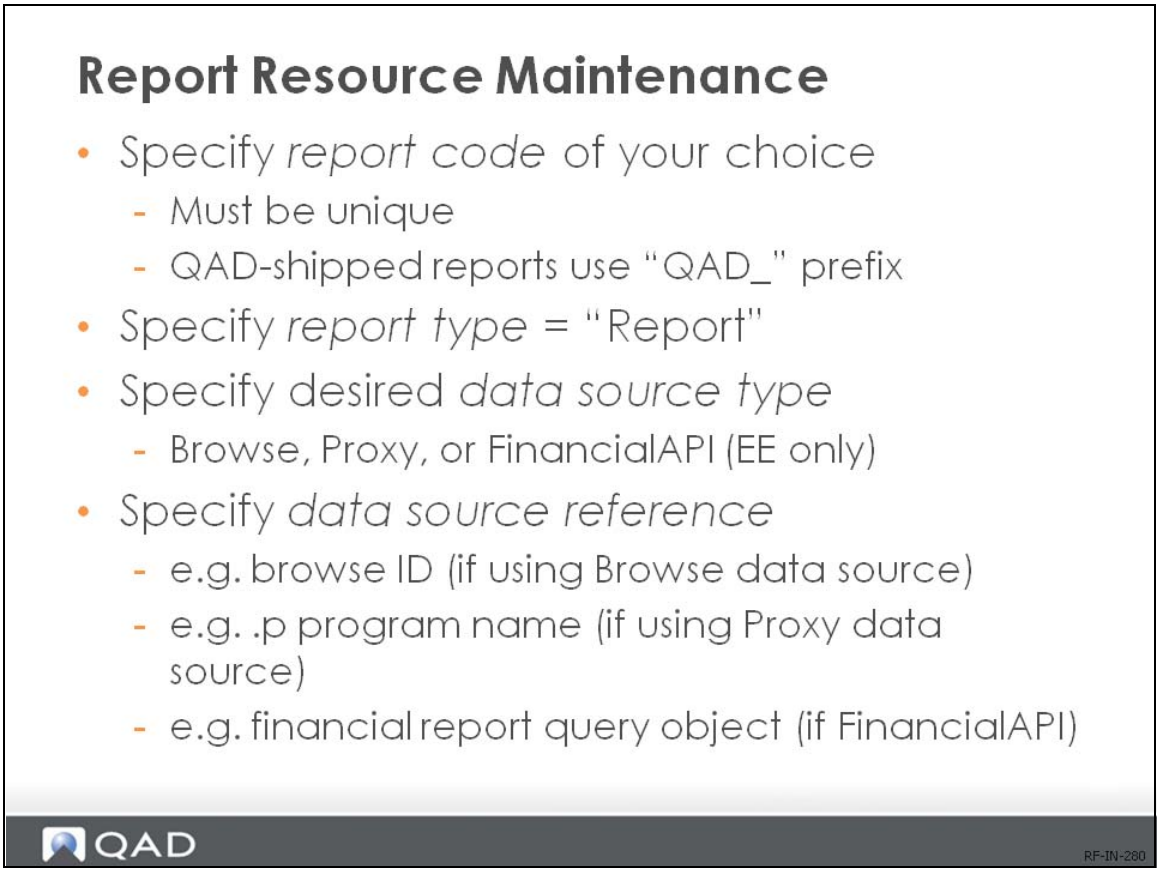
The Windows Task Scheduler should be configured to launch the Report Batch Processor, which is a non-GUI instance of the QAD .NET UI launched from the command line, for a specific batch as scheduled and runs all the scheduled reports grouped in the batch. If already set up, the report outputs are sent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

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



Report Resource Maintenance



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:

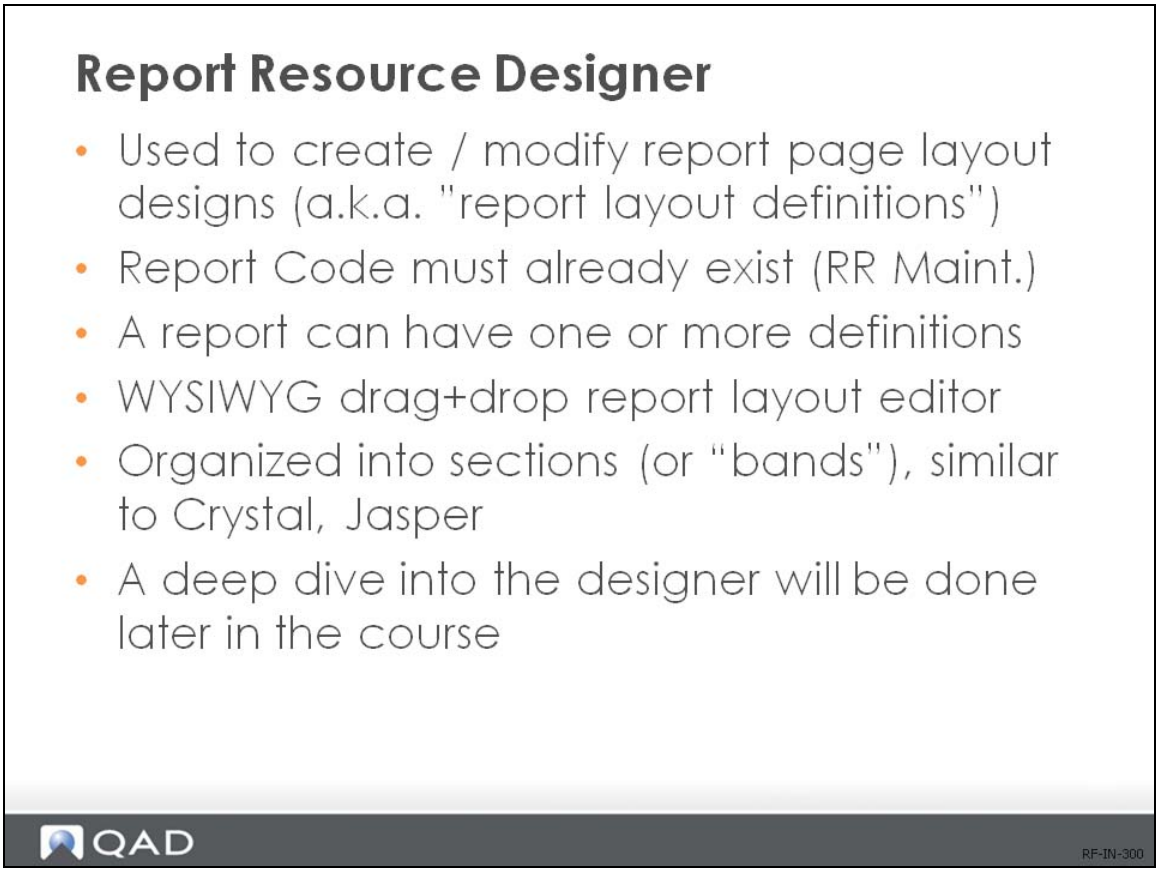
  ```
  BaseAdapters.CBAdapters.QadBrowseAdapter:BJournalEntry.SelectPosting
  ```

- 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



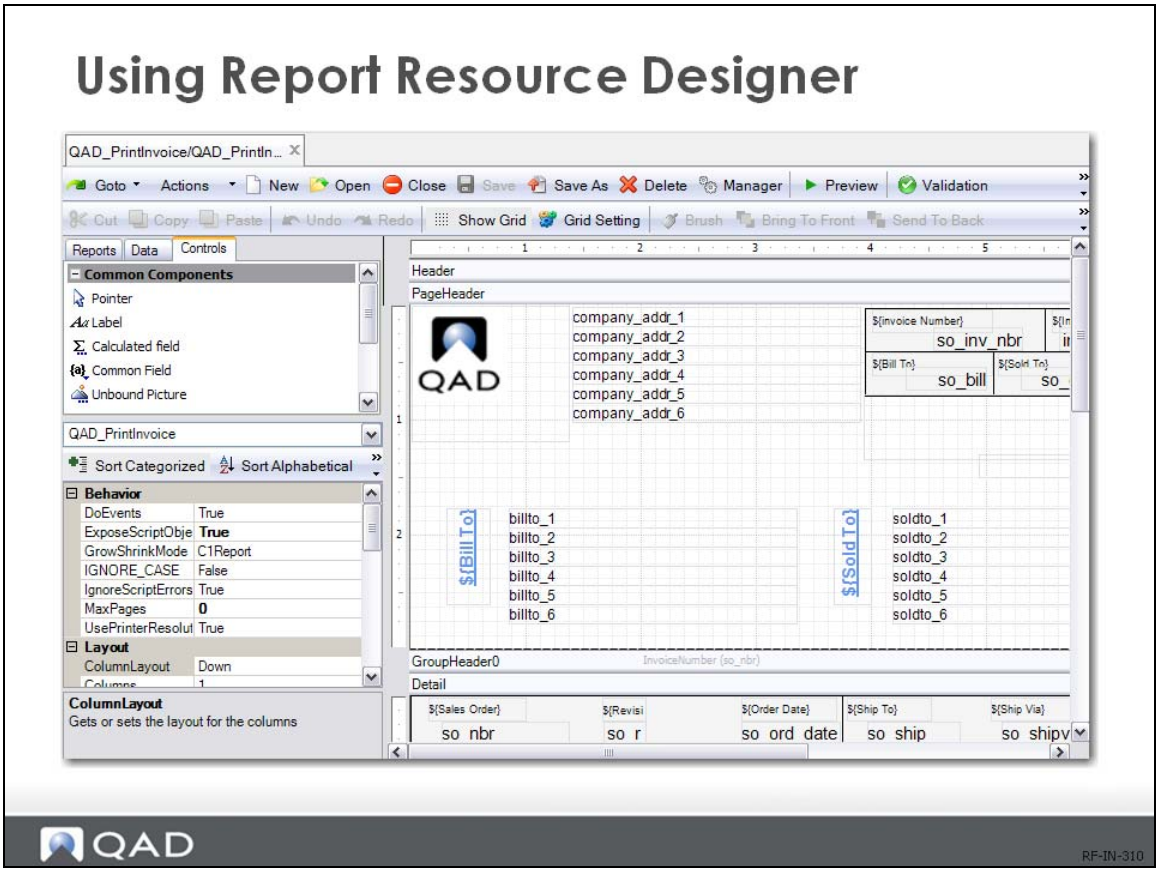
Use the Report Resource Designer to create a report definition.

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.

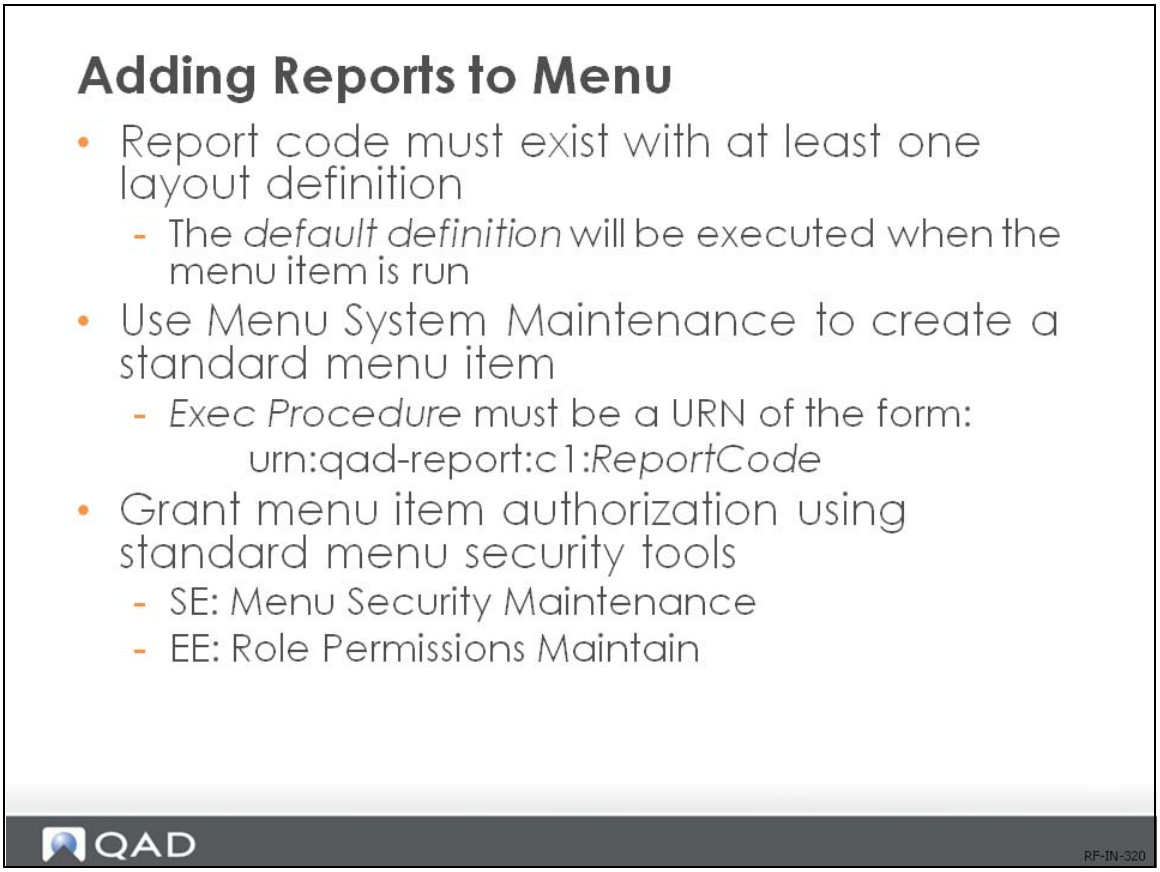
6 This screen offers you several options to define how the data will be organized on the page. Select the layout that best approximates what you want the final report to look like.

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



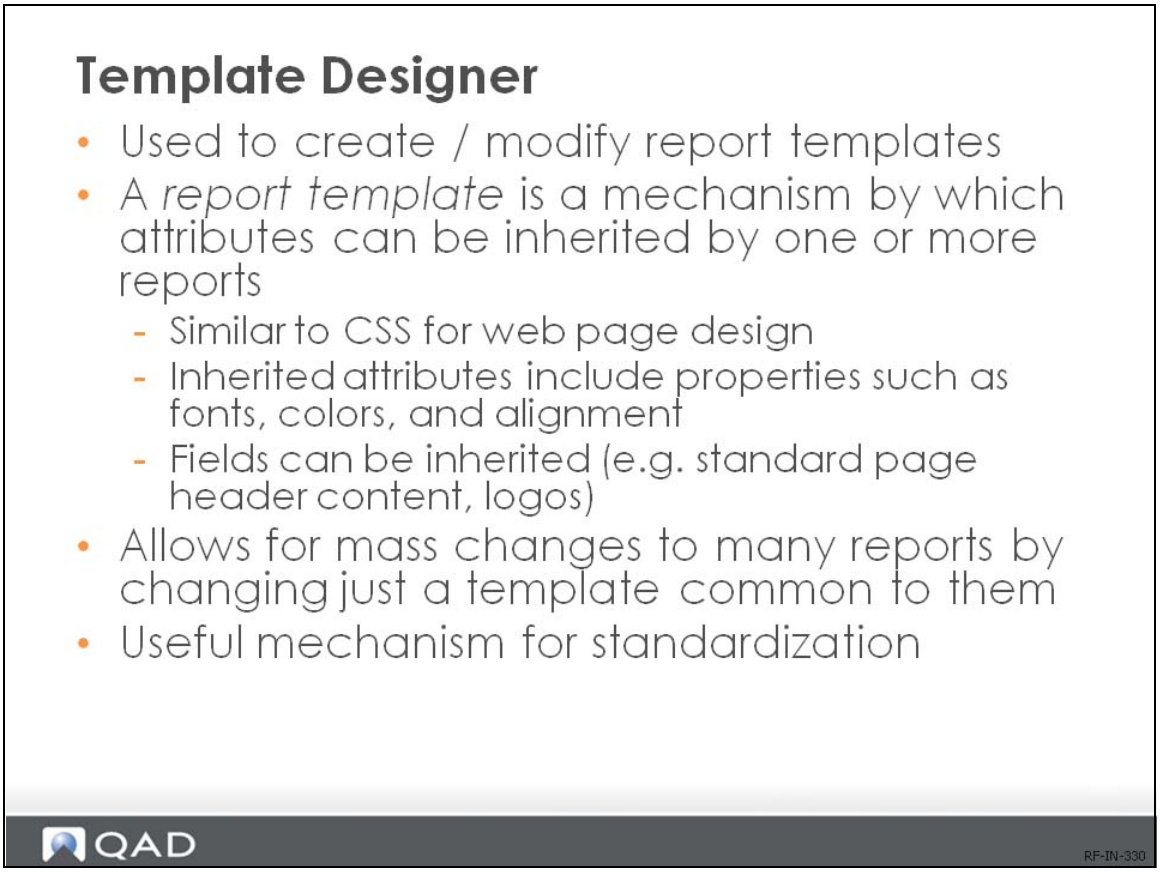
Users open reports through report menu items that they have been given access to based on their role membership. Use Menu System Maintenance to create a menu item to provide access to a report resource.

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
```

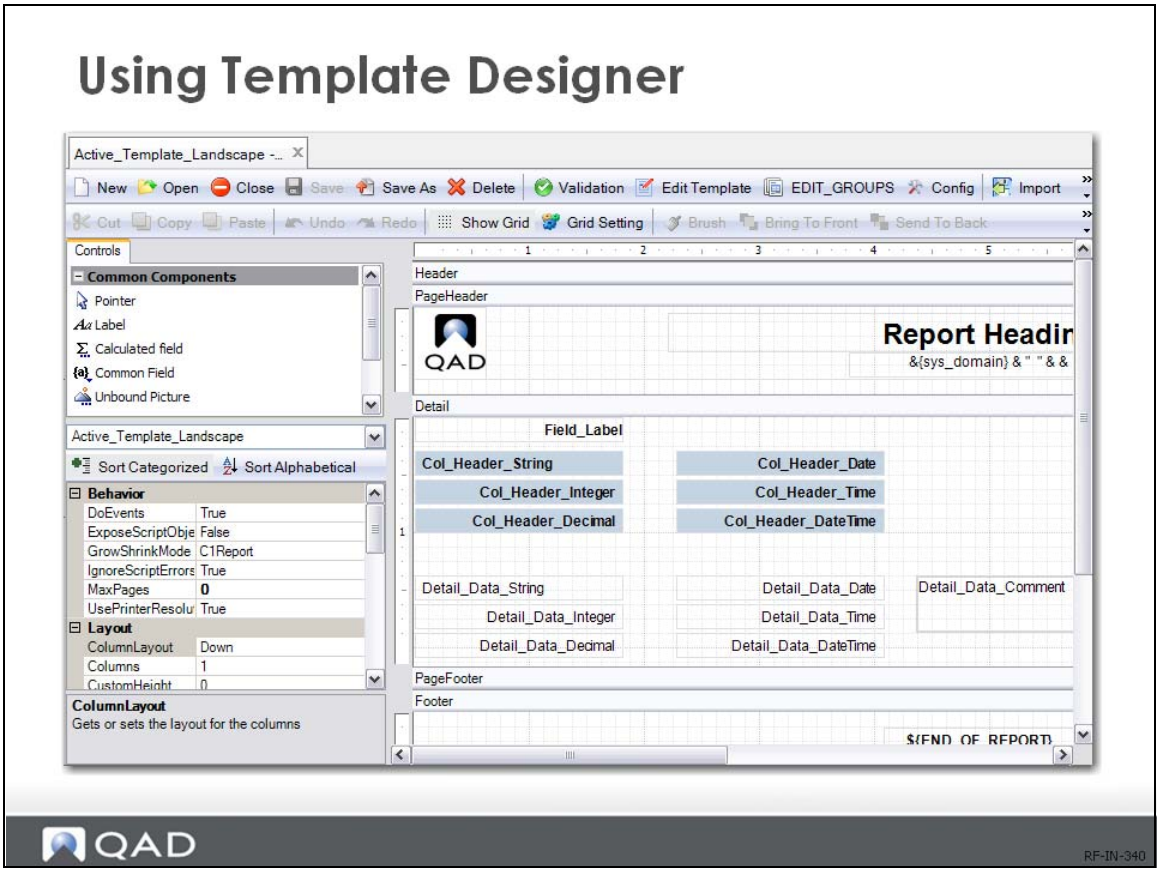
Where *ReportCode* is the report code of the report resource you are creating a menu item for.

Template Designer



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

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.

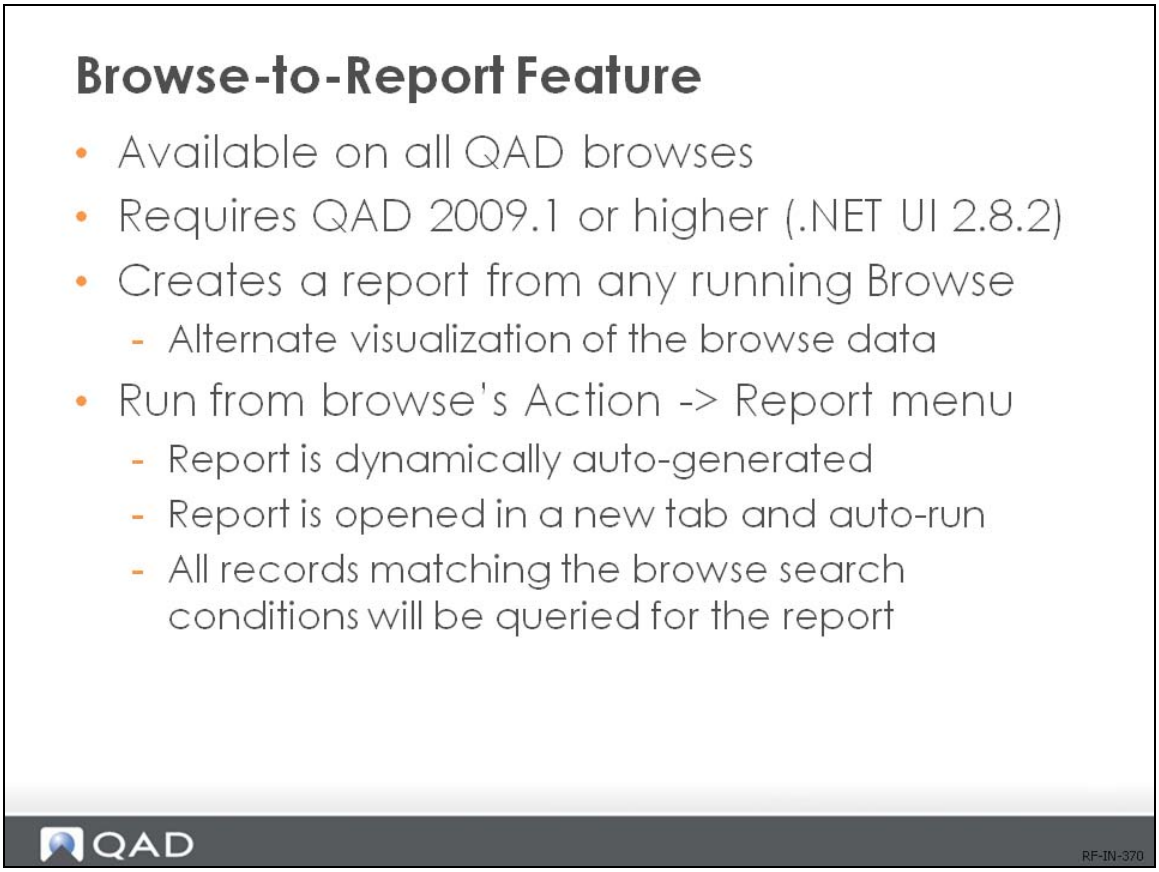
Use Report Resource Import to import report resource data files into the system.

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.

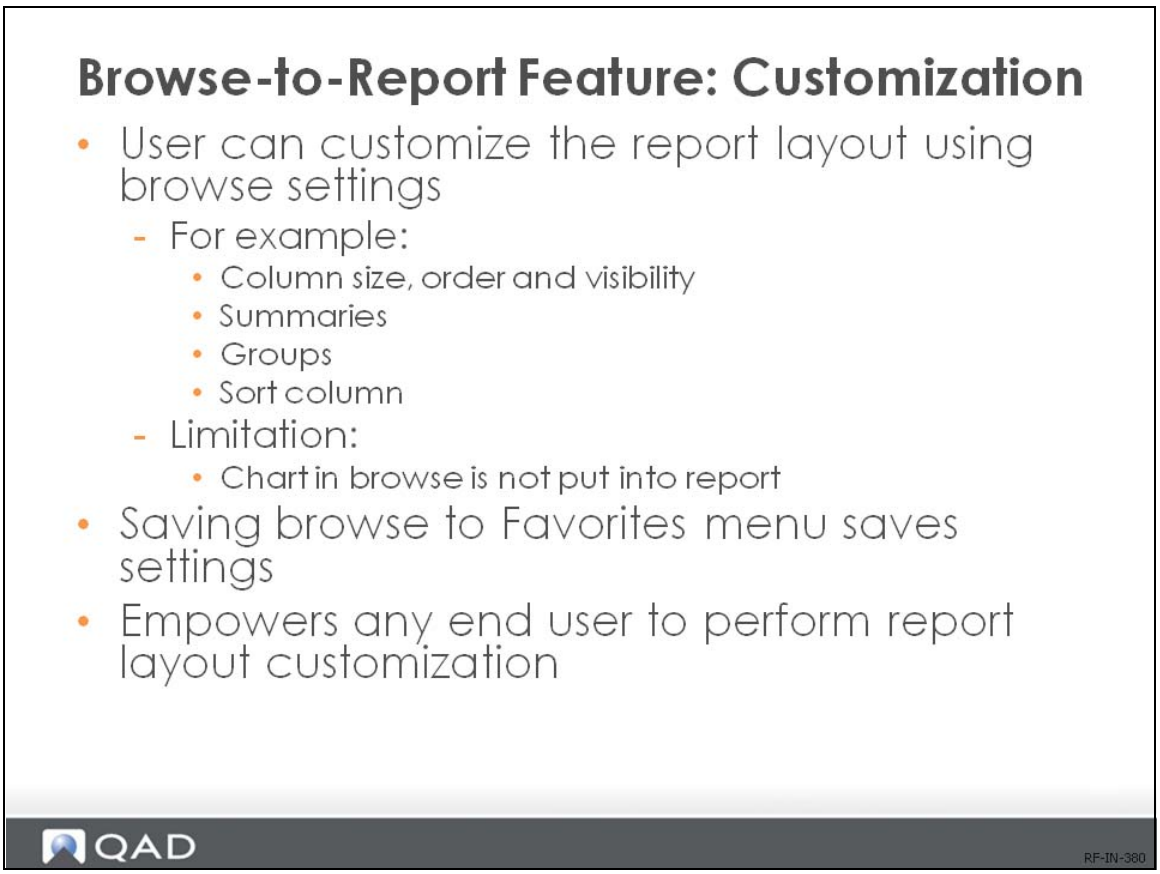
Report Import/Export and Development Process 2

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).

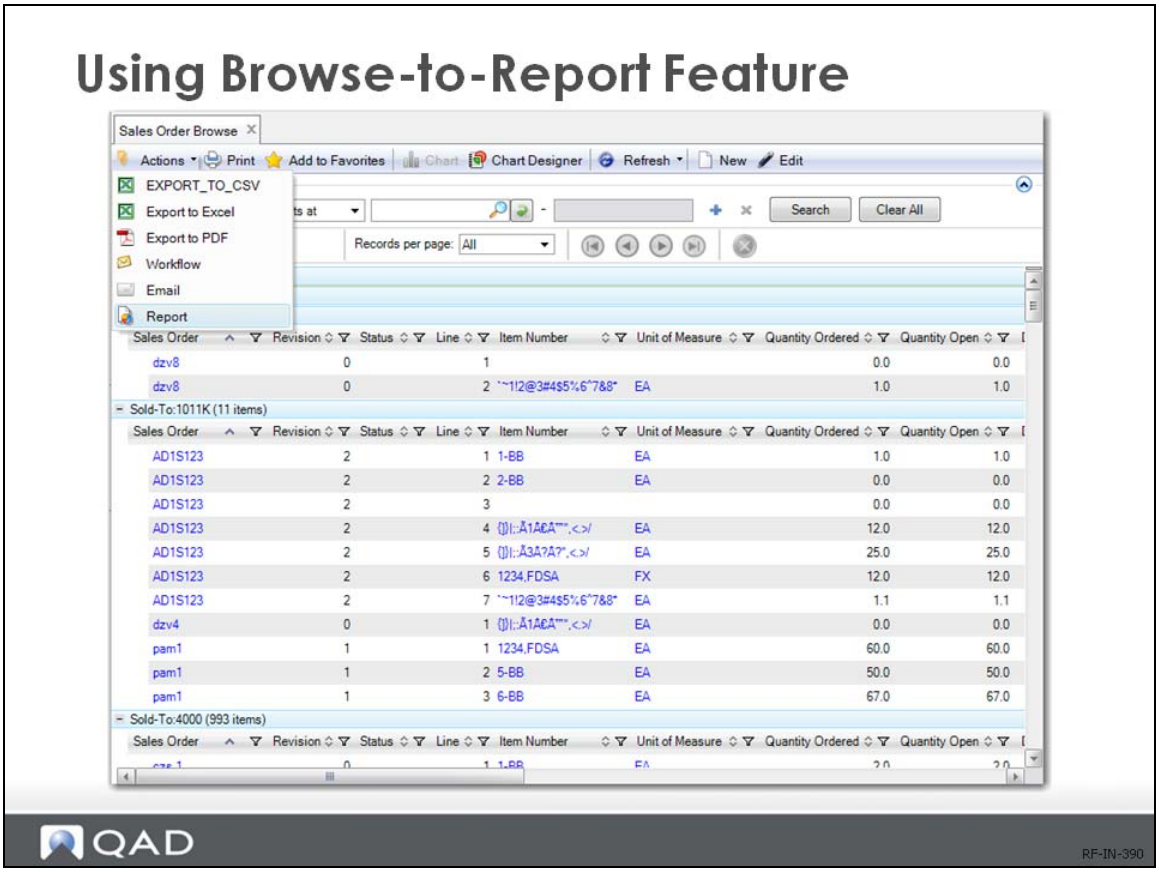
Browse-to-Report Feature



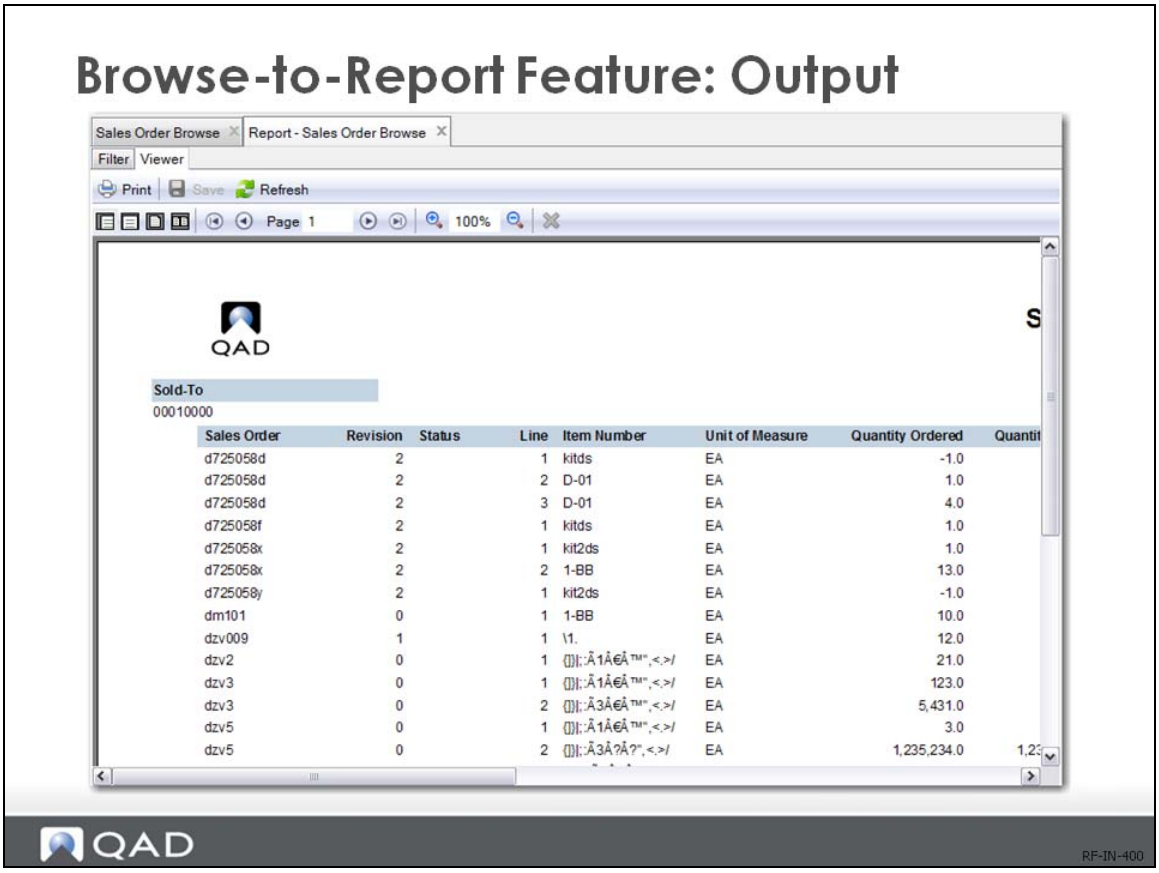
Browse-to-Report Feature: Customization



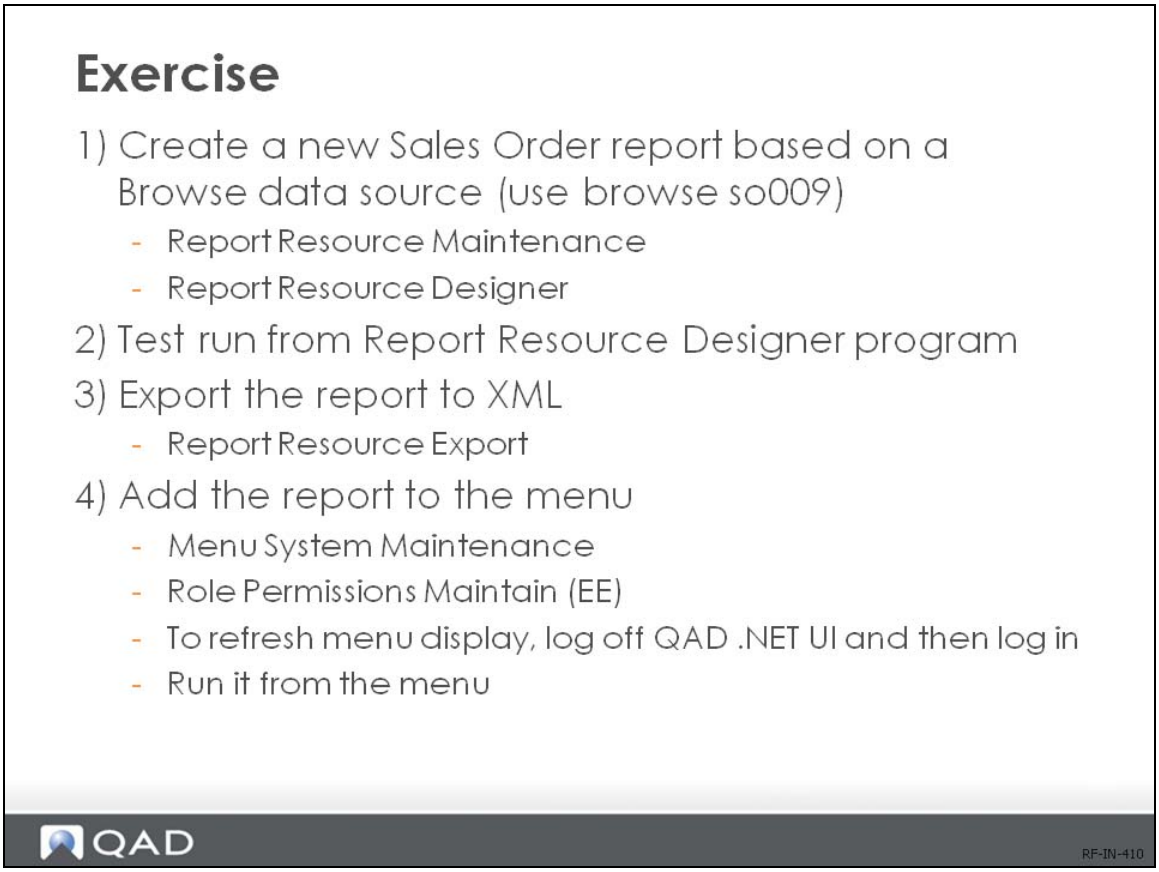
Using Browse-to-Report Feature



Browse-to-Report Feature: Output



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)
- To refresh menu display, log off QAD .NET UI and then log in
- Run it from the menu
QAD
RF-IN-410

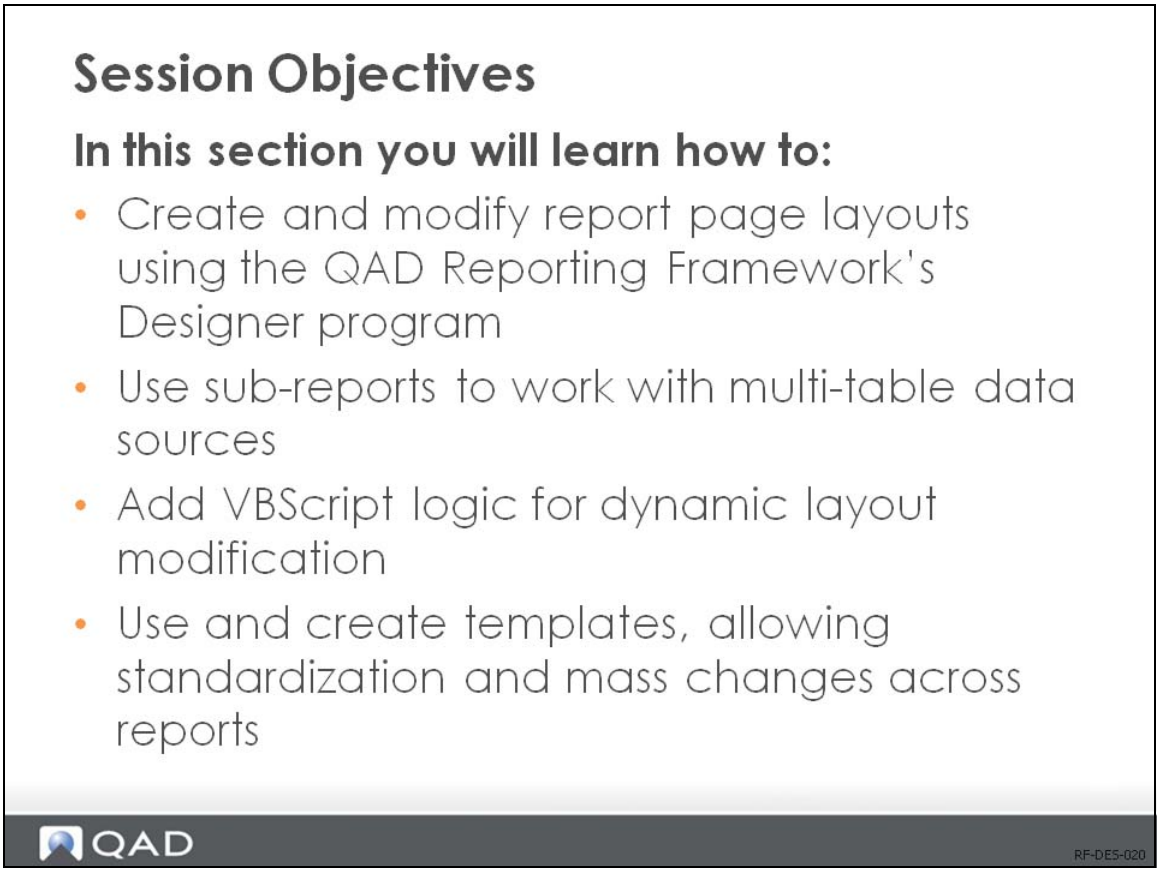
Summary

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.

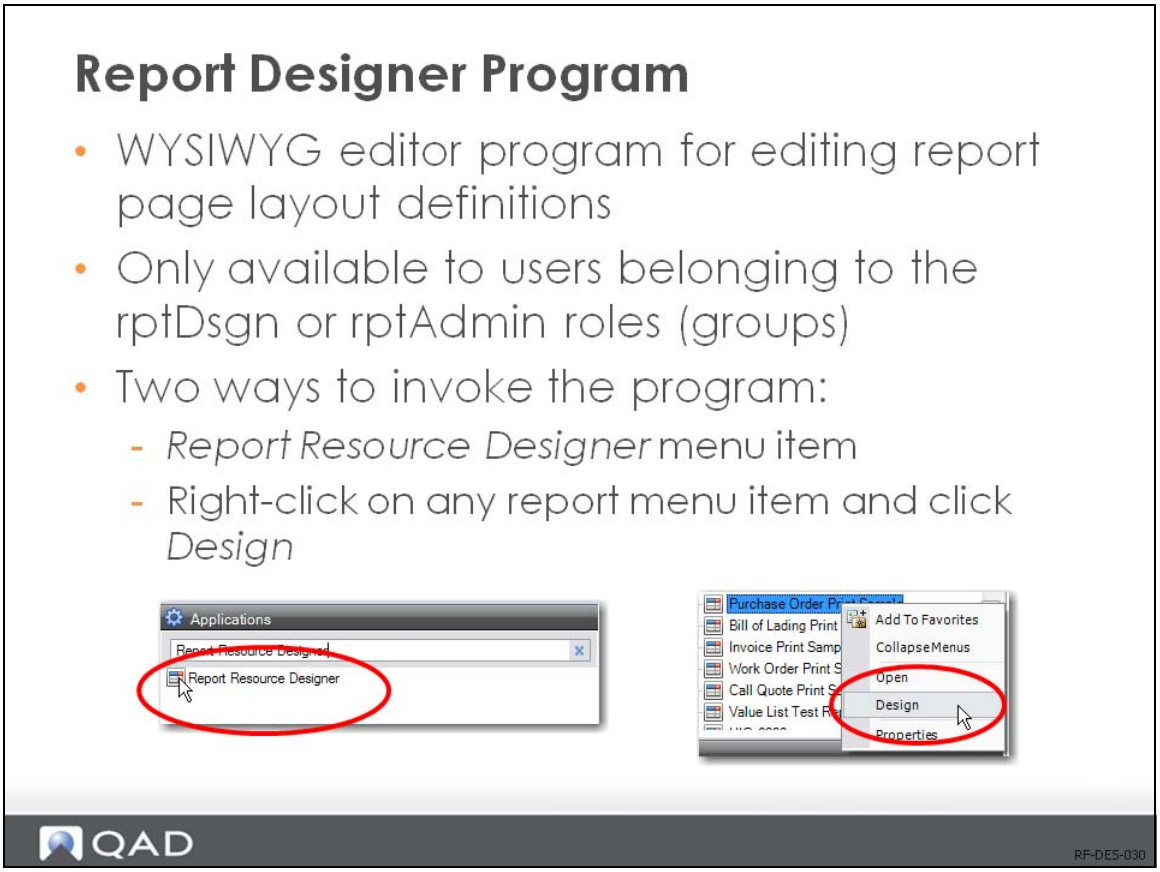
Chapter 2

Development Using Designer

Session Objectives



Report Designer Program

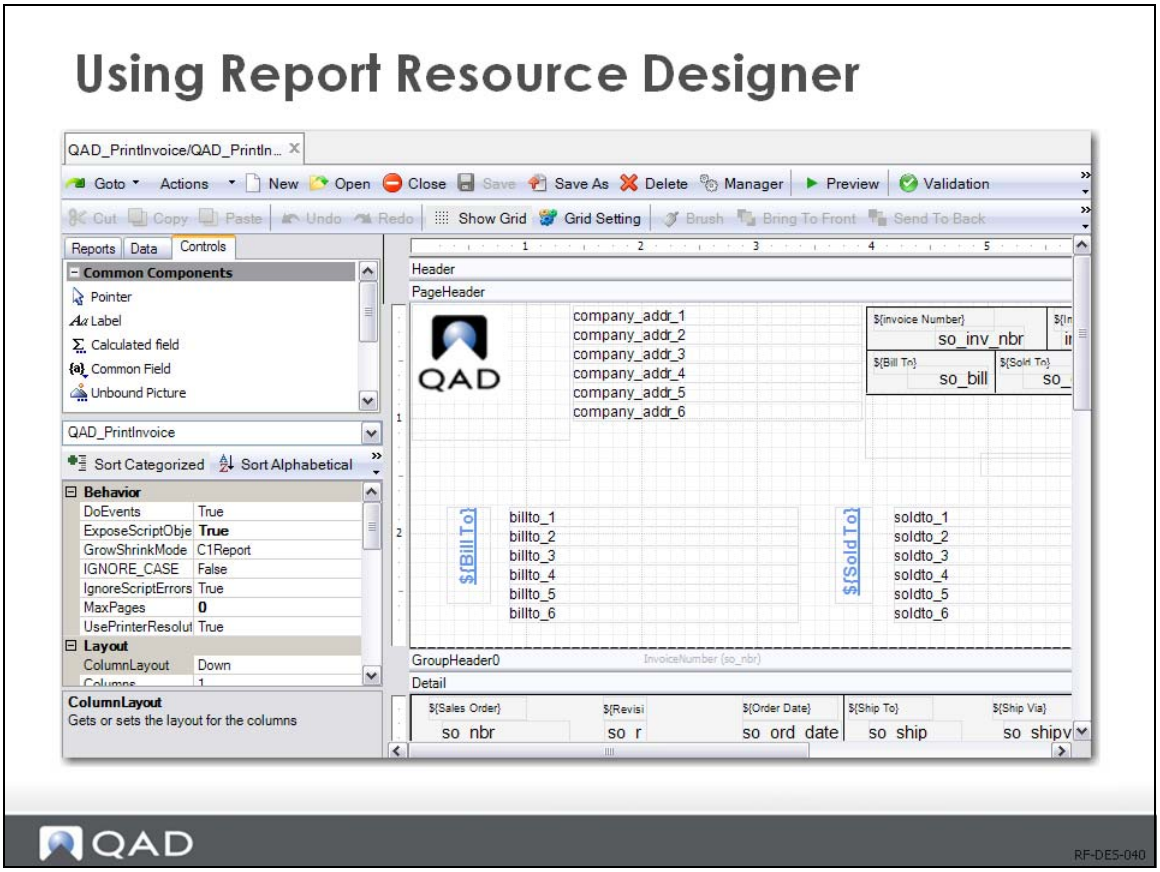


Use one of the following ways to access Report Designer:

- 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

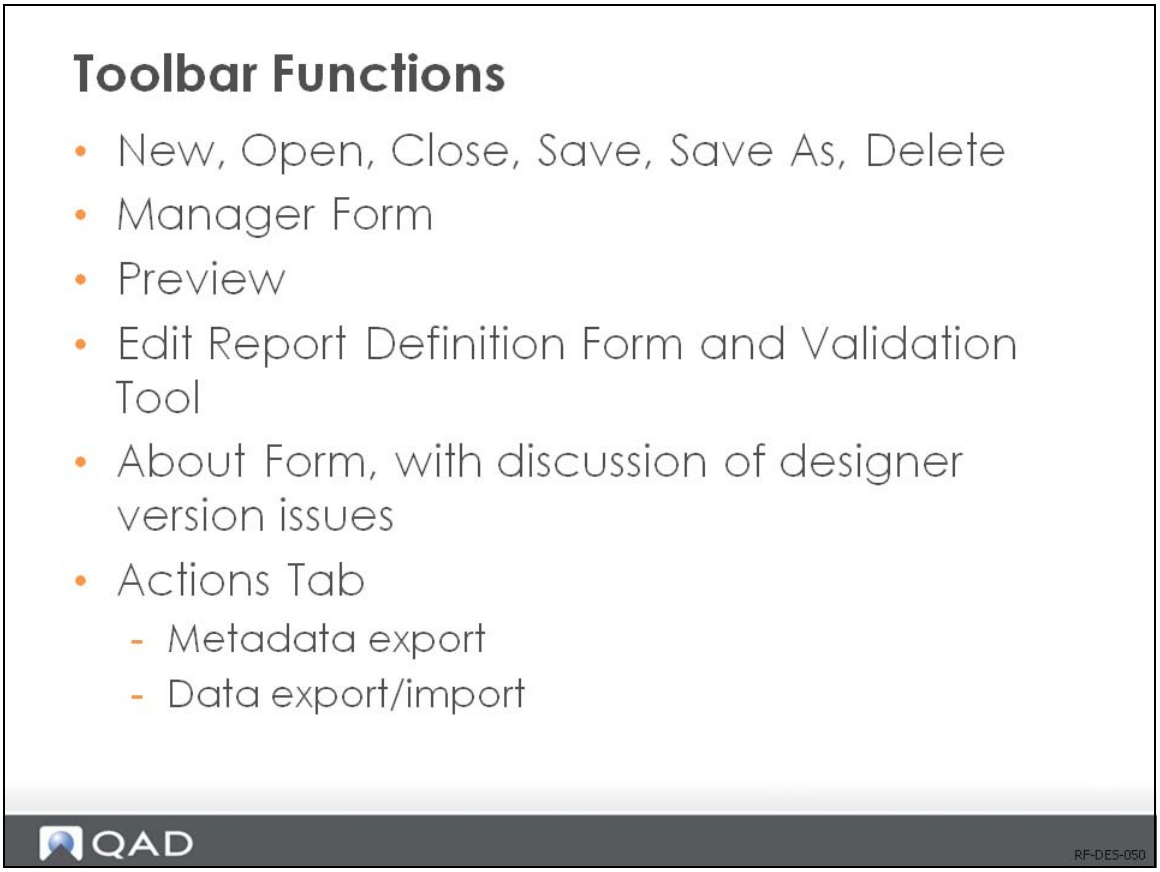


In the Report Designer, you can create, load, and delete report definition files, as well as edit them either in the WYSIWYG (What You See Is What You Get) or code edit mode.

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



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.

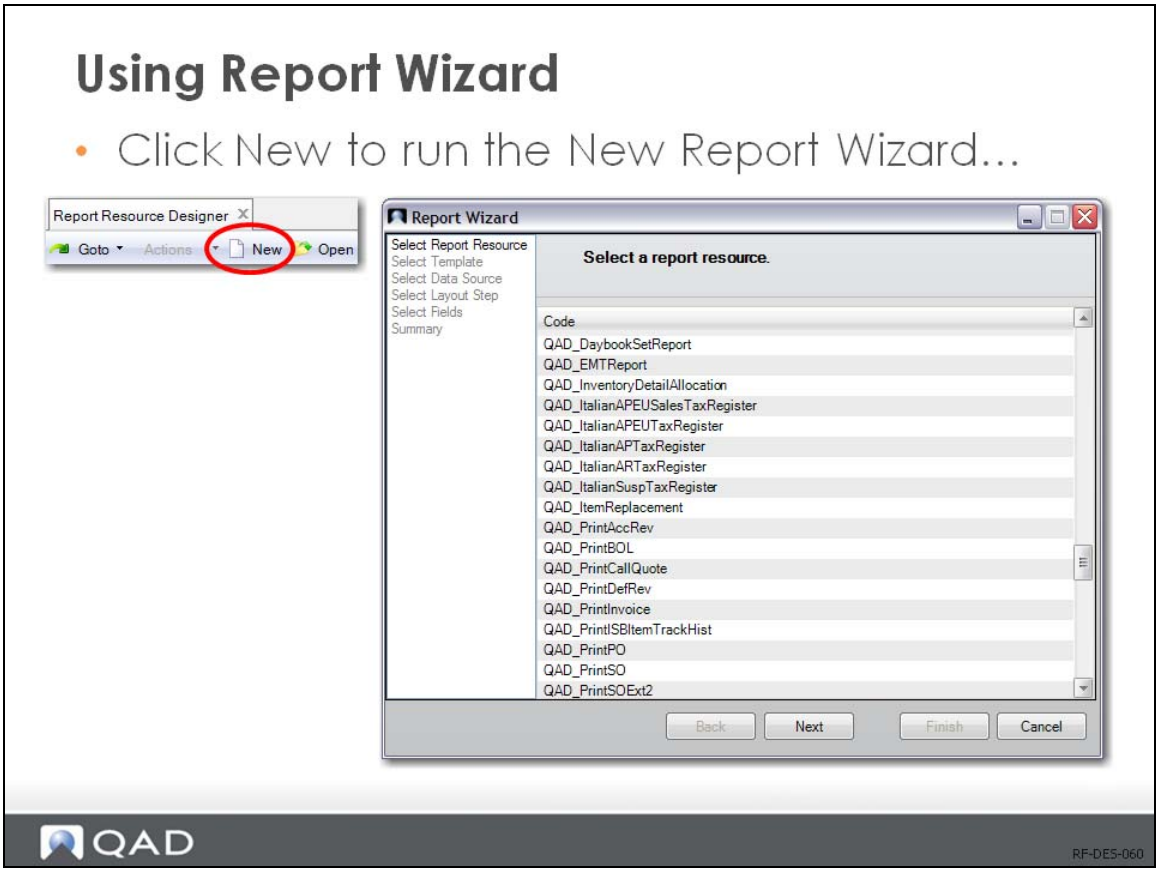
In the Save As dialog box, specify the name of the XML file and where you want to save the file. By default, the file is named after the current report with _meta.xml appended.

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.

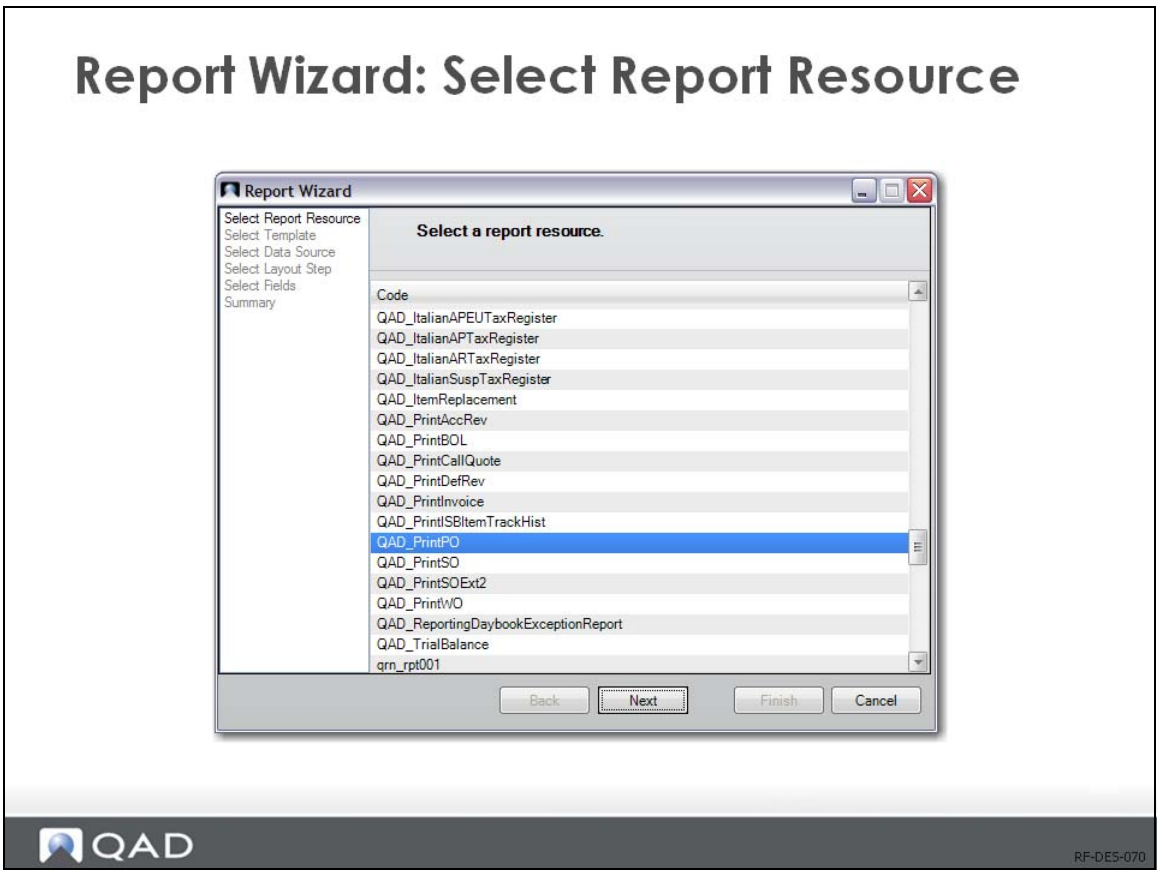
Using Report Wizard



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.

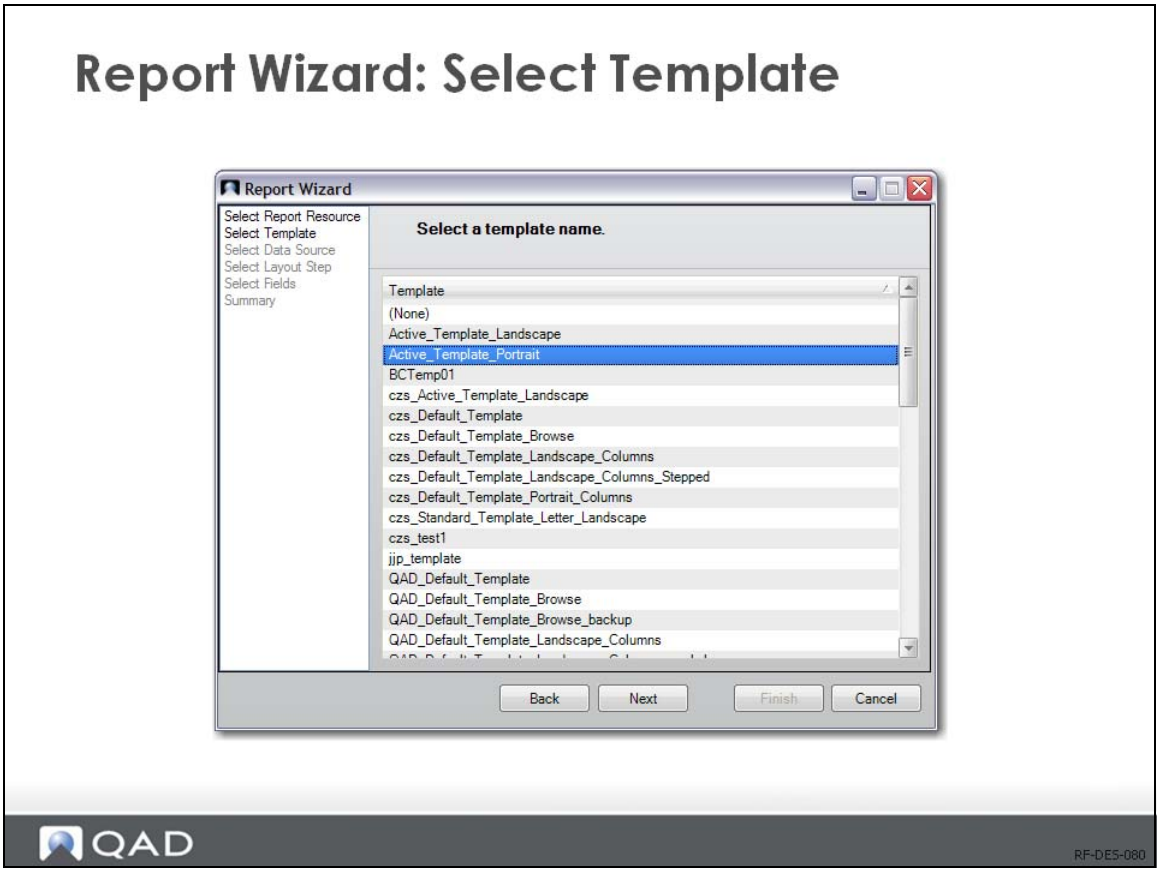
First, click New to launch the Report Wizard.

Report Wizard: Select Report Resource



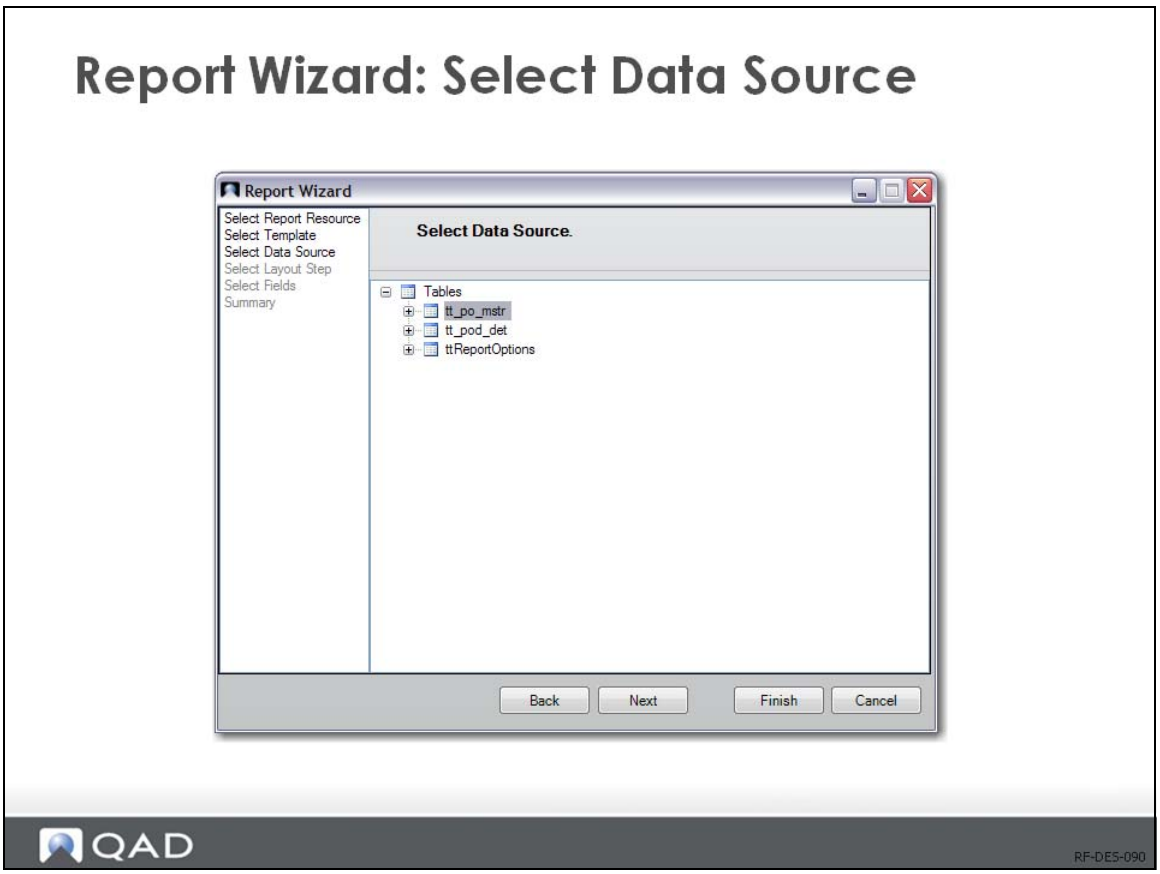
Select a report resource and click Next.

Report Wizard: Select Template



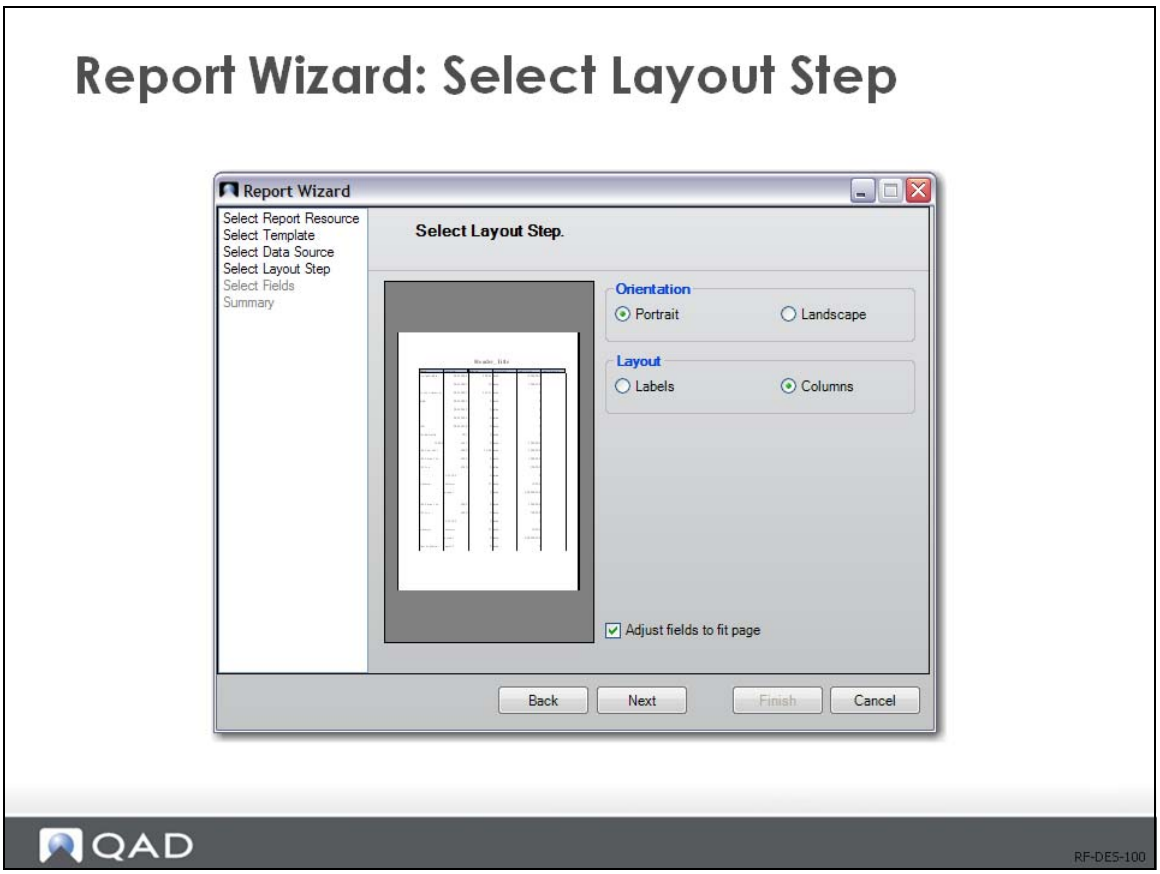
Select a template name such as Active_Template_Portrait and click Next.

Report Wizard: Select Data Source



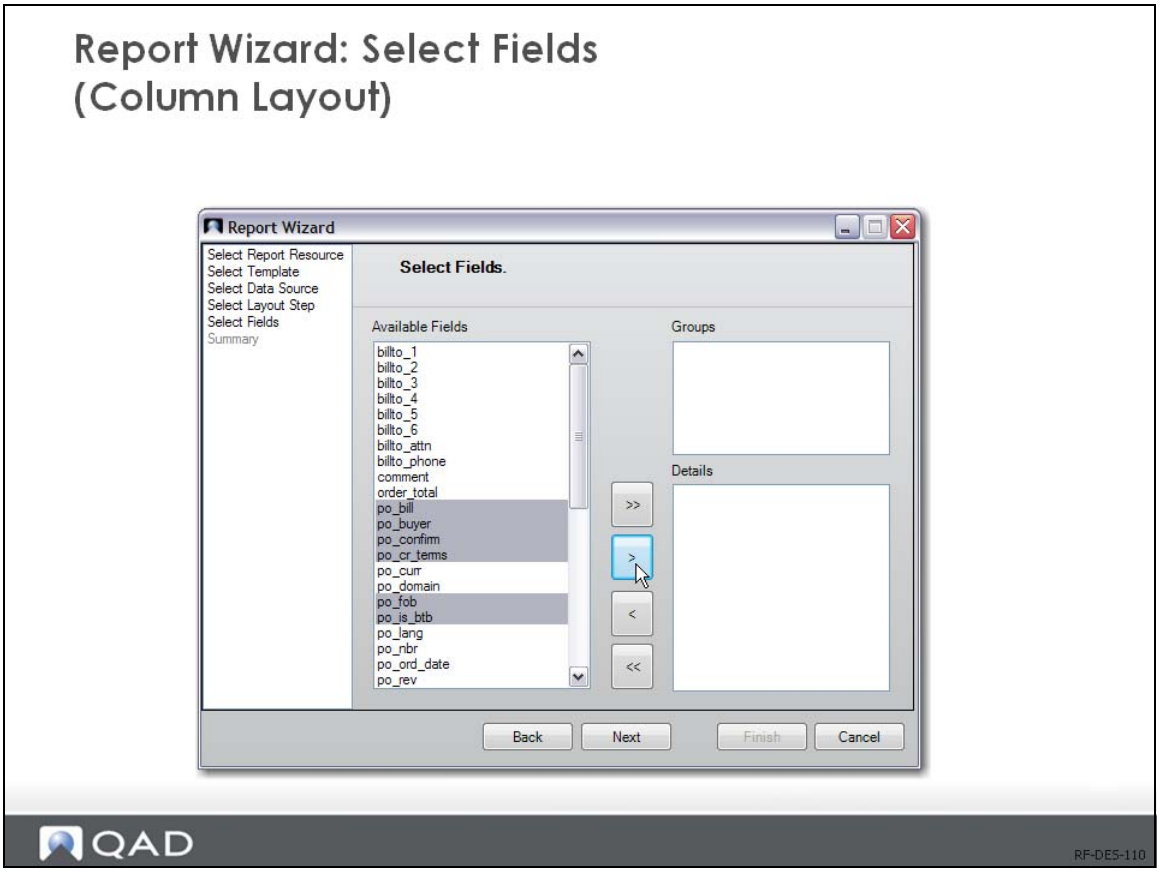
Select a data source and click Next.

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)

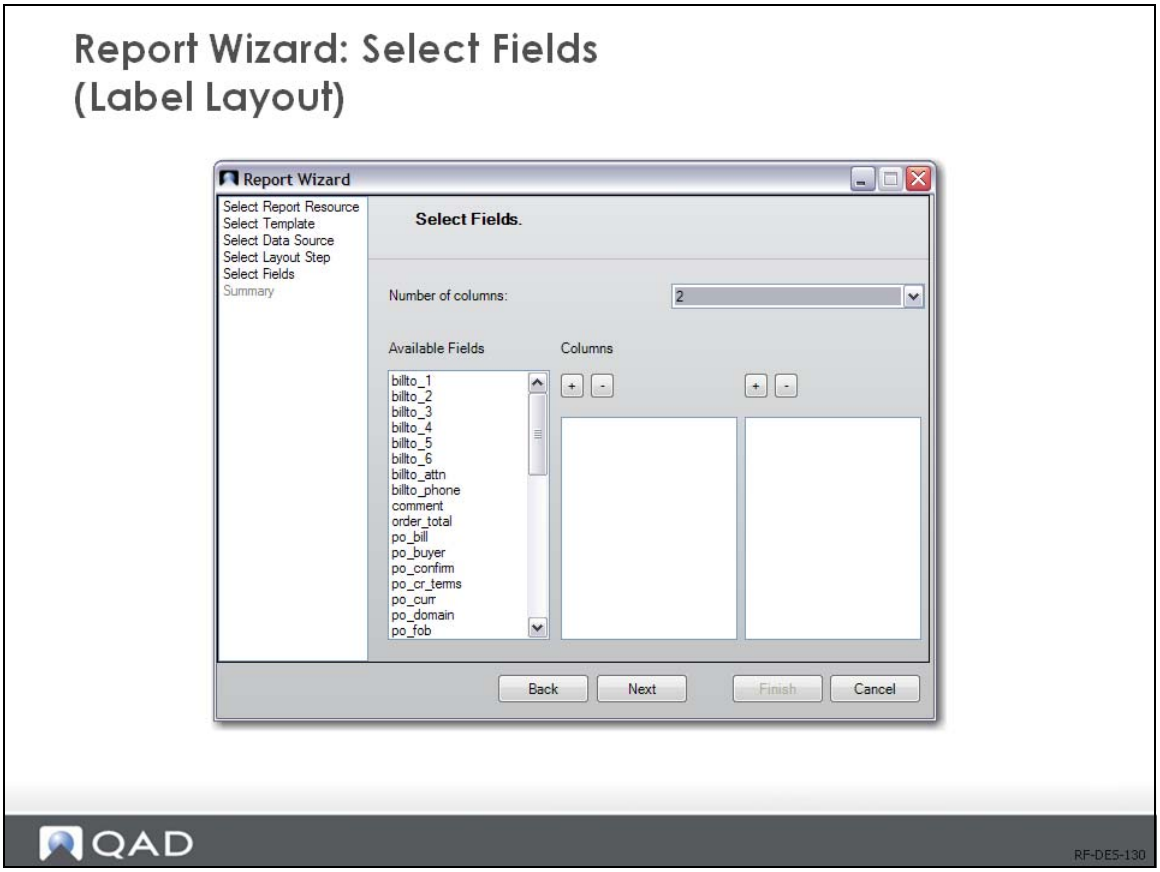


For Column layout, you select the fields you want to use from the available fields of your data source. Optionally, you can organize the columns into groupings. This feature is similar to the “group by” display feature in browses.

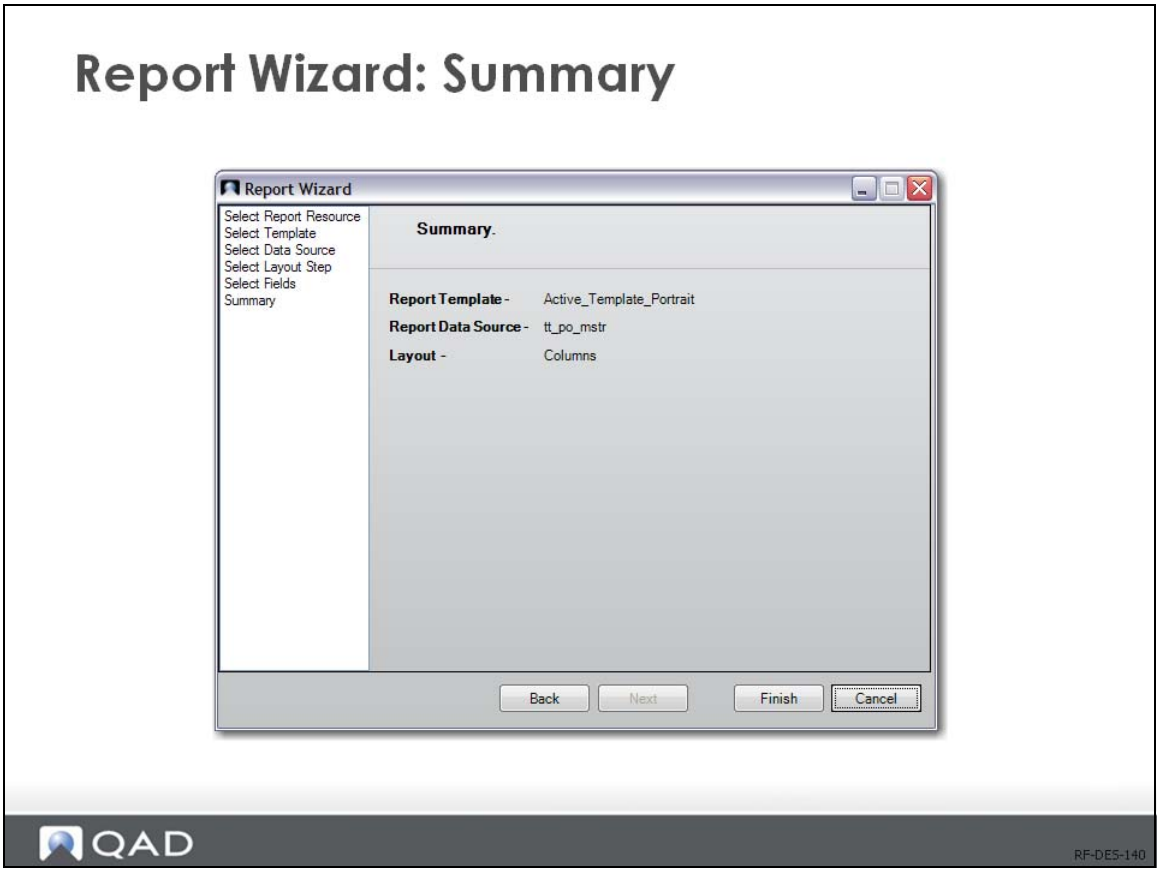
Report Wizard: Select Fields (Column Layout, Groups)

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)

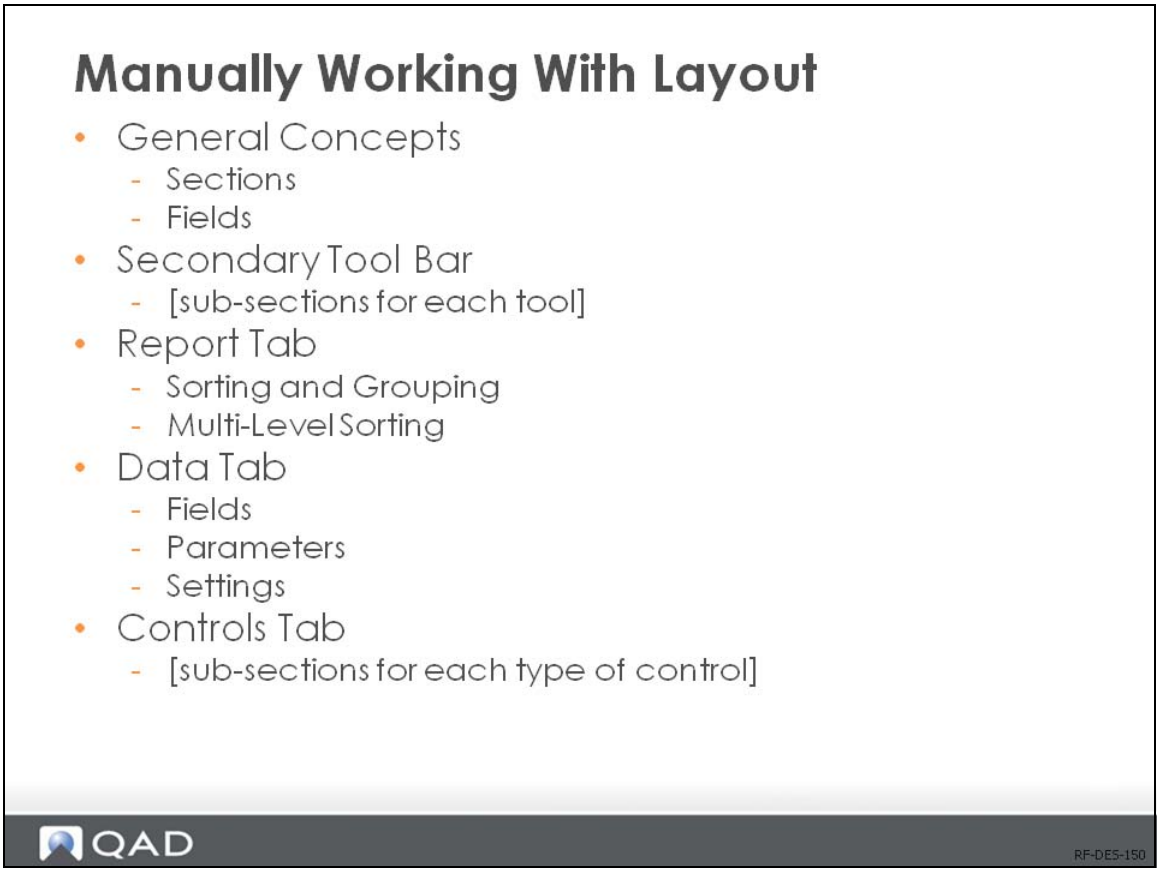


Report Wizard: Summary



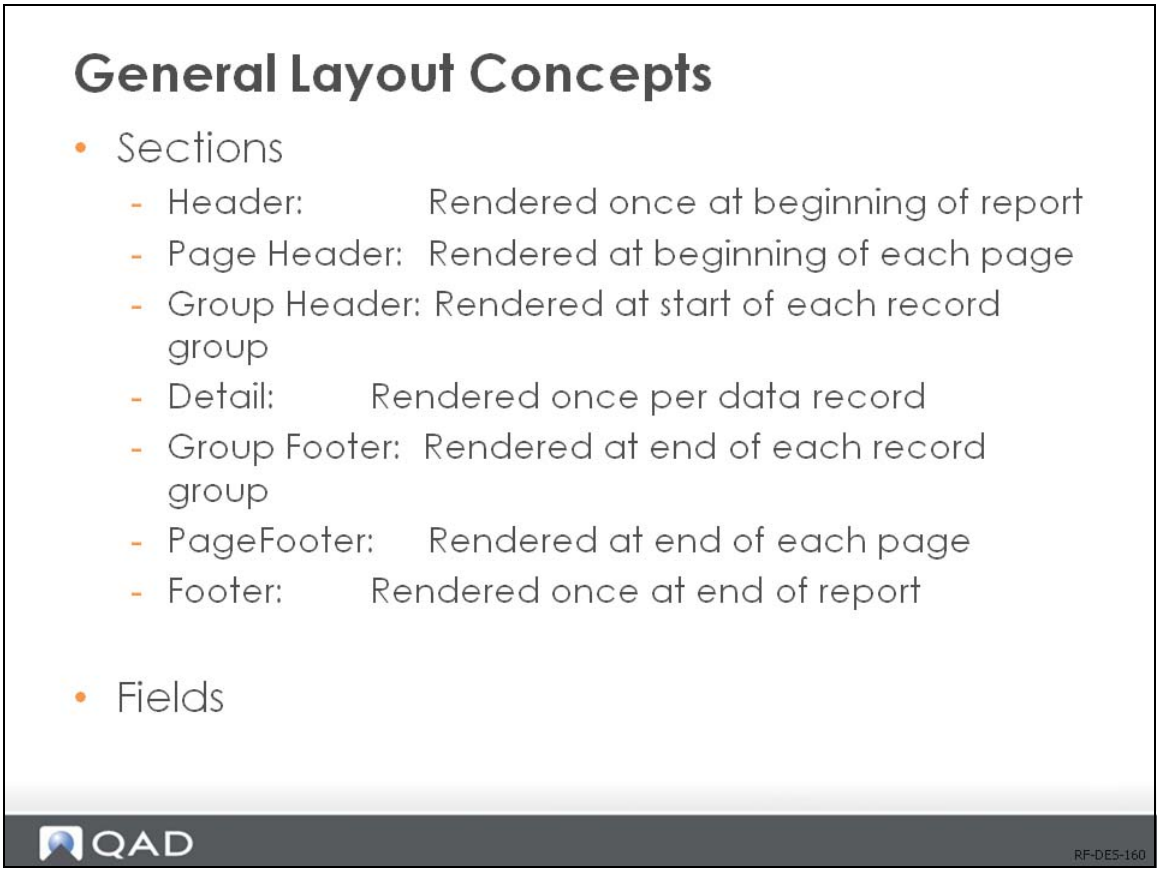
Upon completion, the Report Wizard's summary screen displays information about the template, data source, and layout. Finally, Click Finish.

Manually Working With Layout



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

General Layout Concepts



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

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.

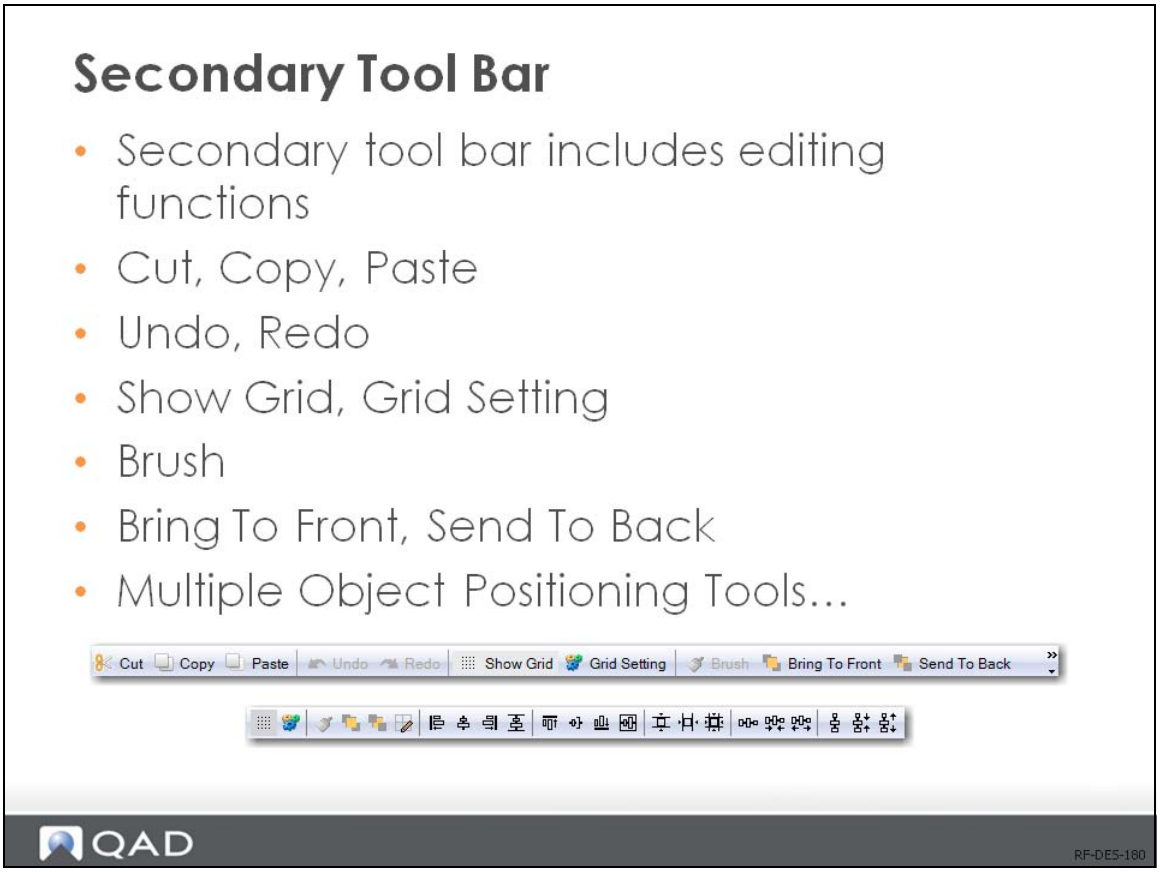
Preview. Display the current report definition in preview mode. In the preview mode, you can only navigate through the generated report.

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



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.

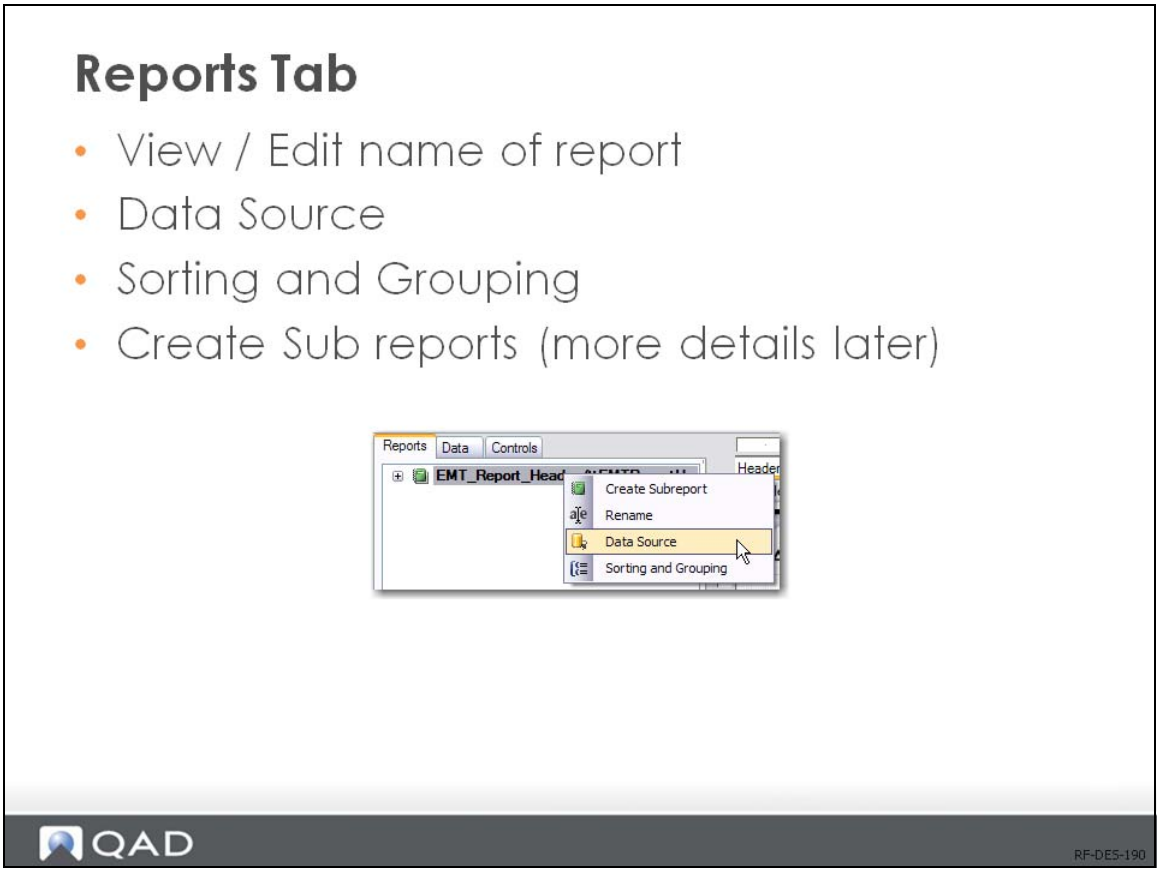
Increase Horizontal Spacing. Horizontally increment spacing between multiple selected objects.

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.

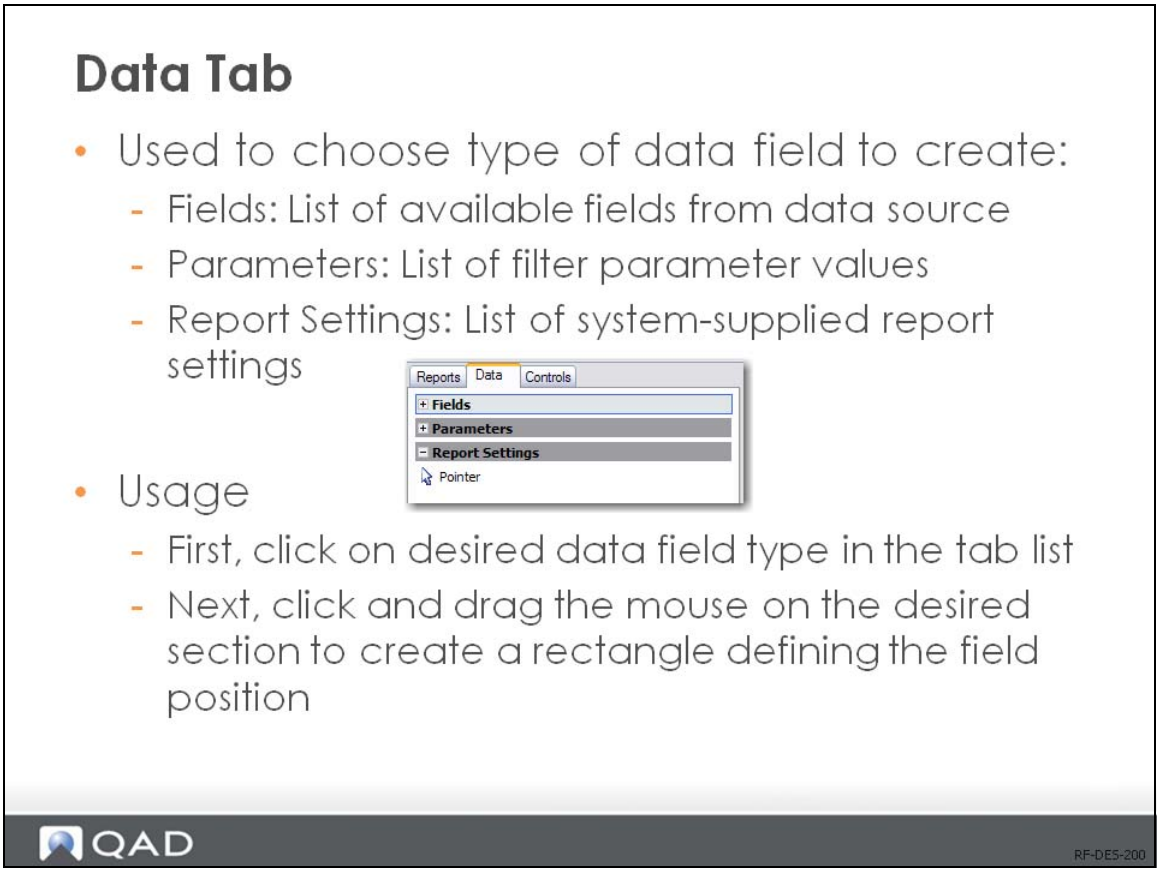
Reports Tab



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



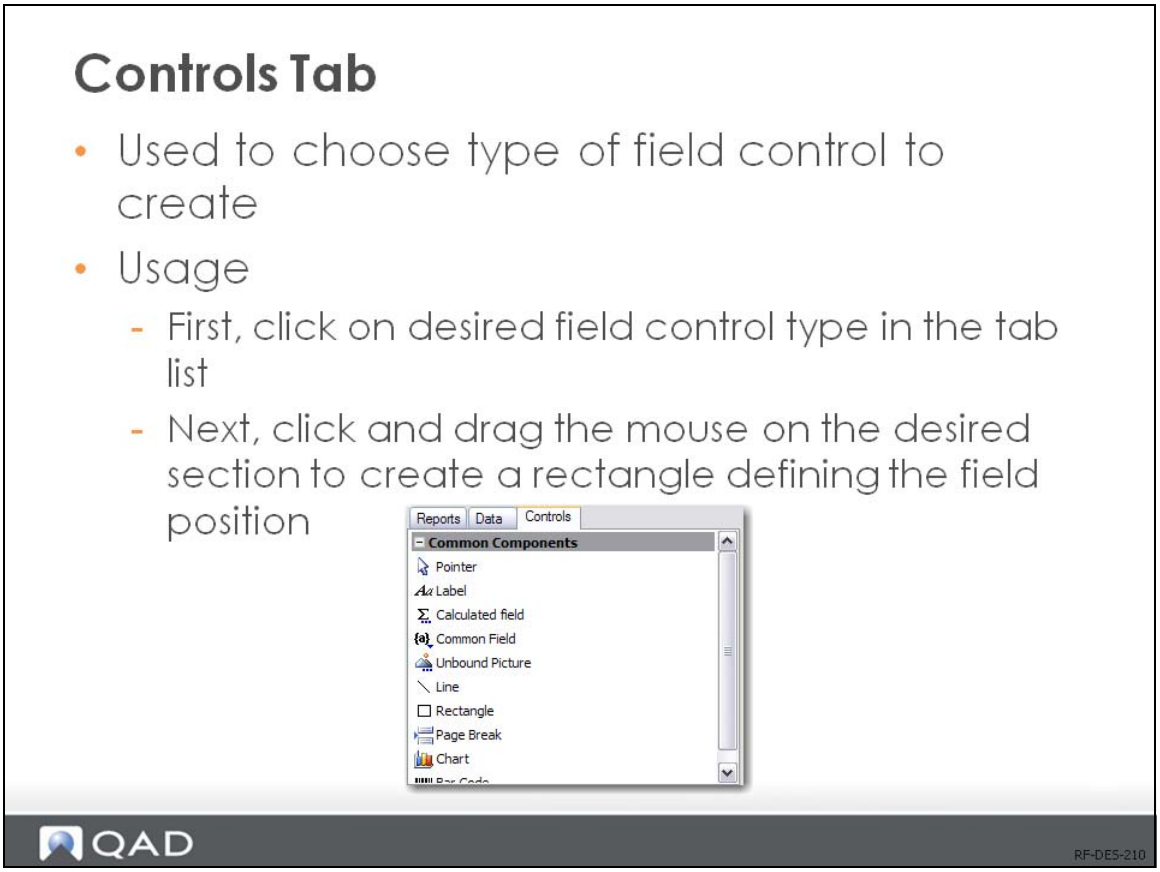
The Data tab contains three groups of data: fields, parameters, and report settings. Except for the Pointer button, which is used to deselect any currently selected object, each button under these groups creates a field on the report and initializes its properties.

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.

Controls Tab



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.

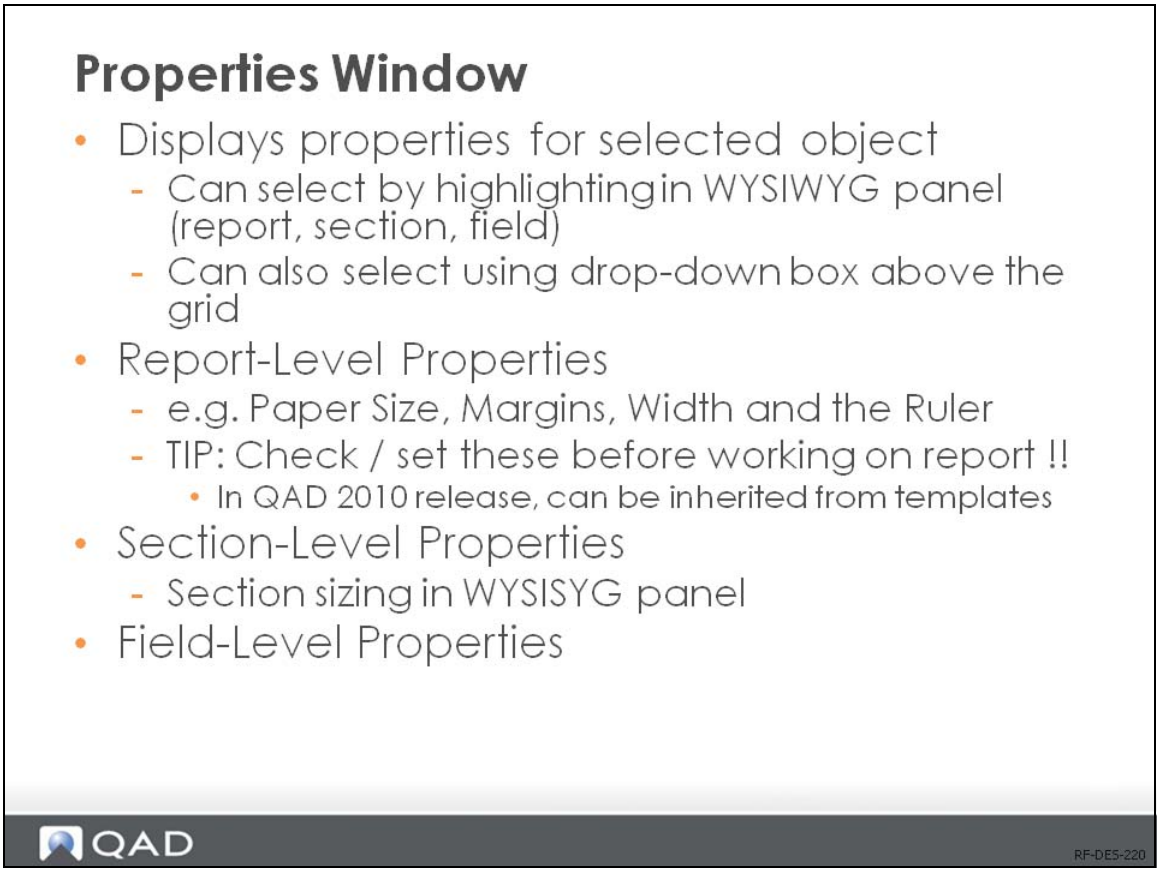
Line, Rectangle. Simple geometric shapes.

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



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

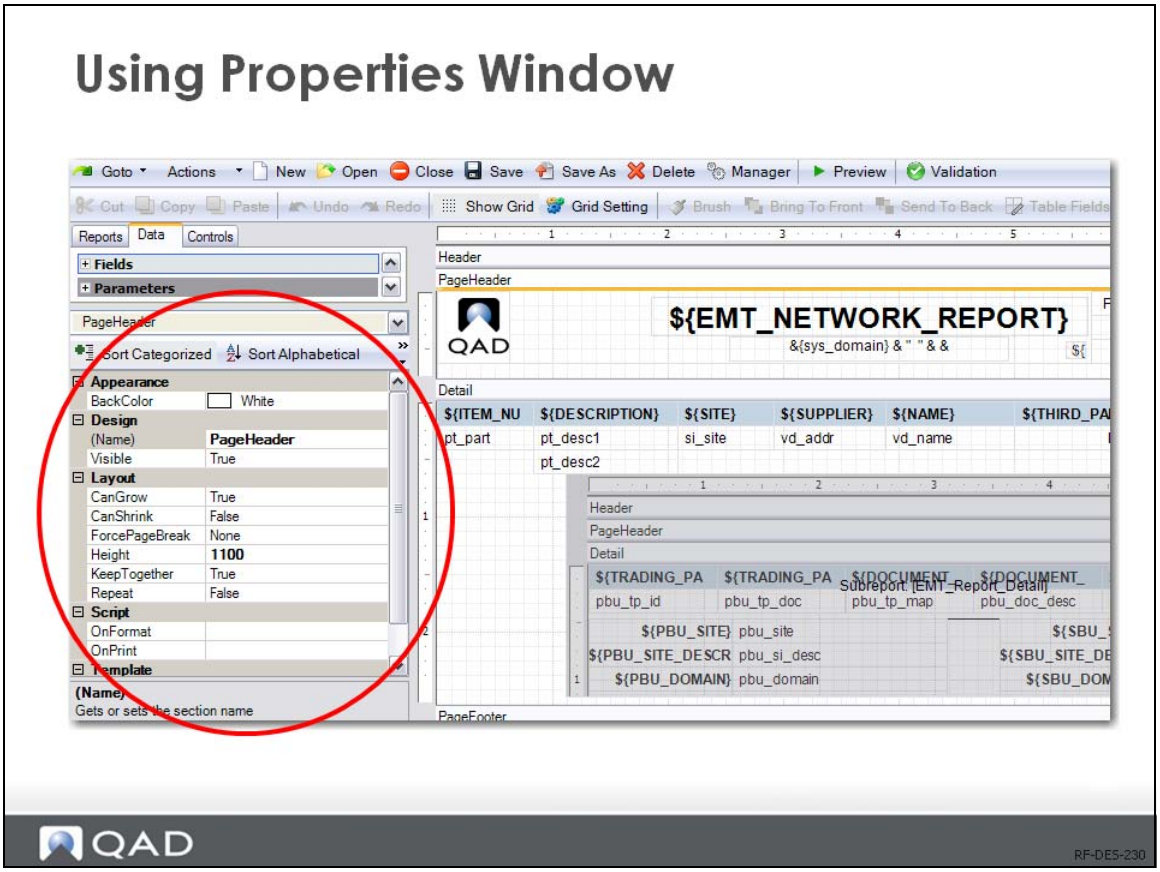
When one or more fields are selected, the Properties window shows property values that all fields have in common, and leaves the other properties blank. If no fields are selected and you click on a section (or on the bar above a section), the Section properties are displayed. If you click the gray area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

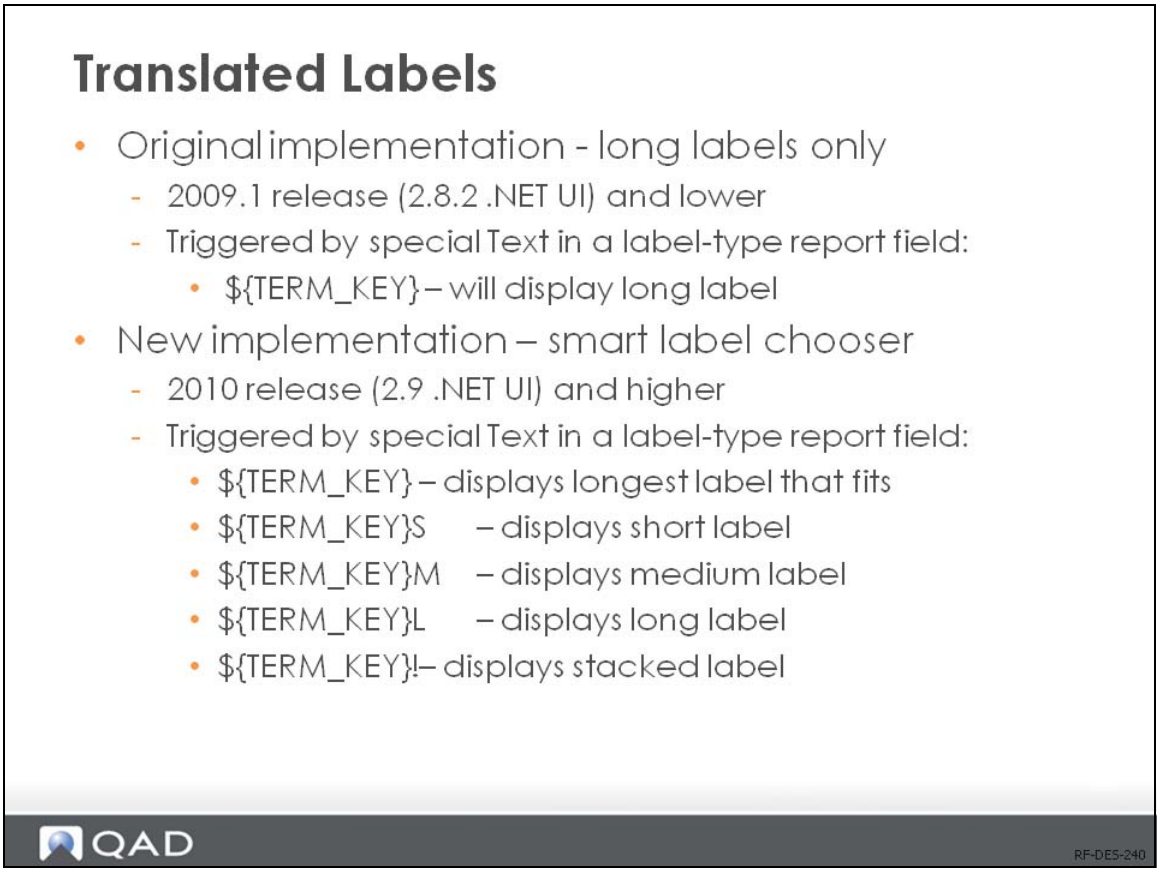
A twip (derived from TWentieth of an Imperial Point) is a typographical measurement, defined as 1/20 of a typographical point. One twip is 1/1440 inch or 17.639 µm when derived from the PostScript point at 72 to the inch, and 1/1445.4 inch or 17.573 µm based on the printer's point at 72.27 to the inch.

Using Properties Window



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

Translated Labels



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

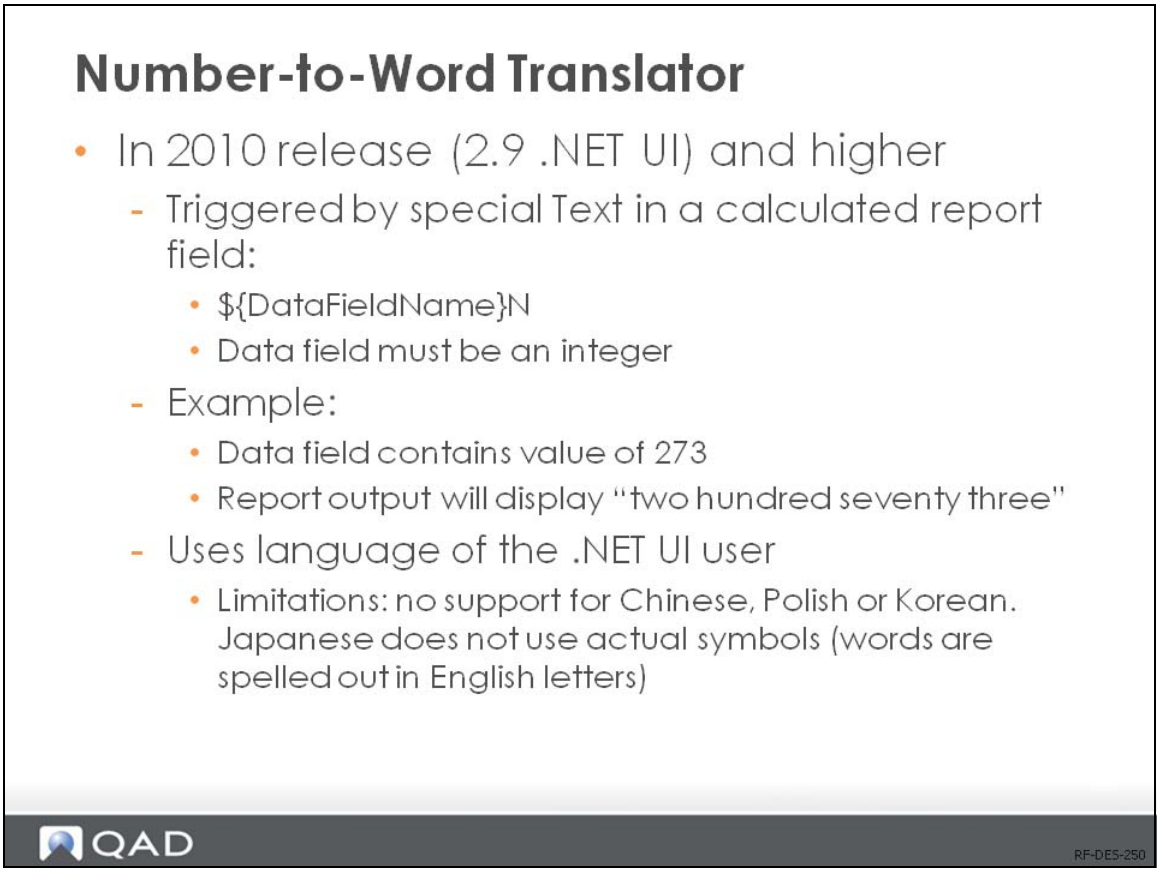
Note that in the case of stacked labels, the height of the field must be large enough to fit all the levels in the stacked label or the display of the label will be truncated.

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.

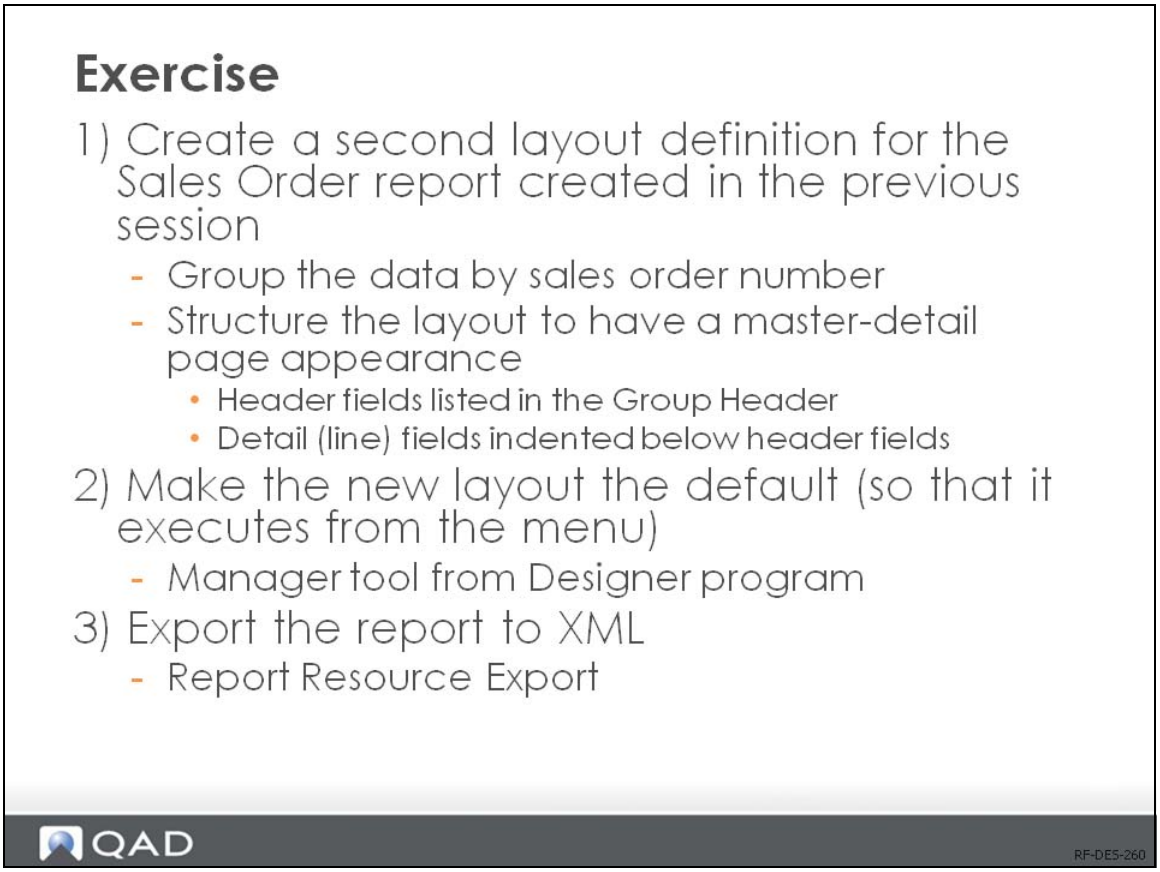
For Calculated fields, you can also specify to have integers converted into the words for the number. For instance, you can have “12” displayed as “twelve.” The format is as follows:

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

Number-to-Word Translator



Exercise



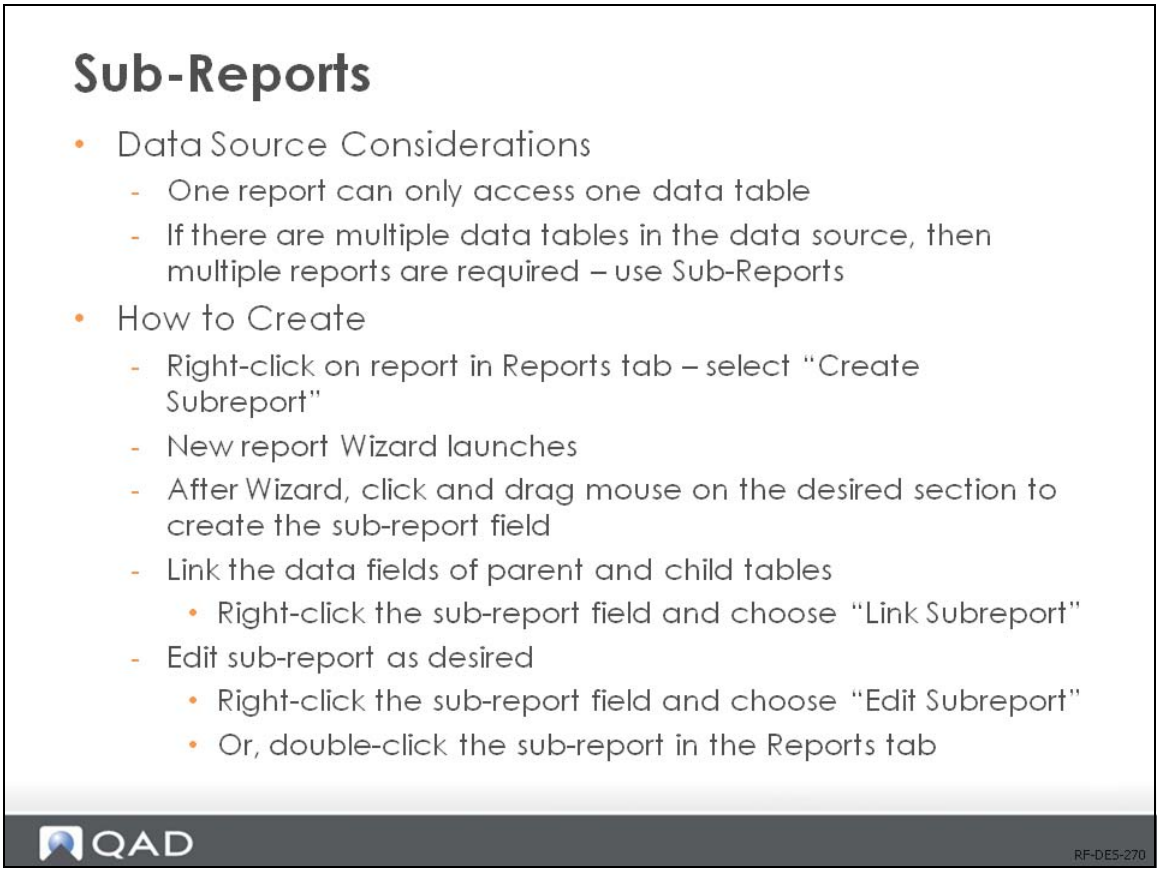
Exercises include the following:

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.

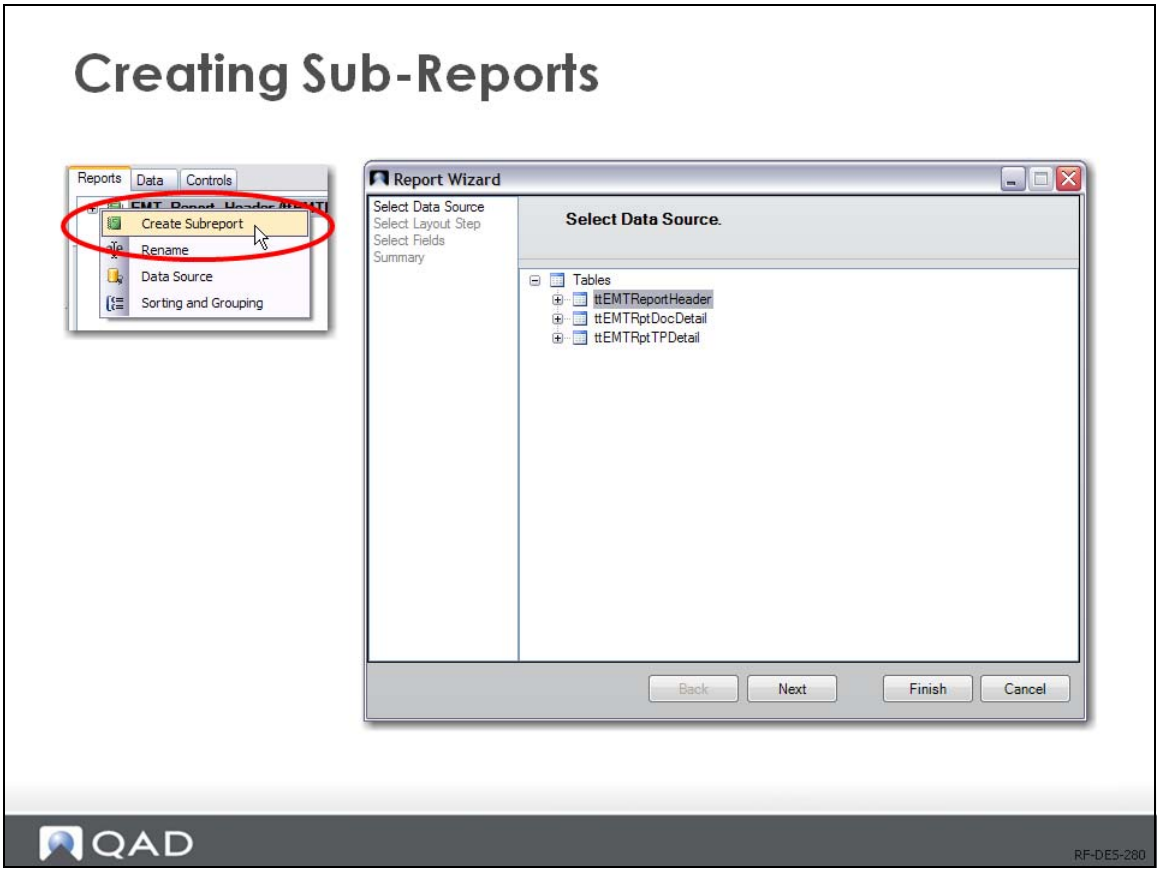
Sub-Reports



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.

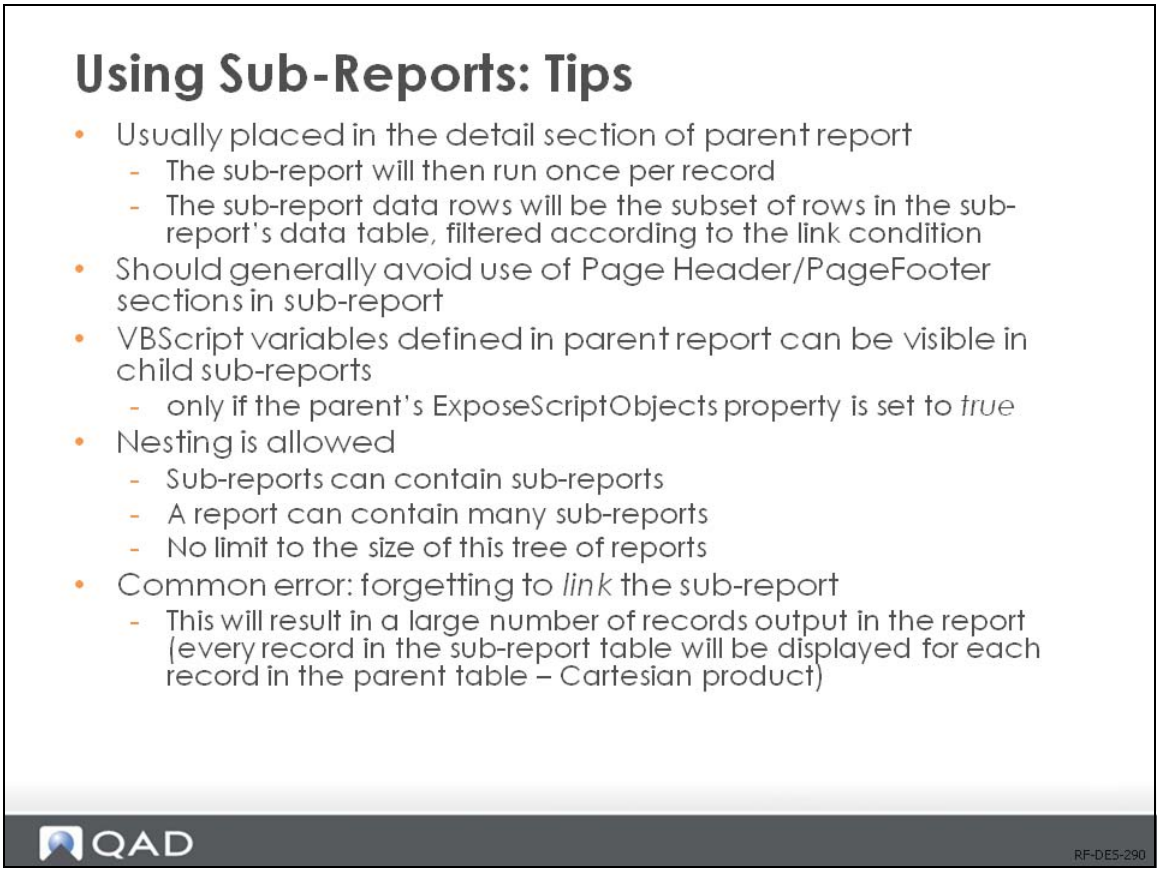
For example, the main report contains product categories and the subreport in the Detail section contains product details for the current category.

Creating Sub-Reports

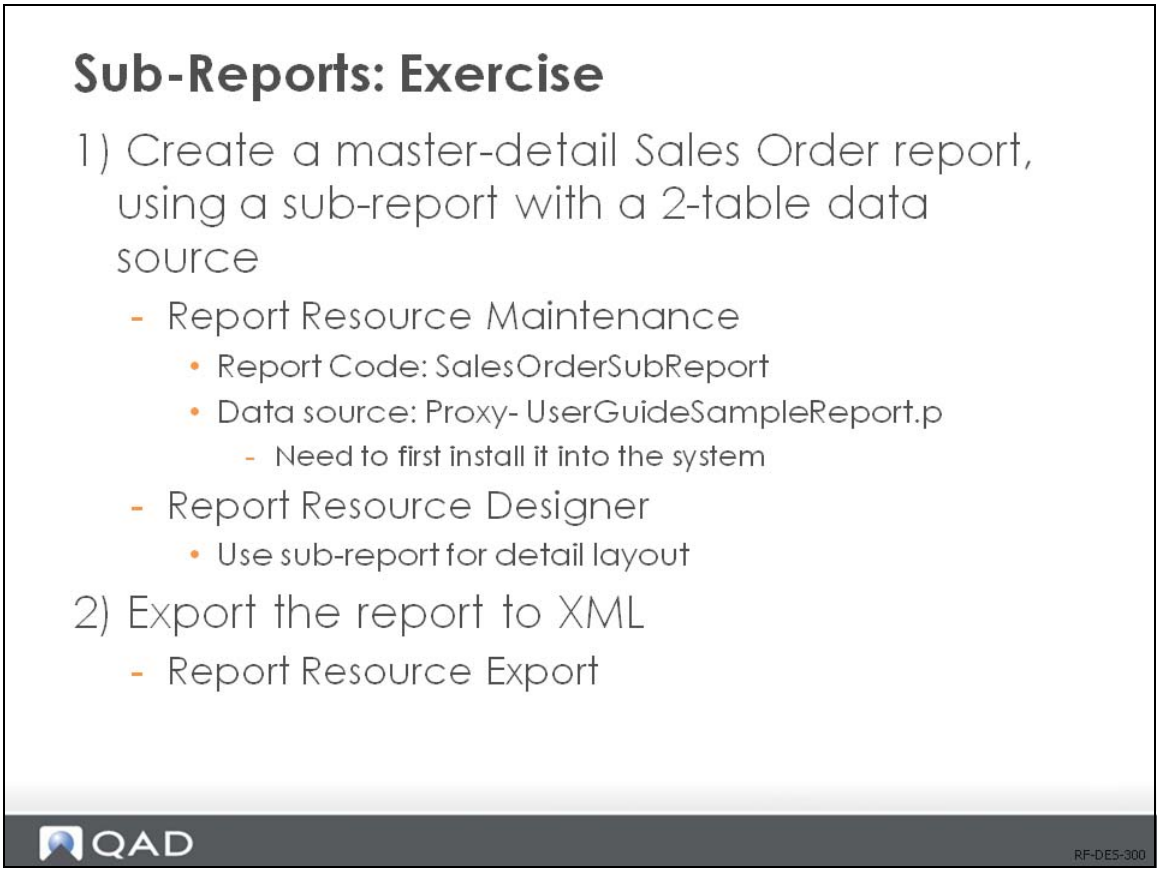


In the Reports tab, choose Create Subreport and then from the Report Wizard, select the data source.

Using Sub-Reports: Tips



Sub-Reports: Exercise

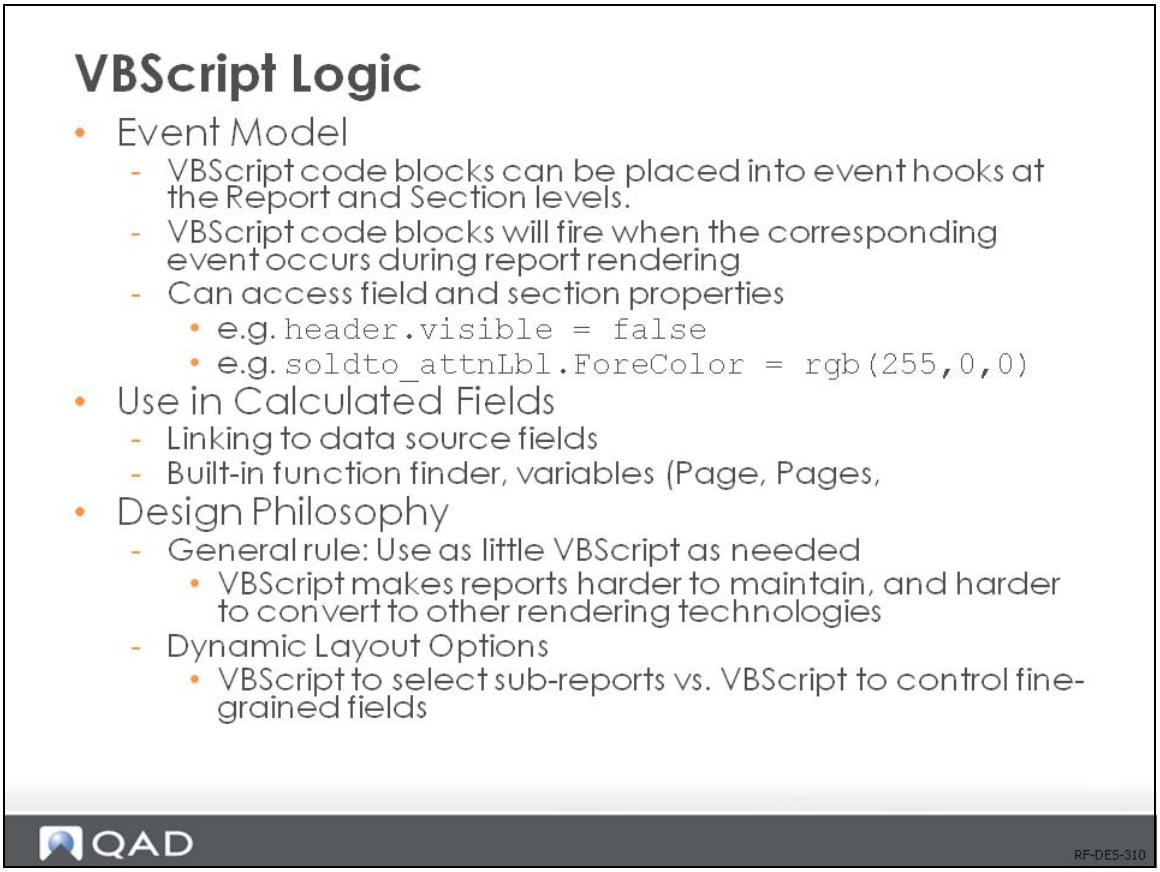


Exercises include the following:

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



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.

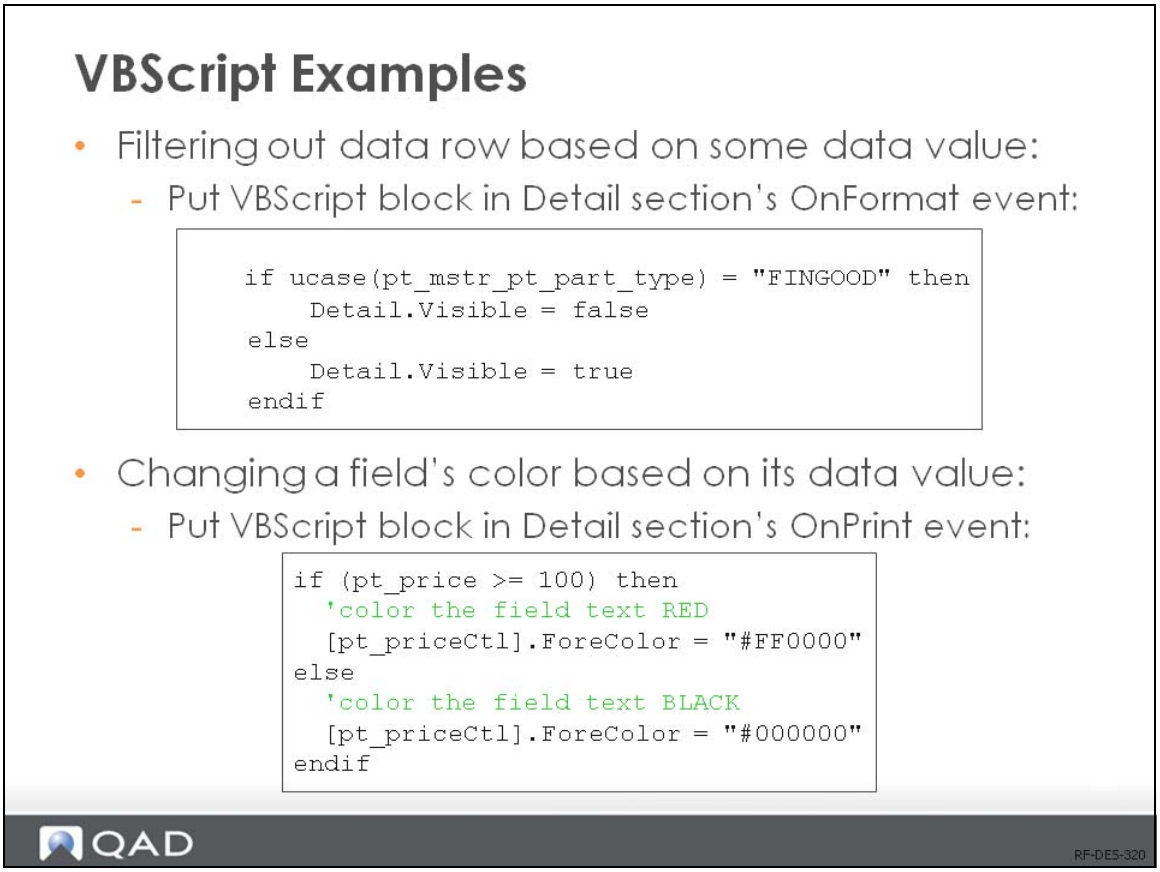
Design Philosophy for using VBScript:

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.

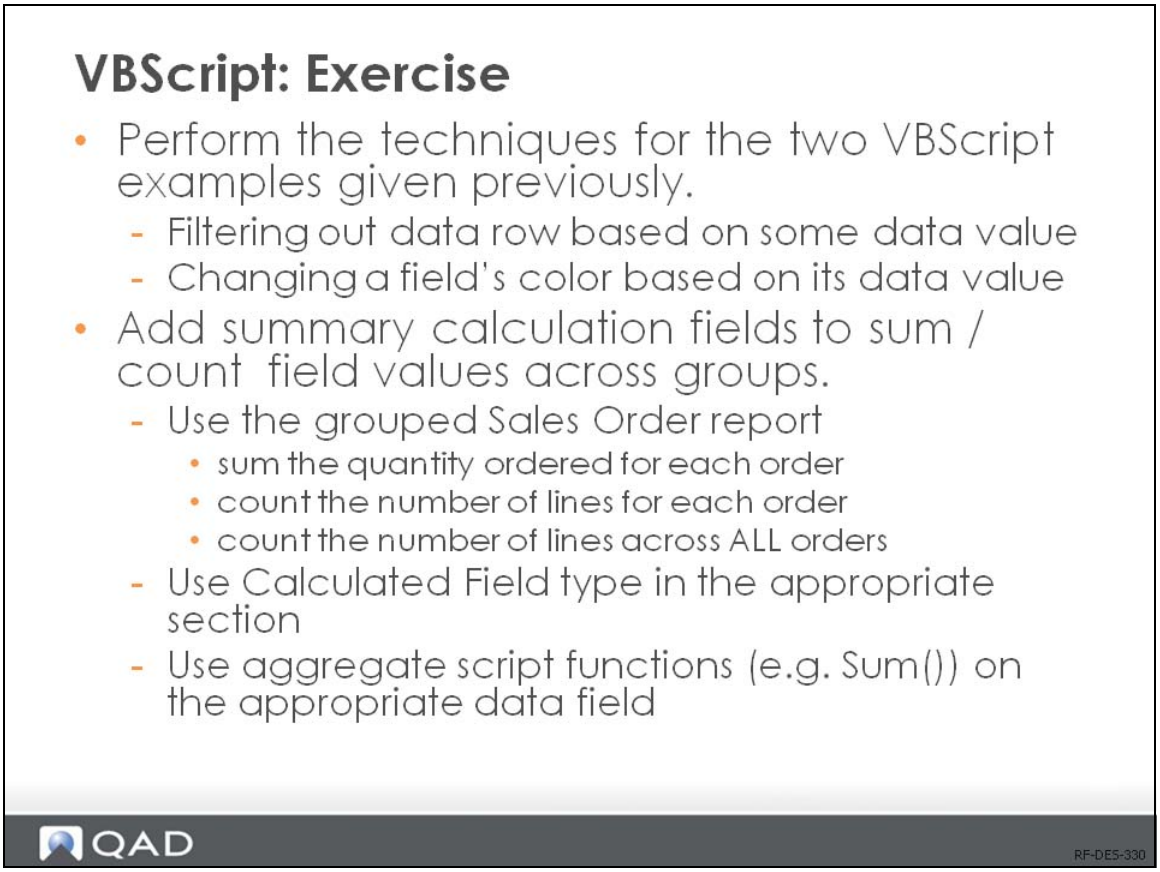
VBScript is often necessary for achieving certain dynamic layout goals, for example changing a file's color or visibility based on its data value.

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 tow 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 Exercise

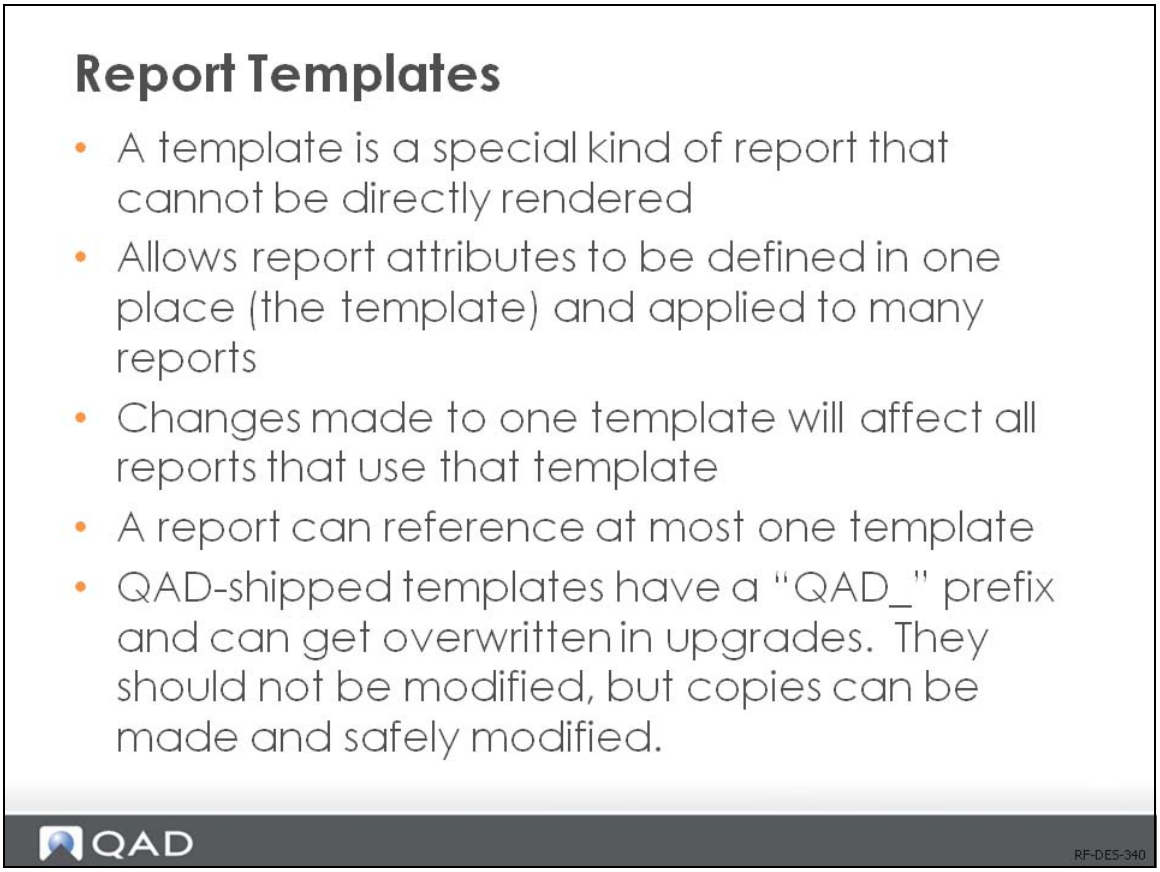


Exercises include the following:

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



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.

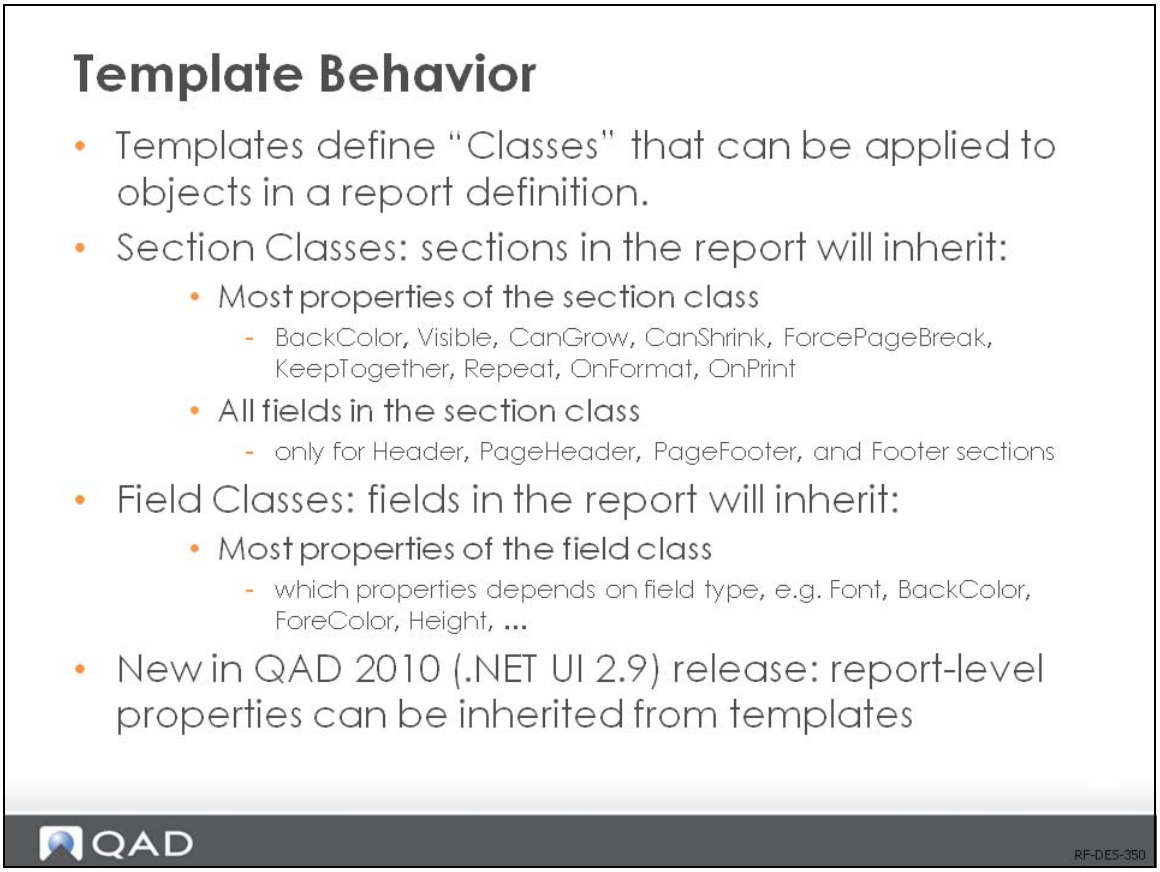
Templates can be edited in the Template Designer, which is very similar to the Report Designer program.

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.

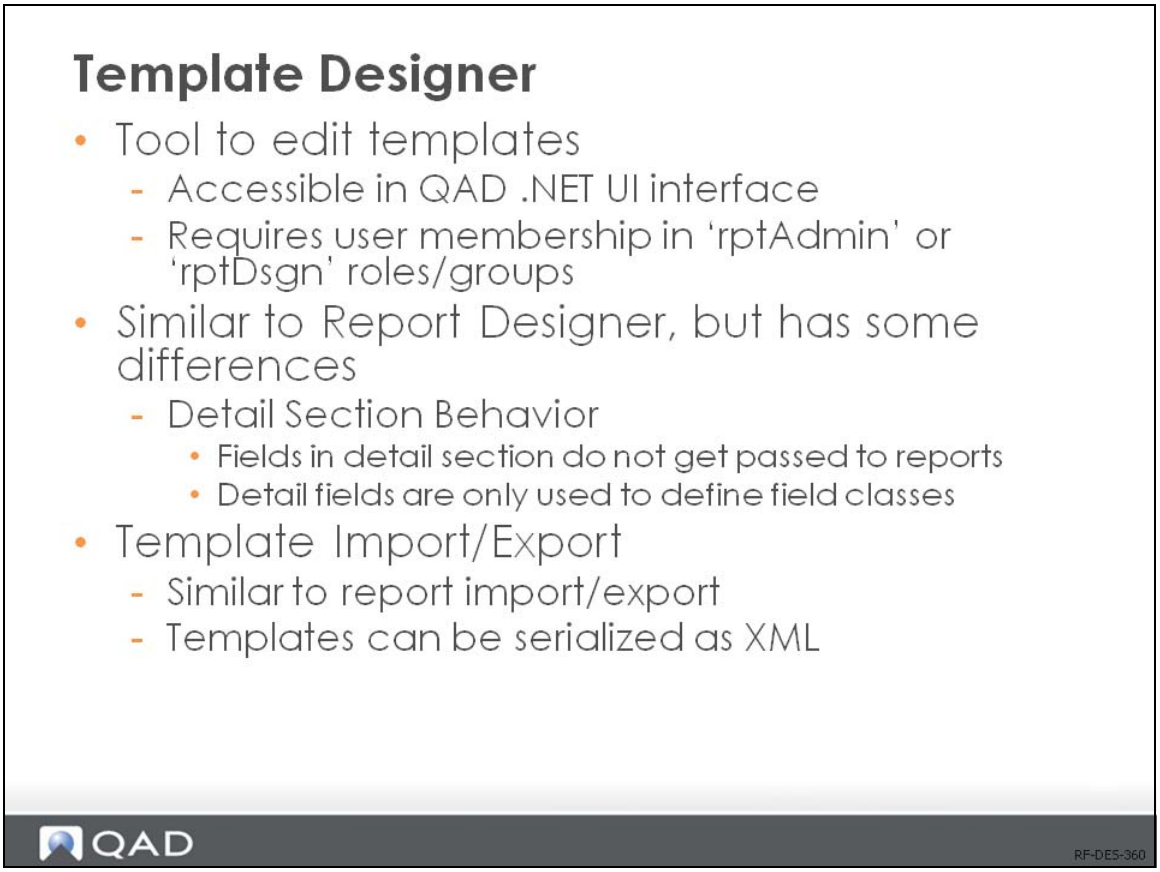
A field defined in the report template represents a field class identified by a unique class name. When the field class is applied to a field in a report definition, most of its properties are carried over to the field so that the field takes on the same formatting and layout.

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



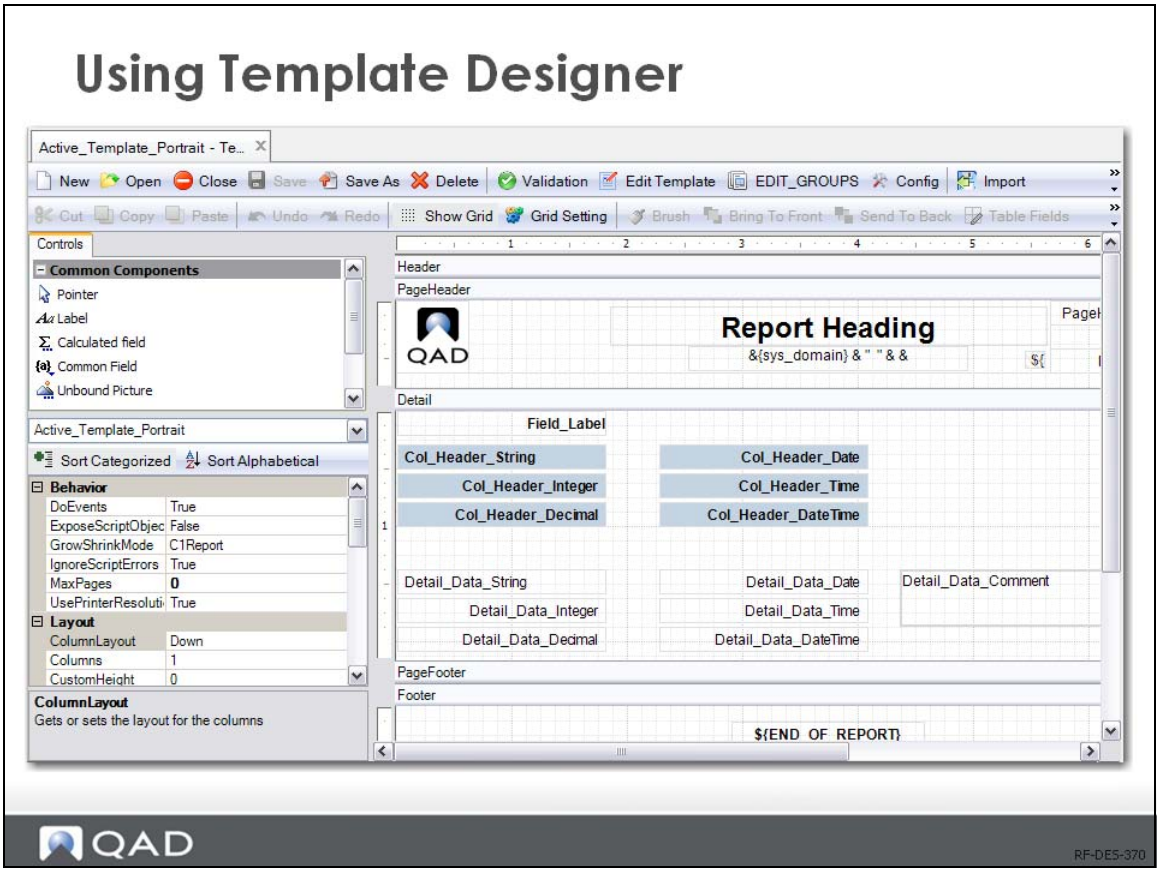
Template Designer



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.

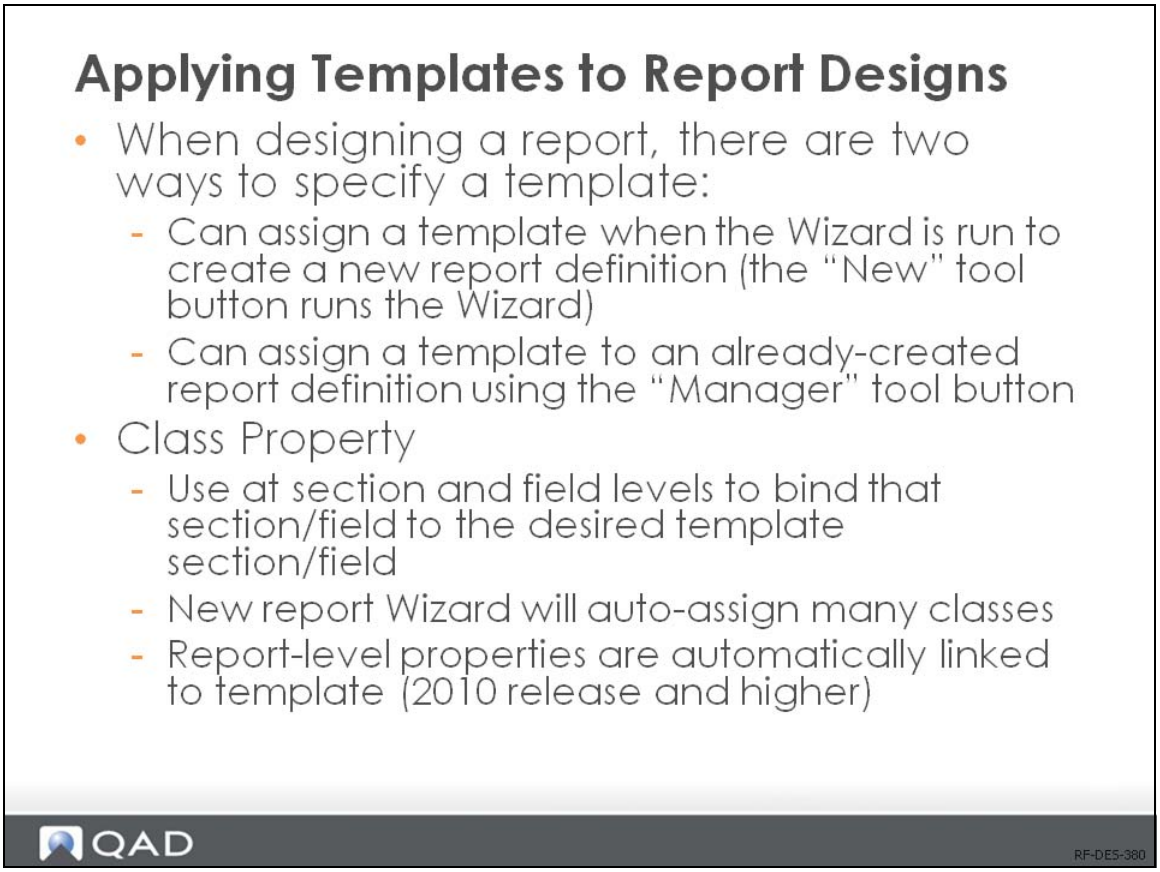
   b Under the Section Configuration tab, specify a class name for each section type.

   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.

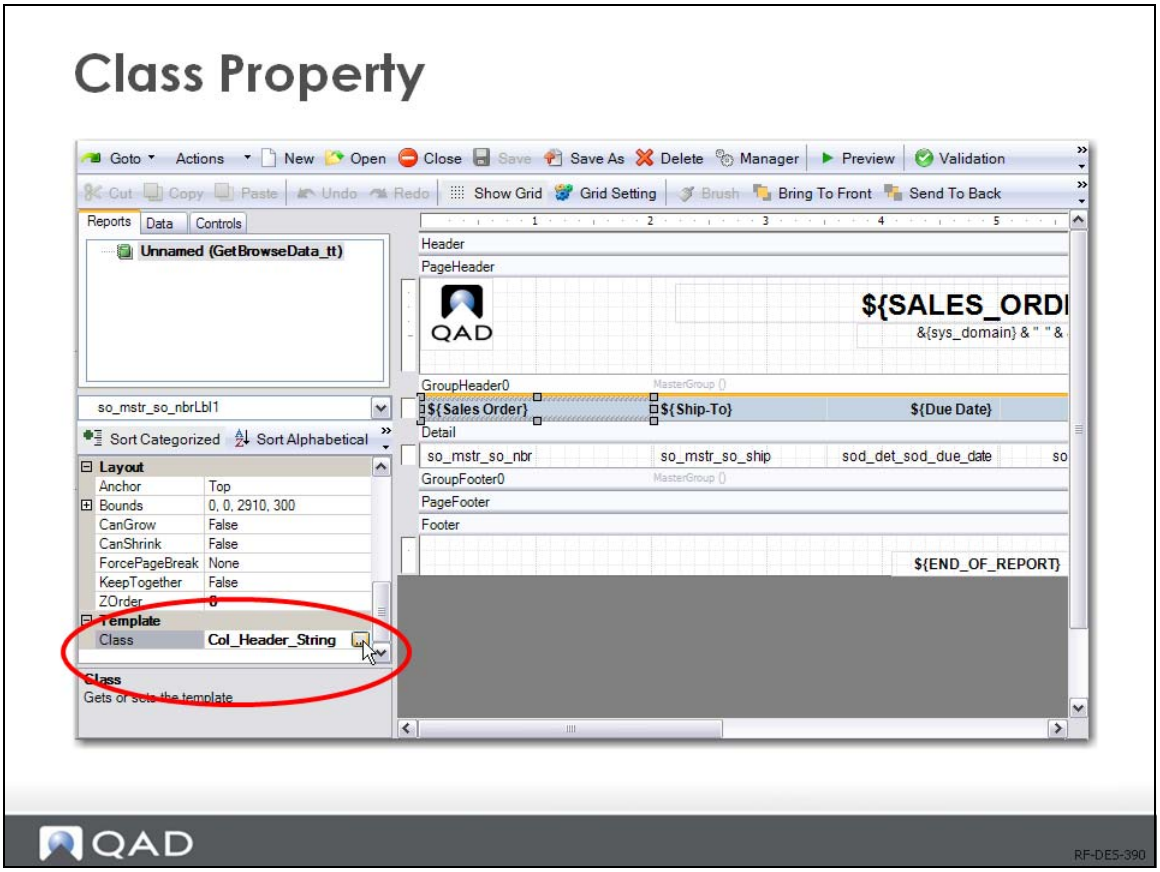
8 Back in the Template Designer main screen, click the Save button on the toolbar to save the template.

Applying Templates to Report Designs

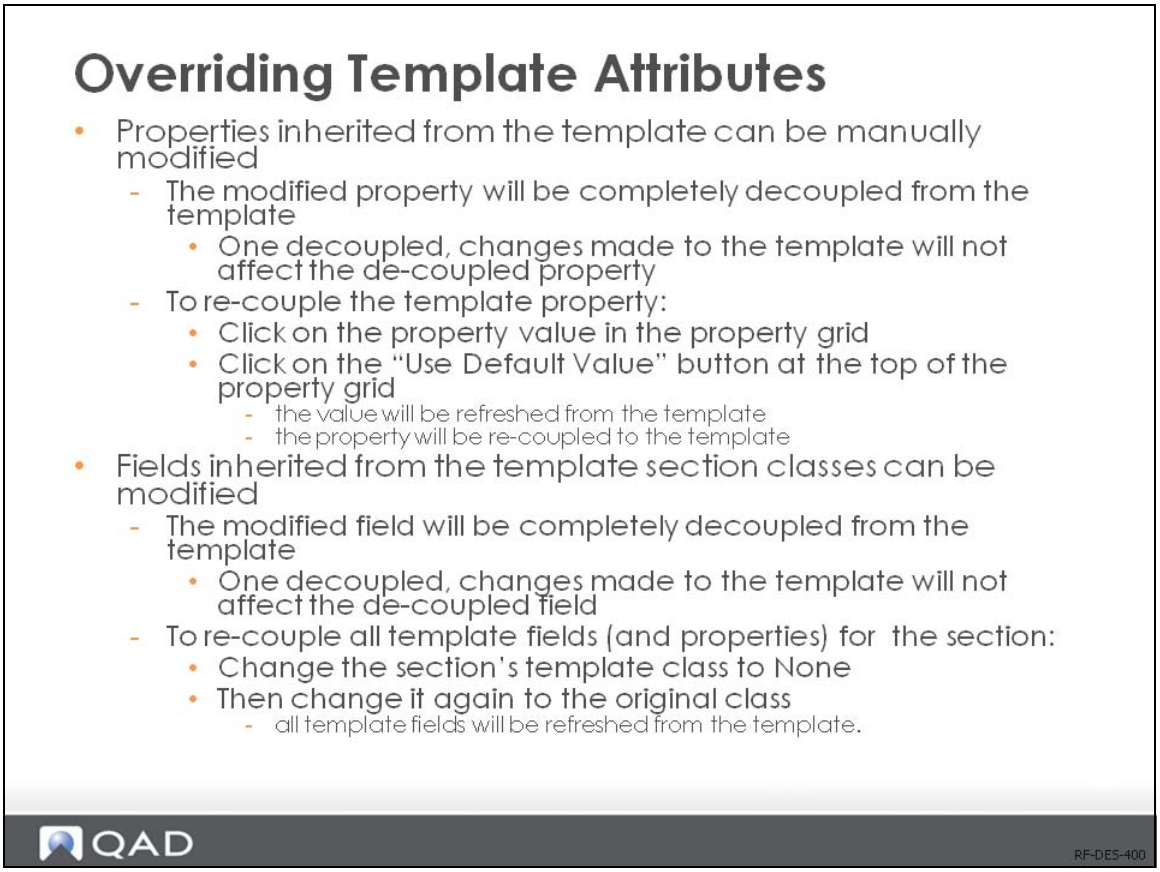


After you design a report template, you can apply it to multiple report definitions to enforce a consistent look and feel across all these reports.

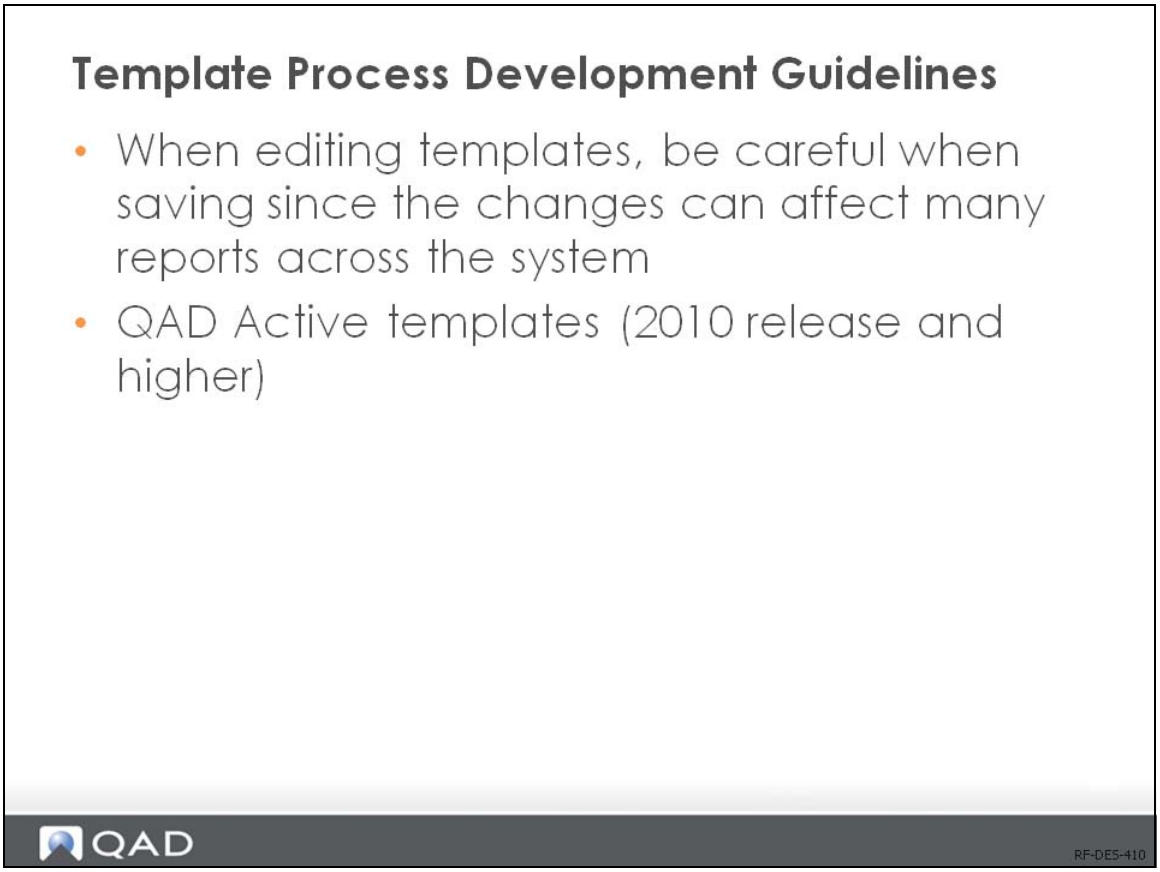
Class Property



Overriding Template Attributes



Template Process Development Guidelines



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.

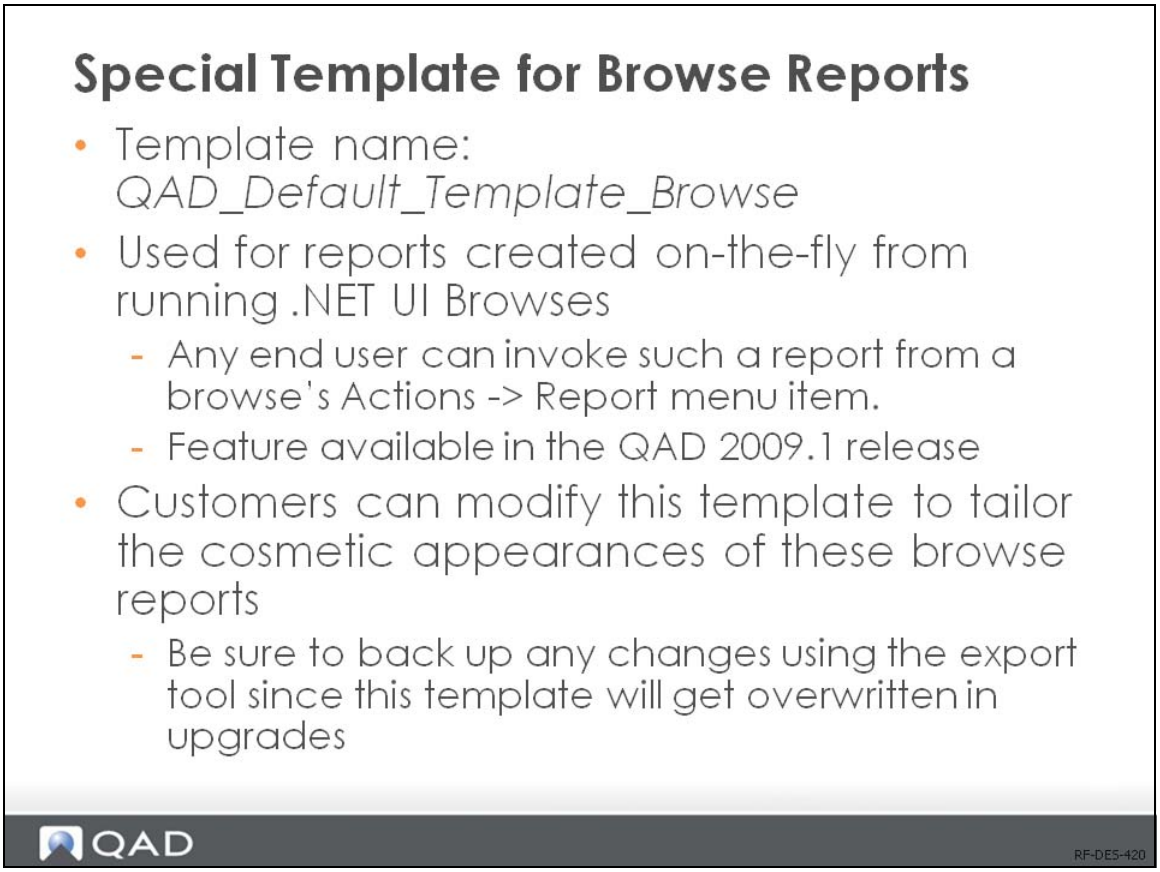
Additionally, four QAD standard templates are included:

- 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.

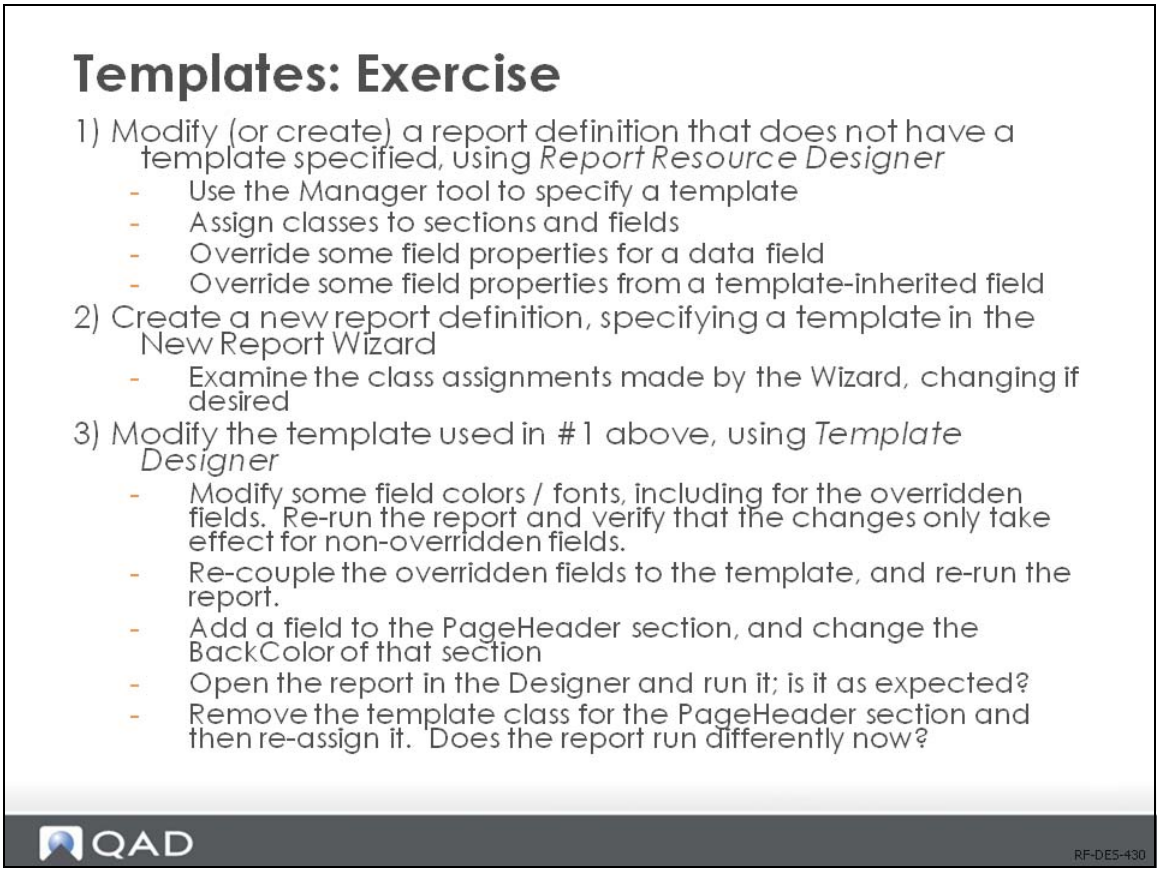
As shipped, the contents of Active_Template_Landscape are identical to QAD_Standard_Template_Letter_Landscape, and the contents of Active_Template_Portrait are identical to QAD_Standard_Template_Letter_Portrait. If template customizations are desired, they can be done in the Active templates (which the 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



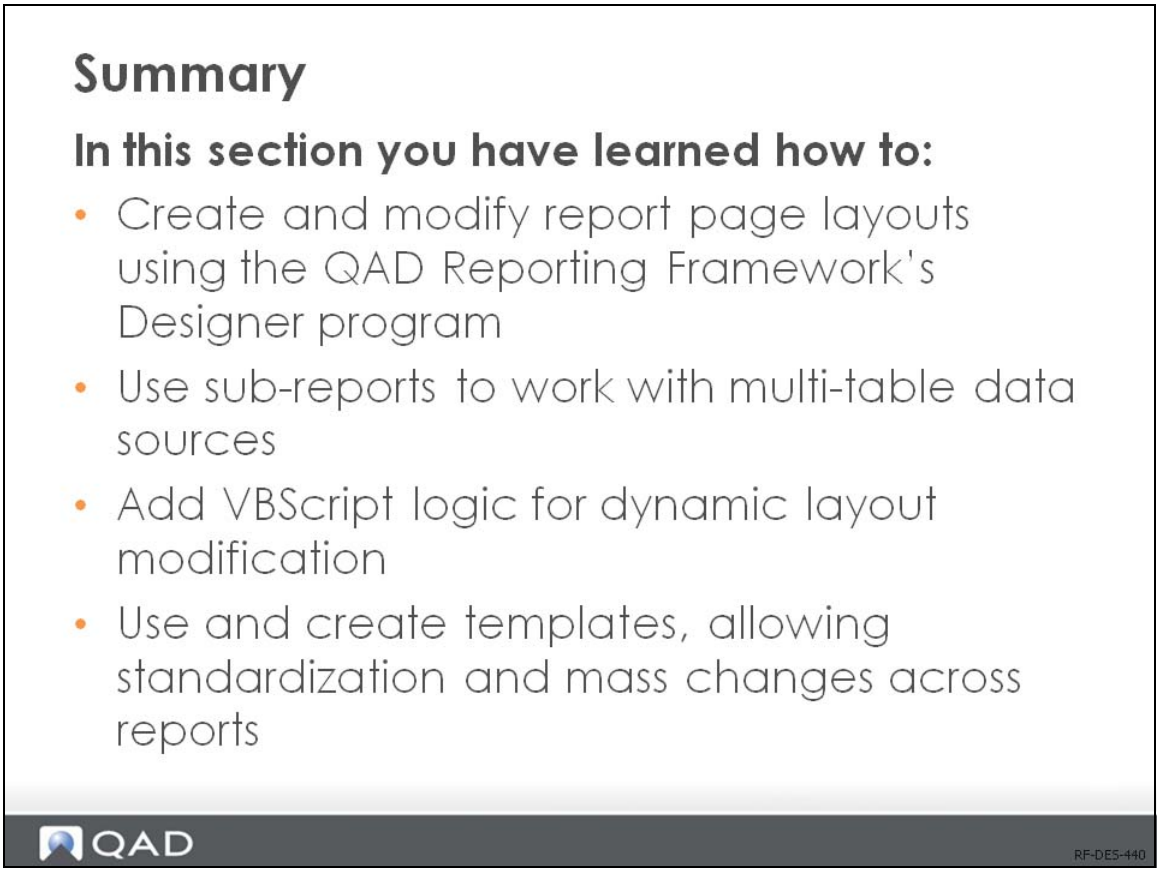
Templates: Exercise



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 step 1 above, using Template Designer

Summary



In this section you have learned how to:

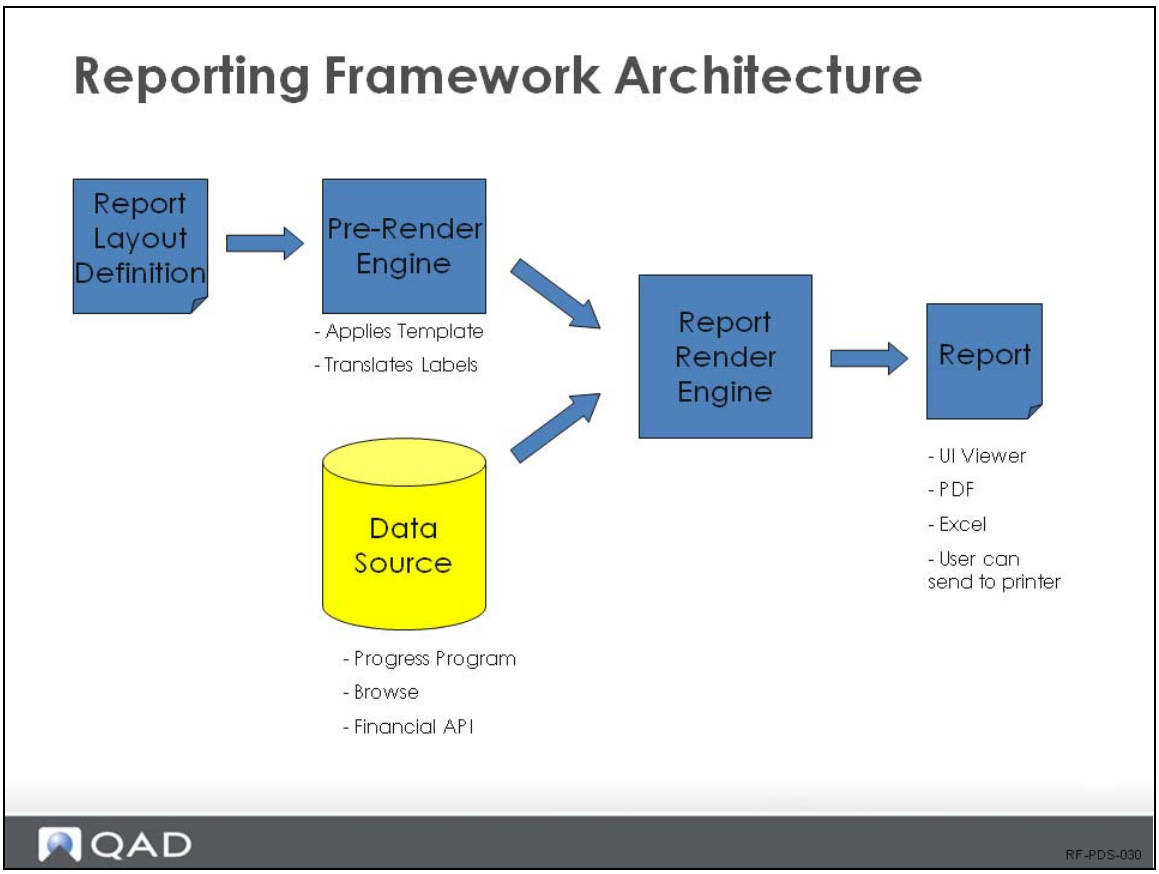
- Create and modify report page layouts using the QAD Reporting Framework’s Designer program
- Use sub-reports to work with multi-table data sources
- Add VBScript logic for dynamic layout modification
- Use and create templates, allowing standardization and mass changes across reports

Chapter 3

Progress Data Source Program

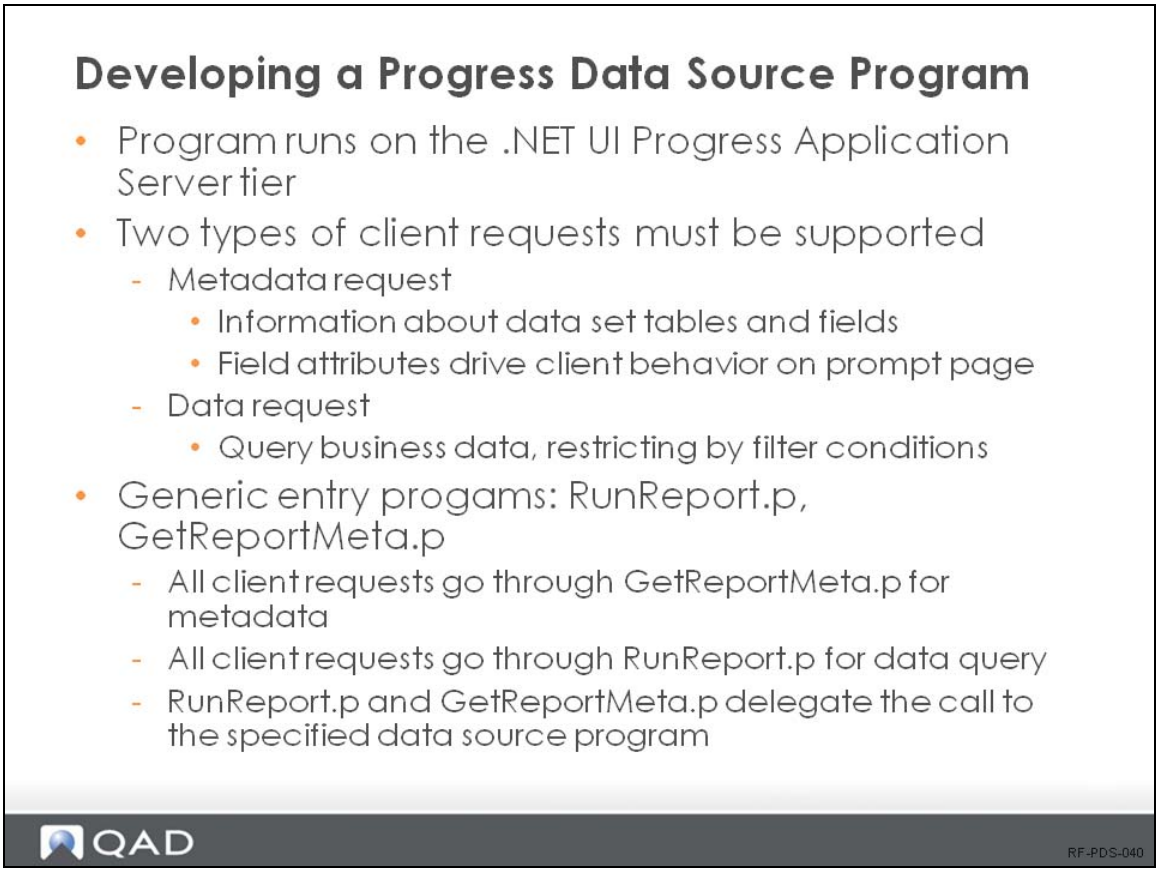
Session Objectives

Reporting Framework Architecture: Data Source

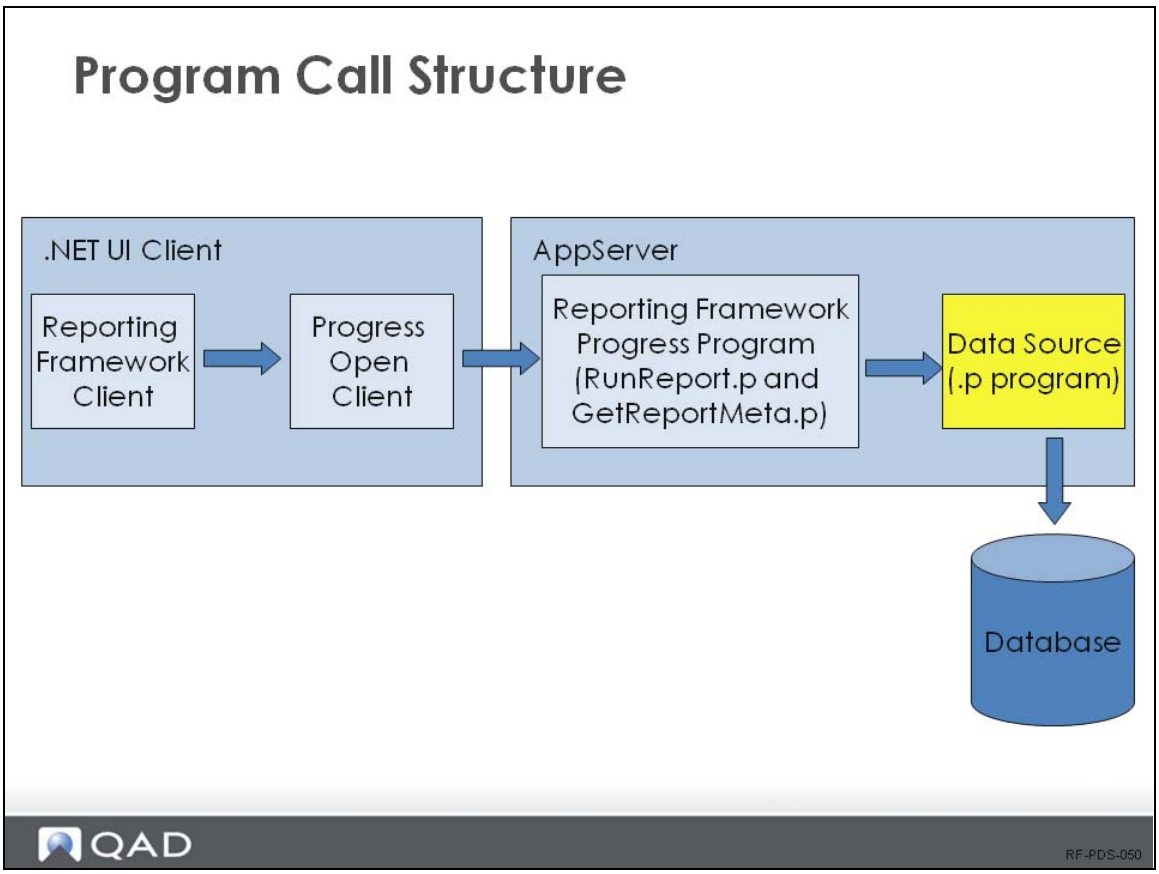


There are three types of data sources that a report can use: Browse, FinancialAPI, and Progress Program. Our focus in this session is on developing and deploying a Progress Program – a proxy data source program.

Developing a Progress Data Source Program

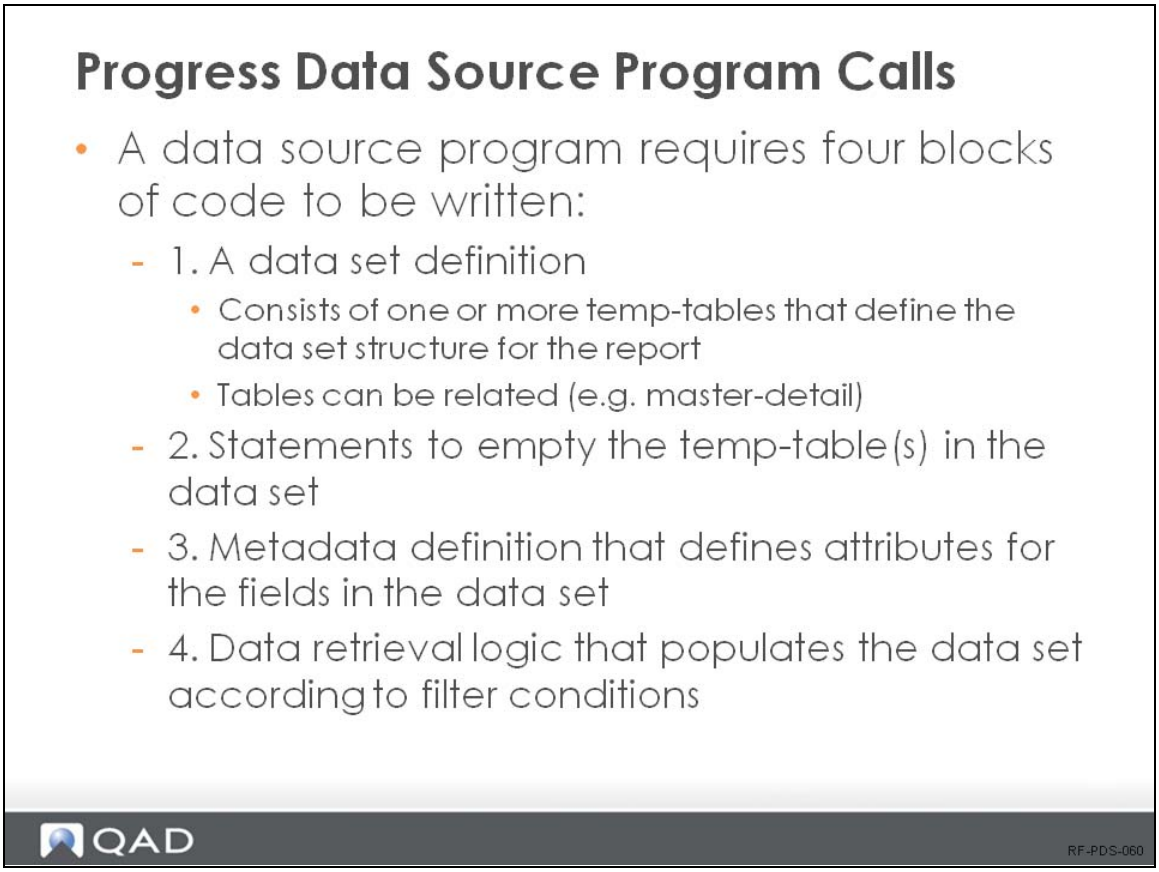


Program Call Structure



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

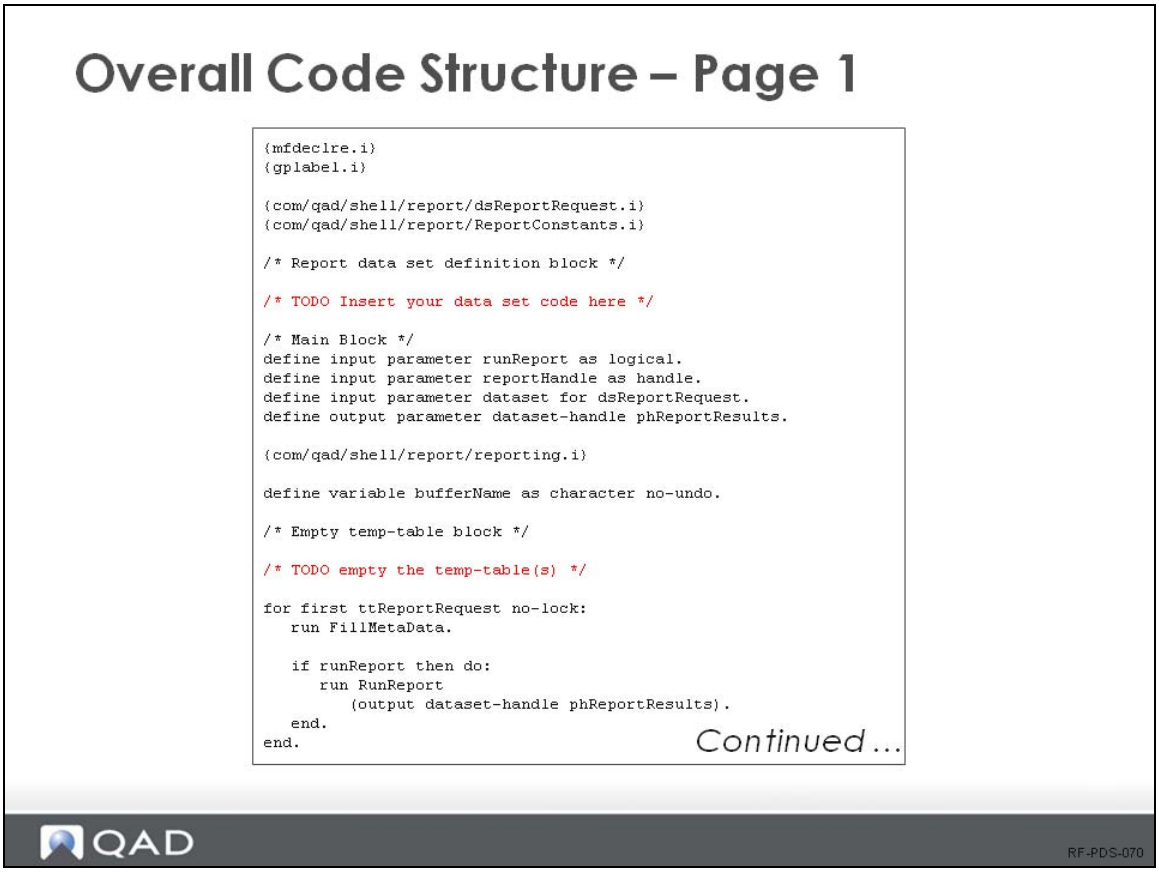
Progress Data Source Program Calls



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



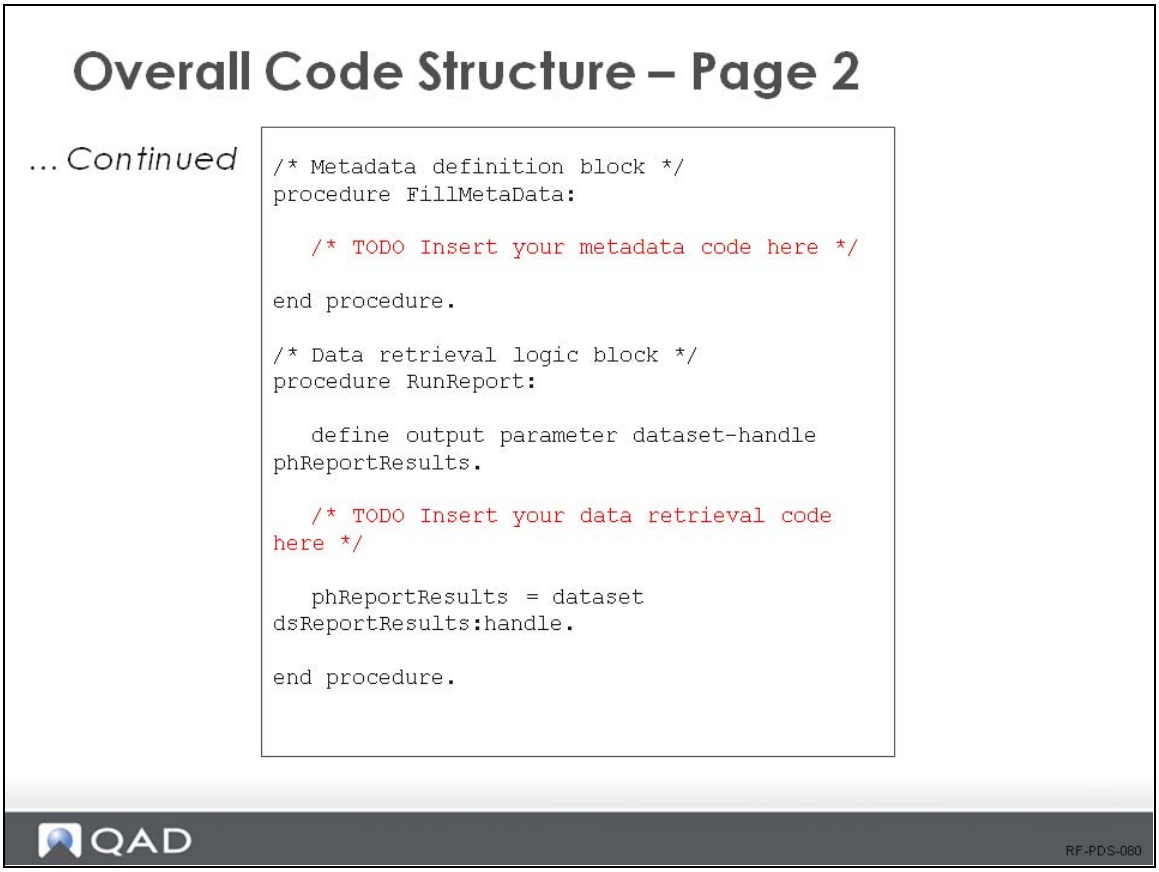
In order for the program to be invoked properly by RunReport.p, it must have a specific structure. The following sample code illustrates this structure, showing where the four blocks of code should be inserted:

```
{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:
```

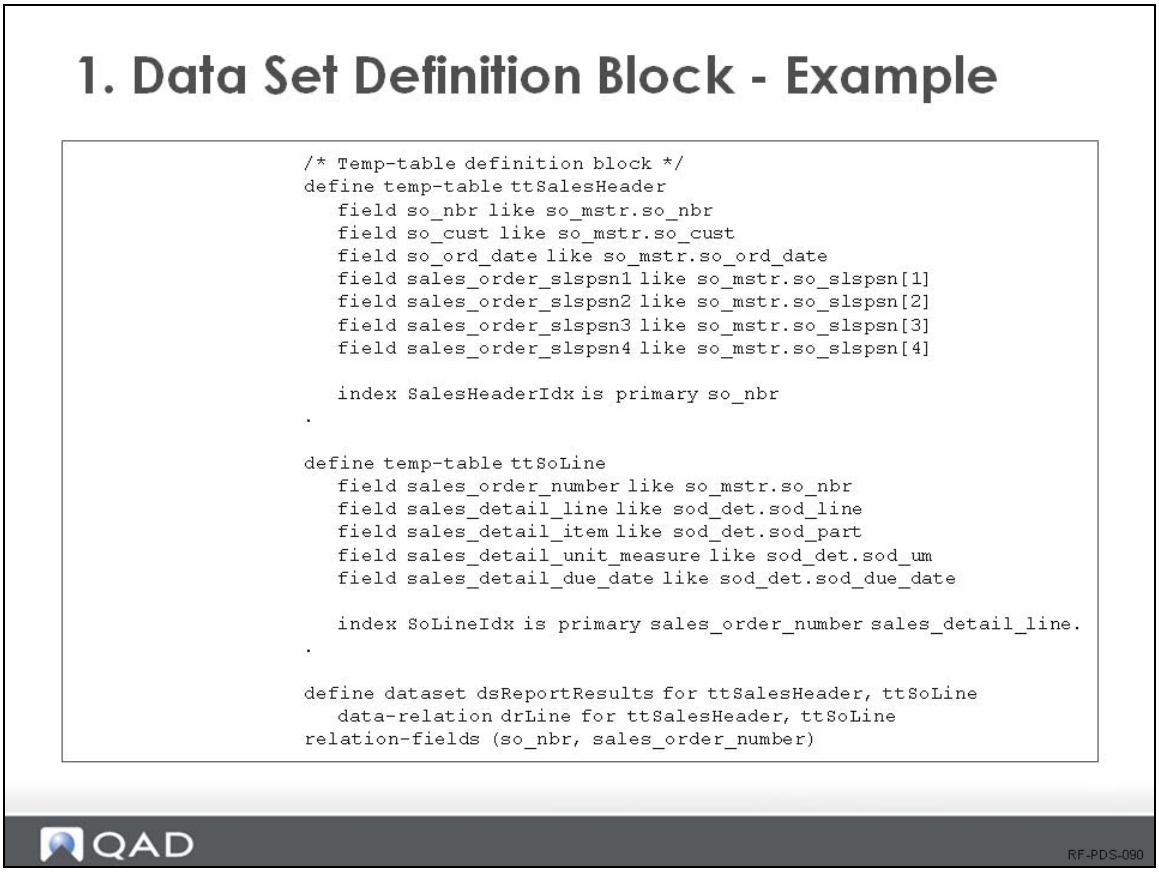
```
      run RunReport
        (output dataset-handle phReportResults).
    end.

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



1. Data Set Definition Block - Example

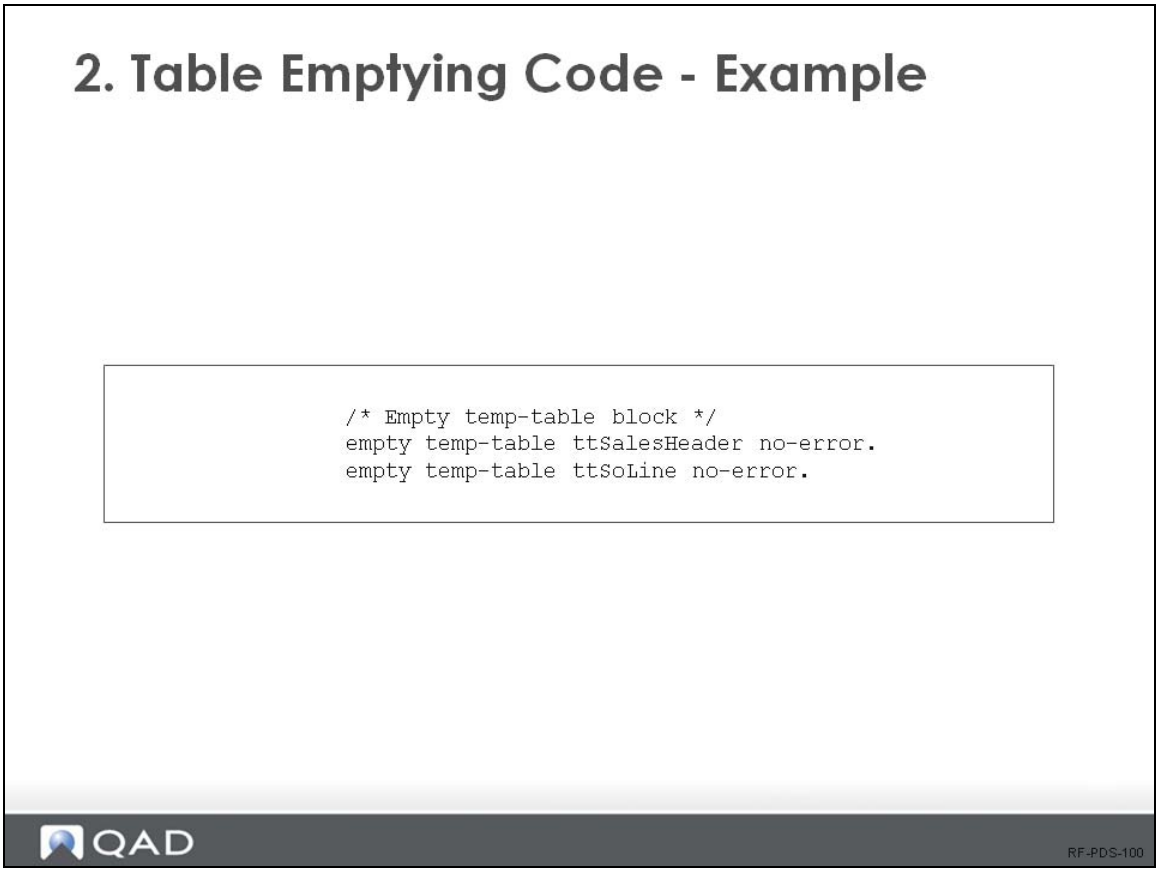


Note If you want to use some fields in the temp-table as search fields, you must use the same field names in the temp-tables as those in the database.

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

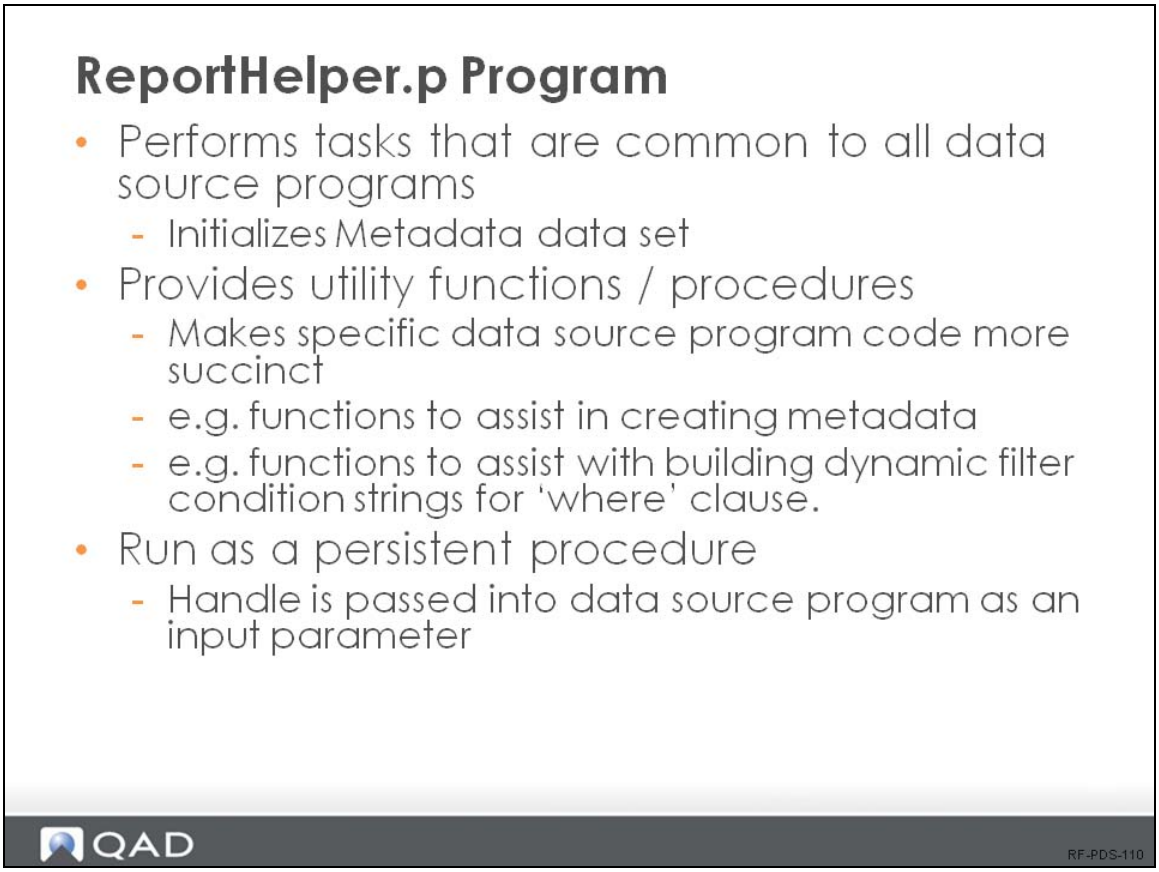
Note There must be a dataset named dsReportResults defined for temp-tables for the program to work, even if they have no relations.

2. Table Emptying Code - Example

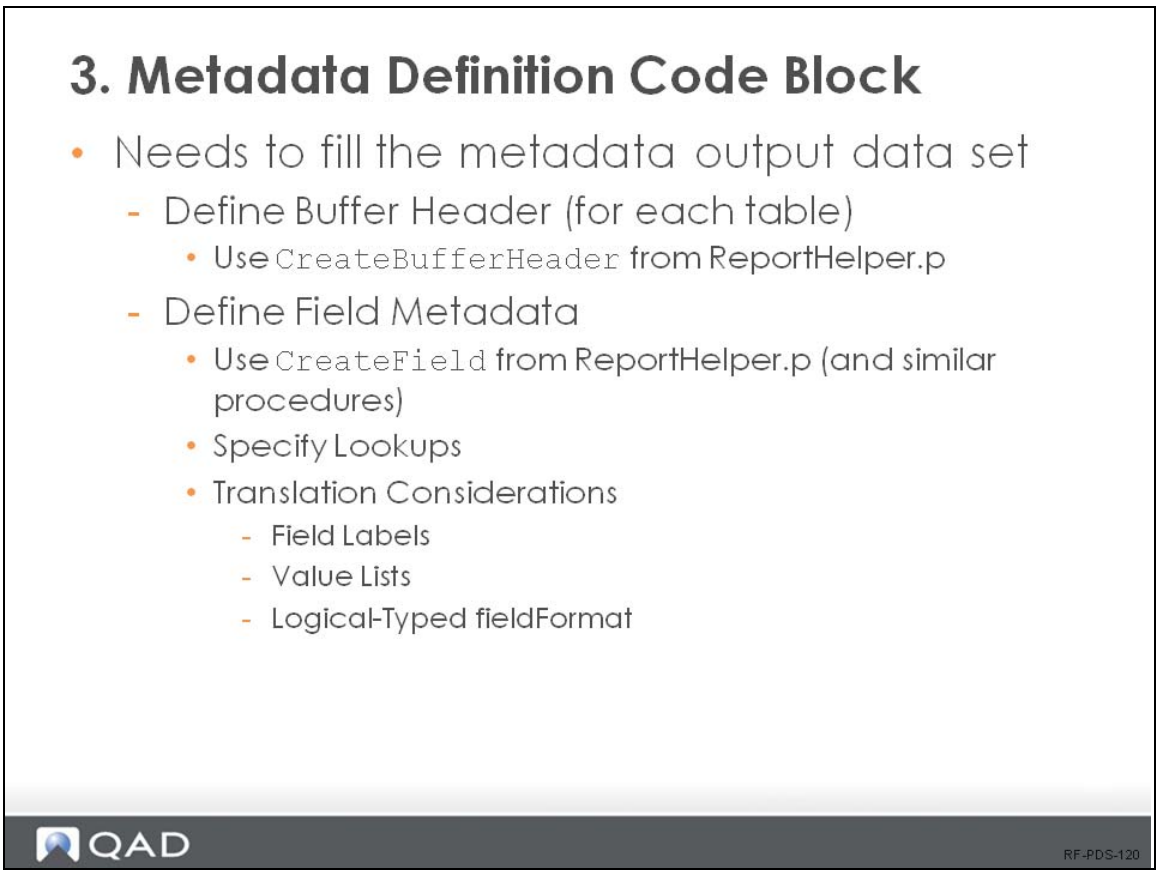


Each temp-table in the report data set should be emptied before the main program logic is executed. This is necessary since temp-table contents could still be cached on the Application Server from previous client requests.

ReportHelper.p Program



3. Metadata Definition Code Block



Metadata: Using CreateBufferHeader

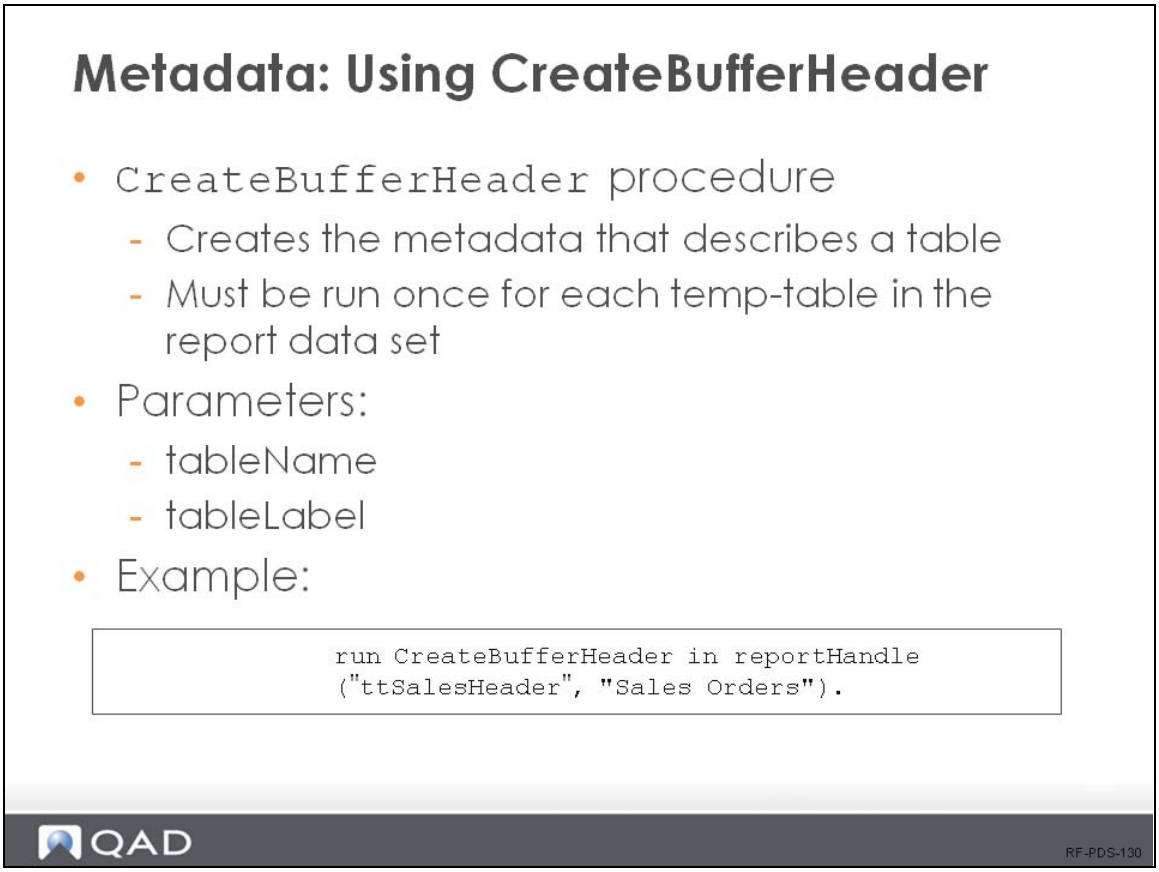
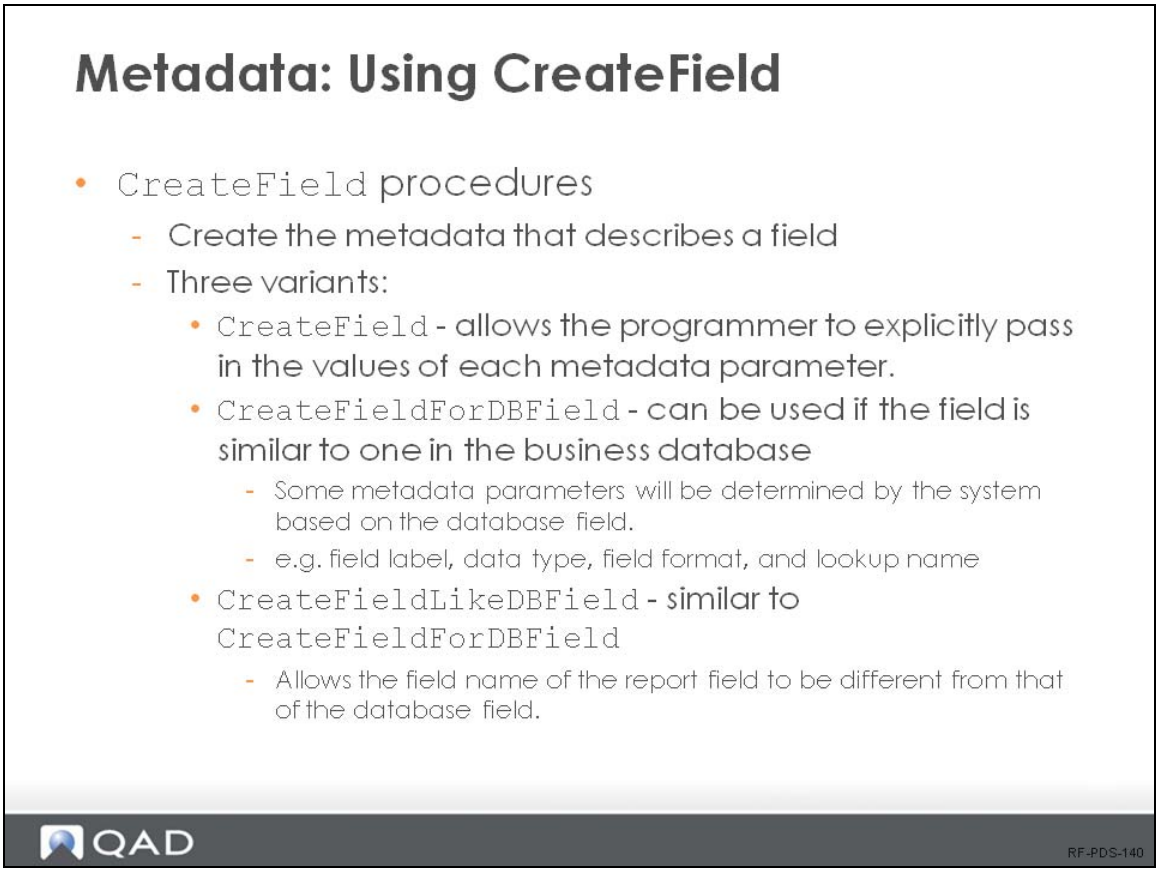


Table 3.1
CreateBufferHeader Parameters

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



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

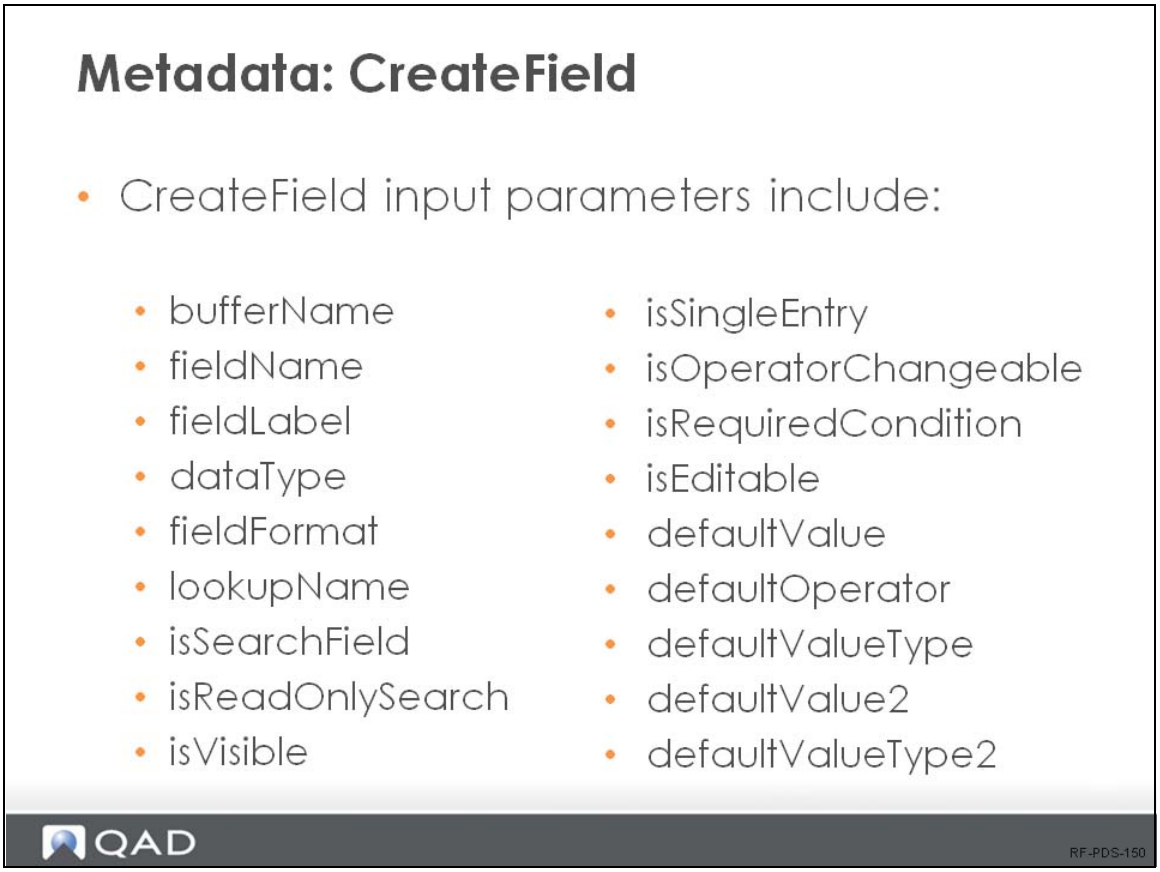


Table 3.2
CreateField Parameters

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: `<trueLabel>/<falseLabel>`. For example, `"Debit/Credit"`. Note: in a multi-language system, these values should not be hard coded but instead dynamically translated using the getLabel 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

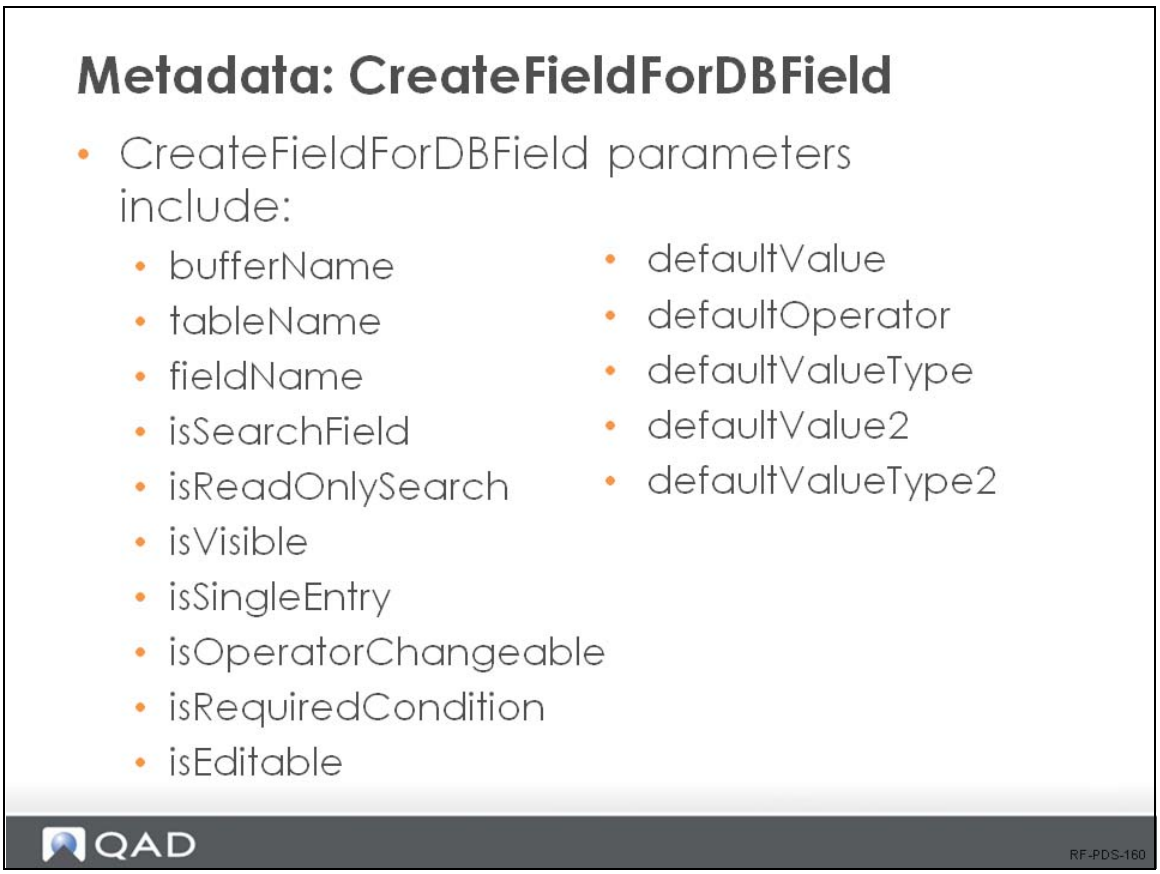


Table 3.3
CreateFieldForDBField Parameters

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

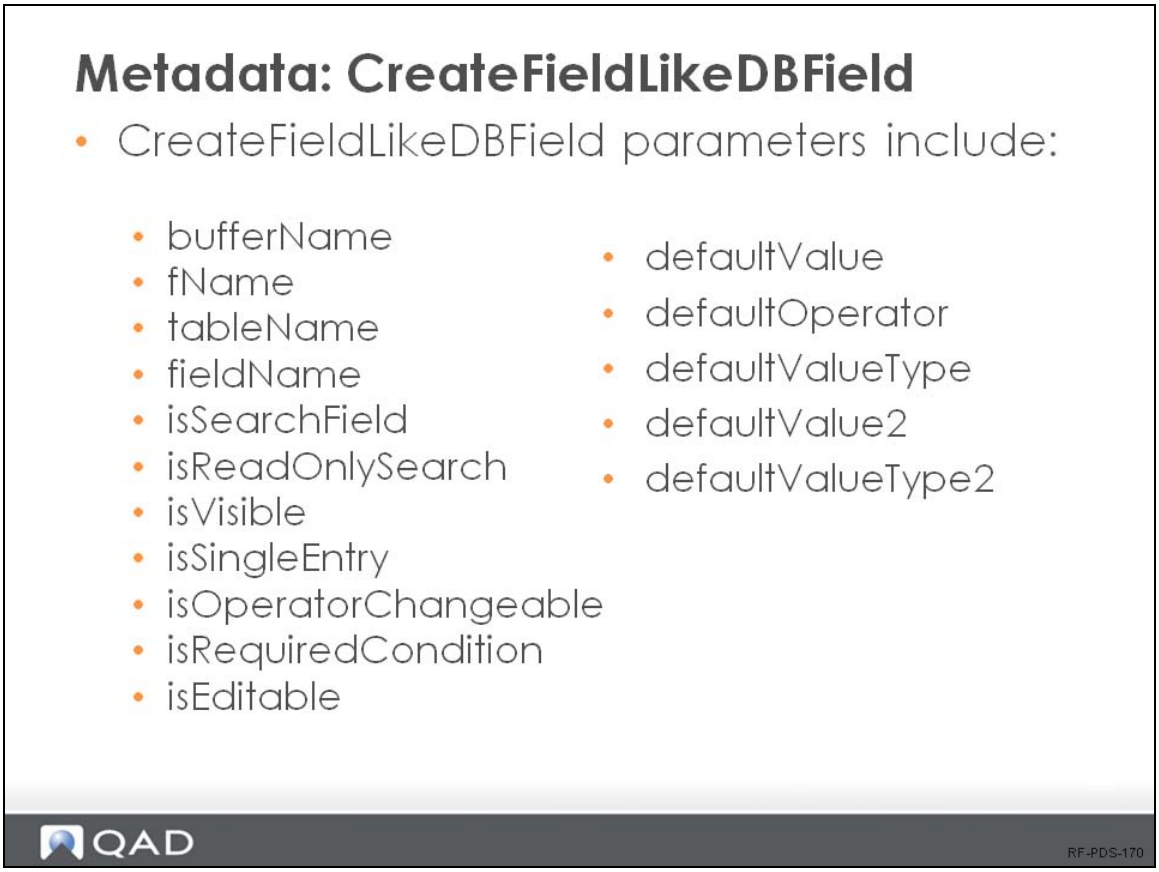
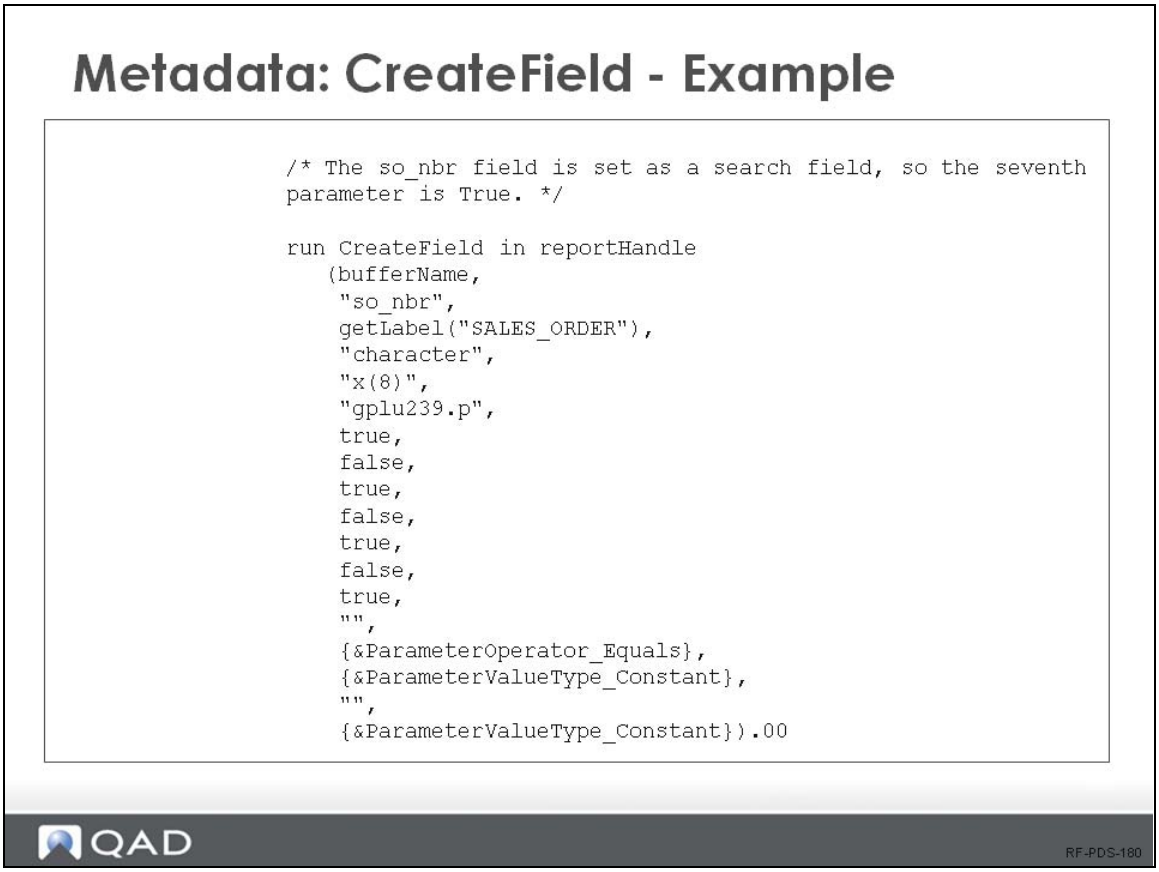


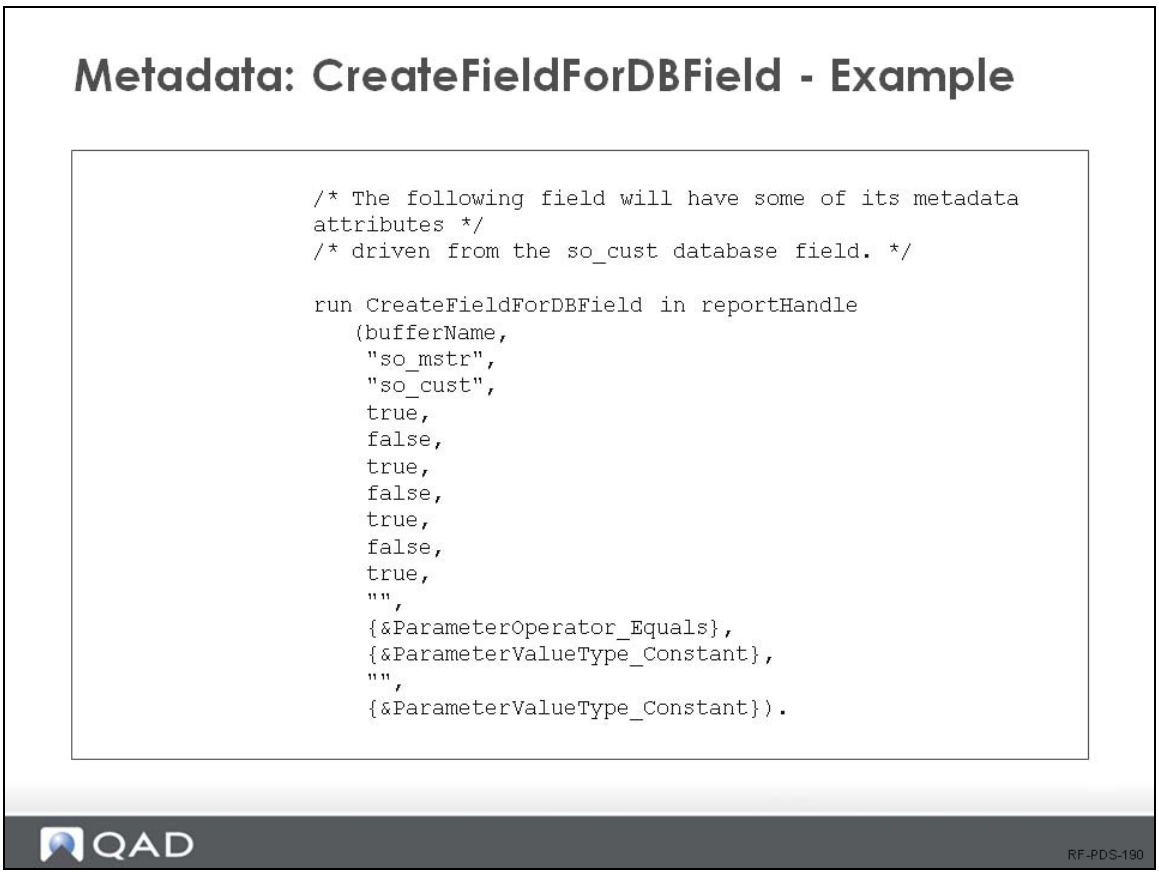
Table 3.4
CreateFieldLikeDBField Parameters

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

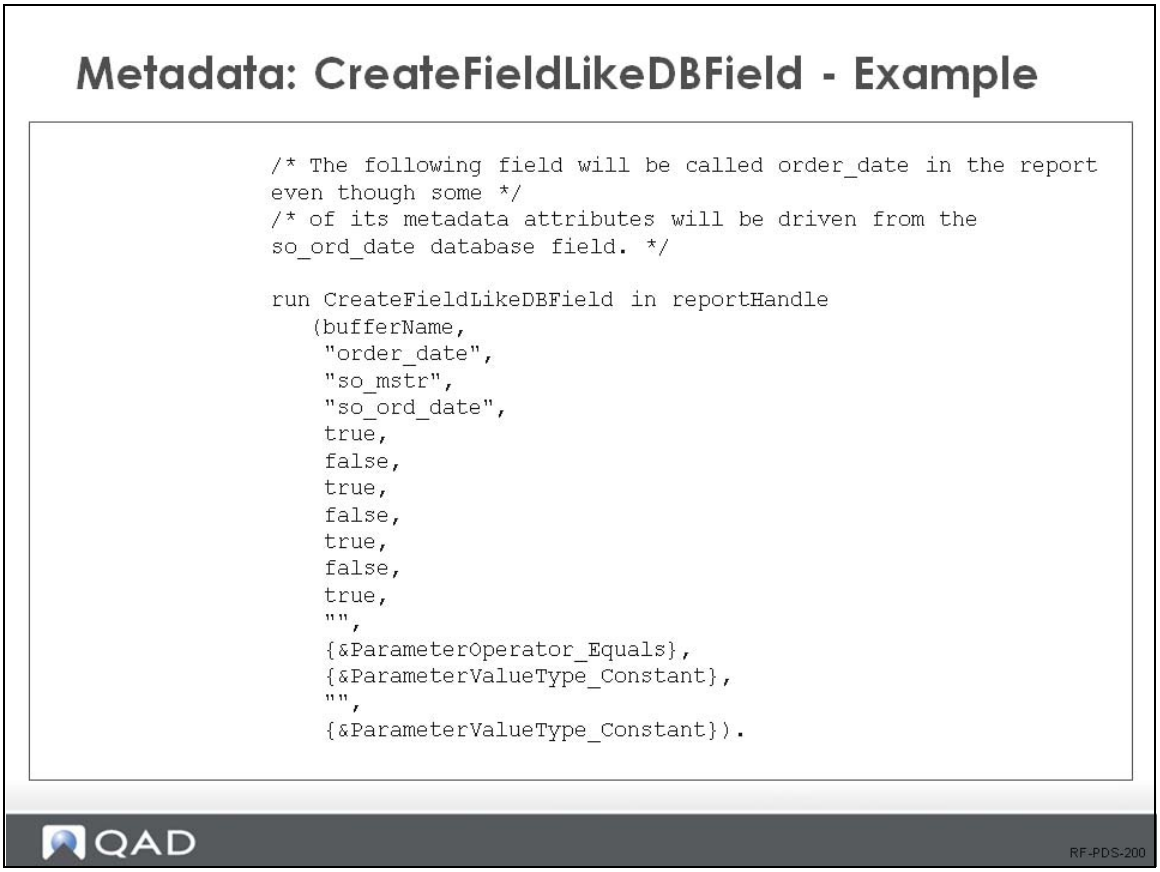
Metadata: CreateField - Example



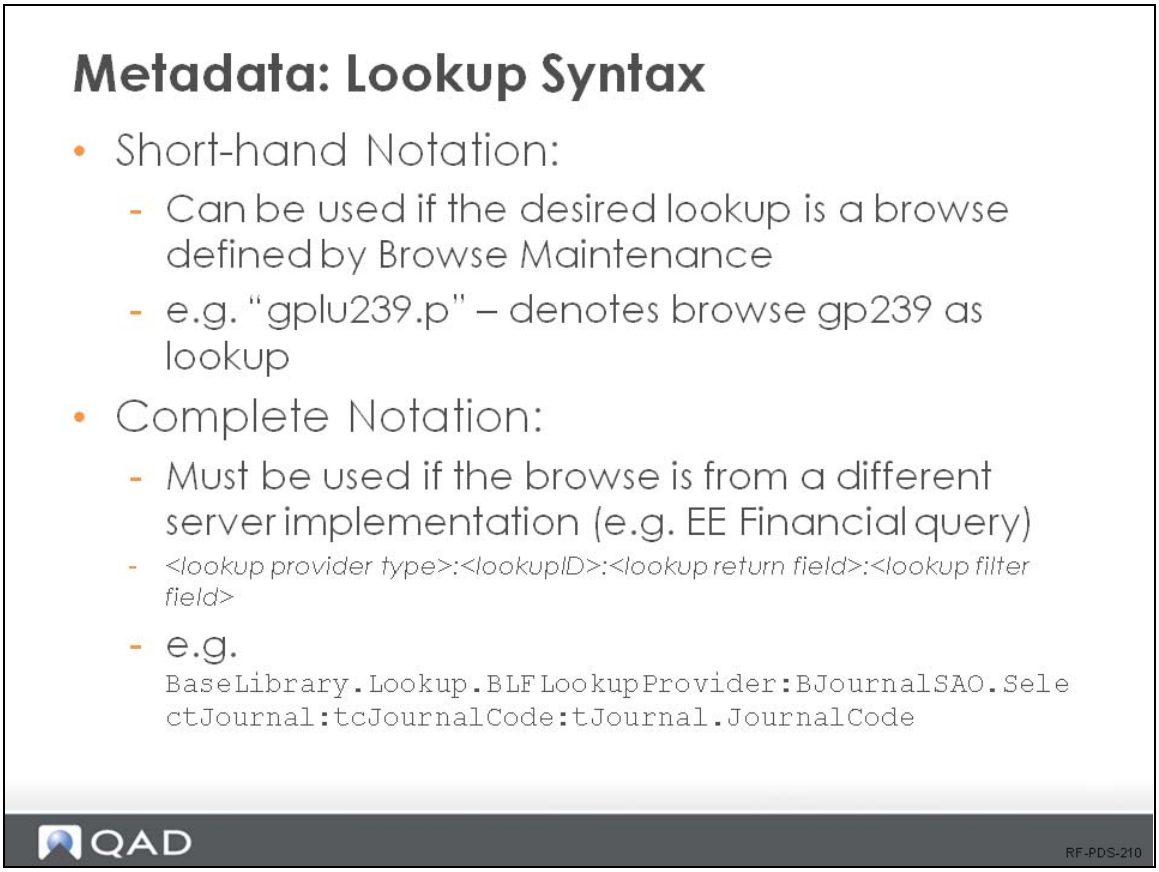
Metadata: CreateFieldForDBField - Example



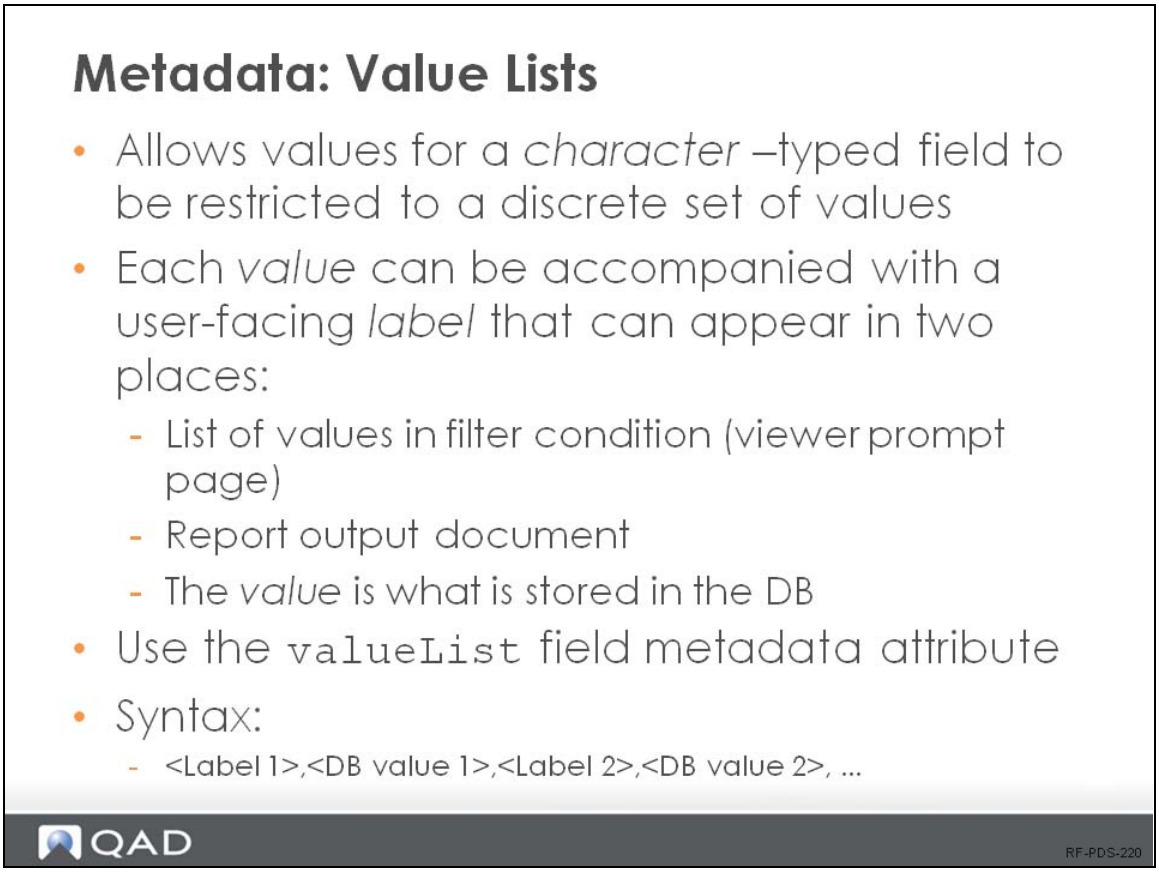
Metadata: CreateFieldLikeDBField - Example



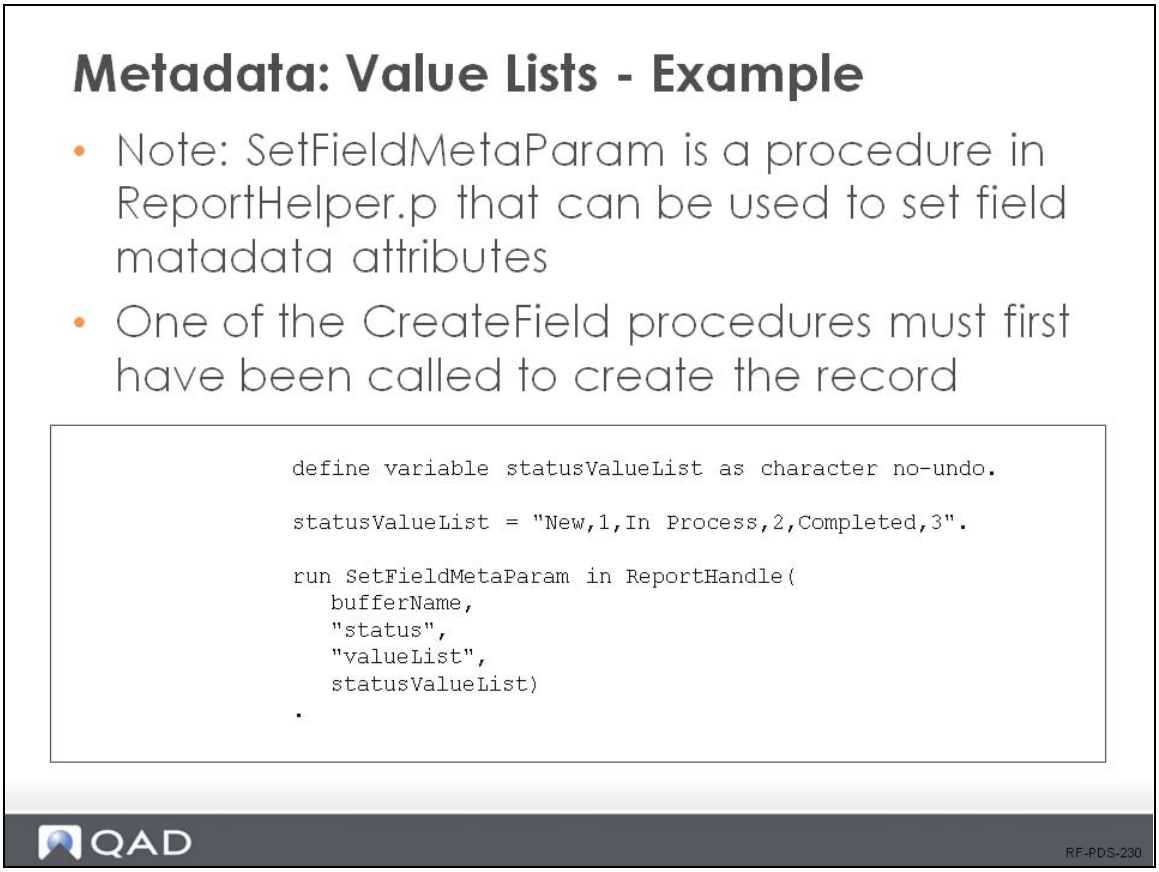
Metadata: Loookup Syntax



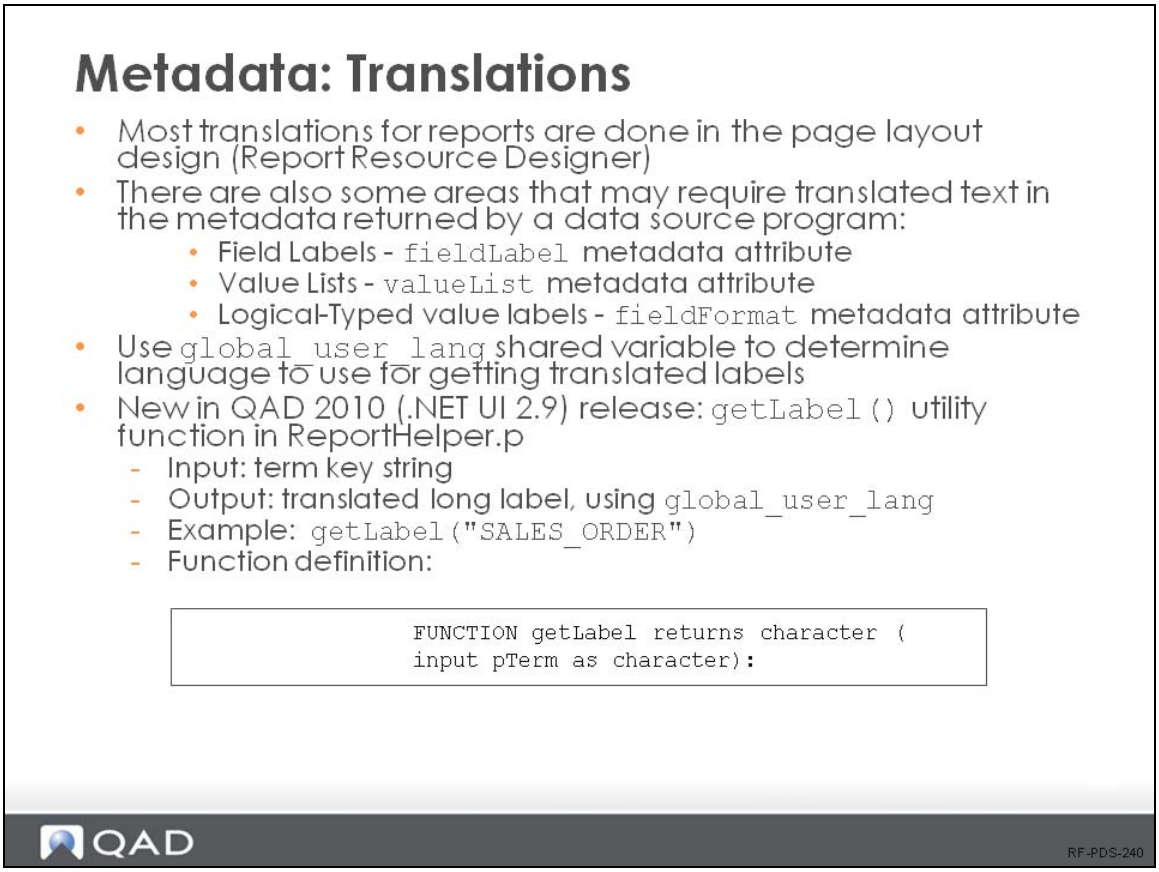
Metadata: Value Lists



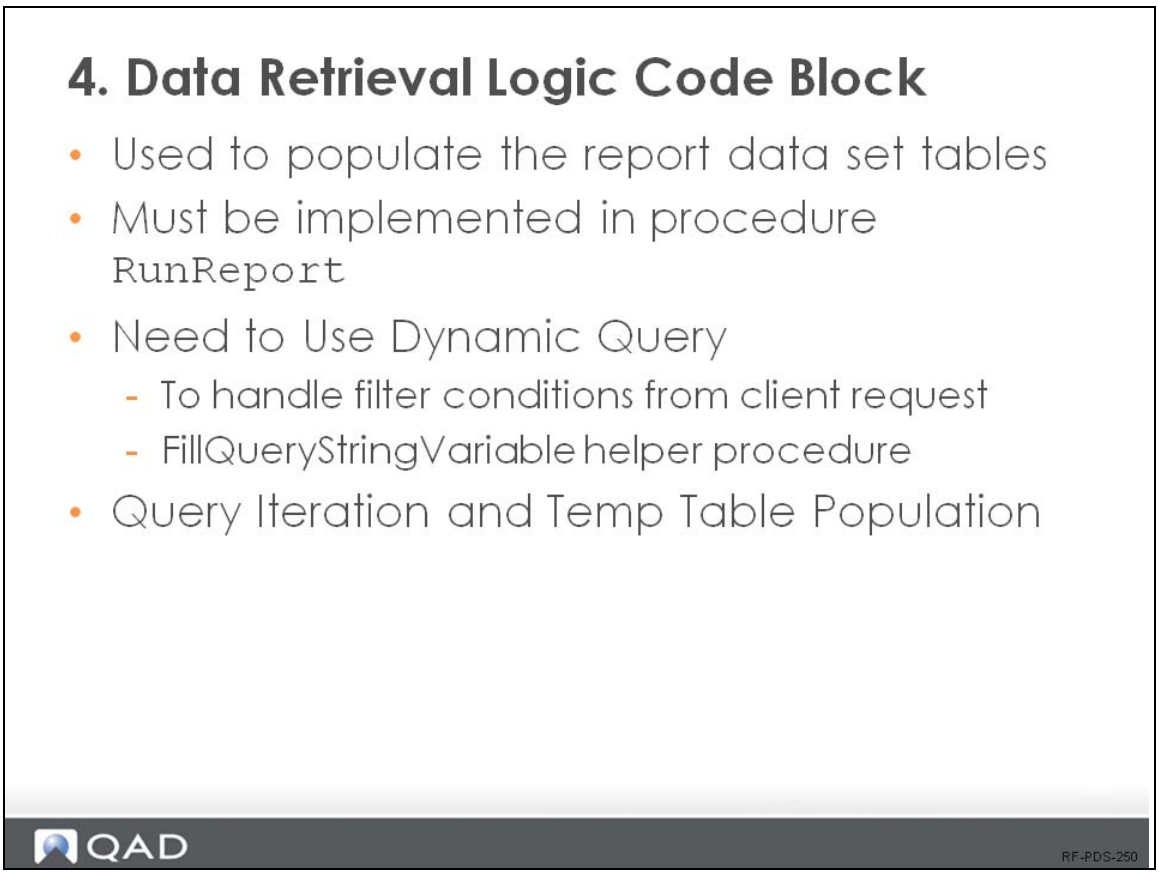
Metadata: Value Lists - Example



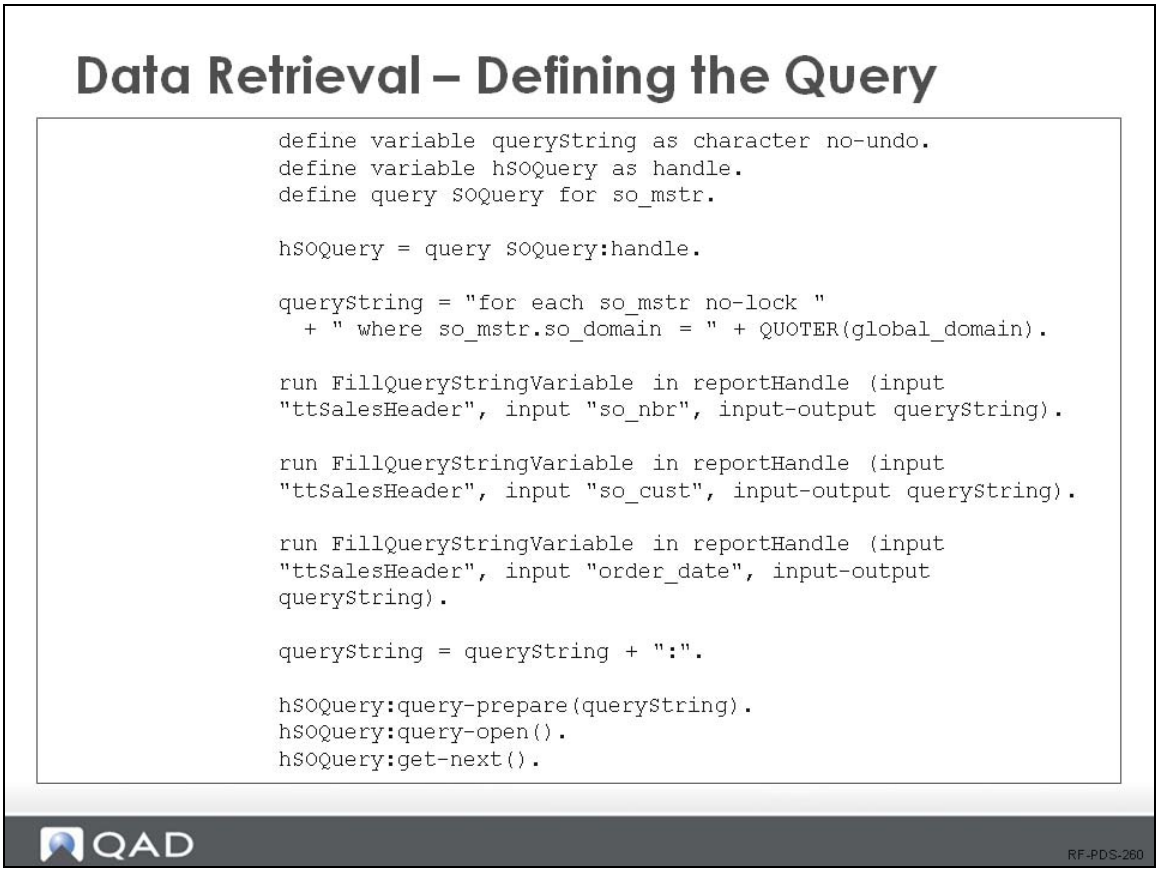
Metadata: Translations



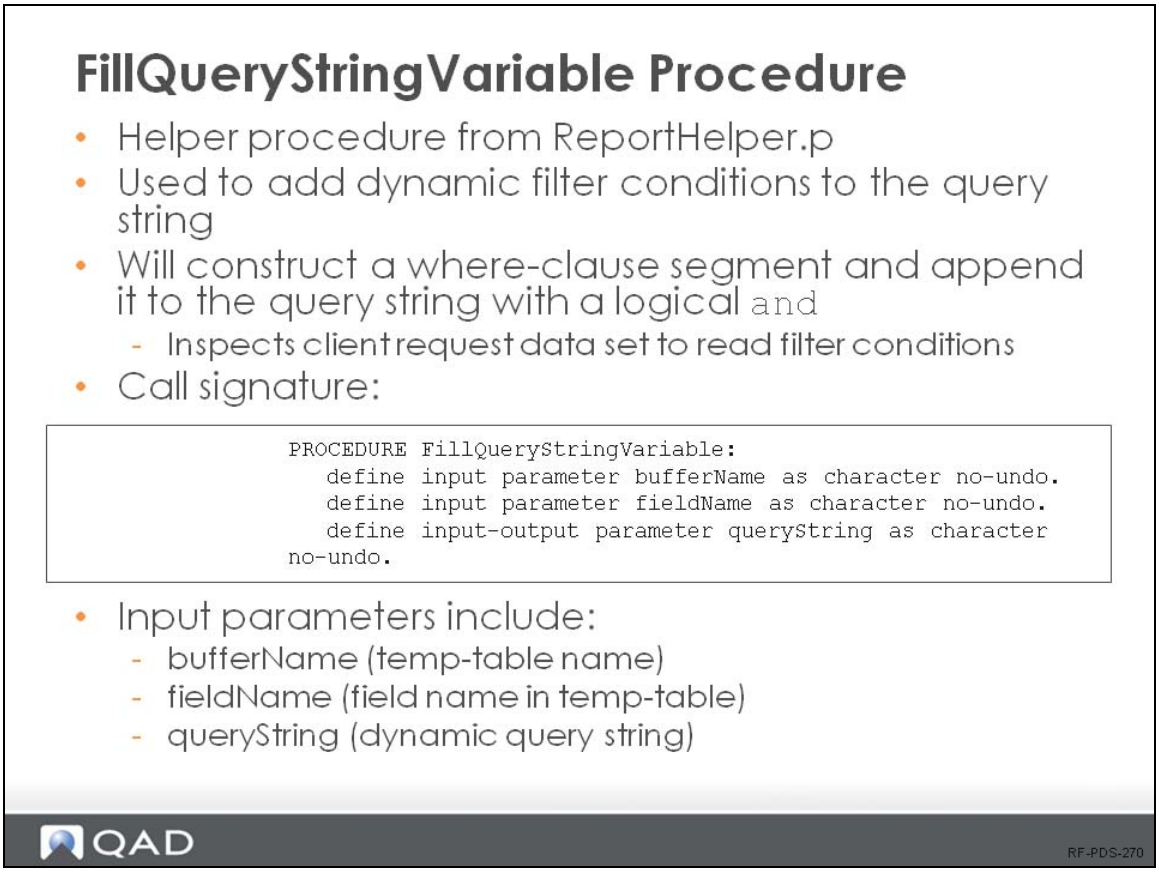
4. Data Retrieval Logic Code Block



Data Retrieval - Defining the Query



FillQueryStringVariable Procedure



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
bufferName	Input	Character	Temp-table name
fieldName	Input	Character	Field name in the temp-table
queryString	Input-Output	Character	Dynamic query string

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

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:

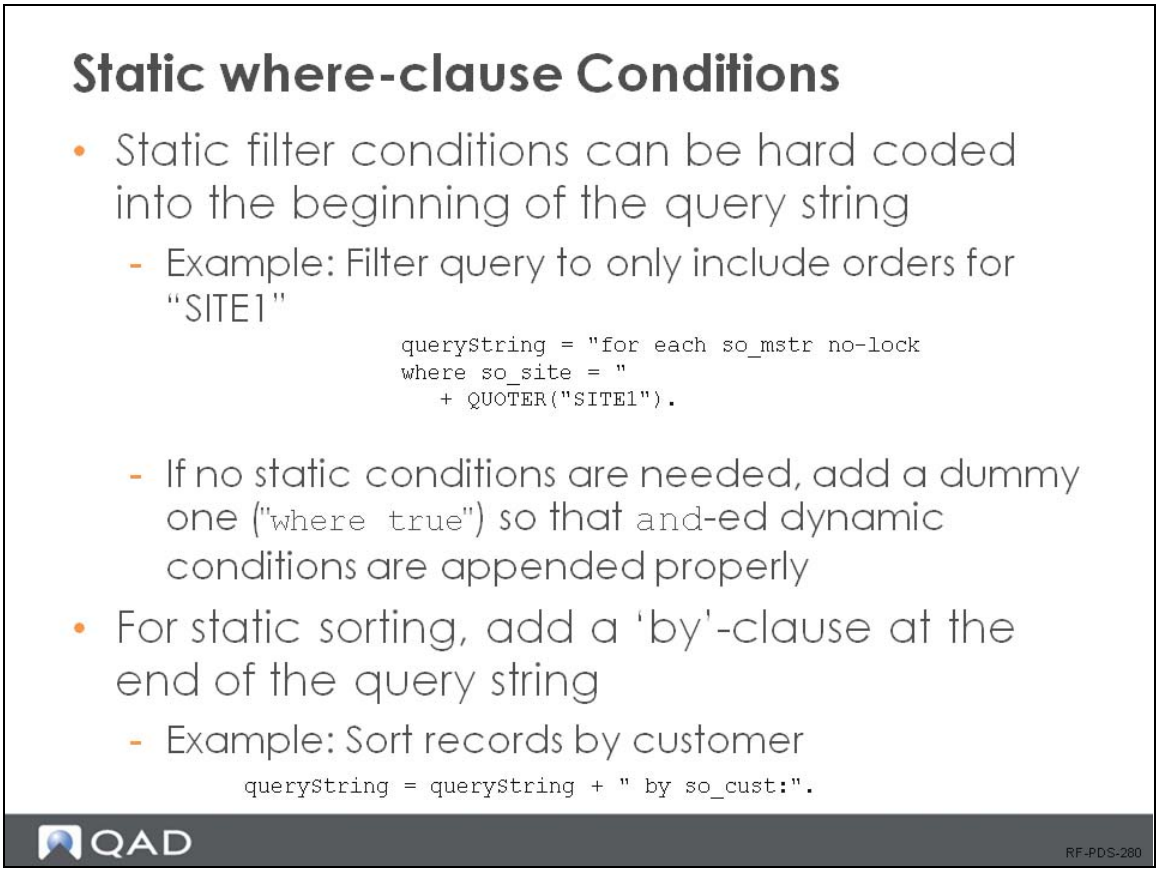
```
queryString = "for each so_mstr no-lock where so_site = " + QUOTER("SITE1").
```

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:

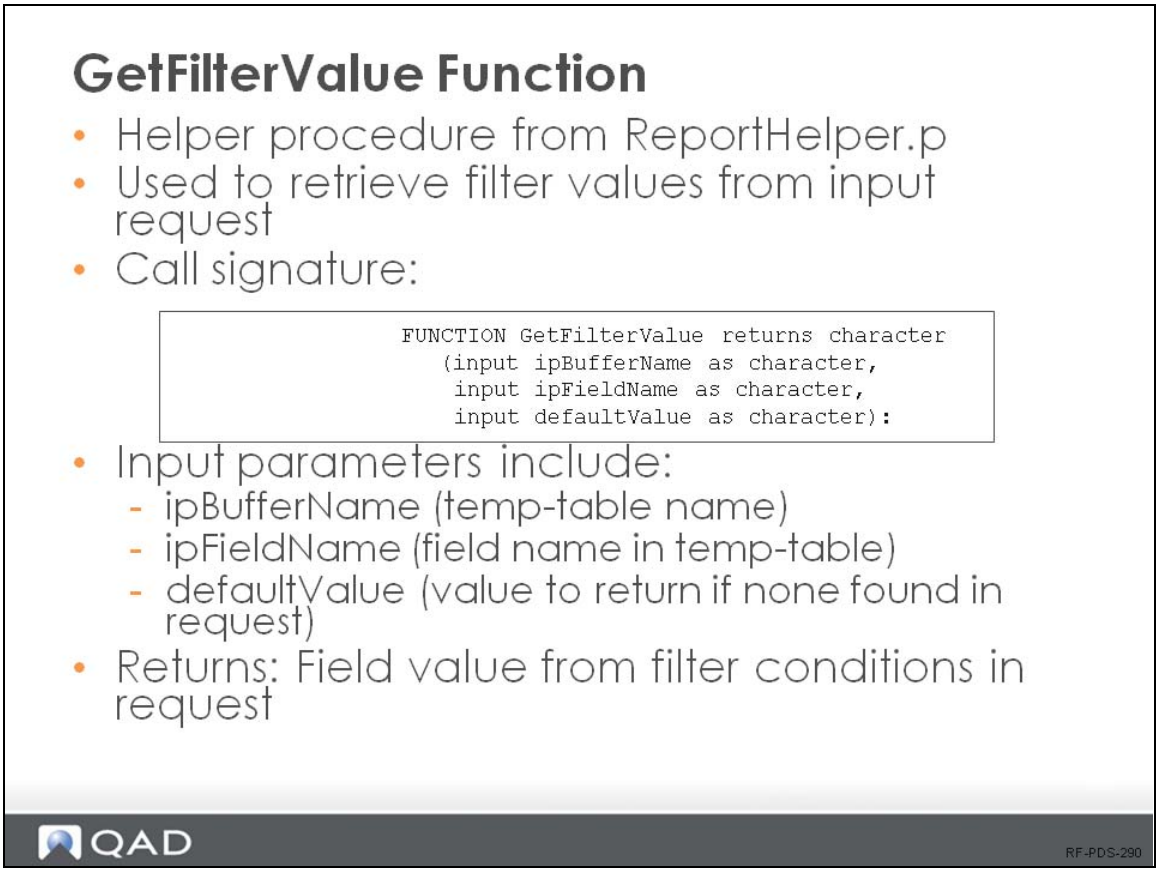
```
queryString = queryString + " by so_cust:".
```

Static where-clause Conditions



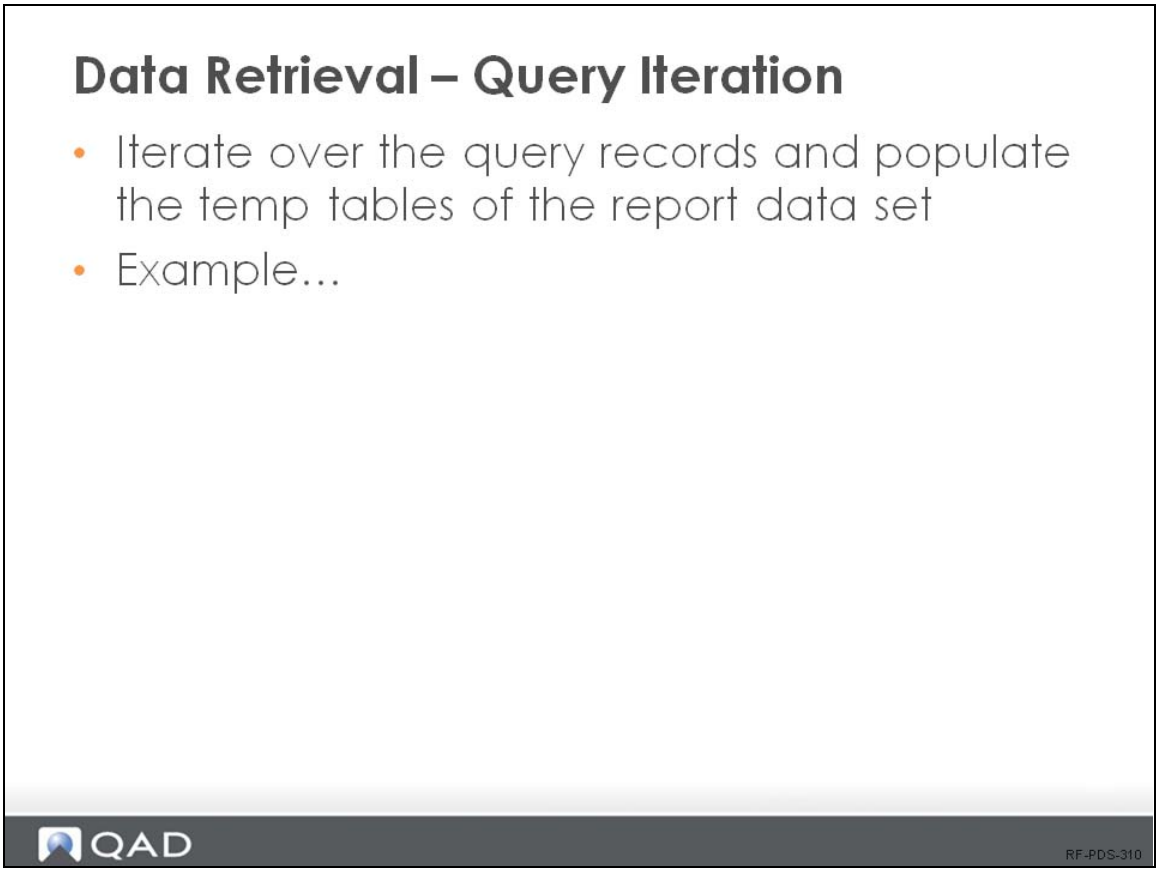
NOTE: The QUOTER() function returns the input string wrapped in quotes, and is useful when specifying text in a dynamic query string that must appear quoted in the final query string.

GetFilterValue Function

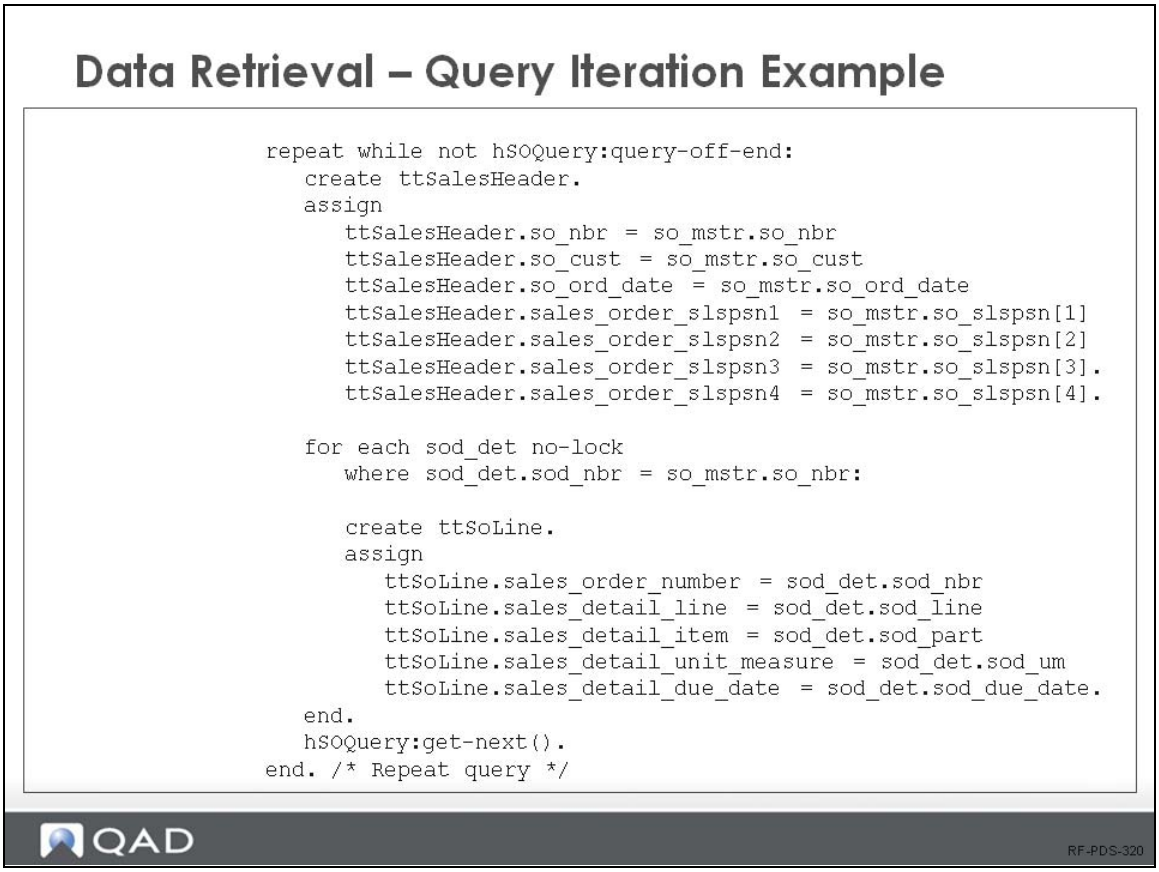


GetFilterValue - Example

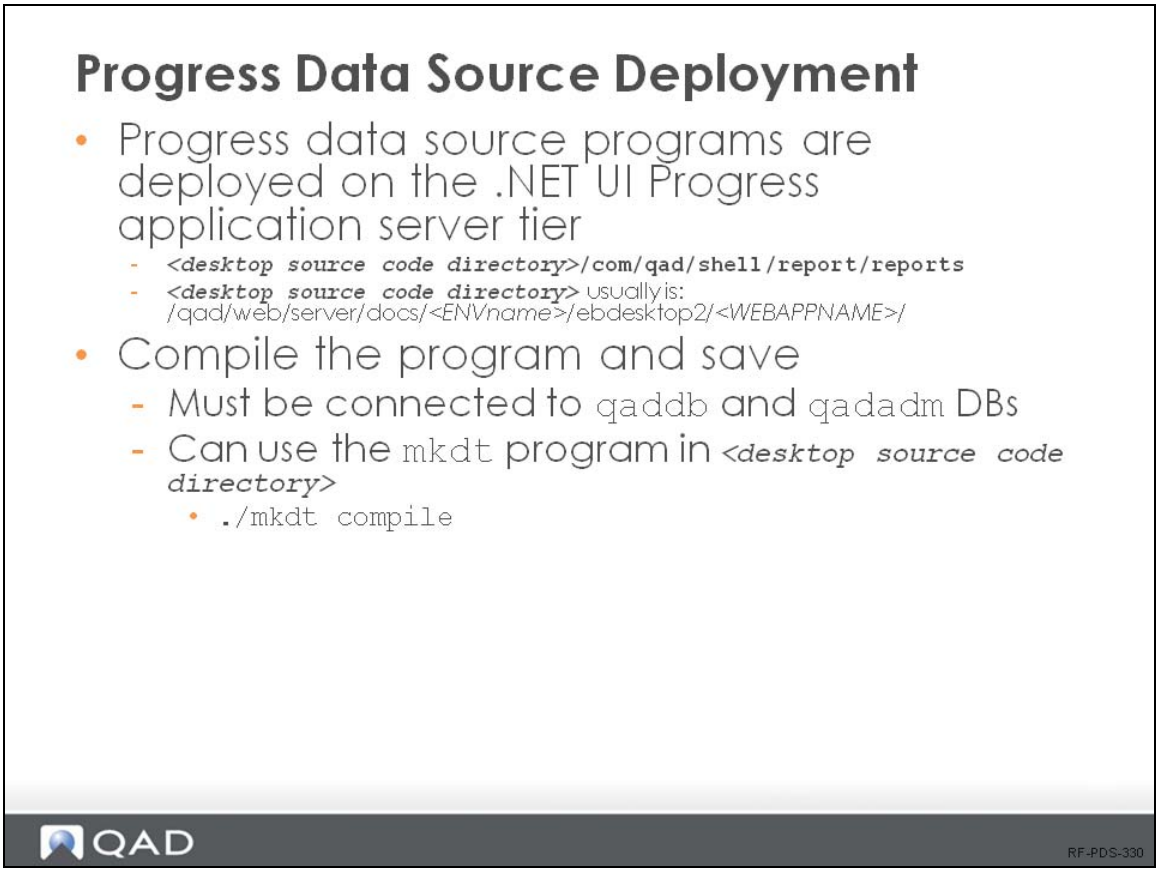
Data Retrieval - Query Iteration



Data Retrieval - Query Iteration Example

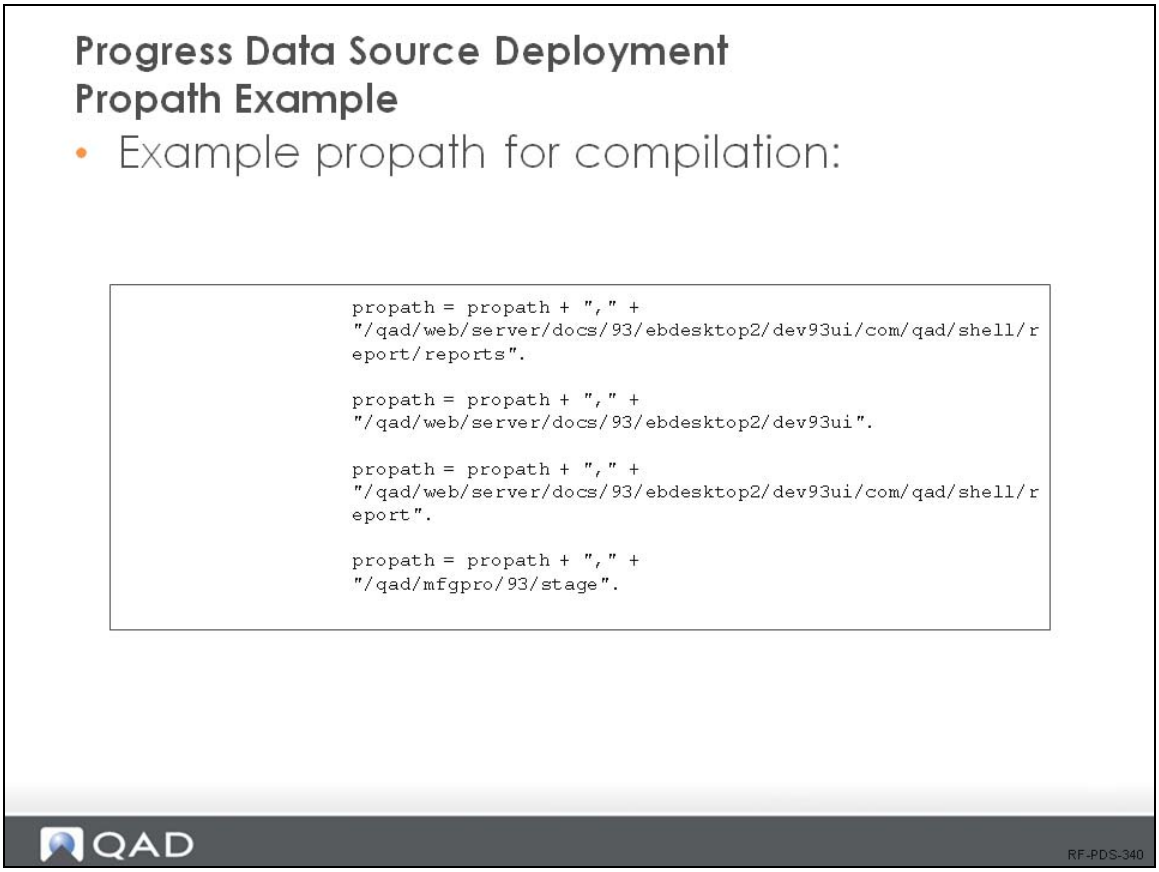


Progress Data Source Deployment

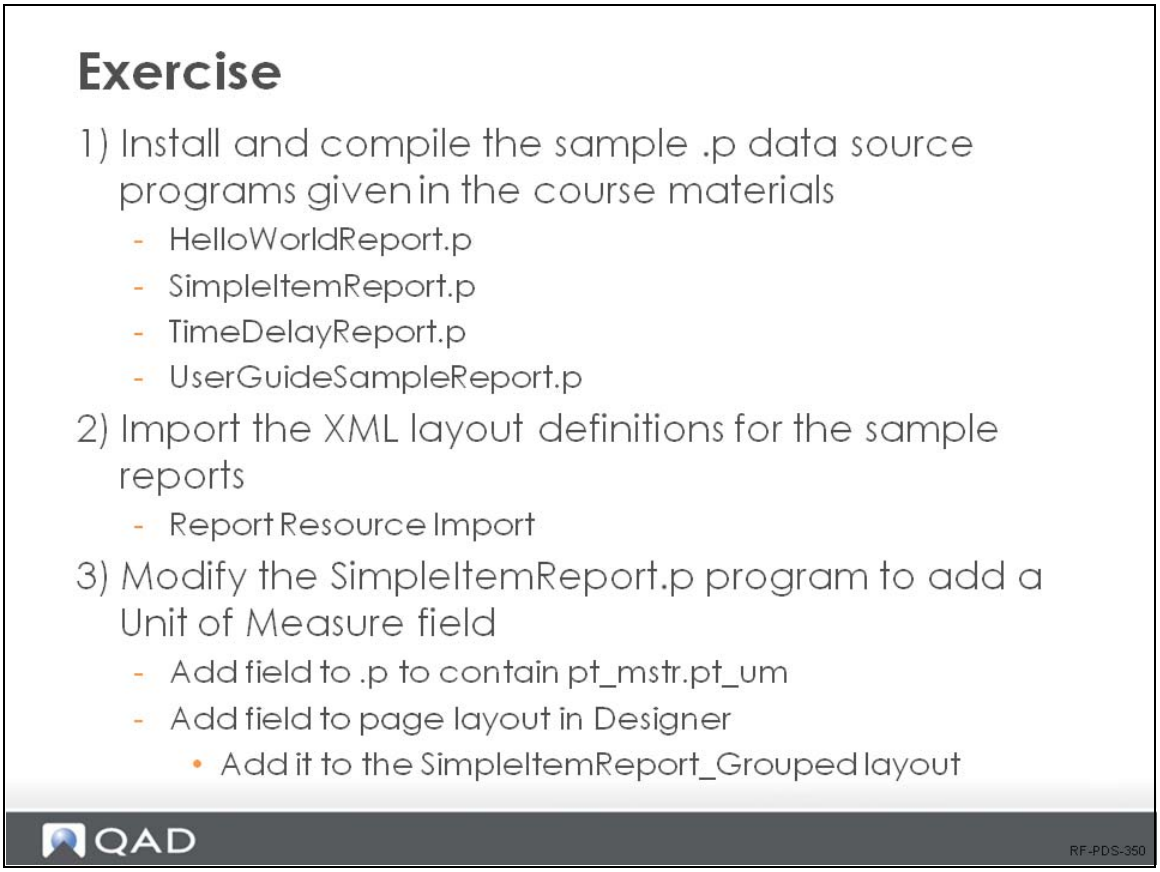


Note If you prefer to physically locate your Progress data source programs in another directory path, that is fine. Just be sure to realize that at run-time the reporting system will attempt to locate the program using the path `com/qad/shell/report/reports/<DataSourceRef>`, where `<DataSourceRef>` is 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

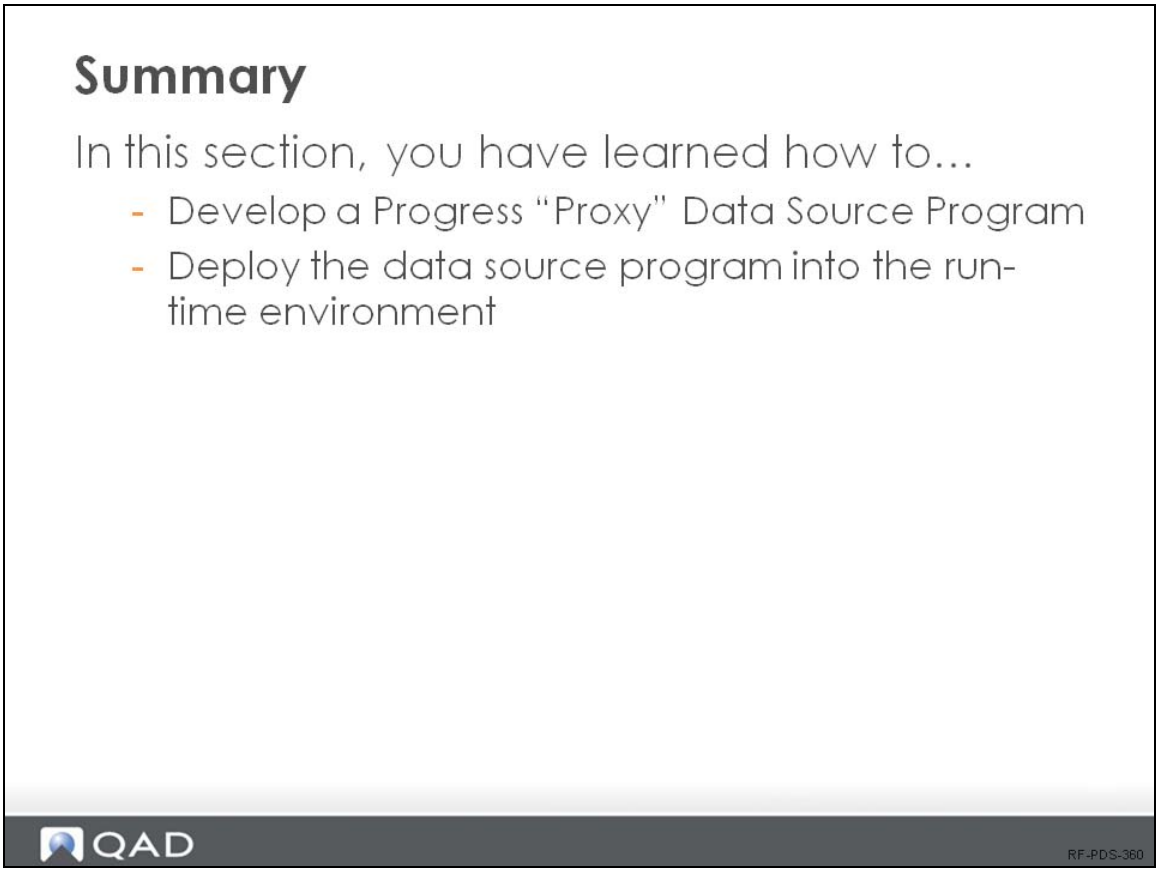


Exercise



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

Summary



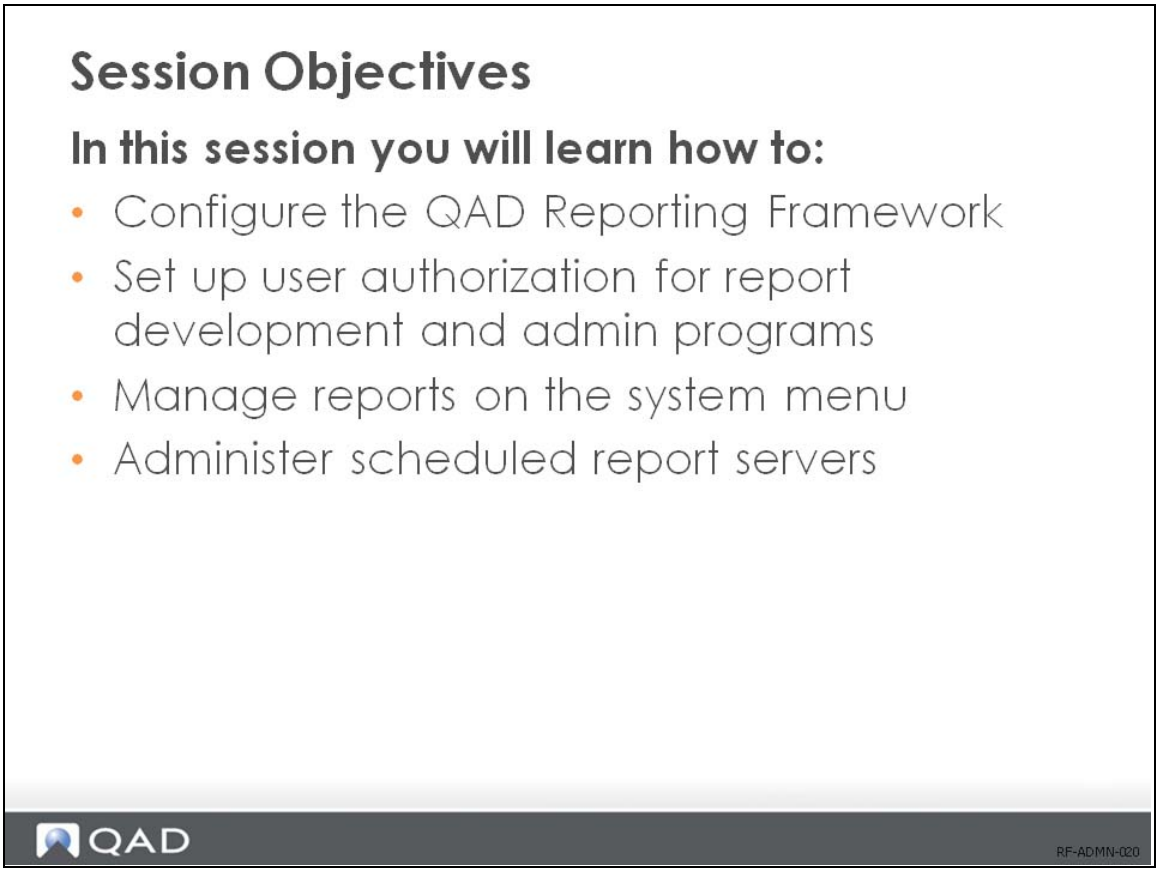
In this section, you have learned how to:

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

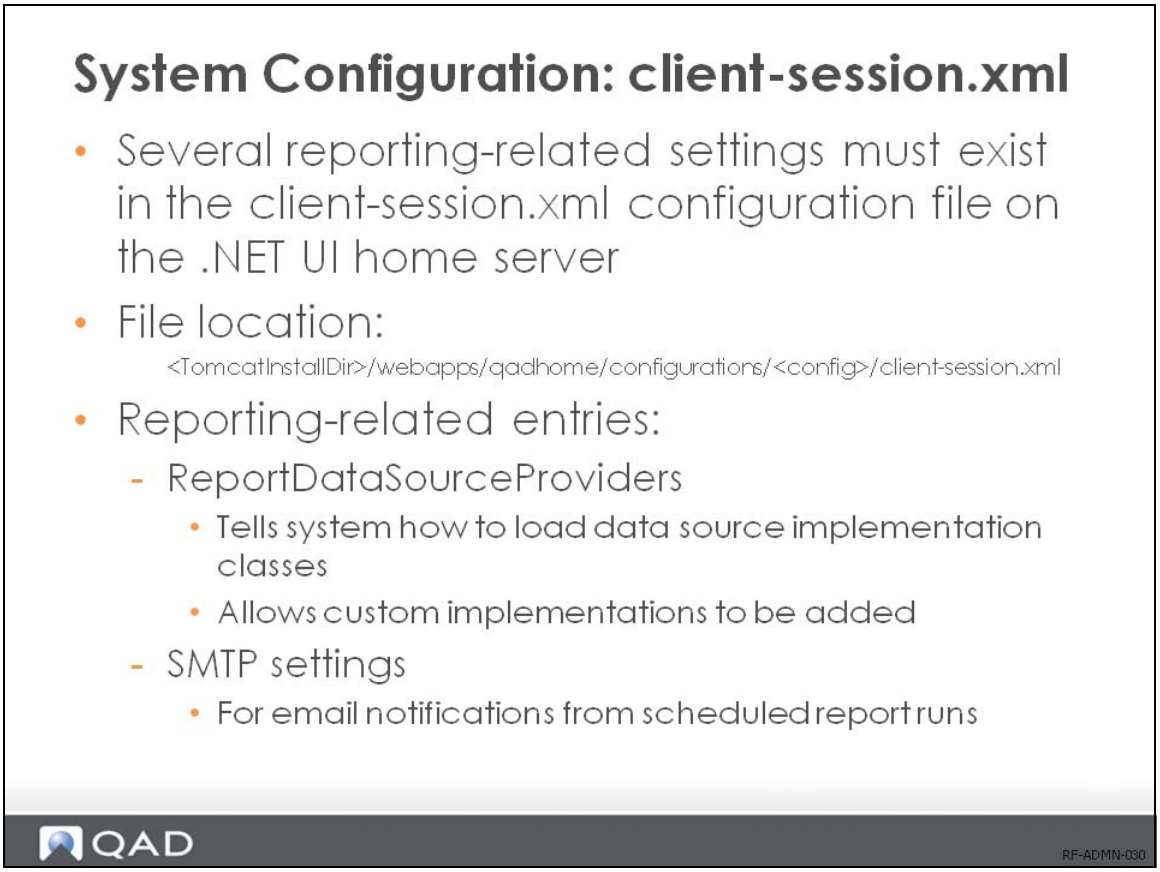
Chapter 4

Reporting Framework Administration

Session Objectives

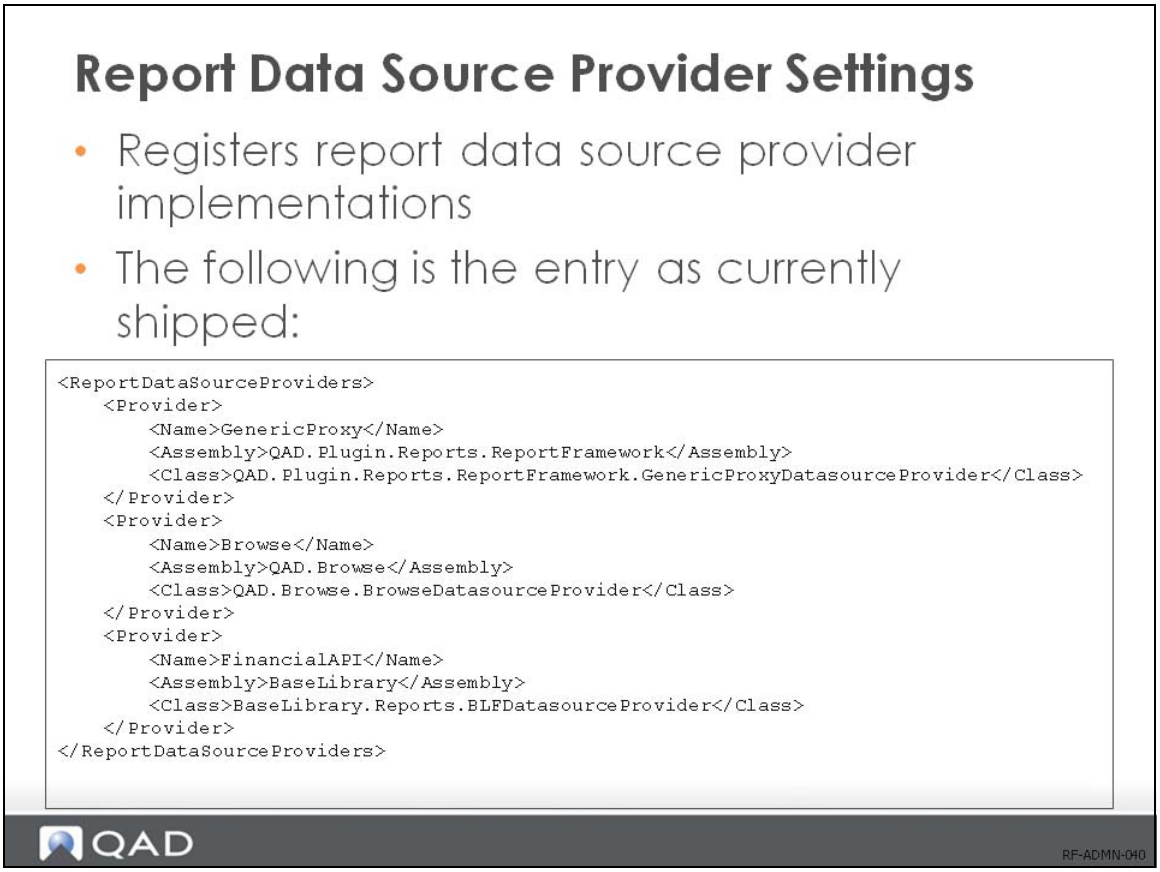


System Configuration: client-session.xml



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 QAD .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.

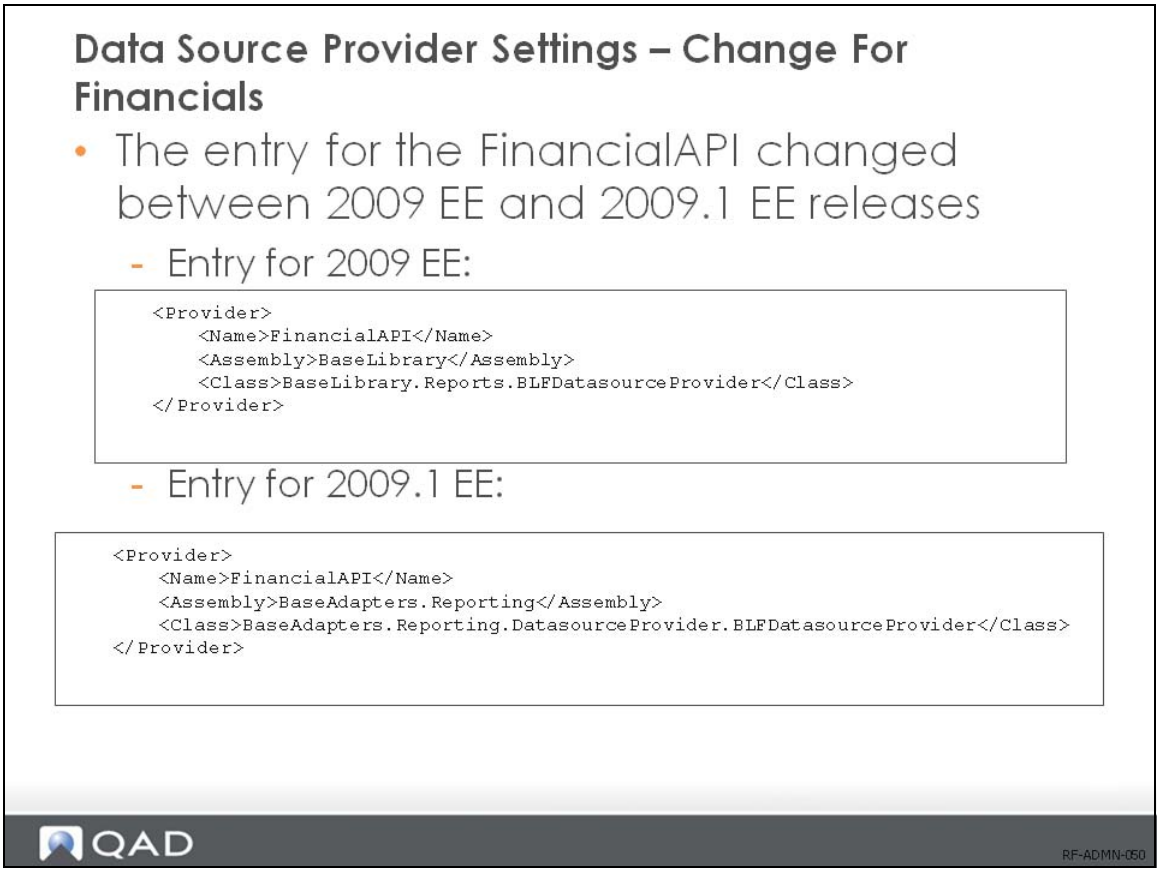
Report Data Source Provider Settings



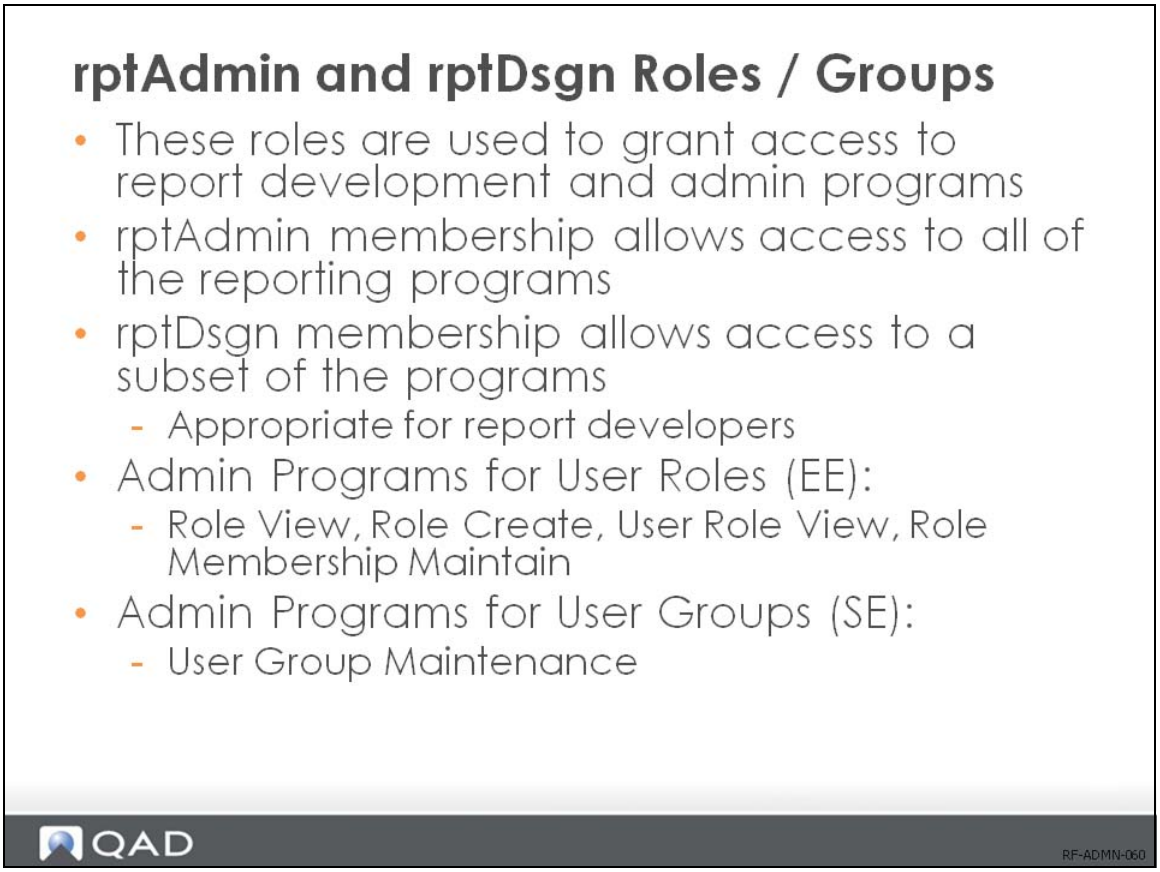
It is possible to write a new custom data source provider implementation. For example, you might have a non-QAD data store that you wish to report against.

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



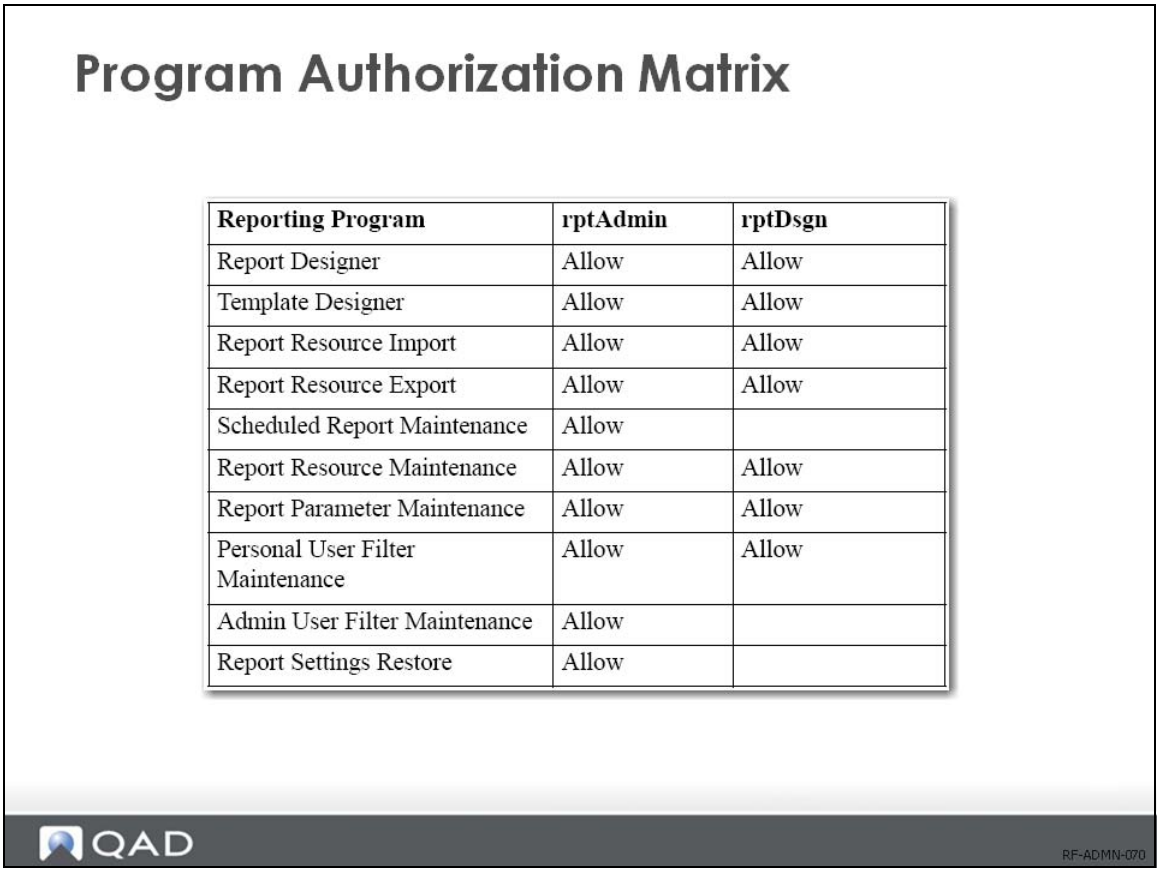
rptAdmin and rptDsgn Roles / Groups



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



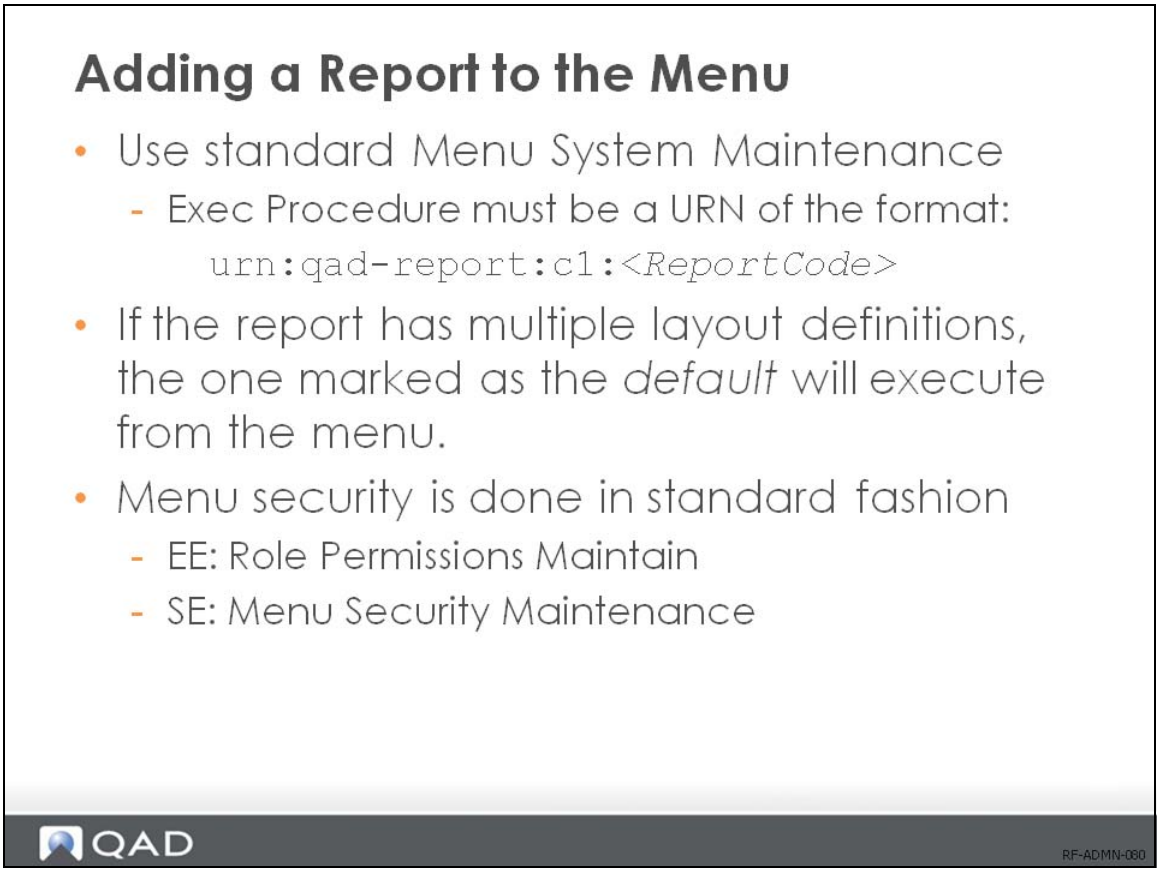
You must grant the rptAdmin and rptDsgn roles access to the following programs based on this table:

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	

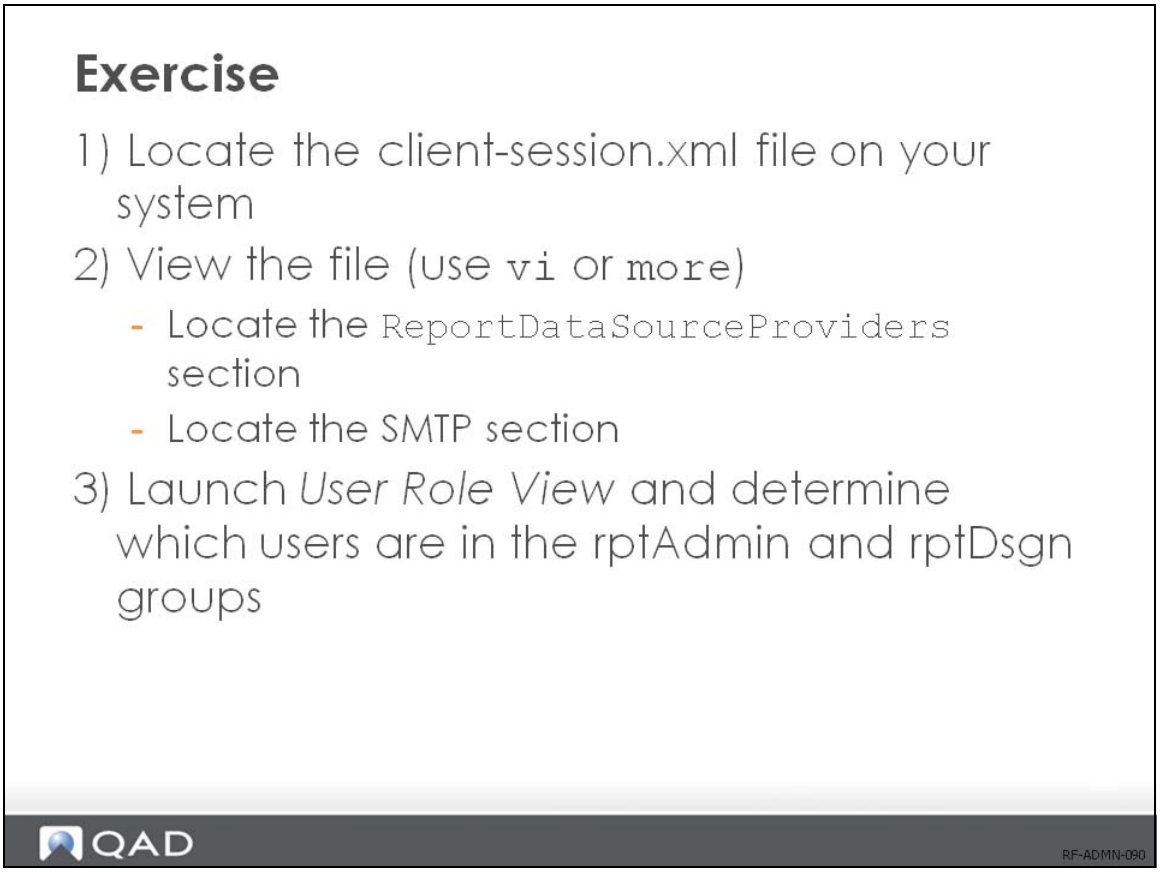
For details about common access security features, see *User Guide: QAD Security and Controls.*

Adding a Report to the Menu



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

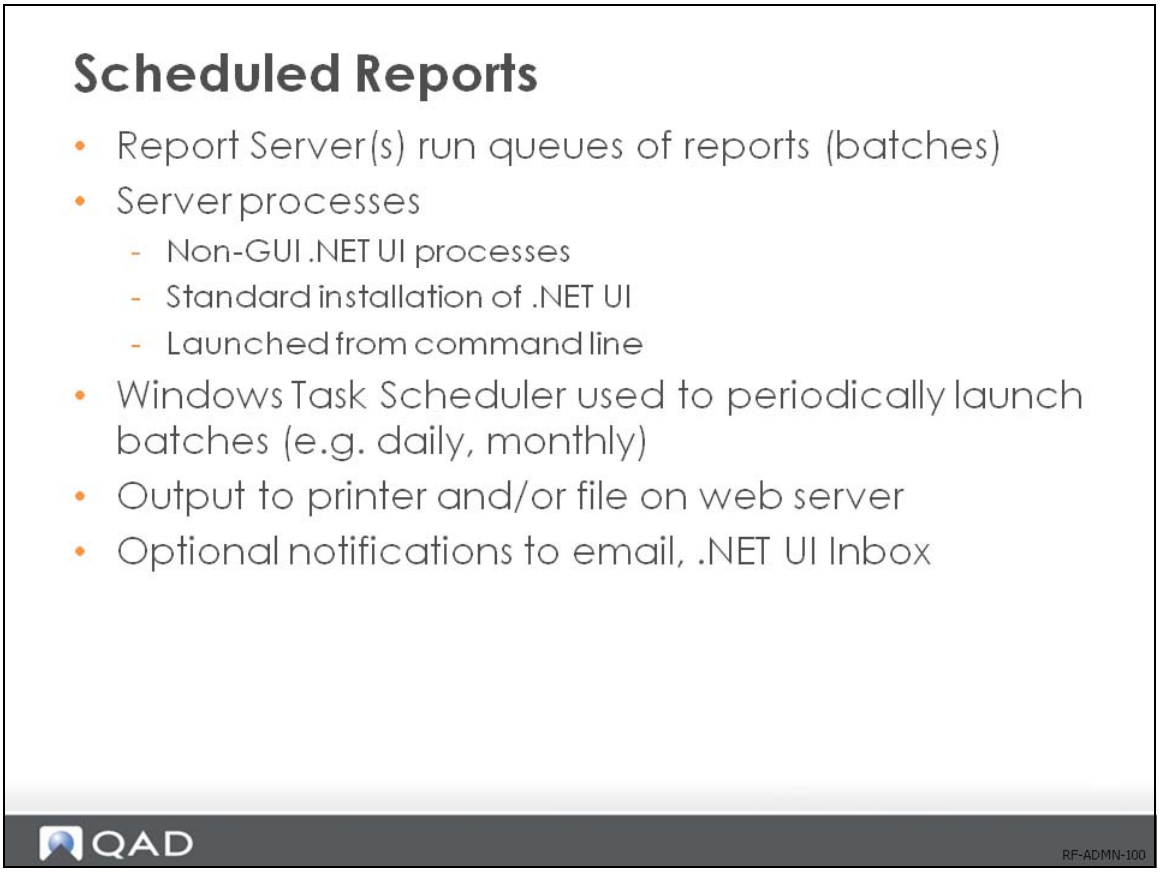
Exercise



Exercises include the following:

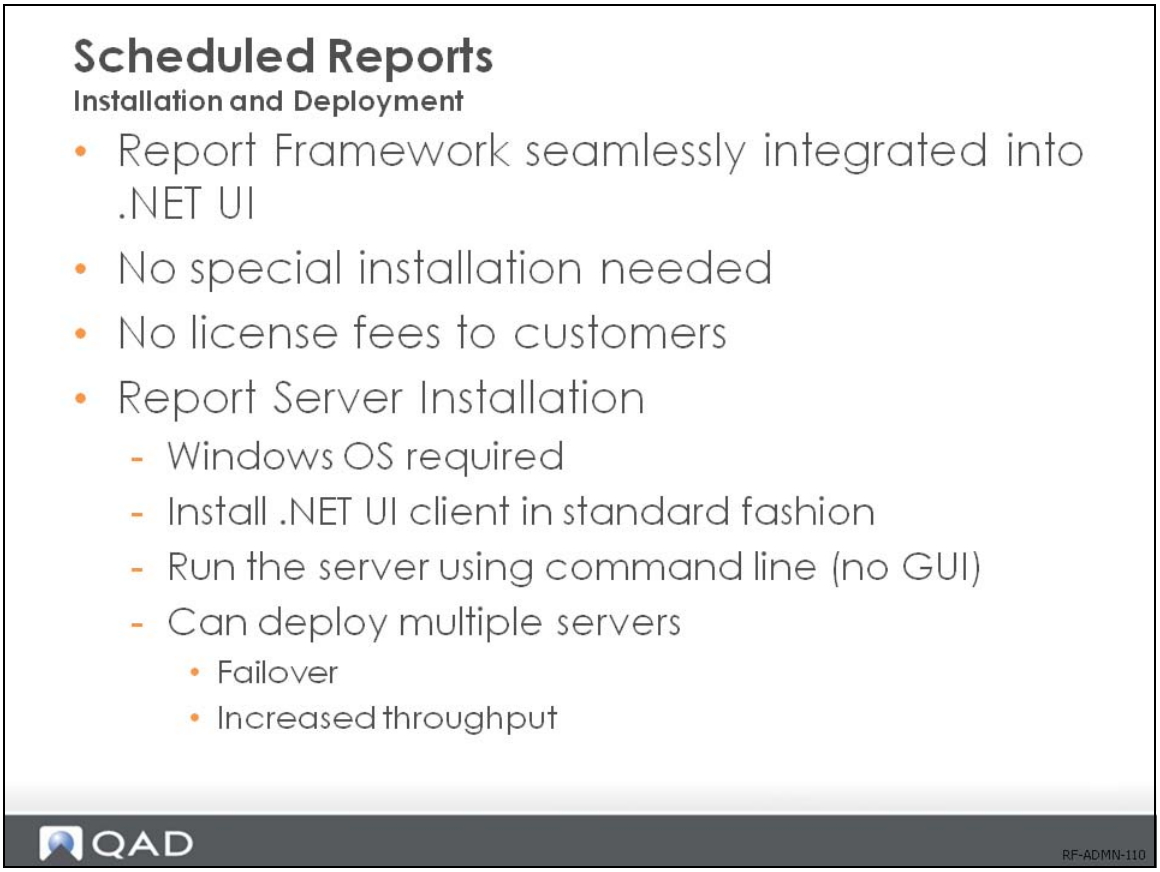
1. Locate the client-session.xml file on your system.
2. View the file (use vi or more):
   - Locate the ReportDataSourceProviders section
   - Locate the SMTP section
3. Launch User Role View and determine which users are in the rptAdmin and rptDsgn groups.

Scheduled Reports

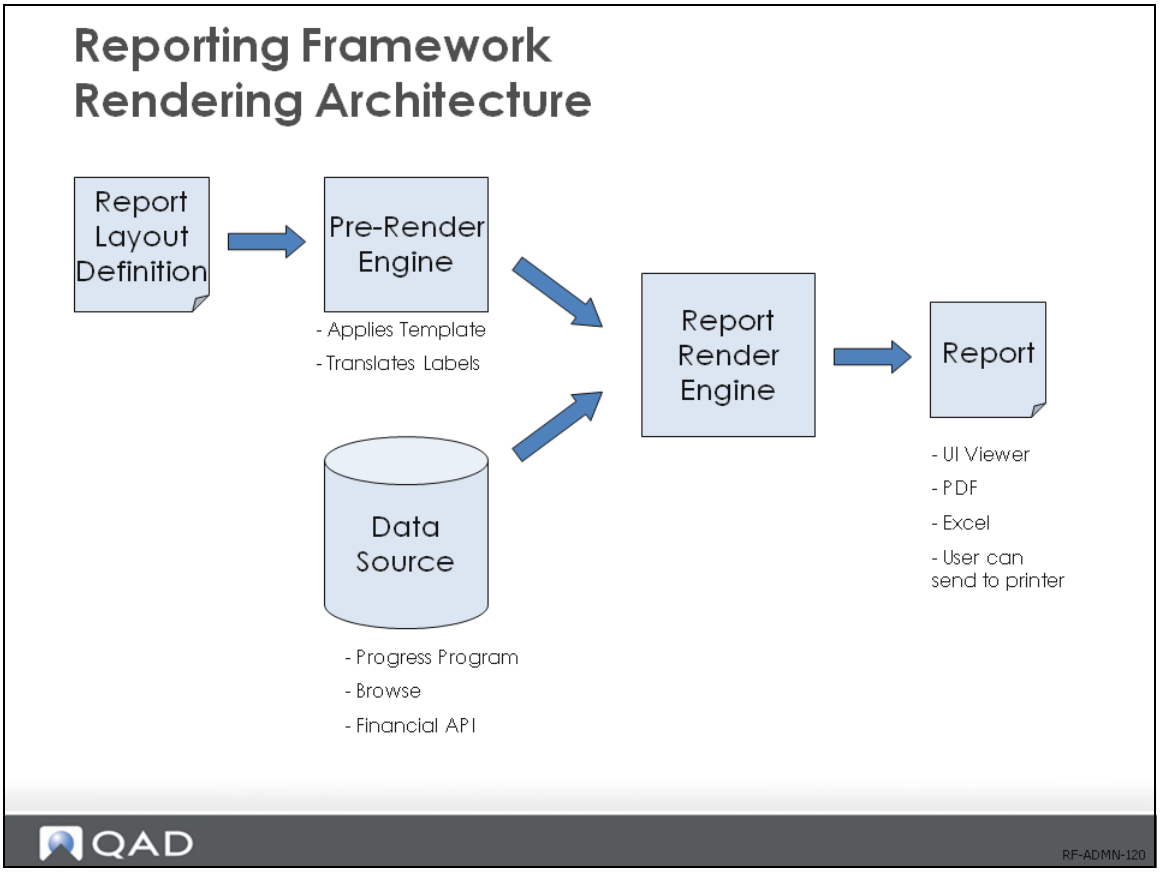


NOTE: QRF scheduled report servers should not be confused with the financial report daemons in the EE product. The latter is separate from the QRF and is used in conjunction with the Crystal Reports for EE financials.

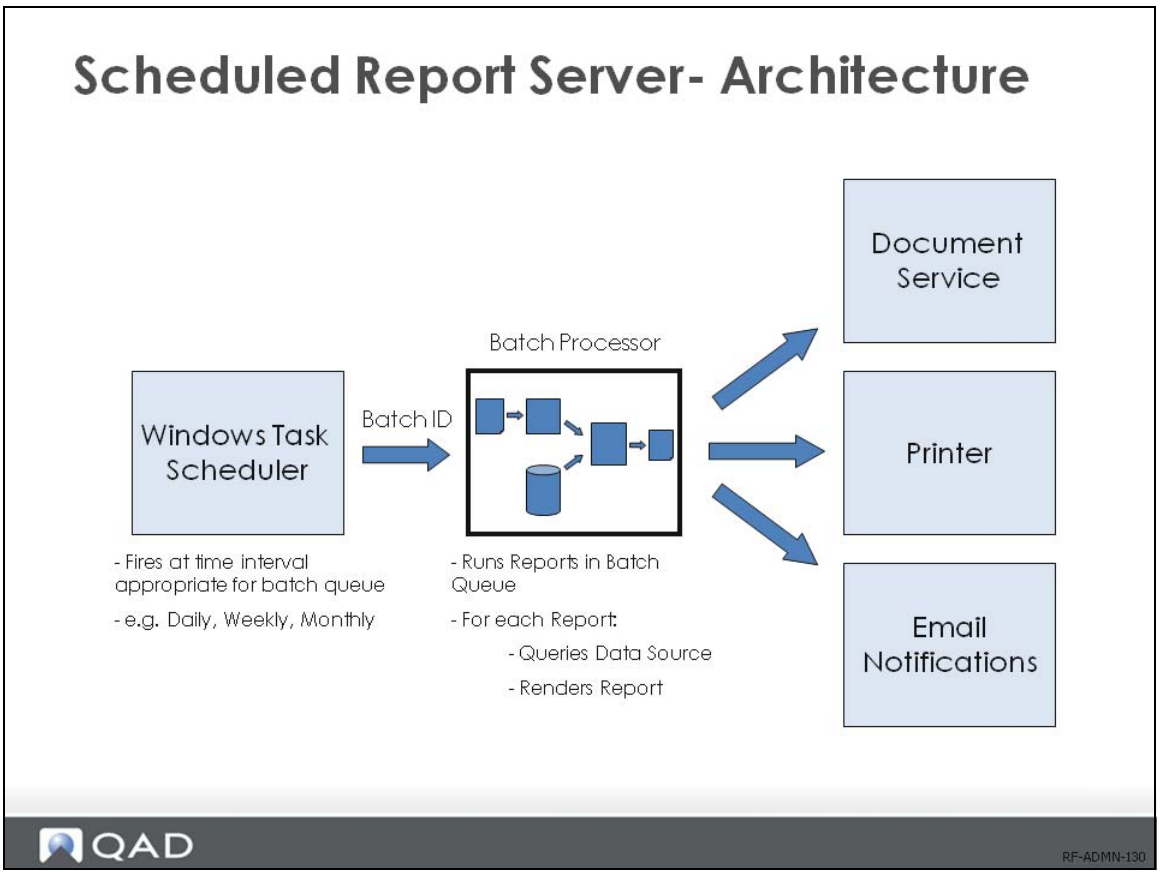
Scheduled Reports: Installation and Deployment



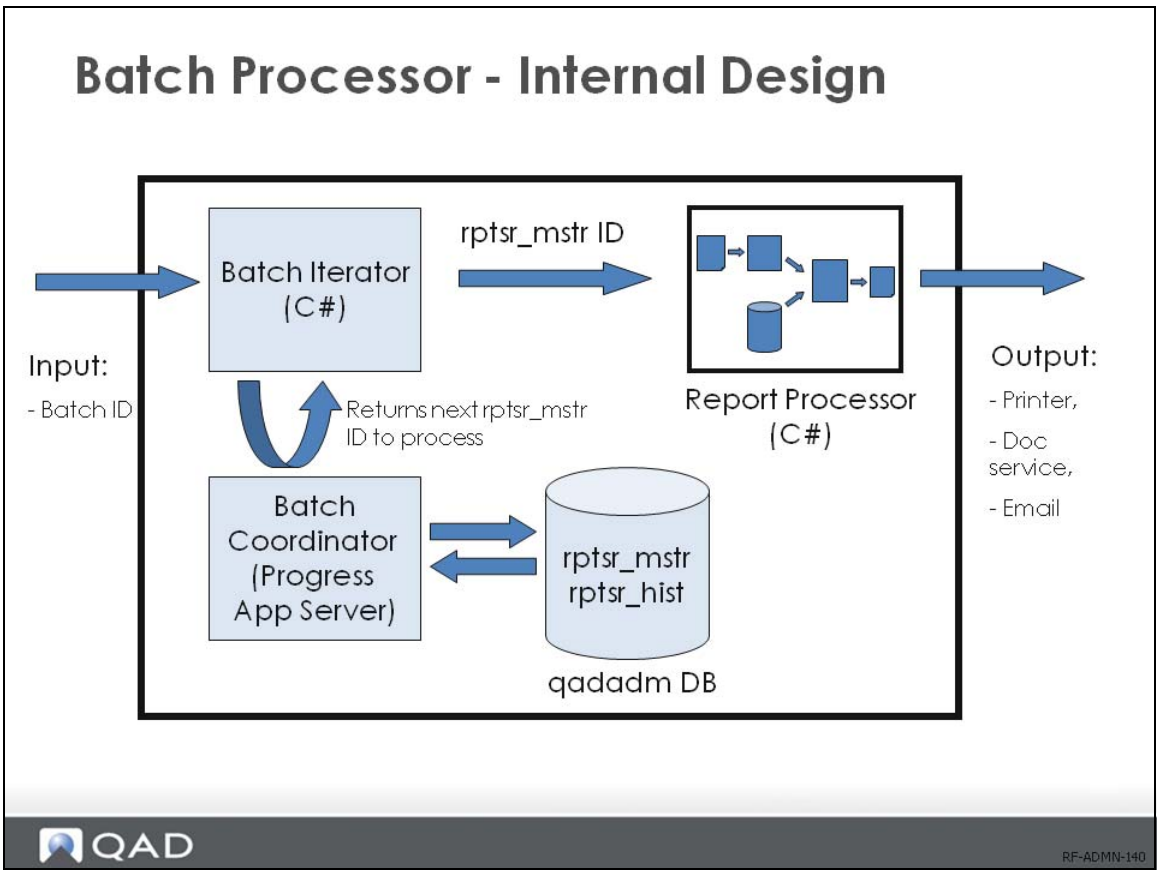
Reporting Framework: Rendering Architecture



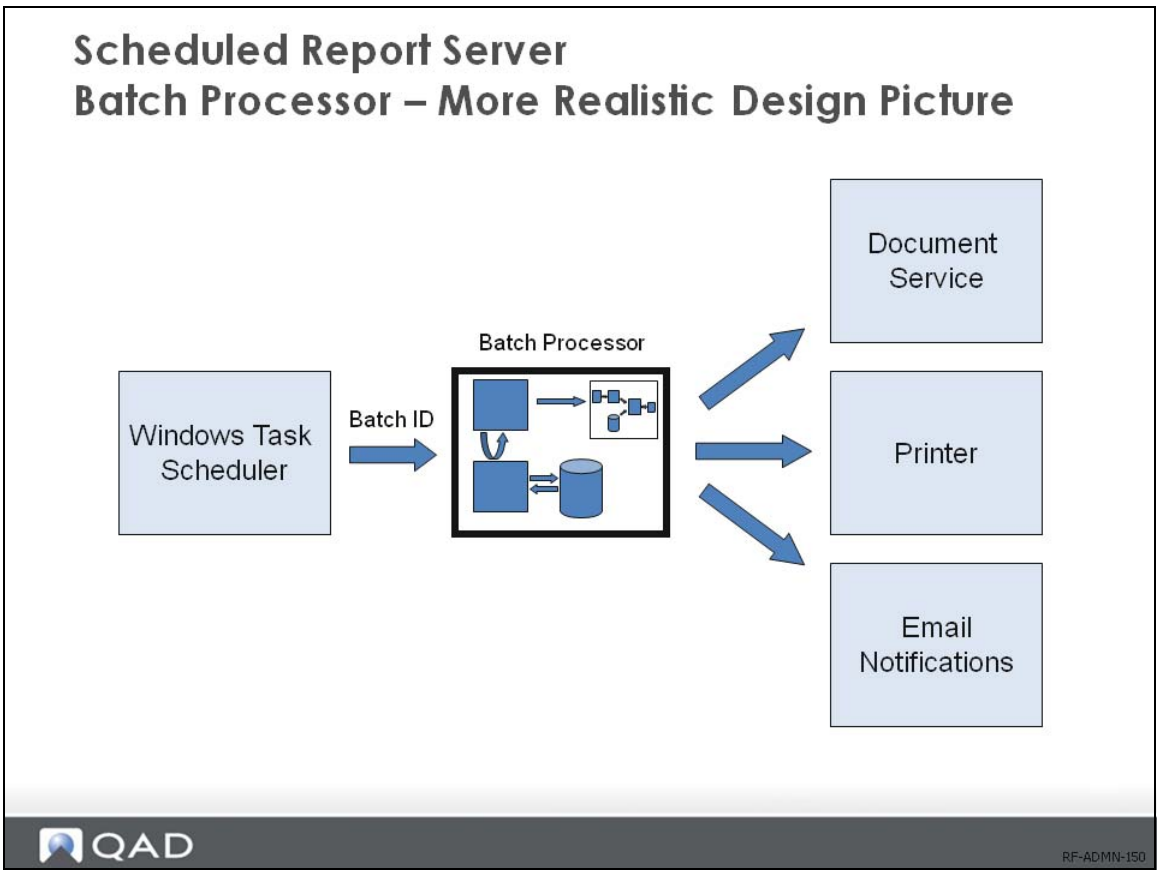
Scheduled Report Server - Architecture



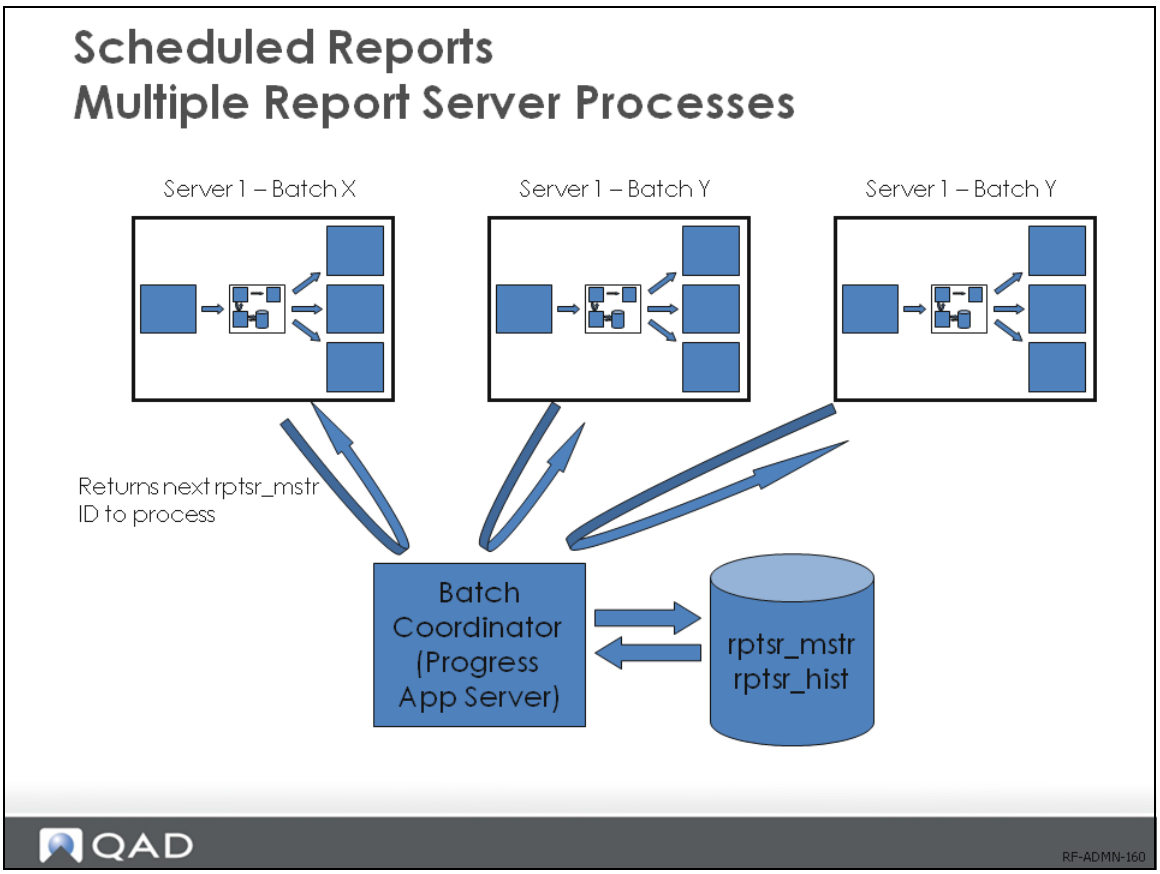
Batch Processor - Internal Design



Scheduled Report Server: Batch Processor Design



Scheduled Reports: Multiple Report Server Processes



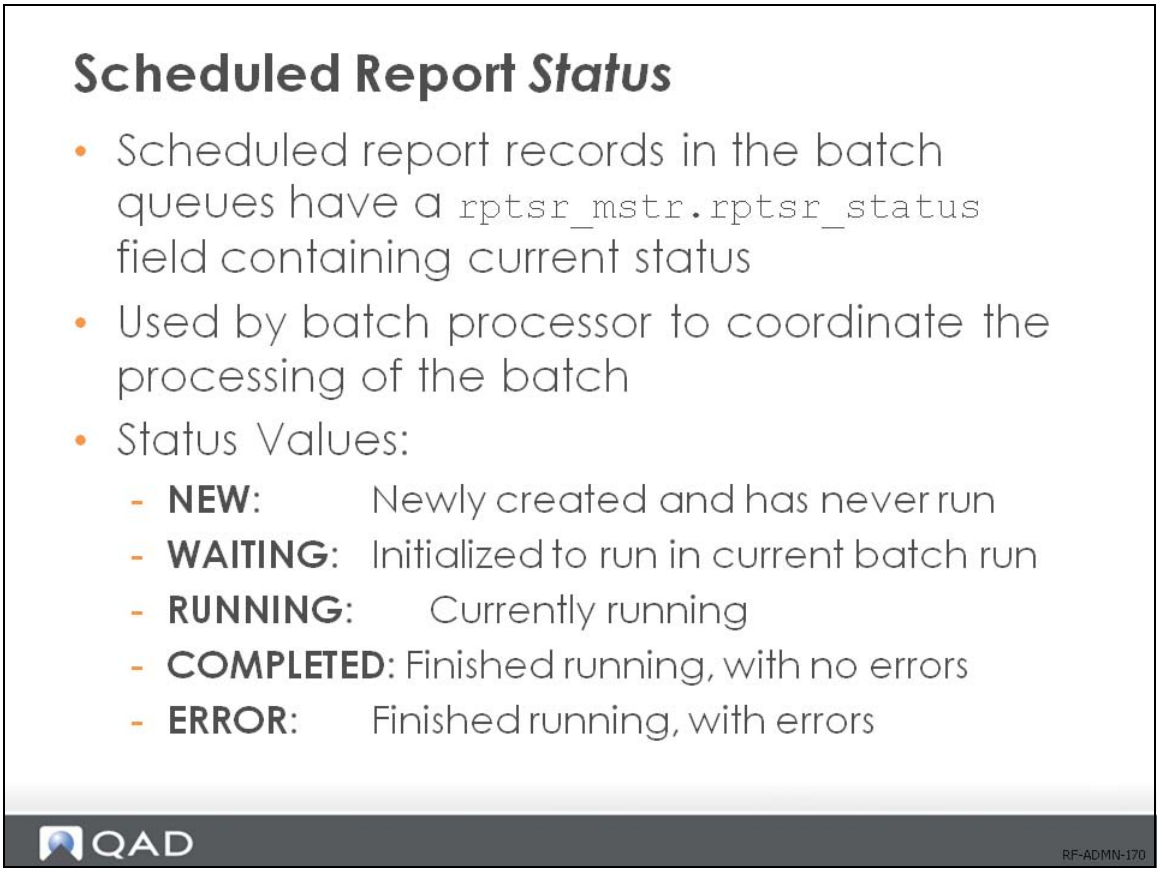
When multiple report server processes are run, they can be either on the same machine or different machines.

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.

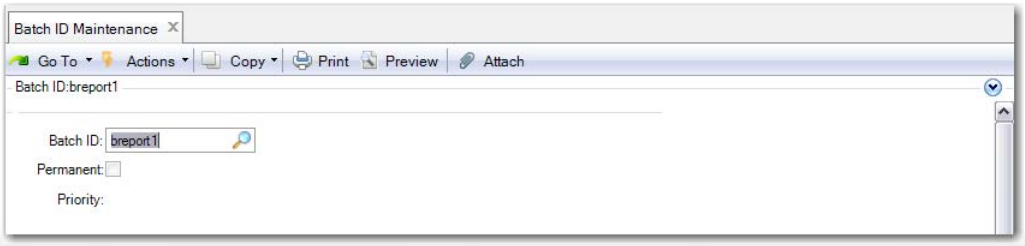
Scheduled Reports Status



Scheduled Report Administration: Batch ID Maintenance

Scheduled Report Administration
Batch ID Maintenance

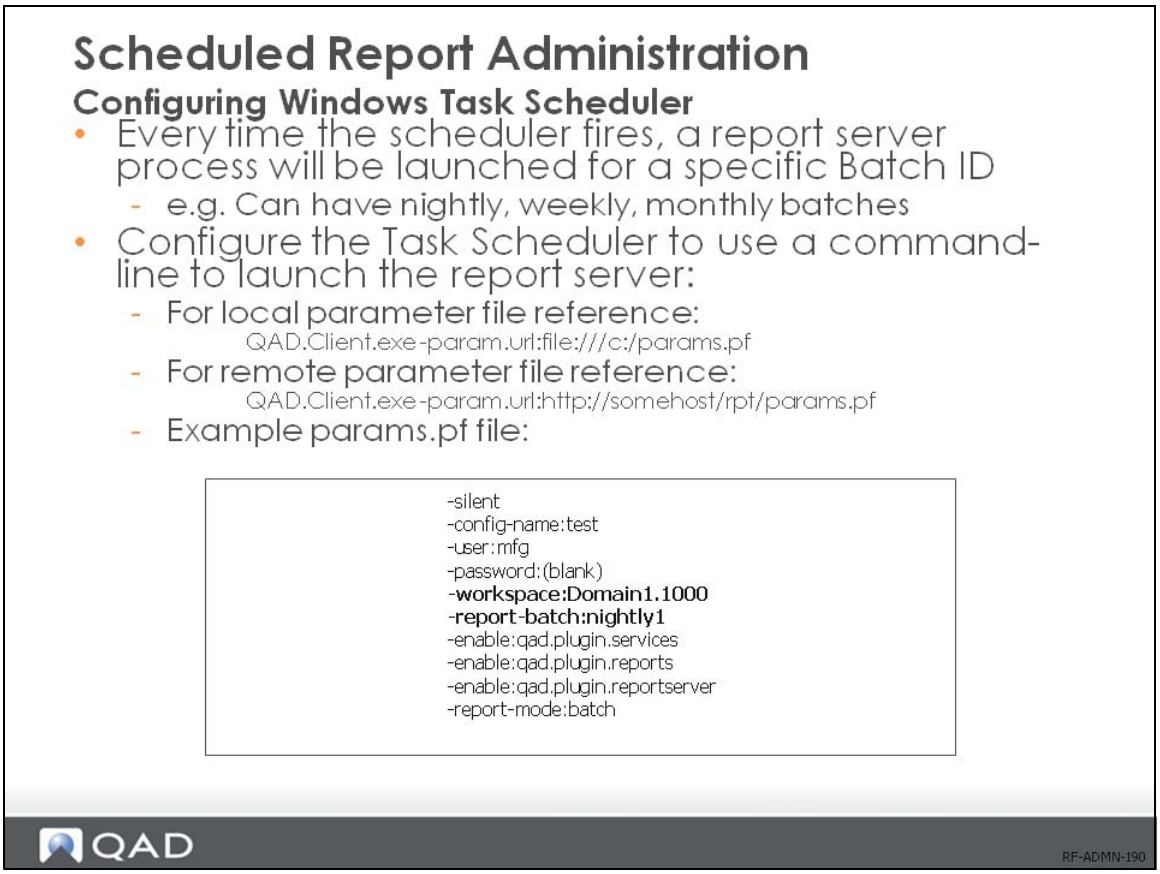
- Same batch DB table used (bc_mstr) as for legacy MFG/PRO reports
  - Different detail table: rptsr_mstr (one record per scheduled report)
  - Each batch ID can logically a queue of scheduled reports (as rptsr_mstr records)
- Same *Batch ID Maintenance* program as for legacy MFG/PRO reports



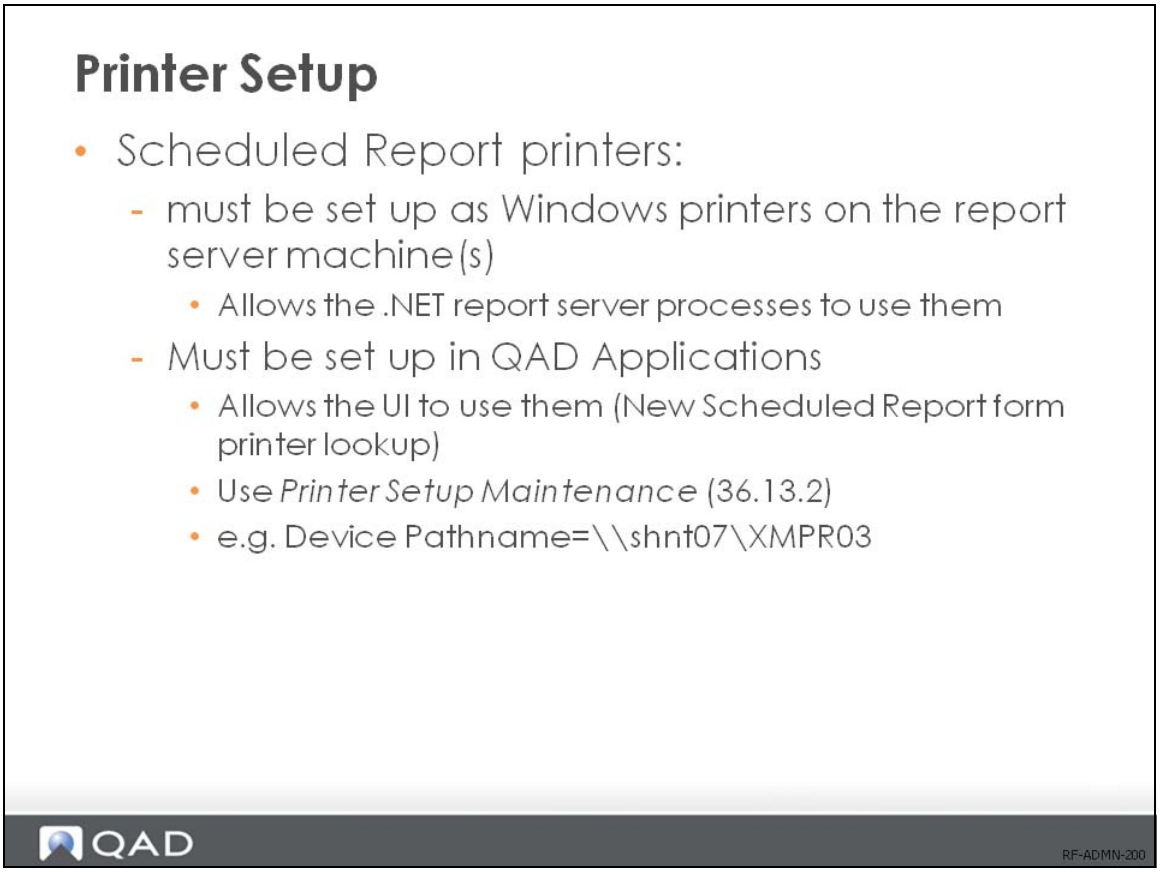
QAD

RF-ADMN-180

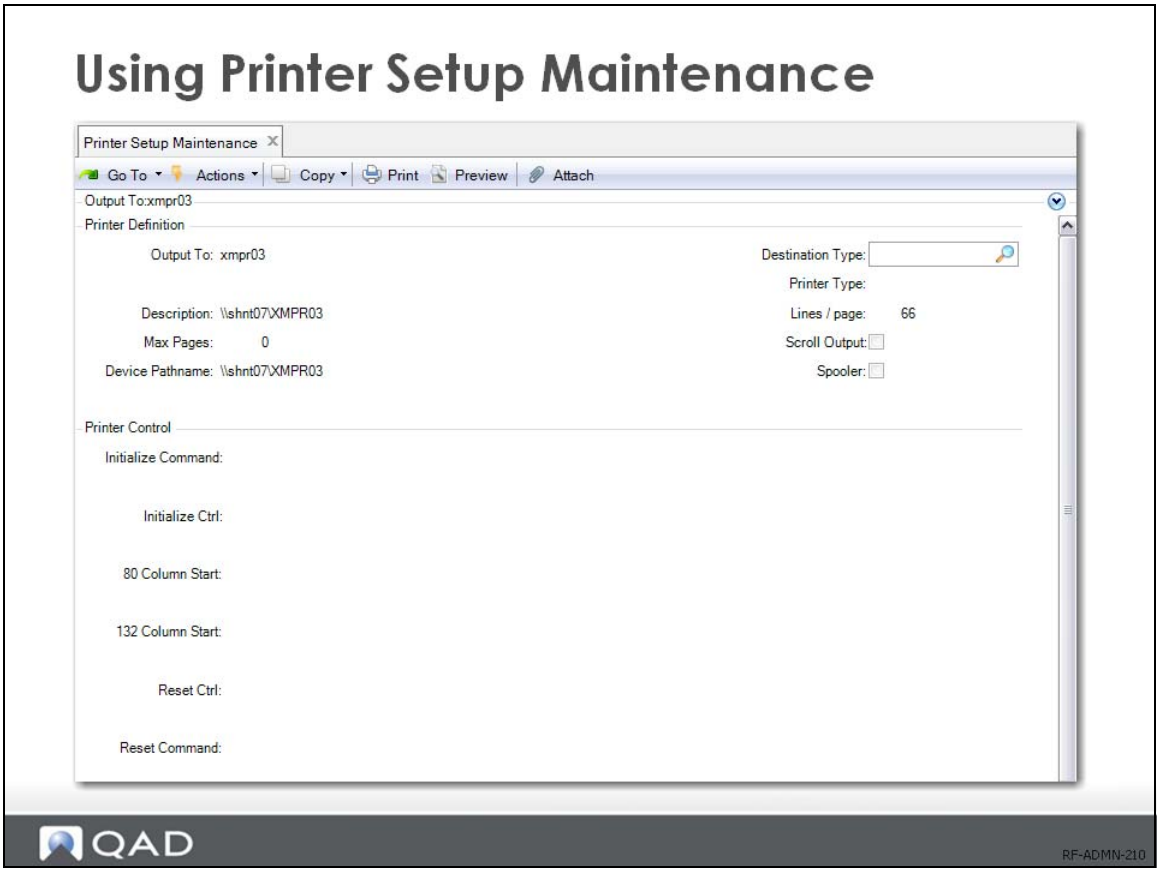
Scheduled Report Administration: Configuring Windows Task Scheduler



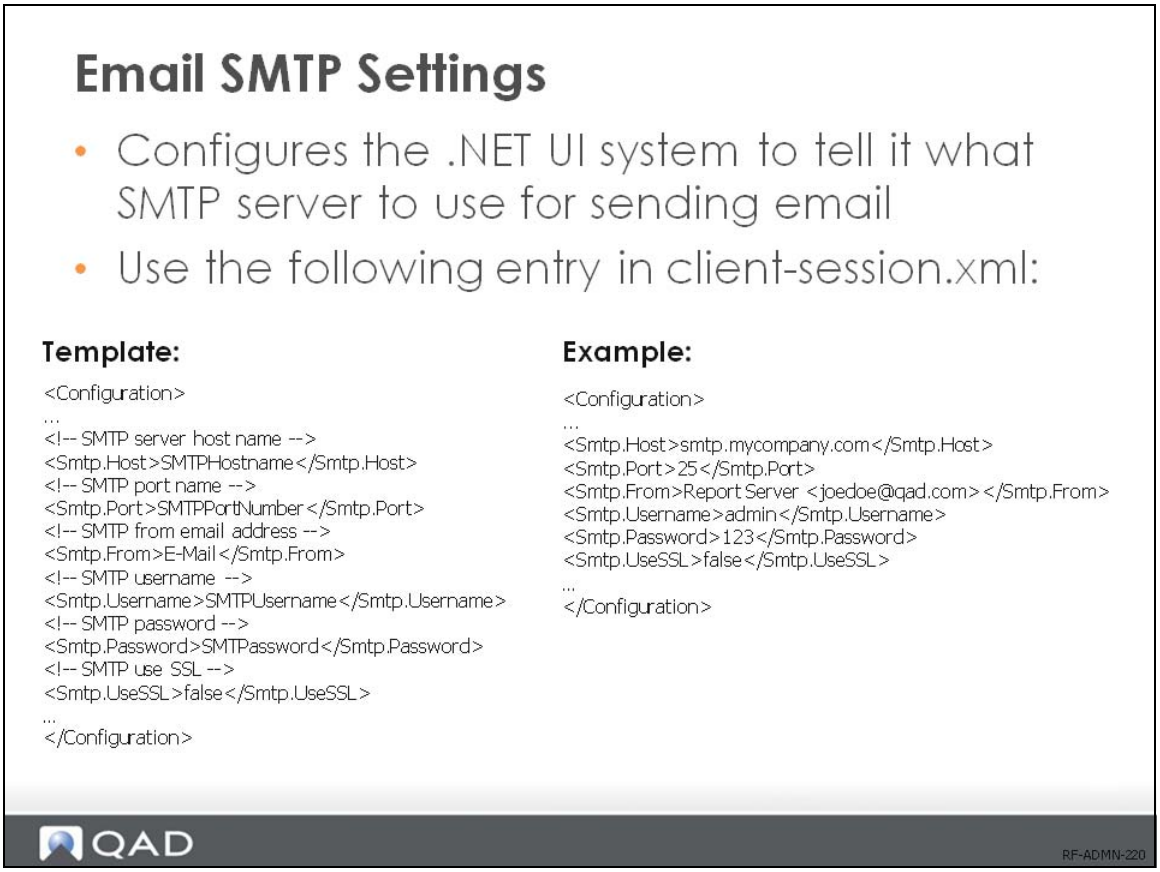
Printer Setup



Using Printer Setup Maintenance



Email SMTP Settings



Email Template File

Available dynamic variables to use in the email template:

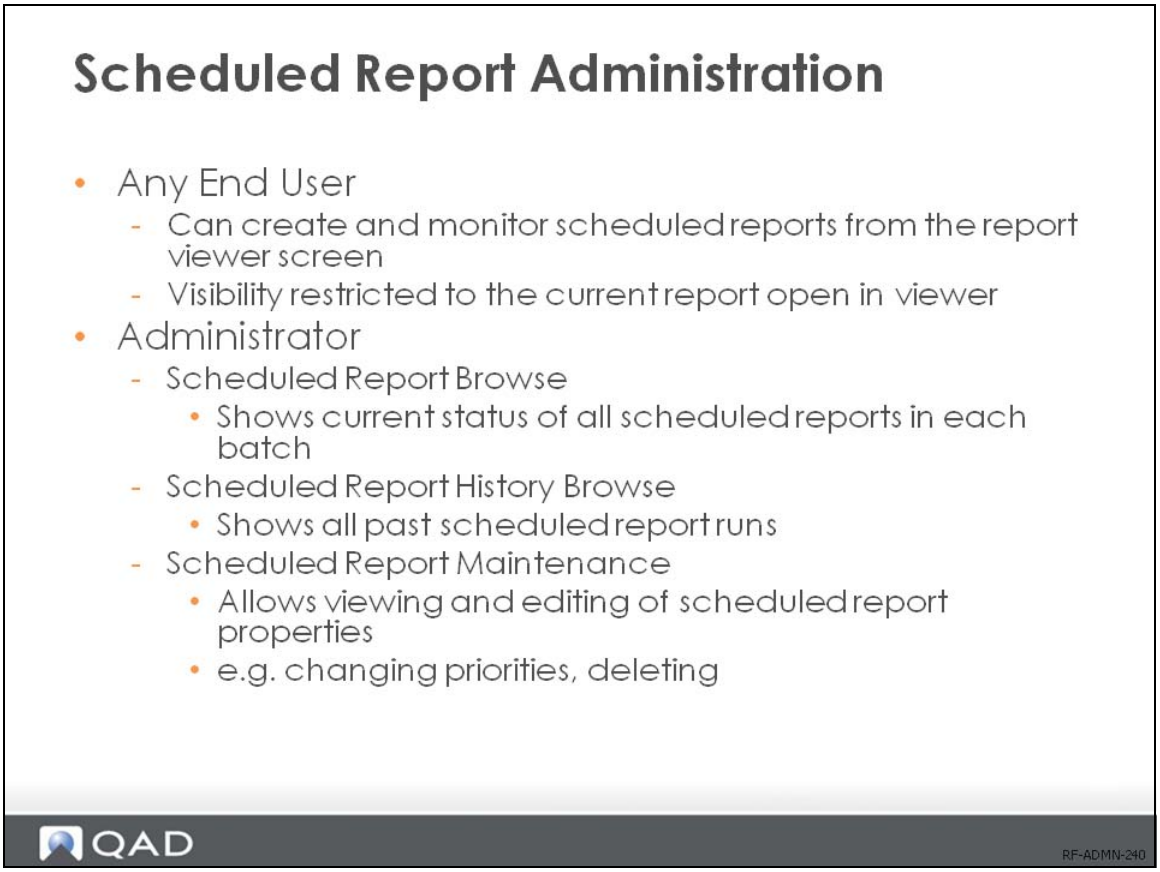
${RRO_CODE} : Report Code (as entered in Report Resource Maintenance).

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

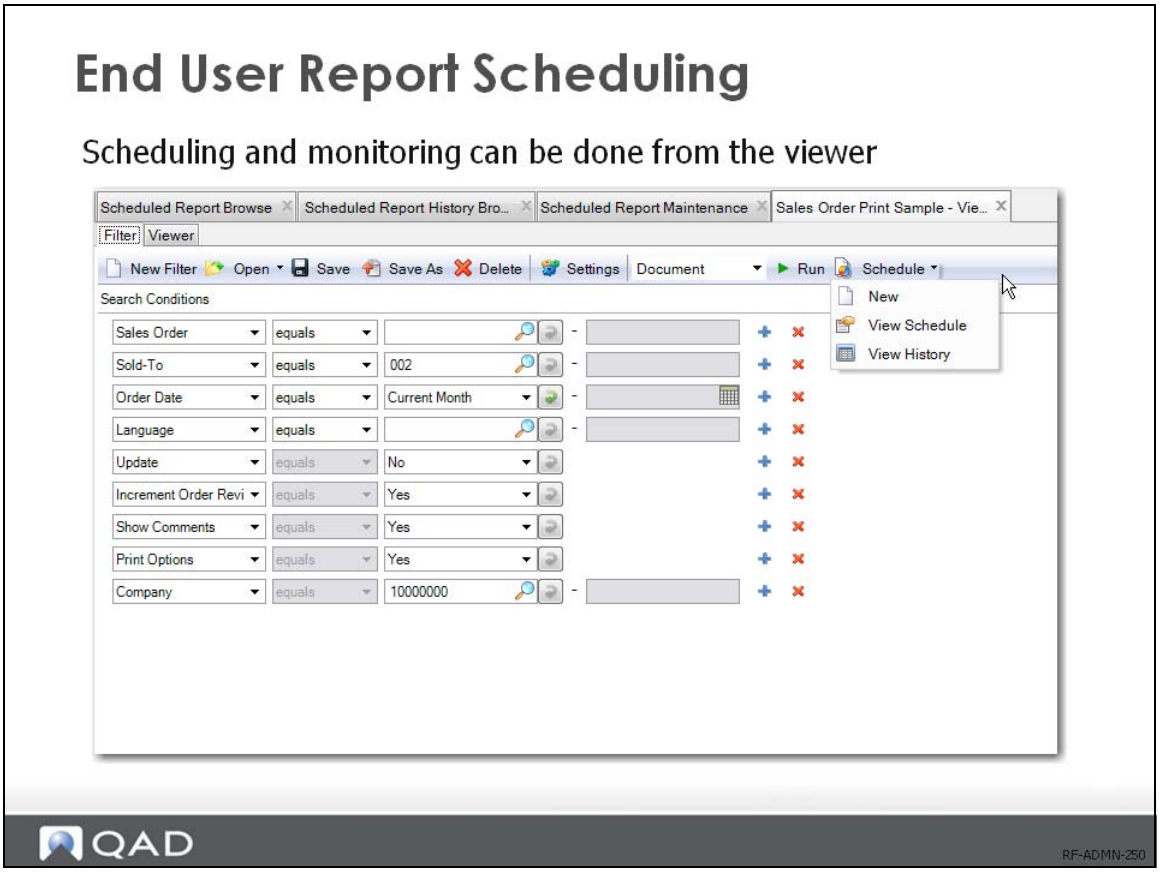
${SR_DESC} : Scheduled Report Description (as entered in the New Schedule Report form).

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

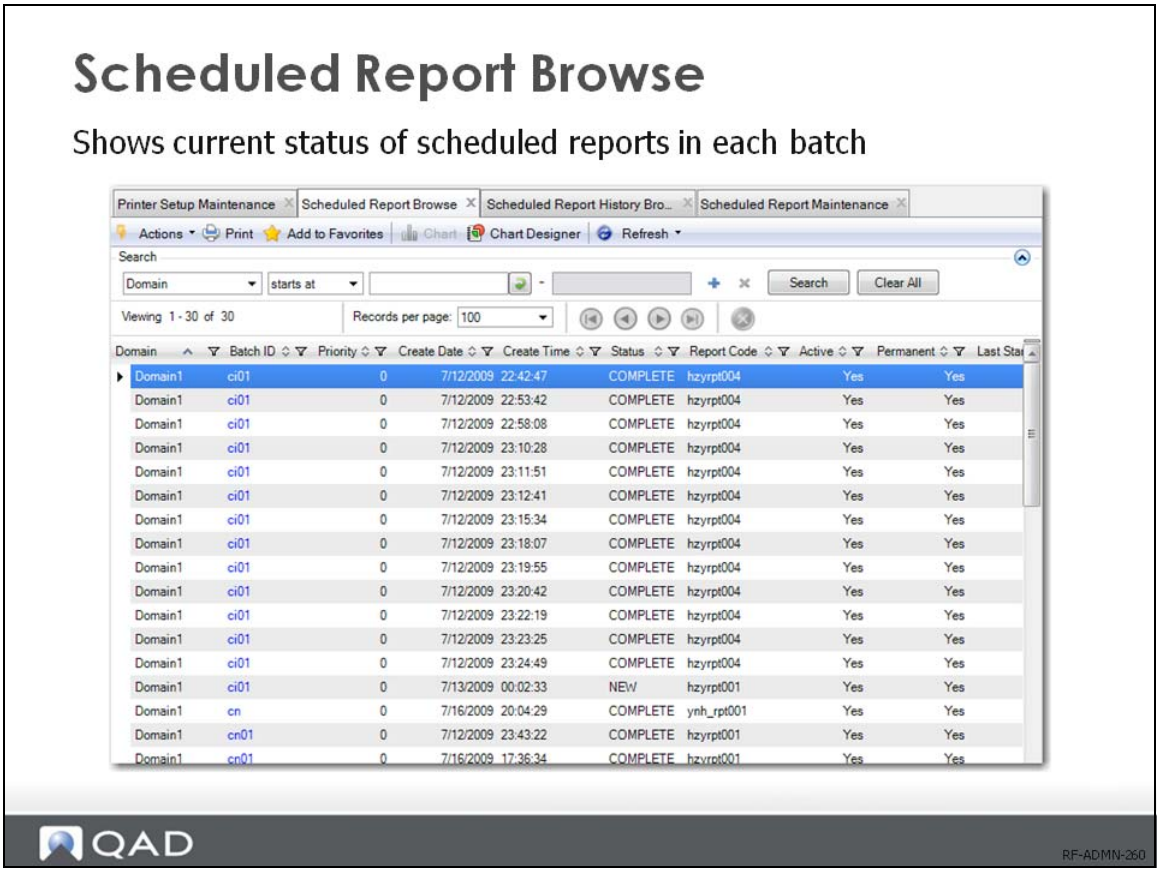
Scheduled Report Administration



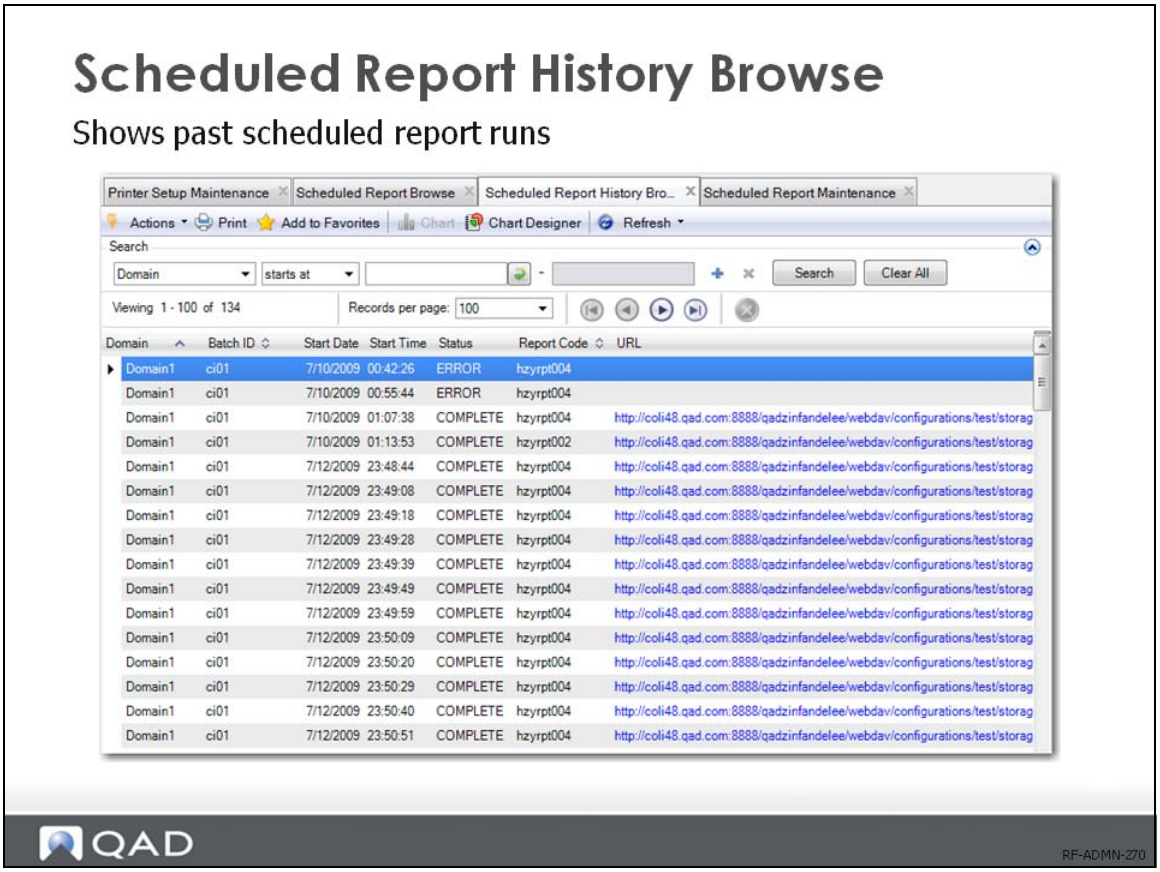
End User Report Scheduling



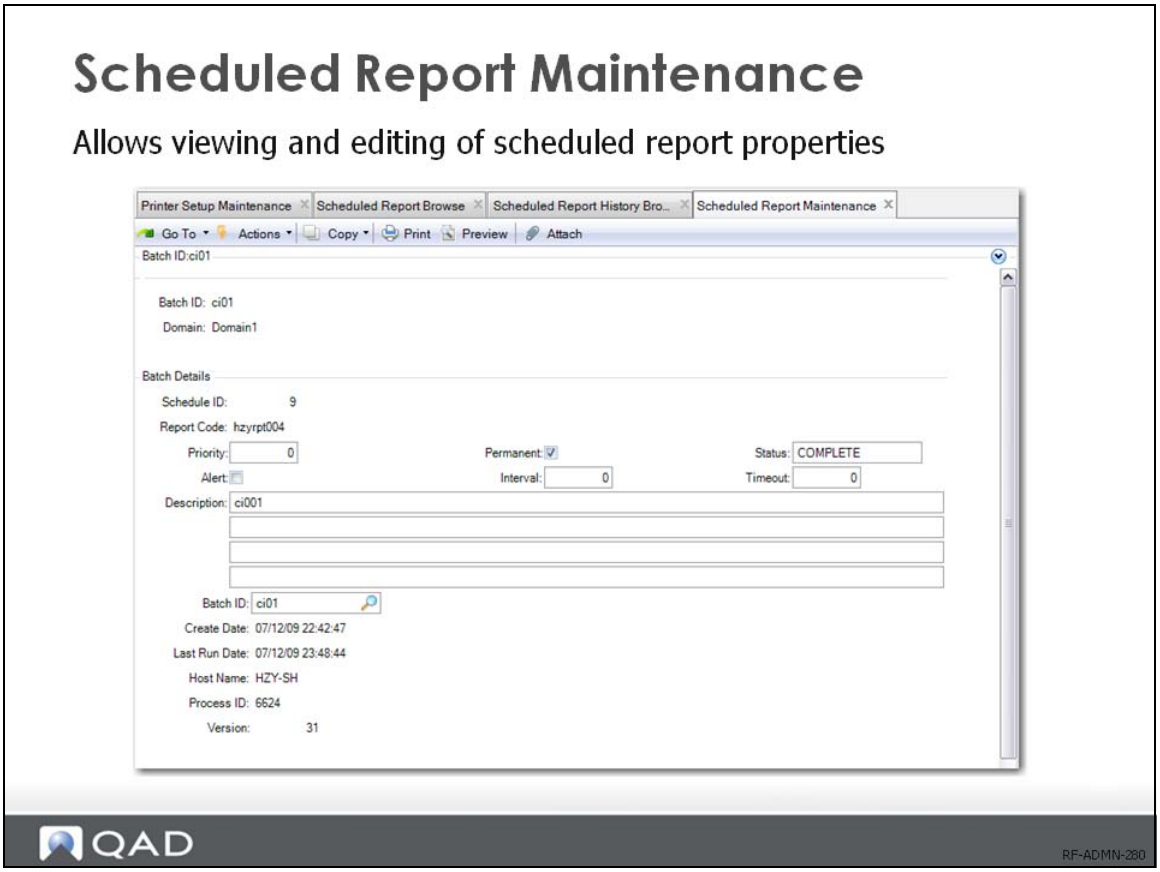
Scheduled Report Browse



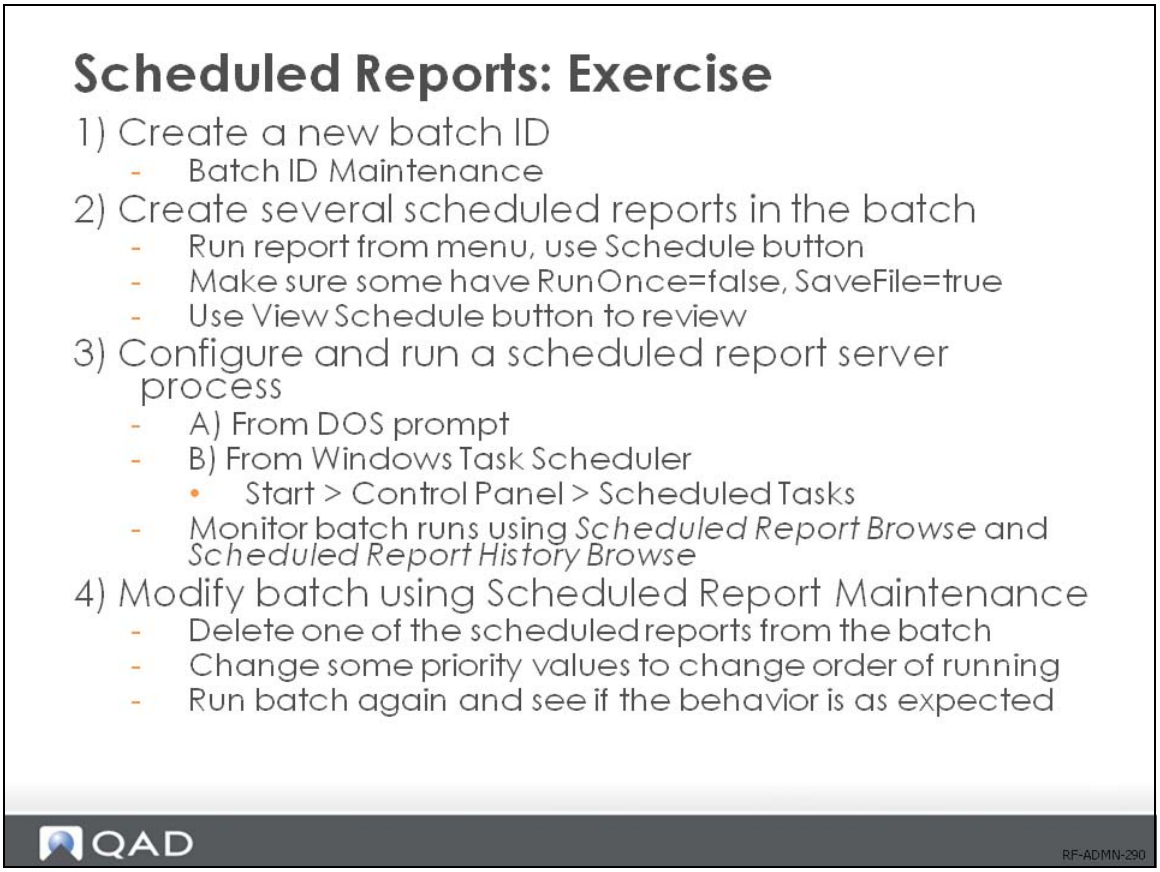
Scheduled Report History Browse



Scheduled Report Maintenance

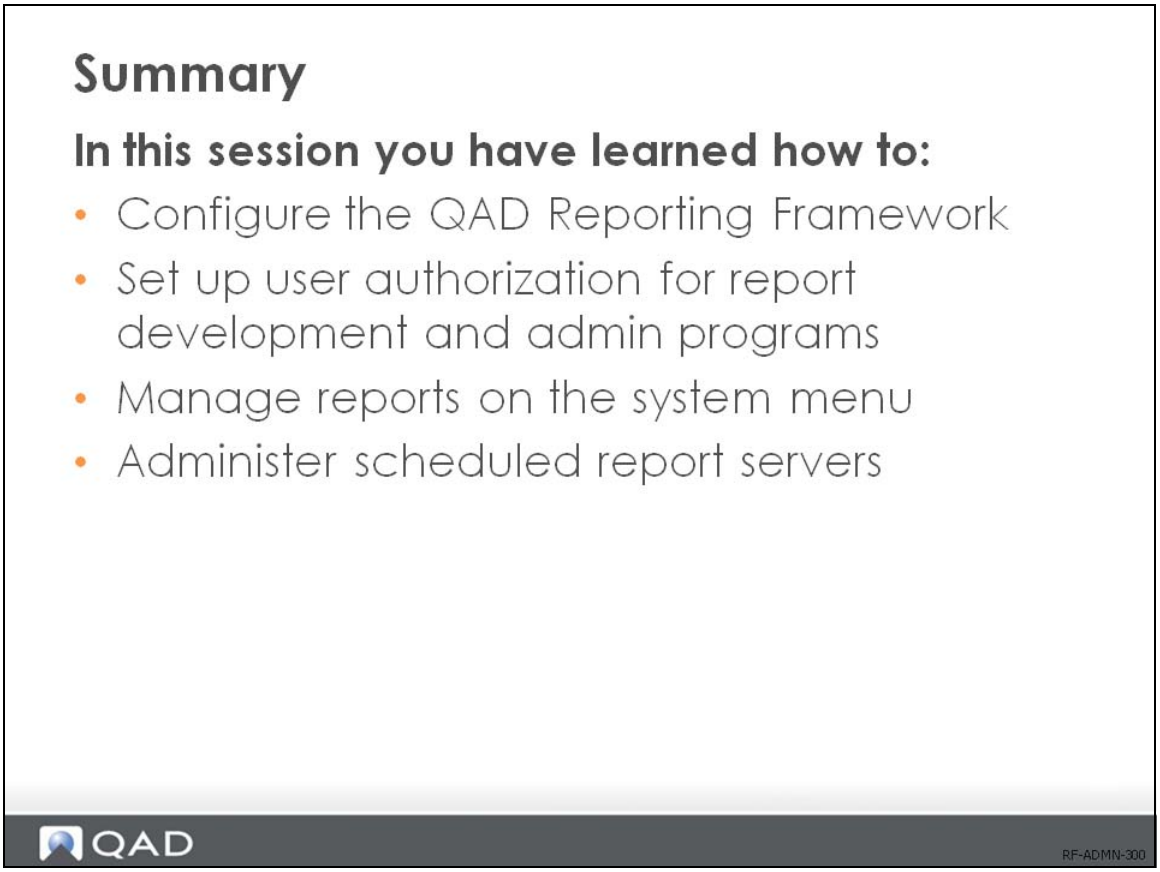


Scheduled Reports: Exercise



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. 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



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