

Power BI  
Paginated Reports  
in a Day

December 2022 release

Instructor Demos

# Overview

This document provides high-level instructions on how to setup and deliver the instructor demos.

The demos allow you to present a story that is in sync with the learning objects delivered in the course modules. They provide you with a minimum set of demos. You can modify or augment the demos, but take care not to deviate from the scripted demo story, otherwise later demos may fail.

To prepare for delivering the instructor demos, you should:

* Complete all the course labs.
* Practice the demos. If possible, commit them to memory to demonstrate during class.
* Make your own notes to supplement the demo instructions.
* Complete the Instructor Setup prior to class.

Solution files, if needed, are available in the <CourseFolder>\PowerBIPRIAD\Demos\Solutions folder. But the data sources will not connect to your database instance, so you will need to update the connection properties.

# Instructor Setup

Complete the setup instructions prior to class.

If you need any assistance with the setup of the tenant, student accounts, or the database, please contact [pbipartnerevents@microsoft.com](mailto:pbipartnerevents@microsoft.com).

## Setup tenant and student accounts

Please refer to the CDX setup for creating a test environment. As well students may choose to use their own power bi accounts.

Setup database

Important: Prior to class starting, you must test that you and your students can connect to the Azure SQL Database. Connection issues may arise in some regions or because firewall rules restrict accessing the database. If necessary, you can complete the steps in this setup task to restore the database locally to SQL Server or to an Azure SQL Database instance that you can control.

1. If necessary, restore the <CourseFolder>\PowerBIPRIAD\Demo\Setup\AdventureWorksDW2021-PRIAD.bak database backup to Azure or SQL Server.
2. Grant db\_datareader role permissions to a user named readonlyuser.
3. If restoring as an Azure SQL Database, create a [server-level firewall rule](https://docs.microsoft.com/azure/azure-sql/database/firewall-create-server-level-portal-quickstart) allowing all client IP addresses (0.0.0.0 to 255.255.255.255) to access the Azure SQL server instance.
4. Open your <CourseFolder>\PowerBIPRIAD\MySolution\MyEnvironment.txt file, and replace the server name value with the new server name.
5. Leave the MyEnvironment.txt file open.
6. In the presentation, unhide slide the Labs\Database server name slide.

When presenting this slide, you can open your MyEnvironment.txt file to allow students to copy the server name into their MyEnvironment.txt files.

## Setup your Power BI environment

1. Sign in to Power BI using the trial tenant admin account
2. Create any type of workspace, the most basic being a Pro workspace.
3. In the workspace, use “get data” to upload the <CourseFolder>\PowerBIPRIAD\Lab01A\Assets\Sales Analysis.pbix file.

It is the same file that students will upload in their first lab.

1. Remove the Sales Analysis.pbix dashboard (if it was automatically created).
2. Open Power BI Report Builder, and then sign in to Power BI using the account provided to you.
3. Open the <CourseFolder>\PowerBIPRIAD\Demo\Setup\Sales Order.rdl report.
4. Edit the data source connection properties to connect to your AdventureWorksDW2021-PRIAD database.
5. Test the data source connection.
6. Publish the report to the workspace as Sales Order.
7. Close Power BI Report Builder.
8. In the Power BI service, apply the credentials to the Sales Order report.
9. Test that you can open the Sales Analysis report and the Sales Order paginated report.
10. Return to the workspace home page, ready to commence the first demonstration.

# Demo 01A

In this demo, you will demo a Power BI report.

1. In the Power BI service, open the Sales Analysis report.
2. Describe that it is a Power BI report, primarily designed for interactive analysis.
3. To demonstrate interactivity:

* In the Fiscal Year slicer, select a different year.
* Click a column to cross-filter the page.

1. Right-click the “2021 Jun” column and select the drill-through option of “Sales Orders”.
2. Explain that this page displays all the sales orders for June 2021.
3. Now right-click sales order **SO71779** and choose the drill-through page called **Sales Order Detail**
4. Explain that this page breaks down all the line items for that individual sales order chosen. In a later lab, we will update this Power BI report to use a paginated report.
5. Use the menu to export the report to PDF.
6. Open the PDF document and point out that the table is a snapshot of an image, and so it is not possible to see all order lines.

# Demo 01B

In this demo, you will demo a Power BI paginated report.

1. In the Power BI service, open the Sales Order report.
2. Describe that it is a Power BI paginated report, designed to print a single sales order.
3. Input **SO71779** in the parameter and explain that it presents the same sales order number as the previous demo.
4. On the View menu, open page view.
5. Navigate between all pages of the report, and point out the page number details in the page footer.

In the series of instructor demos, you will produce a similar sales order report.

# Demo 02A

In this demo, you will create a Wizard-designed report.

1. Open Power BI Report Builder.
2. In the Getting Started window, select the Table or Matrix Wizard.
3. In the first Wizard step, create a new data source:

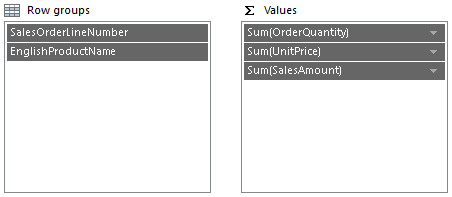
* Name: AdventureWorksDW
* Connection type: Microsoft Azure SQL Database
* Connection details: [Copy from your MyEnvironment.txt file]

1. In the second Wizard step, design the query:

* Expand the dbo folder, and then the Tables folder.
* Expand the FactResellerSales table, and then check the SalesOrderLineNumber, OrderQuantity, UnitPrice, and SalesAmount columns.
* Expand the DimProduct table, and then check the EnglishProductName column.
* Apply a filter to the FactResellerSales table SalesOrderNumber column, for the SO51721 (SO is an abbreviation for “Sales Order”—“O” is the letter O.)
* Check the filter parameter checkbox.
* Run the query to preview the result.

1. In the third Wizard step, drag the available fields as follows:

* Row groups: SalesOrderLineNumber, and EnglishProductName
* Values: OrderQuantity, UnitPrice, and SalesAmount



1. In the Choose the Layout step, uncheck the Show subtotals and grand totals checkbox.
2. Uncheck the Expand/collapse groups checkbox.



1. Complete the Wizard process.
2. In the title text box, enter an expression to return the parameter value:
   1. VB.NET
   2. =Parameters!SalesOrderNumber.Value
3. Preview the report.

Explain that the Wizard can produce a quick result, but it’s not always what you want. You can either modify the Wizard-designed report, or as they will learn in this course, they can design a report from scratch.

You will not develop this report any further.

1. Close Power BI Report Builder without saving the report.

# Demo 03A

In this demo, you will create a data source and dataset.

1. Open Power BI Report Builder.
2. In the Getting Started window, click Blank Report.
3. Save the report as Sales Order to your <CourseFolder>\PowerBIPRIAD\Demo\MySolution folder.
4. Create a data source:

* Name: AdventureWorksDW
* Connection type: Microsoft Azure SQL Database
* Connection details: [Copy from your MyEnvironment.txt file]

1. Create a dataset named dsMain based on the AdventureWorksDW data source.
2. Use the query designer to create the same query defined in Demo 02A, except do not parameterize the filter.

You will parameterize the dataset in Demo 04A.

* Expand the dbo folder, and then the Tables folder.
* Expand the FactResellerSales table, and then check the SalesOrderLineNumber, OrderQuantity, UnitPrice, and SalesAmount columns.
* Expand the DimProduct table, and then check the EnglishProductName column.
* Apply a filter to the FactResellerSales table SalesOrderNumber column, for the SO51721 (SO is an abbreviation for “Sales Order”—“O” is the letter O.)

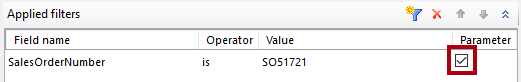
1. Save the report, and leave it open ready for the next demo.

You will design the report by adding a Table data region in Demo 05A.

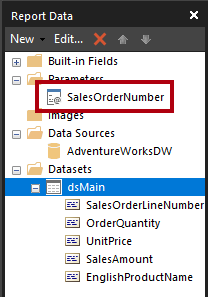
# Demo 04A

In this demo, you will parameterize the report.

1. Continue with the report design from Demo 03A.
2. Explain that presently the dataset retrieves all sales order lines. Yet, the objective is to retrieve lines for a single sales order.
3. Open the dsMain dataset properties, and then open the query designer.
4. In the query designer, in the Applied Filters section, check the Parameter checkbox.



1. Close the query designer.
2. In the dataset properties, point out the query statement and the parameterized WHERE clause.
3. Click OK to close the query designer.
4. In the Report Data pane, expand the Parameters folder and reveal the new report parameter.



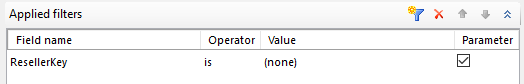
1. Open the SalesOrderNumber report parameter properties, and show and describe the General, Available Values, and Default Values pages.
2. Run the report.
3. Describe that the report now runs for a single sales order, currently defaulting to the order number used to filter the dataset query.

You will now add a new report parameter to filter sales orders by a single reseller.

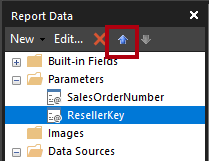
1. Return to the report design, create a new dataset named dsReseller based on the AdventureWorksDW data source.
2. Create a query based on the following two columns of the DimReseller table.

* ResellerKey
* ResellerName

1. In the dataset properties window, modify the query statement to add an ORDER BY clause:
   1. T-SQL
   2. SELECT  
       DimReseller.ResellerKey  
       ,DimReseller.ResellerName  
      FROM  
       DimReseller  
      **ORDER BY  
       ResellerName**
2. Create a new dataset named dsSalesOrderNumber based on the AdventureWorksDW data source.
3. Create a query based on the SalesOrderNumber column of the FactResellerSales table.
4. Add an applied filter to filter the query by the ResellerKey column—do not apply a value.
5. Parameterize the applied filter.

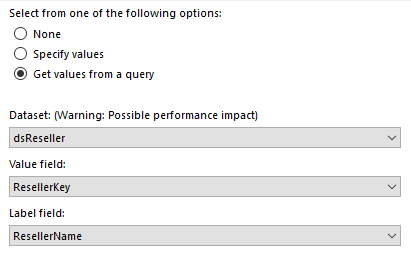


1. In the dataset properties window, modify the query statement to add the DISTINCT operator, and an ORDER BY clause:
   1. T-SQL
   2. SELECT **DISTINCT** FactResellerSales.SalesOrderNumber  
      FROM  
       FactResellerSales  
      WHERE  
       FactResellerSales.ResellerKey = @ResellerKey  
      **ORDER BY  
       SalesOrderNumber**
2. In the Report Data pane, select the ResellerKey report parameter and then click the up icon to move it to become the first in the list.



1. Configure the properties of the ResellerKey report parameter:

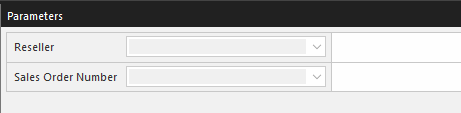
* Prompt: Reseller
* Available values: Get values from a query, using the dsReseller dataset, with the value field set to the ResellerKey field, and the label field set to the ResellerName field.



1. Configure the properties of the SalesOrderNumber report parameter:

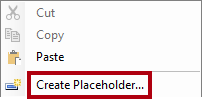
* Available values: Get values from a query, using the dsSalesOrderNumber dataset, with the value field set to the SalesOrderNumber field, and the label field set to the SalesOrderNumber field.
* Default values: No default value

1. In the Parameters pane, move the Sales Order Number parameter beneath the Reseller parameter.



You will now commence designing the report layout.

1. Add a report header.
2. Move the title text box to the top-left of the report header.
3. In the title text box, enter Sales Order, followed by a space.
4. Right-click after the space, and select Create Placeholder.



1. In the Placeholder Properties window, click the fx button for the Value property.
2. In the Expression window, in the Category list, select Parameters.
3. Double-click the SalesOrderNumber parameter to add it to the expression.
4. Run the report.
5. Select different resellers, and show the list of sales orders for the selected reseller.
6. View the report to show how the selected sales order number appears in the report title.
7. Return to the report designer.
8. Save the report, and leave it open ready for the next demo.

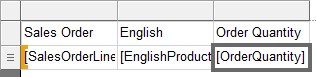
You will design the report by adding a Table data region in Demo 05A.

# Demo 05A

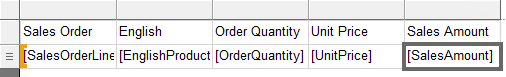
In this demo, you will add a Table data region.

1. Continue with the report design from Demo 04A.
2. Add a Table data region to the top-left corner of the report body.
3. Use the Field picker to add the following dsMain fields to the table details row for the first three columns:

* SalesOrderLineNumber
* EnglishProductName
* OrderQuantity



1. Right-click the third column guide to insert a new column.
2. Select the UnitPrice field for the new column.
3. Add another column for the SalesAmount field.



1. Right-click the Sales Amount detail row cell, and then select Add Total.
2. In the Row Groups pane, edit the (Details) group, and sort by the SalesOrderLineNumber field.
3. Consider applying the following formats to the Table data region:

* Widen the second column to display the product name without word wrap.
* Rename the columns as Line, Product, Quantity, Price, and Amount
* Format the cells of the table header row using a contrasting background color and bold font.
* Format the Price and Amount columns using currency with two decimal places

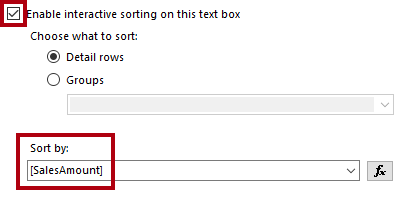
1. Remove any trailing space from the body.
2. Run the report for any reseller and sales order number.
3. Save the report, and leave it open ready for the next demo.

You will add interactive sorting to the Table data region in Demo 06A.

# Demo 06A

In this demo, you will add interactive sorting to the Table data region.

1. Continue with the report design from Demo 05A.
2. Edit the text box properties for the Amount text box.
3. In the Interactive Sorting page, enable interactive sort, and sort by the SalesAmount field.



1. Run the report for any reseller and sales order number.
2. Sort the Amount column in ascending order and then in descending order.
3. Save the report.

This demo completes the series instructor demos.

# Terms of use

© 2022 Microsoft. All rights reserved.

By using this hands-on lab, you agree to the following terms:

The technology/functionality described in this hands-on lab is provided by Microsoft Corporation in a “sandbox” testing environment for purposes of obtaining your feedback and to provide you with a learning experience. You may only use the hands-on lab to evaluate such technology features and functionality and provide feedback to Microsoft. You may not use it for any other purpose. Without written permission, you may not modify, copy, distribute, transmit, display, perform, reproduce, publish, license, create derivative works from, transfer, or sell this hands-on lab or any portion thereof.

COPYING OR REPRODUCTION OF THE HANDS-ON LAB (OR ANY PORTION OF IT) TO ANY OTHER SERVER OR LOCATION FOR FURTHER REPRODUCTION OR REDISTRIBUTION WITHOUT WRITTEN PERMISSION IS EXPRESSLY PROHIBITED.

THIS HANDS-ON LAB PROVIDES CERTAIN SOFTWARE TECHNOLOGY/PRODUCT FEATURES AND FUNCTIONALITY, INCLUDING POTENTIAL NEW FEATURES AND CONCEPTS, IN A SIMULATED ENVIRONMENT WITHOUT COMPLEX SET-UP OR INSTALLATION FOR THE PURPOSE DESCRIBED ABOVE. THE TECHNOLOGY/CONCEPTS REPRESENTED IN THIS HANDS-ON LAB MAY NOT REPRESENT FULL FEATURE FUNCTIONALITY AND MAY NOT WORK THE WAY A FINAL VERSION MAY WORK. WE ALSO MAY NOT RELEASE A FINAL VERSION OF SUCH FEATURES OR CONCEPTS. YOUR EXPERIENCE WITH USING SUCH FEATURES AND FUNCITONALITY IN A PHYSICAL ENVIRONMENT MAY ALSO BE DIFFERENT.

FEEDBACK If you give feedback about the technology features, functionality and/or concepts described in this hands-on lab to Microsoft, you give to Microsoft, without charge, the right to use, share and commercialize your feedback in any way and for any purpose. You also give to third parties, without charge, any patent rights needed for their products, technologies and services to use or interface with any specific parts of a Microsoft software or service that includes the feedback. You will not give feedback that is subject to a license that requires Microsoft to license its software or documentation to third parties because we include your feedback in them. These rights survive this agreement.

MICROSOFT CORPORATION HEREBY DISCLAIMS ALL WARRANTIES AND CONDITIONS WITH REGARD TO THE HANDS-ON LAB, INCLUDING ALL WARRANTIES AND CONDITIONS OF MERCHANTABILITY, WHETHER EXPRESS, IMPLIED OR STATUTORY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. MICROSOFT DOES NOT MAKE ANY ASSURANCES OR REPRESENTATIONS WITH REGARD TO THE ACCURACY OF THE RESULTS, OUTPUT THAT DERIVES FROM USE OF THE VIRTUAL LAB, OR SUITABILITY OF THE INFORMATION CONTAINED IN THE VIRTUAL LAB FOR ANY PURPOSE.

DISCLAIMER This lab contains only a portion of new features and enhancements in Microsoft Power BI. Some of the features might change in future releases of the product.