



# Pro Power BI Architecture

Development, Deployment, Sharing, and  
Security for Microsoft Power BI Solutions

—  
*Second Edition*

—  
Reza Rad

Apress®

# **Pro Power BI Architecture**

Development, Deployment,  
Sharing, and Security for  
Microsoft Power BI Solutions

Second Edition



**Reza Rad**

**Apress®**

# ***Pro Power BI Architecture: Development, Deployment, Sharing, and Security for Microsoft Power BI Solutions***

Reza Rad  
Auckland, New Zealand

ISBN-13 (pbk): 978-1-4842-9537-3  
<https://doi.org/10.1007/978-1-4842-9538-0>

ISBN-13 (electronic): 978-1-4842-9538-0

Copyright © 2023 by Reza Rad

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr  
Acquisitions Editor: Jonathan Gennick  
Development Editor: Laura Berendson  
Editorial Project Manager: Shaul Elson  
Copy Editor: Kezia Endsley

Cover image by wal\_172619 from Pixabay

Distributed to the book trade worldwide by Springer Science+Business Media New York, 1 New York Plaza, Suite 4600, New York, NY 10004-1562, USA. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail [orders-ny@springer-sbm.com](mailto:orders-ny@springer-sbm.com), or visit [www.springeronline.com](http://www.springeronline.com). Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail [booktranslations@springernature.com](mailto:booktranslations@springernature.com); for reprint, paperback, or audio rights, please e-mail [bookpermissions@springernature.com](mailto:bookpermissions@springernature.com).

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <https://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at [www.apress.com/](http://www.apress.com/). For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

*To Leila*

*Who showed me how strong a human being can be*

# Table of Contents

<b>About the Author .....</b>	<b>xxix</b>
<b>About the Technical Reviewers .....</b>	<b>xxxii</b>
<b>Acknowledgments .....</b>	<b>xxxiii</b>
<b>Introduction .....</b>	<b>xxxv</b>
<b>■ Part I: Getting Started.....</b>	<b>1</b>
<b>■ Chapter 1: Power BI Components.....</b>	<b>3</b>
What Is Power BI? .....	4
The Power BI Desktop .....	4
Power Query for Data Transformation .....	5
The Power BI Report.....	6
Analytical Engine (Tabular Engine) .....	7
Datasets .....	8
The Power BI Dashboard .....	8
Dataflows .....	9
Datamarts.....	10
The DAX Language .....	11
The Power BI Service for Hosting Reports .....	12
The Power BI Mobile App .....	13
The Power BI Report Server .....	14
Paginated Reports .....	15
The Power BI Report Builder .....	16

■ TABLE OF CONTENTS

Metrics and Scorecards .....	17
The On-Premises Gateway .....	18
Workspaces.....	18
Power BI App .....	19
Power BI Premium.....	20
Power BI Embedded .....	20
Developer API or REST API.....	20
Summary.....	21
<b>Chapter 2: Tools and Preparation .....</b>	<b>23</b>
Using the Power BI Desktop .....	23
The Power BI Desktop: Report Authoring Tool .....	24
The Power BI Desktop from the Microsoft Store .....	24
The Power BI Desktop as a Separate Download Link.....	26
Auto-Update from the Microsoft Store App .....	26
Flexibility on Installation: The Download Link.....	27
Development Work.....	27
Setting Up Power BI Accounts .....	27
Creating Power BI Accounts from Office 365 .....	28
Is Your Account Pro, Free, or PPU?.....	29
Installing Power BI Mobile App.....	30
Power BI Report Server .....	30
On-Premises Gateway .....	30
Microsoft SQL Server Database and Management Tools .....	30
The Power BI Helper.....	30
Downloading Dataset Files.....	30
Summary.....	30

<b>Part II: Development.....</b>	<b>31</b>
<b>Chapter 3: Import Data or Scheduled Refresh.....</b>	<b>33</b>
The Connection Type Is Not the Data Source Type .....	33
Import Data or Scheduled Refresh .....	33
A Closer Look at Import Data.....	33
Loading Data to the Model .....	35
How Power BI Stores Data in Memory .....	36
How xVelocity Compresses Data.....	36
Where Data Is Stored in Memory .....	38
How About Power BI Service? .....	38
Is There a Limitation on Size? .....	39
Combining Data Sources; Power Query Unleashed.....	40
DAX: A Powerful Analytical Expression Language .....	43
Publishing a Report.....	44
Gateway Configuration .....	44
Scheduled Refresh .....	46
Advantages and Disadvantages of Import Data .....	49
Advantages of Import Data .....	49
Disadvantages of Import Data .....	49
When Should You Use Import Data? .....	50
Summary.....	50
<b>Chapter 4: DirectQuery .....</b>	<b>51</b>
What Is DirectQuery?.....	51
Which Data Sources Support DirectQuery?.....	51
How to Use DirectQuery.....	52
No Data Tab in DirectQuery Mode.....	54
How DirectQuery Works .....	55
Performance.....	58
Performance Tuning on the Data Source .....	58

## ■ TABLE OF CONTENTS

Query Reduction .....	60
Maximum Connections Per Data Source .....	62
Adding Extra Data Sources: Composite Mode .....	64
Limited Power Query .....	65
Limited Modeling and DAX .....	66
No Refresh Needed .....	66
Large Scale Dataset .....	66
Summary .....	67
<b>■ Chapter 5: Live Connection .....</b>	<b>69</b>
What Is Live Connection? .....	69
Create a Report with Live Connection .....	69
Live Connection Behind the Scenes .....	72
Performance .....	73
Single Data Source .....	74
No Power Query Transformations .....	74
Modeling and Report-Level Measure .....	76
Publishing the Report and Gateway Configuration .....	78
Live Connection to Power BI Service .....	83
How Live Connection Differs from DirectQuery .....	84
Summary .....	84
<b>■ Chapter 6: Composite Model .....</b>	<b>87</b>
What Is Composite Model? .....	87
Why Use Composite Model? .....	88
How Does Composite Model Work? .....	90
Dual Storage Mode .....	93
Modeling with the Power BI Composite Model .....	95
Relationships in the Power BI Composite Model .....	95
Calculations and DAX .....	96
DirectQuery to Power BI Datasets .....	96
Summary .....	97

<b>■ Chapter 7: DirectQuery for Power BI Datasets and Analysis Services .....</b>	<b>99</b>
Power BI with Import Data .....	99
One Important Problem with Import Data.....	99
Power BI Using Live Connection.....	100
Big Limitation of Live Connection.....	100
DirectQuery for Power BI Datasets or Analysis Services Datasets.....	101
Why Is This Feature a Big Deal? .....	101
How It Works Under the Hood.....	102
Not from My Workspace .....	102
Creating a DirectQuery to Power BI Dataset.....	103
DirectQuery to Power BI Dataset vs Live Connection to Power BI Dataset .....	106
Multiple DirectQuery Connections Are Supported .....	107
Creating a Relationship Between Different Data Sources .....	107
Publish to Service.....	108
Lineage View .....	113
Sharing the Report.....	114
Building Access to the Source Power BI Datasets.....	114
Summary.....	116
<b>■ Chapter 8: Choosing the Right Connection Type: DirectQuery, Live, Import, or Composite Model .....</b>	<b>117</b>
Why Is This a Tough Decision? .....	117
What Is Import Data (Scheduled Refresh)? .....	118
Compression Engine of xVelocity.....	118
Important Pros and Cons of This Method.....	118
What Is DirectQuery?.....	119
Important Pros and Cons of This Method.....	119
What Is Live Connection? .....	121
Important Pros and Cons of This Method.....	121
Composite Model: The Best of Both Worlds.....	122
Differences Between Live Connection and DirectQuery.....	123
Relationship Configuration .....	123

## ■ TABLE OF CONTENTS

Report-Level Measures .....	124
No Power Query in Live Connection .....	124
<b>Pros and Cons of Each Method .....</b>	<b>124</b>
Import Data or Scheduled Refresh .....	124
DirectQuery.....	124
Live Connection .....	124
Which Method Is the Best and Fastest? .....	125
Which Method Is More Flexible?.....	125
Which Method Is More Scalable? .....	125
<b>Which Architecture Scenarios Are Best for Each Method?.....</b>	<b>125</b>
Import Data for Agility and Performance .....	125
Live Connection for an Enterprise Solution .....	126
Live Connection for Team Development .....	126
Direct Query for Non-Microsoft Sources .....	126
Summary.....	128
<b>■ Chapter 9: Dataflows .....</b>	<b>129</b>
<b>What Is a Dataflow? .....</b>	<b>129</b>
A Dataflow Is Service-Only (Cloud-Only) Object.....	130
A Dataflow Is Not Just for Power BI.....	130
<b>Where Do Dataflows Store Data? .....</b>	<b>131</b>
Standard vs. Analytical Dataflows .....	132
<b>Dataflows Are Powered by Power Query Online.....</b>	<b>133</b>
<b>Dataflows Can Be Used in Power BI, Excel, and Other Services .....</b>	<b>135</b>
<b>Example Dataflow Scenarios.....</b>	<b>136</b>
Using One Power Query Table in Multiple Power BI Reports .....	136
Different Data Sources with Different Refresh Schedules.....	138
Centralized Data Warehouse.....	139
<b>Getting Started with Dataflows in Power BI .....</b>	<b>140</b>
<b>Administrator's Control .....</b>	<b>141</b>
<b>Creating Your First Dataflow .....</b>	<b>142</b>
Sample Datasets.....	145

Power Query Editor Online.....	146
Setting Up a Gateway .....	147
Blank Query as the Data Source.....	148
Scheduled Refreshes.....	150
Get Data from Dataflow in the Power BI Desktop.....	151
Get Data from Dataflows Using Excel .....	154
<b>Summary.....</b>	<b>155</b>
<b>■ Chapter 10: Shared Datasets.....</b>	<b>157</b>
What Is the Dataset in Power BI? .....	157
What Is Included in the Dataset?.....	159
What Is a Shared Dataset? .....	159
How Do You Create a Shared Dataset?.....	162
Sharing Datasets Across Multiple Workspaces .....	162
External Dataset: How Does Shared Dataset Work Behind the Scenes? .....	163
Why Use Shared Datasets? .....	163
Shared Datasets in the Power BI Architecture.....	164
Endorsement: Certified and Promoted Datasets.....	165
Summary.....	168
<b>■ Chapter 11: Datamarts .....</b>	<b>169</b>
Power BI for the Citizen Data Analyst.....	169
Governance and Reusability.....	169
What are Datamarts in Power BI? .....	170
Who Are Datamarts for? .....	172
Power BI Datamart for the Citizen Data Analyst .....	173
Power BI Datamart for Enterprises .....	173
Power BI Datamart for Developers .....	173
What Are the Features of Datamarts that Empower Power BI Development?.....	173
Datamarts Complete the BI Ecosystem .....	173
One Place to Manage and Build, No Other Tools Needed.....	174
One License Is Enough .....	175

## ■ TABLE OF CONTENTS

The Vast Horizon and the Future of Datamarts .....	175
Governance.....	175
<b>Getting Started with Power BI Datamarts .....</b>	<b>175</b>
Premium Workspace.....	175
Creating a Datamart .....	177
Get Data: Power Query Online .....	179
The Datamart Editor.....	182
<b>Power BI Datamart Components .....</b>	<b>190</b>
Power BI Datamarts Under the Hood .....	190
<b>Summary.....</b>	<b>202</b>
<b>■ Chapter 12: The Multi-Layer Architecture .....</b>	<b>203</b>
Challenges of Using a Single PBIX File for Everything .....	203
Using Dataflows to Decouple the Data Preparation Layer.....	204
Shared Datasets for the Data Modeling Layer.....	206
Thin Power BI Reports, Paginated Reports, and the Analyze in Excel Option.....	206
Power BI Architecture with Dataflows, Shared Datasets, and Thin Reports.....	208
Benefits of a Multi-Layered Architecture.....	209
Enhancing the Architecture Using Power BI Datamarts.....	210
Layered Architecture for Dataflow .....	211
Chained Datasets.....	214
What About Enterprise Architecture?.....	215
Summary.....	216
<b>■ Chapter 13: Datamarts vs. Dataflows vs. Datasets .....</b>	<b>217</b>
What Is a Dataflow? .....	217
What Is a Dataset? .....	218
What Is a Datamart? .....	218
Differences Between Dataflows and Datasets .....	219
Dataflows vs. Datasets .....	224

What About Datamarts? .....	225
Will Dataflows Be Replaced by Datamarts?.....	226
Are Datasets Being Replaced by Datamarts? .....	227
Summary.....	228
<b>■ Chapter 14: Paginated Reports.....</b>	<b>229</b>
What Is a Paginated Report in Power BI?.....	229
Why Paginated Reports? .....	231
Differences Between Power BI Reports and Paginated Reports .....	233
Do You Need a Paginated Report?.....	234
Architecture Best Practice: One Dataset to Rule Them All .....	235
The Power BI Report Builder .....	235
Summary.....	240
<b>■ Chapter 15: Excel and Power BI Integration.....</b>	<b>241</b>
The Analyze in Excel Feature.....	241
The Analyze in Excel Feature from the Power BI Service .....	241
Implicit Measures Don't Work in Excel .....	244
Excel Is Connected Live to the Power BI Model in the Service.....	247
Getting Data from Power BI Dataset.....	248
Getting Data from a Power BI Dataflow .....	251
Why is Analyze in Excel Better than Using Export Data? .....	252
Publish to Power BI from Excel .....	254
Upload Your Workbook to Power BI .....	255
Export Workbook Data to Power BI .....	256
Import Excel into the Power BI Desktop.....	256
Summary.....	257
<b>■ Chapter 16: Real-Time Streaming Datasets in Power BI .....</b>	<b>259</b>
Real-time vs. Live or DirectQuery.....	259
Sample Scenario .....	260
Capturing Inputs with a Microsoft Form.....	260

## ■ TABLE OF CONTENTS

<b>The Power BI Streaming Dataset .....</b>	<b>261</b>
<b>Power Automate Pushes Data to the Power BI Dataset .....</b>	<b>265</b>
<b>The Power BI Real-Time Dashboard.....</b>	<b>268</b>
<b>Streaming Dataset Types.....</b>	<b>269</b>
Push Datasets.....	269
Streaming Datasets .....	269
Hybrid Datasets .....	269
<b>Streaming Services .....</b>	<b>270</b>
Power BI Streaming Dataset and REST API .....	270
Azure Stream Analytics.....	270
PubNub .....	270
<b>Creating Calculations on Real-Time Datasets .....</b>	<b>270</b>
Writing Calculations Using Q&A.....	270
Live Connection to a Streaming Dataset .....	274
Summary.....	278
<b>■ Chapter 17: Performance Tuning with Aggregations.....</b>	<b>279</b>
<b>How Does Aggregation Work? .....</b>	<b>280</b>
Aggregated Table .....	281
Layers of Aggregation.....	282
Step 1: Create the Aggregated Table .....	283
Dual Storage Mode; The Most Important Configuration for Aggregations! .....	293
Step 2: Power BI Aggregations .....	293
Step 3: Configure Aggregation Functions and Test Aggregations in Action .....	310
Multiple Layers of Aggregations .....	318
<b>Aggregation for Imported Data.....</b>	<b>328</b>
DAX Measures Instead of Manage Aggregation.....	331
<b>Automatic Aggregation.....</b>	<b>336</b>
Summary.....	337

<b>■ Chapter 18: Big Data with Incremental Refresh and Hybrid Tables.....</b>	<b>339</b>
What Is Incremental Refresh? .....	339
Hybrid Tables.....	339
Configuring Incremental Refresh .....	340
Setting Up Incremental Refresh .....	340
Parameters in Power Query.....	340
Filter Data Based on the Parameters.....	342
Power BI Desktop Incremental Refresh Setup.....	344
Hybrid Table Setup.....	347
Incremental Refresh for Multiple Tables.....	348
Publish to Service.....	349
What Do the Partitions Look Like? .....	350
Detecting Data Changes.....	354
Only Refresh When Period Is Complete .....	355
Incremental Refresh for Dataflows or Datamarts.....	356
Summary.....	357
<b>■ Chapter 19: Development Best Practices .....</b>	<b>359</b>
Reuse Tables Generated in Power Query .....	359
Reuse DAX Calculations .....	359
Use Power Query Functions for Reusable Data Transformation.....	360
Parametrize the Data Transformation Process Using Power Query Parameters .....	360
Categorize Power Query Using Power Query Groups (or Folders).....	360
Disable the Load of Temp Tables in Power Query.....	361
Only Load What You Need for Reporting: Filter the Data .....	361
Use Reference and Duplicate in Their Rightful Places .....	361
Multi-Layer Architecture Everywhere.....	361
Sync Slicers.....	362
Use a Theme File .....	362
Use a Background Image for Report Pages.....	362

## ■ TABLE OF CONTENTS

Incorporate Conditional Formatting Using DAX, Parameters, and Tables .....	362
Create a Measure Table.....	363
Create and Use Hierarchies.....	363
Set Auto Summarization at the Field Level .....	363
Consider Using a Custom Date Table.....	363
Design Mobile Reports .....	364
Use Aggregations for Big Tables.....	364
Use Incremental Refresh and Hybrid Tables When Possible .....	364
Design Star-Schema Models .....	365
Avoid Both-Directional Relationships .....	365
Clean Up Your Models.....	365
<b>■ Part III: Deployment and Collaboration.....</b>	<b>367</b>
<b>■ Chapter 20: The Power BI Service .....</b>	<b>369</b>
The Power BI Desktop: A Report Authoring Tool .....	369
Hosting Options for Power BI Reports .....	370
What Is the Power BI Service? .....	371
Publish Reports to the Service .....	372
Capacity-Based or User-Based.....	373
The Power BI Service .....	374
Template Apps .....	375
Apps.....	375
Workspaces .....	376
Dataflows.....	377
Shared Datasets .....	378
Datamarts .....	378
Deployment Pipelines.....	379
Dashboards.....	380
Metrics.....	380

Content Certification.....	381
Data Lineage.....	382
Summary.....	382
<b>■ Chapter 21: The Power BI Report Server.....</b>	<b>383</b>
What Is the Power BI Report Server? .....	383
Requirements for Setting It Up .....	384
Installing the Power BI Report Server .....	385
Configuring the Power BI Report Server .....	390
Database Set Up .....	392
Web URL Set Up.....	396
Installing the Power BI Desktop Report Server .....	402
Developing Reports with the Power BI Report Server.....	403
Publish Report to the Report Server .....	405
Managing Datasets on the Report Server .....	407
Schedule Refresh Requirement.....	409
Pros and Cons of the Report Server .....	412
No Gateway Is Needed.....	412
All Types of Connections Are Supported .....	413
The Power BI Report Server Is a Fully On-Premises Solution.....	413
The Power BI Service Features Are Not Available.....	413
Licensing the Report Server .....	413
Summary.....	414
<b>■ Chapter 22: Power BI Gateways .....</b>	<b>415</b>
What Is a Gateway?.....	415
Do You Always Need a Gateway?.....	415
Types of Gateways.....	415
Gateway Execution Flow.....	417
Important Points to Consider Before Installing a Gateway .....	418
How Many Gateways Are Required?.....	418
Installing Gateways .....	419

■ TABLE OF CONTENTS

<b>Adding Data Sources .....</b>	<b>424</b>
Set Up a Gateway Connection to the Dataset .....	428
<b>Users and Access Controls for Gateways .....</b>	<b>429</b>
Gateway Installers .....	429
Gateway Users.....	430
Data Source Users .....	431
Additional Settings for Gateway .....	432
<b>Troubleshooting.....</b>	<b>435</b>
<b>Summary.....</b>	<b>436</b>
<b>■ Chapter 23: Power BI Licensing .....</b>	<b>437</b>
<b>Two Types of Licensing.....</b>	<b>437</b>
The Power BI Desktop .....	438
Why Do I Need a License if the Power BI Desktop Is Free?.....	438
<b>User-Based Licensing.....</b>	<b>439</b>
Power BI Free .....	439
Power BI Pro .....	439
Power BI PPU: Premium Per User .....	441
<b>Capacity-Based Licensing.....</b>	<b>442</b>
Power BI Embedded .....	442
Power BI Premium.....	444
The Difference Between A SKUs and P/EM SKUs .....	446
<b>SQL Server Enterprise Edition Plus Software Assurance .....</b>	<b>447</b>
<b>Summary.....</b>	<b>449</b>
<b>■ Chapter 24: Power BI Admin Portal and Tenant Settings .....</b>	<b>451</b>
<b>Power BI Administrator .....</b>	<b>451</b>
<b>Tenant Settings .....</b>	<b>452</b>
Help and Support Settings.....	453
Workspace Settings.....	455
Information Protection.....	455
Export and Sharing Settings.....	456

Content Pack and App Settings .....	459
Integration Settings .....	460
Custom Visual Settings .....	462
Audit and Usage Settings .....	463
Developer Settings .....	464
Dataflow, Template Apps, and Datamarts .....	464
<b>Usage Metrics .....</b>	<b>465</b>
Embed Codes .....	465
Organization Visuals .....	466
Workspaces .....	468
Custom Branding .....	468
Summary .....	469
<b>■ Chapter 25: PowerShell Cmdlets for Power BI .....</b>	<b>471</b>
PowerShell .....	471
PowerShell Cmdlets for Power BI .....	471
Getting Started .....	472
Example: Export a List of All Workspaces in the Organization as a CSV File .....	478
Example: Deploy a Power BI Report from One Workspace to Another .....	479
Are PowerShell Cmdlets the Same as REST APIs in Power BI? .....	483
Summary .....	483
<b>■ Chapter 26: Power BI REST API .....</b>	<b>485</b>
The Power BI REST API .....	485
Getting Started .....	485
Calling the API from .NET .....	490
Embedding into a Custom Application .....	492
Power Automate .....	493
PowerShell to Call REST API .....	493
Summary .....	493

■ TABLE OF CONTENTS

<b>■ Chapter 27: Power BI Audit Log for the Tenant .....</b>	<b>495</b>
The Problem .....	495
The Audit Log .....	498
Using PowerShell to Extract the Audit Log .....	502
The Power BI Report for the Audit Log .....	505
Summary .....	512
<b>■ Chapter 28: XMLA Endpoint .....</b>	<b>513</b>
Behind the Scenes of a Power BI Dataset.....	513
SSAS Is More Than What You See .....	514
Can I Access the SSAS Model of My Power BI Dataset? .....	516
XMLA Endpoint .....	517
How Does an XMLA Connection Work?.....	518
The XMLA Endpoint URL .....	519
Admin Control under Capacity Configuration.....	519
Client Tools .....	520
Sample Connection.....	521
The Power BI Helper .....	522
Summary .....	522
<b>■ Chapter 29: Dashboard and Report Sharing .....</b>	<b>523</b>
Power BI Content Owner .....	523
How Does Dashboard Sharing Work?.....	523
Three Levels of Access.....	528
Manage Permissions .....	528
Advantages of Dashboard Sharing .....	532
Disadvantages of Dashboard Sharing .....	532
No Edit Access.....	533
Share Objects One at a Time .....	533
Licensing for a Large User Base.....	533
Summary .....	533

<b>■ Chapter 30: Power BI Workspaces .....</b>	<b>535</b>
What Are Power BI Workspaces? .....	535
Creating Workspaces.....	536
Adding Content to a Workspace .....	541
Four Levels of Workspace Access .....	543
Tenant Admin Control for Workspaces .....	544
Advantages of Workspaces .....	547
Disadvantages of Workspaces .....	547
Summary.....	548
<b>■ Chapter 31: Power BI Apps.....</b>	<b>549</b>
Power BI Apps .....	549
App Workspaces .....	549
Creating an App .....	551
Setup .....	552
Content .....	554
Audience.....	556
Getting Apps .....	559
Making Changes to an App.....	561
Isolated Environments for Developers and Users.....	562
What About Data Refresh?.....	564
Advantages of Power BI Apps .....	564
Cons of Power BI Apps .....	565
Summary.....	565
<b>■ Chapter 32: Publish to Web .....</b>	<b>567</b>
What Is Publish to Web?.....	567
Security Issues with Publish to Web .....	570
Removing Access .....	572

■ TABLE OF CONTENTS

<b>Central Monitoring for all Embed Codes.....</b>	<b>574</b>
Who Can Publish Reports on the Web?.....	574
<b>Differences with Other Sharing Methods .....</b>	<b>575</b>
Summary.....	578
<b>■ Chapter 33: Power BI Embedded .....</b>	<b>579</b>
<b>Using Power BI Embedded .....</b>	<b>579</b>
Myths about Power BI Embedded.....	580
<b>What Is Power BI Embedded? .....</b>	<b>580</b>
Embedded Playground.....	580
<b>Licensing Power BI Embedded.....</b>	<b>582</b>
Some SKUs Can Be Turned On and Off .....	583
Power BI Users or Custom Application Users .....	583
<b>Implementing Power BI Embedded .....</b>	<b>584</b>
Summary.....	585
<b>■ Chapter 34: Secure Embed .....</b>	<b>587</b>
<b>Why Another Embedding Method? .....</b>	<b>587</b>
Publish to Web.....	587
Power BI Embedded .....	588
Secure Embed: Good Features from Both Worlds!.....	589
<b>Using Secure Embed .....</b>	<b>589</b>
Who Can View the Report? .....	590
Licensing Needs .....	593
<b>Advantages of Secure Embed .....</b>	<b>593</b>
<b>Limitations.....</b>	<b>594</b>
<b>Scenarios for Using this Method .....</b>	<b>594</b>
Summary.....	595

<b>Chapter 35: Embed in SharePoint Online and Teams.....</b>	<b>597</b>
How to Use Embed in SharePoint Online.....	597
Sharing a SharePoint Page with an Embedded Power BI Report.....	604
Access to SharePoint.....	604
Power BI Permissions.....	605
Advantages and Disadvantages of Embedding in SharePoint Online.....	608
Advantages of Embed in SharePoint Online .....	608
Disadvantages of Embed in SharePoint Online.....	609
Power BI and Teams.....	609
Power BI and PowerPoint.....	610
Summary.....	611
<b>Chapter 36: Comparison of Sharing Methods for Power BI.....</b>	<b>613</b>
Types of Sharing Methods .....	613
Basic Sharing for Dashboards and Reports.....	614
Advantages of Dashboard Sharing .....	615
Disadvantages of Dashboard Sharing.....	615
Workspaces.....	615
Advantages of Workspaces.....	616
Disadvantages of Workspaces.....	617
Power BI Apps .....	617
Advantages of Power BI Apps.....	618
Cons of Power BI Apps.....	619
The Publish to Web Option .....	619
Security Issues with Publish to Web.....	620
Embed in SharePoint Online.....	620
Advantages of Embed in SharePoint Online .....	621
Disadvantages of Embed in SharePoint Online.....	622
Power BI Embedded .....	622
Pros and Cons of Power BI Embedded .....	622

■ TABLE OF CONTENTS

<b>Secure Embed .....</b>	<b>623</b>
Advantages of Secure Embed.....	623
Disadvantages and Limitations of Secure Embed .....	624
<b>Summary.....</b>	<b>624</b>
Cheat Sheet for Choosing a Method .....	624
<b>■ Chapter 37: Types of Power BI Users.....</b>	<b>625</b>
The Different Types of Power BI Users .....	625
Access Control .....	626
Training.....	627
The Layers and User Categories Are Different in Each Organization.....	628
Users Can Move Between Layers .....	629
Summary.....	629
<b>■ Chapter 38: Best Practices for Setting Up Workspace Roles.....</b>	<b>631</b>
The Modern Power BI Workspace .....	631
Roles in the Workspace .....	632
Viewer.....	632
Contributor.....	633
Member .....	634
Admin .....	637
Best Practices for Each Role .....	638
Viewer Role: Use Power BI Apps Instead .....	638
Contributor Role: Developers Only .....	639
Member Role: Deployment Group Only .....	639
Admin Role: Admin Group .....	640
Use Groups, Not Individual Accounts .....	640
Summary.....	640

<b>■ Chapter 39: Build Access Level .....</b>	<b>641</b>
Build Access Example .....	641
Are Build and Edit the Same? A Row-Level Security Example .....	643
Providing Build But Not Edit Access .....	645
Manage Permissions: Add Build Access .....	645
Create an App with Build Access .....	646
Who Should Have Build Access? .....	648
Licensing .....	648
Summary .....	649
<b>■ Chapter 40: How to Organize Workspaces in a Power BI Environment.....</b>	<b>651</b>
Considering a Power BI Workspace as a Single Development-Sharing Unit.....	651
Separating Audiences Using Power BI Apps .....	651
Separating Developers with Multiple Workspaces .....	652
Split the Load, Use the Capacity.....	654
Sharing Workspaces Among Multiple Developers .....	654
Layers of Shared Workspaces .....	655
Separating the Environments .....	656
Summary .....	660
<b>■ Chapter 41: Content Certification and Endorsement .....</b>	<b>661</b>
The Importance of Endorsement .....	661
Levels of Endorsement .....	661
What Kinds of Content Can Be Endorsed? .....	663
Who Can Endorse It? .....	663
Dataset Discoverability .....	665
An Example of an Endorsement and Content Certification .....	666
Summary .....	668

■ TABLE OF CONTENTS

<b>■ Chapter 42: Deployment Pipelines .....</b>	<b>669</b>
Why Multiple Environments?.....	669
How Many Environments Do You Need?.....	672
What Is a Deployment Pipeline?.....	673
How Does a Deployment Pipeline Work?.....	674
Creating Deployment Pipelines.....	674
Assigning a Workspace.....	675
Comparing Content.....	676
Deploy.....	678
Rules and Connections .....	680
Automate the Deployment.....	680
What If There Is No Premium License? .....	681
Summary.....	681
<b>■ Chapter 43: Data-Level Security.....</b>	<b>683</b>
Introduction .....	683
Row-Level Security .....	683
Static Row-Level Security .....	684
Dynamic Row-Level Security.....	693
Column-Level Security .....	703
Object-Level Security .....	704
Page-Level Security.....	704
Summary.....	704
<b>■ Chapter 44: Power BI Helper .....</b>	<b>705</b>
Documenting a Power BI File and Report.....	705
Download and Install Power BI Helper for Free .....	705
Open Power BI Helper as an External Tool.....	705
Connect to Model.....	706
Visualization Information .....	707
Documentation .....	708

What Is Included in the Documentation? .....	710
Configurations .....	710
Exporting the Data in a Power BI Table .....	712
Connect to Model.....	712
Export the Data.....	713
Any Size, Any Table Type.....	716
Consider Using Analyze in Excel.....	717
Reducing the Size of Power BI File .....	717
Steps to Reduce File Size .....	718
Documenting Power BI Tenant Objects .....	722
Service Tenant Settings in Power BI Helper .....	723
Scan Service Objects.....	730
Configuring the Documentation's Output.....	732
Exporting the Power BI Audit Log .....	735
What Is the Power BI Audit Log?.....	735
Difficulties Exporting the Audit Log .....	735
A Simple Way to Export the Audit Log Without Limitations.....	736
Power BI File Cleanup in a Few Steps.....	739
Visualization Information.....	740
Hiding Technical Fields .....	743
Summary.....	745
<b>Index.....</b>	<b>747</b>

## CHAPTER 31



# Power BI Apps

You've learned previously about some ways to share content in Power BI, such as using workspaces and dashboard sharing. This chapter explains everything about Power BI apps, a mechanism to share the content in Power BI in a way that includes security and governance. When Power BI apps are used with workspaces, you can build an ultimate sharing strategy in your organization.

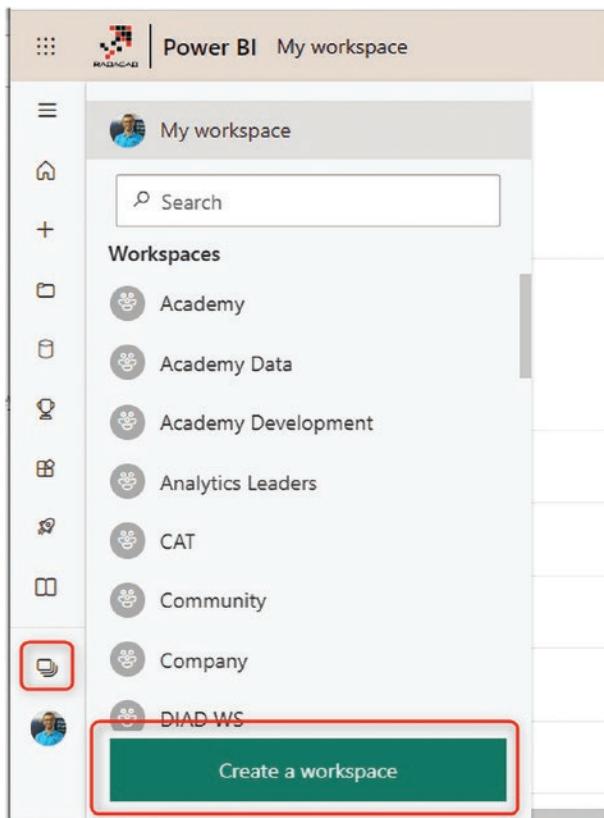
## Power BI Apps

Power BI apps differ from the mobile app for Power BI. Power BI apps share content with end users. You already learned about the limitations of dashboard sharing and workspaces. The Power BI App method provides an extensive approach to sharing content with end users. With Power BI apps, you can share content with end users without worrying about changing something in the development environment. Managing multiple environments is much easier with this approach. An app can be shared with a group of people or the entire organization.

In addition to separating the dev and user environments, apps come with another helpful point. Power BI apps can be shared with free users in the Power BI if the workspace in which the Power BI content is located is under a Premium capacity.

## App Workspaces

To start creating an app, you'll need a workspace (see Figure 31-1). Workspaces in Power BI are called app workspaces because you can create an app on top of a workspace (they are also called organization workspaces). The content that you have in this workspace can be selected and shared with users. In the previous chapter, you learned how to create and manage workspaces.



**Figure 31-1.** Creating workspaces in Power BI

An app workspace is like a shared folder with a group of people. Create a name for the app workspace. You can specify whether users can edit the workspace or have read-only access (there are four access levels). Note that this type of access is only for people you add to the workspace directly, not for the app's users. After creating the app workspace, you should first see a new workspace (folder) that is blank. You can publish the content to this workspace from the Power BI Desktop or create it from the Power BI Service.

Figure 31-2 shows an example of a workspace with some content in it.

The screenshot shows a Power BI workspace interface. At the top, there's a navigation bar with icons for Home, Recent, and Power BI. The workspace title is 'Power BI PW2'. Below the title, there's a toolbar with a camera icon, a 'Test' button, and dropdown menus for 'New' and 'Upload'. A sidebar on the left contains icons for Home, Recent, New, Upload, Content, Datasets + dataflows, and Datamarts (Preview). The main area displays a table of content items:

	Name	Type	Owner	Refreshed	Next refresh
	Datamart2	Datamart	Reza Rad	11/15/22, 2:15:38 PM	N/A
	Datamart3	Dataset (default)	Reza Rad	12/14/22, 7:22:08 PM	N/A
	Datamart3	Datamart	Reza Rad	—	N/A
	Difinity DM	Dataset (default)	Reza Rad	11/28/22, 3:52:01 PM	N/A
	Difinity DM	Datamart	Reza Rad	—	N/A
	Difinity report	Report	PW2	11/28/22, 3:52:01 PM	—
	DM Ukraine	Dataset (default)	Reza Rad	12/15/22, 7:41:02 AM	N/A
	DM Ukraine	Datamart	Reza Rad	—	N/A
	DM Ukraine Report 1	Report	PW2	12/15/22, 7:41:02 AM	—
	Incremental Refresh	Report	PW2	11/15/22, 4:05:13 PM	—
	Incremental Refresh	Dataset	Reza Rad	11/15/22, 4:05:13 PM	N/A
	Pubs 20220609	Report	PW2	11/15/22, 2:18:46 PM	—

**Figure 31-2.** A sample Power BI workspace

## Creating an App

The process of creating a Power BI app is also called publishing an app. To create the app, click the workspace in the Power BI Service, as shown in Figure 31-3.

All	Name	Type	Owner	Refreshed	Next refresh	Endorsement	Send
	Northwind	Dataset (default)	PPU WS	7/22/22, 4:28:08 PM	N/A	—	—
	Northwind	Datamart	Ressa Rad	—	N/A	—	—
	Northwind report	Report	PPU WS	7/22/22, 4:26:08 PM	—	—	—
	Reza Test 1 Viz	Report	PPU WS	7/22/22, 1:40:14 PM	—	—	—
	Reza Test 1 Viz	Report	PPU WS	7/22/22, 2:08:54 PM	—	—	—

**Figure 31-3.** Creating a Power BI app

## Setup

There are three stages for creating the app. In the first stage, you do the general setup by adding details such as the app's name, description, logo, and theme color. Choosing the theme color can help users distinguish the environment when they switch between apps. (A Power BI user can be the user of multiple apps.) You can also set the contact information.

There are three settings for the app that require attention:

- Install this app automatically. I recommend selecting this option. However, the tenant administrator in your organization has to enable this feature first. I explain this shortly.
- Hide app navigation pane. An app can come with a navigation pane. The navigation pane is helpful when sharing multiple reports and dashboards through this app. If you are sharing only a single report, the navigation pane is better hidden.
- Allow users to make a copy of the reports in this app. If you have power users and business analysts, you might want to give them access to create copies of the report (but not change the reports in this app).

Once you set the configurations, you can go to the next step (see Figure 31-4).

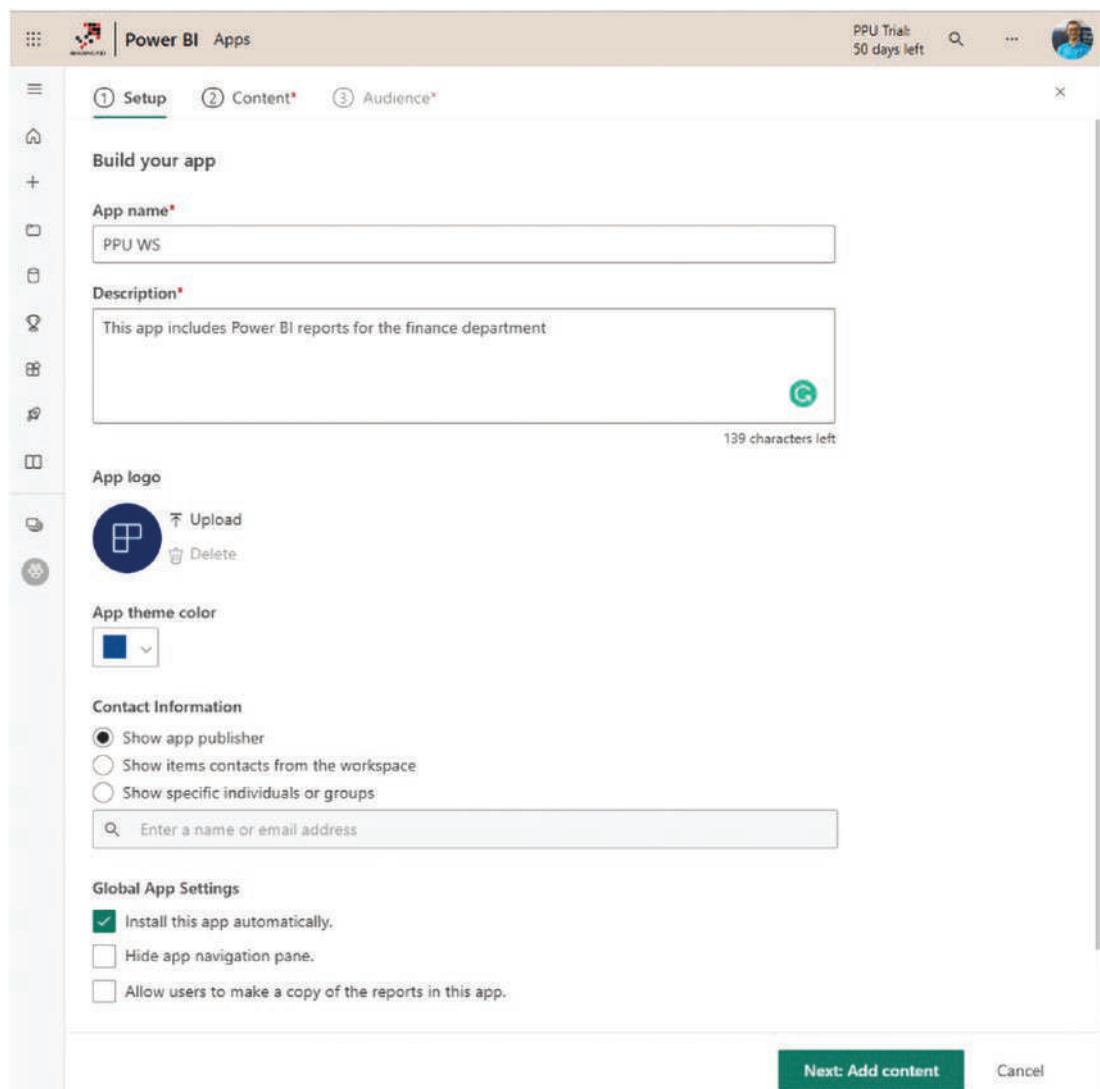
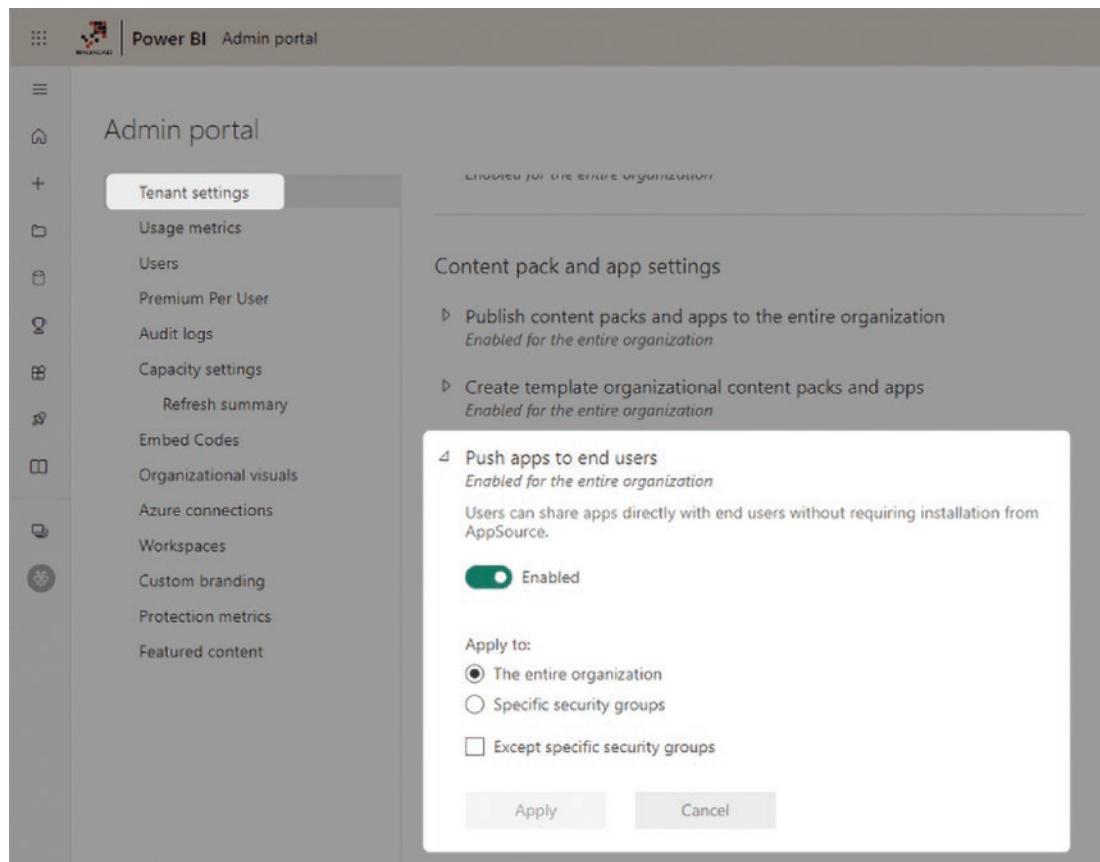


Figure 31-4. Setting up the Power BI App

## Installing Apps Automatically

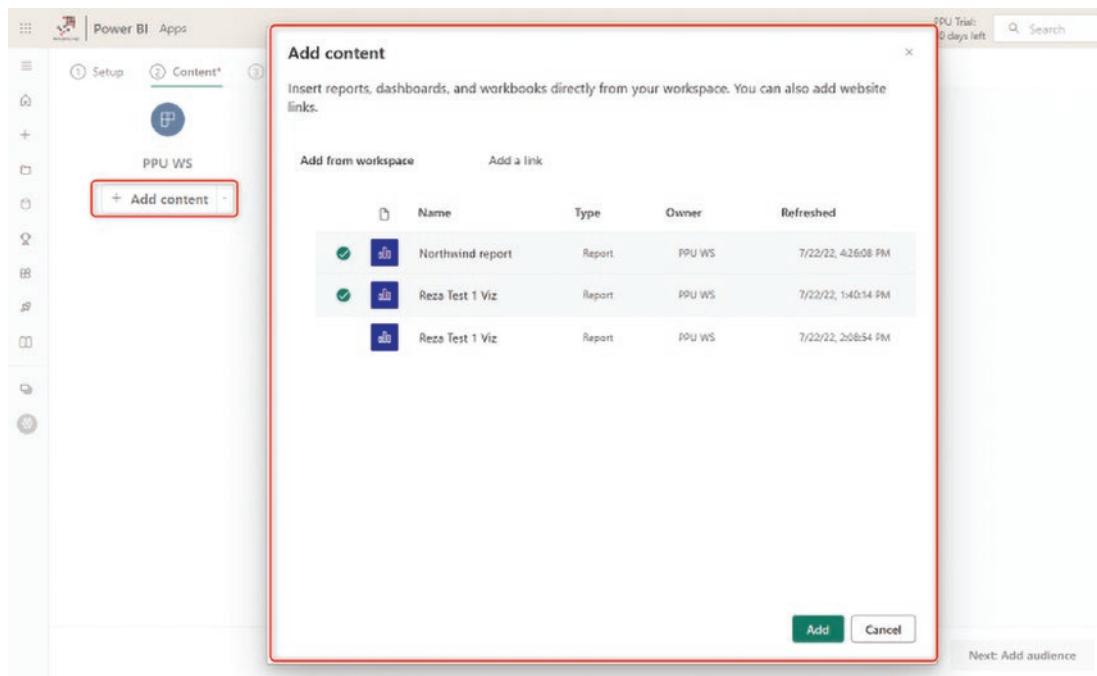
By default, the apps must be installed by the organization's users. This is an extra step that can be done automatically (which means that as soon as the app is created, it will be installed). The Power BI tenant administrator enables this automatic installation of the apps from the Tenant Settings area, as shown in Figure 31-5.



**Figure 31-5.** Pushing the Power BI app to the end users

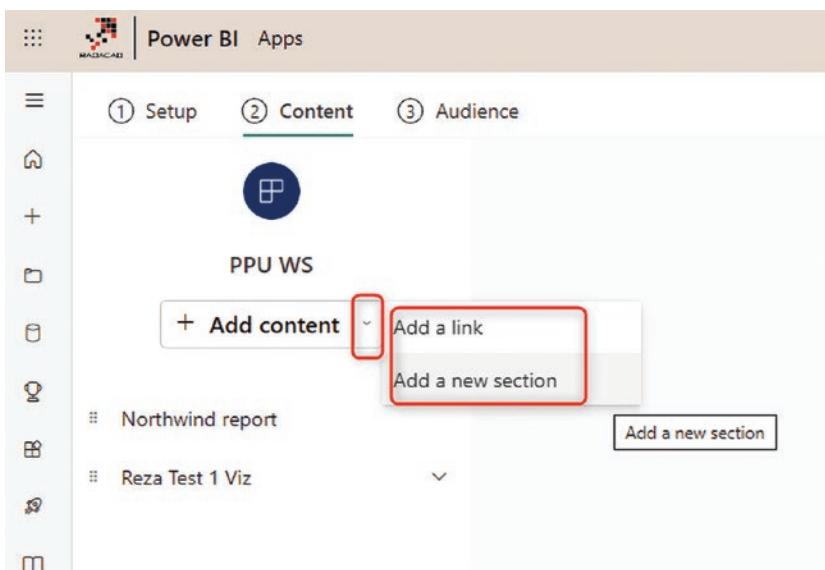
## Content

In the next tab, you will set the content for the app. This can be done by clicking Add Content (see Figure 31-6). If you want to share different content with different audiences, select all the content and add them here.



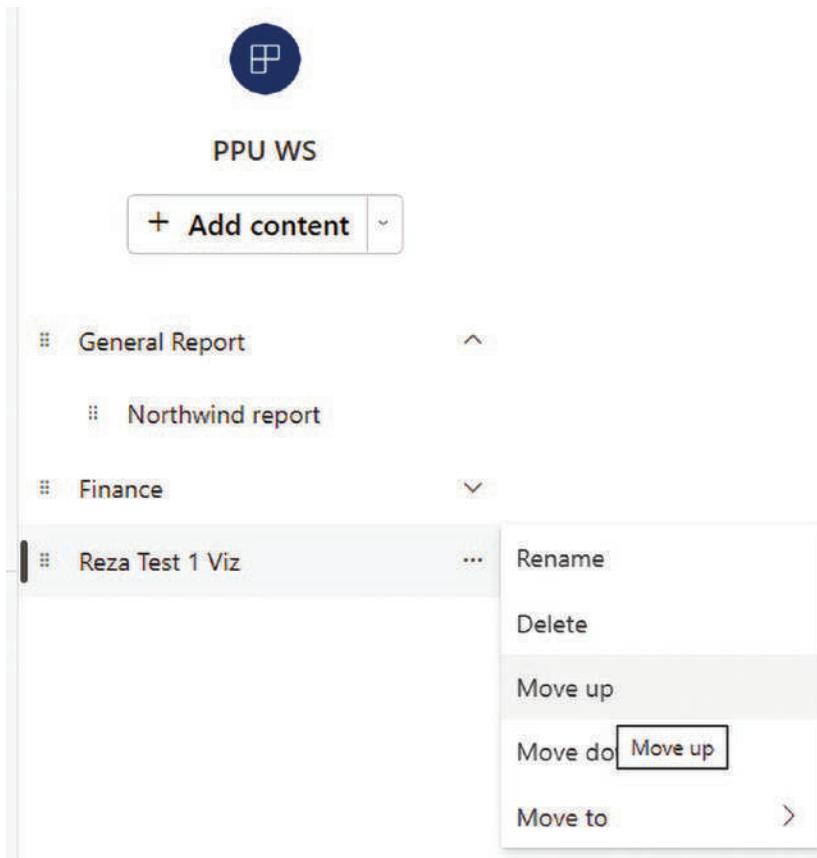
**Figure 31-6.** Adding content to the Power BI app

You can now build a navigation menu for your app. By default, the navigation only includes the content (reports and dashboard). You can build better navigation by adding custom links or sections (see Figure 31-7).



**Figure 31-7.** Adding a link or section to the Power BI app navigation

You can then organize the items under the sections, rename or reorder them, and build the navigation in the way you want your users to consume the content, as depicted in Figure 31-8.

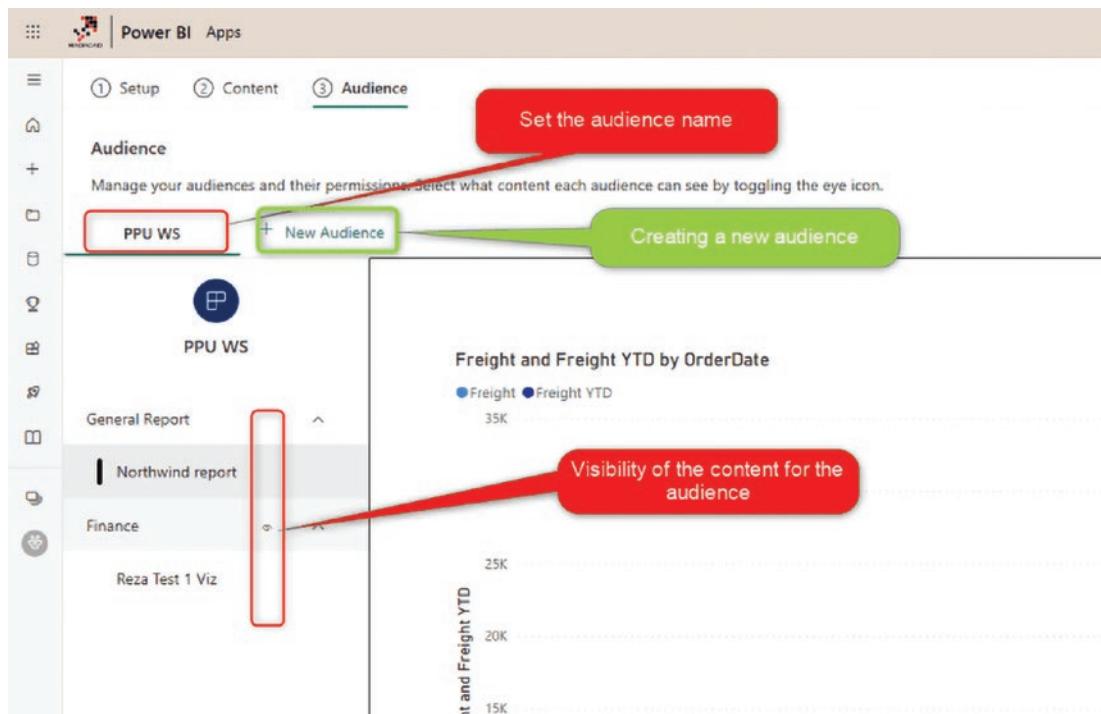


**Figure 31-8.** Building custom navigation in the Power BI app

After building the content and navigation, proceed to the next step.

## Audience

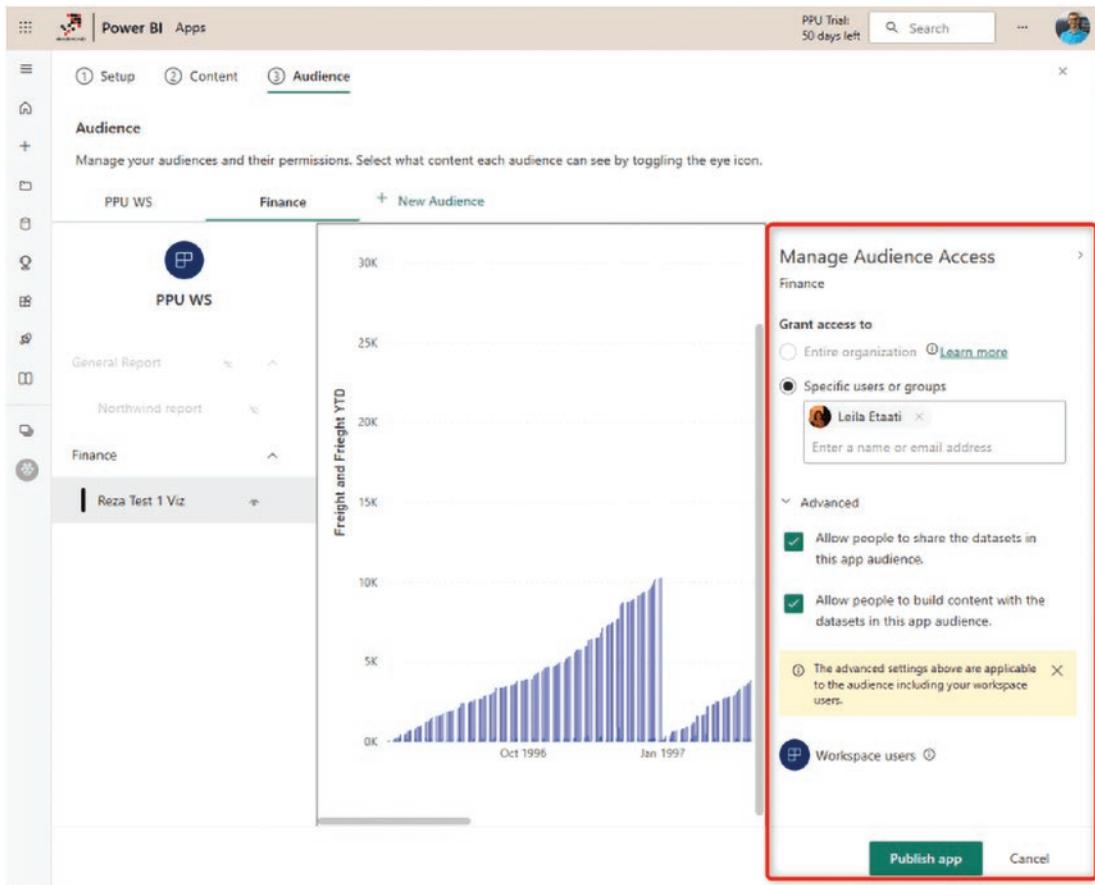
The third and final stage of building a Power BI app is to set the audience. These are the app users. You can include different groups of audiences. In the past, you could only create one audience group per app. Now, you can have a group with access to part of the content and another group with access to another part of the content in the same app. Figure 31-9 illustrates this process.



**Figure 31-9.** Editing the Power BI app's audience

There will always be a default audience with the same name as the app. You can rename it by double-clicking the Audience tab.

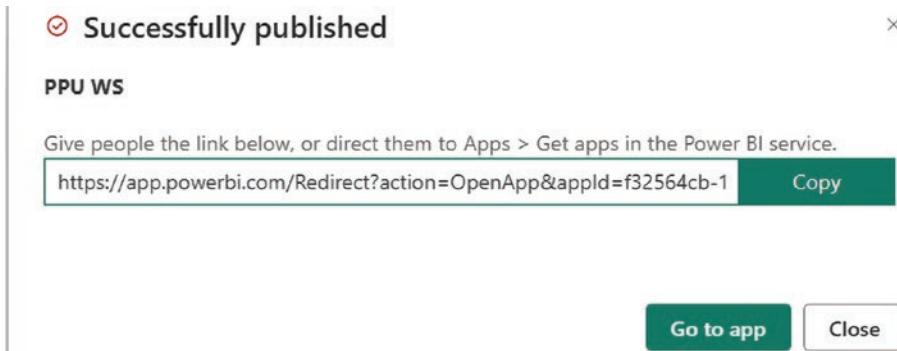
For each audience, you can set the group of users who have access to it (see Figure 31-10). It is better to use Office 365 groups than the individual users. This enables you to manage the user list more easily outside the Power BI environment.



**Figure 31-10.** Setting audience access to the Power BI app

When you create the app, you can give the users access to reshare it or even build new content with the dataset available in the app (using a live connection). As you can see in Figure 31-10, I created a new Audience for Finance, allowed it to see only the finance section and report, and then authorized the users of this audience to reshare or build new content using the dataset of this app.

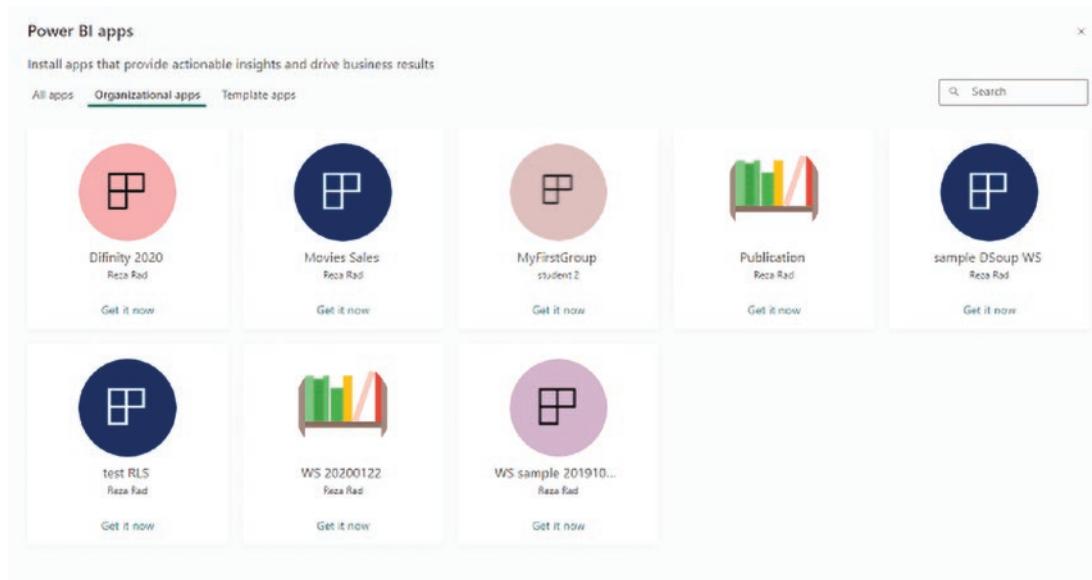
After publishing the app, users can access it immediately. They will either get the app installed automatically or need to install it themselves. You can also share the link with them to make it easier for them to access it, as demonstrated in Figure 31-11.



**Figure 31-11.** The link to access the app

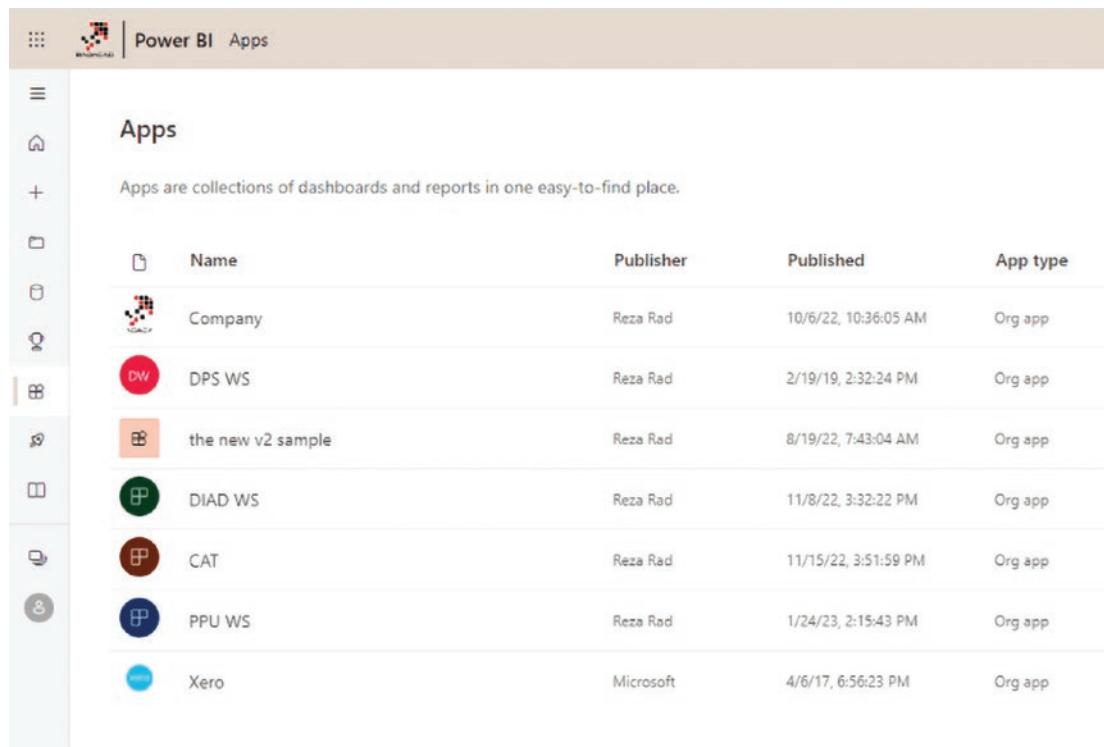
## Getting Apps

Users can go to their Power BI account page, click Apps, and then choose Get Apps. They will be able to see all the apps shared with them (see Figure 31-12).



**Figure 31-12.** Getting apps from the organization's tenant

Users can simply install an app and use it right away. As you can see in Figure 31-13, users can have multiple apps.

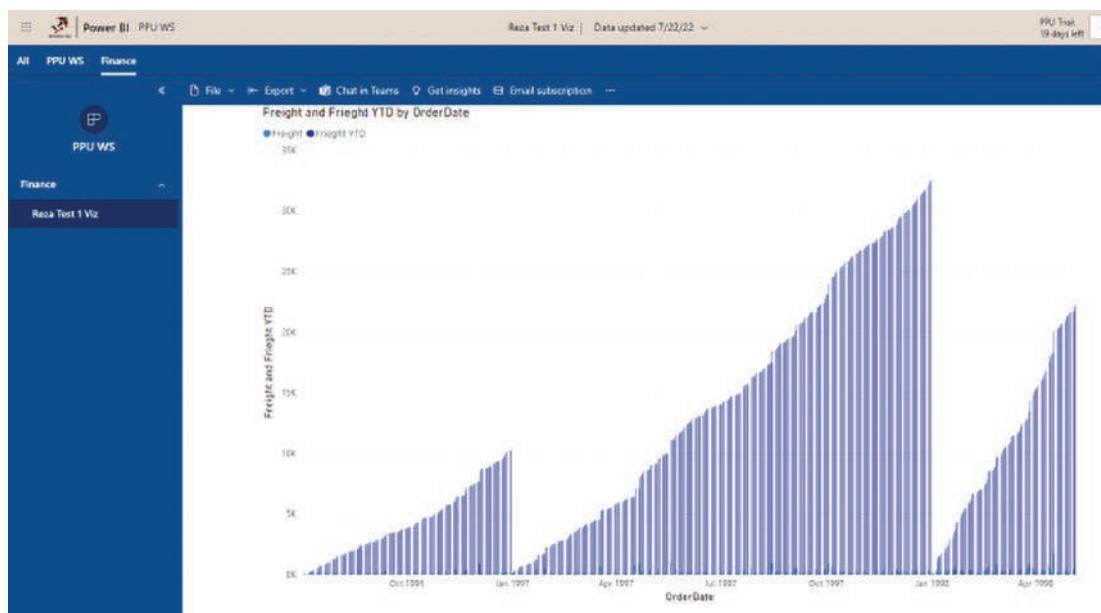


The screenshot shows the 'Power BI Apps' section. On the left is a vertical toolbar with icons for Home, New, Dashboard, Report, and others. The main area has a header 'Apps' and a sub-header 'Apps are collections of dashboards and reports in one easy-to-find place.' Below is a table with columns: Name, Publisher, Published, and App type. The data is as follows:

Name	Publisher	Published	App type
Company	Reza Rad	10/6/22, 10:36:05 AM	Org app
DPS WS	Reza Rad	2/19/19, 2:32:24 PM	Org app
the new v2 sample	Reza Rad	8/19/22, 7:43:04 AM	Org app
DIAD WS	Reza Rad	11/8/22, 3:32:22 PM	Org app
CAT	Reza Rad	11/15/22, 3:51:59 PM	Org app
PPU WS	Reza Rad	1/24/23, 2:15:43 PM	Org app
Xero	Microsoft	4/6/17, 6:56:23 PM	Org app

**Figure 31-13.** Multiple apps can exist per user

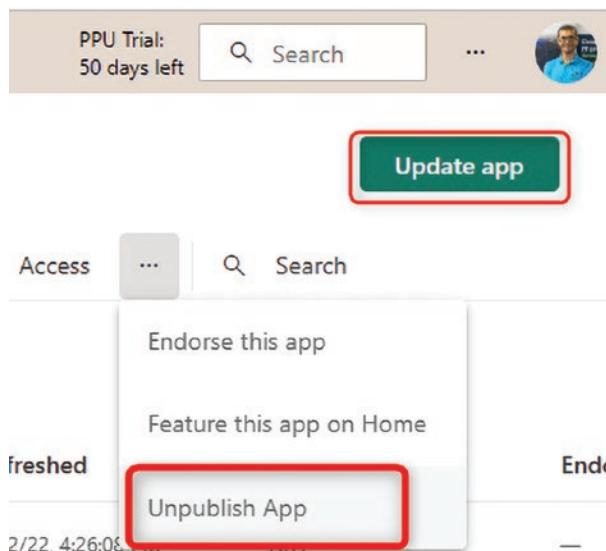
By clicking the app, users will be redirected to it and can explore its content. Depending on the access and the audience group the user is part of, they may have a different view of the app. See Figure 31-14.



**Figure 31-14.** End-user view of the Power BI app

## Making Changes to an App

You can apply any changes to the content in the workspace. The changes on the workspace will not affect users until you update the app. With the new update, the users will get the updated content. You can also unpublish an app (see Figure 31-15).



**Figure 31-15.** Updating or unpublishing a Power BI app

## Isolated Environments for Developers and Users

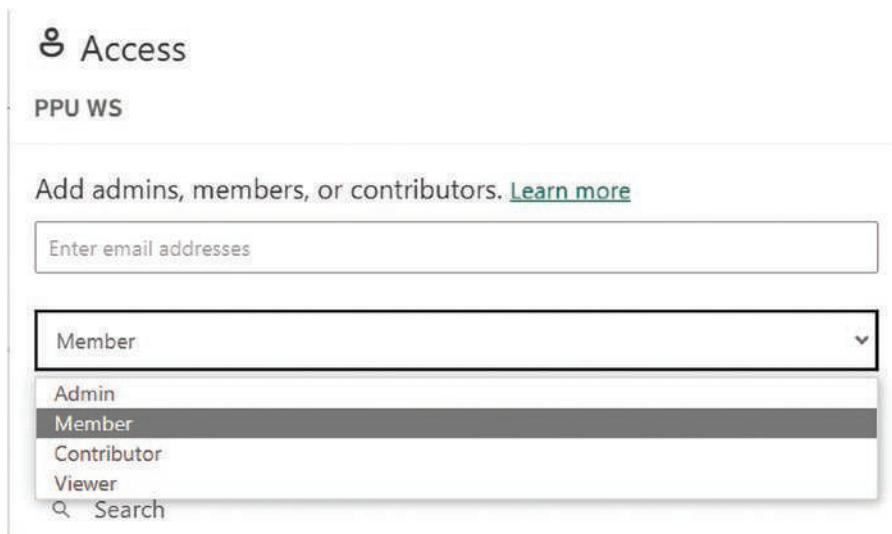
One of the best advantages of using Power BI apps is having two separate environments—one for developers and one for users. The concept works merely because a Power BI app is always associated with a workspace. Workspaces act like developer environments, and the Power BI app serves as an end-user environment.

---

Workspaces are developer environments, and Power BI apps are end-user environments.

---

If you are a developer, you will have edit access to the workspace (through Contributor or Member roles). See Figure 31-16.



**Figure 31-16.** Setting access for the workspace users

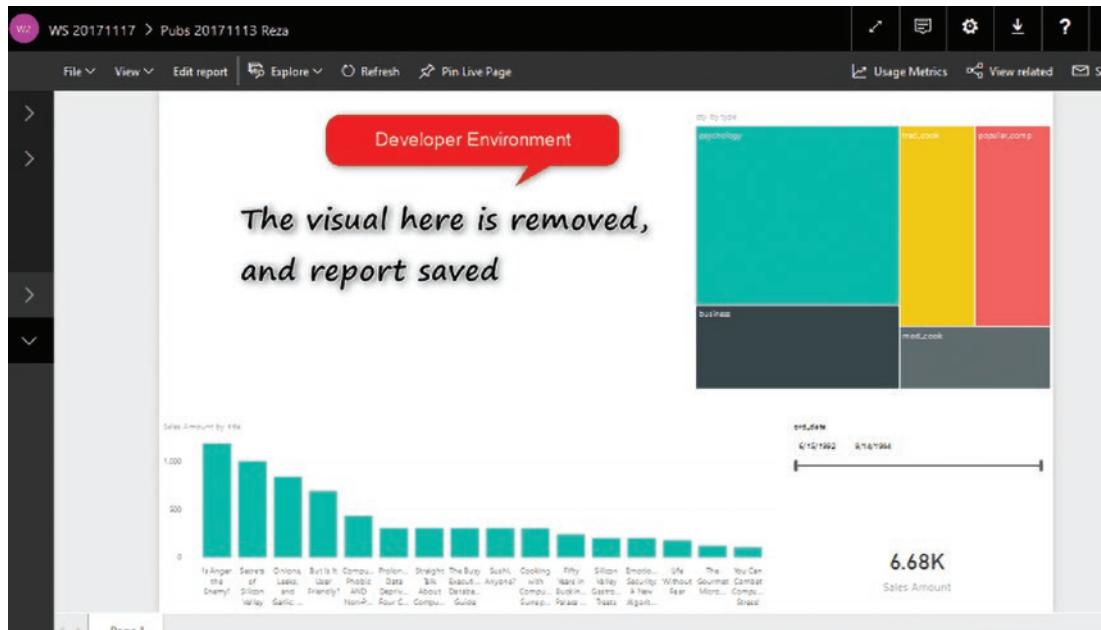
All users who have edit access can change the content in the workspace. Their changes do not apply to the end-user environment until they update the app.

---

Changes in the workspace do not affect the end users' environment until the app is updated.

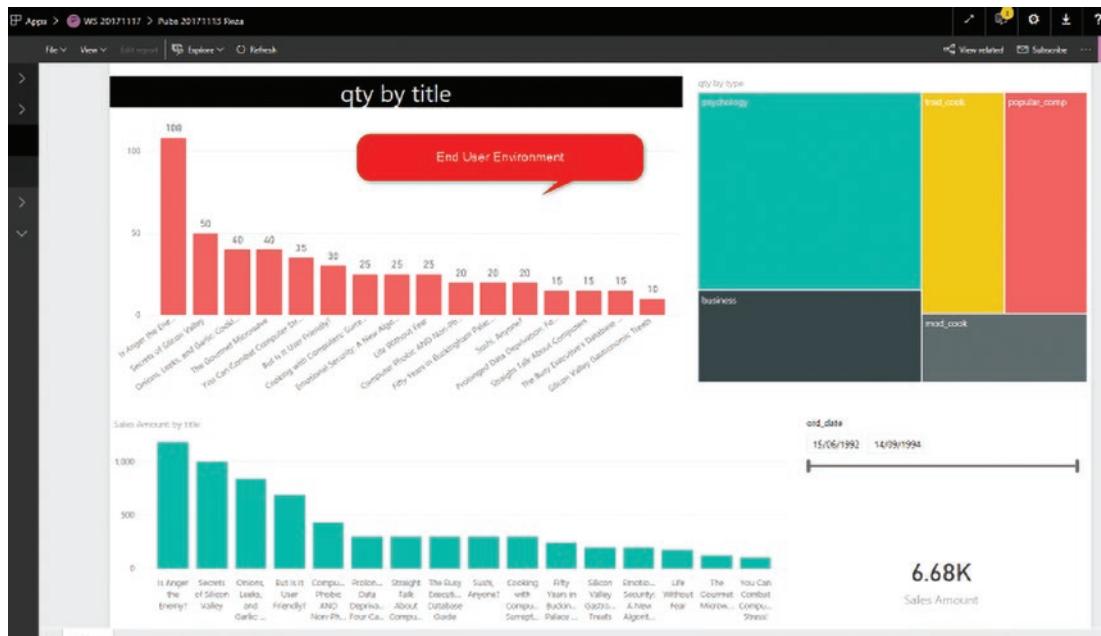
---

As a developer, you can make changes to the workspace, for example, you remove a chart from a report and then save it, as shown in Figure 31-17.



**Figure 31-17.** Changes made to a workspace by a developer

The end users will not be affected by this change. They will still see the full report, as shown in Figure 31-18.



**Figure 31-18.** The full report remains visible to the end users

To sync the developer and end-user environments, you need to update the app and apply the changes to the end-user side. Usually, developers in the workspace cannot do this. The access level of developers in the workspace is Contributor, which doesn't allow them to update the app. The Member is a user who can publish or update the app. This person is also called the deployment user. See [radacad.com/best-practice-for-power-bi-workspace-roles-setup](http://radacad.com/best-practice-for-power-bi-workspace-roles-setup) to learn more about workspace roles and how to set them up.

## What About Data Refresh?

Data Refresh and access to up-to-date data is always a big issue for potential developers. You might wonder if you have to update the app so end users can get the updated data. The answer is that data will be updated if it is scheduled to refresh with a scheduling process determined at the dataset level or if it is Live Connection or DirectQuery. Regardless of the users or developers, the dataset refresh is a different process.

If you have set up your dataset to be refreshed, users will always have access to up-to-date data. You don't need to update the app for data changes. You just need to update the app for structural changes (adding, modifying, or removing tables, fields, relationships, and calculations) or visualization changes (adding, modifying, or removing visuals on the report). However, some structural changes in the dataset might cause the report in the app to fail (such as removing a field or a measure used in a report's visualization), as it depends on the existing dataset in the workspace.

# Advantages of Power BI Apps

### Separate environments for developers and end users

This method has two separate environments; an environment for developers to edit the Power BI content in a collaborative workspace and another environment for end users to consume the reports. End users are only able to view the reports, and developers can make changes.

---

Power BI apps are the best solution for isolated developer and end-user environments.

---

### A cost-effective option for large userbases

If you have thousands of users and want to share Power BI content with them securely, sharing content with options that rely on per-user licensing is not cost-effective. A Power BI workspace can be assigned to a Premium workspace, and then if you create an app on top of that, the app's users can be free Power BI users. This is a big help for organizations with a large userbase. You can purchase a capacity, and as long as it can cover the load of the users over the Power BI content in the service, your free users can access it.

This option is one of the most common reasons why many organizations use Power BI apps combined with workspaces to share Power BI content.

### Controlling multiple Power BI content sources

Similar to the workspace, the Power BI app allows you to share multiple dashboards, reports, and datasets simultaneously. Controlling multiple content sources means less maintenance overhead compared to dashboard sharing, which is one dashboard at a time.

### External sharing

Another great benefit of Power BI apps is the ability to integrate them with Azure B2B services and provide external sharing. If you want to share Power BI content with people outside your company, you can do that with a combination of Azure B2B and the Power BI app.

## Cons of Power BI Apps

Power BI apps do have an important limitation that you need to consider.

### **Changes to the dataset apply immediately**

Power BI apps separate the developer and end-user environments, and the changes in a report in a workspace don't affect the end user until you update the app. However, this functionality doesn't work with the dataset in that way. The Power BI apps and workspaces share the same datasets, so any changes from a Scheduled Refresh or structural changes apply to both.

## Summary

Power BI apps separate the development and user environments. They also provide a more cost-effective licensing option in some scenarios. Using Power BI apps, you can create a navigation menu for users with different audiences per app. These advantages make Power BI apps the most common method of sharing Power BI content.

## CHAPTER 32



# Publish to Web

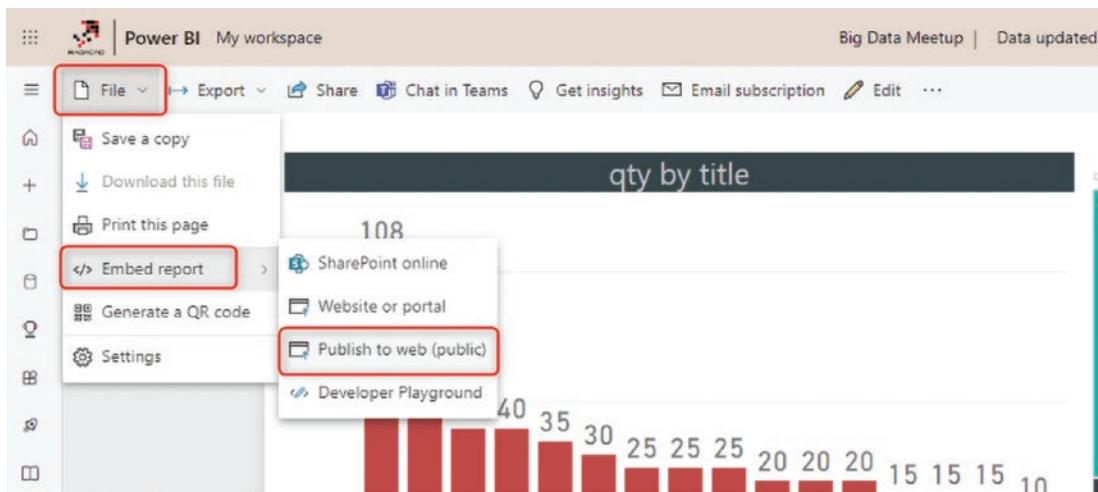
All of the methods for sharing Power BI files in the Power BI Service (such as dashboard sharing, workspaces, apps, and so on) need paid Power BI subscriptions for consuming reports. Users need to be either Power BI Pro or free accounts under the Power BI premium capacity. The Publish to Web option is the only way to share Power BI content through the service for free. With this method, you can share Power BI content with users who don't have a Power BI account.

Publish to Web is an easy way to share public data. However, it has some disadvantages as well. In this chapter, you learn about this feature in more detail, and you learn how it is different from Power BI Embedded. I explain Publish to Web, and in another chapter, you'll learn about Power BI Embedded. In this chapter, you learn how easy it is to share your report with the public through a web page, which can be your blog post, an HTML page, or any other web page. I answer some frequently asked questions about this feature as well.

## What Is Publish to Web?

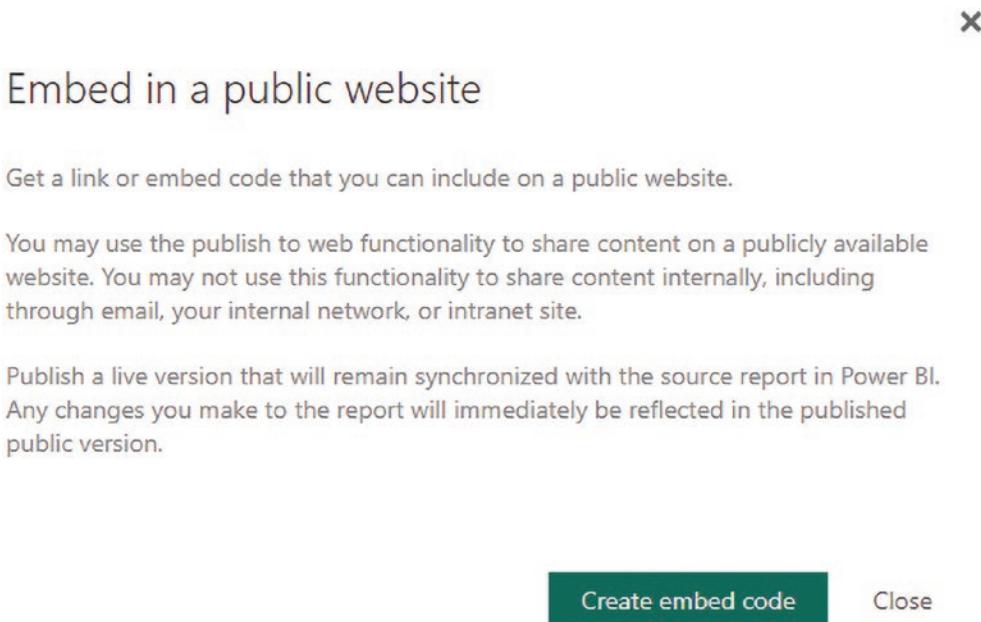
Once you publish your Power BI report into a Power BI Service, you can share it with others by sharing it directly or using workspaces in Power BI. What if you want to share it with the public through the web? Let's say you want everyone to see the report and play with it (with all the interactive features in Power BI). In this case, you can use Publish to Web. Publish to Web allows you to create an embedded code for the Power BI report and use that code on the web page. This simple feature enables everyone to access the report. They won't be able to edit the report, but they will see the report, and the report will be fully interactive so they can highlight items, select slicers, and drill down.

It's very simple to use this feature. You must first deploy or publish the report into the Power BI website or service. Then click the report (not the dashboard). Once you open the report, click the File menu option, and under Embed Report, choose Publish to Web (see Figure 32-1).



**Figure 32-1.** Using Publish to Web from a Power BI report

Then you will be informed about this feature in a message box that mentions this step will create a link and embed code for you to share with the world through a website or email. Click Create Embed Code, as shown in Figure 32-2.



**Figure 32-2.** Embedding in a public website

Because this is all about sharing a report, you will be informed again to check the confidentiality of the data and make sure you are sharing content that is not harmful to the organization or someone when it is viewable to the public (see Figure 32-3).

## Embed in a public website

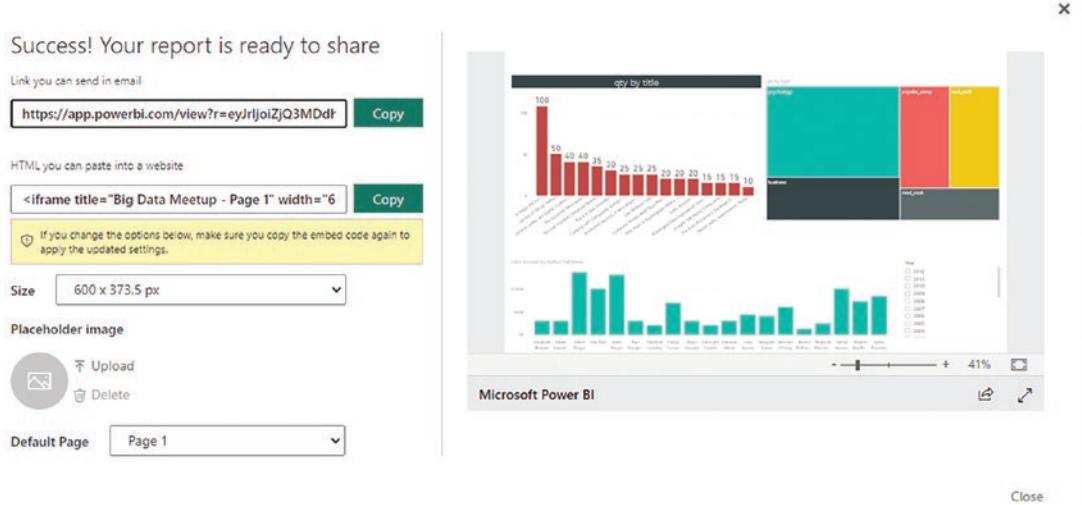
**⚠** You are about to create an embed code for this report. Once published, anyone on the internet will be able to access the report and the data it contains, and Microsoft may display the report on a public website or a public gallery.

Before publishing this report, ensure you have the right to share the data and visualizations publicly. Do not publish confidential or proprietary information, or an individual's personal data. If in doubt, check your organization's policies before publishing.

**Publish**      **Close**

**Figure 32-3.** Confidentiality warning

After clicking Publish, you will see the embed code plus a link to share through email. You can also choose the size of the screen for the embed code (see Figure 32-4).



**Figure 32-4.** The embed code and the public link for the Power BI report

You can now browse the link to see the report in a browser, even if you open it in Incognito (or Private) mode, as illustrated in Figure 32-5.



**Figure 32-5.** Open the public link in a web browser

To browse the report, you don't need to log in to the Power BI service. Users don't need anything except a browser to view this report. The report is fully interactive; users can highlight, select, and deselect items.

You can also add the embed code to your HTML page, blog post, or wherever you want users to see the report.

## Security Issues with Publish to Web

### Everyone Can See What You Share

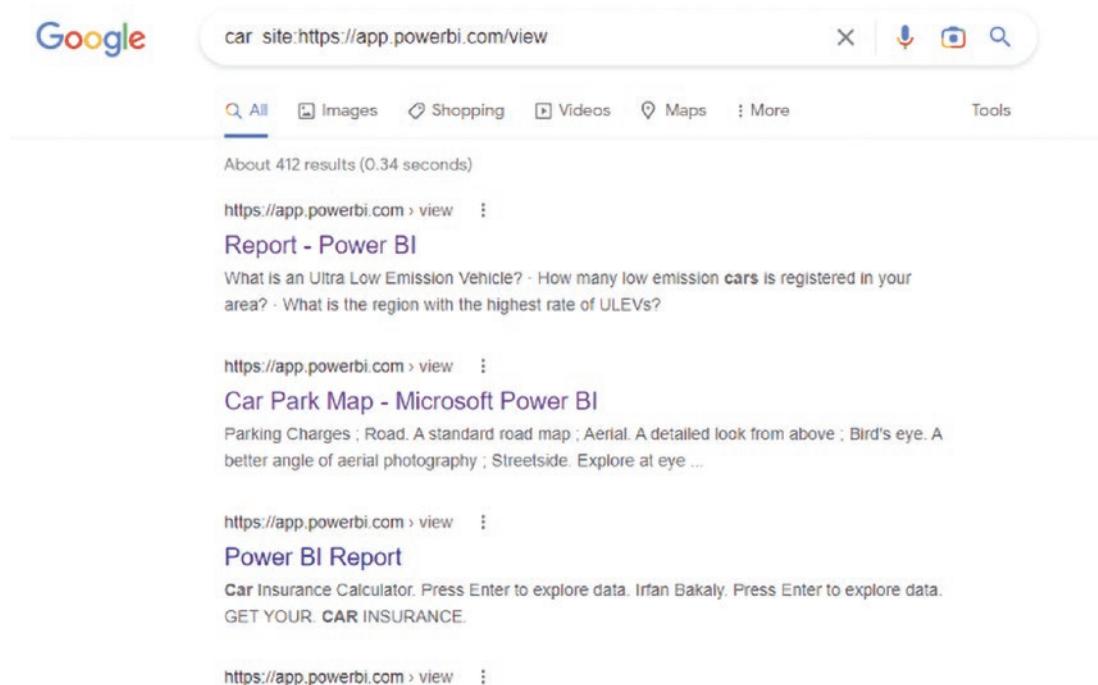
The first thing you might think of is usually security. How can you manage security with this kind of setup? The short answer is that there is no security here. The report is shared through the web or email with everyone. So everyone who has the link or the embed code can access the report. They cannot edit it. But they can view it with no restrictions.

### Users can share it with others

A report published on the web has a share section on the bottom-right side. Everyone can share this report with anyone else through all social media channels—Facebook, Twitter, and LinkedIn. This method of sharing is not secure. I recommend sharing the data only on your company or organization's website.

### The report is public

This report is not shared only with those with the link, it is shared globally on the Internet. A search engine such as Google can find the report. All the reports with Publish to Web links are available. Figure 32-6 shows an example search result from Google.



**Figure 32-6.** Searching for Power BI reports published publicly

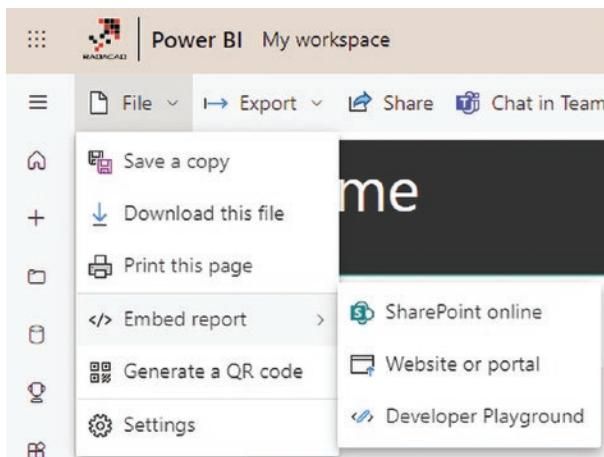
You must be sure that the report and its data don't reveal confidential information.

**All report pages are visible**

If you have a report with ten or more pages, it will be visible to browsers. You cannot limit which pages you want to show and which you don't. I recommend creating different reports if you want to restrict some pages and share them separately.

**What if the report has row-level security applied?**

If you have a report with row-level security applied to it, you can't create a Publish to Web link. Figure 32-7 shows an example of such a case.



**Figure 32-7.** Publish to Web cannot be applied to row-level security enabled reports

---

Publish to Web is only recommended for public data sharing on your organization's website. There is no security option for Publish to Web; this method should not be used for confidential reports.

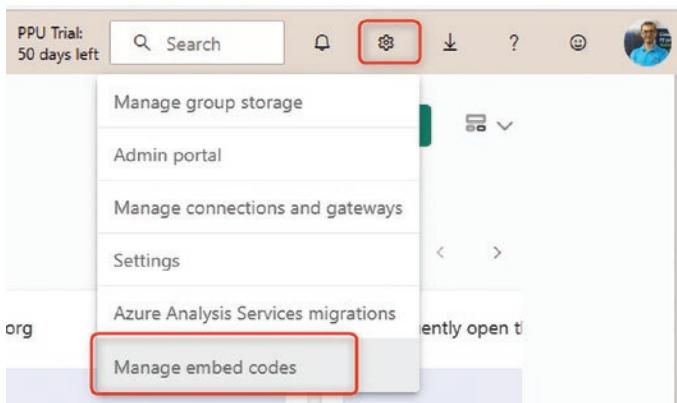
---

#### The link and embed code are synchronized with the report

If you make any changes to the report, all the changes will be synchronized because links and embed codes are simply references to this report. So people will always see the latest version of your report. If you also want to keep the report up-to-date, you can schedule it for data refresh.

## Removing Access

If you want to revoke access to the report, you can do so quickly. Go to the Power BI service, and under Settings, click Manage Embed Codes (see Figure 32-8).



**Figure 32-8.** Finding all your embed codes in the Power BI Service

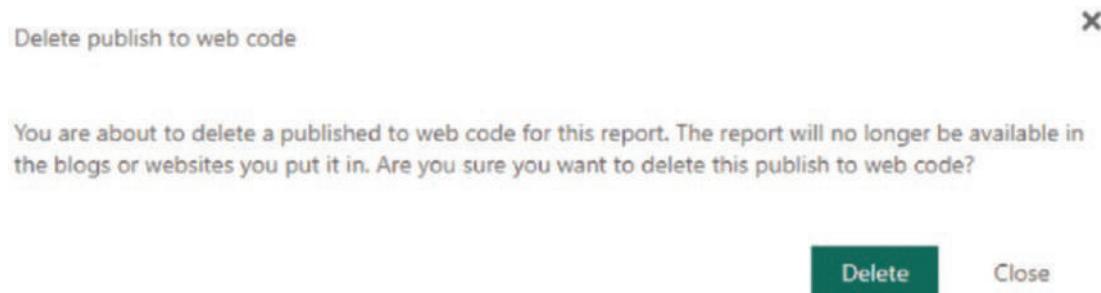
You will see the embed code and can delete it (see Figure 32-9). It is essential you be in the correct workspace because the list of public reports will show the reports only from that workspace.

The screenshot shows the 'Public Reports' section of the Power BI Service. The left sidebar has icons for Home, New Report, Recent Reports, and others. The main area is titled 'Public Reports' and lists eight reports with columns for 'Associated Report', 'Status', 'Date Created', and 'Actions'. Each report row includes a trash icon for deletion.

Associated Report	Status	Date Created	
2020 Salary Survey	Active	1/2/2020, 5:20:31 PM	<code>/&gt;</code> <span style="color: red;">Delete</span>
2021 detailed Salary Survey	Active	1/11/2021, 1:43:53 PM	<code>/&gt;</code> <span style="color: red;">Delete</span>
MVP	Active	7/24/2018, 12:15:01 PM	<code>/&gt;</code> <span style="color: red;">Delete</span>
Power BI Summit Content View	Active	3/26/2022, 11:00:51 AM	<code>/&gt;</code> <span style="color: red;">Delete</span>
RD	Active	12/20/2019, 1:01:57 PM	<code>/&gt;</code> <span style="color: red;">Delete</span>
Salary Survey 2020	Active	12/17/2019, 2:55:16 PM	<code>/&gt;</code> <span style="color: red;">Delete</span>
Salary Survey 2021	Active	12/29/2020, 11:10:46 AM	<code>/&gt;</code> <span style="color: red;">Delete</span>
SQL Saturdays	Active	12/18/2019, 6:32:48 PM	<code>/&gt;</code> <span style="color: red;">Delete</span>

**Figure 32-9.** Finding your embed codes and deleting them in the Power BI Service

Once you delete the embed code, no one can access the report from the web. You will see a notification message about this, as shown in Figure 32-10.



**Figure 32-10.** Deleting a public report link

If you delete the embed code, the link and the embed code will show a message to the public web users that this content is not available.

This content is not available.  
[Learn more about Power BI.](#)

## Central Monitoring for all Embed Codes

Publish to Web might seem like a frustrating option with all its security holes. There is a need for an administration page to manage all embed codes across the Power BI tenant. There is a place where the Power BI tenant administrator can find all reports that are published to the web, and you can remove those from being published to the web.

To go to the Admin Portal, click the Setting icon in the Power BI Service, and then go to the Admin Portal. Click Embed Codes on the left side, and you will see all embed codes published by anyone from your organization (see Figure 32-11).

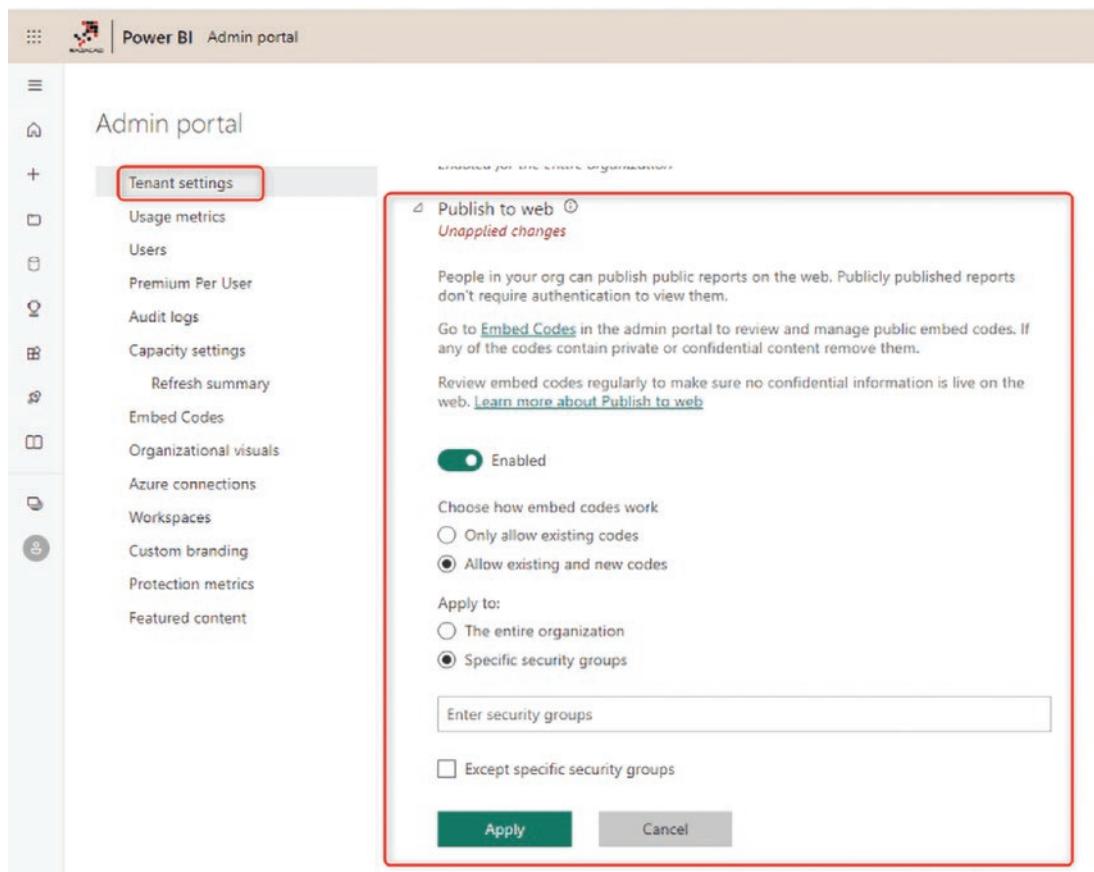
Report name	Workspace name	Published by	Status
Productivity_Final	My workspace	Reza Rad	Active
Car Accidents	My workspace	Reza Rad	Active
Agenda	My workspace	Reza Rad	Active
Speakers	My workspace	Reza Rad	Active
MVP	Public Reports	Reza Rad	Active
Events	My workspace	Reza Rad	Active

**Figure 32-11.** Finding all the public reports in the Power BI tenant

There are two options for each report published to the web—view the published report or delete it. Once you delete it, no one can use the published web link anymore.

## Who Can Publish Reports on the Web?

The Power BI tenant administrator can turn off the Publish to Web feature entirely or authorize it only for a specific group of users. I highly recommend using this setting and either turning it off entirely or enabling it for a group of users from the BI or data analytics team who know all the security problems of this method. Do not turn on this feature for the entire organization. This configuration is in the Tenant Settings area (see Figure 32-12).



**Figure 32-12.** Enabling or disabling Publish to Web in the Tenant Settings area

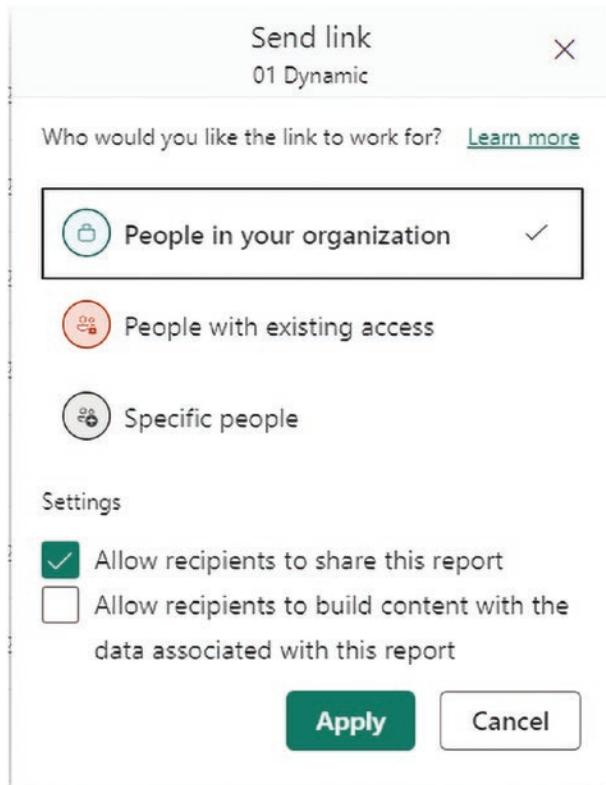
## Differences with Other Sharing Methods

You can share your dashboards and reports with people in your organization. This feature also gives you a link to the report. However, it is different from using the public link. Here are some differences between sharing dashboards and using a public link:

- Only those who have access to the dashboard will see the content. Sharing a dashboard link with the public won't show anything if they are not authorized to see it.
- Users need to have Power BI accounts.
- Power BI workspaces and apps are only for authorized groups of users, not for everyone.
- Power BI Embedded is different from Publish to Web.

**Only those who have access to the dashboard will see the content**

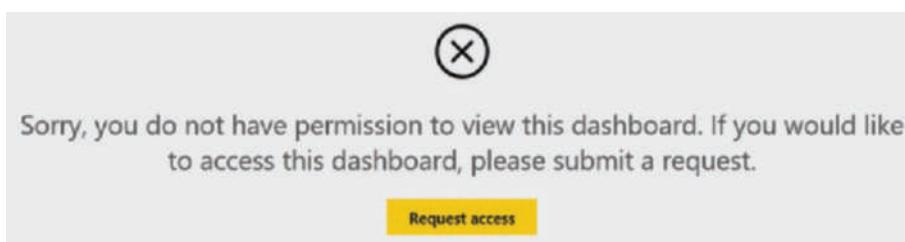
Once you share a report, you can choose who has access to it based on their Power BI accounts (see Figure 32-13). If they don't have a Power BI account or they don't have access to that content, they won't see the content.



**Figure 32-13.** Sharing content using dashboard (or report) sharing

#### The dashboard link works only for authorized users

Despite having the dashboard link available, only authorized users can see the content when browsing the link. Otherwise, they will get a message that says they don't have permission to view this dashboard (see Figure 32-14).



**Figure 32-14.** The message displayed to unauthorized users

**Power BI workspaces and apps are for authorized groups of users inside or outside of your organization, not for everyone**

You can share dashboards, reports, and datasets with your organization's workspace. Users will have access to all the content or part of it (using the audience settings in the Power BI app). However, with Publish to Web, any users who have access to the page can see the report, regardless of whether they have a Power BI account (see Figure 32-15).

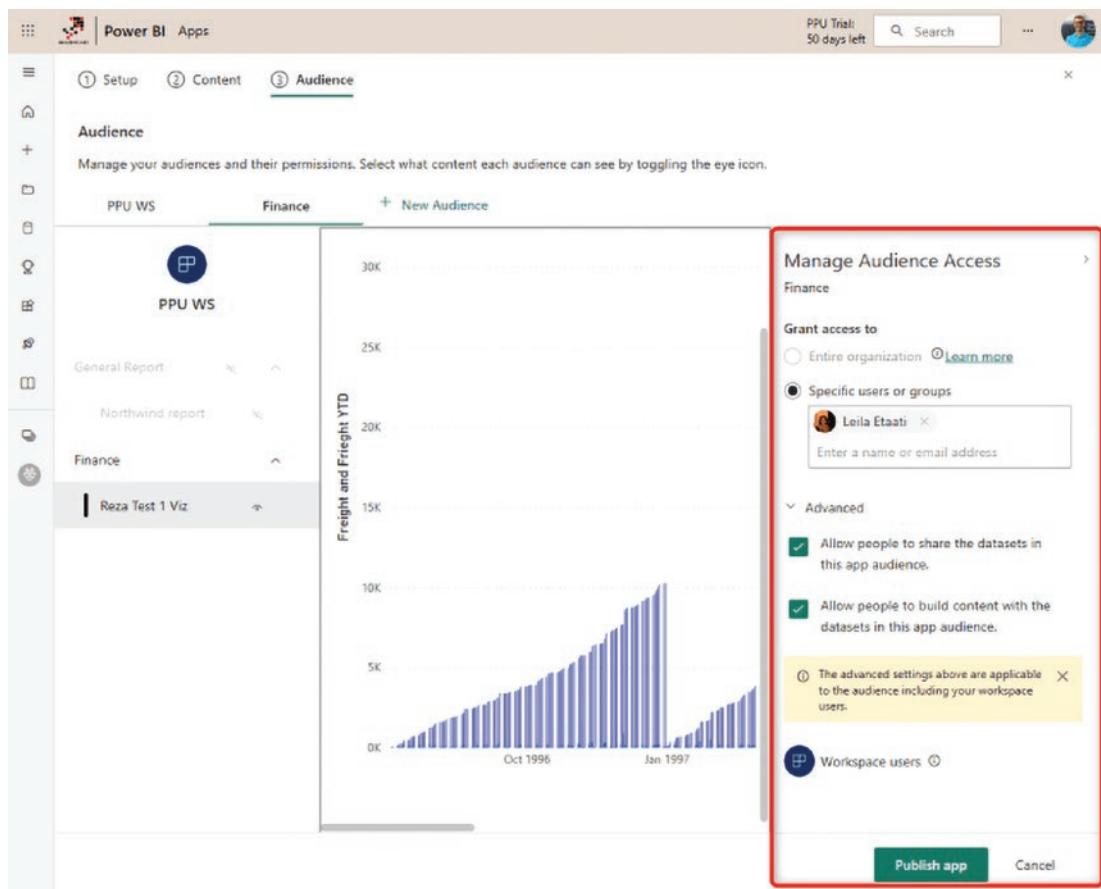


Figure 32-15. Sharing content through Power BI apps

#### Public access or organizational sharing

The difference between Publish to Web and the other sharing methods is about giving public access or sharing content through the organization:

- When you share dashboards or use the Power BI workspaces and apps, you share content with other users. These users should be Power BI users; they cannot access content anonymously. They must log in to the Power BI Service to access the content.
- With Publish to Web, anyone can access the report, even if they don't have a Power BI account. They don't need to log in. They can browse the page that contains Power BI embed code with no restrictions on viewing the report.

**Power BI Embedded is different from Publish to Web**

Power BI Embedded brings Power BI into applications. Yes, you can share your Power BI report through an application with API keys. You can share the report with application users, even if they don't have Power BI accounts. However, this is much more flexible. You can choose which reports you want to share with which users in the application. Power BI Embedded brings the Power BI experience into an application with security configuration enabled for users. Power BI Embedded is fully secured in a custom application, but the public reports are not.

---

Publish to Web is free but it's not a secure way of sharing, and it is for the public. Power BI Embedded is secure, paid service, and it's for specific people who you authorize.

---

## Summary

In summary, Publish to Web is the only free way to share Power BI content. This method doesn't have any security bound to it. As soon as you publish a report to the web, anyone with that link can access the report and the data. This method of sharing is easy. However, it is not recommended for confidential data. This method is a reasonable option if you want to share public reports on your company's public website.

Publish to Web is entirely different from Power BI Embedded or Secure Embed; these methods should not be considered the same.

## CHAPTER 33



# Power BI Embedded

Power BI includes a very powerful method for sharing called Power BI Embedded. This method of sharing is powerful because you can share Power BI content in your custom application, and you can share it with your custom application users, regardless if they have Power BI accounts. In this chapter, you learn everything about Power BI Embedded, including but not limited to its scenarios and how to implement it.

## Using Power BI Embedded

Power BI has strong and compelling sharing methods for organizations, such as workspaces and Power BI apps. However, there are scenarios where you may find a better sharing method. Consider using Power BI Embedded when you have scenarios like these:

- You want to share a Power BI report with users who do not have Power BI accounts, but your report needs to be shared securely so that only authorized users can see the content.
- You want to embed the Power BI report inside a custom application. This might be because your users are already using a custom application for other operational tasks, and having the Power BI report in the same application will give them a central point of work.

An example of these scenarios is an ISV company with customers from all other companies; their Office 365 tenant might be entirely different for every company. If the ISV company wants to share the Power BI report with all its clients, it first has to create Power BI accounts for each client in the tenant and then publish the Power BI report separately to each tenant. This process involves lots of repetitive work.

Power BI Embedded can really help this ISV company. They can embed Power BI reports in a custom ASP.NET application and then create users and passwords as part of the application. The username and password of the application are not Power BI users, of course. Everyone who has access to that page can consume the report.

There are a few things you need to know before choosing Power BI Embedded as your method of sharing:

- Licensing rules for Power BI Embedded are different. The page renders matter.
- Power BI Embedded doesn't have all the features of Power BI services (such as subscriptions, alerts, and so on).
- You need a web developer to handle embedding and any further changes.

Power BI licensing is either per user (Free, Pro, and PPU) or capacity-based (Premium or Embedded). Power BI Embedded uses a capacity-based licensing. However, the capacity specification for entry nodes for the Embedded is very limited. It is important to choose a capacity that can cover the workload. Page renders can be a good indication of which licensing plan is appropriate.

When you use Power BI Embedded, you aren't using a Power BI Service. As a result, you will not have features such as usage metrics (per-user details) reports, alerts, subscriptions, and features that are specific to the service. You can obviously implement anything you want, but you need to have a good web developer on your team.

Power BI Embedded embeds a report into an application and needs a web developer to take care of this process. It is not just about the embedding itself; that may be a one-off process. When you embed the report into your application, all other requests come after that. You need to manage users and memberships for your web application. You may want to have a usage analysis of the report. You may want to have different levels of access to each report. Every extra feature comes as a request for the web developer to implement. It is possible to implement, but these all come at the cost of hiring a web developer.

## Myths about Power BI Embedded

Power BI Embedded was recently enhanced. There are many myths about this method that need clarification:

- Power BI Embedded only works with Live Connection, not Import Data or Scheduled Refresh. This statement is a myth.
- Power BI Embedded does not support all data sources. This statement is a myth.
- You can only embed the Power BI report, not the dashboard or Q&A. This statement is a myth.
- Row-level security is not possible with Power BI Embedded. This statement is a myth.

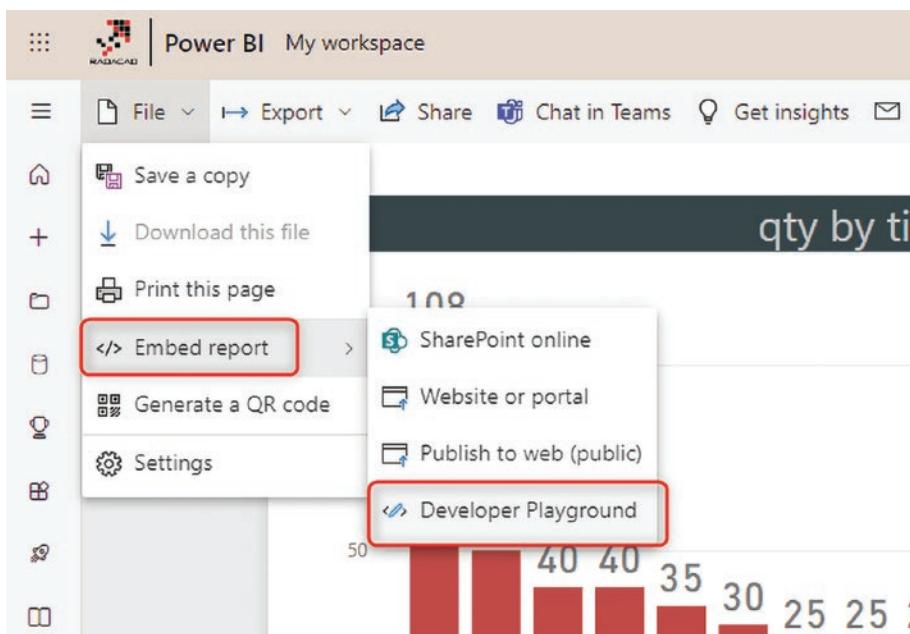
Many of these statements were correct at the beginning phase of the Power BI Embedded (perhaps in 2015 or part of 2016). However, with recent updates and enhancements, you can do these things with Power BI Embedded. For your peace of mind, a Power BI team is only focused on embedded features, and they are working hard to bring more features to this technology regularly.

## What Is Power BI Embedded?

After explaining these aspects of Power BI Embedded, it is now time to talk about what Embedded is. Power BI Embedded adds a report, dashboard, or dataset with Q&A to a custom application. This embedding is fully secure because it leverages the authentication keys and tokens. This way of embedding is not like Publish to Web. Publish to Web is not a secure method; however, Power BI Embedded is fully secure and customizable.

## Embedded Playground

There is no place better than the Power BI Embedded Analytics Playground to familiarize yourself with the Embedded functionality and all its necessary codes. You can get to Playground from a Power BI report, as shown in Figure 33-1, or you can directly go to [playground.powerbi.com/](https://playground.powerbi.com/).



**Figure 33-1.** Getting your report in the Power BI Embedded Playground

If you navigate to the Developer Playground using the method shown in Figure 33-1, it will open the Playground with your report. You can then see what report in an embedded solution, as shown in Figure 33-2.

The screenshot shows the Microsoft Power BI Embedded Analytics Playground (Preview) interface. On the left, there is a sidebar with various operational and configuration tabs like General Operations, Properties, Navigation, Authoring, Accessibility, Filters & Slices, Edit & Save, Menu Operations, Layout, Bookmarks, and Data. The main area has a search bar at the top. Below the search bar, there is a yellow banner with the text 'Set up your Power BI Embedded demo environment'. Underneath the banner, there is a code editor window containing the following JavaScript code:

```

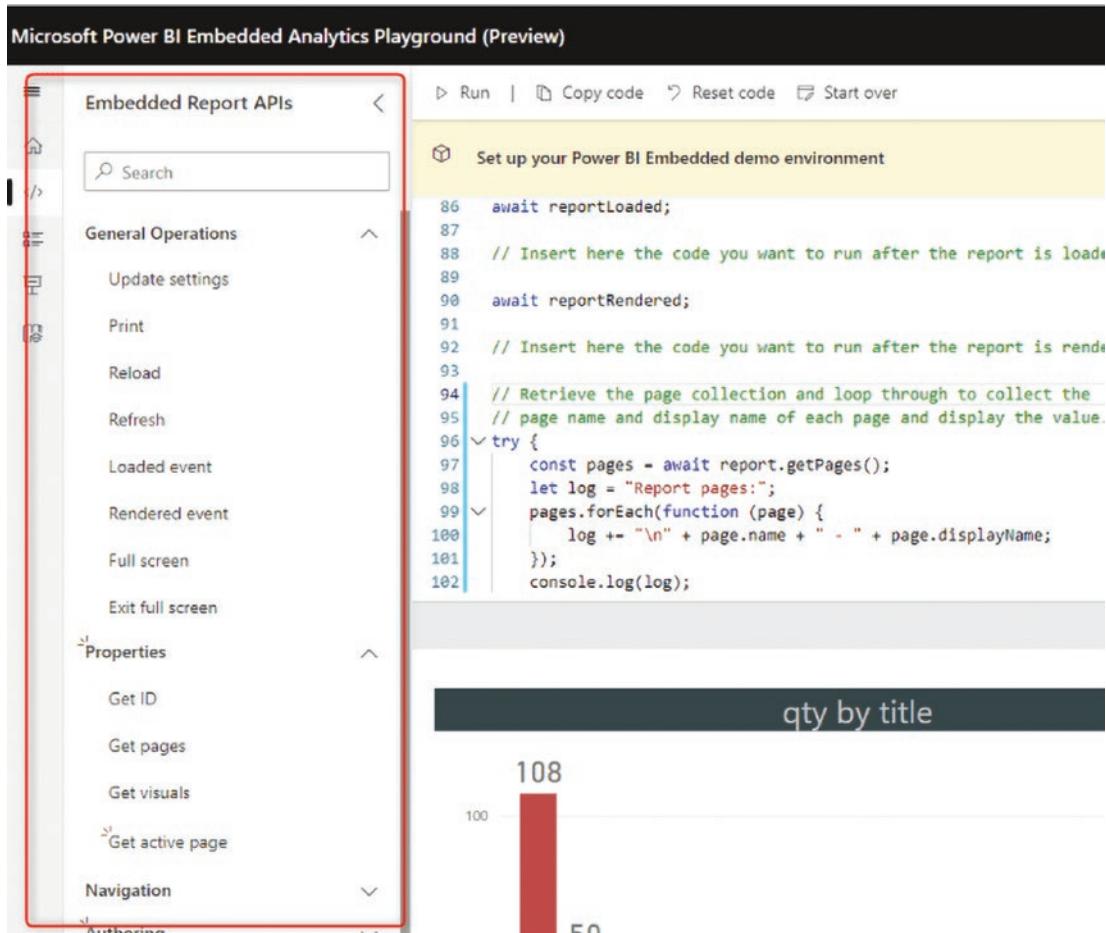
6 // Inject a Power BI report in the given HTML element with the given configuration
7 // Read more about how to embed a Power BI report in your application here: https://go.microsoft.com/fwlink/?linkid=2155100
8 > Function embedPowerBIReport() { ...
9
10
11     embedPowerBIReport();
12     await reportIntialized();
13
14     // Insert here the code you want to run after the report is loaded
15
16     await reportInitialized();
17
18     // Insert here the code you want to run after the report is rendered.
19
20 }
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94

```

Below the code editor, there is a preview of the report. The report title is 'qty by title'. It contains a bar chart with values [108, 50, 40, 40, 35, 30, 25, 25, 25, 20, 20, 20, 15, 15, 15, 10] and a treemap visualization. The treemap categories are 'psychology' (teal), 'business' (dark grey), 'popular\_camp' (red), and 'sport\_cook' (yellow). There are also filters and search bars on the right side of the preview area.

**Figure 33-2.** The Power BI Embedded Playground

The Playground is where you can also see other APIs and functions. Try them to see their code in action (see Figure 33-3).



**Figure 33-3.** Trying APIs in the Power BI Embedded Playground

The Embedded Playground is a perfect way to start learning about functions and how they work in the code.

## Licensing Power BI Embedded

Power BI Embedded is based on capacity sizes, and each capacity can cover some workload. Every time you open a page with a Power BI object, it is considered a page render. If you click a slicer, the page renders again; if you click a column chart and it causes other parts of the report to slice and dice, this counts as another page render. Knowing how many page renders you use in the peak hours of operation is important. The pricing list in Figure 33-4 is from [azure.microsoft.com/en-us/pricing/details/power-bi-embedded/](https://azure.microsoft.com/en-us/pricing/details/power-bi-embedded/).

Node Type	Virtual Cores	Memory	Frontend / Backend Cores	Price
A1	1	3 GB RAM	0.5 / 0.5	\$1.0081/hour
A2	2	5 GB RAM	1 / 1	\$2.0081/hour
A3	4	10 GB RAM	2 / 2	\$4.0242/hour
A4	8	25 GB RAM	4 / 4	\$8.0565/hour
A5	16	50 GB RAM	8 / 8	\$16.121/hour
A6	32	100 GB RAM	16 / 16	\$32.2506/hour
A7	64	200 GB RAM	32 / 32	N/A
A8	128	400 GB RAM	64 / 64	N/A

**Figure 33-4.** Power BI Embedded pricing

The cost may look high; however, you get a different result for different scenarios. For a month of service with Power BI Embedded for about 300 page renders per hour (which an A1 should be enough), you pay about \$700 USD, and 300 page renders should be enough for normal users of about 100 users. If you have 100 Pro users, on the other hand, you need to pay \$1,000 USD. Power BI Embedded can be cheaper than Power BI Service in certain scenarios. However, for every scenario, the process needs to be considered separately.

## Some SKUs Can Be Turned On and Off

A very helpful feature of Power BI Embedded is that it can be turned off and on with SKUs. There are two ways to get Power BI Embedded licensing:

- A SKUs (see Figure 33-4), which are part of Azure licensing
- EM SKUs, which are part of Office 365 licensing

A SKUs can be turned off when there is no operation. For example, the embedded capacity can be turned off when no one works after hours. As a result, the organization can save money (that is why you see hourly prices). Or if it is close to the end of the month, that is, peak days for the reporting, the SKU can be upgraded to a higher level and then downgraded to a lower level. This will ensure the reports' high performance and keep costs at a minimum.

To learn more about the differences between A and EM SKUs for Power BI Embedded, check out the licensing chapter earlier in this book.

## Power BI Users or Custom Application Users

Power BI Embedded can work in two ways. You can use the users in your custom applications. These users might not have a Power BI account; they might not even have an Office 365 account. The other method is to leverage the Power BI accounts for the users and share content using Embedded. This latter method works similarly to the Secure Embed code.

# Implementing Power BI Embedded

If you are not a developer and don't want to get into the coding part of Power BI Embedded, you can skip the rest of this chapter (or send it to your web developer). If you are a developer and want to learn how to use Power BI Embedded, keep reading. The following articles are all about implementing Power BI Embedded and using REST API step by step.

- **Part 1: Register your application:** The first step is to register your application. Any application that wants to interact with Azure should be registered and authorized through a process. (See [radacad.com/integrate-power-bi-into-your-application-part-1-register-your-app](http://radacad.com/integrate-power-bi-into-your-application-part-1-register-your-app).)
- **Part 2: Authenticate:** Every application that wants to interact with the Azure environment must undergo an authentication process. This authentication process passes a Client ID (and sometimes a Client Secret) to Azure and gets an authentication code. From this authentication code, an access token can be fetched. This access token should be involved in every request sent from your application to Power BI and Azure. Without this access token, your requests will fail. (See [radacad.com/integrate-power-bi-into-your-application-part-2-authenticate](http://radacad.com/integrate-power-bi-into-your-application-part-2-authenticate).)
- **Part 3: Embed Content:** In this step, you learn about codes and the method to embed the content into your application after registration and authentication of the application. (See [radacad.com/integrate-power-bi-into-your-application-part-3-embed-content](http://radacad.com/integrate-power-bi-into-your-application-part-3-embed-content).)
- **Part 4: Refresh Dataset:** The Power BI REST API is not just for embedding content or getting a list of dashboards and reports. It also has many functions to work with datasets, gateways, and data sources. One of the many exciting features of that is that you can quickly refresh a dataset from an application. You can refresh it as many times as you want, with any frequency you want. You can refresh your dataset after ETL runs through a console application. Or you can have a service application that kicks off a refresh every minute! There is no limitation on your data refresh anymore. (See [radacad.com/integrate-power-bi-into-your-application-part-4-refresh-data-set](http://radacad.com/integrate-power-bi-into-your-application-part-4-refresh-data-set).)
- **Part 5: Data Sources and Gateway Management:** The REST API in Power BI is powerful. In addition to embedding content in Power BI and refreshing datasets from API, it also gives you many functions to work with gateways and data sources. With this API, you can set up new data sources, clone datasets, check a data source's credentials, get a list of all data sources on a gateway, and do many other operations. In other words, Data source management can be fully automated with REST API. (See [radacad.com/integrate-power-bi-into-your-application-part-5-data-source-management](http://radacad.com/integrate-power-bi-into-your-application-part-5-data-source-management).)
- **Part 6: Real-time Streaming and Push Data:** In this article, I explain one of REST API's functionalities: pushing data to a streaming dataset. I've previously explained how to do real-time streaming using Azure Streaming Analytics and Power BI. However, this post explains real-time functionality just with the REST API. You don't need Azure services. (See [radacad.com/integrate-power-bi-into-your-application-part-6-real-time-streaming-and-push-data](http://radacad.com/integrate-power-bi-into-your-application-part-6-real-time-streaming-and-push-data).)

## Summary

Power BI Embedded brings Power BI functionalities into your application. Imagine the analytical power of Power BI combined with your custom application. The result is revolutionary. You can leverage your users, interact with your web application's embedded object of Power BI, and implement many scenarios, including customized refresh and many other features. This method, however, works on a different licensing plan, and you need a web developer to enable it.

## CHAPTER 34



# Secure Embed

There is a method for sharing Power BI content called Secure Embed. This method of sharing is as simple as Publish to Web to implement, but it doesn't have the security issues of that method. You can securely publish Power BI reports through a custom web portal or application. This chapter explains Secure Embed in detail.

## Why Another Embedding Method?

To understand why there is another method of sharing as embedded, you need some background information about the two other methods of sharing—Publish to Web and Power BI Embedded. Briefly speaking, Publish to Web (see Figure 34-1) is a method of sharing that can be implemented very simply, with just a few clicks, but has security issues and some limitations. Power BI Embedded (see Figure 34-2) is a secure method of sharing, but it needs a developer's touch and code lines to implement it. The Secure Embed method combines these two to achieve something secure but simple! Let's look at the pros and cons of each method first.

### Publish to Web



**Figure 34-1.** Publish to Web

**Pros**

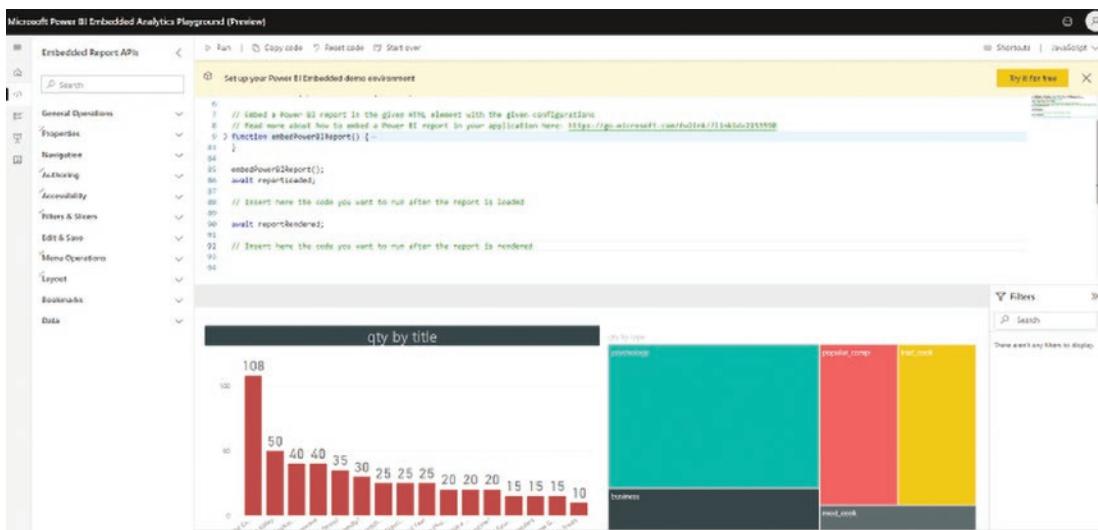
- It can be embedded in any web page easily with just a few clicks.

**Cons**

- Everyone can see what you share. No security is available.
- All report pages are visible.
- Users can share the link and data with others.
- Row-level security is not supported.

Publish to Web is a very quick way to share, but it's not suitable for confidential data because it cannot be secured. Users won't even need Power BI accounts to access the content. They can see the content and even share it with others. Everyone with that link will have access to the data.

## Power BI Embedded



**Figure 34-2.** Power BI Embedded

**Pros**

- You can share Power BI content securely with users.
- Users don't need to have Power BI accounts. It can work with Embedded licensing.
- Row-level security is supported.

**Cons**

- Implementation requires a web developer, because coding is involved in this method.

Power BI Embedded is a secure method of sharing that works perfectly for confidential data and when you have row-level security implemented. You can even share it with users who don't have Power BI accounts. However, the disadvantage of this method is the need for a web developer to take care of every change.

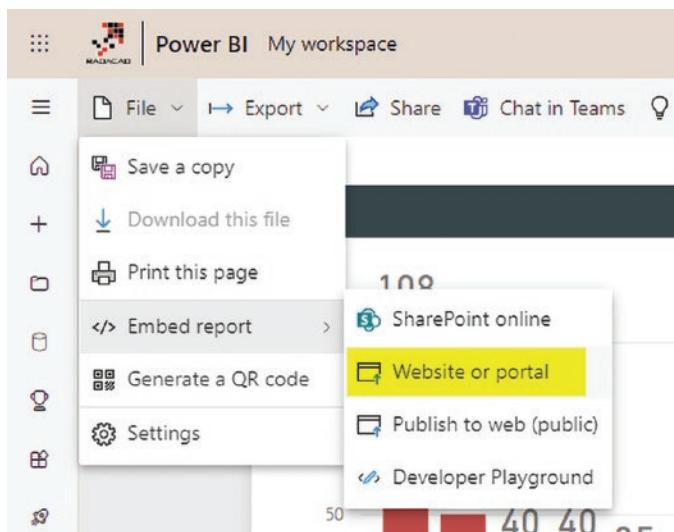
## Secure Embed: Good Features from Both Worlds!

Now that you know the advantages and disadvantages of the previous two methods, you can see the need for a method that can simplify the process of Power BI Embedded and be secure enough for confidential data. That method is called Secure Embed. Using this method of sharing, the process of creating the embed code is as simple as doing it for the Publish to Web method. However, this method is secure, as users need to be authorized. They either need to have Power BI pro accounts or be part of an embedded licensing authorized by the person sharing the report. So with this method, you get good features from both methods! With Secure Embed, you can simply share Power BI content securely in a web portal or application. The next section explains how this method works.

## Using Secure Embed

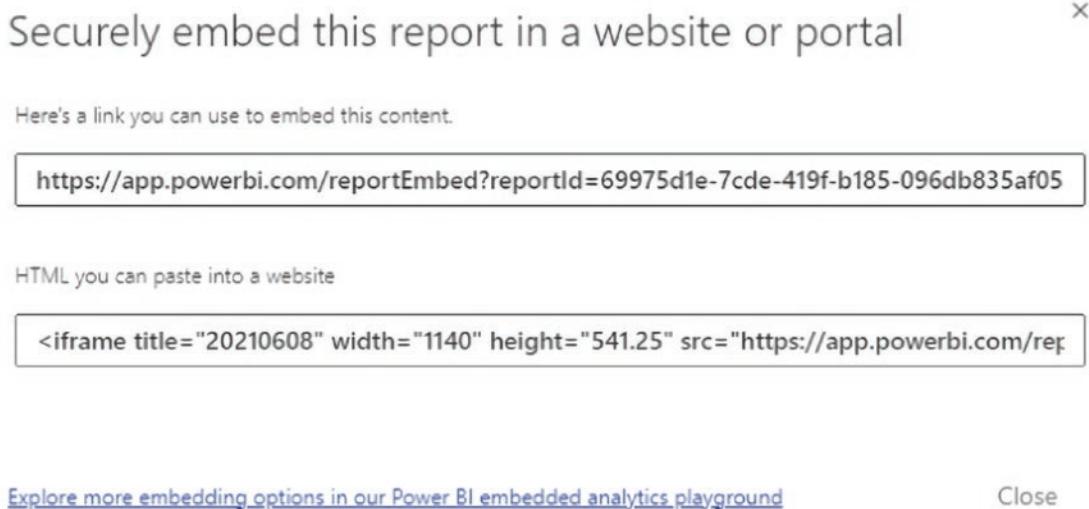
To use the Secure Embed method, go to a report in the Power BI Service. From the File menu, select Embed report, and then Website or Portal, as shown in Figure 34-3.

Secure Embed works only with reports in the Power BI Service, not with dashboards.



**Figure 34-3.** Secure Embed in the Power BI Service

As mentioned, this method makes creating the embed code very simple! With just a click, you get the embed code and the URL to share the reports on any platform you want (see Figure 34-4).

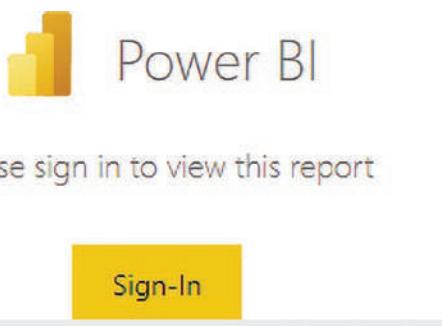


**Figure 34-4.** Getting the Secure Embed link from the Power BI Service

There are two outputs for the Secure Embed code—the URL and the HTML embed code. Using the URL, you can directly share the reports with a link. You can share the link with any users you want. Using the HTML code, you can embed this report in web pages and custom web portals.

## Who Can View the Report?

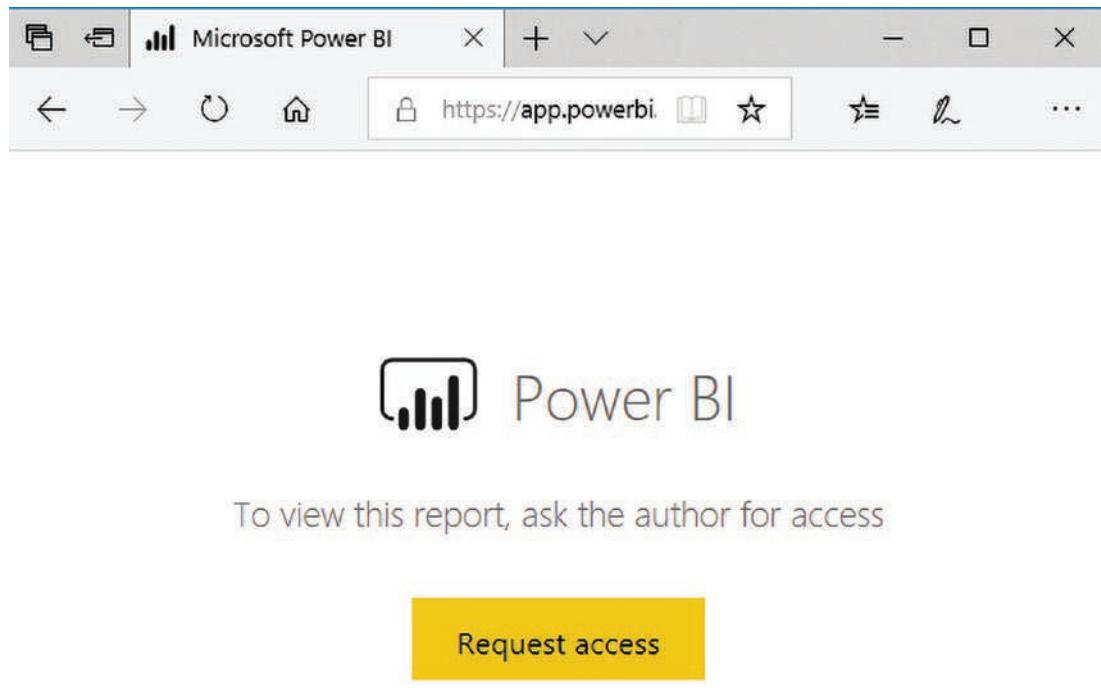
You've seen that sharing using this method is simple, but who can view the report? Is it available to everyone with the URL link or access to the page with the embed code? This method of sharing is not the same as Publish to Web. Users need a Power BI Pro (or PPU) account or a report to be shared through dedicated capacity licensing. If you send the URL to someone, they get a message that says, Please sign in to view this report, as depicted in Figure 34-5.



**Figure 34-5.** Users of the Secure Embed code must sign in

Reports shared through Secure Embed work with Power BI Pro (or PPU) user accounts of (users need to log in) or by adding the report under the embedded capacity.

This is not the whole story. If a user logs in, they still might not be authorized to see the report because it is not shared with them. If a user logs in through a Power BI Pro account but still doesn't have access to the report, they will see a message that says, To view this report, ask the author for access (see Figure 34-6).



**Figure 34-6.** The author must grant access to certain users to view a report

If users are accessing the report through Secure Embed using Power BI Pro accounts, the report should be shared with them by the report owner.

There are multiple ways to share accounts. The most straightforward way is via report sharing. To use report sharing, click the Share button on the report in the Power BI Service. You can also do this using Manage Permissions, as demonstrated in Figure 34-7.

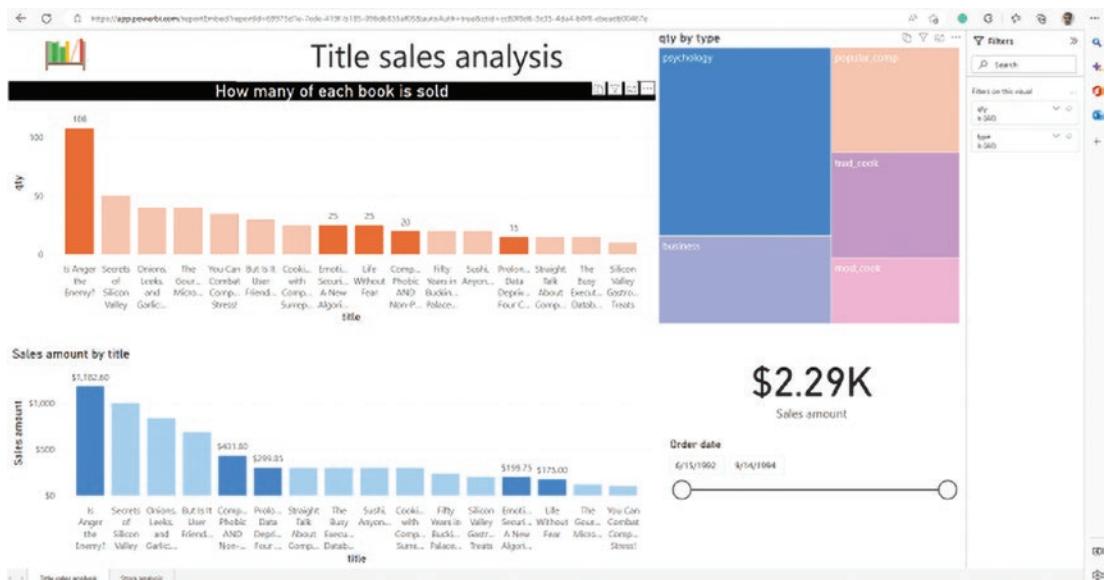
The screenshot shows the Power BI service interface. At the top, there are navigation links for '+ New', 'Upload', 'All', 'Content', and 'Datasets + dataflows'. Below this is a table listing eight reports. The columns are 'Name', 'Type', and 'C'. The first report, '00 Static', has its context menu open, with the '...' button highlighted by a red box. The menu options are: 'Analyze in Excel', 'Delete', 'Quick insights', 'Save a copy', 'Settings', 'View usage metrics report', 'View lineage', 'Create paginated report', and 'Manage permissions'. The 'Manage permissions' option is also highlighted by a red box.

Name	Type	C
00 Static	Report	R
00 Static		R
01 Dynamic		R
01 Dynamic		R
20210608		R
20210608		R
20210608		R
20221115		R

**Figure 34-7.** Manage Permissions on the Power BI report

This access can also be set up through workspaces or other sharing methods.

After you share the report, users can access it using Secure Embed. Figure 34-8 shows how this looks.



**Figure 34-8.** A Secure Embed report opened in a web browser

Users will have access to all the pages. At the time of writing this chapter, isolating access at the page level is impossible.<sup>1</sup> Access to the report means access to all pages in the report. This applies to all methods of sharing in Power BI.

## Licensing Needs

This method of sharing is not free. You need to use one of these licensing paths:

- Using Power BI Pro or PPU (Premium per User)
- Using Power BI dedicated capacity (Premium or Embedded)

To use Power BI Pro or PPU, users need to have accounts, and then you can simply share the report with them. Another option is to use dedicated capacity. An important point here is that even if you have embedded licensing, the user has to log in to the page to see the Secure Embed content. This is different from Power BI Embedded, where you can use Power BI users or custom application users. A custom application user setting won't work with Secure Embed.

## Advantages of Secure Embed

This section explains the advantages of the Secure Embed method.

### Sharing with only a few clicks; no developer is needed

As you have seen in this chapter, I haven't written a single line of code. This method works by using the Secure Embed option. All you need to do is use the URL output of Secure Embed or embed the HTML code on a web page. Unlike Power BI Embedded (which needs a developer's touch), this method is very simple to implement.

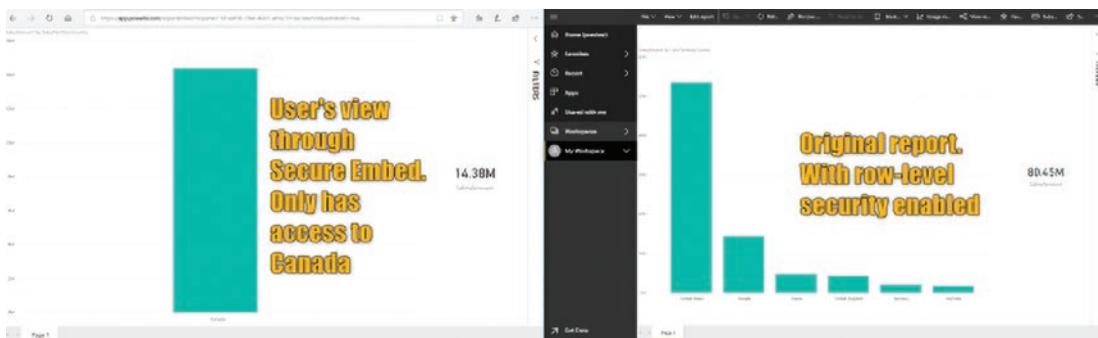
<sup>1</sup>There is, however, a workaround. See [radacad.com/page-level-security-workaround-in-power-bi](http://radacad.com/page-level-security-workaround-in-power-bi)

### Secure sharing

Just by its name, Secure Embed, you can guess that it's secure. You also saw in this chapter, through an example, how the content is secured for those users who are authorized to see the content. Unlike Publish to Web, this method provides a secure method of sharing.

### Row-level security is supported

Publish to Web doesn't support row-level security because there is no concept of the user logging in because no login is required. However, using Secure Embed, only authorized users will have access to the report, and as a result, row-level security is possible (see Figure 34-9).



**Figure 34-9.** The Publish to Web does not support row-level security

## Limitations

Some limitations of Secure Embed are related to the content that can be shared. For example, ArcGIS Maps is not currently supported, as you can see in Figure 34-10.



**Figure 34-10.** ArcGIS Maps are not supported in Secure Embed

Secure Embed, at the moment, doesn't support dashboards or paginated reports.

Another significant limitation of Secure Embed is that it won't work with custom applications' user setups. If you use the Secure Embed method, you must use Power BI user accounts. If you want to use a custom application's userbase, the better choice is Power BI Embedded.

## Scenarios for Using this Method

This method provides a quick way to share data using an HTML embed code. If you are using SharePoint Online, there is a quick method to use for that (read the next chapter), but what about other web applications? What about SharePoint on-premises? This method can be a great option to securely share quickly through a web page.

## Summary

Secure Embed is neither Publish to Web nor Power BI Embedded. It is combines features from both of these tools. Using Secure Embed, you can share your reports through a web portal with just a few clicks. However, unlike Publish to Web, Secure Embed is a secure method of sharing, because only authorized users have access to the data. As a result, this method also supports row-level security-enabled reports. However, there are scenarios where Power BI Embedded is a better option (for example, if you want to use the custom application's userbase).

## CHAPTER 35



# Embed in SharePoint Online and Teams

In this chapter, you learn about another method of sharing called Embed in SharePoint Online. Embedding in SharePoint Online is an excellent method for sharing Power BI content through a SharePoint portal. Because Power BI and Office 365 accounts are bound to each other, this method of sharing is prevalent for SharePoint users. You can use SharePoint as the portal. Power BI content can then be easily shared with Office 365 users through that portal.

## How to Use Embed in SharePoint Online

To use this method, you must have a Power BI report published in the service. This method only works with Power BI reports (not dashboards). To share a report using this method, after logging into the service and opening a Power BI report, click the File menu. Under Embed Report, choose SharePoint Online (see Figure 35-1).

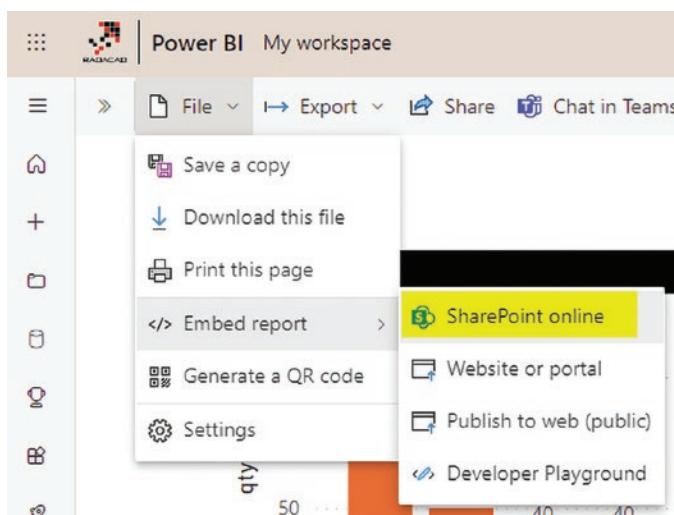
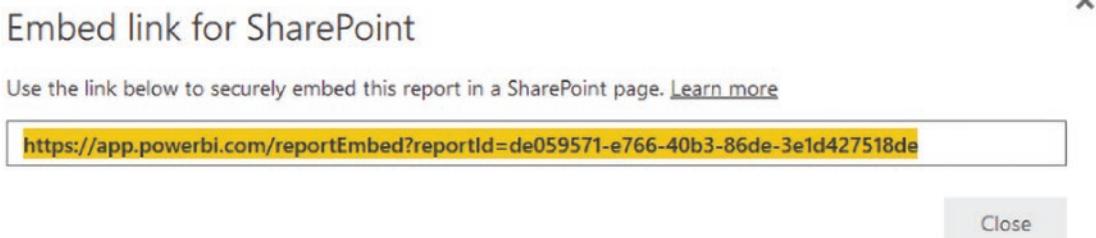


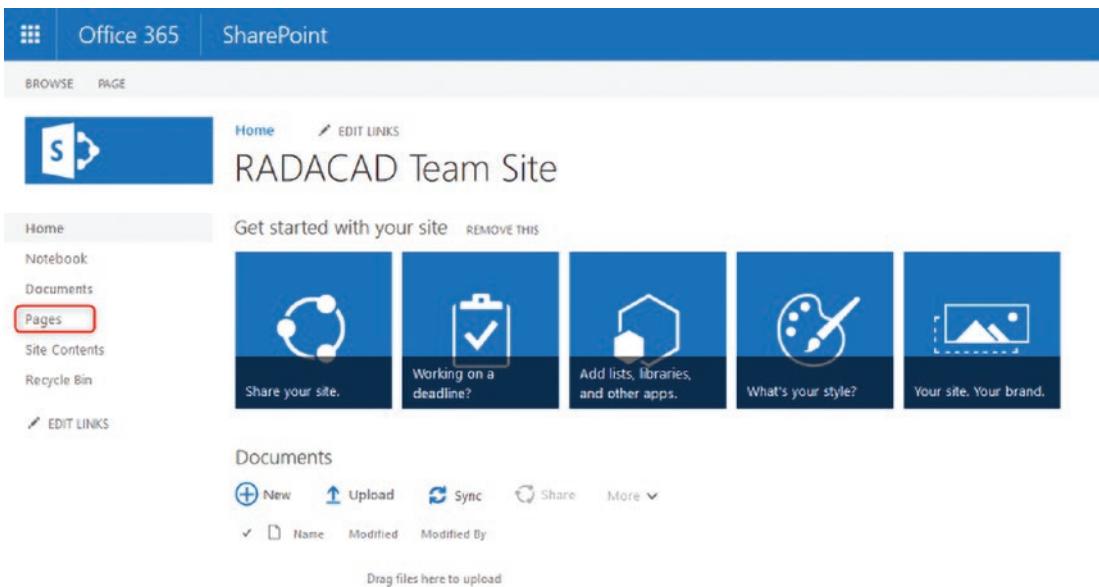
Figure 35-1. Embedding a Power BI report in SharePoint Online

The next step generates a link that can be used in SharePoint. Just copy the link from this step (see Figure 35-2).



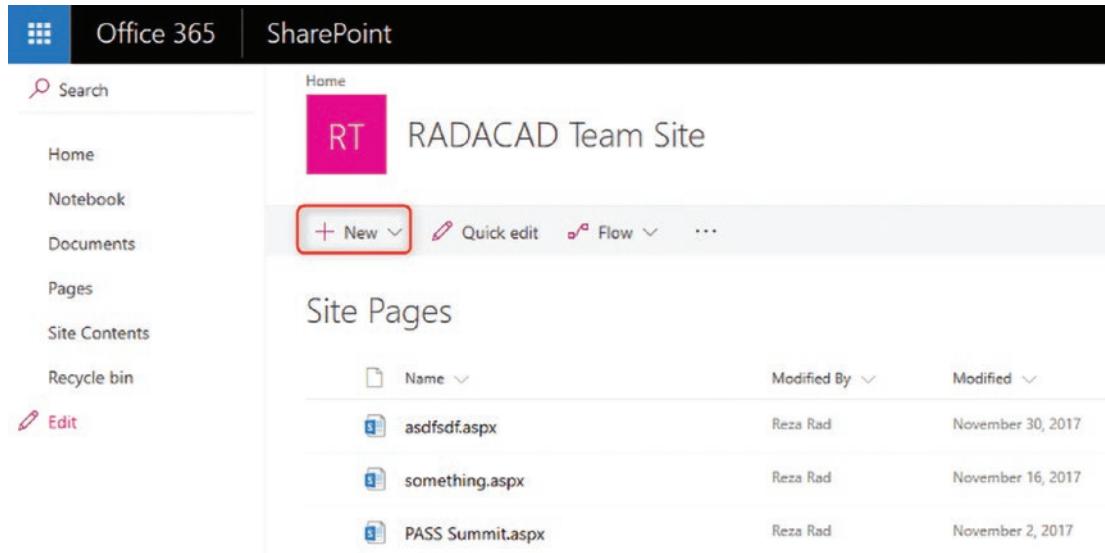
**Figure 35-2.** Link generated in SharePoint

The URL is needed in SharePoint Online for embedding the Power BI report. Log in to your SharePoint Online tenant. Go to the Pages section, as shown in Figure 35-3.



**Figure 35-3.** Navigating to the Pages section in SharePoint

Create a new site page (or edit an existing page), as demonstrated in Figure 35-4.

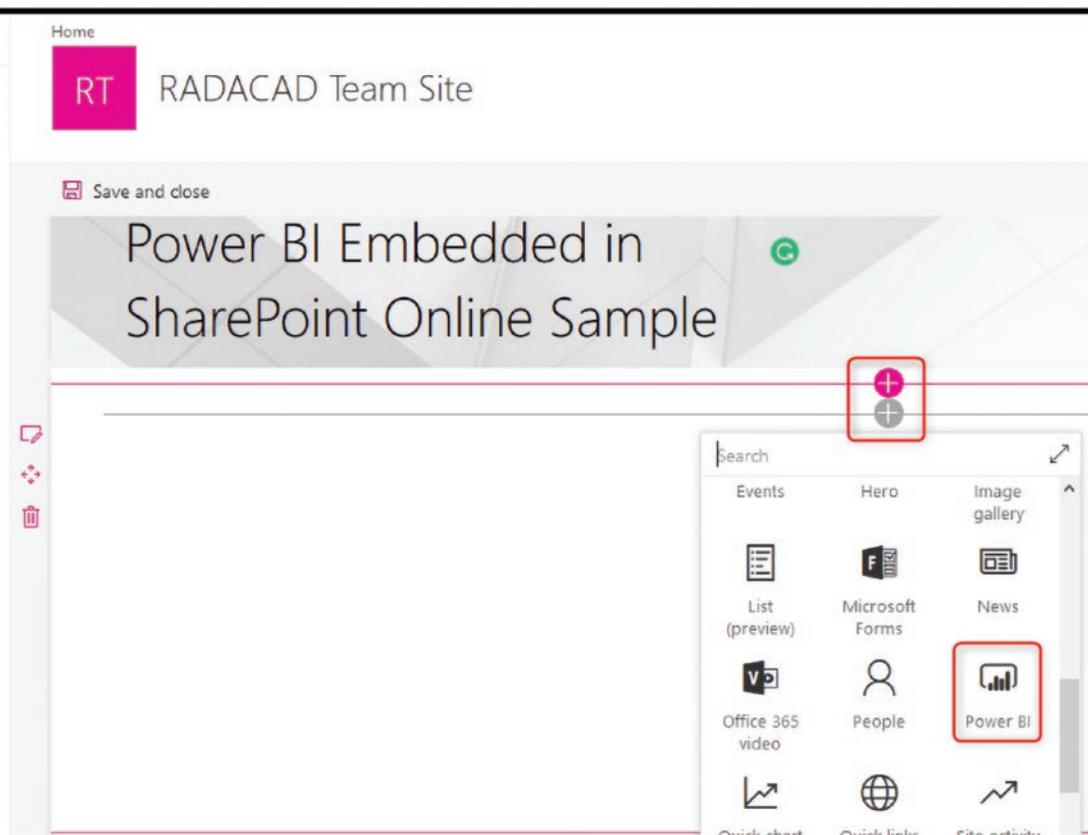


The screenshot shows the SharePoint Online interface. At the top, there's a navigation bar with the Office 365 logo, the word "Office 365", and the "SharePoint" logo. Below the navigation bar is a left-hand navigation pane with links for "Search", "Home", "Notebook", "Documents", "Pages", "Site Contents", "Recycle bin", and an "Edit" link. The main content area is titled "RADACAD Team Site". It features a "Home" button with the letters "RT" and a "New" button highlighted with a red box. Below these are "Quick edit", "Flow", and a three-dot menu. The title "Site Pages" is displayed, followed by a table listing three items:

Name	Modified By	Modified
asdfasdf.aspx	Reza Rad	November 30, 2017
something.aspx	Reza Rad	November 16, 2017
PASS Summit.aspx	Reza Rad	November 2, 2017

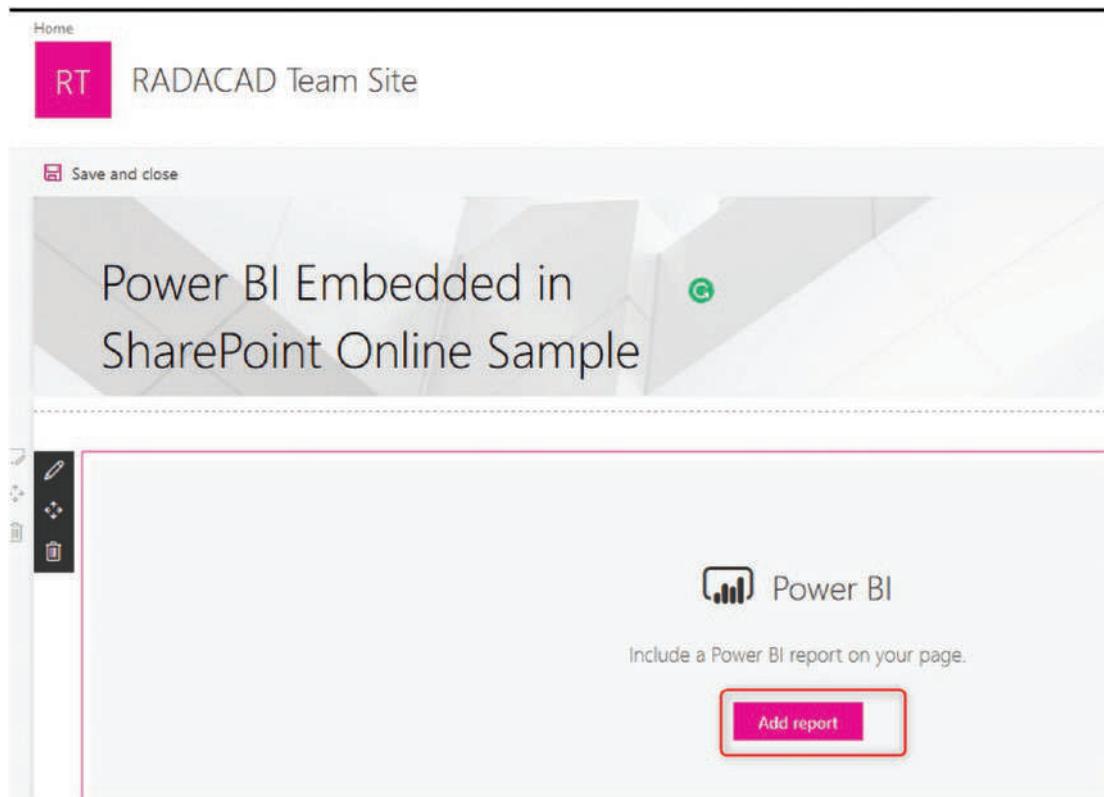
**Figure 35-4.** Creating a new site page in SharePoint

You can name this page something like “Power BI Embedded into SharePoint Online,” then click Add Icon to add a new object. From the All Items list, select Power BI (see Figure 35-5).



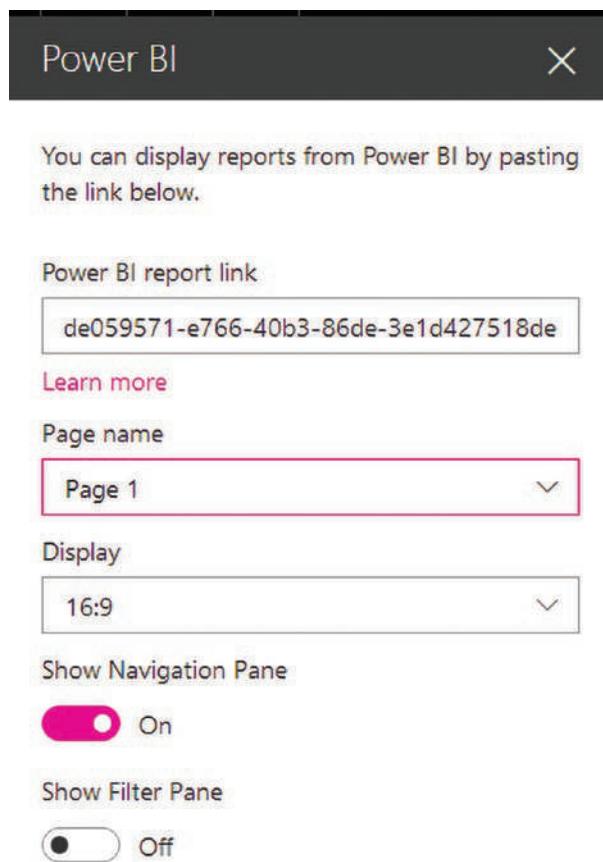
**Figure 35-5.** Naming your new site and selecting Power BI

The Power BI component is now inserted into your page. As shown in Figure 35-6, you can click Add Report. (At the time of writing this chapter, you can only embed reports, not dashboards, into SharePoint Online.)



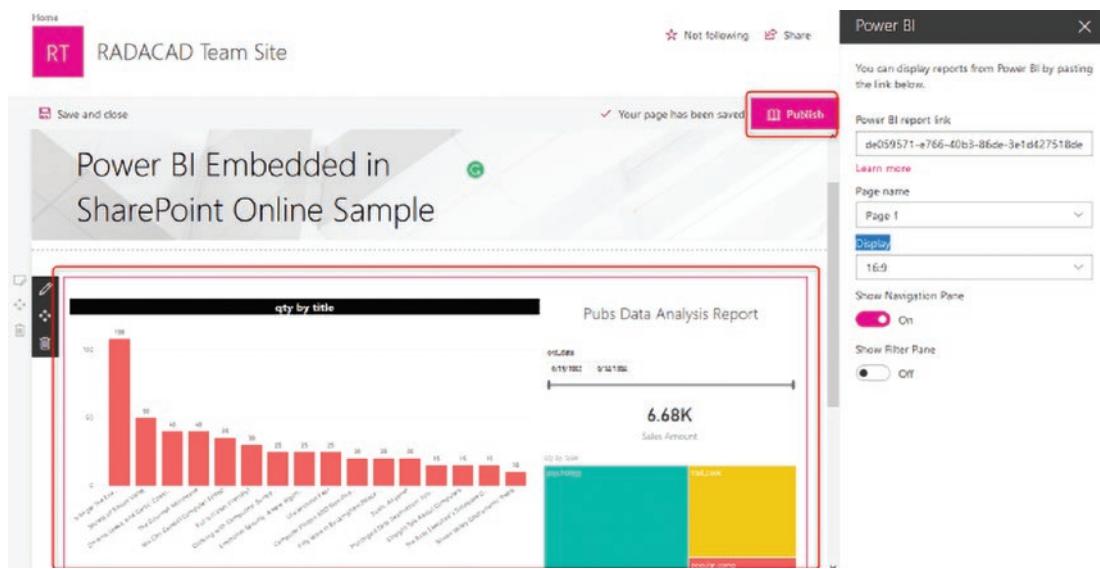
**Figure 35-6.** Adding a report to your new team site

You can now paste the embedded URL from the previous step for the Power BI report into this section and select the page (if the Power BI report has multiple pages), as shown in Figure 35-7.



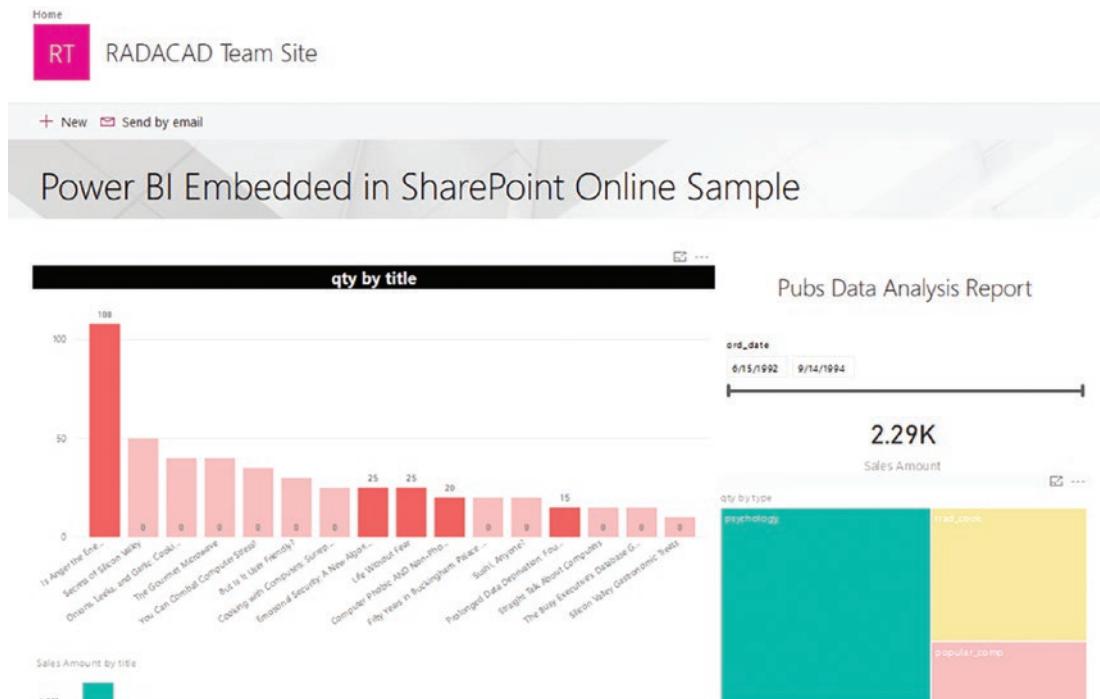
**Figure 35-7.** Pasting in the embedded URL

As soon as you make these changes, you'll see the report preview on the page (see Figure 35-8).



**Figure 35-8.** The results generated by pasting in the embedded link

You can now publish the page. Once the page is published, the Power BI embedded part will be part of it. The Power BI embedded in the SharePoint Online page will be interactive, like a standard Power BI report is (see Figure 35-9).



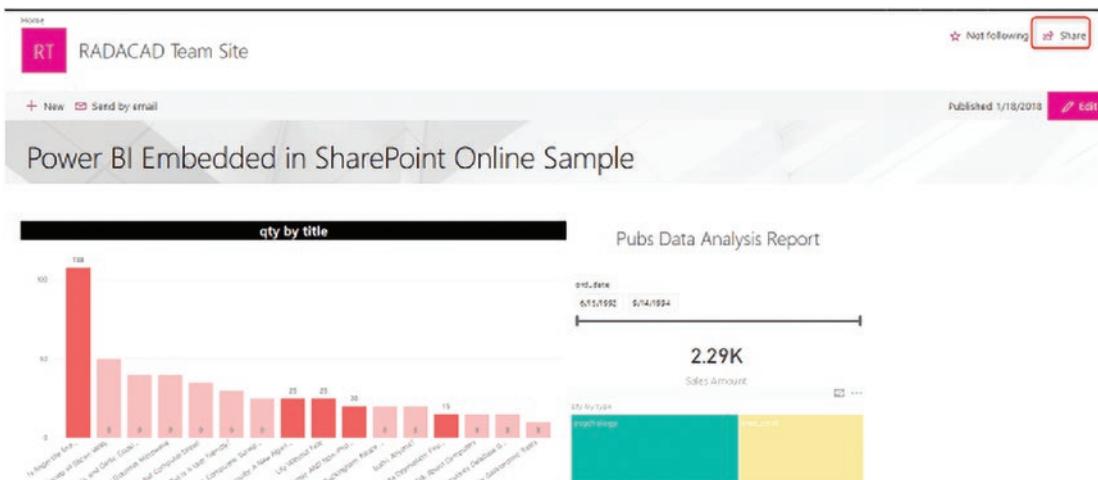
**Figure 35-9.** The published page displaying the Power BI module

# Sharing a SharePoint Page with an Embedded Power BI Report

After embedding the Power BI report into the page, you can share it with others. A SharePoint page can be shared with other SharePoint Online users (who usually are also Office 365 users). However, one important note is that if a Power BI report is embedded in this page, users need to have Power BI accounts to see it, and their account should have access to that Power BI report. You must manage permission in two locations—from a SharePoint Online page and from the Power BI report.

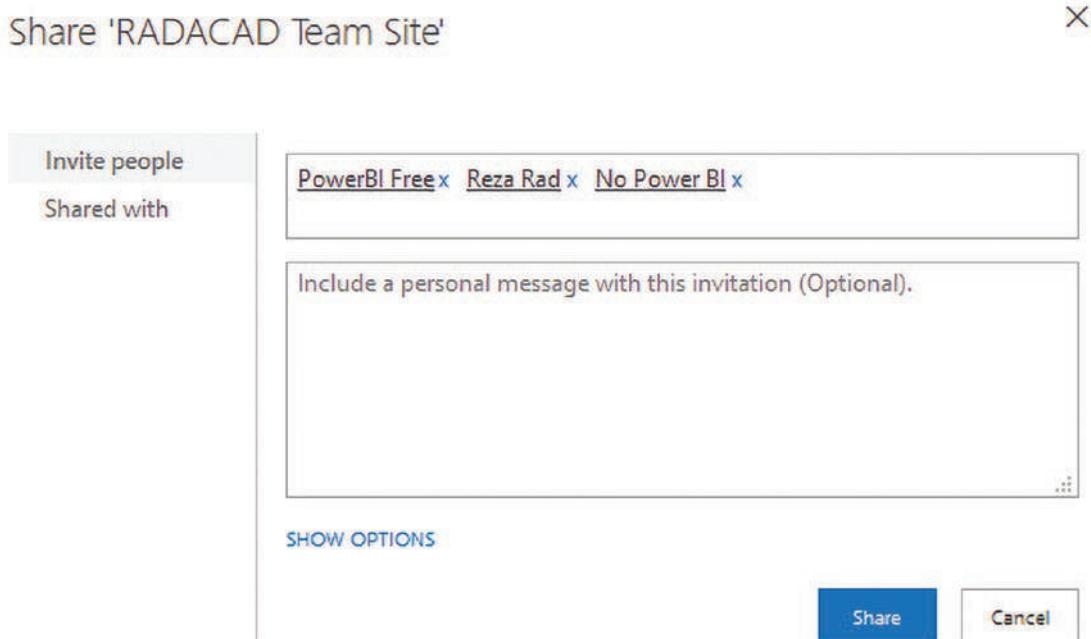
## Access to SharePoint

You can share the page with other users simply using the Share icon at the top-right side, as in Figure 35-10.



**Figure 35-10.** Sharing the page with other users

You can then add people from Office 365 accounts to the list (see Figure 35-11).



**Figure 35-11.** Adding people to your team site

If you take a closer look at Figure 35-11, you'll see that I shared it with three types of accounts:

- Power BI Pro account, which is the account named Reza Rad in the figure
- A Power BI free account
- An Office 365 account with no Power BI license

## Power BI Permissions

Users also need to be permitted to access the Power BI report. You can use the Manage Permissions part of the dashboard to handle this, as shown in Figure 35-12.

The screenshot shows a list of datasets in a Microsoft Power BI environment. At the top, there are navigation links for '+ New', 'Upload', 'All', 'Content', and 'Datasets + dataflows'. Below this is a table with columns for Name, Type, and a series of icons. The first dataset, '00 Static', has its 'More options' menu open, revealing several options: 'Analyze in Excel', 'Delete', 'Quick insights', 'Save a copy', 'Settings', 'View usage metrics report', 'View lineage', 'Create paginated report', and 'Manage permissions'. The 'Manage permissions' option is highlighted with a red box.

Name	Type
00 Static	Report
00 Static	R
01 Dynamic	R
01 Dynamic	R
20210608	R
20210608	R
20210608	R
20221115	R

**Figure 35-12.** Managing permissions of your new team site

Choosing Manage Permissions shows you a detailed access list to the dashboard, reports, and datasets. You will see related reports and datasets on the left side of the Manage Permissions section (see Figure 35-13). You can click the report.

The screenshot shows the 'Manage Permissions' page in the Power BI Service. On the left, there's a sidebar with 'Related content' sections for 'Dashboards', 'Workbooks', and 'Datasets'. A red box highlights the '2021Q608' report under the 'Dashboards' section. To the right, there's a table titled 'Access' with columns for 'Links', 'Who has Access', 'Permissions', 'Create', and 'Email Address'. It lists two entries: one for 'People in your organization' with 'Read, Refresh' permissions and another for 'Reza Rad' with 'Read' permissions. Both entries have an 'Email Address' column with 'Reza.RAD@RADACAD.com'.

**Figure 35-13.** A detailed access list displayed in Manage Permissions

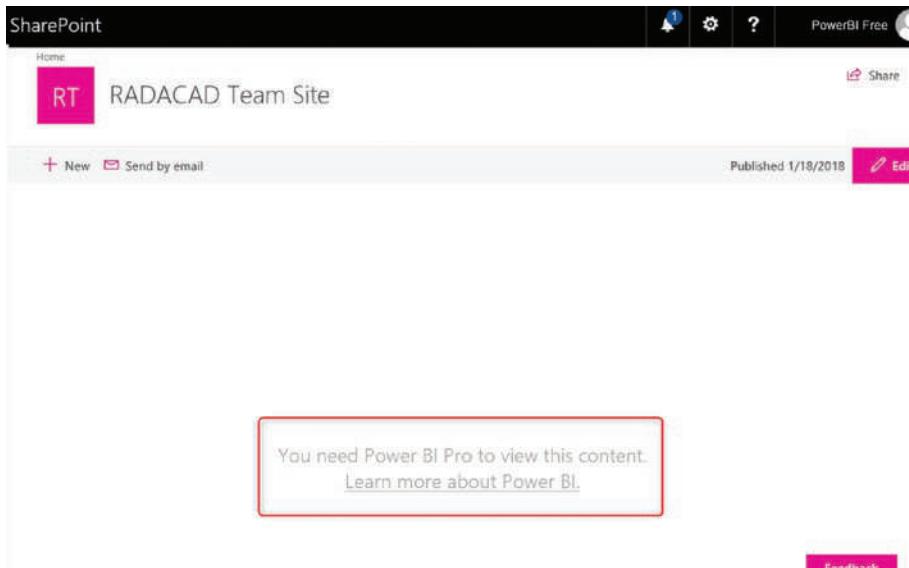
You will see the permission specified for that object by clicking a report or dataset. Now, let's see what each user will see when they log in to this page.

A Pro user, when the content is shared with them in the Power BI Service and the page is shared with them in SharePoint Online, will see the full page (see Figure 35-14).

The screenshot shows a SharePoint page for the 'RADACAD Team Site'. On the left, there's a Power BI visualization displaying various charts and graphs. On the right, there's a 'My accounts' sidebar for 'Reza Rad' (reza@radacad.com). The sidebar includes links for 'My profile', 'My account', and 'Sign out'.

**Figure 35-14.** A page that is fully accessible to a Pro user

Other users, however, can't see the entire page. If the users don't have a Pro subscription, or if it is not part of a Power BI premium capacity, they can't see the content and will see an error, as shown in Figure 35-15.



**Figure 35-15.** Non-pro users cannot access all elements of the page

---

Users need to have a Power BI account. The Power BI report should be shared with them in the service, and the SharePoint page should be shared with them. Power BI accounts can use this page only when they are part of a Premium capacity or Power BI Pro accounts.

---

## Advantages and Disadvantages of Embedding in SharePoint Online

Like all other methods, this method has its pros and cons.

### Advantages of Embed in SharePoint Online

#### One portal for all the content

With SharePoint Online, you can share other documents as well. Why not use it for all other documents and the Power BI report? You can have one portal, which is the central sharing portal for your Office 365 tenant. Users usually love integrity.

#### Embedding is simple

Unlike with Power BI Embedded, embedding in SharePoint Online is easy. You just get the URL and embed it into a Power BI object in SharePoint Online. You don't need to write a single line of code for that purpose, but with Power BI Embedded, you need a developer.

## Disadvantages of Embed in SharePoint Online

### The Power BI Service is underutilized

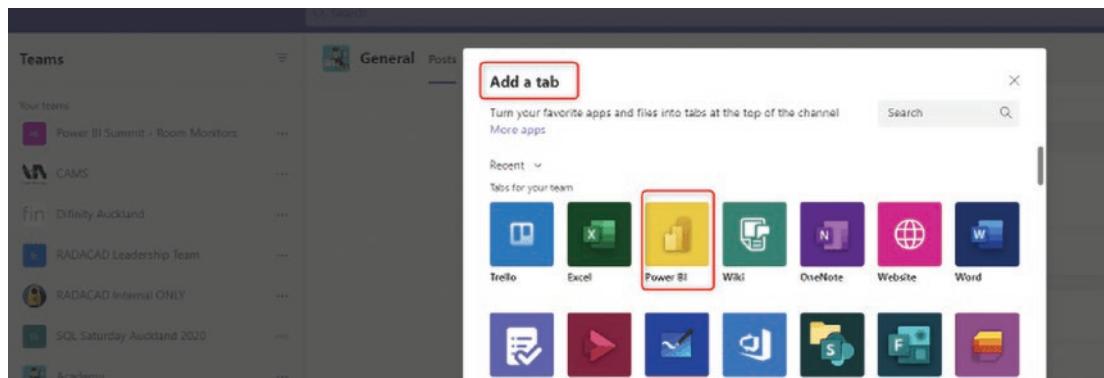
One of the great aspects of Power BI components is the service. If you use SharePoint embedded, users will use that as the portal for reports. The Power BI Service has many exciting features that may not be well used in this scenario, such as alerts, feature dashboards, the dashboard itself, Q&A, and many other items. Users can still log in to the Power BI service and see the report, but the experience you created for them with SharePoint Online is not there.

### Two places for managing permissions

At the time of writing this chapter, you need to manage permissions in the Power BI Service and in SharePoint Online. This takes time for maintenance and reconciliation to check whether the people who have access to the page are permitted to read the report.

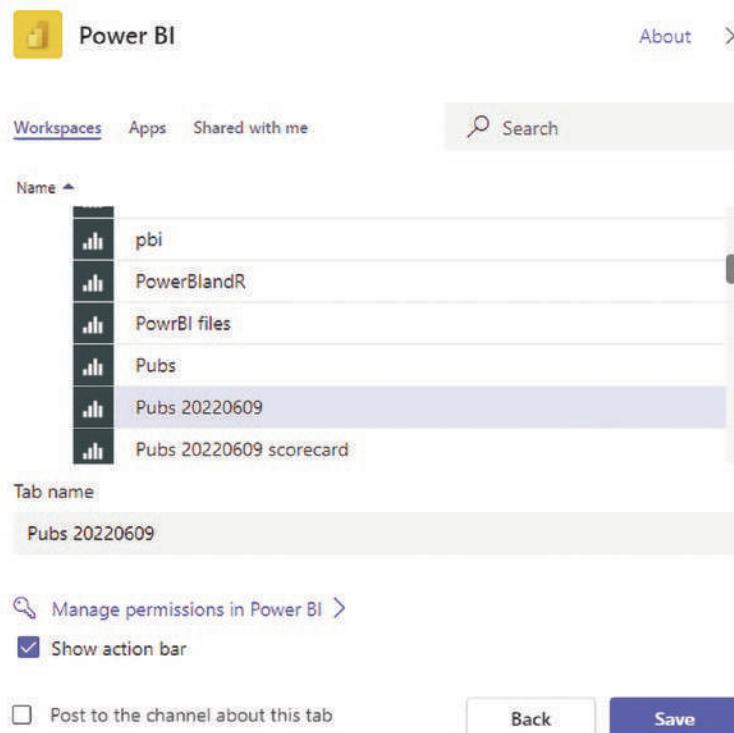
## Power BI and Teams

Power BI also integrates with Teams. Instead of starting data from the Power BI service, you can go to a Teams channel tab, add a Power BI item, select the report, and share it (see Figure 35-16).



**Figure 35-16.** Adding a Power BI tab to Teams

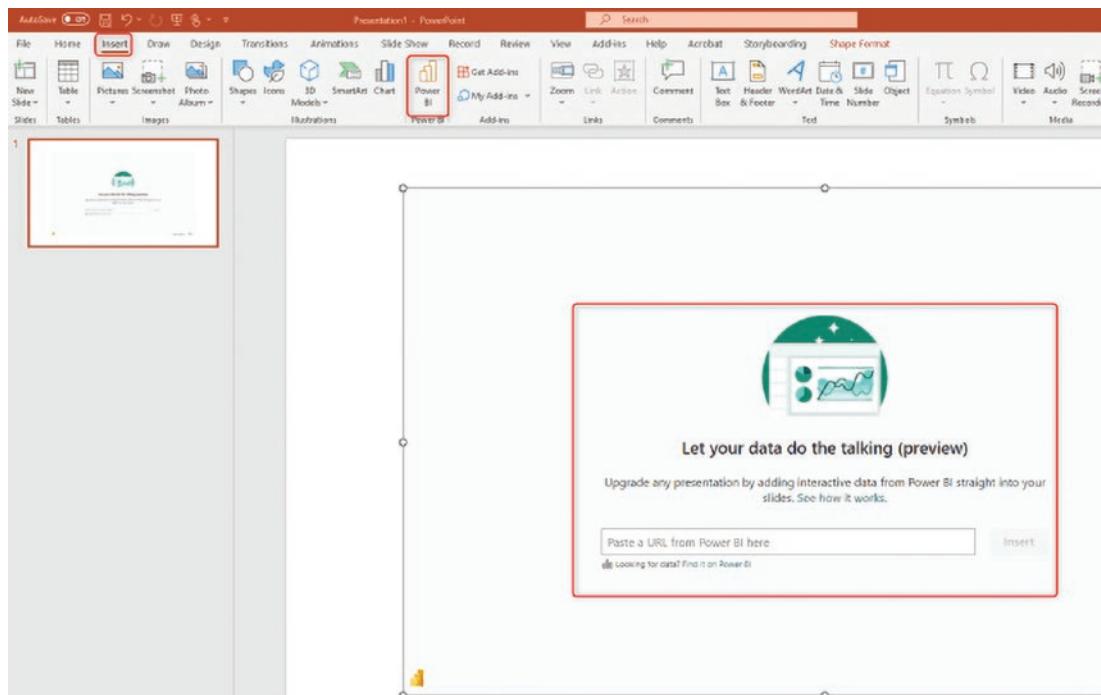
There is also a link to set the Manage Permissions from there (see Figure 35-17).



**Figure 35-17.** Embedding Power BI reports in Teams

## Power BI and PowerPoint

Power BI can also be integrated and embedded into Microsoft PowerPoint. This process can be done by selecting the Insert tab and selecting Power BI. You can then paste the URL of the Power BI report (see Figure 35-18). All of these methods use the Secure Embed option.



**Figure 35-18.** Embedding Power BI reports in PowerPoint

## Summary

In summary, embedding in SharePoint Online is an easy way to add an interactive Power BI report to a SharePoint Online page. This method allows you to have a central SharePoint Portal for all your content and Power BI reports. Users need to be part of a paid Power BI subscription to use this feature. They will lose the Power BI service's functionalities because they are not available in SharePoint.

Power BI report also integrates well with Microsoft Teams and PowerPoint. The concept is similar; it uses Power BI user access to shared content.

## CHAPTER 36



# Comparison of Sharing Methods for Power BI

You have published your Power BI report and want to share it with others. You can share it using various methods—basic sharing, workspaces, apps, Publish to Web, Power BI Embedded, Secure Embed, and SharePoint Online. The wide range of sharing methods makes it confusing to choose the best one. It is very important to determine which method is best before sharing your content with users. In this chapter, you learn about all the different methods of sharing, the pros and cons of each, and scenarios in which to use each method. By the end of this chapter, you'll be able to choose the best sharing mechanism for sharing your Power BI reports.

## Types of Sharing Methods

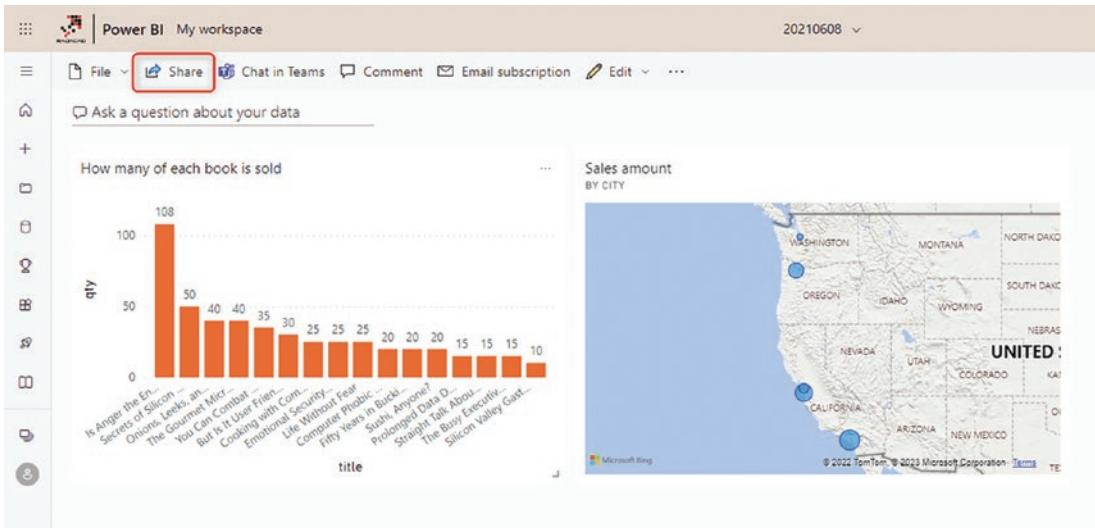
This chapter only covers the sharing methods that are interactive and cloud-based. It doesn't cover sharing a \*.pbix file with others, which is a method of sharing, obviously, but not a proper sharing method. It doesn't cover exporting a Power BI report as a PDF and sharing it with others because that is not interactive. It also doesn't cover taking screenshots of a Power BI report and sharing it with others. Also, it doesn't cover sharing through on-premises solutions using the Power BI Report Server.

This chapter focuses on the cloud-based, interactive methods of sharing Power BI reports. It covers the following sharing methods:

- Basic sharing
- Workspaces
- Power BI apps
- The Publish to Web option
- Embed in SharePoint Online
- Power BI Embedded
- Secure Embed

## Basic Sharing for Dashboards and Reports

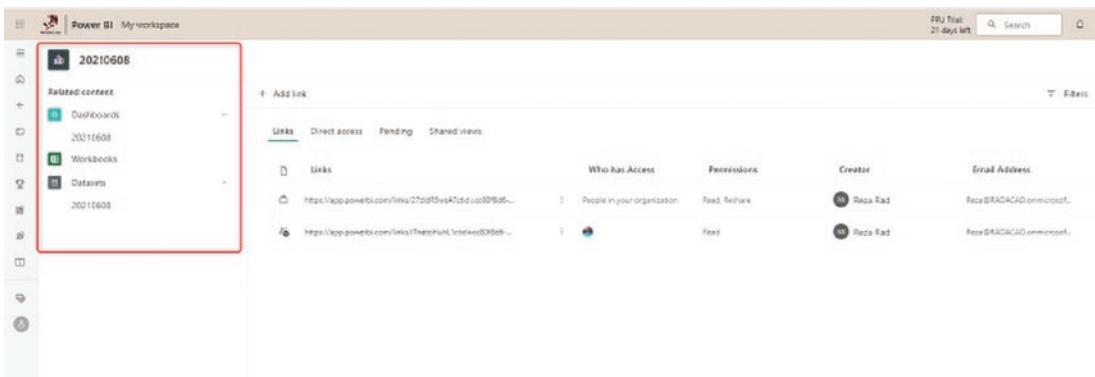
This is one of the most common ways to share Power BI content. However, it is not always the best way. Basic sharing is a very simple and easy-to-use method. You simply click the Share button in the dashboard or report and then share it with other users, as shown in Figure 36-1.



**Figure 36-1.** Dashboard or report sharing

This method of sharing gives users three levels of access—Read-only, Reshare, and Build.

By default, when you share a Power BI dashboard using this method, the report and the dataset are shared as well. However, you can go to Manage Permissions and set up permission for every item if you want, as depicted in Figure 36-2.



**Figure 36-2.** Manage Permissions on the Power BI content

Users can easily click the Shared with Me section of their profile, and they will see all the reports and dashboards shared with them (see Figure 36-3).

The screenshot shows the Power BI 'Browse' interface. On the left, there's a sidebar with navigation icons: Home, Recent, Favorites, and Shared with me (which is highlighted with a red box). The main area is titled 'Shared with me' and lists five reports. Each report entry includes a small thumbnail icon, the report name, its type (Report), the date it was shared, and the owner. The reports are as follows:

Name	Type	Shared	Owner
[Redacted]	Report	8/13/19, 8:36:58 AM	sample DSCup WS
[Redacted]	Report	1/2/20, 4:59:02 PM	Public Reports
[Redacted]	Report	1/20/20, 6:38:51 AM	General Dataset Wor...
[Redacted]	Report	1/11/21, 1:15:59 PM	Public Reports
[Redacted]	Report	8/13/21, 7:57:14 AM	P AT

**Figure 36-3.** Power BI reports that are shared with me

This method of sharing has some pros and cons, discussed next.

## Advantages of Dashboard Sharing

Dashboard sharing is the most basic way to share content in Power BI. This method is quick and easy to set up. You don't need to go through a lot of steps to set up dashboard sharing. The ability to share it very quickly makes this method the most common method of sharing for testing purposes.

If you create Power BI content and want to share it with others easily for testing purposes, dashboard sharing is a good option.

## Disadvantages of Dashboard Sharing

Dashboard sharing is simple; however, it has many drawbacks, which make it unwise in a production environment. I do not recommend using this method to share Power BI content with users in a production environment because of the reasons mentioned next.

### No Edit Access

With dashboard sharing, you cannot specify edit access. End users should never have edit access; however, if you are working with a team of developers and you want to provide them with access to edit the content, you cannot do that with dashboard sharing.

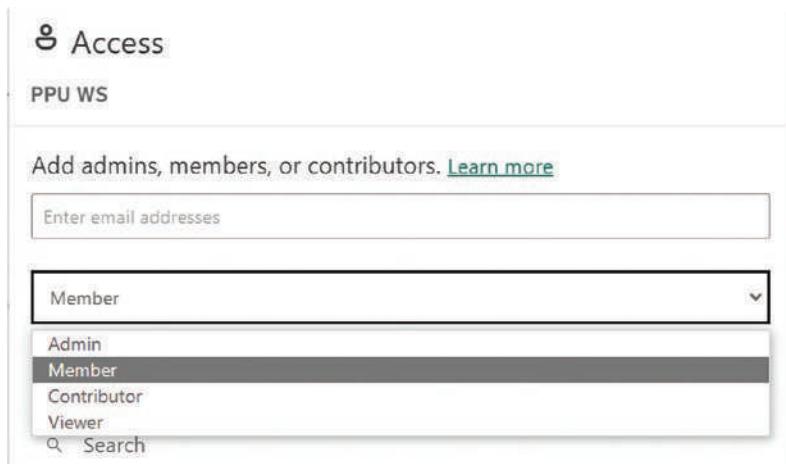
### Share objects one at a time

You can only share one dashboard at a time. What if you wanted to share hundreds of dashboards? You must go to each dashboard and share items individually. Sharing every dashboard would add a lot of maintenance overhead to your work. The best method would be having all contents under a group and sharing them with others at once.

## Workspaces

Workspaces are created to address the main two limitations of basic sharing—lack of edit access and not being able to share multiple objects. With a workspace, you can share as many items as you want in that workspace at once. You can also determine the access level of the workspace to be Edit or Read-Only (through four access levels). Workspaces, because of these two features, are heavily used as collaborative development environments.

There are four levels of workspace access—Admin, Member, Contributor, and Viewer—as shown in Figure 36-4.



**Figure 36-4.** Access levels for the users of the Power BI workspace

Workspaces also have their advantages and disadvantages. Let's check these out.

## Advantages of Workspaces

### Sharing multiple datasets with the team

You may have shared a dashboard with a couple of your colleagues in your organization, but after a few weeks, a dashboard comes up, and you share that dashboard with them. A couple of months later, another team member asks for access to a Power BI dataset to create a report and share it with others. Power BI workspaces enable you to share content (dashboards, reports, and datasets) with all group members. You don't have to share each dashboard with each user; groups make it easy for you.

### Sharing all types of objects

The dashboard only allows sharing of the Power BI report, dashboard, and dataset. However, sharing through the workspace allows all the content to be shared. This includes, but is not limited to, dataflows, datasets, datamarts, dashboards, reports, metrics, and paginated reports.

### Multiple workspaces

It is hectic when you are part of multiple teams, and each team has its own dashboards, reports, and datasets. Your Shared with Me section in Power BI could have hundreds of items. Power BI workspaces create a separate environment for all members of the group. You can easily switch between workspaces in Power BI.

### Isolated user/group administration

When you share content with an individual in the organization, and that person leaves the company or is replaced by someone from another team, you have to remove sharing from a previous user account and assign it to the new user account. The best practice is to share content with groups. Members of workspaces can then easily be managed by an administrator. Power BI workspaces can be shared with Office 365 groups. Once you use a group in Power BI, it is only the admin's task to add/remove members.

### Best developer environment

You need an environment to share multiple Power BI content for a team of developers. Everyone needs to have edit access to the content provided by the team. Power BI workspaces are a perfect solution for the development environment. You can create a workspace as a development environment and then share it with other developer team members with edit access. Then you all have access to the same content in your development workspace.

---

Power BI workspaces are a perfect solution for development environments.

---

## Disadvantages of Workspaces

Workspaces are useful, but there are some drawbacks.

### **Not suitable for end users**

Workspaces are not suitable for sharing content with end users. You may wonder why that is. You can give users read-only access to the content. However, this is half of the requirement. In an end-user sharing environment, one of the primary requirements is to have the development and user environment separated.

Assume that you created a workspace and shared it with end users. If you suddenly make changes to the workspace while they are using it, their view breaks and changes.

---

With one workspace, your development and user environment are the same.

---

You cannot use one workspace to be shared between developers and users. Creating multiple workspaces also introduces another challenge. To overcome this challenge, you can use apps on top of the workspaces to share content with end users.

### **Complications of the workspace structure**

Setting up a good workspace structure is a challenge. You need a workspace structure that covers the development, user needs, and deployment structure. This is more of a caution than a limitation. Use workspaces with care, and make sure you have a good setup.

### **Requires Power BI Pro or PPU (Premium Per User)**

Creating Power BI workspaces, or even being part of one (even at the viewer access level), requires a Pro or PPU option and is not part of a Power BI free user account. However, it is possible to create an app for the workspace in a Premium capacity and assign free Power BI users to it. This limitation is one of the main reasons this is method not the most cost-effective option for sharing content with end users.

## Power BI Apps

Workspaces are a great way to share content with users, but when it comes to having a development and user environment, managing multiple workspaces is not the best method. Power BI apps are best for multiple environment approaches. With Power BI apps, your development environment (the workspace) and user environment (the app) are isolated. Power BI apps can also be a cost-effective option for organizations with large userbases.

Creating an app for a workspace is very simple. Users can easily access it through the Apps section of their Power BI account, as shown in Figure 36-5.



**Figure 36-5.** Creating an app for a workspace

Apps are great sharing methods for multiple environments and the best way to share with users in a production environment. However, apps also have their pros and cons, discussed next.

## Advantages of Power BI Apps

### Separate environments for the developer and end users

This method has two separate environments—an environment for developers to edit the Power BI content in a collaborative workspace and another environment for end users to consume the report. End users are only able to view the reports, and developers can make changes.

Power BI apps are the best solution when you need isolated developer and end user environments.

### A cost-effective option for a large userbase

If you have thousands of users and want to share Power BI content with them securely, then sharing with options that rely on per-user licensing is not cost-effective. A Power BI workspace can be assigned to a Premium workspace, and then if you create an app on top of it, the app's users can be free Power BI users. This is a big help for organizations with large userbases. You can purchase a capacity, and as long as it can cover the load of the users over the Power BI content in the service, free users can use it.

This is one of the most common reasons that many organizations use Power BI apps combined with workspaces to share Power BI content.

### Control over multiple Power BI data sources

Similar to workspaces, the Power BI app allows you to share multiple dashboards, reports, and datasets simultaneously. Controlling multiple sources means less maintenance overhead compared to dashboard sharing, which shares one dashboard at a time.

### External sharing

Another great benefit of Power BI apps is the ability to integrate them with Azure B2B services and provide external sharing. If you want to share Power BI content with people outside your company, you can do that with a combination of Azure B2B and Power BI apps.

## Cons of Power BI Apps

Power BI apps are one of the best and the most common methods of sharing in Power BI. There is a small thing that you need to be careful of, though.

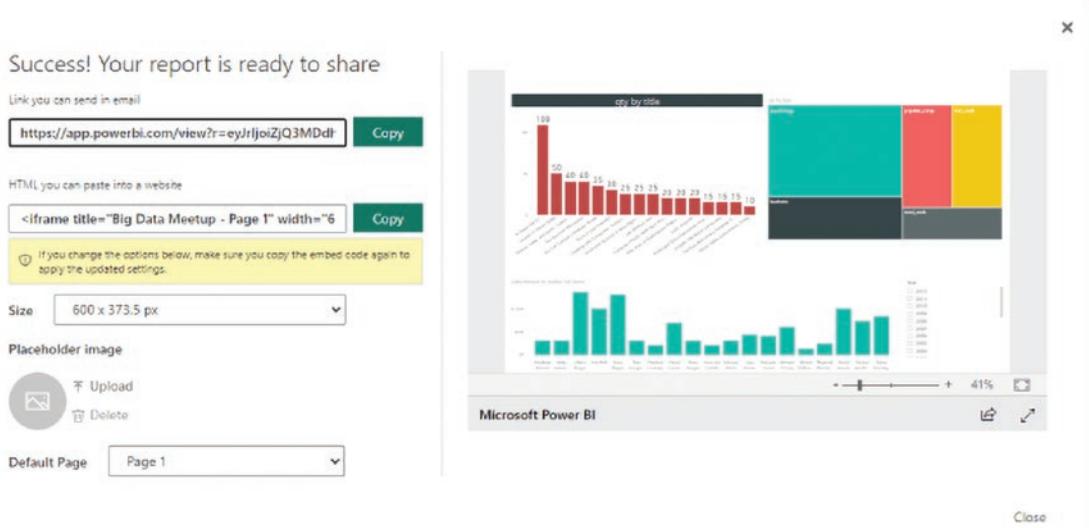
### Changes in the dataset are applied immediately

Power BI app separates developer and end-user environments, and the changes to a report in a workspace don't affect the end user until you update the app. However, this functionality doesn't work with the dataset in that way. The Power BI app and the workspace share the same datasets, so any changes to the schedule refresh or any structural changes will apply to both.

## The Publish to Web Option

Sometimes, you don't need a secure way of sharing. You need an easy and free way of sharing, and your content is not confidential or sensitive. In that case, Publish to Web is your friend. This is the only free way to share data in Power BI, but be aware that this method is not secure.

The Publish to Web method gives you an embed code, which you can use on any web page to embed the Power BI report in it (see Figure 36-6).



**Figure 36-6.** The Publish to Web option

The embedded content is available to anyone who has access to that page. Publish to Web is a free way of sharing. If you think you might use this method, consider the following issues first.

## Security Issues with Publish to Web

### **Everyone can see what you share**

The first thing you might think of is usually security. How can you manage security with this method? The short answer is there is no security. The report is shared through the web or email with everyone. So everyone who has the link or the embed code can access the report. They cannot edit it. But they can view it without restriction.

### **Users can share the reports with others**

A report published on the web has a share section on the bottom-right side. Everyone can share this report with anyone else through all social media channels—Facebook, Twitter, and LinkedIn. This method of sharing is not secure. I recommend sharing data this way only when you have a public report to place on your company or organization's website.

### **The report is public, and not only to those with the link**

This report is not shared only with those with the link. It is shared globally on the Internet. That means a search engine such as Google can find the report. All reports with Publish to Web links are available in search results. You must be sure that the report doesn't reveal confidential information before using this method.

### **All report pages are visible**

If you have a report with more than ten pages, all the pages will be visible to browsers. You cannot limit which pages you want to show and which you don't. I recommend creating different reports if you want to restrict some pages and share them separately.

### **What if the report has row-level security applied?**

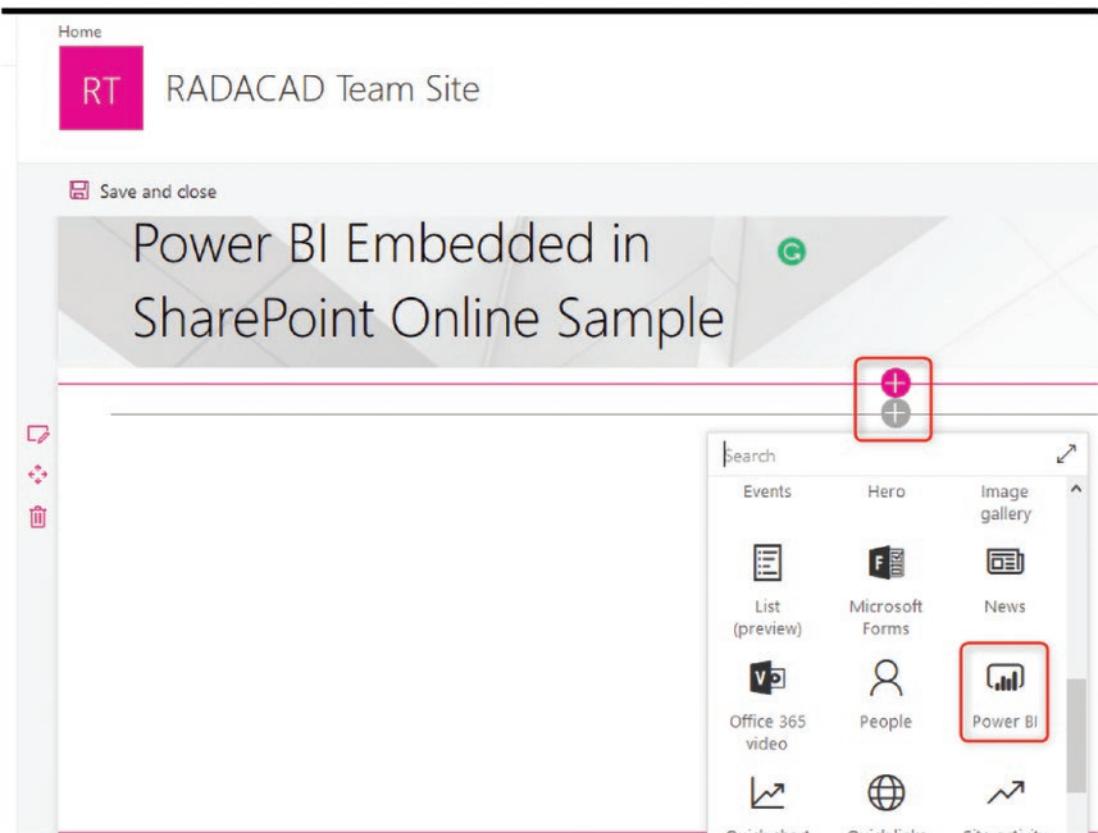
If you have a report with the row-level security applied to it, you can't create a Publish to Web link.

Publish to Web is only recommended for public data sharing on your organization website with the public. There is no security option for Publish to Web; this method should not be used for confidential reports.

## Embed in SharePoint Online

If you are already using SharePoint Online as a portal for document management, consider using Embed in the SharePoint Online feature of Power BI reports. This method is secure, and you can share the report only with the Power BI users you want.

Power BI content can easily be embedded into a SharePoint Online page, as shown in Figure 36-7.



**Figure 36-7.** Embedding Power BI content into a SharePoint Online page

You manage security and sharing in two different places—from the SharePoint site, and in Power BI. This method also has its pros and cons.

## Advantages of Embed in SharePoint Online

### One portal for all content

With SharePoint online, you can share other documents as well. Why not use it for all other documents and the Power BI report? You can have one portal, which is the central sharing portal for your Office 365 tenant. Users usually love integrity.

### Embedding is simple

Unlike Power BI Embedded, embedding in SharePoint Online is easy. You just get the URL and embed it into a Power BI object in SharePoint Online. You don't need to write a single line of code, but with Power BI Embedded, you need a developer.

## Disadvantages of Embed in SharePoint Online

### Power BI service golden plate is underutilized

One of the great aspects of Power BI components is the service. If you use SharePoint embedded, users will use it as the portal for reports. The Power BI Service has many exciting features that may not be well used in this scenario, such as alerts, feature dashboards, the dashboard itself, Q&A, and many other items. Users can still log in to the Power BI Service and see the report, but the experience you created for them with SharePoint Online is not there.

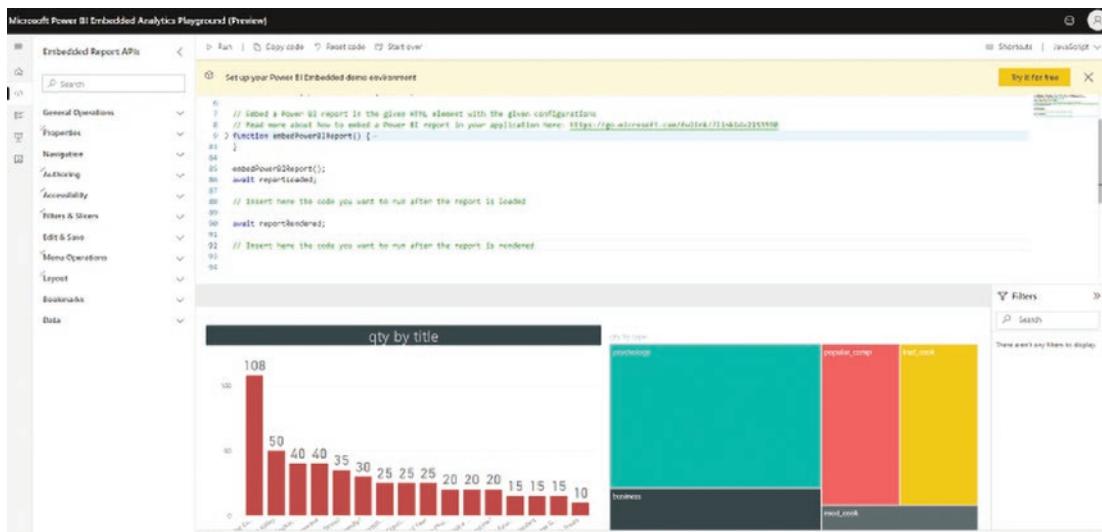
### Two places for managing permissions

At the time of writing this chapter, you need to manage permissions in the Power BI Service and in SharePoint Online. This would take some time for maintenance and reconciliation to check whether those people who have access to the page are always permitted to read the report.

## Power BI Embedded

Sometimes you want to embed the Power BI content into your custom application, and you want the content to be secured. In most cases, you want to leverage the custom user management of your current application rather than Power BI accounts. Power BI Embedded gives you all of these features. The only side effect is that you need a web developer.

Power BI Embedded uses an API called Power BI REST API, and it has many great features for interacting with Power BI content. Users can easily access reports through your application (see Figure 36-8).



**Figure 36-8.** Accessing reports through an application via Power BI Embedded

## Pros and Cons of Power BI Embedded

With Power BI Embedded, you get a fully customizable solution. You can do whatever you want inside your application with Power BI content. You can embed reports, dashboards, tiles, and even Q&As. You can interact with those elements from the web page.

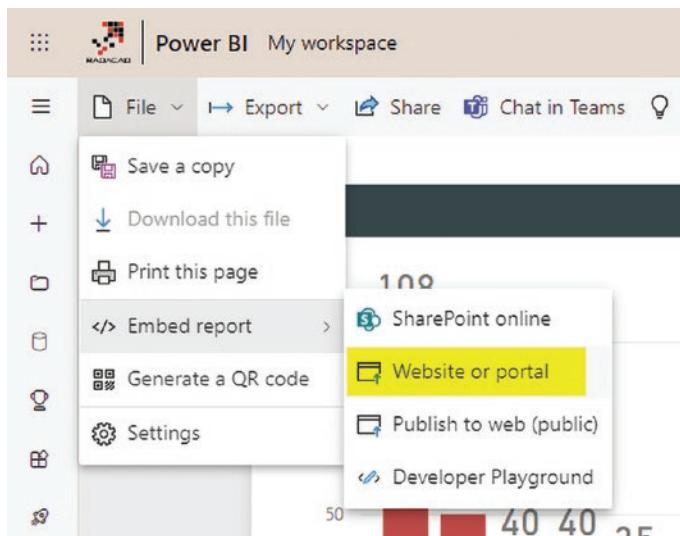
Power BI Embedded can work without the need for Power BI accounts. If you have a set of users without accounts or your users are not part of a single company, Power BI Embedded can be a great solution.

To implement Power BI Embedded, you need a web developer. It is not just about one-off embedding your first content; every change after that, or every new functionality you add to your application, needs a web developer's touch.

Another big advantage of Power BI Embedded is its ability to scale up or down on some SKU capacities and bring the costs down while keeping performance high.

## Secure Embed

This method combines good aspects of Publish to Web and Power BI Embedded. Using Secure Embed (see Figure 36-9), you can share your reports through a web portal with just a few clicks. However, unlike Publish to Web, Secure Embed is a secure method of sharing, and only authorized users have access to the data. As a result, this method also supports row-level security-enabled reports.



**Figure 36-9.** Secure Embed

Like all the other methods, this method also has its pros and cons.

## Advantages of Secure Embed

### Sharing with a few clicks, no developer is needed

You don't have to write a single line of code; it works by using the Secure Embed option. All you need to do is use the URL output of Secure Embed or embed the HTML code on the web page. Unlike Power BI Embedded (which needs a developer's touch), this method is very simple to implement.

### Secure sharing

Since this method is called "Secure Embed," you probably guessed that it includes secure sharing. Content is available only to authorized users. Unlike Publish to Web, this is a secure method of sharing.

### Row-level security is supported

The Publish to Web method doesn't support row-level security because there is no concept of the user logging in, because no login is required. However, using Secure Embed, only authorized users have access to the reports, and as a result, row-level security is possible.

## Disadvantages and Limitations of Secure Embed

Power BI Embedded does limit the content that can be shared. For example, ArcGIS Maps is not currently supported. Secure Embed, at the moment, also doesn't support dashboards or paginated reports.

Another significant limitation of Secure Embed is that it doesn't work with user setups of custom applications. If you use the Secure Embed method, you must use the Power BI user accounts. If you want to use a custom application's userbase, your best bet is Power BI Embedded.

## Summary

So far, you've learned what every method of sharing does; now you can see how they compare, all in one place. This is shown in Figure 36-10.

	<b>Basic Sharing</b>	<b>Workspace</b>	<b>Power BI App</b>	<b>Publish to web</b>	<b>SharePoint Online</b>	<b>Power BI Embedded</b>	<b>Secure Embed</b>
<b>FREE</b>							Yes
No Power BI account needed				Yes		Yes	
Access Levels	Read Secure Yes	Read/Edit Yes	Read Yes	Read	Read Yes	Read/Edit Yes	Read Yes
Dev/User Environment			Yes				
Sharing Multiple Items		Yes	Yes				
Extras							the need for Web Developer

**Figure 36-10.** Comparing the Power BI sharing methods

## Cheat Sheet for Choosing a Method

Basic sharing	A fast and quick method for sharing test reports and dashboards.
Workspaces	A great option for collaborative development environments between Power BI Developer teams.
Power BI apps	The best option to share reports with end users in a user environment that is isolated from the developer environment.
Publish to Web	The free method of sharing that is best for public datasets where the data is not confidential.
SharePoint Online	A good choice when SharePoint Online is the current portal for users in the organization.
Power BI Embedded	An option to bring Power BI content into your application, especially when user management can be done in the custom application.
Secure Embed	A great way to share simply and securely in SharePoint on-premises or a custom web application without the need for a web developer.

## CHAPTER 37



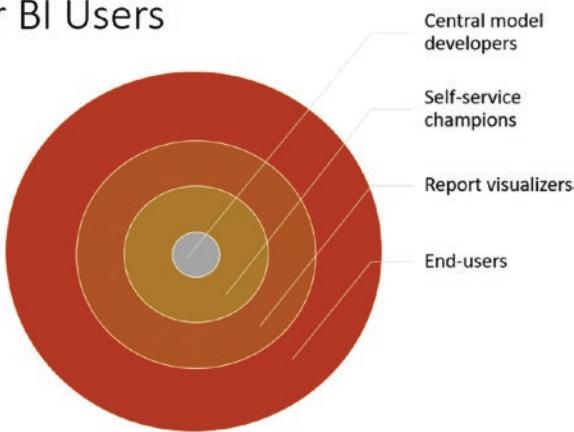
# Types of Power BI Users

Power BI users come in all different types. If you treat everyone equally, you will not have good Power BI adoption. You will likely spend too much on training and still gain poor adoption. This chapter explains the different types of users and the proper actions for each type.

## The Different Types of Power BI Users

In the world of Power BI, there are different types of users in an organization. Some users develop the model, and others consume what is created. On bigger teams, there are more layers. Figure 37-1 shows the four types of Power BI users.

### Power BI Users



**Figure 37-1.** Different categories of Power BI users

These users are defined as follows:

- **End user:** Someone who browses the report (in the mobile app, web browser, or through an embedded web experience). This user might interact with the report by filtering, highlighting, drilling through it, and so on, but they never change the data or create a report.

- **Report visualizer:** This is a much smaller category than the end-user group. For this group, the reports that were built are not enough; they need to modify the visuals. This request might have come from their managers, or they prefer another way to view that data. This group does not change the data model or import new data tables. They might, however, add a few calculations added here and there to support their visualizations.
- **Self-service champions:** This is a smaller group than report visualizer group. This group does visualizations and brings new data into the model. They might have data sources that they want to integrate into the existing analysis, and they do some extra calculations based on that data too.
- **Central model developers:** This is the smallest group of Power BI users in every organization. This group builds data models (and reports) used by the rest of the organization (the end users). Their data models are often an important source for report visualizers and self-service champions.

## Access Control

Depending on the user group, you should also have different types of access and different ways of sharing content. Figure 37-2 shows some suggestions.

### Controlled access



**Figure 37-2.** Access control for Power BI user groups

Giving an end user full edit rights to a data model and report can be problematic. Every user type requires a different level of access:

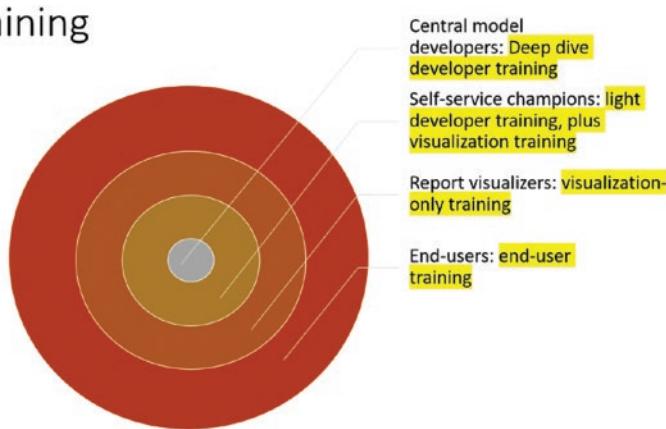
- **End users:** Access the model through Power BI apps or embedded with a read-only option in a web application. This ensures that these users can't change reports.
- **Report visualizers:** Have read-only access to the dataset. A report visualizer only needs a live connection to the data model to build visualizations. However, they can build their own reports. This level of access can be provided in different ways. Using Viewer access to the workspace or using apps are some methods for this purpose. These users do not have edit access to the dataset.

- **Self-service champions:** This group also has read-only access to the central dataset and models. They can import data from other sources, combine that data with this model, build a new model, and create their own reports. They can also create a chained dataset with a DirectQuery connection to the Power BI dataset.
- **Central model developers:** The only group with full edit access to the datasets and reports.

## Training

Each group requires a different level of training because they are performing different functions, as shown in Figure 37-3.

### Training



**Figure 37-3.** Each group's different levels of training

- **End users:** This is the quickest Power BI training option; a couple of hours of training is usually enough. This includes showing them how to navigate the app and drill through in reports or learn about drill down and up and some ways to get most of the existing Power BI reports. Most of the training relates to understanding the underlying data and how it is presented.
- **Report visualizers:** This group should be fully trained in visualization. Give this training as the module of Power BI for data analysts, which takes two days. Users learn how to create useful visualizations in Power BI. There isn't any point in training this group of users on data modeling.
- **Self-service champions:** Because this group wants to build its own data models, they must learn about modeling in Power BI in addition to visualization. Training takes longer and includes training for the Report Visualizer, as well as some parts of modeling, Power Query, and DAX. This can sometimes be close to a week of training, but consider splitting it into multiple sessions.

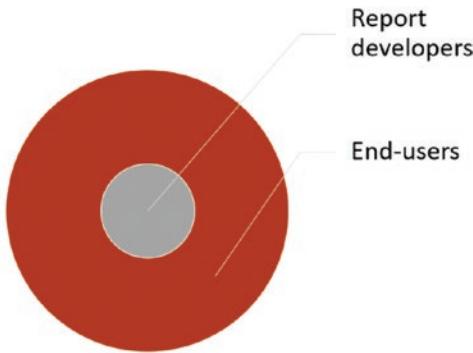
- **Central model developers:** For this group, you require the deepest dive Power BI training option. Deep dive training on Power Query, DAX, M scripting, modeling, and visualization is required. This group is likely to be the go-to resource for the self-service champions and report visualizers when they have questions. This group also requires a bit of architecture, administration, and governance training. This group requires multiple weeks of training, which needs to be split into multiple sessions.

Visit [radacad.com/power-bi-training](http://radacad.com/power-bi-training) for a comprehensive agenda of a Power BI training, which covers all the agenda items mentioned here.

## The Layers and User Categories Are Different in Each Organization

Each organization can have different layers. In small businesses, you often find two or three layers. The two layers can be the report developer and the end users, as shown in Figure 37-4.

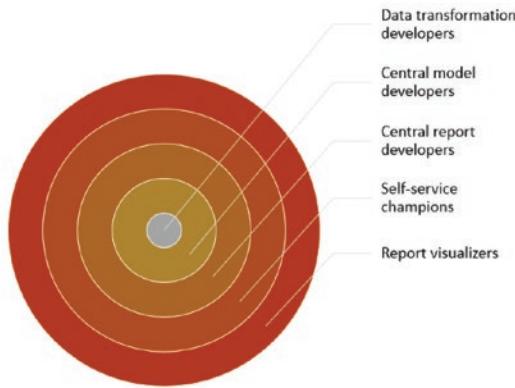
### Small businesses



**Figure 37-4.** Power BI users in small businesses

In larger organizations, you sometimes find layers within layers. The central developer layer can also be split into data transformation developers, data model developers, and core visualizer developers, as illustrated in Figure 37-5.

## Layers within layers



**Figure 37-5.** Power BI users in a large organization

You should explore the culture of the users in your organization and determine the structure of these layers.

## Users Can Move Between Layers

Once a user is in a layer, it doesn't mean they will stay in that layer forever. Users change functions over time. You might have a report visualizer interested in doing some self-service stuff, so their job function changes. You should be ready to upskill users to ease the process of moving them to another layer.

## Summary

It's important to understand the nature of the Power BI users in your organization and not mix these user groups. If you mix all these groups together and treat them all the same, you will likely end up with these problems, including:

- Spending too much training budget when it's not necessary
- Losing interest in downstream user layers
- Upstream users will likely start developing their own way of doing things
- The mix-up will hurt governance more than anything else
- Trust in Power BI will slowly vanish
- Power BI adoption may fail

## CHAPTER 38

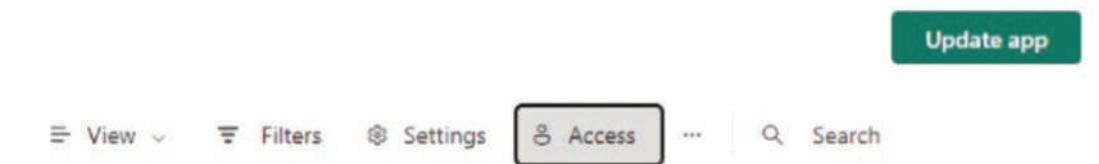


# Best Practices for Setting Up Workspace Roles

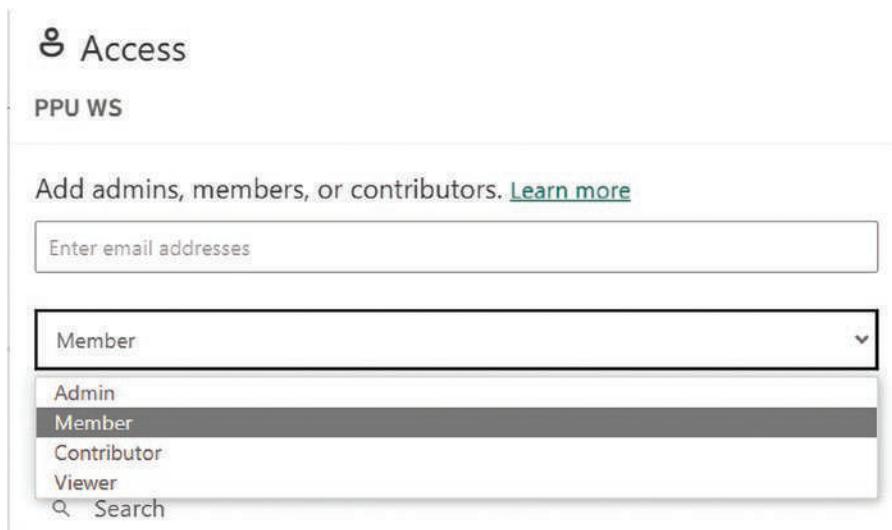
Power BI workspaces have changed from the old days where there was Edit and View access only. You now have more options for roles in a workspace, and in my courses, I have found that many people choose the incorrect role without knowing what the role does. This chapter explains all the roles in the workspace and discusses the best way to set them up to create a secure workspace.

## The Modern Power BI Workspace

There is an old version of Power BI workspace that a few people are still using. Microsoft started migrating those older workspaces (called workspace version 1) to the new version (workspace version 2), but you still might have some older versions in your environment. The older workspace version included Edit, View, and Admin access. It was easy to understand. In the new workspace (see Figure 38-1), there are four roles, as shown in Figure 38-2.



**Figure 38-1.** Assigning roles in a Power BI workspace



**Figure 38-2.** Roles in the workspace

## Roles in the Workspace

This section looks at the roles and access levels in workspaces, beginning with Viewer.

### Viewer

This role does what its name says in terms of viewer access. It authorizes the users to only view the report's content, dashboards, and workbooks (see Figure 38-3). The view role has read-only access to the content. They cannot access dataflows but can access the data stored in the dataflow.

The screenshot shows the 'Dashboards', 'Reports', and 'Workbooks' tabs selected in the top navigation bar. A search bar is present above the list of items. The list displays three items, each with a star icon, a small thumbnail, and three action icons (grid, lightbulb, and link). The columns are labeled 'NAME ↑', 'ACTIONS', 'OWNER', and 'INCLUDED IN APP'. Each item is owned by 'Company' and has a toggle switch set to 'No' under 'INCLUDED IN APP'. A red box highlights the 'Dashboards', 'Reports', and 'Workbooks' tabs. Below the list, a large red text box contains the following text:

**Viewer Role**  
**can access all reports and dashboards**  
**no access to dataset, dataflow, or Analyze in Excel**

**Figure 38-3.** The Viewer role

If this role is also given build permission, the role can build content with Analyze in Excel or make a live connection to the dataset. I am not a big fan of this role. I explain why later in this chapter. There is a better option for people with this role.

## Contributor

The Contributor role is for developers in the workspace. These people can access reports and dashboards, as well as datasets and dataflows. They can edit the content as well as delete it. They can publish a report to the workspace or remove it. The contributor role allows them to do all their development work in the workspace. See Figure 38-4.

The screenshot shows the Power BI workspace interface. At the top, there's a navigation bar with 'Create' (highlighted with a yellow box), 'Settings', 'View', and a three-dot menu. Below the navigation bar is a search bar labeled 'Search...'. Underneath the search bar is a horizontal menu with tabs: 'Dashboards', 'Reports', 'Workbooks', 'Datasets' (highlighted with a yellow box), and 'Dataflows'. To the right of the menu, it says 'Showing 3 items'. The main area displays a list of three items, each represented by a small icon, a star icon, and a name. To the right of the list are columns for 'OWNER' (all listed as 'Company') and 'INCLUDED IN APP' (all set to 'No'). Below the list is a red box containing a set of icons for actions: a plus sign, a gear, a lightbulb, a magnifying glass, a gear, a trash can, and a copy icon. Overlaid on the bottom half of the screenshot is a large red box containing the following text:

**Contributor Role**  
**Edit access to all the content in the workspace**  
**Create/Delete/Edit access for the content**  
**Cannot publish APP**

**Figure 38-4.** Contributor role

The Contributor role is the ideal role for Power BI developers. They can perform all of these actions:

- Publish a report to the workspace
- Edit the content in the workspace
- Delete the content in the workspace
- Access all workspace objects—reports, dashboards, workbooks, datasets, and dataflows
- Copy content, use Analyze in Excel, and more

This role should be assigned to all developers in the team. However, this role should not be confused with the next role: the Member role.

## Member

The Member role has access to all the Contributor role's actions, plus has the ability to publish, unpublish, and update apps (see Figure 38-5). The Member role has the worst name of all the roles, and that is one of the reasons that many people don't understand it.

The screenshot shows the Power BI workspace interface. At the top, there's a navigation bar with 'Power BI Company' and icons for search, settings, access, view, and publish. The 'Access' and 'Publish app' buttons are highlighted with red boxes. Below the navigation bar is a search bar labeled 'Search...'. Underneath is a list of items categorized as Dashboards, Reports, Workbooks, Datasets, and Dataflows. The 'Reports' tab is selected. The list shows three items, each with a 'NAME' column (containing three report icons), an 'ACTIONS' column (with icons for edit, publish, copy, etc.), an 'OWNER' column (listing 'Company'), and an 'INCLUDED IN APP' column (with a toggle switch set to 'Yes'). A large red box highlights the 'INCLUDED IN APP' column for all three items. Below this list, there is a summary stating 'Showing 3 items'.

**Member Role**

**Can Publish, Un-publish, or Update App**

**Can do all things that Contributor role can do**

**Can set Access for Member role or underneath**

**Figure 38-5.** The Member role

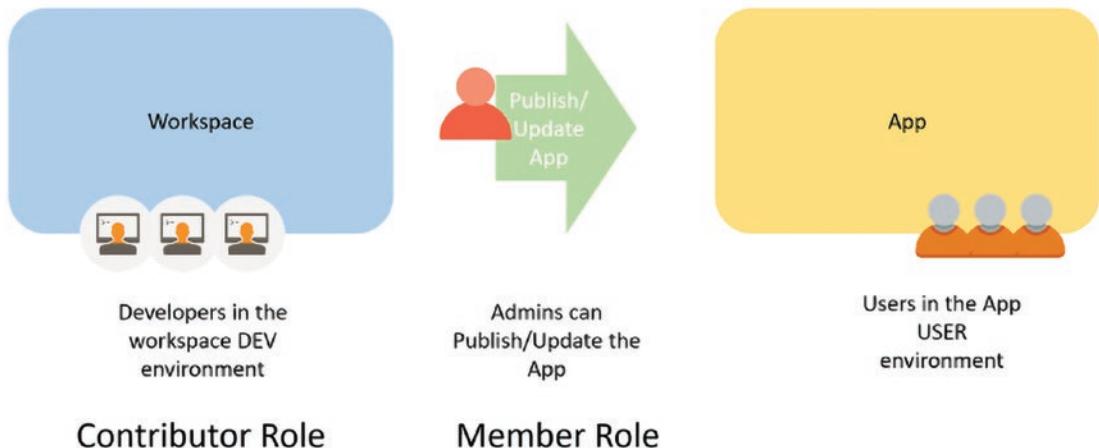
Consider this role as the Deployment Role. This role can publish the content of a workspace as an App for the end-users.

The Member role has one of the most important actions in the workspace—pushing content from the DEV environment to the USER environment.

I have seen a lot of workspaces in which all the developers had Member access. The big problem with this is that any developer can publish apps to end users.

The separation of the Member and Contributor roles is one of the best advantages of the new workspace version. Check out my article at [radacad.com/workspace-v2-of-power-bi-what-are-advantages-vs-old-version](http://radacad.com/workspace-v2-of-power-bi-what-are-advantages-vs-old-version) for more information about this. See Figure 38-6.

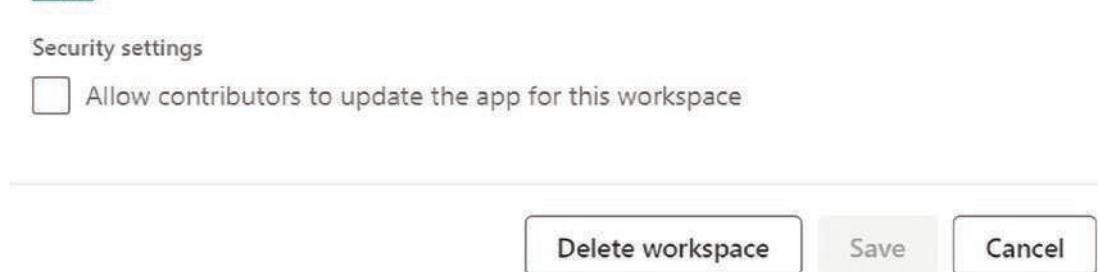
## New Version of Workspace (V2): Developers CANNOT Publish/Update app mistakenly. Member Role can do that.



**Figure 38-6.** Member versus Contributor role

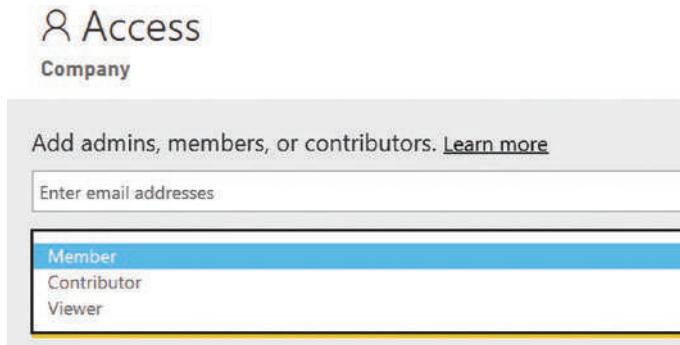
The last thing you want is for a developer from the DEV team to mistakenly push the button and publish content that is not user-ready to the end users. You can avoid that by separating people who have Member role access and restricting it only to the group of people who are your deployment managers.

There is, however, a tricky checkbox in the Advanced tab when you create a workspace that gives the Contributor role access to publish an app (see Figure 38-7). I strongly recommend *not* using this option. If you do this, you are making the Contributor and Member the same and going back to the old version of the workspace, where there is no control over who publishes content.



**Figure 38-7.** Do not select this option—being able to update an app should remain in the Member's domain only

The Member role can also set access for roles in the workspace, as you can see in Figure 38-8.



**Figure 38-8.** Access levels from the Member role

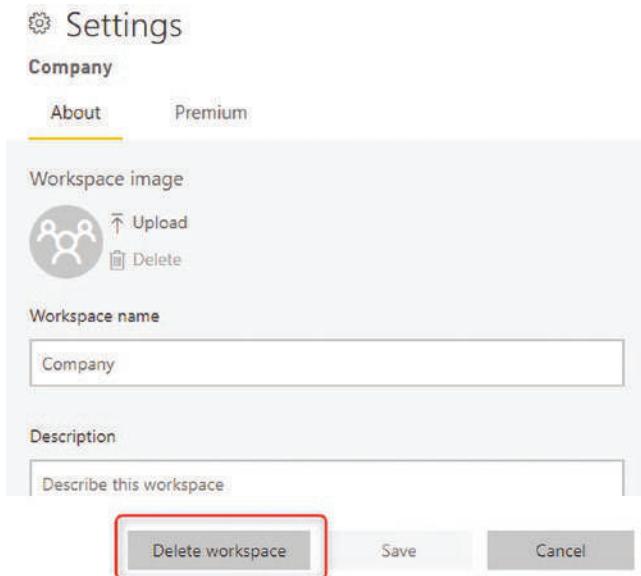
Another thing that the Member role can do is share reports and dashboards individually in the workspace (see Figure 38-9).

NAME ↑	ACTIONS	OWNER	INCLUDED IN APP
[Report 1]	[Sharing Icon] [Settings Icon]	Company	Yes
[Report 2]	[Sharing Icon] [Settings Icon]	Company	Yes
[Report 3]	[Sharing Icon] [Settings Icon]	Company	Yes

**Figure 38-9.** Sharing content individually

## Admin

The Admin role has full access to the workspaces. In addition to doing all the Member's role actions, members of the Admin role can add or remove administrators to/from the workspace and can update or delete workspaces. See Figure 38-10.



**Figure 38-10.** The Admin role

## Best Practices for Each Role

Now that you are familiar with all the roles in the workspace, this section explains how to use them properly.

### Viewer Role: Use Power BI Apps Instead

The Viewer role has read-only access to all content and cannot use Analyze in Excel.

#### End users

I have seen many organizations use the Viewer role to give access to their end users. This is a big mistake because you are giving users access to content in the developer environment. The DEV environment content is likely to be changed by any development team members.

Instead, use the Power BI apps, which create a good separation between the DEV and USER environments. Users have access to reliable content while the developers work on changes behind the scenes.

When you build a Power BI app, you can specify a navigation menu, design a great look and feel for your users, and give them the ability to slice and dice the data as they want. This is much better than Viewer access on the workspace. See Figure 38-11.

The screenshot shows the Power BI Publish app interface. At the top, there's a navigation bar with 'Power BI' and 'PPU WS'. On the right side of the bar, it says 'PPU Trial 50 days left' and has a 'Search' field. Below the bar is a header with 'PPU WS' and a 'Development' status indicator. A 'Create app' button is also visible. The main area is a table with columns: 'Name', 'Type', 'Owner', 'Refreshed', 'Next refresh', 'Endorsement', and 'Sem'. There are five items listed:

Name	Type	Owner	Refreshed	Next refresh	Endorsement	Sem
Northwind	Dataset (default)	PPU WS	7/22/22, 4:26:08 PM	N/A	—	—
Northwind	Dataset	Reza Rad	—	N/A	—	—
Northwind report	Report	PPU WS	7/22/22, 4:26:08 PM	—	—	—
Reza Test 1: Viz	Report	PPU WS	7/22/22, 1:40:14 PM	—	—	—
Reza Test 1: Viz	Report	PPU WS	7/22/22, 2:08:54 PM	—	—	—

**Figure 38-11.** The Publish app

### Test Users

Some organizations give Viewer access to test users. That way, the user can test the content before sending it to the end users using Power BI apps.

Although this might seem like a good use for the Viewer role, it's not. If you have 15 Power BI reports in the workspace, and out of those, only three are test-ready, you should only share those three reports. As you have already learned, the Member role can share reports and dashboards individually. Sharing items individually is a better way to give test users access than using the Viewer role. That way, you don't give users access to the entire DEV content, and users can focus only on the test materials.

Other organizations prefer to have a DEV workspace and a TEST workspace. If that is the case, Publish Power BI apps can be used for test users.

## Contributor Role: Developers Only

Developers on your team should all have Contributor access to the workspace. There is no need to add them to the Member role. Reserve the Member role for the next group. Contributor access gives developers all the access they need. Developers can upload content, update it, change it, delete it, build new objects, and more.

## Member Role: Deployment Group Only

On your team, there should be a deployment person or a group of deployment gatekeepers. This group of people ensures that the right content reaches the audience. They go through a process of checks to make sure the content is ready to publish to users. These people should have access to the Member role. They can publish content to the end users.

## Admin Role: Admin Group

Don't assign the Admin role because you want to give someone a lot of access. Remember that even the Member role can give access to other Member-level or lower-level roles. For those reasons, there is little need for a person or a group of people to have the Admin role. Only give access to the group who needs to control the creation or deletion of the workspace or admin-level access.

## Use Groups, Not Individual Accounts

The golden rule in security is to never use individual accounts. In the Power BI world, there are some places where you can, and others where you cannot, use security groups instead of individual accounts. Everywhere you can use a security group, make sure to use it instead of an individual account. Doing it this way makes access control even easier, because you can simply add and remove people from that security group.

## Summary

Roles are not split into just View and Edit anymore. Figure 38-12 encapsulates the recommendations discussed in this chapter.

Role	Usage
Viewer	Don't use. Use Power BI App instead
Contributor	Developer Group only
Member	Deployment Group
Admin	The workspace admin

**Figure 38-12.** Recommendations for the various roles

## CHAPTER 39



# Build Access Level

I have often seen Power BI users with access to areas they shouldn't be able to access. In Power BI, you can share a report with a user for viewing only or give them access to view, build, and edit. These are all different levels of access. Build access is often confused with Edit access. This chapter explains the differences between these two access levels.

## Build Access Example

One of the easiest ways to explain the difference between the Edit and Build access levels is through an example of building a Power BI report on top of an existing dataset. If you get data from a Power BI dataset, you are building a new report (or even a new dataset) on top of the existing dataset.

As shown in Figure 39-1, when using Get Data from a Power BI dataset, the user has access to some datasets.

A screenshot of the 'Select a dataset to create a report' dialog in Power BI. At the top, there is a search bar containing 'movies' and a dropdown menu labeled 'All datasets'. Below the search bar is a table with columns: Name, Endorsement, Owner, Workspace, and Refreshed. The table lists several datasets: 'Movies 20181210' (Promoted, Reza Rad, DIAD WS, a year ago), 'Composite model movies' (Reza Rad, DIAD WS, 5 days ago), 'Movies' (Reza Rad, General Dataset Workspaces, 11 months ago), 'Movies 20181210' (Reza Rad, the new v2 sample, a year ago), 'Movies 20181210' (Reza Rad, Radacad, a year ago), and 'Movies OMV' (Reza Rad, workspace v2 dif, a year ago). At the bottom right of the dialog are 'Create' and 'Cancel' buttons.

Name	Endorsement	Owner	Workspace	Refreshed
Movies 20181210	Promoted	Reza Rad	DIAD WS	a year ago
Composite model movies		Reza Rad	DIAD WS	5 days ago
Movies		Reza Rad	General Dataset Workspaces	11 months ago
Movies 20181210		Reza Rad	the new v2 sample	a year ago
Movies 20181210		Reza Rad	Radacad	a year ago
Movies OMV		Reza Rad	workspace v2 dif	a year ago

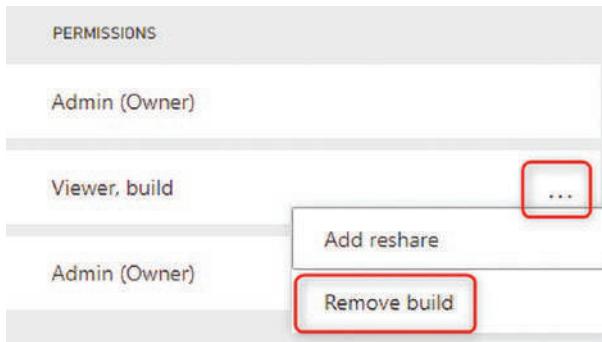
**Figure 39-1.** Getting data from a Power BI dataset with Build access

The user profiled in Figure 39-1 sees all the datasets with movies in their name. The highlighted dataset, as an example, is shared with users who have View and Build access (see Figure 39-2).



**Figure 39-2.** The user has Build access to the Power BI dataset

If you remove Build access from this user, as shown in Figure 39-3, they lose access to that dataset when building a report or new content.



**Figure 39-3.** Removing build access from a Power BI dataset

As you can see in Figure 39-4's simple example, removing Build access from the dataset means that the user can no longer build any new content on top of that dataset.

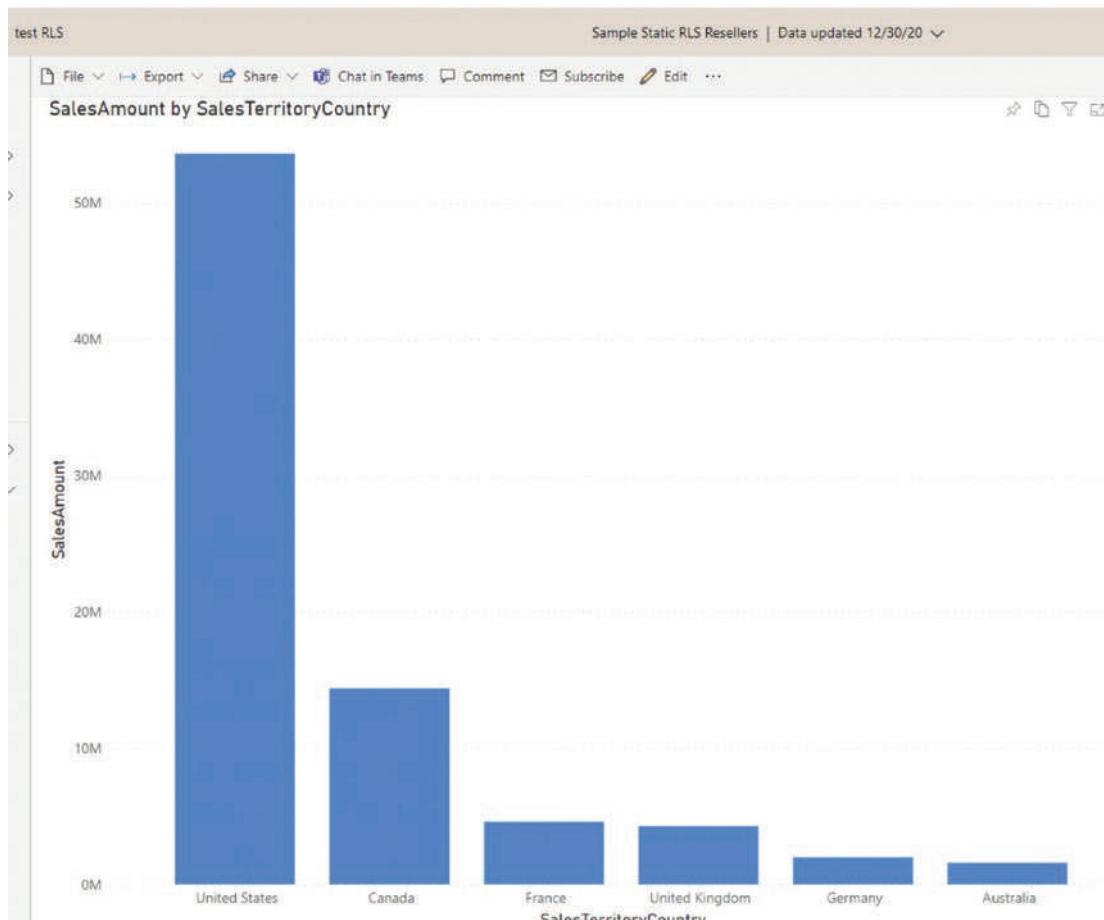
Name	Endorsement	Owner	Workspace	Refreshed
Movies 20181210	(?) Promoted	Reza Rad	DIAD WS	a year ago
Composite model movies		Reza Rad	DIAD WS	5 days ago
Movies 20181210		Reza Rad	the new v2 sample	a year ago
Movies 20181210		Reza Rad	Radacad	a year ago
Movies OMV		Reza Rad	workspace v2 dif	a year ago

**Figure 39-4.** Removing Build access will remove the dataset from the list of Get Data from the Power BI dataset

## Are Build and Edit the Same? A Row-Level Security Example

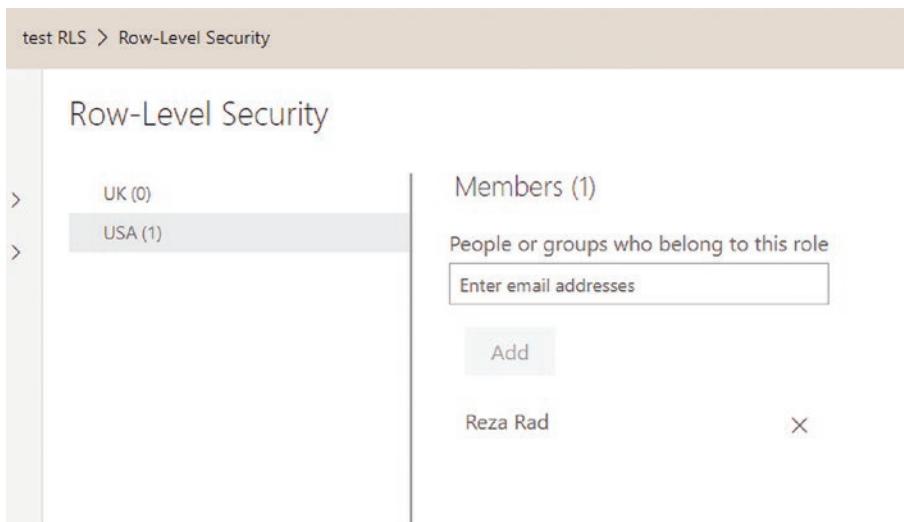
Now that you have been introduced to the Build access level, let's see if the Build and Edit access levels are the same. The short answer is no, they are not. The following row-level security example explains why.

Figure 39-5 shows a report on top of a dataset with RLS (row-level security) configuration defined.



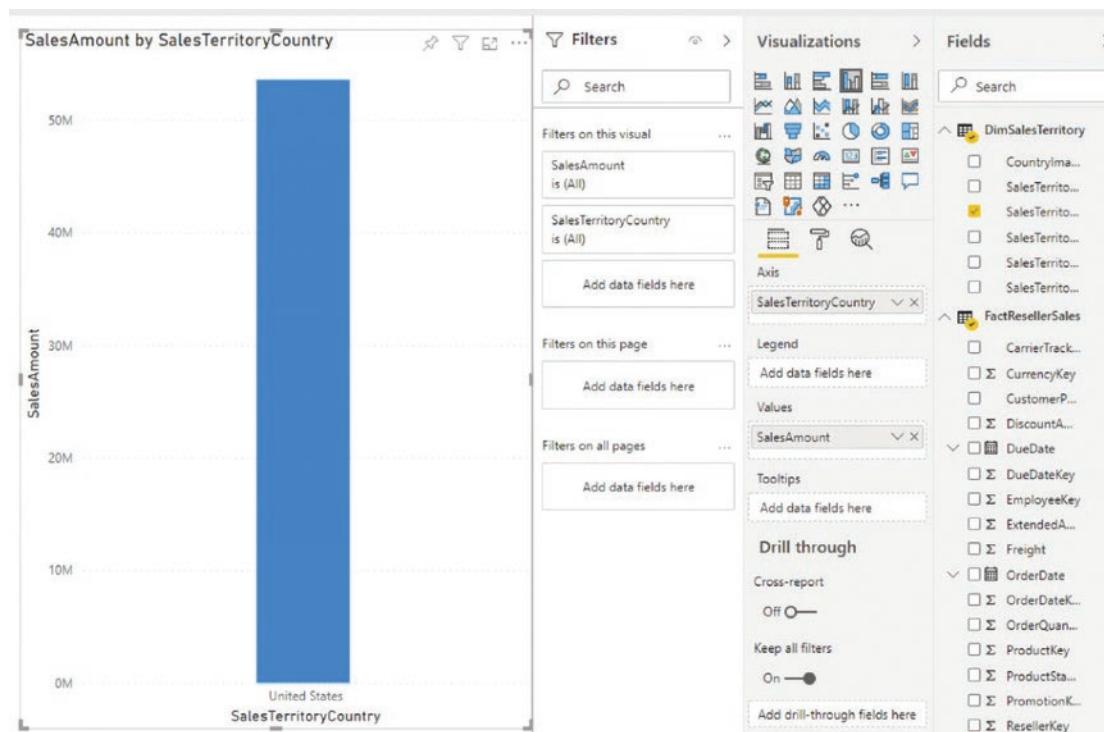
**Figure 39-5.** A report on a dataset with RLS defined

If I have a user with View and Build access to a dataset, the reports that the user can build depends on their access to the roles configured at the dataset level (see Figure 39-6).



**Figure 39-6.** Row-level security configuration of the dataset

This user, if they build a report on top of that dataset, will see something like Figure 39-7.



**Figure 39-7.** A limited view of the data for the user with Build access on a dataset with RLS configuration

As you can see, the user can see only part of the data that they are allowed to see. If this user had Edit access to that dataset, all the data would be visible.

The difference between Build and Edit is not just on the RLS, it is also for editing the content. When a user has Edit access to a dataset, that user can change or delete the content. But the user with Build and View access can only build new content on top of that dataset and cannot change the original content.

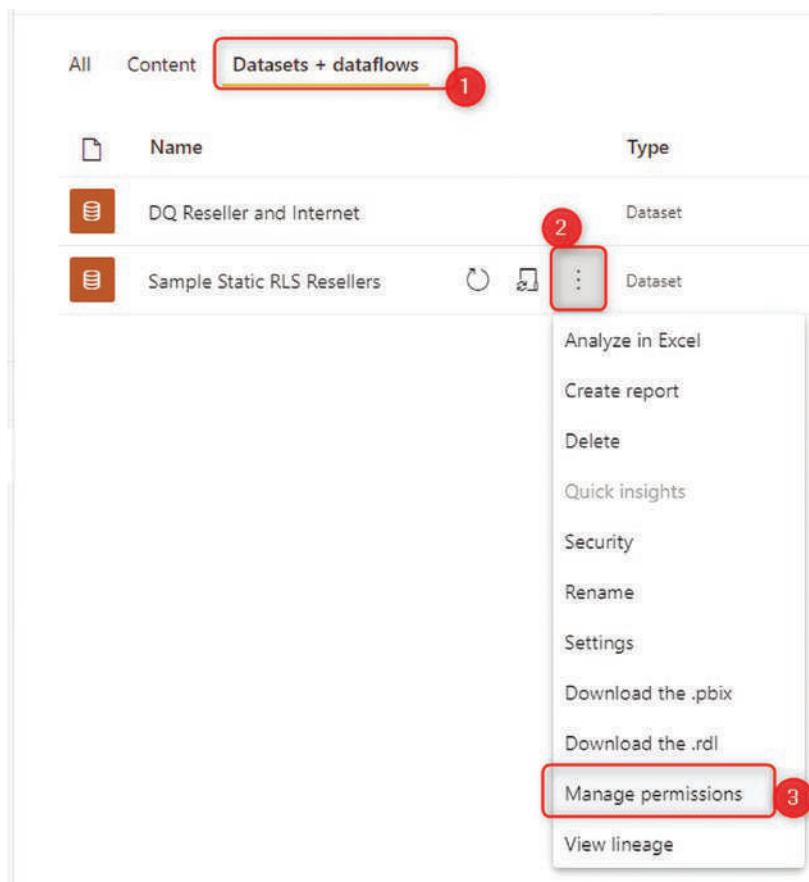
This point is very important, because there are different types of users, and you should not give someone greater access than they need.

## Providing Build But Not Edit Access

To give a user Edit access to a dataset, you can give them proper access to the workspace. Different methods provide Build and View access but not Edit access. Some of the most common methods are discussed next.

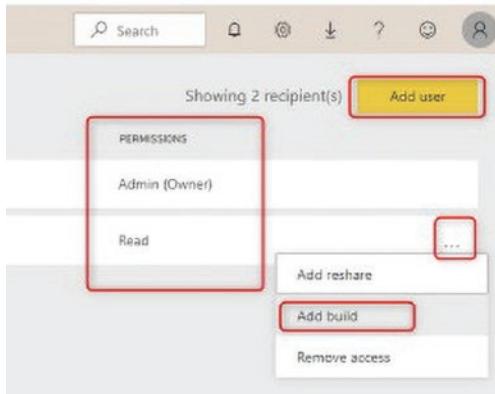
### Manage Permissions: Add Build Access

On any given dataset, you can choose the Manage Permissions option (see Figure 39-8).



**Figure 39-8.** Manage Permissions on the Power BI dataset

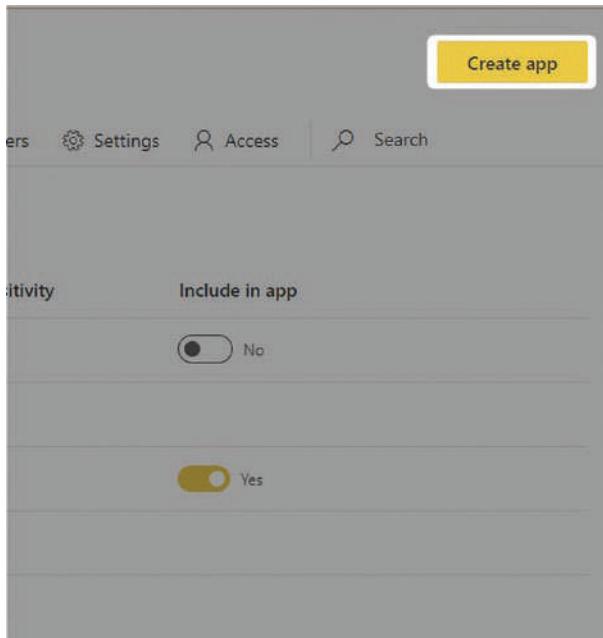
From the Manage Permissions setting, you can add a user with the permissions of Build and View or add permission to an existing user in the list. You can also remove Build permission in the same place (see Figure 39-9).



**Figure 39-9.** Adding and removing Build access to a Power BI dataset through Managing Permissions

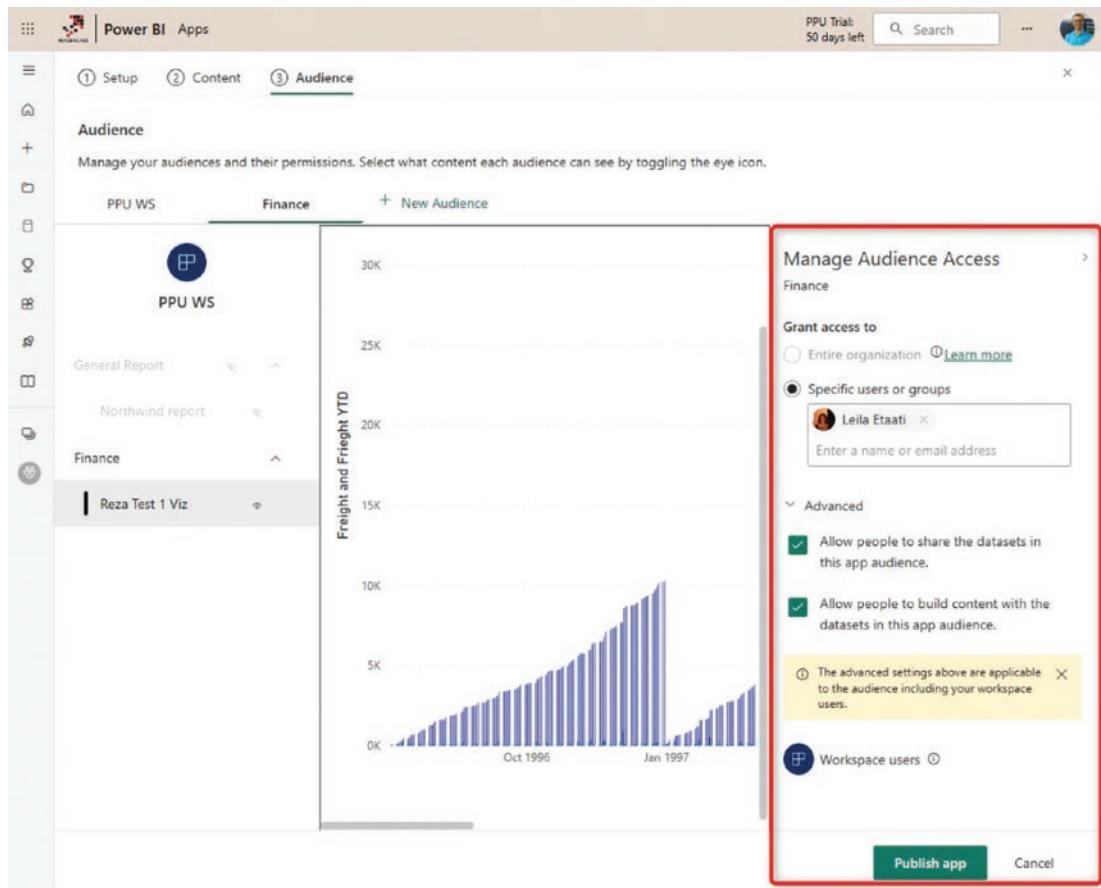
## Create an App with Build Access

Another common way to provide Build access to users is enabling access when creating an app, as shown in Figure 39-10.



**Figure 39-10.** Creating a Power BI app

On the Audience tab of the App Creation wizard, you can choose whether the users can have Build permission, as shown in Figure 39-11.



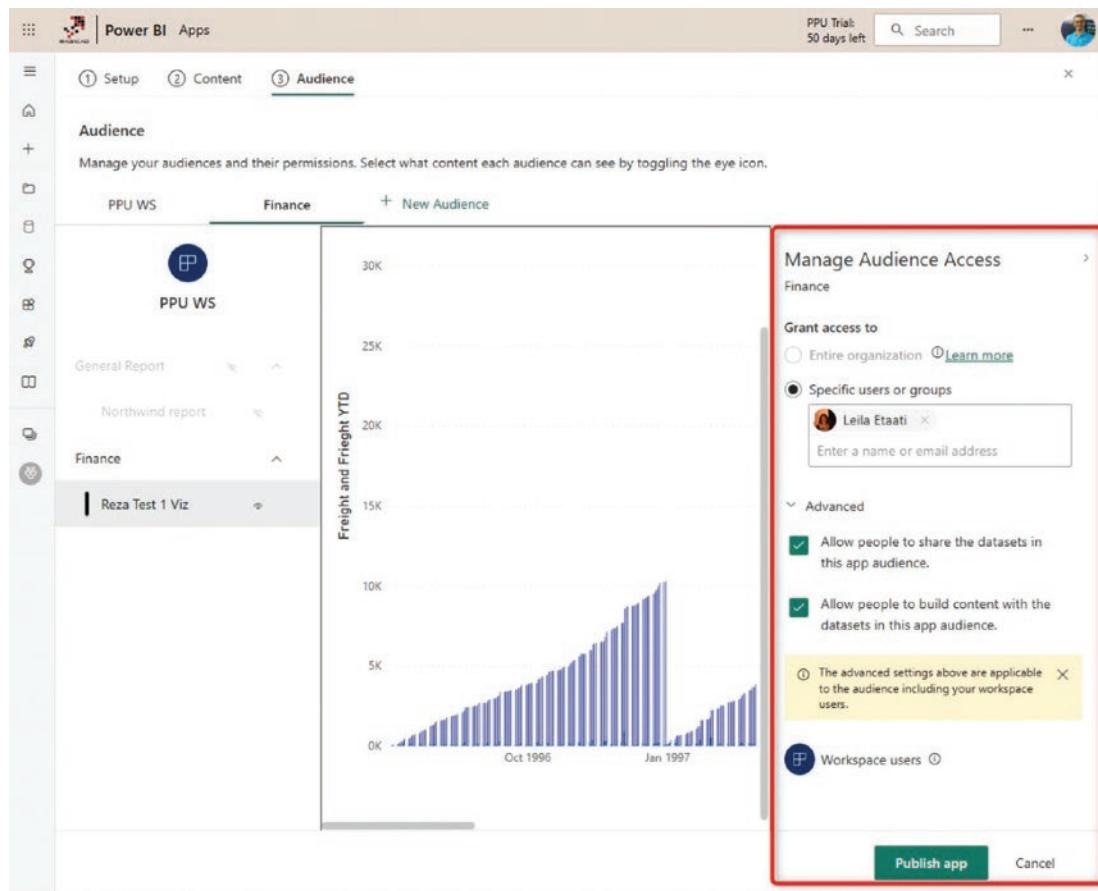
**Figure 39-11.** Enabling Build permission for Power BI app users

This will create Build access in the permissions area of the dataset (see Figure 39-12).



**Figure 39-12.** Build access is provided through the Power BI app

Providing access through the Power BI app is my recommendation. Because you can create different audiences in Power BI apps, set Build permission for an audience, and keep it unchecked for others, this is a perfect way to give Build permission, all of which you can see in Figure 39-13.



**Figure 39-13.** Providing access through a Power BI app

## Who Should Have Build Access?

Report visualizers and self-service champions are among the users who need Build access to a dataset. Developers usually have Edit access, but other users who want to build content on top of an existing dataset require the Build access level.

## Licensing

Users need a Pro license to run Build operations on Power BI datasets. Free Power BI users with Premium capacity can't build content unless they have a Pro license. (Note that the licensing options mentioned here could change in the future.)

## Summary

In this chapter, you learned that the Build access provided alongside View differs from Edit access. A user with Edit access will see everything regardless of row-level security. But Build and View access respects the RLS configuration. Edit access also provides the ability to modify and delete content. You learned how to provide Build access but not Edit access. This is an important consideration for your Power BI sharing method.

## CHAPTER 40



# How to Organize Workspaces in a Power BI Environment

The question I often get during my Power BI architecture consulting sessions is “How should I organize my workspaces? Should I have one workspace with all the reports in it, or multiple workspaces? This chapter provides a guide on how to organize and set up workspaces in your organization.

In a nutshell, a *workspace* is an environment that hosts and shares Power BI content. This Power BI content includes but is not limited to datasets, dataflows, datamarts, reports, paginated reports, metrics, and dashboards.

## Considering a Power BI Workspace as a Single Development-Sharing Unit

You can have multiple Power BI objects inside a workspace, but when you share the workspace, you share all of the objects. When I speak of sharing, I am referring to using one of the workspace sharing methods. By using Power BI apps, you can share a subset of the content, and using basic sharing, you can share an individual object. However, I don’t recommend using basic sharing for content in the workspace (unless it is for testing reports). In the rest of this chapter, I do not consider that option for sharing workspace contents.

This means that the entire contents in your workspace can be shared with someone who has an access role in your workspace (the Administrator, Member, Contributor, or Viewer role) and a subset of that to the Power BI app users.

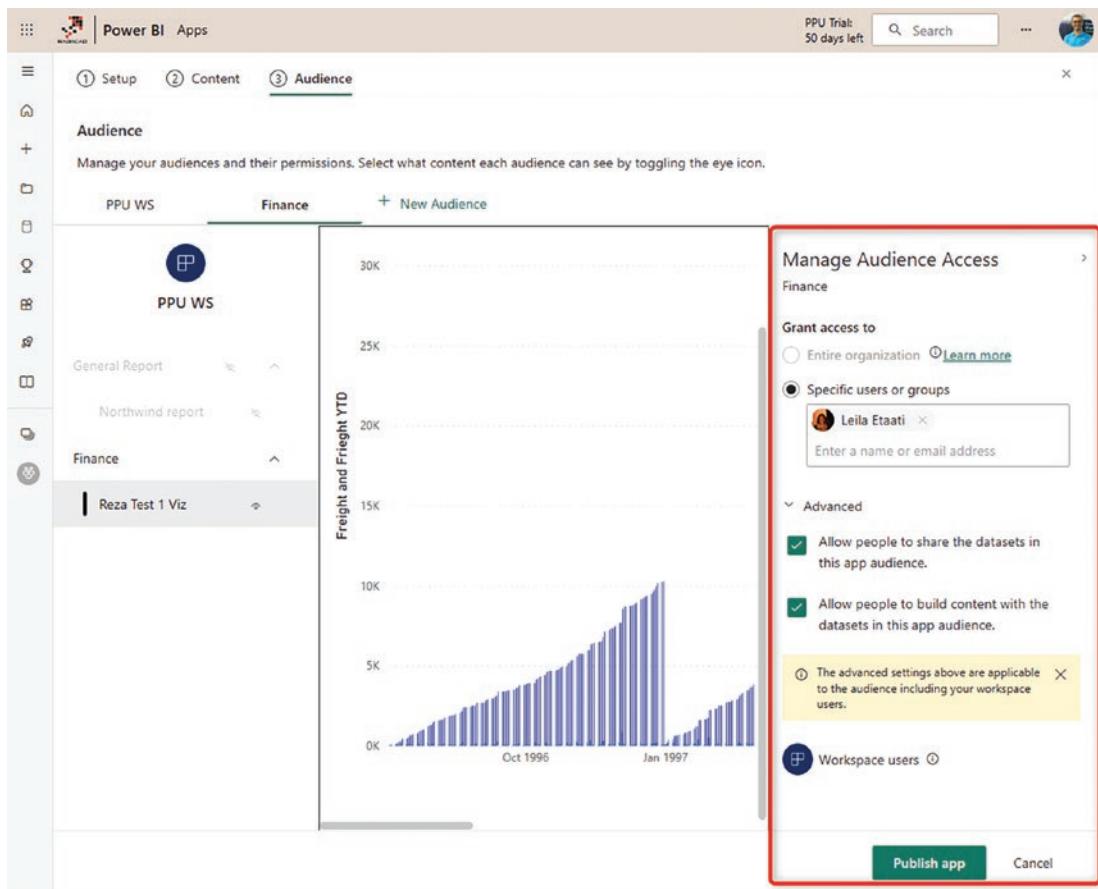
---

You cannot share part of a workspace’s content with some users, and another part with other users. The Power BI workspace is one single sharing unit.

---

## Separating Audiences Using Power BI Apps

You should consider a workspace as a single development-sharing unit because you can use one workspace for multiple groups of audiences. In a previous chapter, you learned how to create multiple audiences and share different content from the same workspace with a different group of users. However, developers are users with Edit access to the content, which is only for Admin, Member, or Contributor users. These access levels permit the user to access the entire contents of the workspace, not just part of it. So, the Power BI workspace is a single unit of sharing for developers or data analysts (see Figure 40-1).

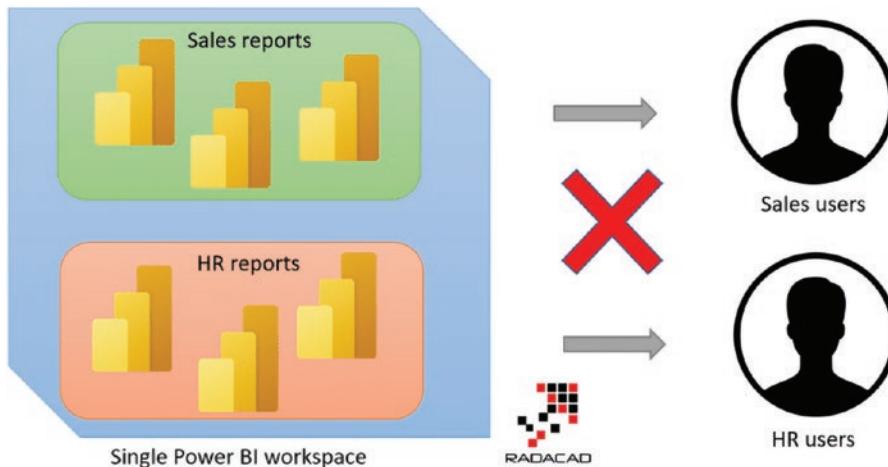


**Figure 40-1.** The Power BI workspace is one unit of sharing for developers or data analysts

## Separating Developers with Multiple Workspaces

Based on this explanation, it is understandable that you will need separate workspaces for different groups of developers (or data analysts). Figure 40-2 shows two sets of reports that should be shared with two different audiences; they won't all be hosted in one workspace.

## All the users will have the same view



**Figure 40-2.** Sharing the content of one workspace with a single group of analysts

You need two different workspaces if you have two different sets of reports for two different groups of developers, as illustrated in Figure 40-3.

## Separate workspaces for different audiences



**Figure 40-3.** Separating Power BI workspaces into different audiences

Note that there is a difference between developers and end users. Developers need access to the workspace to edit the content, whereas end users use Power BI apps.

This means that if you have 12 groups of developers for 12 sets of different reports, you need 12 workspaces.

## Split the Load, Use the Capacity

Another important reason for using separate workspaces is to split the load. This is normally the case when you use a dedicated capacity plan (Power BI Premium or Embedded). If you have a very high consumption rate report, you might want to keep it separate from other reports with low consumption rates and host it in a separate workspace. For each workspace, you can choose the dedicated capacity on which it will be hosted (see Figure 40-4).

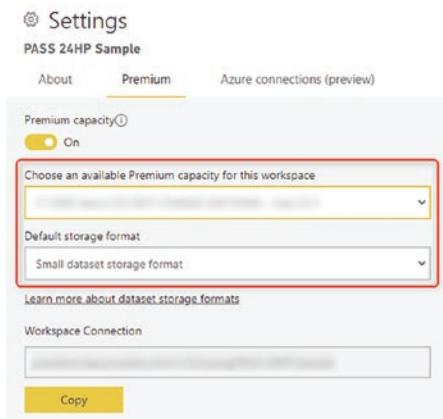


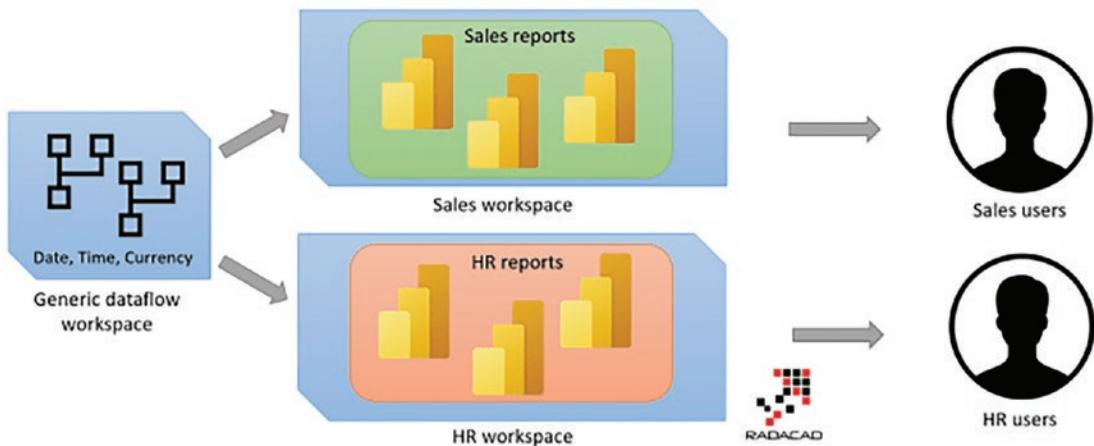
Figure 40-4. Capacity settings for a workspace

## Sharing Workspaces Among Multiple Developers

I've mentioned that one normal practice of having multiple workspaces is having one per developer group. This means, for example, having HR reports in the HR workspace and Sales reports in the Sales workspace. However, what if both of those reports use the Date table? Then you need that Date table to be accessible to both groups.

Having a shared workspace (see Figure 40-5) is an important part of the Power BI architecture. This technique reduces the redundancy in the implementation, increases consistency, and helps in the overall development process of the Power BI content.

## Shared workspaces: Less redundancy

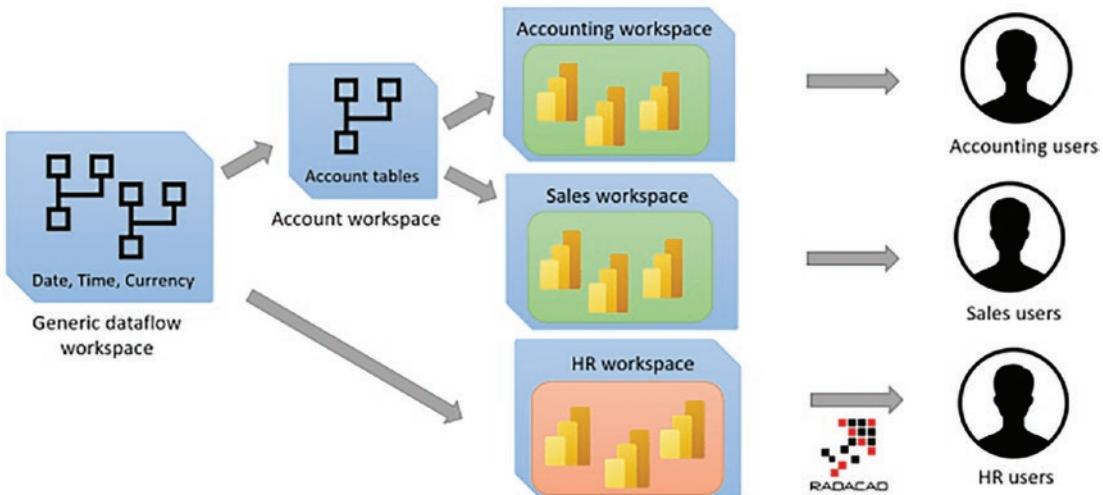


**Figure 40-5.** Sharing Power BI workspaces to reduce redundancy

## Layers of Shared Workspaces

Shared workspaces can have more than just dataflows; they can have datasets too. You can also have layers of shared workspaces. For example, the Date table is likely used by many workspaces. On the other hand, something like the Account table might be only needed in a handful of workspaces (see Figure 40-6).

## Shared workspaces: Layers

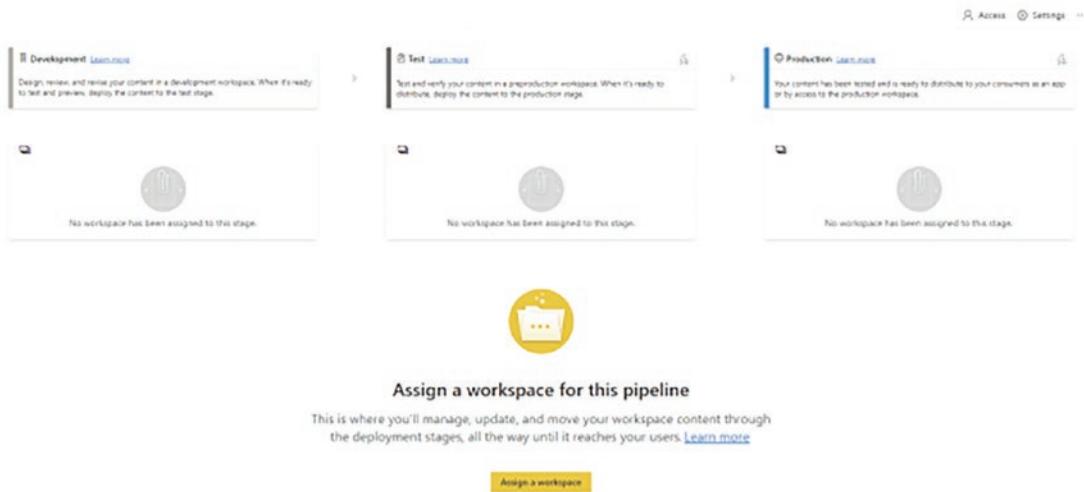


**Figure 40-6.** Layers of Power BI shared workspaces are often needed

## Separating the Environments

Another good use case for having multiple workspaces is to separate environments. For a proper Power BI implementation (or any other software development implementation), you need to have a different environment for the Development, Test, and Production environments. This has many benefits to the development process and will bring trust into the adoption because the content in the production environment will be passed through multiple checks and validations.

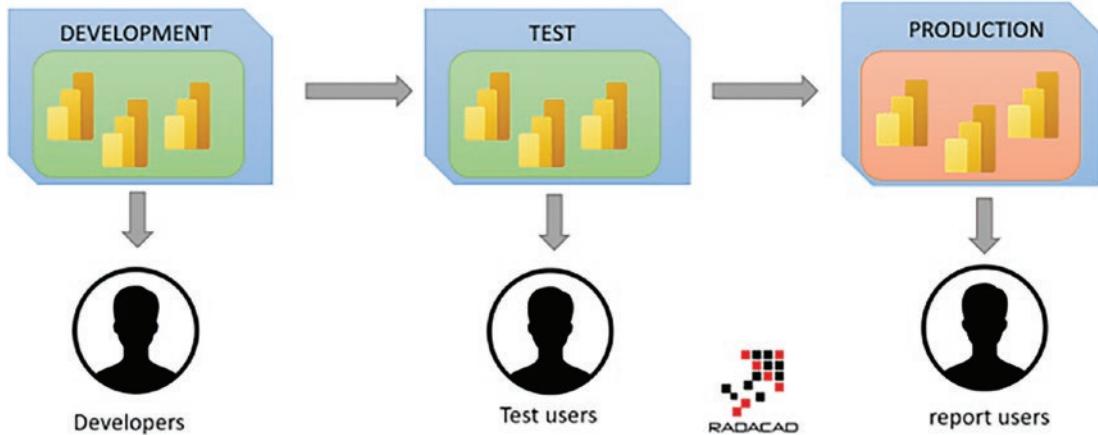
Figure 40-7 shows the deployment pipeline in Power BI Premium, which helps you set up deployment between the environments.



**Figure 40-7.** Deployment pipeline to manage DEV, TEST, and PROD environments

Even if you do not have the Premium license, you can still use the concept of the DEV, TEST, and PROD environments and PowerShell scripts to handle deployment between them (see Figure 40-8).

# Separating environments



**Figure 40-8.** Separating DEV, TEST, and PROD environments in Power BI workspaces

## Workspaces as development layers

Having the DEV, TEST, and PROD environments is not the only workspace structure that helps the development of Power BI solutions. There are some other development practices that you can use by separating workspaces including staging dataflows.

### Staging dataflows

One of the key points of any data integration system is to reduce the number of reads from the source's operational system. In the traditional data integration architecture, this reduction is done by creating a new database called a *staging database*. The purpose of the staging database is to load data as is from the data source into the staging database on a regular schedule.

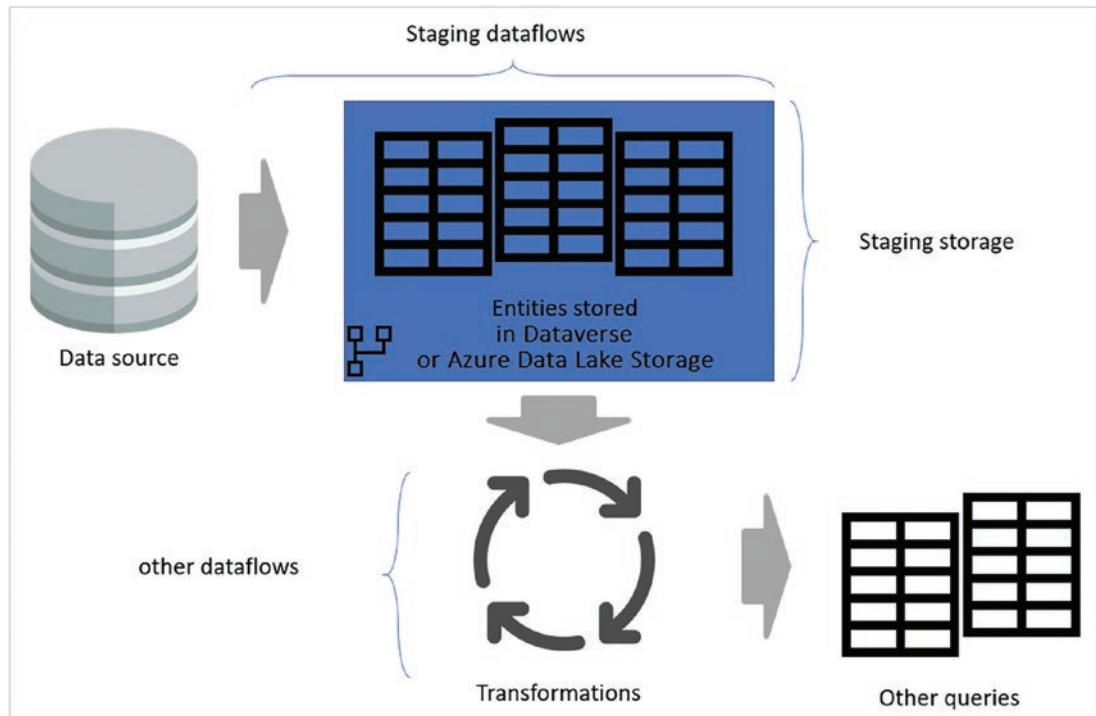
The rest of the data integration will then use the staging database as the source for further transformation and will convert it into the dimensional model structure.

I recommended that you follow the same approach using dataflows. Create a set of dataflows that are responsible for loading data as is from the source system (and only for the tables you need). The result is then stored in the storage structure of the dataflow (either in Azure Data Lake Storage or in a dataverse). This change ensures that the read operation from the source system is minimal.

Next, you can create other dataflows that source their data from staging dataflows. The benefits of this approach include:

- Reduce the number of read operations from the source system and reduce the load on the source system as a result.
- Reduce the load on data gateways if an on-premises data source is used.
- Have an intermediate copy of the data for reconciliation purposes, in case the source system data changes.
- Make the transformation dataflows source-independent.

Figure 40-9 illustrates staging dataflows and staging storage, and it shows the data being accessed from the data source by the staging dataflow. Entities are stored in cadavers or in Azure Data Lake Storage. The entities are then transformed along with other dataflows, which are sent out as queries.

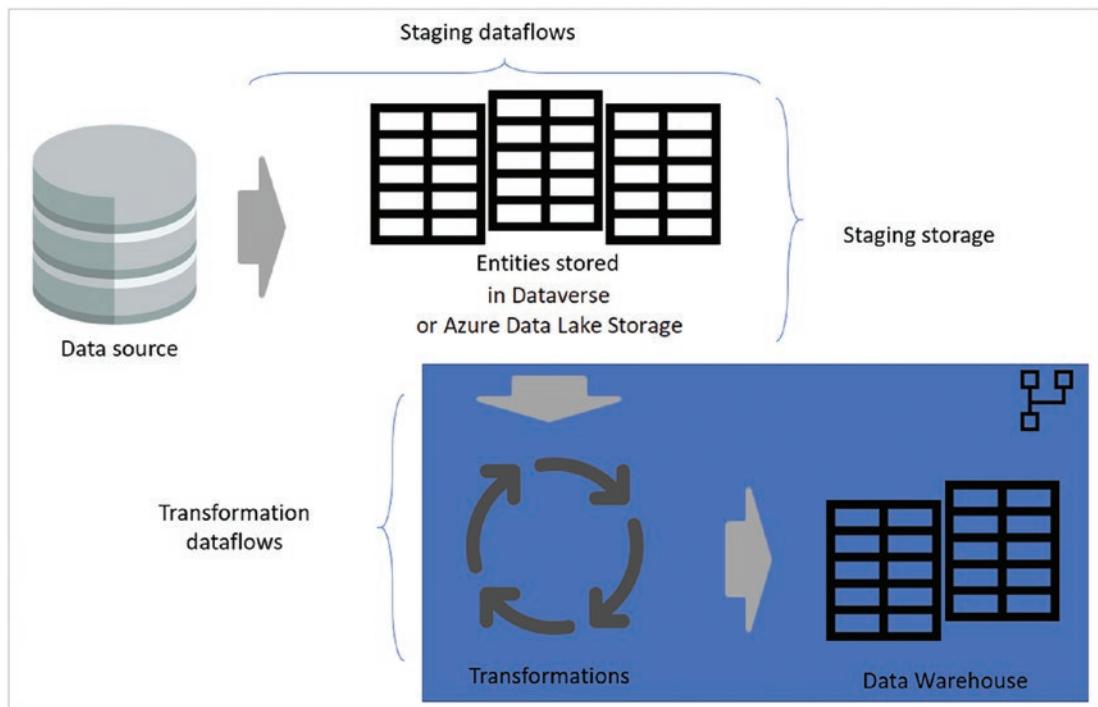


**Figure 40-9.** Staging dataflows

#### Transformation dataflows

When you've separated your transformation dataflows from the staging dataflows, the transformation will be independent from the source. This separation helps if you're migrating the source system to a new system. All you need to do in that case is change the staging dataflows. The transformation dataflows are likely to work without any problems, because they're sourced only from the staging dataflows.

This separation also helps when the source system connection is slow. The transformation dataflow won't need to wait for a long time to get records coming through a slow connection from the source system. The staging dataflow has already done that part, and the data will be ready for the transformation layer. See Figure 40-10.

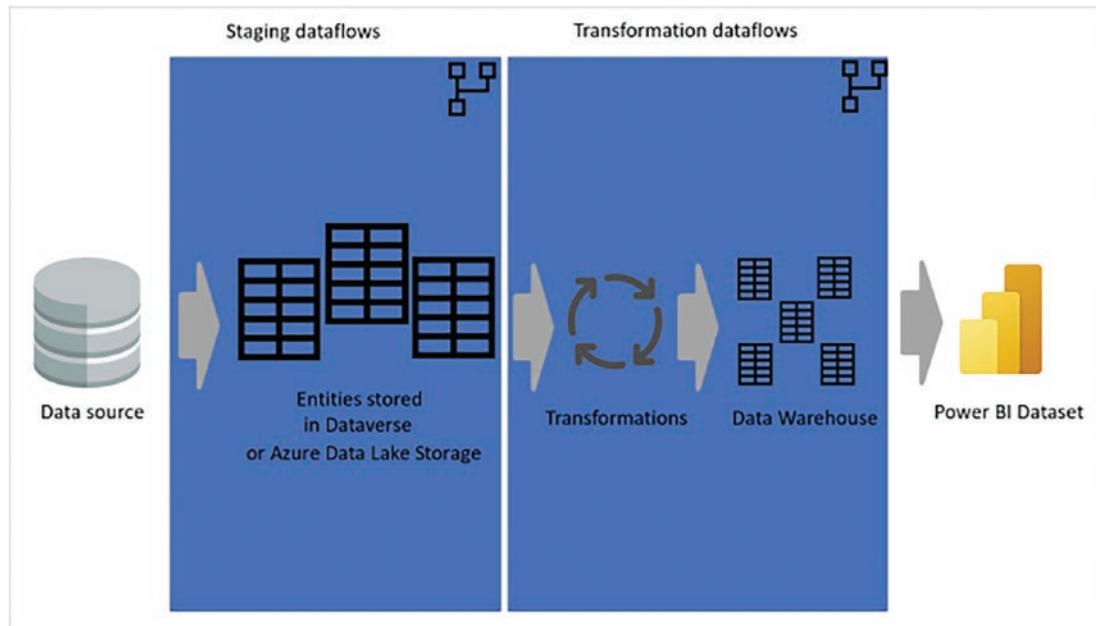


**Figure 40-10.** Separating transformation dataflows from staging dataflows

### Layered architecture

A layered architecture is an architecture in which you perform actions in separate layers. The staging and transformation dataflows can be two layers of a multi-layered dataflow architecture. Trying to do actions in layers ensures the minimum maintenance required. When you want to change something, you just need to change it in the layer in which it's located. The other layers should continue to work fine.

Figure 40-11 shows a multi-layered architecture for dataflows in which their entities are then used in Power BI datasets.



**Figure 40-11.** Multiple layers of development workspaces for Power BI implementation

Figure 40-11 shows multiple workspaces for staging dataflows and transformations. In real-world scenarios, there can be more than these two layers, depending on the complexity of the implementation.

## Summary

To summarize, there are many important factors when organizing a workspace structure. You need to have separate workspaces based on the analysts' groups and then sometimes based on the audience. It is recommended to have DEV, TEST, and PROD layers through workspaces. You must consider using shared workspaces, which reduce redundancy and increase consistency. You can also split the load on the reports using multiple Power BI workspaces.

As you can see, there is a lot to think about when you design the workspace structure. This design will be different from organization to organization. I usually go through many of these processes in my architecture advisory gigs, so I thought it is better to explain some of them here, to help you too.

## CHAPTER 41



# Content Certification and Endorsement

Governance is an important aspect of every application and system. Your Power BI solution architecture needs good rules and processes set up for governance to ensure the content is trustworthy. In Power BI, the content can be endorsed and certified so that it conveys a level of trustworthiness to users. This chapter explains content certification and endorsement and how to use them.

## The Importance of Endorsement

When your organization first begins using Power BI, there may not be a lot of data. But after a while, when the amount of Power BI content (reports, datasets, dataflows, and apps) increases throughout the organization, the need for better governance arises.

If a user faces multiple Power BI objects that present similar information, how does the user know which of these is more accurate? If users can't distinguish the trustworthiness of the content, they might present content that is not correct. Once it's revealed that the information was inaccurate, the trust of the organization in the entire Power BI adoption may collapse. A successful Power BI adoption requires trustworthy Power BI content.

Power BI content endorsement is a labeling system that shows the level of trustworthiness of each piece of content. This system itself is not complicated; it is just labeling. However, it uses an infrastructure for testing and checking reports and certifying them, which helps the entire Power BI trustworthiness and adoption process as a result.

## Levels of Endorsement

The levels of endorsement for the Power BI content follow the principles of the Gold, Silver, and Bronze approach, which is a three-layer approach to determine trustworthiness of content. Gold is the most trusted and Bronze is the least. In the Power BI content endorsement, however, there are different names for it. Power BI content that has the highest level of endorsement is called Certified. The next level down is Promoted, and the last level has no endorsement. By default, Power BI content has no endorsement, which means it is at the lowest level of trustworthiness. See Figure 41-1.



**Figure 41-1.** Levels of certifications

The labels don't really mean anything. These simply label the content. You have to come up with standards and define what exactly these mean. For example, here are some suggestions:

- The Power BI content developed by business analysts in different teams and departments across the organizations, with the data pulled into Power BI from all kinds of data sources, without any peer-review of the quality of the data source or the data quality itself, is considered having no endorsement (or you can call it Bronze-level content).
- Suppose the Power BI content is peer-reviewed by a second team (such as the BI or Data Insight team), and the quality of the data sources is verified. In that case, this is Promoted content, which is the next level up in trustworthiness (or you can call it Silver-level content).
- When the Power BI content is developed by the Data Insight or BI team, or a data analyst competent enough in the technology and understands the business requirements well. If the data sources have been all tested and the quality of the data source and the way in which they are pulled from is fully trusted (such as a data warehouse), and if the measures and reports are all tested and produce high-quality output, then the content can be considered Certified (this is the highest level of trustworthiness, which can be also called Gold-level content).

The endorsement levels should be clearly defined, detailed, and placed on a wiki page or an internal SharePoint site that is easily accessible (the wiki page or the SharePoint site can be linked to the Power BI environments using the Get Help links setup in the Power BI Admin Portal). This will help all the users know what to expect from each level of endorsement. If they see content that has no endorsement, they know what to expect.

## What Kinds of Content Can Be Endorsed?

At the time of writing this chapter, four types of content can be endorsed in the Power BI service environments. These are as follows:

- Power BI datasets
- Power BI dataflows
- Power BI reports
- Power BI apps

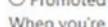
### Endorsement and discovery

Help coworkers find your quality content by endorsing this dataset and making it discoverable. [Learn more](#)



None

This dataset will appear in search results but isn't endorsed.



Promoted

When you're ready to distribute the dataset to your coworkers, promote it to let them know.



Certified

Certify your dataset to show coworkers that it's been reviewed and meets your org's certification criteria. [How do I get my dataset certified?](#)



Make discoverable

Allow users without access to this dataset to discover it and request permissions to access the data. [Learn more](#)

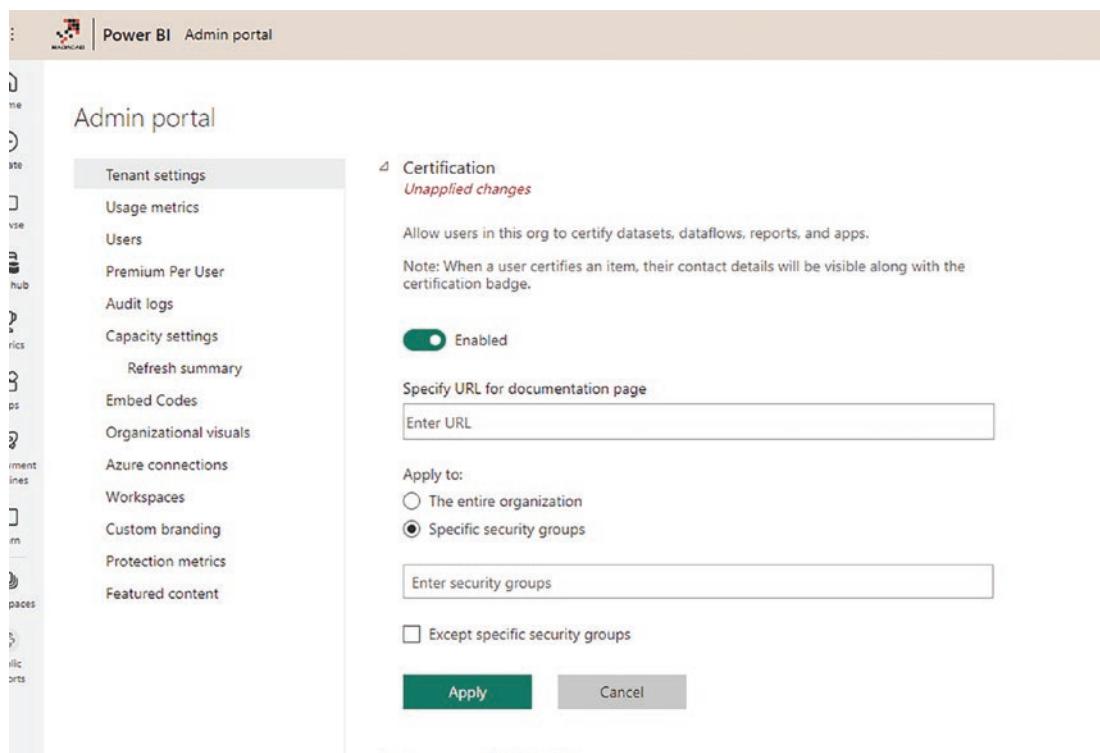
Apply

Discard

**Figure 41-2.** Certifying contents in Power BI

## Who Can Endorse It?

It is important to understand who can endorse content and move it across the level of trustworthiness. It is not recommended to have this functionality open to everyone, because everyone will start certifying their content. You need to set up a specific team and define a specific procedure that the team follows in order to determine if the content is certified-worthy, promoted-worthy, or neither (see Figure 41-3).



**Figure 41-3.** Assigning a specific security group for certifying content

Once they do the test and pass the check, the content can be promoted or certified (see Figure 41-4). This team will then take responsibility for their test because the content is shown as certified under their name.



**Figure 41-4.** The user view of an endorsed report

The Power BI tenant administrator can choose who can certify the contents in the Power BI environment. It is always best to work with Active Directory (or Office 365) groups rather than individual users so that if users are added or removed from the team, it will be easier to manage them in the Office 365 group.

## Dataset Discoverability

As part of certifying datasets, you can check the box to make them discoverable (see Figure 41-5). Ideally, you want the most trusted datasets to be discoverable throughout the organization so that power users can use that contents to build reports with a live connection to the Power BI dataset. This is an important governance step that will increase the adoption of shared datasets in your organization.

### Endorsement and discovery

Help coworkers find your quality content by endorsing this dataset and making it discoverable. [Learn more](#)

None

This dataset will appear in search results but isn't endorsed.

Promoted

When you're ready to distribute the dataset to your coworkers, promote it to let them know.

Certified

Certify your dataset to show coworkers that it's been reviewed and meets your org's certification criteria. [How do I get my dataset certified?](#)

Make discoverable

Allow users without access to this dataset to discover it and request permissions to access the data. [Learn more](#)

This dataset will be made discoverable. Others in your org will be able to find it by such details as name, tables, columns, etc. [Learn more](#)

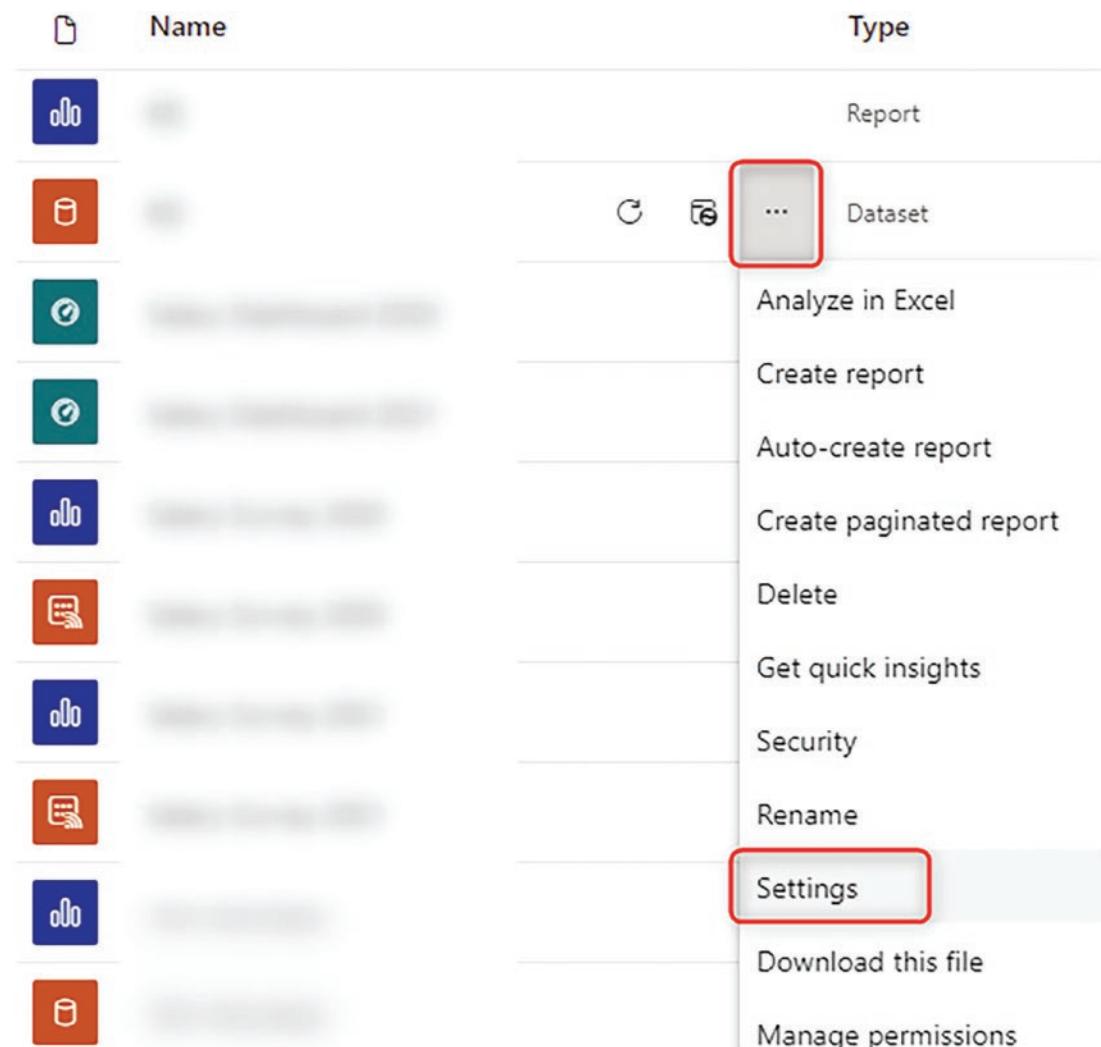
**Apply**

Discard

**Figure 41-5.** Making a Power BI dataset discoverable

## An Example of an Endorsement and Content Certification

To certify content, go to the Power BI Service and choose the Setting area (see Figure 41-6). Then certify it.



**Figure 41-6.** Setting an object in the Power BI service

Once you're in the settings, you can choose the appropriate endorsement (depending on your access level), as shown in Figure 41-7.

#### Endorsement and discovery

Help coworkers find your quality content by endorsing this dataset and making it discoverable. [Learn more](#)

None

This dataset will appear in search results but isn't endorsed.

Promoted

When you're ready to distribute the dataset to your coworkers, promote it to let them know.

Certified

Certify your dataset to show coworkers that it's been reviewed and meets your org's certification criteria. [How do I get my dataset certified?](#)

Make discoverable

Allow users without access to this dataset to discover it and request permissions to access the data. [Learn more](#)

[Apply](#)

[Discard](#)

**Figure 41-7.** Endorsement setup

The content user can see if it is certified or not. Figure 41-8 shows the view of the content from the Power BI Desktop developer's point of view.

The screenshot shows the Data hub interface with a list of datasets. A red box highlights the 'Endorsed in your org' button in the top navigation bar. Another red box highlights the 'Endorsement' column in the main table, which lists various endorsement status for each dataset.

Name	Owner	Location	Refreshed	Endorsement	Sensitivity
[REDACTED]	[REDACTED]	[REDACTED]	9/26/16, 6:31:21 PM	<input checked="" type="radio"/> Certified	-
[REDACTED]	[REDACTED]	[REDACTED]	10/13/19, 3:31:33 AM	<input checked="" type="radio"/> Certified	-
[REDACTED]	[REDACTED]	[REDACTED]	10/16/19, 4:35:55 AM	<input checked="" type="radio"/> Certified	-
[REDACTED]	[REDACTED]	[REDACTED]	4/10/19, 6:07:45 AM	<input checked="" type="radio"/> Certified	-
[REDACTED]	[REDACTED]	[REDACTED]	12/29/20, 10:59:54 AM	<input type="radio"/> Promoted	-
[REDACTED]	[REDACTED]	[REDACTED]	10/3/22, 6:06:53 AM	<input type="radio"/> Promoted	-
[REDACTED]	[REDACTED]	[REDACTED]	4/11/19, 2:05:29 AM	<input type="radio"/> Promoted	-
[REDACTED]	[REDACTED]	[REDACTED]	8/7/19, 10:05:28 AM	<input type="radio"/> Promoted	-

**Figure 41-8.** The endorsed content's view from the Power BI Desktop

Figure 41-9 shows the content from the end user's point of view.

The screenshot shows a Power BI service interface. At the top, it says "Salary Survey 2021" and has a "Certified" button with a red border. To its right is the text "Data updated 12/29/20". Below this, there's a "Get insight" section with a "Title" field containing "Salary Survey 2021", a "Contact" field with "Reza Rad", and a "Data updated" field showing "12/29/20, 10:59 AM". A red box highlights the "Endorsement" section, which contains the text "Certified by Reza Rad". To the right of this section, there's a large number "1" and the word "Average".

**Figure 41-9.** The endorsed content view from the Power BI Service

## Summary

As you learned in this chapter, a certification or endorsement is nothing but a simple labeling system. It is the *process* that makes it important. If everyone in the organization can certify their content, endorsement would be meaningless. If a process isn't defined that clearly explains in detail the difference between certified and promoted content and the tests needed to gain a certification level, then endorsement won't help the Power BI implementation.

Endorsement and certification of the Power BI content are features whereby the *process* defined within the organization is more important than the feature itself. With the right process, you can define a good governance strategy and have a better Power BI implementation, and, as a result, Power BI will be successfully adopted across your organization.

## CHAPTER 42



# Deployment Pipelines

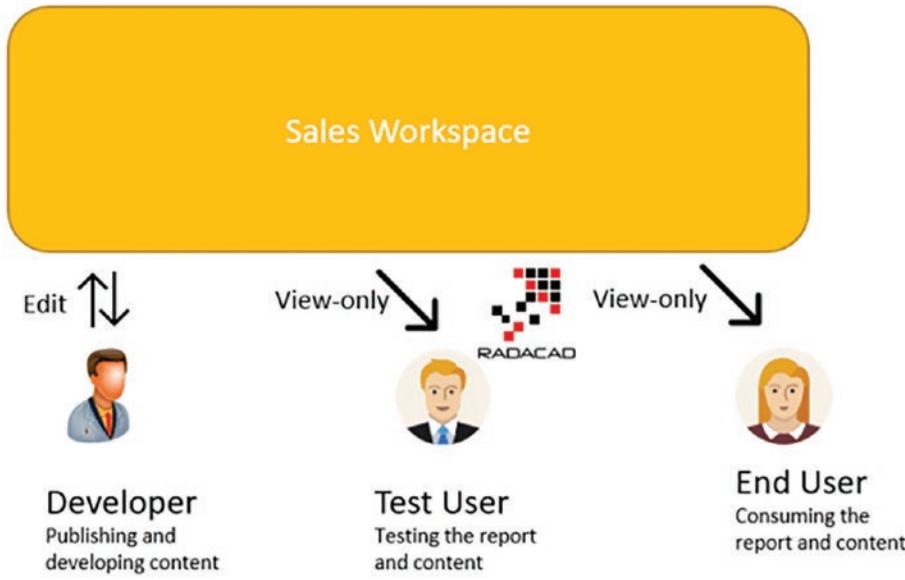
All software applications need multiple environments. Separating the user's environment from the developer's environment comes with many benefits. Power BI is no exception. Having separate environments for different types of users can be helpful in many aspects. This chapter explains how Power BI handles this separation using deployment pipelines.

## Why Multiple Environments?

Before discussing the deployment pipelines, it is important to understand why you need multiple environments in a Power BI implementation. The best way to understand it is to go through a sample scenario.

Jack is a Power BI developer who is part of the data analytics team in his organization. The data analytics team created a workspace for Sales reports, and they asked Jack to publish his reports, datasets, and dataflows to that workspace. Others in the data analytics team also have edit access to that workspace. Some test users use the same workspace to check the reports and the results. The end users connect to this environment through the same workspace. The structure is shown in Figure 42-1.

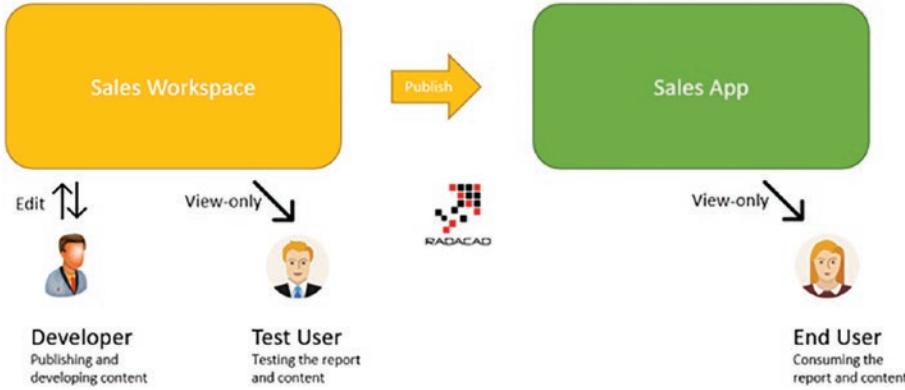
End-user, Developer and Test environments are combined



**Figure 42-1.** Sample Power BI Implementation with all the environments combined

The big problem with this structure is that as soon as the development team applies a change, the test user and the end user are impacted, even if the change is not final. This might lead to frustrating experiences for the end users, resulting in losing their trust in the reports. To separate the end-user experience, the analytics team can use a Power BI app instead.

To use an app, the data analytics team creates a workspace for Sales reports, and they ask Jack to publish his reports, datasets, and dataflows to that workspace. Others on the data analytics team also have edit access to that workspace. Test users use the same workspace to check for reports and results. The end users connect to this environment through an app. This structure is shown in Figure 42-2.



**Figure 42-2.** Sample Power BI implementation with a separate end-user environment

This environment uses one workspace to share between the developers (Jack and the data analytics team) and the test users. It then uses an app on top of that workspace for end users. In this scenario, the developers and end users are separated, but the developer is not separated from the test users. What if the developer is trying some features and is not ready for testing? At the same time, users are testing the content, getting different output, and becoming confused.

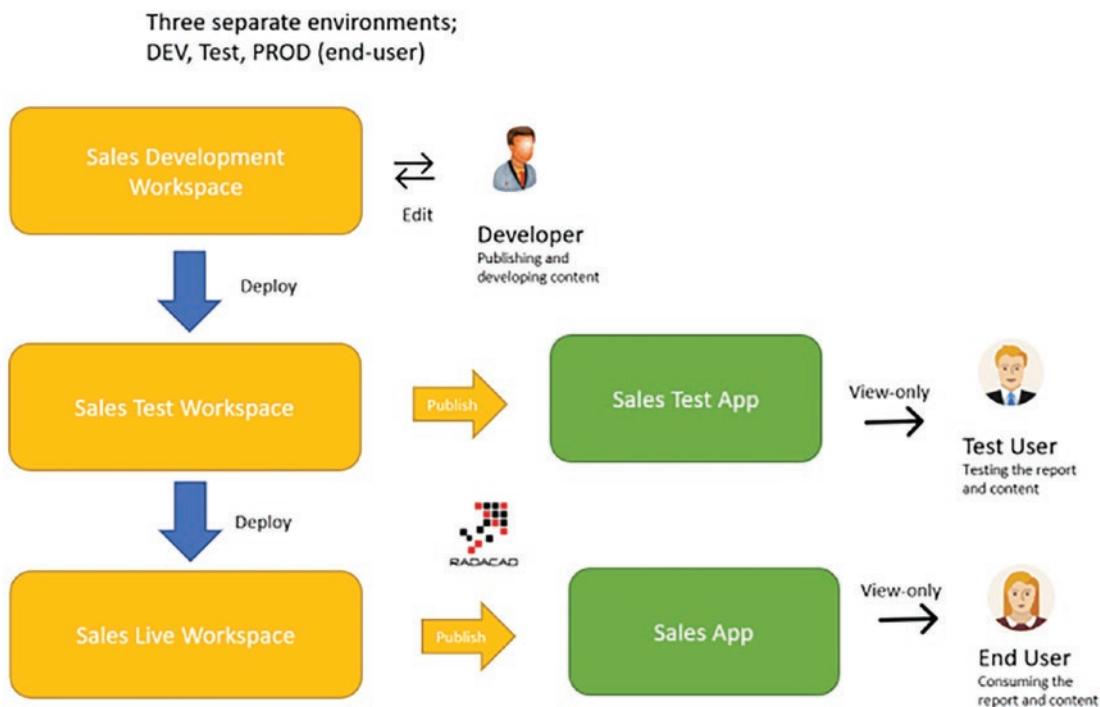
You can separate the developer and test environments by using another workspace. Figure 42-3 shows this new structure.



**Figure 42-3.** Separating the Power BI developer and test user's environment

This structure is much better. When the developer makes changes, the test users won't be impacted. The development team can do all their work, and when they are ready, they can deploy (or copy) the content into the Sales workspace. In that Sales workspace, the test user can check the content before publishing it to the end user.

Although the structure might look perfect, it still has problems. What if this structure uses a shared dataset? In that case, the reports that are shown in the Sales app are connected to the dataset, which is in the workspace, and when the dataset is updated, the reports may show the new results (which might not be correct because the test users haven't tested it yet). So, the environment can change to something like Figure 42-4 shows.



**Figure 42-4.** Three separate environments in Power BI with separate workspaces and apps

The structure in Figure 42-4 separates the test environment using its workspace and app and then the live (or Production/Prod) environment with its workspace and app. This minimizes the risk of unwanted changes appearing in other environments. These levels of isolation for each environment make them easier to change.

This is why having multiple environments in a software application system is recommended. There are different types of users, and separating their environments makes the change process more reliable and, as a result, leads to better adoption of that system.

## How Many Environments Do You Need?

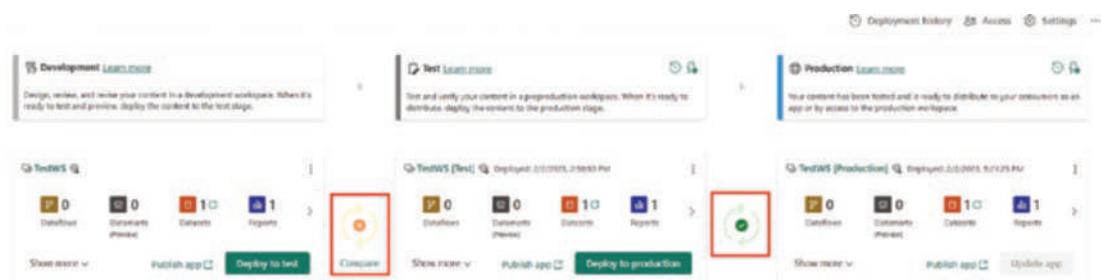
When I explain the structure of multiple environments, I often get the question, do I always need three environments? The answer depends on your solution architecture. Sometimes you may need more, sometimes fewer. Sometimes you may be fine with only two environments—Dev and Live—where the workspace of Live is used for test users. Sometimes you may need four or five environments, depending on roles and levels of test users in your organization.

You must carefully choose the right environment setup based on the Power BI solution architecture, the use of shared components such as dataflows, datasets, and datamarts, the culture of users in your organization, the self-service use of content, and many other factors.

## What Is a Deployment Pipeline?

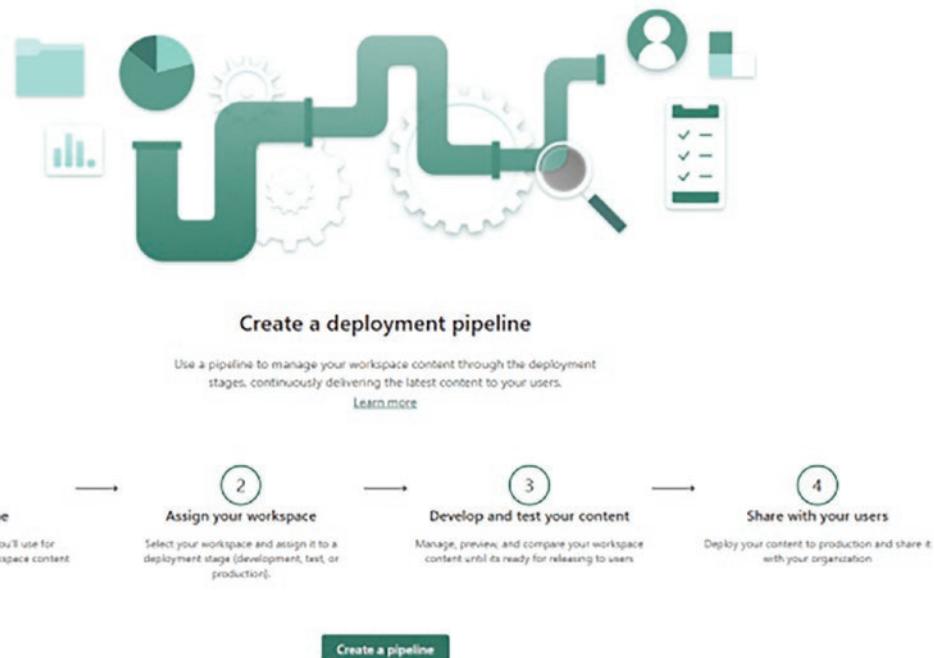
Now that you understand why you need multiple environments, it is the right time to discuss deployment pipelines in Power BI.

When you set up multiple environments, you need a mechanism to determine which environment is which (they may not always be called TEST, DEV, PROD, and so on). You also need a process for copying the content from one environment to another (for example, from DEV to TEST). You also need a process to compare the content in one environment with another, look for changes, and deploy any changes. (For example, comparing the TEST and LIVE environments to see which reports have changed.) You may need to change some connections during this process (for example, connect the TEST dataset to the test data source and the LIVE dataset to the live data source). Many more requirements come with having multiple environments. In other words, you need a tool for managing the deployment between multiple environments. This is where the deployment pipeline comes in (see Figure 42-5).



**Figure 42-5.** Deployment pipelines in Power BI for managing multiple environments  
Source: [learn.microsoft.com/en-us/power-bi/create-reports/deployment-pipelines-compare](https://learn.microsoft.com/en-us/power-bi/create-reports/deployment-pipelines-compare)

Deployment pipelines in Power BI are components of the Power BI Service, and they enable you to manage multiple environments, define each environment, assign workspaces to the environments, compare the contents of two environments, deploy the changes, view histories of deployment, roll back the changes if needed, change the connections through the deployment process, and more. See Figure 42-6.



**Figure 42-6.** Deployment pipelines in Power BI

If you manage multiple environments in Power BI, deployment pipelines are a big help. This is a tool for the deployment team (which usually is a smaller section of the data analytics team in the organization). Deployment pipelines are not an end-user feature; they impact how the end user adopts Power BI in the organization.

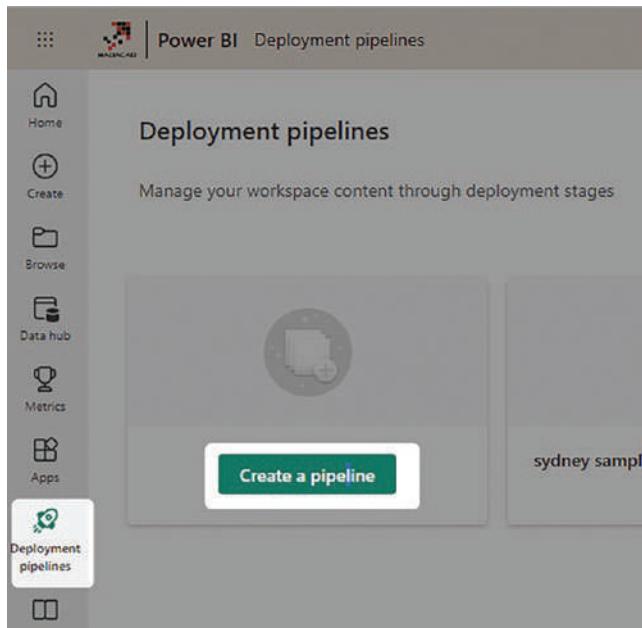
Deployment pipelines are a Premium function in Power BI. This means you need workspaces to be part of a Premium capacity or Premium Per User accounts to use them.

## How Does a Deployment Pipeline Work?

Let's now dive into the experience of deployment pipelines in Power BI and see how they work.

### Creating Deployment Pipelines

To create a deployment pipeline, you must first log in to the Power BI Service and create a pipeline (see Figure 42-7).

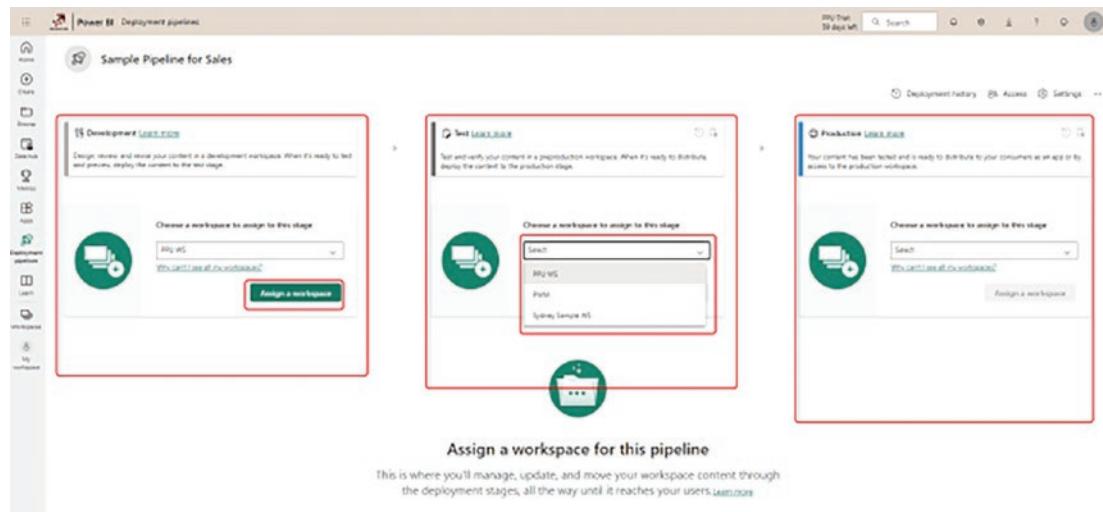


**Figure 42-7.** Creating Power BI deployment pipelines

Add a name and description for the pipeline. The next step is to assign workspaces to each environment.

## Assigning a Workspace

A deployment pipeline in Power BI only comes with an option for three environments. You can assign workspaces to these environments by selecting them in the drop-down (see Figure 42-8).



**Figure 42-8.** Assigning workspaces to the deployment pipelines in Power BI

If you don't see the workspace in the drop-down, it might be because you don't have a Premium workspace created yet, or maybe the workspace is already assigned in another deployment pipeline. Remember that one workspace can be used only in one deployment pipeline.

## Comparing Content

Once the workspaces are assigned, the deployment pipeline summarizes the content in each environment (see Figure 42-9).

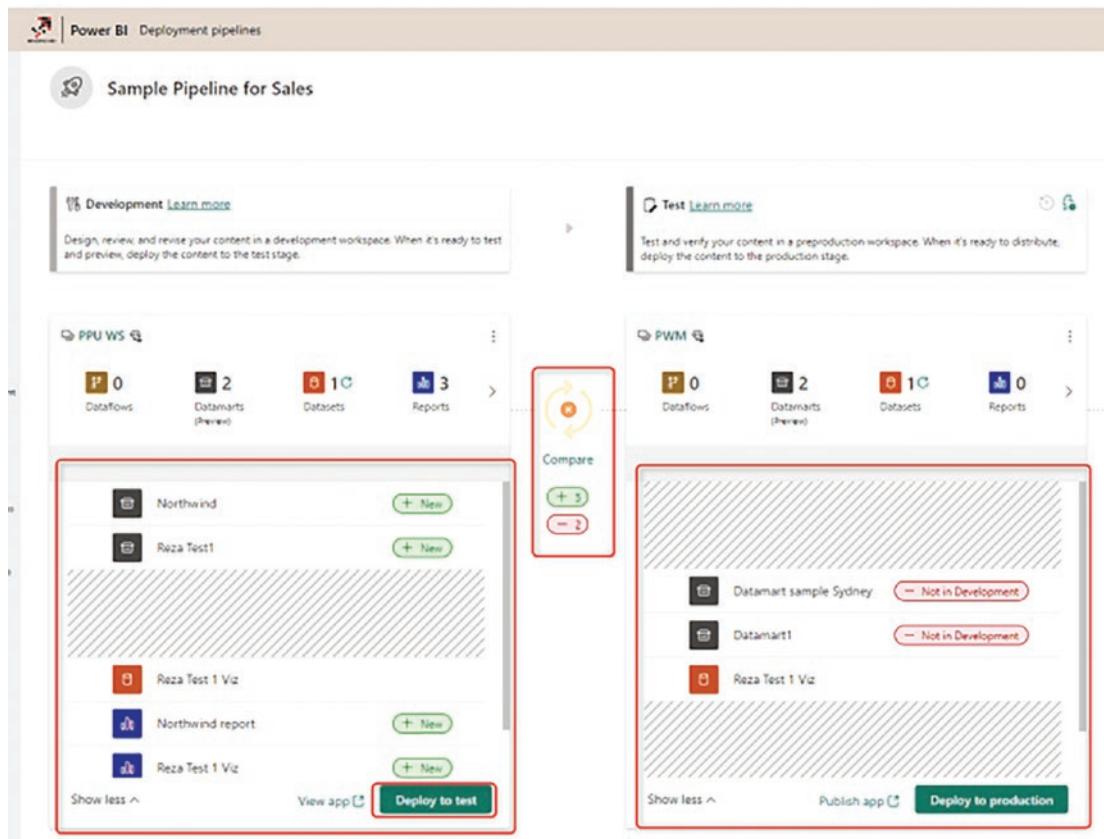
The screenshot shows the Power BI Deployment pipelines interface. At the top, there's a navigation bar with 'My PBI' and '50 days left'. On the right are 'Search', 'Deployment history', 'Access', 'Settings', and a user icon. Below the navigation is a title 'Sample Pipeline for Sales'. Underneath are three cards representing different environments:

- Development (LIVE)**: Shows 15 items. Buttons: 'Show more', 'View app', 'Deploy to test'.
- Test (LIVE)**: Shows 15 items. Buttons: 'Show more', 'Publish app', 'Deploy to production'.
- Production (LIVE)**: Shows 15 items. Buttons: 'Show more', 'Publish app', 'Update app'.

Below these cards are three comparison buttons: 'Compare' between Development and Test, 'Compare' between Test and Production, and 'Compare' between Development and Production.

**Figure 42-9.** The deployment pipeline shows the summary of the content in each environment

You can then compare two of the environments using the Compare action. The Compare process gives you detailed output as to which content items need adding or updating, among other things (see Figure 42-10).

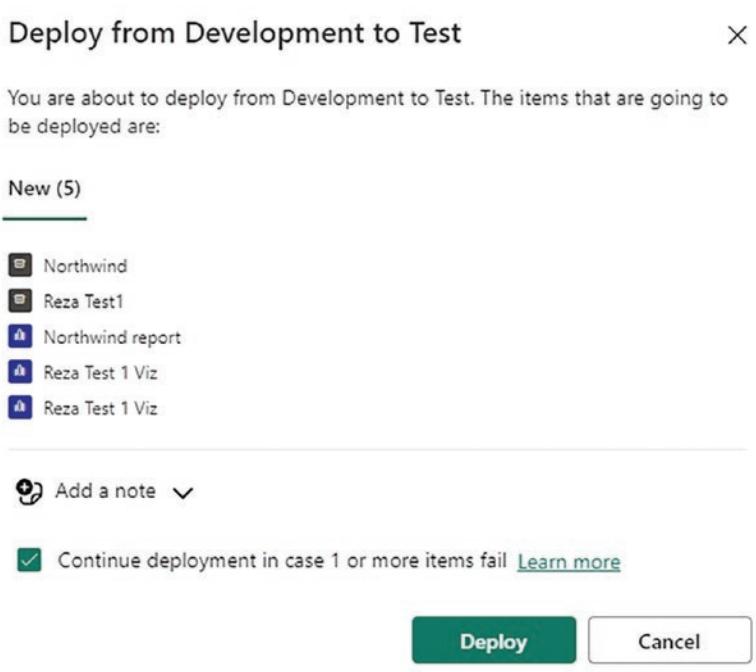


**Figure 42-10.** Comparing content between two environments using the Power BI deployment pipeline

After the comparison, you can choose to deploy. You can deploy to the test if you have compared the development with test environment. If you have compared the test with production, you can deploy it to production.

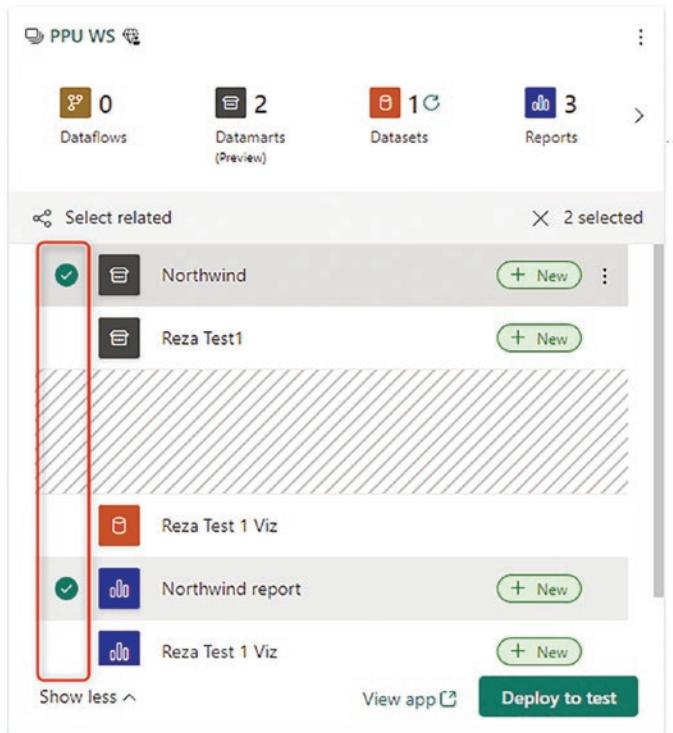
## Deploy

The deploy process is simple to use. You can see the outcome as well, as shown in Figure 42-11.



**Figure 42-11.** Deploy Power BI content

If you want only a few items deployed, select them before clicking the Deploy button, as shown in Figure 42-12.



**Figure 42-12.** Selecting content before deployment

Once the deployment is done, you will get a report of the outcome. You can also compare the two environments after the deployment to see if the changes were applied.

It is important to keep in mind that the deployment will overwrite the destination content (if it exists in the destination).

## Deployment History

One of the most important things in the deployment process is the ability to see the history of deployments and roll them back if needed. Fortunately, this feature exists as part of the deployment pipeline in Power BI (see Figure 42-13).

The screenshot shows a 'Deployment history' report with the following data:

Deployed to	Date and time	Deployed by	Items	Note	ID	Status
Test	07/31/22, 10:23 AM	RP Rene Pellet...	<span style="background-color: yellow;">- 6</span> <span style="background-color: green;">+ 3</span> <span style="background-color: blue;">= 64</span>	<span style="border: 1px solid black; padding: 2px;">?</span> <span style="border: 1px solid green; padding: 2px;">i</span> <span style="border: 1px solid green; padding: 2px;">✓</span>		
Test	07/06/22, 4:58 PM	RP Rene Pellet...	<span style="background-color: green;">+ 74</span>	<span style="border: 1px solid green; padding: 2px;">i</span> <span style="border: 1px solid green; padding: 2px;">✓</span>		
Test	06/06/22, 10:14 AM	RP Rene Pellet...	<span style="background-color: green;">+ 1</span>	<span style="border: 1px solid black; padding: 2px;">?</span> <span style="border: 1px solid green; padding: 2px;">i</span> <span style="border: 1px solid green; padding: 2px;">✓</span>		
Test	06/02/22, 4:41 PM	RP Rene Pellet...	<span style="background-color: blue;">= 1</span>	<span style="border: 1px solid black; padding: 2px;">?</span> <span style="border: 1px solid green; padding: 2px;">i</span> <span style="border: 1px solid green; padding: 2px;">✓</span>		
Test	06/02/22, 4:32 PM	RP Rene Pellet...	<span style="border: 2px solid red; border-radius: 15px; padding: 2px 10px;">Failed</span>	<span style="border: 1px solid red; padding: 2px;">?</span> <span style="border: 1px solid red; padding: 2px;">i</span> <span style="border: 1px solid red; padding: 2px;">✗</span>		

**Figure 42-13.** Deployment history

Source: [learn.microsoft.com/en-us/power-bi/create-reports/deployment-pipelines-history](https://learn.microsoft.com/en-us/power-bi/create-reports/deployment-pipelines-history)

The history also has a detailed result that can be helpful for detailed checks.

## Rules and Connections

If you want to set up different connection configurations for each environment, you can do that using deployment rules. For example, you can change the connection from the dataset to the data source when you deploy it from one environment to another. This function works even better when you use Power Query parameters.

## Automate the Deployment

A user manually triggers deployment pipelines. However, you can automate that process using the Power BI REST API.

## What If There Is No Premium License?

Deployment pipelines are a Premium feature in Power BI. If you don't have Premium licensing and still want to use this feature, you must implement it on your own. Fortunately, the REST API for Power BI can greatly help. You can use that to get the content of one workspace and then compare it with another. You can use the same REST API to download the content from one workspace and then upload it to the destination workspace. You can also change the connection details of content when publishing it. In short, it is possible to build a functionality like the deployment pipelines using PowerShell scripts and REST API calls. Still, it would take some time, and you would need to maintain the script, as the REST API functions might change. It is, however, a method that doesn't require Premium licensing.

## Summary

Having multiple environments in a Power BI implementation is a very important aspect that helps the adoption of Power BI in your organization. You can create separate environments and manage the deployment process between these environments using deployment pipelines in Power BI. Deployment pipelines are a Premium function in Power BI. Still, they provide many great features, making them a very helpful tool for deployment.

Deployment pipelines in Power BI are only part of the big deployment strategy. Other technologies can be used with deployment pipelines, such as Azure DevOps and GitHub Actions, and combining the Power BI deployment pipelines (using their REST API) with those technologies gives you a more comprehensive pipeline solution.

## CHAPTER 43



# Data-Level Security

Power BI supports data security at the dataset level. This means everyone can see the data they are authorized to see. There are different levels of authorization in Power BI, including row-level security, column-level security, and object-level security. These levels help Power BI developers create one dataset and give users different views of the data from the same report. This chapter explains each of those methods and provides guidance on how to use them.

## Introduction

If you want to create one piece of content and give users different views of it, data-level security is what you need. There are four levels of data-level security, as follows:

- Row-level security
- Column-level security
- Object-level security
- Page-level security

Among these four options, at the time of writing this chapter, the last one (page-level security) is not yet supported in Power BI. However, this chapter explains some workarounds for it.

## Row-Level Security

Row-level security is perhaps the most common type of when discussing security data in the Power BI world. Let's understand it by looking at an example.

Jack built a Power BI report for the sales department. However, his organization has five different sales branches, one in each country. Each country's sales manager should see only their country's data, not others. (For example, Diana, the sales manager in the United States, should only see the U.S. data.) Everything else about the report and the layout and calculations is the same.

In such a scenario, your first solution might be to create five copies of the same report and filters in each report for each country, then share the report of each country with the sales managers of that country. This is not a good solution because you will have five reports to manage. Every change means you must create five copies or apply the change in five reports. This is not a maintainable solution. What if there were 20 countries? You can imagine the hardship of maintaining that many copied reports.

Suppose the data is stored in a table like the one shown in Figure 43-1. There is a column for Country, and the country values are in each table row.

User	Country
Diana	USA
Mike	UK

**Figure 43-1.** Sample data for row-level security

The right way to implement this is to apply a security filter on the data and assign that security filter to the users. In that case, when users log in to the Power BI report, they can only see their own data and no one else's. You don't need copies of the report because everyone sees the data based on the security filter applied for their users. Because this type of security is applied at the data row level, it is called row-level security, sometimes RLS for short.

Row-level security is applied at the dataset level. That means even if you have reports with a live connection to the dataset, they will still follow the settings applied for each user. Maintaining a row-level security solution is simple because you only need to configure the security rules rather than have multiple copies of the same table.

Now that you know what row-level security is, let's look at how it is implemented.

## Static Row-Level Security

Static row-level security is the simplest way to implement RLS. The following section details an example of how static RLS can be implemented.

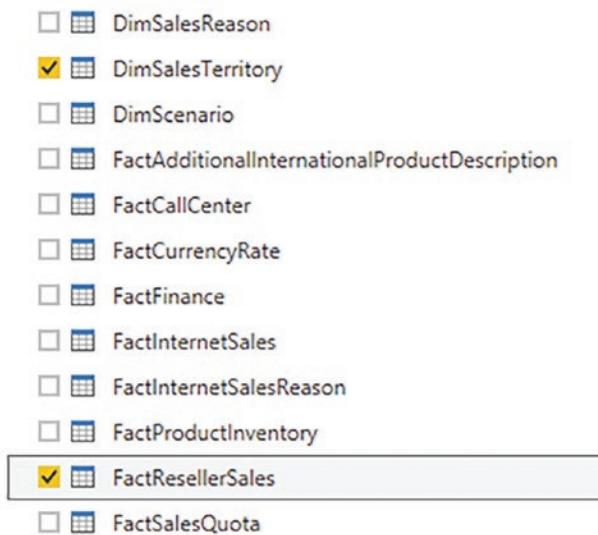
For this example, I use the AdventureWorksDW Excel sample data source. You can download it from [radacad.com/files/AdventureWorksDW2012.xlsx](http://radacad.com/files/AdventureWorksDW2012.xlsx). There is a DimSalesTerritory table in this dataset with country details, as shown in Figure 43-2. This example applies security filters to this table.

SalesTerritoryKey	SalesTerritoryAlternateKey	SalesTerritoryRegion	SalesTerritoryCountry	SalesTerritoryGroup	CountryImage
1	7	Northwest	United States	North America	C:\Users\Reza\Dropbox\Speaking\TechDays Hong Kong 2014\Power View and Pcs
2	2	Northeast	United States	North America	C:\Users\Reza\Dropbox\Speaking\TechDays Hong Kong 2014\Power View and Pcs
3	3	Central	United States	North America	C:\Users\Reza\Dropbox\Speaking\TechDays Hong Kong 2014\Power View and Pcs
4	4	Southwest	United States	North America	C:\Users\Reza\Dropbox\Speaking\TechDays Hong Kong 2014\Power View and Pcs
5	5	Southeast	United States	North America	C:\Users\Reza\Dropbox\Speaking\TechDays Hong Kong 2014\Power View and Pcs
6	6	Canada	Canada	North America	C:\Users\Reza\Dropbox\Speaking\TechDays Hong Kong 2014\Power View and Pcs
7	7	France	France	Europe	C:\Users\Reza\Dropbox\Speaking\TechDays Hong Kong 2014\Power View and Pcs
8	8	Germany	Germany	Europe	C:\Users\Reza\Dropbox\Speaking\TechDays Hong Kong 2014\Power View and Pcs
9	9	Australia	Australia	Pacific	C:\Users\Reza\Dropbox\Speaking\TechDays Hong Kong 2014\Power View and Pcs
10	10	United Kingdom	United Kingdom	Europe	C:\Users\Reza\Dropbox\Speaking\TechDays Hong Kong 2014\Power View and Pcs
11	0	NA	NA	NA	C:\Users\Reza\Dropbox\Speaking\TechDays Hong Kong 2014\Power View and Pcs

**Figure 43-2.** Sample data source for establishing static row-level security

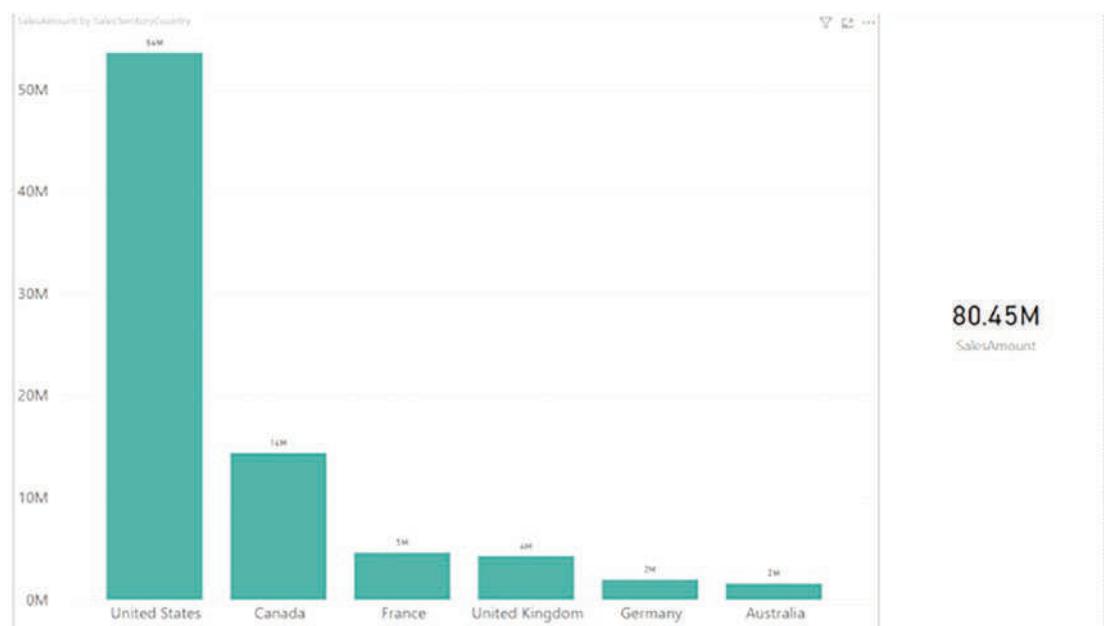
## Creating a Sample Report

Let's start by creating a sample Power BI Desktop report from the AdventureWorks Excel file. This example uses DimSalesTerritory and FactResellerSales (see Figure 43-3).



**Figure 43-3.** Creating a sample Power BI Desktop report

Without making any changes to the Power Query Editor, load the report and build a simple column chart using Sales Amount (from FactResellerSales) and Country (from DimSalesTerritory). The chart shows sales by country, which can be used to create row-level security on geo-location information. Now add one card visualization for the total sales amount. Figure 43-4 shows the layout of this sample report.

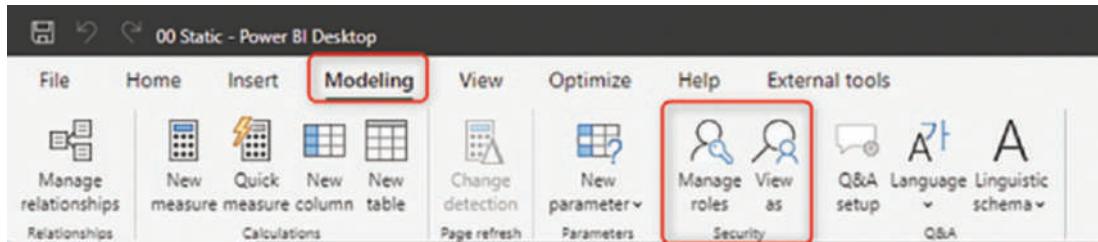


**Figure 43-4.** Layout of the sample report

In this view, the total reseller sales amount is \$80M, and you have sales values for Australia, Canada, France, Germany, the UK, and the United States. Next, you see how to create some roles.

## Creating Roles

The goal is to build roles for the sales managers in Australia and Canada. They should each only see their group in the dataset. To create roles, go to the Modeling tab on the Power BI Desktop (see Figure 43-5). You will see a section named Security.



**Figure 43-5.** Creating roles in the Modeling tab

Click Manage Roles to create a new role. You will see the Manage Roles window, which has three panes, as shown in Figure 43-6.



**Figure 43-6.** The Manage Roles window

You can create or delete roles in the number one pane. You can see tables in your model in the number two pane. Then you can write your DAX filtering expression in the number three pane, or you can use the user-interface profited to define the filters.

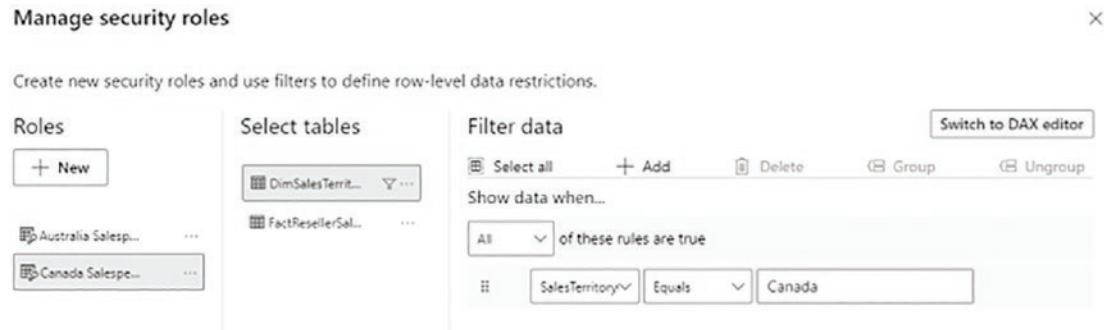
Create a role and name it Australia Sales Manager. You will see two tables in the Tables section—FactResellerSales and DimSalesTerritory. Click Add to add the rule you want to the SalesTerritoryCountry column (see Figure 43-7).

**Figure 43-7.** Creating a role for the Australian sales manager

You can also switch to the DAX Editor and see the DAX expression (see Figure 43-8).

**Figure 43-8.** Switching to the DAX Editor

Now create another role. Call it Canada Sales Manager, use the SalesTerritoryGroup filter this time, and change the value to Canada, as shown in Figure 43-9.



**Figure 43-9.** Creating the Canadian sales manager role

## Testing Roles on the Desktop

Now that you have created two sample roles, you can test them. You can test them using the Power BI Desktop with the View As Roles menu option. This option allows you to view the report exactly as the user will see it. You can even combine multiple roles and see a consolidated view of a person who has multiple roles. Go to the Modeling tab and choose the View As Role option (see Figure 43-10).

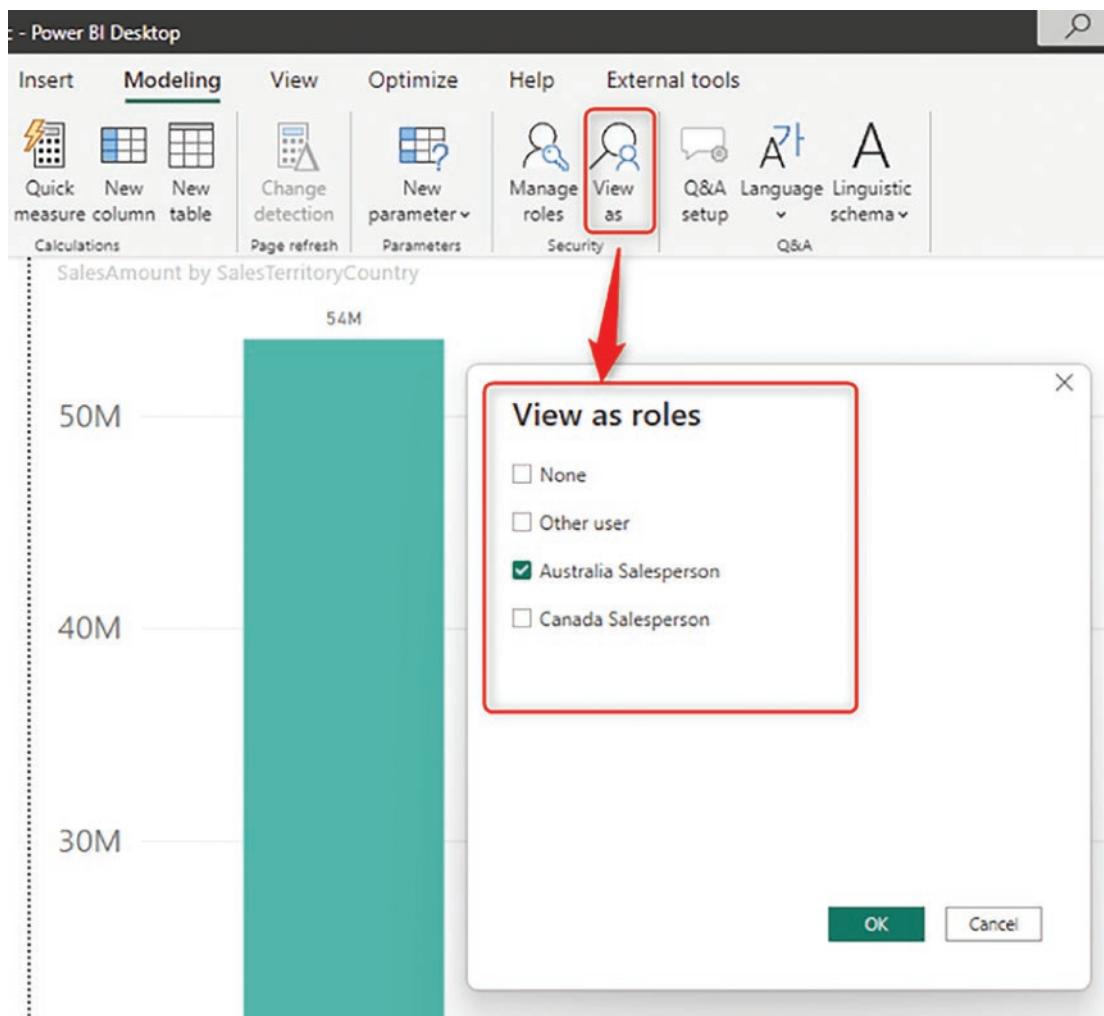


Figure 43-10. The View As Role option

Choose the Australian and Canadian sales managers and click OK. You will see sales for Australia and Canada only, with a total of \$15.97M; you'll also see only the Australia and Canada countries. See Figure 43-11.

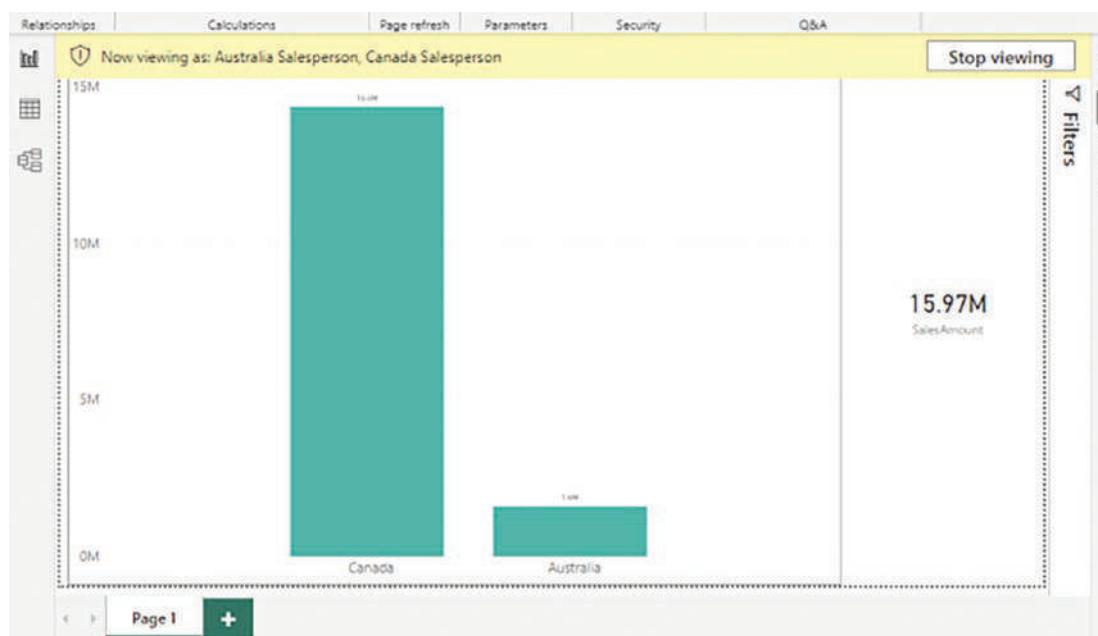


Figure 43-11. Displaying sales data for Australia and Canada

At the top of the report, there is a highlighted information line indicating that the view is showing the Australian and Canadian sales managers. If you click Stop Viewing, you will see the report in normal view (total view).

## Assigning Users to Roles in the Power BI Service

Roles should be assigned to Power BI users (or accounts, in other words), and this should be done in the Power BI Service. Save and publish the report in Power BI. I named this report `00_Static`. You can name it whatever you want. After publishing the report, choose Security for the dataset (see Figure 43-12).

The screenshot shows the Power BI 'My workspace' interface. On the left is a sidebar with icons for Home, Create, Browse, Data hub, Metrics, Apps, Deployment pipelines, Learn, Workspaces, and My workspace (which is selected). The main area has a title 'My workspace' with a user icon. Below it are buttons for '+ New' and 'Upload'. A navigation bar at the top includes 'All', 'Content', and 'Datasets + dataflows', with 'All' being the active tab. The main content area displays a table with columns for Name and Type. The table contains the following data:

Name	Type
00 Static	Report
00 Static	Dataset
01 Dynamic	Analyze in Excel
01 Dynamic	Create report
20210608	Auto-create report
20210608	Create paginated report
20210608	Delete
20221115	Quick insights
	Security
	Rename
	Settings

A red box highlights the '...' button next to the second dataset row, and another red box highlights the 'Security' option in the context menu for the last row.

Figure 43-12. Saving and publishing the report in Power BI

Here, you can see roles and assign them to Power BI accounts in your organization (see Figure 43-13).

The screenshot shows the Power BI service interface with the title "Row-Level Security". On the left, there is a vertical navigation bar with icons for Home, Create, Browse, Data hub, Metrics, and Apps. The main area displays two roles: "Australia Sales Manager (1)" and "CANADA Sales Manager (1)". A red box highlights the "Members (1)" section for the "Australia Sales Manager" role. This section contains a list with one member, "Reza Rad", and an "X" button to remove them. Below the list is a text input field labeled "Enter email addresses" and a "Add" button.

**Figure 43-13.** Assigning roles to Power BI accounts

You can set each user to more than one role, and the user will then have a consolidated view of both roles. For example, a user with Australia and Canada sales manager roles will see data from Australia and Canada.

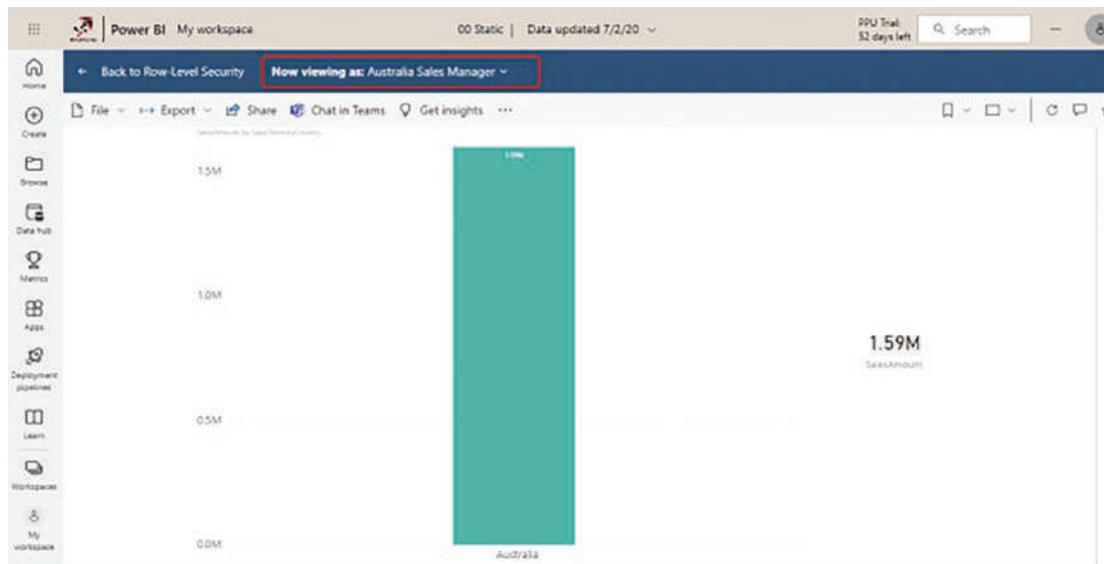
## Testing Roles in the Power BI Service

You can also test each role. Just click the ellipsis button beside the role and then choose Test as Role (see Figure 43-14).

This screenshot is similar to Figure 43-13, showing the "Row-Level Security" page. The "Australia Sales Manager (1)" role is selected, and its details are shown. A red box highlights the ellipsis button next to the role name. A context menu has appeared, with the "Test as role" option highlighted by a red box. Other options in the menu include "Edit role" and "Delete role". To the right of the menu, the "Members (1)" section is visible, showing the member "Reza Rad" and the "Add" button.

**Figure 43-14.** Testing a role

The Test as Role option shows you the report in view mode for that role. You can change the role if you like. See Figure 43-15.



**Figure 43-15.** Displaying a report in view mode

Setting users in each role enables row-level security. If the user logs in with their account, they will only see data for their roles.

## Republishing Won't Hurt

As mentioned in the first paragraph of this chapter, the great thing about this new feature is that RLS is part of the Power BI model. If you publish your Power BI model repeatedly with changes, you won't lose web configuration. You also won't lose users assigned to each role if you don't change the role names.

You've learned about a specific type of row-level security called static row-level security. It is called static because the filter values are statically determined in DAX expressions. Maintenance costs are very high if you want to apply such a filter to thousands of roles. In an ideal world, you want to apply security automatically based on users' login.

## Dynamic Row-Level Security

Although static RLS is simple to set up, it is hard to maintain when there are many roles. Dynamic row-level security makes maintenance of the RLS much simpler. However, it requires more steps to set up. The following section is an example of dynamic RLS.

## Sample Data

In this example, I use data entered in Power BI. There aren't any external data sources. This doesn't mean that dynamic security can't work with external data sources. Dynamic security works with any data sources as long as you have related data rows in the tables. For the simplicity of this example, it uses data sources inside Power BI.

For this example, you can create two simple tables—Sales Rep and Transactions. Sales Rep has information about sales representatives, and Transactions contains information about sales transactions. Each sales transaction is handled by a sales rep. You can create sample tables in Power BI. Open the Power BI Desktop, and, from the Data section, choose Enter Data.

Create a table like the one shown in Figure 43-16, with three columns of data in it. You have to use usernames similar to Power BI accounts that you want to set up security for. Name this table Sales Rep.

ID	Name	Email
1	Reza Rad	reza-zb\reza_
2	Leila Etaati	leila@radacad.com
3	Jack Horlock	jack.horlock@radacad.com
4	xyz	student2@radacad.com

**Figure 43-16.** Sample data used to establish dynamic row-level security

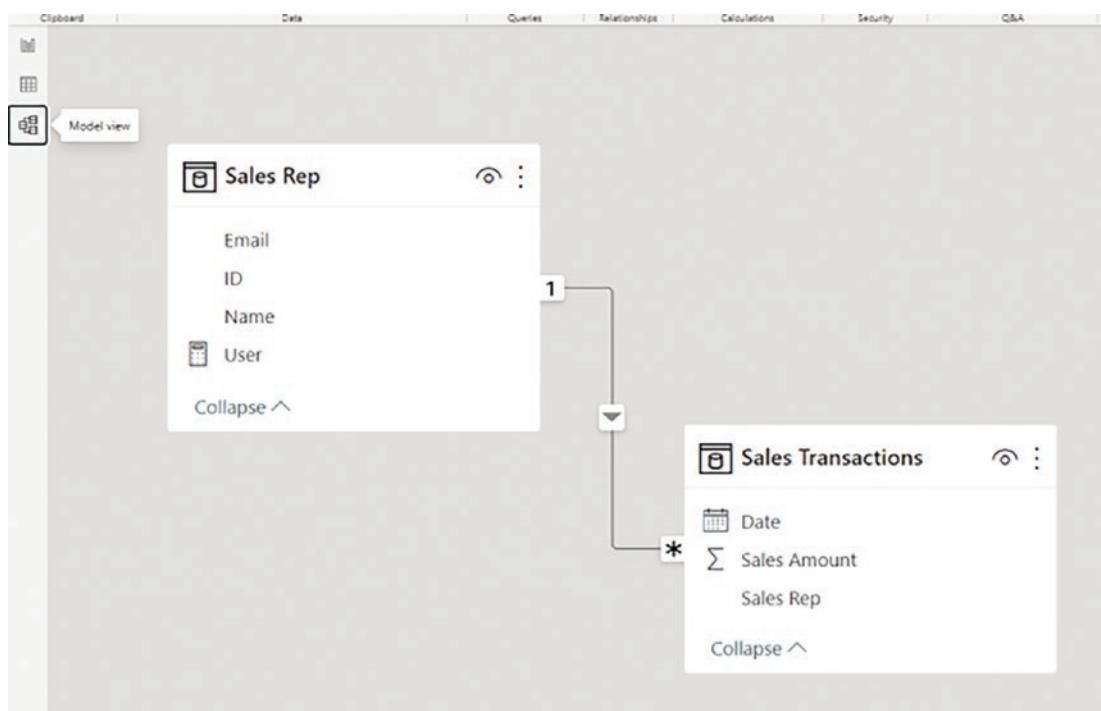
Create another table for transactions using the structure shown in Figure 43-17 and name it Transactions.

Date	Sales Rep	Sales Amount
Sunday, January 1, 2017	1	\$100
Wednesday, February 1, 2017	2	\$300
Wednesday, March 1, 2017	1	\$50
Monday, February 12, 2018	3	\$1,000,000

**Figure 43-17.** Sample transactions table

As you can see in Figure 43-17, each sales transaction is handled by a sales rep. Again I mention that these tables are added to Power BI for the simplicity of this example. Tables can come from everywhere.

Load the tables into Power BI; you don't need to do anything with Power Query at this stage. Go to the Relationship tab and verify the relationship between Sales Rep (ID) and Transactions (Sales Rep) to be as shown in Figure 43-18.



**Figure 43-18.** Verifying the relationship between Sales Rep (ID) and Transactions (Sales Rep)

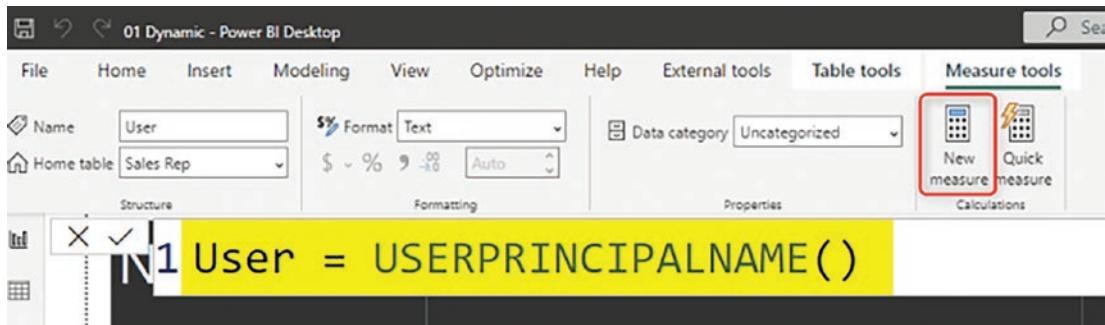
## Sample Report

This example uses basic table visualization. The table visualization shows the Date, Sales Amount (from Transactions), and Name (from Sales Rep), as shown in Figure 43-19.

Name	Email	Year	Month	Day	Sales Amount
Jack Horlock	jack.horlock@radacad.com	2018	February	12	\$1,000,000
Leila Etaati	leila@radacad.com	2017	February	1	\$300
Reza Rad	reza-zb\reza_	2017	January	1	\$100
Reza Rad	reza-zb\reza_	2017	March	1	\$50
<b>Total</b>					<b>\$1,000,450</b>

**Figure 43-19.** Sample report data

The main reason for this visualization is to simply show that each user will see only their data rows from all tables. I also add a measure for `USERPRINCIPALNAME()` in DAX to see the user logged in from my report. So in the Data tab, create a new measure (see Figure 43-20) and name it User, with a value of `USERNAME()`.



**Figure 43-20.** Creating a new measure

Now you can add a Card visualization to the report. Add the User measure to the card visual. Figure 43-21 shows the report's final view.

The screenshot displays a Power BI report with a card visualization. The card contains the text 'REZA-ZB\reza\_'. To the right of the card, a red arrow points from the 'User' item in the 'Filters' pane to the card itself, indicating that the user measure is being displayed. The report also includes a table visualization showing sales data for four users.

Name	Email	Year	Month	Day	Sales Amount
Jack Horlock	jack.horlock@radacad.com	2018	February	12	\$1,000,000
Leila Etaati	leila@radacad.com	2017	February	1	\$300
Reza Rad	reza-zb\reza_	2017	January	1	\$100
Reza Rad	reza-zb\reza_	2017	March	1	\$50
<b>Total</b>					<b>\$1,000,450</b>

**Figure 43-21.** A card visualization with the user measure

## The UserName() and UserPrincipalName() DAX Functions

The USERNAME() function in DAX returns the username of the logged-in user. However, there is a small trick to it. If you don't set up row-level security on your report, the USERNAME() function will return the user ID. To understand what I mean, publish your report to Power BI and browse it.

The UserPrincipalName() function in DAX works similarly to the UserName() function, with the difference that it always returns the username (not the unique identifier). Basically, UserPrincipalName() is a better function for testing, but it works the same in a production environment. Now you'll set up row-level security and assign users to it to see how it works. See Figure 43-22.

The screenshot shows a Power BI report with a table containing sales data. The columns are Name, Email, Year, Month, Day, Sales, and Amount. The data includes four rows: Jack Horlock, Leila Etaati, Reza Rad, and another Reza Rad entry. A bolded 'Total' row at the bottom shows a total sales amount of \$1,000,450. Below the table, a 'User' section displays the email address 'Reza@RADACAD.onmicrosoft.com' with a red border around it.

Name	Email	Year	Month	Day	Sales	Amount
Jack Horlock	jack.horlock@radacad.com	2018	February	12	\$1,000,000	
Leila Etaati	leila@radacad.com	2017	February	1	\$300	
Reza Rad	reza@radacad.com	2017	January	1	\$100	
Reza Rad	reza@radacad.com	2017	March	1	\$50	
<b>Total</b>					<b>\$1,000,450</b>	

Figure 43-22. Setting up row-level security and assigning users

## Row-Level Security in the Power BI Desktop

This example uses the row-level security technique to filter each role based on the username with the DAX `username()` function. To create security, go to the Modeling tab and choose Manage Roles. Create a role and name it Sales Rep. Then define a filter on the Sales Rep table, as shown in Figure 43-23.

`[Username] = USERPRINCIPALNAME()`

The screenshot shows the 'Manage security roles' dialog in Power BI Desktop. It has three main sections: 'Roles' (with a '+ New' button), 'Select tables' (listing 'Sales Rep' and 'Sales Transaction'), and 'Filter data'. In the 'Filter data' section, there is a single-line formula: `[Email] = UserPrincipalName()`. This formula filters the 'Sales Rep' table based on the user's email address.

Figure 43-23. Filtering roles based on username

This filter simply indicates that the logged-in user will only see their records in the whole dataset. As you might remember, the `username` field in the Sales Rep table is defined as the usernames of the Power BI accounts. The Transactions table is also related to this table and is based on Sales Rep ID. So filtering one table will affect others. As a result, this single-line filter will enable dynamic row-level security in the whole Power BI solution.

## Assigning Users to Power BI Security

Now save and publish your solution to Power BI. In the Power BI Service, go to the Security setting of the dataset you just published (I called mine 01 Dynamic), as shown in Figure 43-24.

The screenshot shows the Power BI Service interface with the title bar "Power BI My workspace". On the left is a sidebar with icons for Home, Create, Browse, Data hub, Metrics, Apps, Deployment pipelines, Learn, Workspaces, and My workspace. The main area is titled "My workspace" and shows a list of items under "Content". The list includes:

Name	Type
00 Static	Report
00 Static	Dataset
01 Dynamic	Report
01 Dynamic	Dataset
20210608	
20210608	
20210608	
20221115	
Agenda	
Agenda	
AS Composite model	

A context menu is open over the "01 Dynamic" dataset, with options: Analyze in Excel, Create report, Auto-create report, Create paginated report, Delete, Quick insights, Security (which is highlighted with a red box), Rename, Settings, and Download this file. The "Dataset" option in the menu is also highlighted with a yellow box.

**Figure 43-24.** Security setting for the published dataset

In the Security tab, add all the users to the Sales Rep role (see Figure 43-25).

The screenshot shows the 'Row-Level Security' settings in Power BI. On the left, there's a sidebar with icons for Home, Create, Browse, Data hub, Metrics, and a grid icon. The main area has a title 'Row-Level Security' and a section for 'user (1)'. Below this, there's a 'Members (1)' section with a sub-section for 'People or groups who belong to this role'. A text input field says 'Enter email addresses', and below it, the name 'Reza Rad' is listed with a delete 'X' icon. The overall interface is clean with a light gray background and blue headers.

**Figure 43-25.** Adding all the users to the Sales Rep role

Adding a user here doesn't mean that they will see data in the report. Remember that this security is dynamic, which means that they will see their data rows only if the underlying dataset has a record for their username, and they will only see data rows related to their username, not others.

If you refresh the report in Power BI, you will see actual usernames, as shown in Figure 43-26.

A screenshot of a dropdown menu with the word 'User' and a small arrow icon. Below it, the email address 'reza@radacad.com' is listed. The background is white, and the text is black.

**Figure 43-26.** Actual usernames displayed after refreshing the report

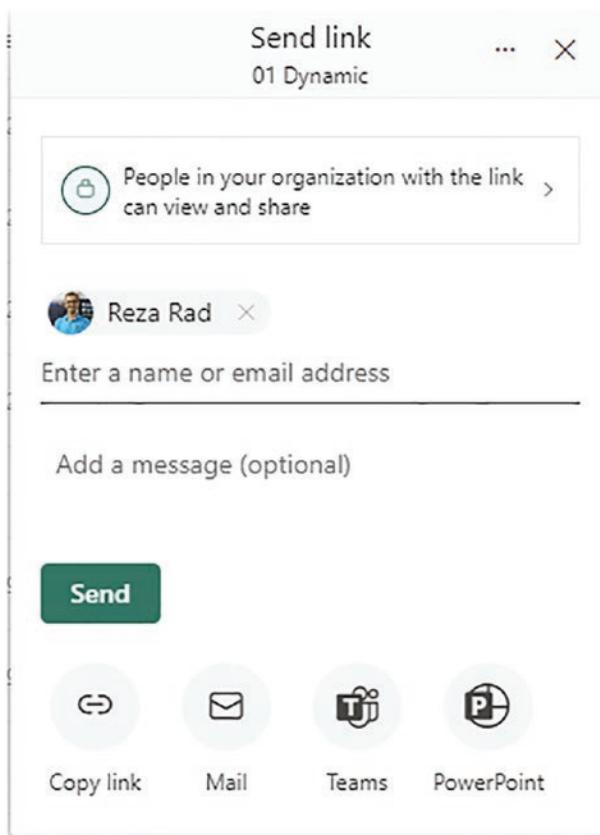
## Sharing the Report or Dashboard

Other users should have access to the dashboard and report to see them. You can share the report using any method that gives the user a read-only view. In Figure 43-27, I used the individual report-sharing option.

The screenshot shows the Power BI 'My workspace' interface. On the left is a vertical navigation bar with icons for Home, Create, Browse, Data hub, Metrics, Apps, and Deployment pipelines. The main area is titled 'My workspace' and contains a search bar with '+ New' and 'Upload' dropdowns. Below is a filter bar with 'All' (selected), 'Content', and 'Datasets + dataflows'. The main content area lists reports: '00 Static' (blue icon), '00 Static' (orange icon), '01 Dynamic' (blue icon), and '01 Dynamic' (orange icon). To the right of each report are three buttons: a grey square with a white icon, a star, and three dots. A 'Share' button is highlighted with a red box.

**Figure 43-27.** Granting users access using the individual report-sharing option

Now you can share it with other users (see Figure 43-28).



**Figure 43-28.** Sharing the report with other users

## Testing the Security

If other users open the report, and if their usernames match one of the entries in the Sales Rep table, they will see their names and data rows in the report (see Figure 43-29).

The screenshot shows a Power BI report interface. At the top, there's a navigation bar with 'Power BI', 'My workspace', and other options like 'Export', 'Share', and 'Get insights'. Below the navigation, a message says 'Data updated 7/2/20'.

The main area contains a table with the following data:

Name	Email	Year	Month	Day	Sales	Amount
Reza Rad	reza@radacad.com	2017	January	1		\$100
Reza Rad	reza@radacad.com	2017	March	1		\$50
<b>Total</b>						<b>\$150</b>

On the right side of the report, there's a sidebar titled 'Filters' with a search bar. It displays the message 'There aren't any filters to display.'

Below the table, there's a user profile section for 'reza@radacad.com' labeled 'User'.

**Figure 43-29.** Users granted access can view data rows related to them in the report

As you can see, my account only sees my transactions in the Sales Rep and Sales Transactions tables. Other users will have a different view.

You have seen how easy it is to use dynamic row-level security in Power BI using the DAX `USERNAME()` or `UserPrincipalName()` functions. Users can see their view of the world. However, you must ensure that your Power BI model's relationship is set up properly. Otherwise, people might see other table data when there is no relationship between their profile table and those tables. Dynamic row-level security is highly dependent on your data model, so keep your data model right.

To set up dynamic row-level security, you must set up your data model properly. Here are some details about that: [radacad.com/what-do-you-need-to-implement-dynamic-row-level-security-in-power-bi](http://radacad.com/what-do-you-need-to-implement-dynamic-row-level-security-in-power-bi).

### Dynamic Row-Level Security Patterns

Dynamic row-level security can get complicated because the way that tables are related makes a big difference in how the filter propagates through the model. The following are examples of dynamic RLS patterns:

- Dynamic Row-level security with manager-level access in Power BI ([radacad.com/dynamic-row-level-security-with-manager-level-access-in-power-bi](http://radacad.com/dynamic-row-level-security-with-manager-level-access-in-power-bi))
- Dynamic row-level security with profiles and users in Power BI : Many-to-many relationship ([radacad.com/dynamic-row-level-security-with-profiles-and-users-in-power-bi](http://radacad.com/dynamic-row-level-security-with-profiles-and-users-in-power-bi))
- Dynamic row-level security with organizational hierarchy Power BI ([radacad.com/dynamic-row-level-security-with-organizational-hierarchy-power-bi](http://radacad.com/dynamic-row-level-security-with-organizational-hierarchy-power-bi))
- Dynamic row-level security in Power BI with organizational hierarchy and multiple positions in many-to-many relationship: Part 1 ([radacad.com/dynamic-row-level-security-in-power-bi-with-organizational-hierarchy-and-multiple-positions-in-many-to-many-relationship-part-1](http://radacad.com/dynamic-row-level-security-in-power-bi-with-organizational-hierarchy-and-multiple-positions-in-many-to-many-relationship-part-1))

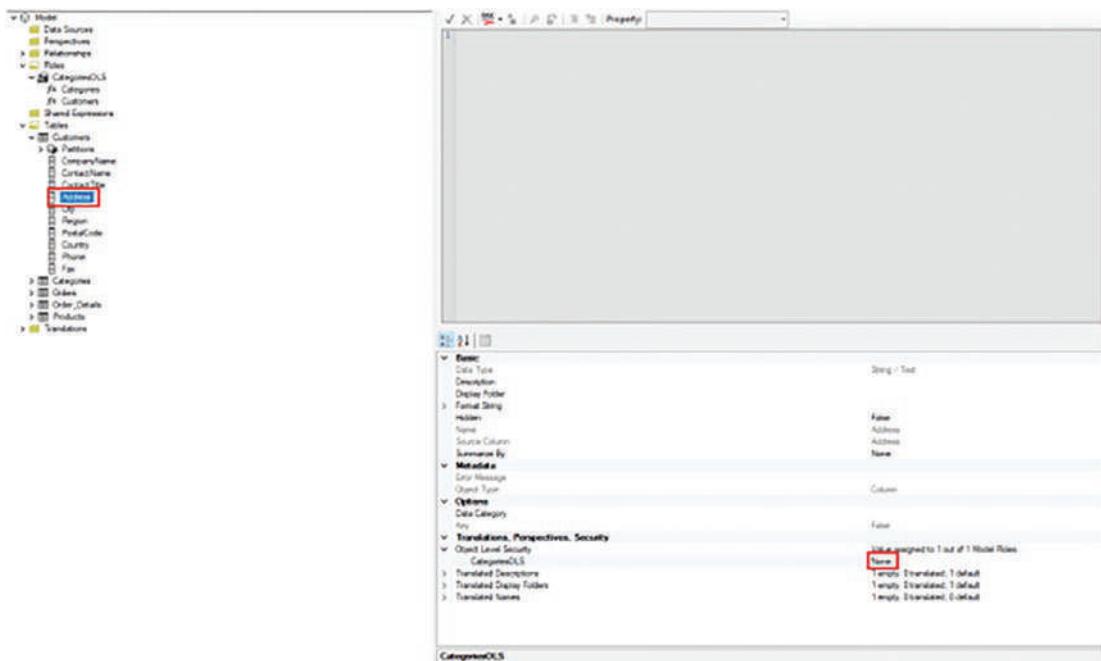
- Dynamic row-level security in Power BI with organizational hierarchy and multiple positions in many-to-many relationship: Part 2 ([radacad.com/dynamic-row-level-security-in-power-bi-with-organizational-hierarchy-and-multiple-positions-in-many-to-many-relationship-part-2](http://radacad.com/dynamic-row-level-security-in-power-bi-with-organizational-hierarchy-and-multiple-positions-in-many-to-many-relationship-part-2))

For more information about row-level security, I suggest reading my book at [www.amazon.com/Row-Level-Security-Power-BI-different/dp/1651119287](http://www.amazon.com/Row-Level-Security-Power-BI-different/dp/1651119287).

## Column-Level Security

What if you wanted to control access to a whole column? Say that the Sales team cannot see the Profit column but can see the Sales column in the same table. This is called column-level security.

Although column-level security is supported in Power BI, the implementation of that is not yet supported in the Power BI Desktop. This means it has to be done using another tool, called Tabular Editor, Visual Studio, or any other tool that can connect to the Power BI dataset and apply changes to it. Figure 43-30 shows an example of implementing it using the Tabular Editor.



**Figure 43-30.** Column-level security

Source: [learn.microsoft.com/en-us/power-bi/enterprise/service-admin-ols](https://learn.microsoft.com/en-us/power-bi/enterprise/service-admin-ols)

## Object-Level Security

What if you wanted access to a whole table to be controlled? Consider some users seeing the Sales table and others not. This is called object-level security, and similar to column-level security, it has to be implemented using another tool.

The screenshot shows the 'Basic' tab of a dataset's properties. Under 'Table Permissions', there is a table with columns 'Category' and 'Role'. The 'Category' column lists various tables like 'Categories', 'Customers', and several date tables. The 'Role' column for all these tables is set to 'None', which is highlighted with a red box. A tooltip above the table says 'RLS enabled on 0 out of 9 tables' and 'OLS enabled on 1 out of 9 tables'.

Category	Role
Categories	None
Customers	Default
DateTableTemplate_8340ca1c-13cc-417d-881d-176281d83ed3	Default
Local Date Table_7ef183e6-2cb9-4290-aa60-606d185c654f	Default
Local Date Table_cde564de-d289-40fd-b18f-100e83821e85	Default
Local Date Table_d5647606-e75a-42be-a141-5b5dc67aebedf	Default
Order_Details	Default
Orders	Default
Products	Default

Below the table, under 'Translations, Perspectives, Security', it shows '0 model role members' and 'Read' permission.

**Figure 43-31.** Object-level security

Source: [learn.microsoft.com/en-us/power-bi/enterprise/service-admin-ols](https://learn.microsoft.com/en-us/power-bi/enterprise/service-admin-ols)

## Page-Level Security

RLS, CLS, and OLS are set at the dataset level. This means all the reports connected live to the shared dataset will follow the security rules and setup. Sometimes, you may need security at the visual level, though. Visual-level security means that some users see some visuals or pages, and others see others. At the time of writing this chapter, visual-level security is not supported in Power BI. However, there are a few workarounds you can use, which are explained in the following articles:

- Share different visual pages with different security groups in Power BI ([radacad.com/share-different-visual-pages-with-different-security-groups-in-power-bi](https://radacad.com/share-different-visual-pages-with-different-security-groups-in-power-bi))
- Page-level security workaround in Power BI ([radacad.com/page-level-security-workaround-in-power-bi](https://radacad.com/page-level-security-workaround-in-power-bi))

## Summary

There are different levels of data-related security in Power BI. Row-level security involves filtering rows of data for some users. Column-level security and object-level security are about controlling access to particular columns or tables, respectively. When data-level security is applied to a dataset, all the reports connected to it will follow its security rules. Row-level security can be static or dynamic. Dynamic row-level security is the preferred method for most situations because it is easier to maintain. However, it can come with some complexities of design. This chapter explained data-level security and row-level security.

## CHAPTER 44



# Power BI Helper

Microsoft has developed built-in tools for Power BI development—the Power BI Desktop, the Power BI Report Builder, the Power BI Gateway, and the Power BI Service for hosting the reports. Although these tools help you develop a Power BI solution, there are certain aspects of the process that can be further enhanced by using third-party tools. The Power BI community has a wide range of tools that can help with different aspects of Power BI. At RADACAD, we developed a free tool named Power BI Helper. This tool assists with many aspects of Power BI implementation, from the report's development to maintenance and performance tuning, to administration, documentation, and file cleanup. Knowing how to use this tool from the Power BI architecture perspective is helpful.

## Documenting a Power BI File and Report

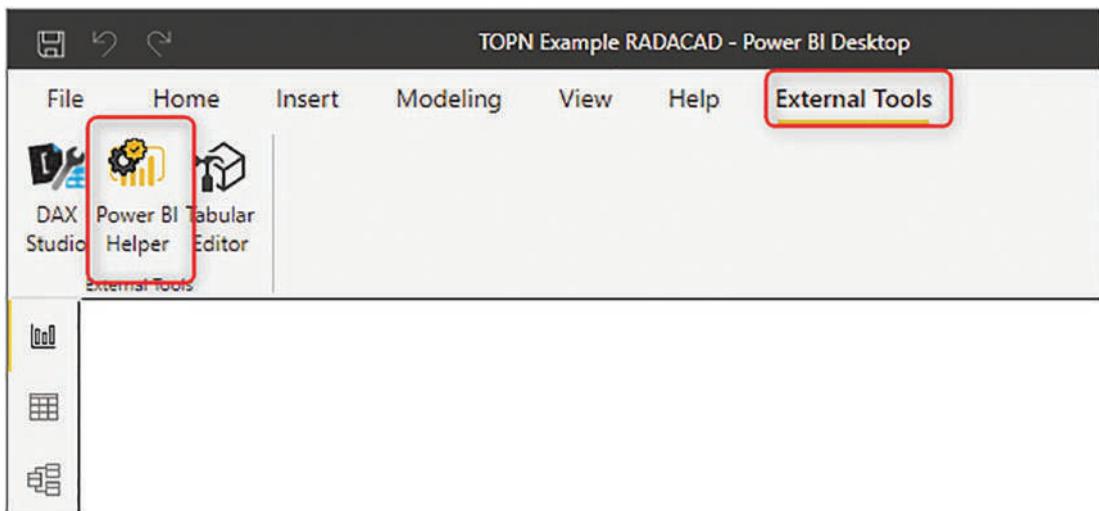
This section explains how to document everything in a Power BI file in a few clicks. The document output will contain all the DAX code (measures, columns, and tables) with the expressions, the tables with the Power Query scripts, information about visualization such as how many pages, bookmarks, and visuals are in each page, and even which tables and columns are used in visualization. All in just a few clicks.

## Download and Install Power BI Helper for Free

If you haven't done so already, download and install Power BI Helper. It is a free application and it can help you do many useful things related to the development of the Power BI reports as well as the documentation process. You can download it from [powerbihelper.org/](http://powerbihelper.org/).

## Open Power BI Helper as an External Tool

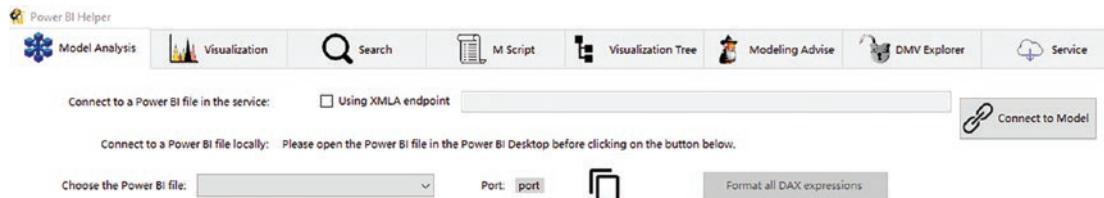
After the install, while you have the Power BI file open in the Power BI Desktop, go to the External Tools tab and click Power BI Helper to open it, as shown in Figure 44-1. (Alternatively you can open it from Start and then from the programs on your Windows machine.)



**Figure 44-1.** Opening Power BI Helper from the Power BI Desktop

## Connect to Model

Once Power BI Helper is open, click Connect to Model, as shown in Figure 44-2. (Make sure that the Power BI file is already open in the Power BI Desktop.)



**Figure 44-2.** Connecting to the model from Power BI Helper

This option will detect all instances of Power BI files open in your desktop and will list them in a drop-down, as shown in Figure 44-3. You can choose the one you want to document.

The screenshot shows the Power BI Helper application window. At the top, there's a navigation bar with links like Model Analysis, Visualization, Search, M Script, Visualization Tree, Modeling Advisor, Data Explorer, Service, Compare, and Documentation. Below the navigation bar, there's a message: "Connect to a Power BI file in the service" with an "Using XMLA endpoint" checkbox, and a "Connect to Model" button.

Below this, a dropdown menu says "Choose the Power BI file: T0PN Example RADACAD" and a "Port: 2185" input field. There are also "Format all DAX expressions" and "Filter" buttons.

The main area is divided into several sections:

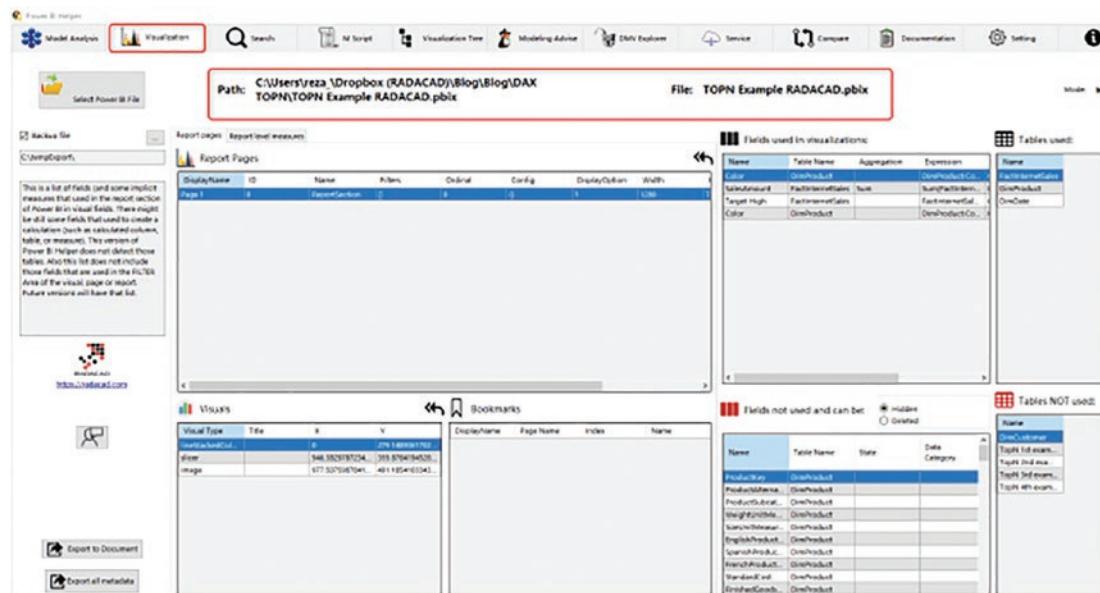
- Tables:** A grid showing tables like DimProduct, DateTableTemplate\_4930f756-500c..., LocalDateTable\_f6a5ad2-2f5c-4ec..., FactInternetSales, DimCustomer, LocalDateTable\_1e93d309-23d8-4a..., and LocalDateTable\_f6db4fe-2f1d-4db... with columns for Name, Description, Storage Mode, Source, and N.
- Columns:** A grid showing columns like FactCustomer-35d297bb-1735-487b..., FactProduct-Top100-exam..., and Order Quantity with columns for Name, Table Name, State, Data Category, and Data Type.
- Measures:** A grid showing measures like Target High (average sales of top c...) and Sales Color based on top group with columns for Name, State, and Data Category.
- fx Expression:** A section showing a complex DAX expression involving TOPN, SUMMARIZE, DimCustomer, DimCustomer[FullName], Order Quantity, and sum(FactInternetSales[OrderQuantity]) - [Order Quantity].
- Dependency List:** Two tables showing dependencies between objects. The first table lists dependencies for the table itself, and the second lists dependencies for the columns.
- fx Measure Expression:** A section showing the measure expression for the Target High measure, which includes variables like topGroupValue, selectedColorSale, and SalesTarget.
- Reverse Dependency Tree:** A tree diagram showing dependencies for the Back Color measure, including FactCustomer, FactProduct, SalesAmount, and DimCustomer.
- Measure Dependency Tree:** A tree diagram showing dependencies for the Target High measure, including FactCustomer, FactProduct, SalesAmount, and DimCustomer.

**Figure 44-3.** Selecting the Power BI file in Power BI Helper

If you have only one Power BI file open, the drop-down will show just that one. The rest of the page shows information about the model. It displays information such as the tables, columns, and measures with their expressions and expression trees.

## Visualization Information

Power BI Helper should automatically find all the visualization information of the selected model. It shows you this information in the Visualization tab (see Figure 44-4).

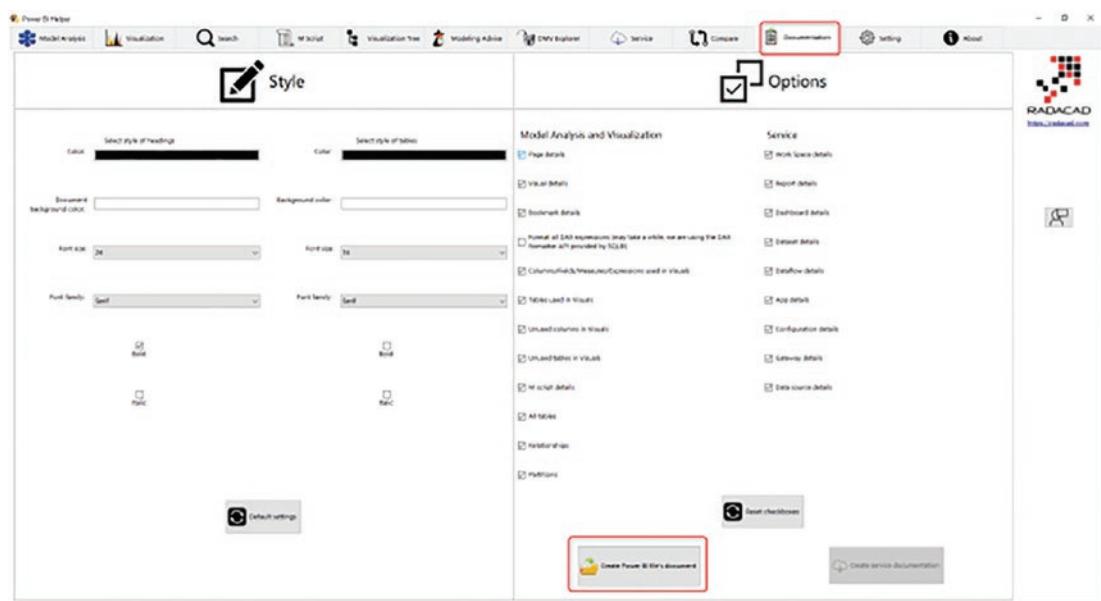


**Figure 44-4.** Power BI Helper automatically shows the visualization information after selecting the model

There are a few cases where Power BI Helper doesn't show the visualization information automatically (for example, if the Power BI file was just generated and has not yet been saved). In those cases, you can use the Select Power BI File option to choose the file.

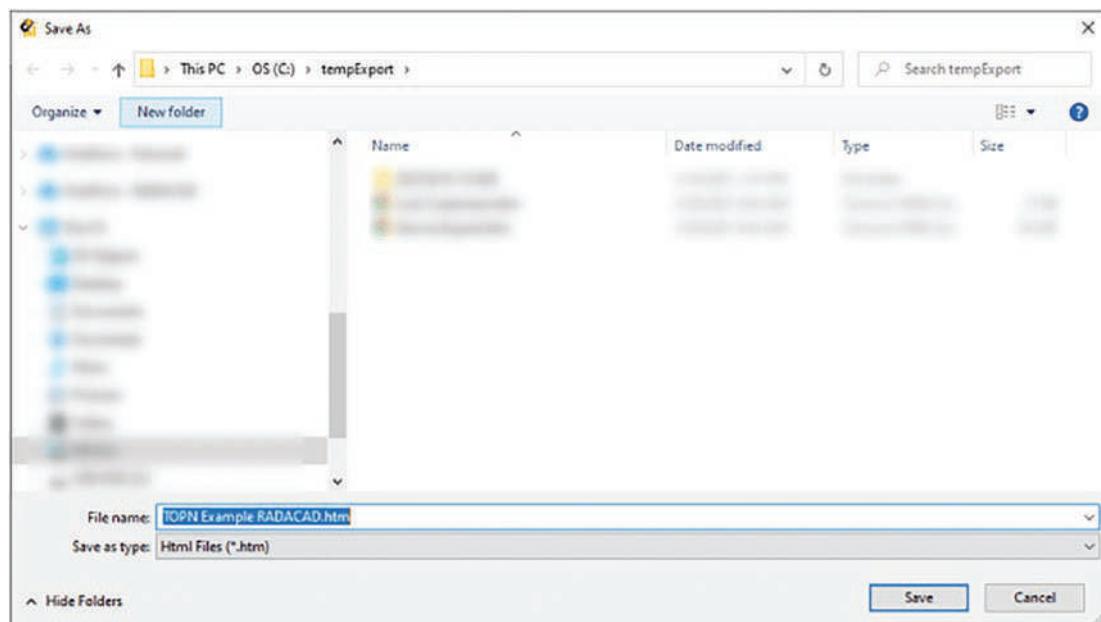
## Documentation

After that, you are just a click away from the documentation (this is in fact the second click after opening Power BI Helper). In the Documentation tab, click the Create Power BI document, as demonstrated in Figure 44-5.



**Figure 44-5.** Generating the Power BI file

Choose the location for the file (which is of HTML type), as shown in Figure 44-6.



**Figure 44-6.** Saving the Power BI file documentation

And that's it, you can now open the documentation.

## What Is Included in the Documentation?

The documentation includes a large amount of information. Here is just some of the information available in the output document:

- Filename, path, and date of documentation
- All visualization pages and their details
- All visuals on every page with their details
- All the bookmarks and the pages the bookmarks are bound to
- All the columns and measures used in Power BI visuals
- All the tables used in Power BI visuals
- All the columns, tables, and measures not used in Power BI visuals (good source to clean up, although Power BI Helper can do that for you in a few clicks)
- All the tables in the model with their details. If they are calculated tables, their DAX expressions will be there too
- All the measures with their DAX expressions, and two tables of dependency tree and reverse dependency tree (useful for detecting where the measure is used, or what other fields/tables/measures used in the expression of this measure)
- All the relationships and their details
- All the roles (defined for row-level security) and their expressions
- Report-level measures if there are any

## Configurations

As you have seen, creating the documentation is easy. If you want to configure the output document, you can do that from the Documentation tab, in two main sections—Style and Options.

### Configure the Style of the Documentation

The Style configuration (see Figure 44-7) determines the fonts, colors, size of tables, headings, and other parts of the generated document.

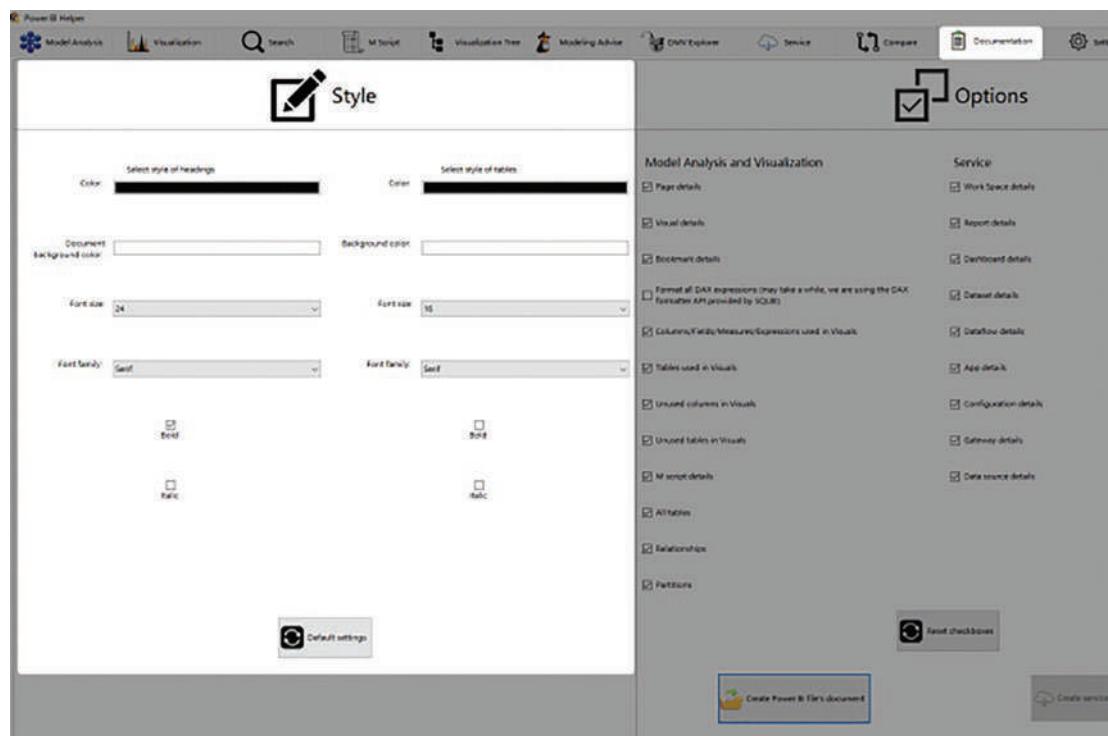
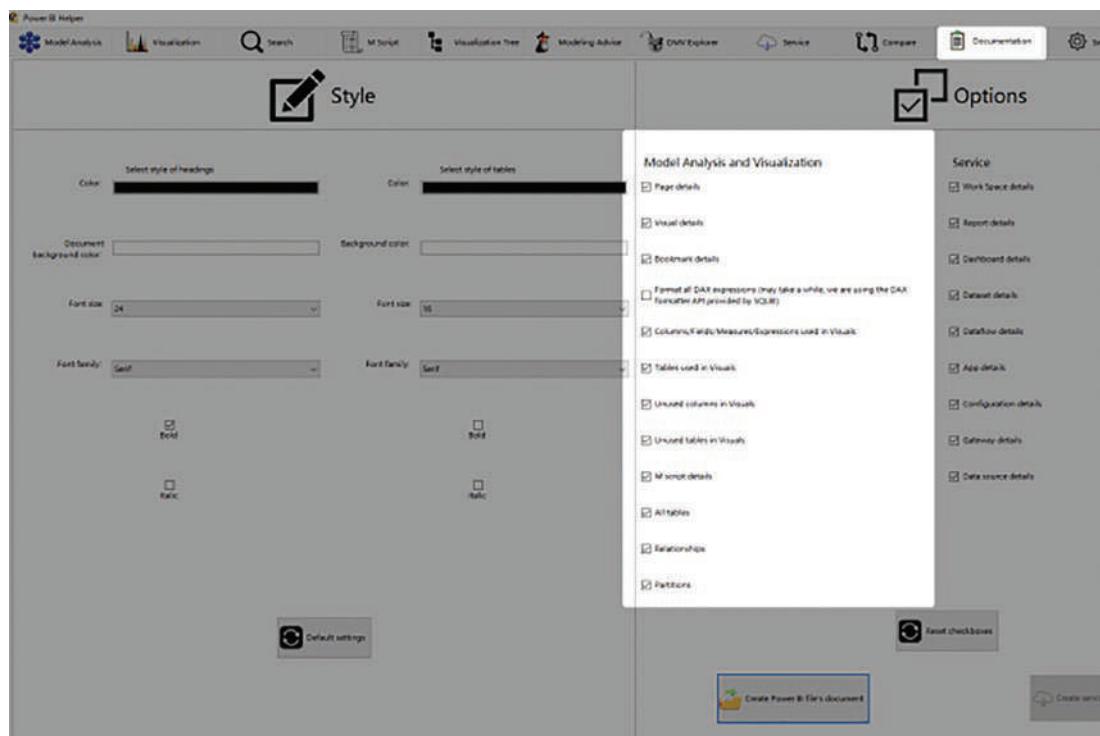


Figure 44-7. Configuring the style of the documentation

## Sections and Information to Document

You can choose what information you want to document from the Power BI files. I recommend keeping them all selected for the full documentation, as shown in Figure 44-8.



**Figure 44-8.** Choosing the information to document from a Power BI file

You can format all the DAX expressions before documentation, and this option uses the beautiful DAX formatter service from our friends at SQLBI (see [www.sqlbi.com](http://www.sqlbi.com)). If you use it, remember that every measure's expression will be sent to an API and the result of that will come back, so if you have too many DAX expressions, this will slow down the documentation process.

## Exporting the Data in a Power BI Table

If you want to export the data in a Power BI table, you have options such as copy and paste, or you can put it in a visual and export it. However, there is an easier way. You can use Power BI Helper to export the table in a few clicks. The following sections explain how it is possible. First, open Power BI Helper from the Power BI Desktop.

### Connect to Model

Once Power BI Helper is open, click Connect to Model. (Make sure that the Power BI file is already open in the Power BI Desktop.)

This option will detect all the Power BI files open on the desktop and will list them in a drop-down. You can choose the one you want to document (see Figure 44-9).

The screenshot shows the Power BI Helper application window. At the top, there's a navigation bar with links like Model Analysis, Visualization, Search, M Script, Visualization Tree, Modeling Advisor, Data Explorer, Service, Compare, and Documentation. Below the navigation bar, there's a message: "Connect to a Power BI file in the service" and "Using XMLA endpoint". A "Connect to Model" button is also present.

In the center, there's a dropdown menu "Choose the Power BI File: T0PN Example RADACAD" and a "Port: 2185" input field. To the right of these are buttons for "Format all DAX expressions" and "Import the data".

The main area is divided into several sections:

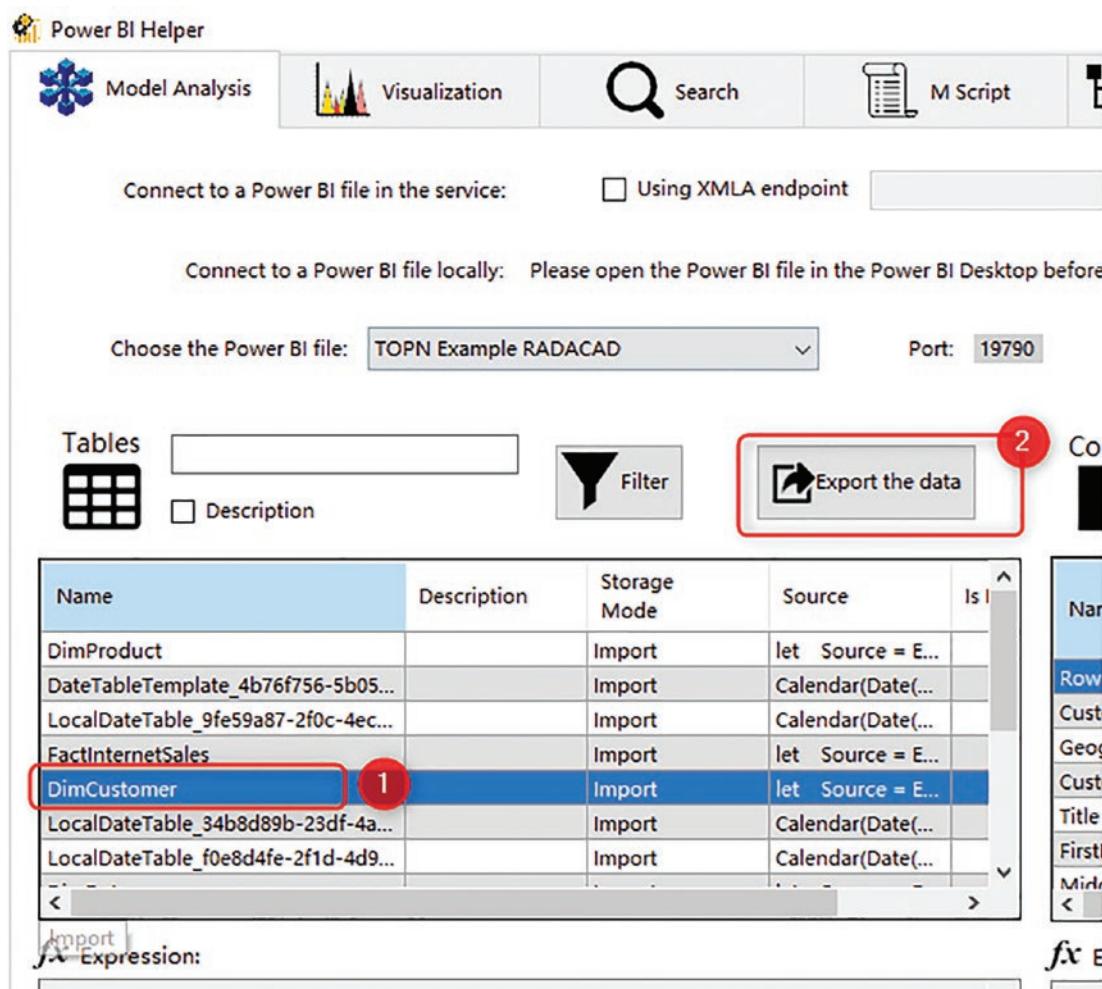
- Tables:** A list of tables with columns for Name, Description, Storage Mode, Source, and N.
- Columns:** A list of columns with columns for Name, Table Name, State, Data Category, and Data Type.
- Measures:** A list of measures with columns for Name, State, and Data Category.
- fx Expression:** A section showing a complex DAX expression for a measure named "TOPN( 10, SUMMARIZE( DimCustomer, DimCustomer[FullName], 'Order Quantity', sum(FactInternetSales[OrderQuantity]) ), [Order Quantity] )".
- fx Expression:** Another section showing a DAX expression for a measure named "TargetHigh(average sales of top 10%)".
- fx Measure Expression:** A section showing a DAX measure expression for "Target High (average sales of top 10%)".
- Dependency List:** Two sections titled "Dependency List: (In which measures or columns this table is used)" and "Dependency List: (In which measures or columns this column is used)" both show empty tables.
- Reverse Dependency Tree:** A section titled "Reverse Dependency Tree: (In which measures this measure is used)" showing a tree structure for "Measure: Back Color based on top group".
- Measure:** A section titled "Measure: Back Color based on top group" showing a tree structure for "Measure: Back Color based on top group (Table: FactInternetSales)" and "Column: SalesAmount (Table: FactInternetSales)".
- Measure:** A section titled "Measure: Target High (average sales of top 10%) (Table: DimProduct)" showing a tree structure for "Measure: Target High (average sales of top 10%) (Table: DimProduct)" and "Table: FactInternetSales".

**Figure 44-9.** Selecting the Power BI file in Power BI Helper

If you have only one Power BI file open, the drop-down will show just that one. The rest of the page shows information about the model. It lists information such as the tables, columns, and measures with their expressions and expression trees.

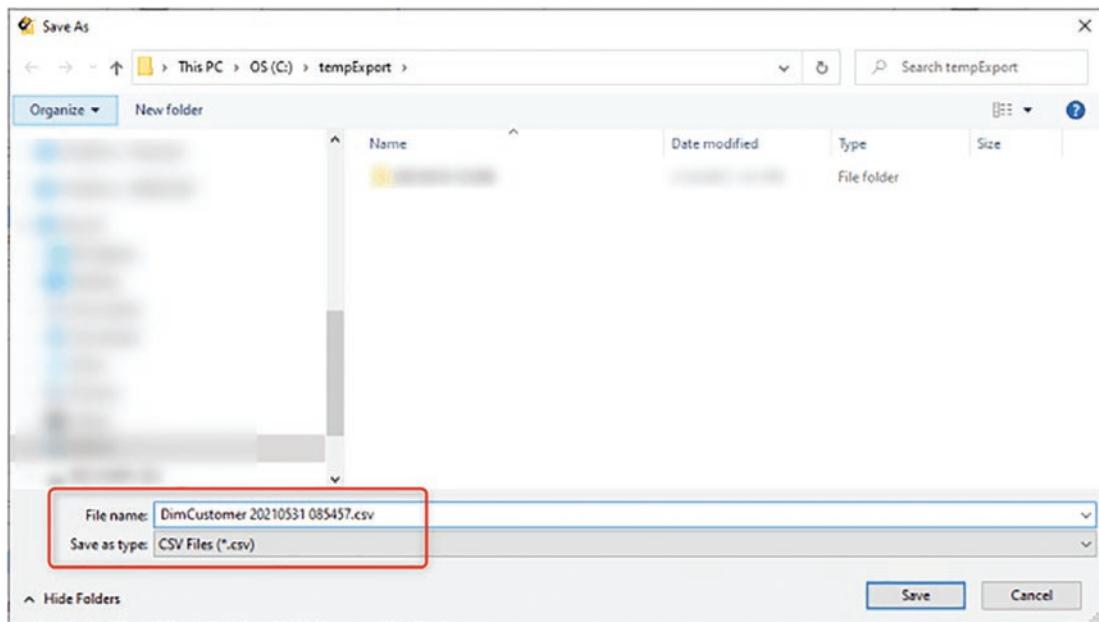
## Export the Data

In the same Model Analysis tab, select any table you want from the list of tables, and then click Export the Data, as shown in Figure 44-10.



**Figure 44-10.** Exporting data from a Power BI table

This will export the data as a CSV file to the location you choose, as illustrated in Figure 44-11.



**Figure 44-11.** Exporting table data as a CSV file

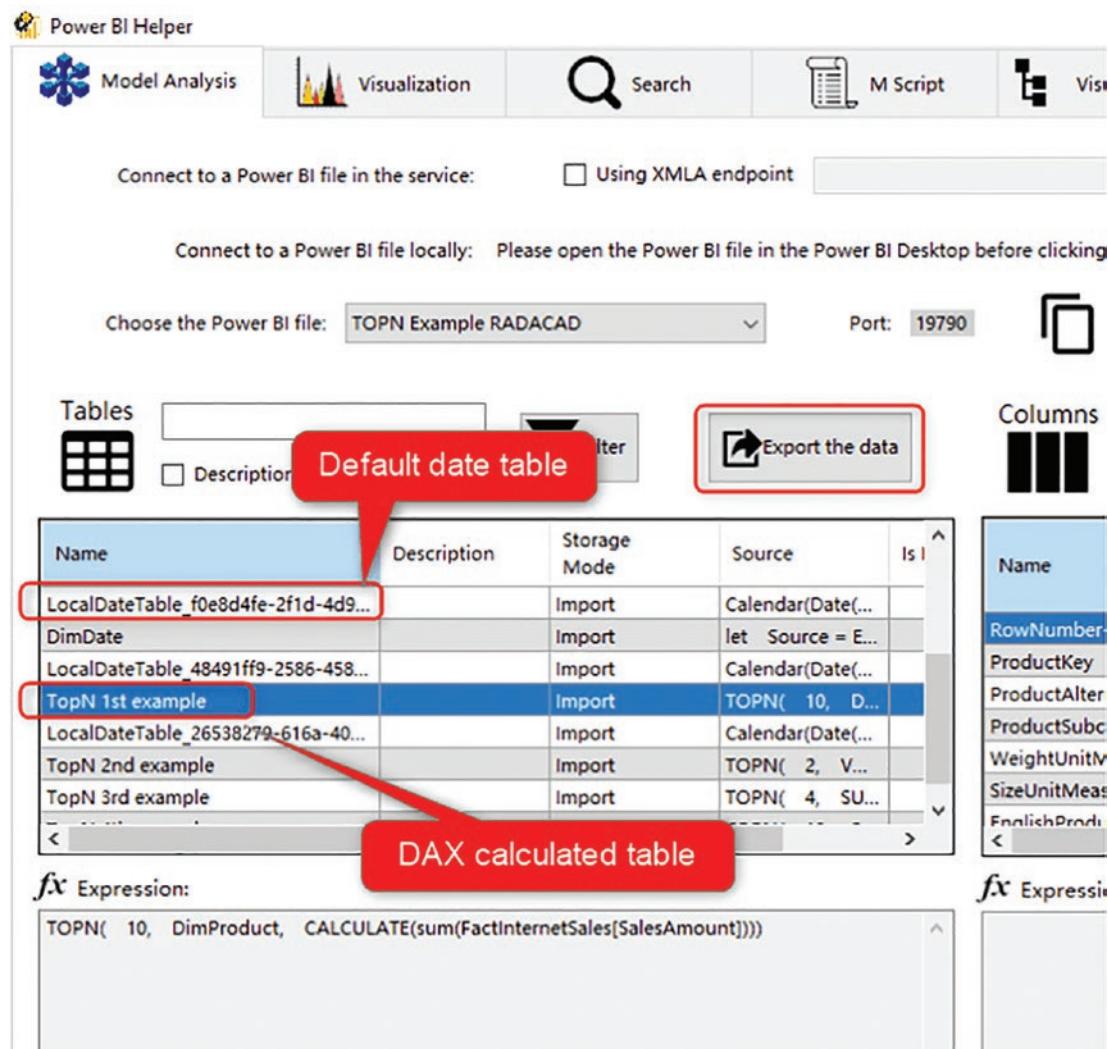
Figure 44-12 shows example CSV output.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	DimCusto	DimCusto	DimCusto	DimCusto	DimCusto	DimCusto	DimCusto	DimCustomer	DimCustomer[BirthDate]	DimCusto	DimCusto	DimCusto	DimCusto	DimCusto	DimCusto	DimCusto	DimCusto	DimCusto	DimCusto	
2	11471	207	AW00011471	Latasha	Suarez	FALSE	9/25/1973	0:00	5	F	latasha12@e	30000	0	0	Partial Co Estudios L Baco					
3	11602	115	AW00011602	Larry	Gill	FALSE	4/18/1970	0:00	5	M	larry16@e	30000	0	0	Partial Co Estudios L Baco					
4	11603	244	AW00011603	Geoffrey	Gonzalez	FALSE	2/6/1970	0:00	5	M	geoffrey11	30000	0	0	Partial Co Estudios L Baco					
5	11604	275	AW00011604	Edgar	Sanchez	FALSE	6/1/1970	0:00	5	M	edgar21@e	30000	0	0	Partial Co Estudios L Baco					
6	11610	269	AW00011610	Blake	Collins	FALSE	4/23/1970	0:00	5	M	blaked7@e	30000	0	0	Partial Co Estudios L Baco					
7	12515	189	AW00012515	Shelby	Barley	FALSE	6/1/1970	0:00	5	F	shelby16@e	30000	0	0	Partial Co Estudios L Baco					
8	12517	133	AW00012517	Alexa	Watson	FALSE	8/25/1970	0:00	5	F	alexad9@e	30000	0	0	Partial Co Estudios L Baco					
9	12518	161	AW00012518	Jacquelyn	Domingue	FALSE	9/27/1970	0:00	5	F	jacquelyn	30000	0	0	Partial Co Estudios L Baco					
10	12519	265	AW00012519	Casey	Gutierrez	FALSE	12/17/1970	0:00	5	M	casey14@e	30000	0	0	Partial Co Estudios L Baco					
11	12524	211	AW00012524	Kate	Shan	FALSE	1/24/1970	0:00	5	F	kate8@ad	30000	0	0	Partial Co Estudios L Baco					
12	12714	157	AW00012714	Colleen	Lu	FALSE	7/17/1970	0:00	5	F	colleen11	30000	0	0	Partial Co Estudios L Baco					
13	12716	185	AW00012716	Dale	Shen	FALSE	3/16/1970	0:00	5	M	dale2@ad	30000	0	0	Partial Co Estudios L Baco					
14	12718	193	AW00012718	Tammy	Sai	FALSE	11/14/1970	0:00	5	F	tammy6@e	30000	0	0	Partial Co Estudios L Baco					
15	12228	131	AW00012728	Jeremiah	Stewart	FALSE	6/26/1970	0:00	5	M	jeremiah4	30000	0	0	Partial Co Estudios L Baco					
16	12871	213	AW00012871	Leah	Li	FALSE	10/6/1970	0:00	5	F	leah2@ad	30000	0	0	Partial Co Estudios L Baco					
17	13671	173	AW00013671	Frank	Ramos	FALSE	2/7/1974	0:00	5	M	frank25@e	30000	0	0	Partial Co Estudios L Baco					
18	13825	127	AW00013825	Candice	He	FALSE	11/25/1970	0:00	5	F	candice2@e	30000	0	0	Partial Co Estudios L Baco					
19	13830	237	AW00013830	Andrea	Cor	FALSE	8/3/1970	0:00	5	F	andrea11	30000	0	0	Partial Co Estudios L Baco					
20	13837	120	AW00013837	Alyssa	Lee	FALSE	8/13/1970	0:00	5	F	alyssaz22@e	30000	0	0	Partial Co Estudios L Baco					
21	13838	263	AW00013838	Jill	Rubio	FALSE	6/27/1970	0:00	5	F	jill20@ad	30000	0	0	Partial Co Estudios L Baco					
22	14837	144	AW00014837	Dennis	U	FALSE	7/17/1970	0:00	5	M	dennis4@e	30000	0	0	Partial Co Estudios L Baco					
23	14838	121	AW00014838	Darren	Alvarez	FALSE	7/26/1970	0:00	5	M	darren26@e	30000	0	0	Partial Co Estudios L Baco					

**Figure 44-12.** A Power BI table exported to a CSV file

## Any Size, Any Table Type

When you use this method, you don't get any of the hassles you get with copy and paste. It's all very simple and easy to export. The size of the data in the table doesn't matter (it will take longer of course for bigger tables). Power BI Helper exports data of any size and works with DAX calculated tables, Power Query generated tables, and even automatically generated hidden tables by Power BI such as the default date tables. You can export them all. See Figure 44-13.



**Figure 44-13.** Power BI Helper exports DAX calculated tables, default tables, and hidden tables

## Consider Using Analyze in Excel

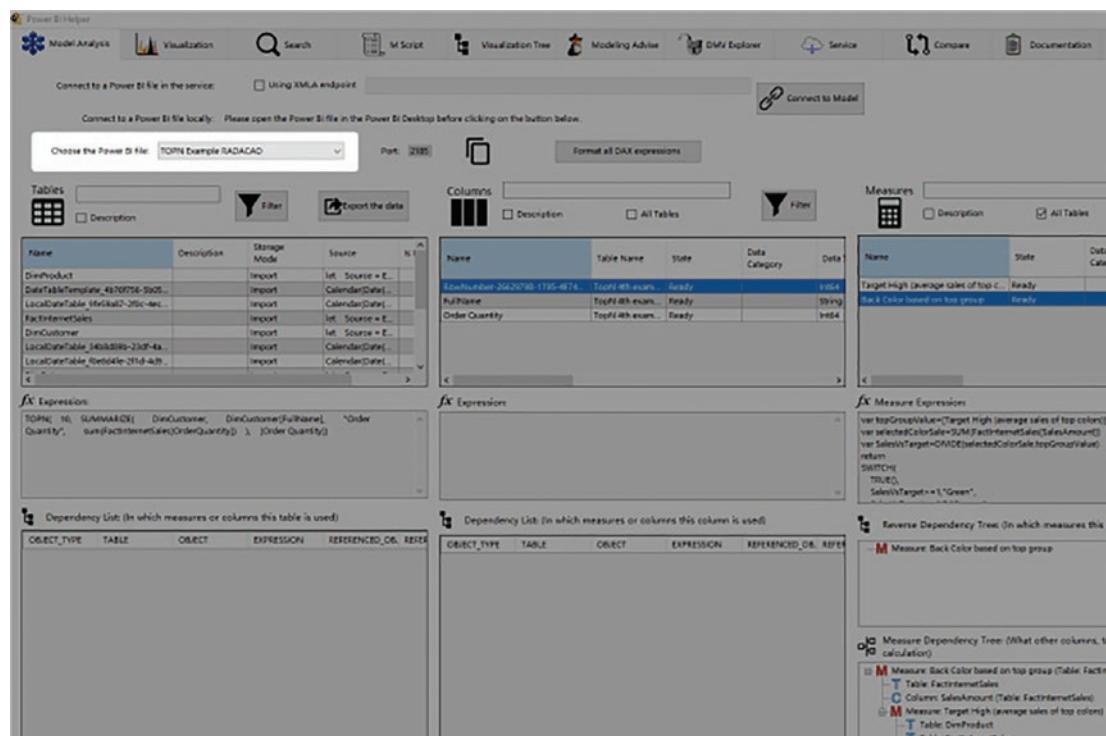
Before I leave you to use this feature, I want to mention something. If the purpose of exporting the data is to analyze it further in Excel, then I strongly suggest you use Analyze in Excel instead of Export. Read the chapter about Analyze in Excel in this book to learn more about it.

## Reducing the Size of Power BI File

One of the performance considerations of Power BI files is reducing the size as much as possible. This helps with development speed, because smaller files load faster in the Power BI Desktop. One way to reduce the size significantly is to determine which columns are consuming the most size and remove them. This section explains how you can easily do that using Power BI Helper.

Once Power BI Helper is open, click Connect to Model. (Make sure that the Power BI file is already open in the Power BI Desktop.)

This option will detect all the instances of Power BI files open in the desktop and will list them in a drop-down. Choose the one you want to review (see Figure 44-14).



**Figure 44-14.** Selecting a Power BI file in Power BI Helper

If you have only one Power BI file open, the drop-down will show just that one. The rest of the page shows information about the model. It includes information such as the tables, columns, and measures with their expressions and expression trees.

## Steps to Reduce File Size

My sample Power BI file is 72MB, as shown in Figure 44-15. I want to determine which columns are consuming the most space in this model.

Name	Date modified	Type	Size
Model.pbix	6/2/2021 8:48 AM	Microsoft.Microso...	73,230 KB

Figure 44-15. Power BI file size before tuning

When I open Power BI Helper and connect to that model, from the Modeling Advise tab, I can see how big each column is when it is in the Power BI model (see Figure 44-16).

Columns in the model and their sizes: (Sizes are in MB)		All columns in the model with their allocated size.
Dimension_name	Attribute_name	dictionary_size
FactInternetSales	SalesOrderNumber	52.96153450012207
LocalDataTable_4c984d9c-cb85-49...	Date	1.32513427734375
DimCustomer	EmailAddress	0.691594123540332
FactInternetSales	CustomerKey	0.57082366943359375
DimCustomer	AddressLine1	0.49065303302490214
DimCustomer	CustomerAlternateKey	0.46635341644287109
DimCustomer	BirthDate	0.303375244140625
DimCustomer	Phone	0.23897457122002734
LocalDataTable_4724380f-a059-4a...	Date	0.18123204345703125
LocalDataTable_595b6c11-a218-46...	Date	0.17479705810546875
DimDate	FullDateAlternateKey	0.17479705810546875
LocalDataTable_1ab20a25-e607-4b...	Date	0.0902633666992175
LocalDataTable_13145d78-77d0-4a...	Date	0.0902633666992175
LocalDataTable_7d3b4477-0899-4b...	Date	0.0828704833984375
LocalDataTable_4104540c-33c9-42...	Date	0.0328704833984375

Figure 44-16. Checking the memory space consumed for each column in a Power BI file

The dictionary\_size column is in megabytes, and it is the size of the column when expanded in memory. This might not be necessarily the size of the file when it's stored. For example, the SalesOrderNumber column doesn't use 52MB of the 72MB size of the file, but it is definitely a big part of it.

## Step 1: Remove Large Columns

Once you identify columns that are using lots of space and you know you don't need them, you can go to the file in the Power BI Desktop and remove them, as demonstrated in Figure 44-17.

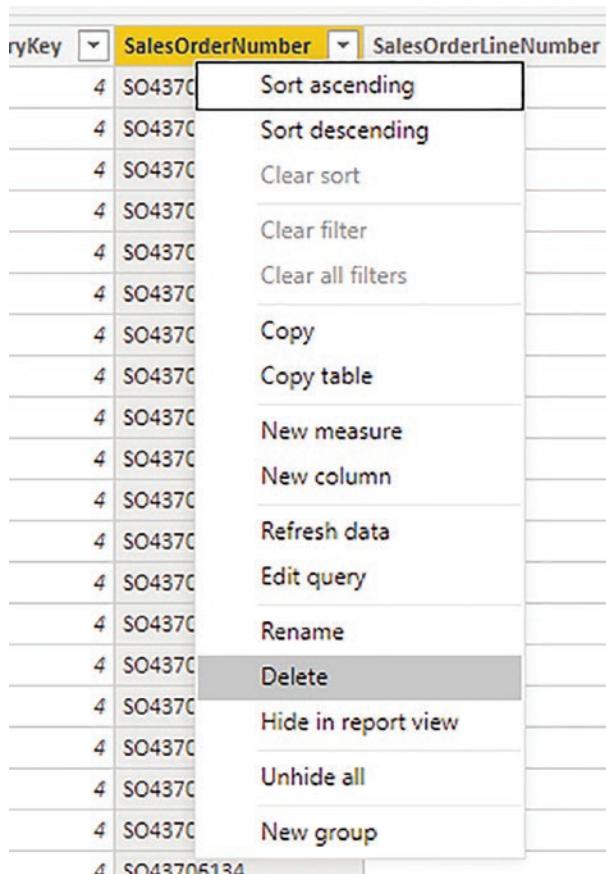
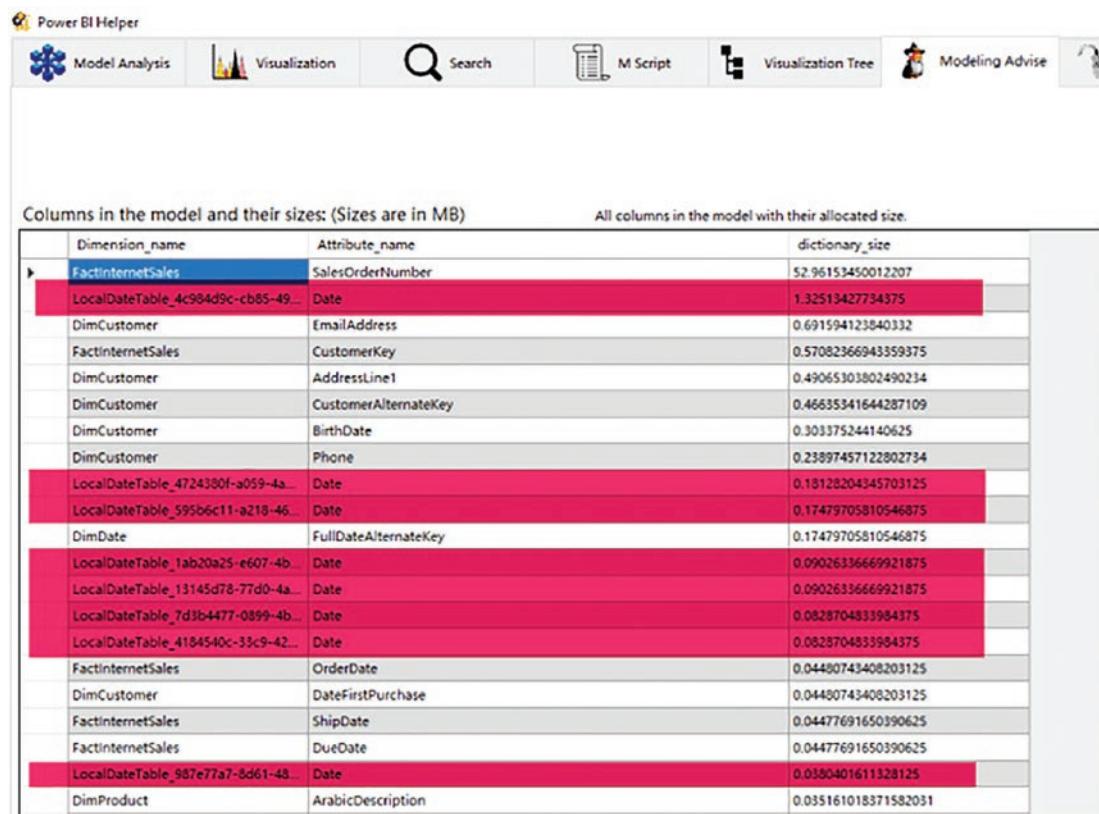


Figure 44-17. Removing columns that are consuming a lot of space and are not useful for analysis

## Step 2: Turning Off the Auto Date/Time

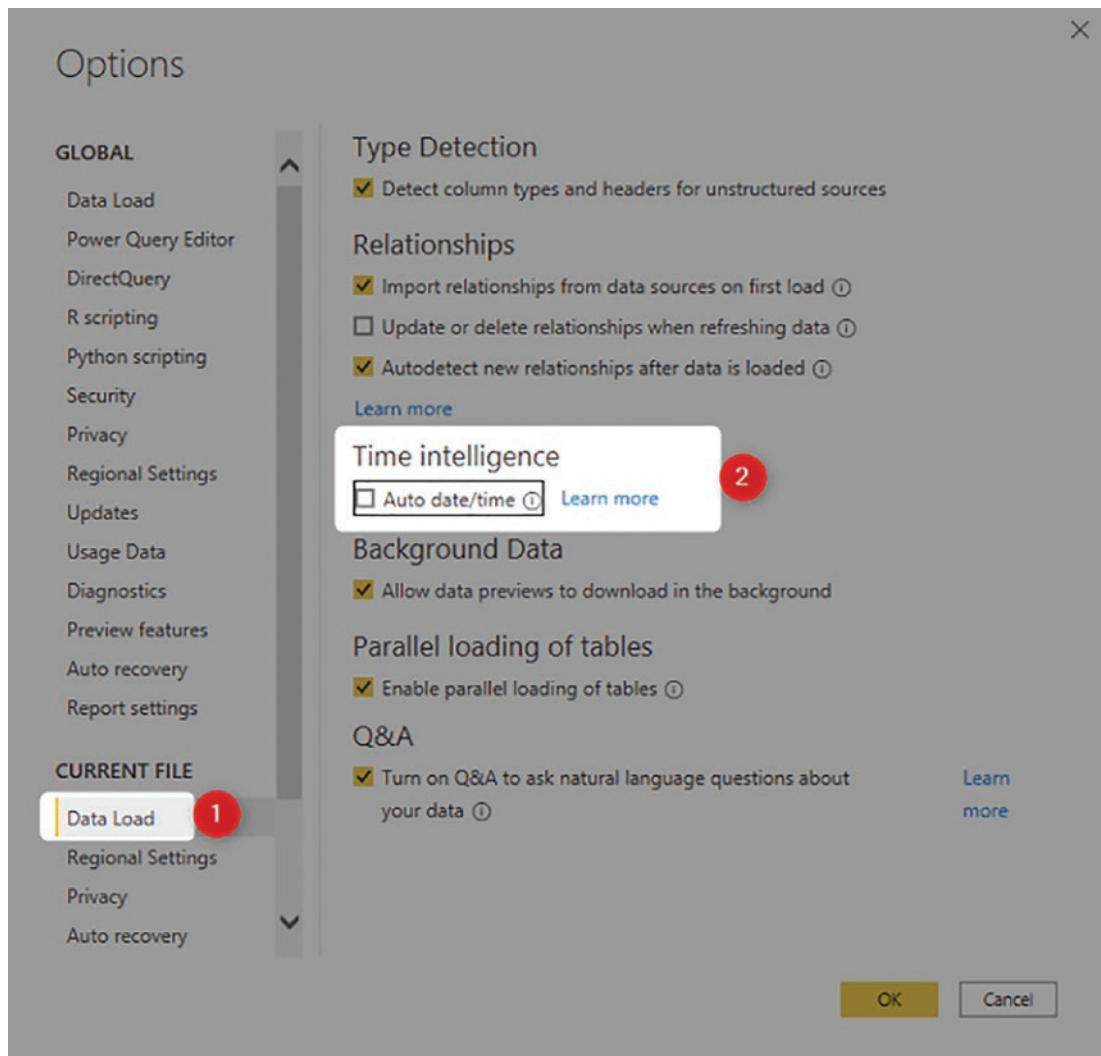
In the Modeling Advise report, you can also see evidence of having a default date table in Power BI (see Figure 44-18).



**Figure 44-18.** Using the default date table in a Power BI file

The default date table can be helpful. However, if you have too many date fields in your data model, and the range of dates is very wide, the ideal is to have a custom date table and disable the default date tables. Read my article at [radacad.com/power-bi-date-dimension-default-or-custom-is-it-confusing](http://radacad.com/power-bi-date-dimension-default-or-custom-is-it-confusing) to understand why.

You can go to the Power BI Desktop and choose File ► Options, Options. Uncheck the Auto Date/Time setting under Time Intelligence, as shown in Figure 44-19.



**Figure 44-19.** Turning off the default date table in the Power BI file

After making these changes and saving the file, the model size is reduced by 30 percent, as you can see in Figure 44-20.

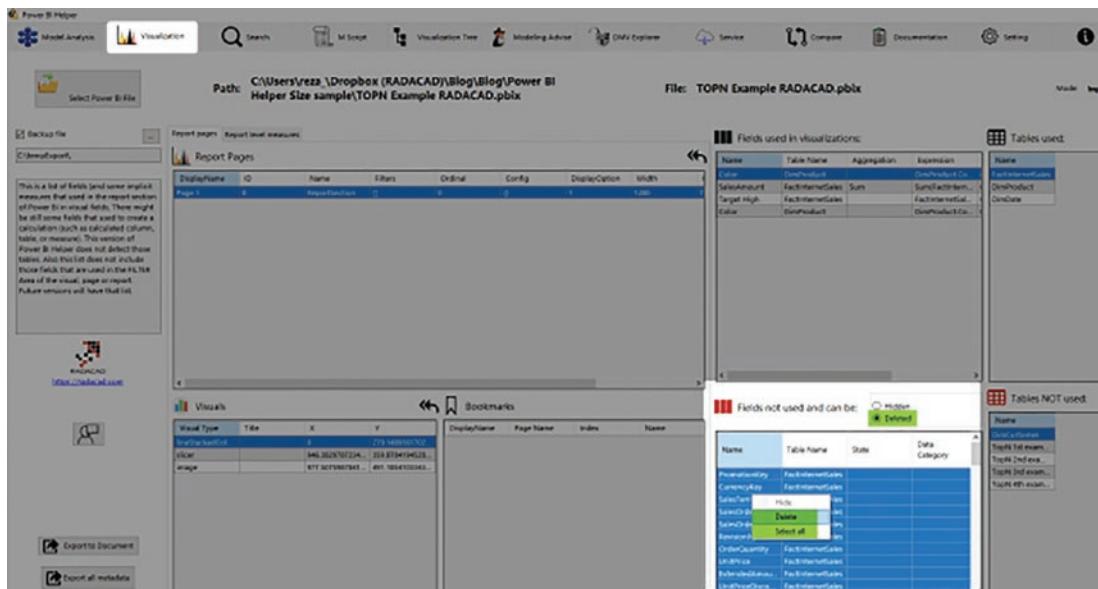
Model.pbix	6/2/2021 8:48 AM	Microsoft.Microso...	73,230 KB
Model after change.pbix	6/2/2021 9:36 AM	Microsoft.Microso...	49,873 KB

**Figure 44-20.** Power BI file size is reduced more than 30 percent after tuning

## Step 3: Removing Unused Columns

In a lot of Power BI reports, it might not be easy to determine which columns are not used in the file. You would have to consider visualizations, relationships, filters, and even in other calculations.

Power BI Helper can help you identify which columns are not used anywhere. You can check that by going to the Visualization tab in Power BI Helper (see Figure 44-21).



**Figure 44-21.** Deleting the columns not used in visualizations or in other calculations

This option simply deletes all the columns that are not needed. Figure 44-22 shows that the file was reduced in size by more than 80 percent.

File	Last modified	Size
TOPN Example RADACAD.pbix	5/27/2021 11:34 AM	Microsoft.Micro... 2,543 KB
TOPN Example RADACAD after chang...	6/2/2021 9:55 AM	Microsoft.Micro... 656 KB

**Figure 44-22.** Power BI file size reduced by more than 80 percent

## Documenting Power BI Tenant Objects

Previously I explained how you can document a single Power BI file. Another very common documentation requirement is to determine which objects and workspaces exist in the Power BI tenant and document them all. You might even want to download all the PBIX files from the service and keep them for versioning purposes. This section explains how you can do that in just a few clicks.

722

## Service Tenant Settings in Power BI Helper

To use any of the Power BI Service features in Power BI Helper, you need to register Power BI Helper as an app under your tenant, which is explained next.

---

**Note** Power BI Helper does *not* store any information from your tenant. The list of workspaces, datasets, dashboards, reports, dataflows, users—everything—is populated at runtime in the application. Nothing is stored on our servers or databases—not even your username, password, or application ID.

---

## Register the App

Go to this URL in a web browser: [dev.powerbi.com/apps](https://dev.powerbi.com/apps). Sign in using your Power BI account (see Figure 44-23).



## Register your application for Power BI

Register your application with Azure Active Directory (Azure AD). You'll be able to manage and modify  
[Learn more](#)

### STEP 1 Sign in to Power BI

Sign in to your Power BI account to start registering your application.

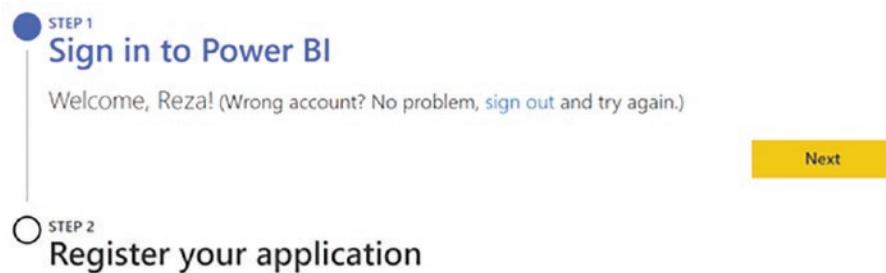
[Sign in](#)

Don't have an account? [Sign up now, for free](#)

### STEP 2 Register your application

**Figure 44-23.** Signing in to Power BI

After logging in, go to the next step (see Figure 44-24).



**Figure 44-24.** Registering your application

Apply the settings shown in Figure 44-25 in the Register Your Application dialog box.

## STEP 2 Register your application

Register your application with Azure AD to allow your application to access the Power BI REST APIs and to set resource permissions for your app. You can change this later in the Microsoft Azure portal. [Learn more](#)

### Application Name

Enter a display name to identify your application in Azure

Power BI Helper

### Application Type

Choose the type of application you are developing

Native (for apps that run on client devices, such as Android, iOS, Windows, etc.)

### API access

Select the APIs and the level of access your app needs. You can change these settings later in the Azure portal.

[Learn more](#)

Select all

#### Read only APIs ⓘ

- Read all datasets
- Read all dashboards
- Read all reports
- Read all workspaces
- Read all capacities
- Read all storage accounts
- Read all dataflows
- Read all gateways
- Read all Power BI apps

#### Read and write APIs ⓘ

- Read and write all datasets
- Read and write all dashboards
- Read and write all reports
- Read and write all workspaces
- Read and write all capacities
- Read and write all storage accounts
- Read and write all dataflows
- Read and write all gateways

#### Create APIs ⓘ

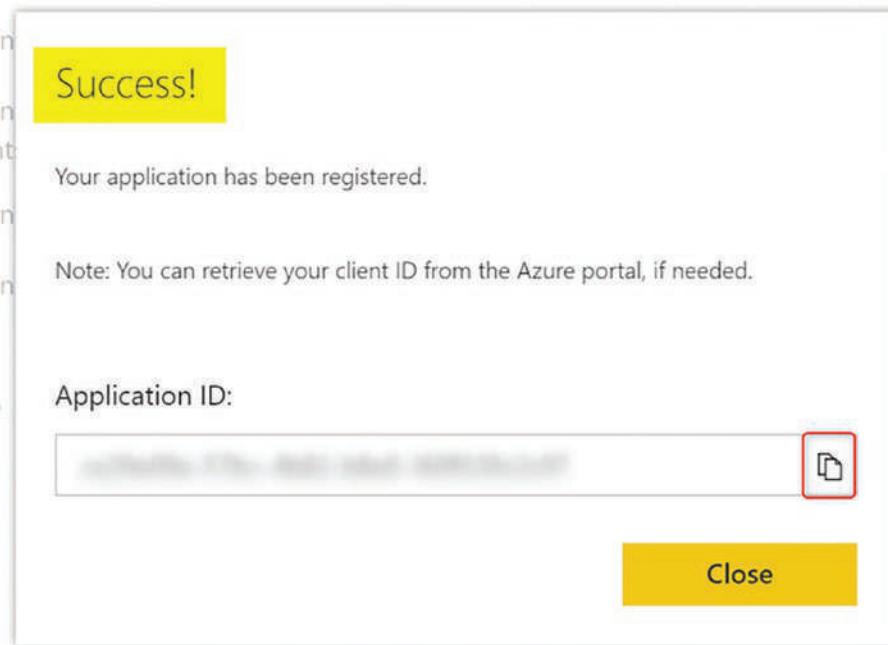
- Create APIs

By clicking Register, you agree to the [terms of use](#)

Register

**Figure 44-25.** Entering the desired settings in the Register Your Application dialog box

After successfully registering the app, copy the Application ID (see Figure 44-26).



**Figure 44-26.** Copying the application ID

This Application ID will be added in the About tab of Power BI Helper later, but there are still a few more steps to do first.

## Set Up Permissions in the Azure Portal

Log in to Azure Portal at [portal.azure.com](https://portal.azure.com). Go to Azure Active Directory, and then go to App Registrations (see Figure 44-27). Find the Power BI Helper app and click it.



**Figure 44-27.** Locating App Registrations in Azure Active Directory

Select the app, as shown in Figure 44-28, and continue.

The screenshot shows the 'App registrations' section of the Azure portal. At the top, there are tabs for 'All applications' and 'Owned applications'. Below them is a search bar with placeholder text 'Start typing a name or Application ID to filter these results'. A single application entry is listed: 'Power BI Helper'. This entry includes fields for 'Application (client) ID' (e65d7702-b067-4a30-a399-2965c8514382), 'Created on' (2/2/2021), and 'Certificates & secrets'. A red box highlights the 'Power BI Helper' entry.

**Figure 44-28.** Selecting the Power BI Helper app

Choose Request API Permissions and select Power BI Service, as shown in Figure 44-29.

The screenshot shows the 'Power BI Helper | API permissions' page. On the left, there's a sidebar with navigation links like Overview, Quickstart, Integration assistance, Manage (Branding, Authentication, Certificates & secrets, Token configuration, API permissions, Expose an API, Applications | Preview, Owners, Roles and administrators | Preview, Manifest, Support + Troubleshooting, New support request), and Help + Feedback. The 'API permissions' link is highlighted with a red box and a circled '1'. The main area shows 'Configured permissions' and a 'Request API permissions' grid. In the 'Request API permissions' grid, the 'Power BI Service' row is highlighted with a red box and a circled '1'. Other rows include Dynamics 365 Business Central, Dynamics CRM, Flow Service, Office 365 Management API, OneNote, SharePoint, and Yammer.

**Figure 44-29.** Selecting Request API permissions

Next, add all the requests, as shown in Figure 44-30.

The screenshot shows the Microsoft Azure portal interface. On the left, the navigation pane includes Home, App registrations, Power BI Helper, Overview, Quickstart, Integration assistant, Manage (with sub-options like Branding, Authentication, Certificates & secrets, Token configuration, API permissions, Expose an API, App roles | Preview, Owners, Roles and administrators | Preview, Manifest, Support + Troubleshooting, and New support request), and Troubleshooting.

The main content area displays the "Power BI Helper | API permissions" page. It shows a table of configured permissions:

API / Permissions name	Type	Description	Admin consent req...	Status
User.Read	Delegated	Sign in and read user profile	-	***
<b>Power BI Service (10)</b>				
App.Read.All	Delegated	View all Power BI apps	-	***
Capacity.Read.All	Delegated	View all capacities	-	***
Capacity.ReadWrite.All	Delegated	Read and write all capacities	-	***
Content.Create	Delegated	Create content	-	***
Dashboard.Read.All	Delegated	View all dashboards	-	***
Dashboard.ReadWrite.All	Delegated	Read and write all dashboards	-	***
Dataflow.Read.All	Delegated	View all dataflows	-	***
Dataflow.ReadWrite.All	Delegated	Read and write all dataflows	-	***
Dataset.Read.All	Delegated	View all data	-	***

A modal dialog titled "Request API permissions" is open over the table. It has sections for "Configured permissions" (with a note about Azure AD Graph API deprecation) and "Select permissions". Under "Select permissions", there are two expanded sections: "App (1)" and "Capacity (2)". Both sections have checkboxes for "App.Read.All" and "Capacity.Read.All". A red box highlights the "Add permissions" button at the bottom right of the dialog.

**Figure 44-30.** Adding all requests

Then choose Grant Admin Consent, as shown in Figure 44-31.

This screenshot shows the same "Power BI Helper | API permissions" page as Figure 44-30, but with a different state. The "Grant admin consent for RADACAD" checkbox is checked. The table now shows the status column for each permission row, with all entries showing a green checkmark icon indicating successful admin consent.

API / Permissions name	Type	Description	Admin consent req...	Status
User.Read	Delegated	Sign in and read user profile	-	***
<b>Power BI Service (10)</b>				
App.Read.All	Delegated	View all Power BI apps	-	***
Capacity.Read.All	Delegated	View all capacities	-	***
Capacity.ReadWrite.All	Delegated	Read and write all capacities	-	***
Content.Create	Delegated	Create content	-	***
Dashboard.Read.All	Delegated	View all dashboards	-	***
Dashboard.ReadWrite.All	Delegated	Read and write all dashboards	-	***
Dataflow.Read.All	Delegated	View all dataflows	-	***
Dataflow.ReadWrite.All	Delegated	Read and write all dataflows	-	***
Dataset.Read.All	Delegated	View all data	-	***

**Figure 44-31.** Grant admin consent

Go to authentication and enable LiveSDK, as shown in Figure 44-32.

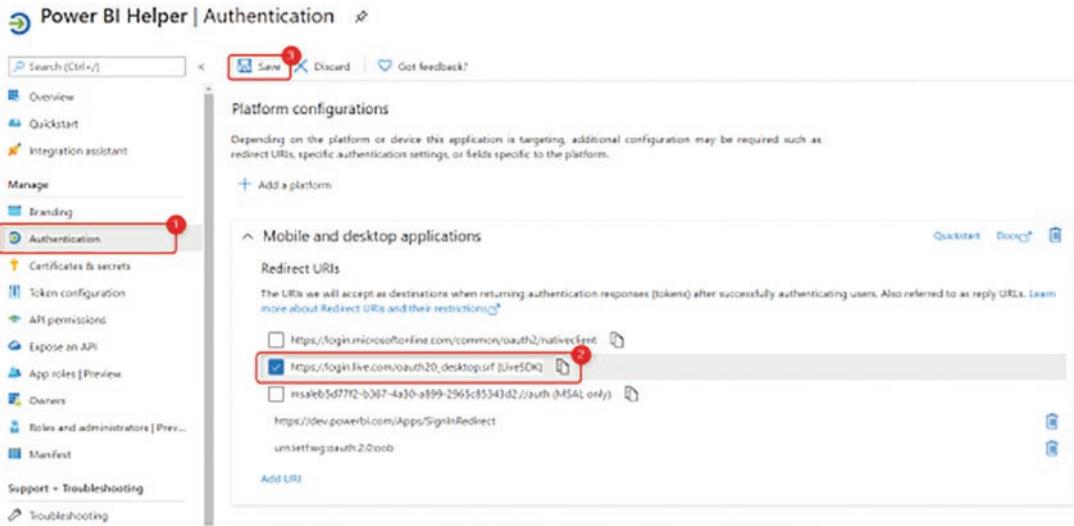
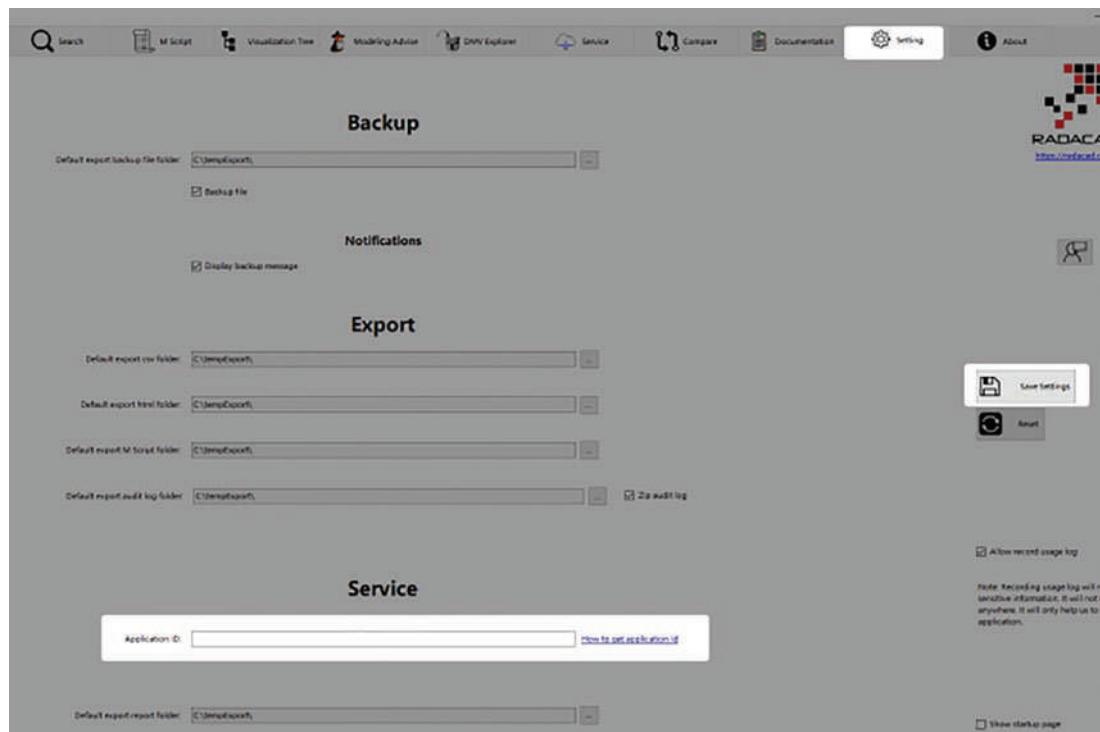


Figure 44-32. Enabling LiveSDK

## Add the Application ID to Power BI Helper

The last step is to add the Application ID to the Setting tab of Power BI Helper and then save it (see Figure 44-33).

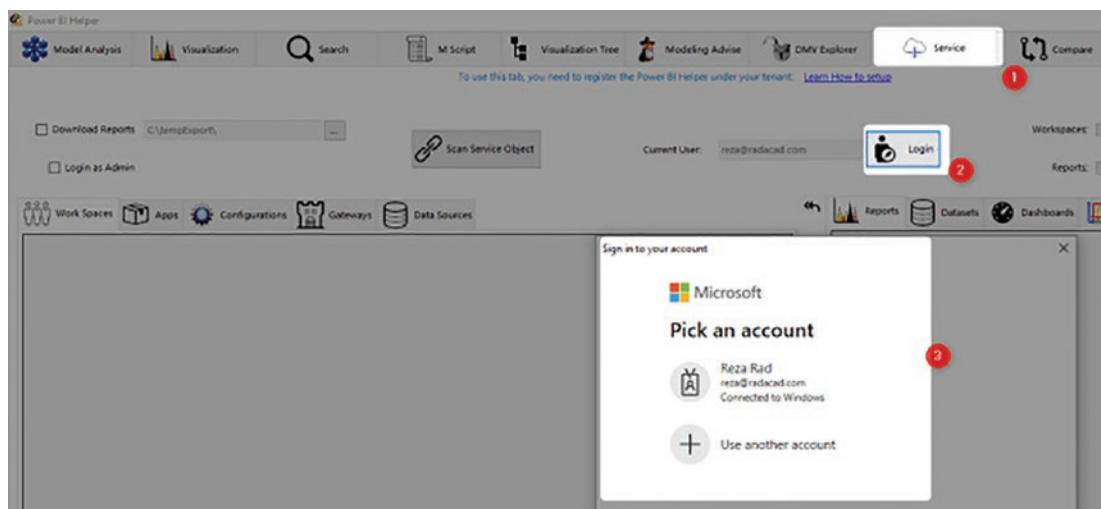


**Figure 44-33.** Adding the ApplicationID to the Setting tab of Power BI Helper

You can now use Power BI Helper's Service tab.

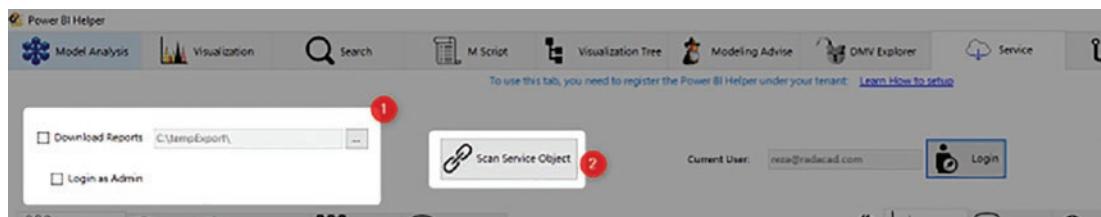
## Scan Service Objects

After you register the app (this has to be done only once), you can log in with your Power BI account. As Figure 44-34 shows, go to the Service tab in Power BI Helper and log in. You should use the Power BI account that contains the objects you want to be document. (Power BI Helper can only see the objects that this Power BI account has access to. If you want all the content on the service, try the Power BI Administrator account.)



**Figure 44-34.** Logging in to the service from Power BI Helper

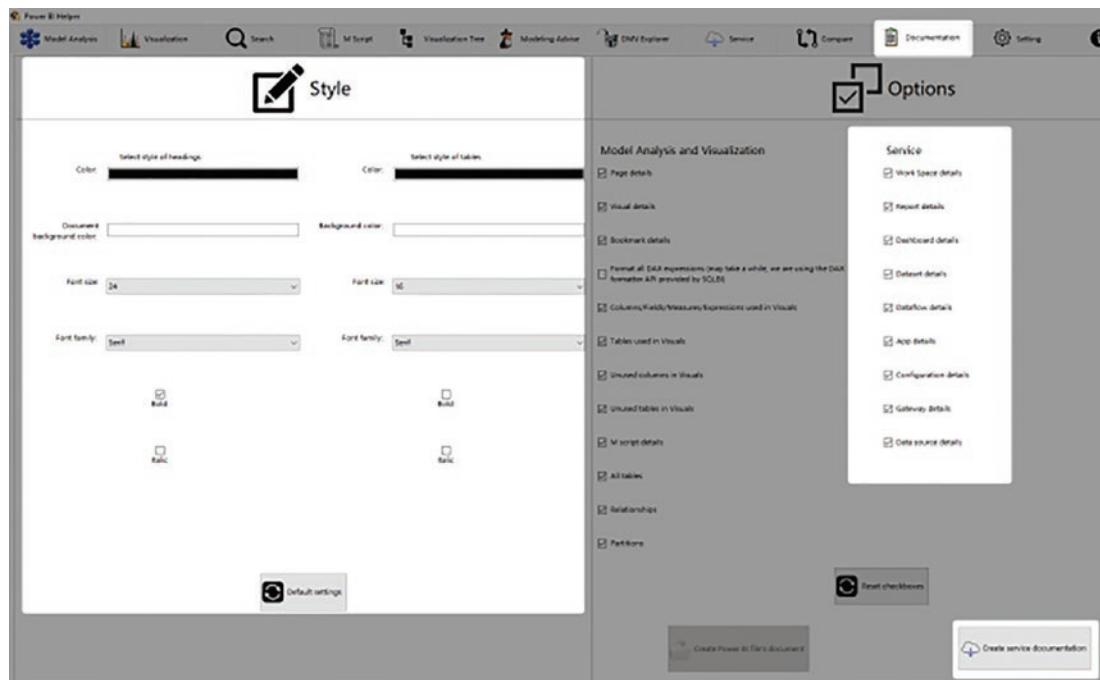
After logging in, you can choose if you want to download reports (and choose the folder to do so). You also need to check the box that says Login as Admin if you are using the Power BI Administrator account to log in. After that, you can choose Scan Service Object (see Figure 44-35).



**Figure 44-35.** Service settings for Power BI Helper

Scanning service objects might take some time; it depends on your Internet connection, whether you selected the Download Reports option, and the number of objects (reports, datasets, dataflows, and workspaces) you have on the Power BI tenant. After the process is done, you should see the objects listed in Power BI Helper.

In this tab of Power BI Helper, you can perform many operations on the objects. However, this chapter focuses on documentation. You can simply click Export to Document, or go to the Documentation tab and click Create Service Documentation (see Figure 44-36).



**Figure 44-36.** Creating Power BI Service documentation

Objects that are documented (at the time this chapter was written) include the following:

- Lists of workspaces
- Lists of reports
- Lists of dashboards
- Lists of datasets
- Lists of dataflows
- Lists of apps
- Configuration details
- Lists of gateways
- Lists of data sources

## Configuring the Documentation's Output

As you have seen, creating the documentation takes just a few clicks. If you want to configure the output document, you can do that from the Documentation tab, in two main sections. The first is the Style section.

## Configuring the Documentation's Style

As noted earlier in the chapter, the Style configuration determines the fonts, colors, size of tables, headings, and other parts of the generated document (see Figure 44-37).

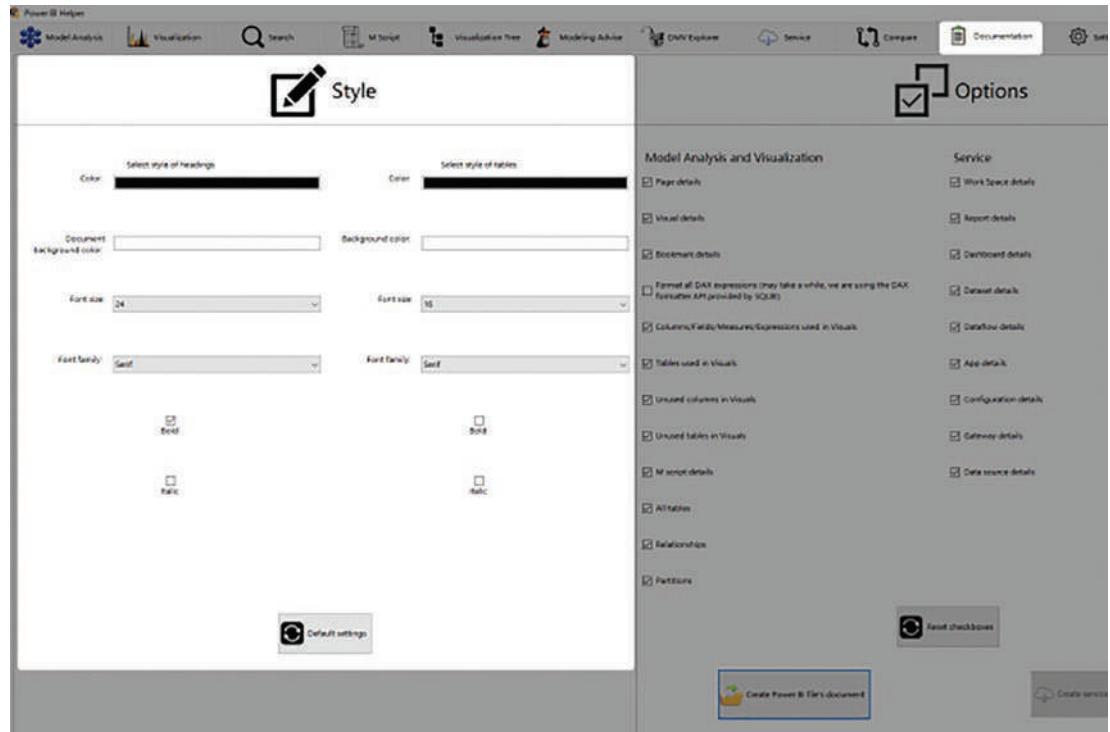
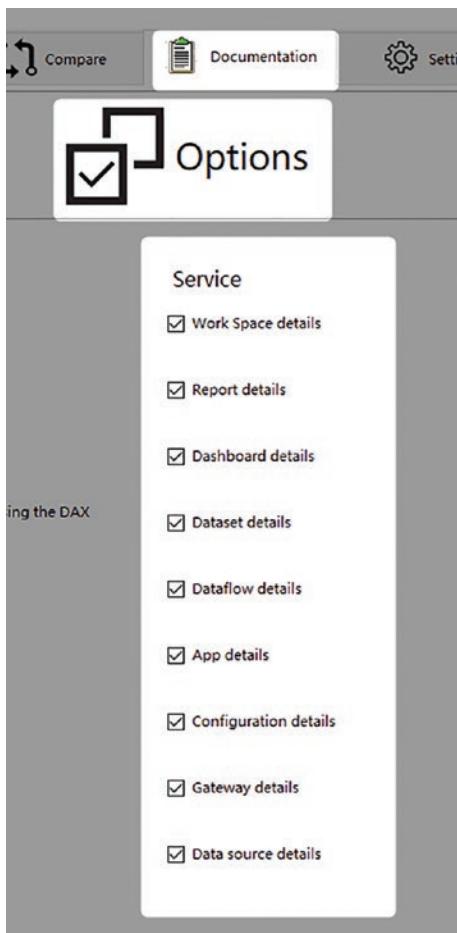


Figure 44-37. Configuring the style of the documentation

## Sections and Information to Document

You can choose what information you want to document from the Power BI tenant (see Figure 44-38). I recommend keeping them all selected for the full documentation.



**Figure 44-38.** Choosing the objects list to be documented from the Power BI Service

## Export the Information as CSV Files (Export Metadata)

I find it helpful to export this information as a CSV file for each object type. Power BI Helper can do that for you, simply by clicking the Export Metadata option (see Figure 44-39).

The screenshot shows the Power BI Helper interface with the 'Reports' tab selected. On the left, there's a table with columns: Name, Dashboards, Reports, Datasets, Dataflows, Date, Type, On-Dedicated Capacity, and Status. The table lists several items like 'DPIs (1)', 'Dashboards (1)', 'Dataflows (1)', etc. On the right, there's a preview of the CSV export with columns: Group, Report, Dataset, Downloaded, and Opened. The preview shows rows for 'DPIs (1)' and 'Power BI Helper'.

**Figure 44-39.** Exporting a Power BI Service objects list as a CSV file

Figure 44-40 illustrates the exported result.

<input type="checkbox"/> Apps service export 20210604 104850.csv	6/4/2021 10:48 AM	Microsoft Excel C...	2 KB
<input checked="" type="checkbox"/> Configurations service export 20210604 104850.csv	6/4/2021 10:48 AM	Microsoft Excel C...	1 KB
<input type="checkbox"/> Dashboards service export 20210604 104850.csv	6/4/2021 10:48 AM	Microsoft Excel C...	1 KB
<input type="checkbox"/> Dataflows service export 20210604 104850.csv	6/4/2021 10:48 AM	Microsoft Excel C...	8 KB
<input type="checkbox"/> Datasets service export 20210604 104850.csv	6/4/2021 10:48 AM	Microsoft Excel C...	75 KB
<input type="checkbox"/> DataSources service export 20210604 104850.csv	6/4/2021 10:48 AM	Microsoft Excel C...	6 KB
<input type="checkbox"/> Gateways service export 20210604 104850.csv	6/4/2021 10:48 AM	Microsoft Excel C...	2 KB
<input type="checkbox"/> Reports service export 20210604 104850.csv	6/4/2021 10:48 AM	Microsoft Excel C...	93 KB
<input type="checkbox"/> WorkSpaces service export 20210604 104850.csv	6/4/2021 10:48 AM	Microsoft Excel C...	3 KB

**Figure 44-40.** Exporting the Power BI Service metadata (document) as CSV files

## Exporting the Power BI Audit Log

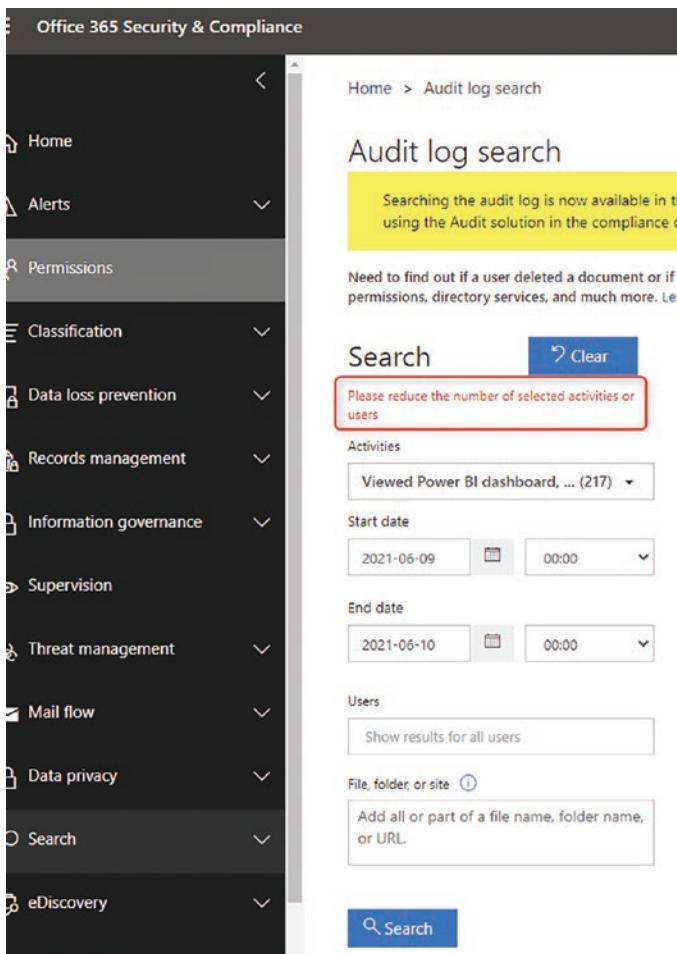
This can be done in a few simple steps, without the need for code. Power BI-related activity can be often useful for determining the behavior of the users and for enhancing the adoption of Power BI. Getting the audit log isn't simple though. This section shows you an easy way (with no scripting required) to export the audit log of Power BI.

### What Is the Power BI Audit Log?

The Audit log of Power BI lists all the Power BI-related activities done under your organization's tenant. You can get heaps of information from report usage, such as who opened the reports or dashboards, who shared them, who published them, who accessed them using Analyze in Excel, and much more. The Power BI Audit log includes (but is not limited to) a log of operations.

### Difficulties Exporting the Audit Log

The Power BI Administrator can access the audit log in two ways. One is by going to the Office 365 admin center and exporting the log from there (see Figure 44-41).



**Figure 44-41.** Exporting a limited version of the Power BI log from the admin center of Office 365

This method, however, only exports a limited result. As you see in Figure 44-41, you'll often get a message saying, "Please reduce the number of selected activities or users."

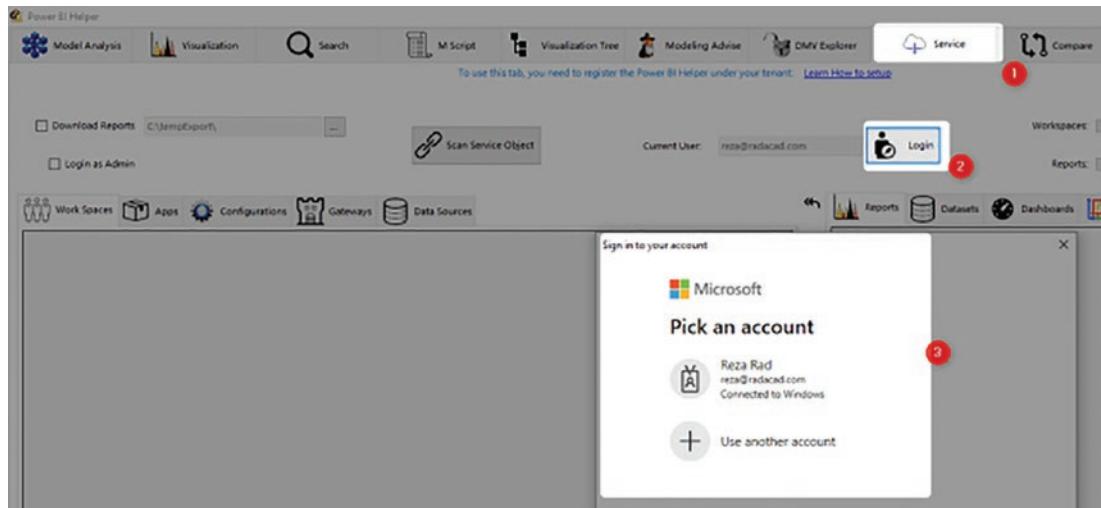
Another method is to use the PowerShell commands and scripts to export the log. The following section explains how to do this.

## A Simple Way to Export the Audit Log Without Limitations

You don't need to write or run PowerShell scripts to export the audit log. All you need is the Power BI Helper tool.

## Logging In

After you register the app, you can log in with your Power BI account. Go to the Service tab in Power BI Helper and log in. You should use the Power BI account that has either Power BI Administrator or Office 365 Administrator access to export the audit logs. You may also need to check the Login as Admin box (see Figure 44-42).



**Figure 44-42.** Logging in to the service from Power BI Helper

After logging in, go to the Export Audit Log section of the Service tab. From there, you can set the configuration of the period, set the export location, determine if you want the file to be zipped or not, and then export the log (see Figure 44-43).

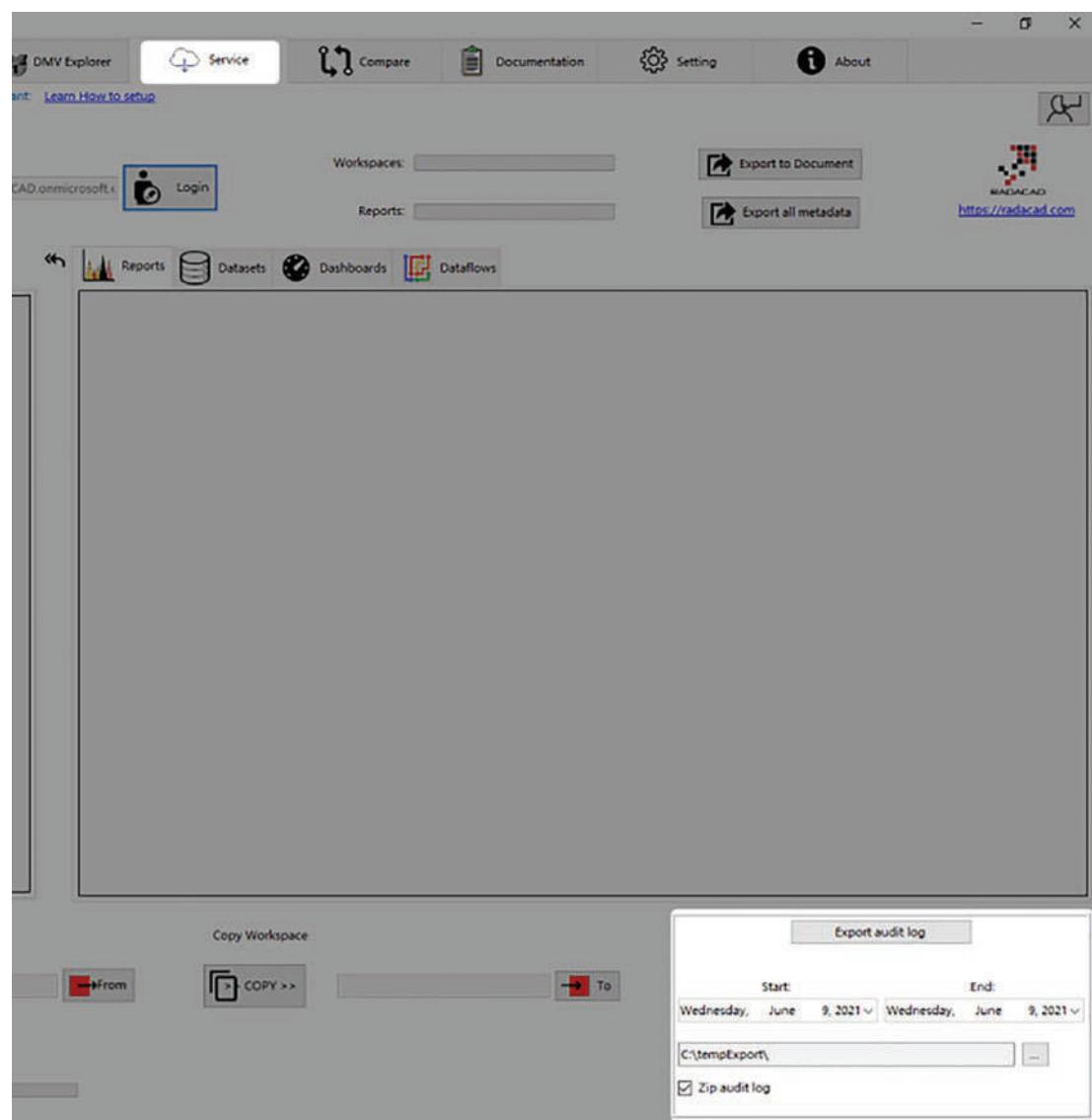
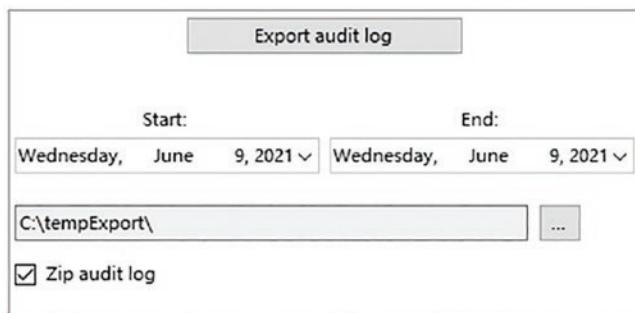


Figure 44-43. Exporting the audit log of Power BI using Power BI Helper

The settings for the export audit log are shown in Figure 44-44.



**Figure 44-44.** Exporting audit log settings in Power BI Helper

As shown in Figure 44-45, the audit log will be exported as CSV files (which can be zipped).

AuditLog from 2021-05-01 to 2021-06-09.zip							
	Name	Type	Compressed size	Password ...	Size:	Ratio	Date modified
<input type="checkbox"/>	AuditLog 2021-05-10.csv	Microsoft Excel Comma S...	1 KB	No	2 KB	60%	6/9/2021 11:44 AM
<input type="checkbox"/>	AuditLog 2021-05-11.csv	Microsoft Excel Comma S...	2 KB	No	5 KB	77%	6/9/2021 11:44 AM
<input type="checkbox"/>	AuditLog 2021-05-12.csv	Microsoft Excel Comma S...	1 KB	No	2 KB	60%	6/9/2021 11:44 AM
<input type="checkbox"/>	AuditLog 2021-05-13.csv	Microsoft Excel Comma S...	1 KB	No	1 KB	42%	6/9/2021 11:44 AM
<input type="checkbox"/>	AuditLog 2021-05-14.csv	Microsoft Excel Comma S...	1 KB	No	1 KB	41%	6/9/2021 11:44 AM
<input type="checkbox"/>	AuditLog 2021-05-15.csv	Microsoft Excel Comma S...	1 KB	No	2 KB	60%	6/9/2021 11:44 AM
<input type="checkbox"/>	AuditLog 2021-05-16.csv	Microsoft Excel Comma S...	1 KB	No	2 KB	60%	6/9/2021 11:44 AM
<input type="checkbox"/>	AuditLog 2021-05-17.csv	Microsoft Excel Comma S...	1 KB	No	2 KB	60%	6/9/2021 11:44 AM
<input type="checkbox"/>	AuditLog 2021-05-18.csv	Microsoft Excel Comma S...	1 KB	No	1 KB	41%	6/9/2021 11:44 AM

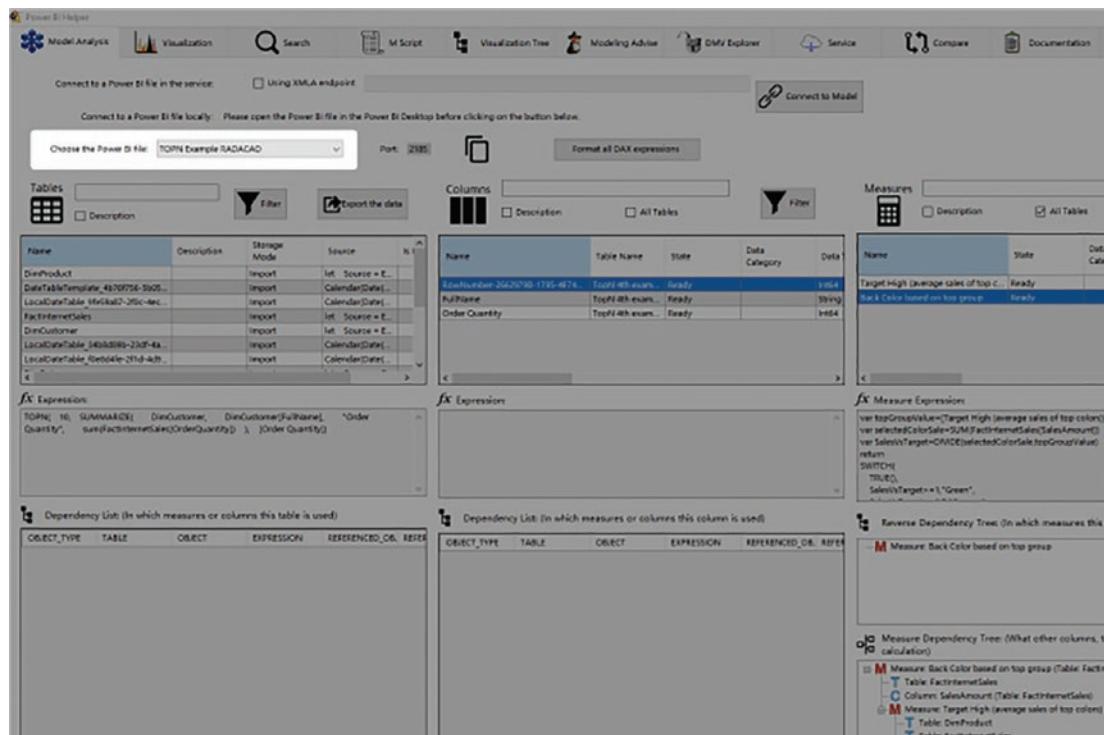
**Figure 44-45.** Sample audit log exported as CSV files

## Power BI File Cleanup in a Few Steps

If you have used a Power BI file for a while, you might have noticed that the model gets bigger and bigger. The file size increases and the number of pages, tables, fields also increase. You might get to a point where it becomes hard to determine which tables and fields are useful and which are not. You have to search for something to find it, and you have many duplicate calculations. This section explains how you can clean up the Power BI file by removing unused fields and measures simply and quickly.

Once Power BI Helper is open, click Connect to Model (Make sure that the Power BI file is already open in the Power BI Desktop.) Then, connect to the model from Power BI Helper.

This option will detect all the instances of Power BI files open in the desktop and will list them in a drop-down. Choose the one you want (see Figure 44-46).

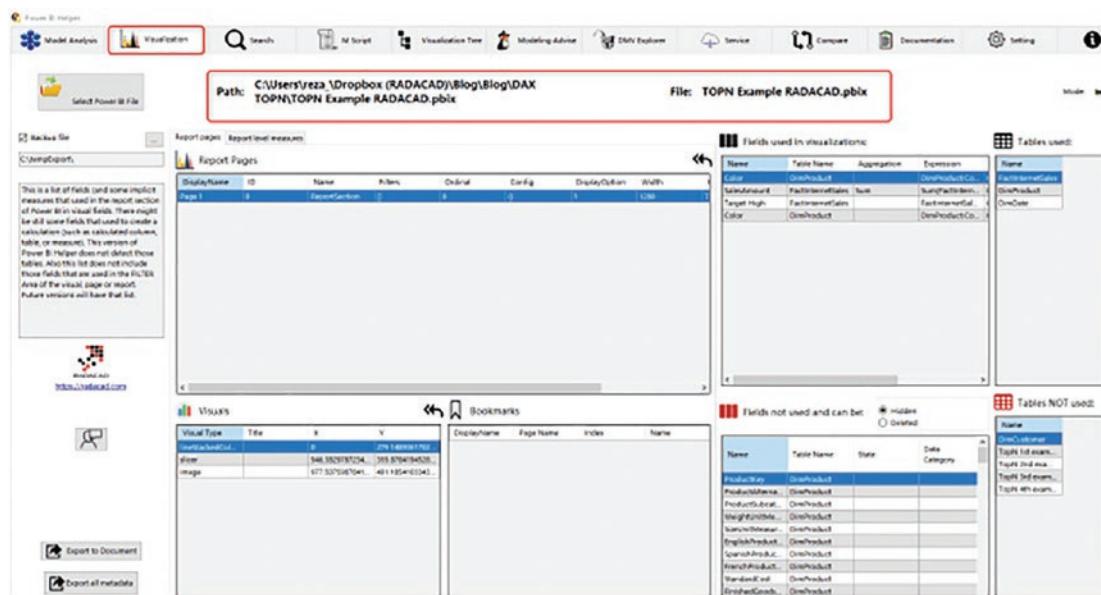


**Figure 44-46.** Selecting the Power BI file in Power BI Helper

If you have only one Power BI file open, the drop-down will show just that one. The rest of the page shows information about the model. It lists information such as the tables, columns, and measures with their expressions and expression trees.

## Visualization Information

Power BI Helper automatically finds all the visualization information of the selected model and shows you the information in the Visualization tab (see Figure 44-47).

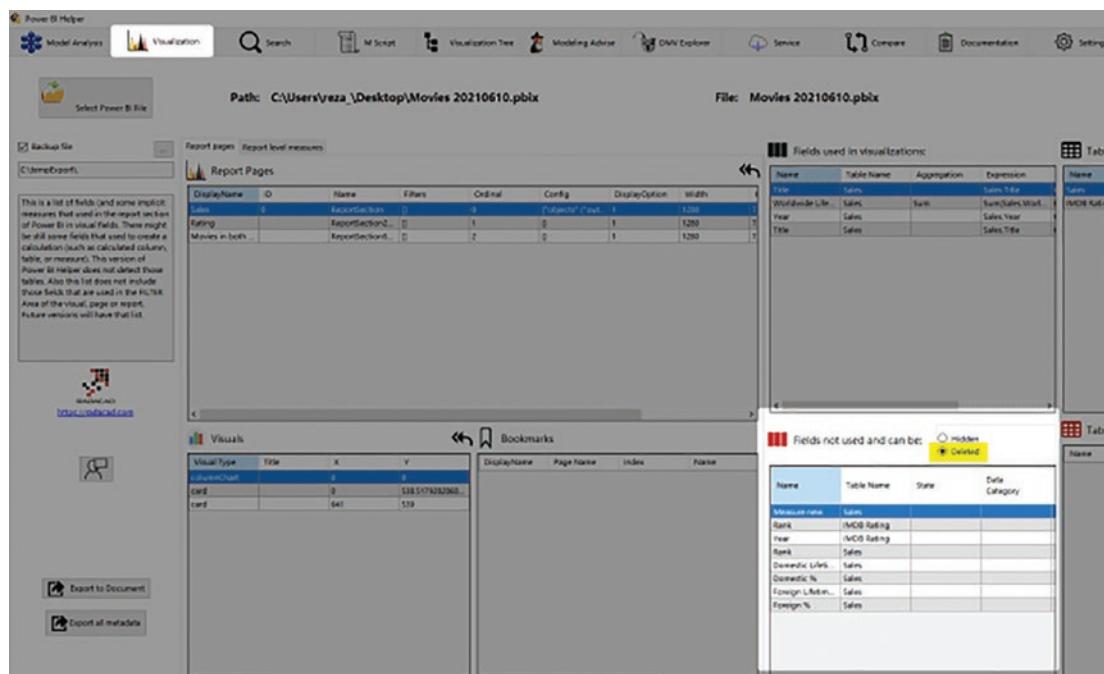


**Figure 44-47.** Power BI Helper automatically shows the visualization information after selecting the model

There are a few cases where Power BI Helper doesn't show the visualization information automatically (for example, if the Power BI file was just generated and has not yet been saved). In those cases, you can use the Select Power BI File option to choose the file.

## Removing Unused Fields

If you find a field that's not used in any visual, filter, or other calculations, it is safe to remove this field. Power BI Helper can identify these fields even when they are DAX calculated columns or measures. Using the dependency tree of the measure, Power BI Helper will identify if the field is used in another calculation that is used in a visual or filter. You can see the fields that can be removed in Figure 44-48.



**Figure 44-48.** Unused fields in Power BI detected using Power BI Helper

As you can see in Figure 44-48's list, there is also a measure that is not used, and no other measures use this measure in any visuals or filters.

Figure 44-49 shows how you can remove them.

Name	Table Name	State	Data Category
Measure new	Sales		
Rank	IMDB Rating		
Year	IMDB Rating		
Rank	Sales		<span>Hide</span>
Domestic Lifeti...	Sales		<span>Delete</span>
Domestic %	Sales		<span>Select all</span>
Foreign Lifetim...	Sales		
Foreign %	Sales		

**Figure 44-49.** Deleting unused fields in Power BI

You can delete all the fields by using Select All and then Delete. Or you can delete them one by one.

## Hiding Technical Fields

Technical fields are necessary for the model and cannot be deleted, but they are not used directly in visualizations. Examples of these fields include:

- Fields used in relationships
- Fields used to sort other columns
- Fields used to create a hierarchy

Technical columns are best hidden. You can see the fields to hide in Figure 44-50.

The screenshot shows a window titled 'Fields not used and can be:' with two radio button options: 'Hidden' (selected) and 'Deleted'. Below the list is a table with four columns: Name, Table Name, State, and Data Category. Two rows are visible: one for 'MovieKey' in the 'IMDB Rating' table and another for 'MovieKey' in the 'Sales' table. The 'Hidden' radio button is highlighted with a yellow box.

Name	Table Name	State	Data Category
MovieKey	IMDB Rating		
MovieKey	Sales		

**Figure 44-50.** Detecting fields in Power BI that can be hidden from the report view

In this example, the fields are used in a relationship (that is why they don't appear in the list to be deleted). Hiding them is recommended if they are not used directly in visuals in Power BI.

Figure 44-51 shows how you can hide them.

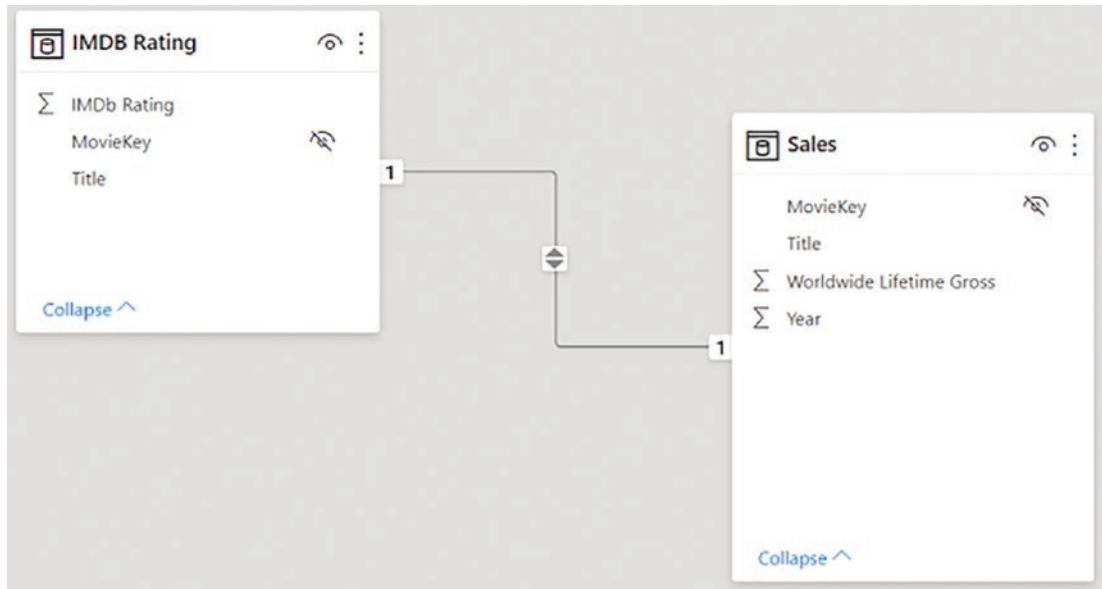
The screenshot shows the same interface as Figure 44-50. A context menu is open over the second row of the table, specifically over the 'MovieKey' entry in the 'Sales' table. The menu has three options: 'Hide' (selected), 'Delete', and 'Select all'. The 'Hide' option is highlighted with a blue box.

Name	Table Name	State	Data Category
MovieKey	IMDB Rating		
MovieKey	Sales		

**Figure 44-51.** Hiding Power BI fields in a few clicks using Power BI Helper

## Sample Output

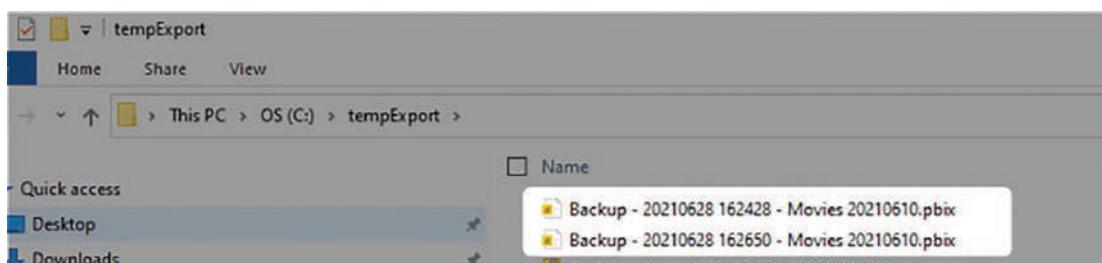
The output will be saved in the same Power BI file that is open on the Power BI Desktop. You can see the result of the sample in Figure 44-52.



**Figure 44-52.** Cleaned Power BI file

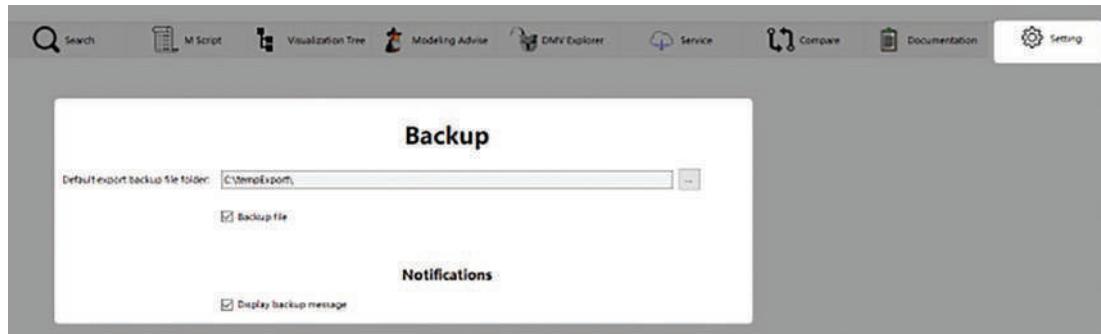
## Backup File

Power BI Helper, by default, keeps a version of the file before the changes are saved as a backup. This backup can be helpful if unwanted behavior is detected (see Figure 44-53).



**Figure 44-53.** Auto backup of Power BI file before changes

You can customize the backup file's settings from the Setting tab of Power BI Helper (see Figure 44-54).



**Figure 44-54.** Backup file settings in Power BI Helper

## Sample Scenarios

- If a field is directly used in a visual, it will not be removed or hidden
- If a measure is not used in visuals or other measures, it will be removed
- If a measure is not used directly in a visual, it will be hidden

## Not Supported at Present

Fields and measures that are used across multiple files are not included in these actions (such as if another Power BI file uses the fields from this Power BI file).

## Summary

For this last chapter of this book, I wanted to leave you with a useful tool, and that is why I included Power BI Helper. Power BI Helper is a free-to-use tool that helps with development, documentation, performance tuning, cleanup, and administration of your Power BI solution.

# Index

## A

- AAS dataset, 100
- Active Directory, 371, 491, 665, 726
- Admin Portal
  - custom branding, 468, 469
  - embed codes, 465, 466
  - Office 365 administrator, 451
  - organization visual, 466, 468
  - usage metrics, 465
  - workspaces, 468
- AdventureWorksDW database, 52, 53, 91
- Adventure Works Internet Sales model, 70
- AdventureWorks Tabular model, 69
- Aggregated table, 282, 333
  - add aggregations, 288
  - advanced Group By, 287
  - CalendarYear, 281
  - columns
    - Count function, 291
    - decimal data type, 290
    - operation, 289
    - Whole Number data type, 291
  - create, 285–287, 289
  - data model, 292
  - data results, 335
  - vs. dimension tables, 292
- DirectQuery, 284
- field selection, Group By, 288
- Group By, 287
  - hide
    - test result, 316–318
    - user, 315
    - view, 315
  - import, memory, 281
  - Power BI, 285
  - querying, 283
  - Sales Agg, 291
  - sample, 289
  - sample model, 283, 285
  - setting up Power query, 286
- source table, DirectQuery, 293
- SQL Server database, 283
- T-SQL statements, SQL Server, 285
- Aggregations, 280, 364
  - aggregated table, 281 (*See also* Aggregated table)
  - automatic, 336, 337
  - awareness, 330
  - big tables, 364
  - column
    - count function, 314
    - sum function, 313
  - configuration, 311
  - create relationships, 281
  - DirectQuery connection, 280
  - dual storage mode, 293, 302 (*See also* Dual storage mode)
  - Group By fields, 312
  - layers, 282, 283
  - Manage Aggregations, 310, 314
  - multiple layers (*see* Multiple layers, aggregations)
  - Power BI, 310
  - selection, 311
  - setting aggregation as sum, 313
  - setting Group By fields, 313
  - setup, 312
  - storage modes (*see* Storage modes)
- Analysis Services database, 73, 78
- Analysis Services model, 123
- Analysis Services Tabular, 7, 8, 11, 30, 36, 43, 69
- Analytical Engine (Tabular Engine), 7, 8, 69, 73, 84, 85, 125, 127
- App Workspaces/Organizational Workspaces, 536
- Audit Log
  - extract logs, using PowerShell
    - command, 503
    - export process, 505
    - install/import package, 503
    - methods, 502
    - Power BI report, 505–509, 511, 512
    - running administrator, 502
  - Search-UnifiedAuditLog function, 504

- Audit Log (*cont.*)  
problem  
    Admin Portal, 499  
    error message, 501  
    search criteria, output, 500  
    search dialog, 499  
    selecting activities, 500  
    usage metrics, Admin Portal, 497  
    usage metrics report, 496, 497  
    viewing usage metrics, 495
- Audit log, export  
    difficulties, 735, 736  
    login, 737–739  
    operations, 735
- Automatic aggregation, 336, 337
- AW Internet Sales Tabular  
    Model 2014, 78
- Azure Analysis Services (AAS), 7, 11, 69, 83, 84, 100, 121, 126, 248, 514
- Azure Data Lake, 164, 209
- Azure Data Lake storage, 9, 130–133, 139, 140, 211, 212, 217, 219, 359, 469, 657, 658
- Azure DevOps, 681
- Azure SQL database, 148, 170, 171, 173, 175, 183, 190, 193–197, 201
- Azure Stream Analytics, 262, 270
- Azure Synapse, 172, 227
- **B**
- Background image, 362
- BI Ecosystem, 173, 174
- BIRADACAD, 78
- Blank Query, 148, 149
- Build access level  
    example, 642  
    licensing, 648  
    row-level security, 643–645, 649  
    user edit access  
        build access, 646–648  
        manage permissions, 645, 646
- **C**
- Capacity-based licensing  
    option, 442
- Power BI Embedded, 442–444
- Power BI Premium, 444–446
- SKUs  
    capacity considerations, 447  
    scenarios, 447  
    vs. P/EM SKUs, 446
- C# application, 265, 270
- Categorize Power Query objects, 360
- Centralized data warehouse, 139, 140, 164
- Chained datasets, 96, 122, 214, 215, 227, 228, 362, 458, 627
- Citizen data analysts, 169, 173, 186
- Cloud-based sources, 159
- Cloud-based technology, 4, 14, 18, 413, 416
- Clustered Column Store index, 59, 120
- Column-level security, 683, 703, 704
- Column-store in-memory technology, 118
- Composite mode, 64–67, 74, 85, 280, 302, 328
- Composite model  
    definition, 87  
    DirectQuery, 88, 89, 122, 123  
    Import Data, 88, 89  
    needs, 88  
    working mechanism, 90–93  
        dual storage mode, 93–95
- Computed entity, 132, 133, 223, 442
- Conditional formatting, 78, 362
- Configuring Report Server, 389
- Connections, 128
- Connection string, 193, 198, 351, 352
- Content certification/endorsement  
    assigning specific security group, 664  
    dataset discoverability, 665  
    example, 666–668  
    labeling system, 661  
    levels, 661, 662  
    report, 664  
    service environments, 663
- Count function, 291, 314
- Custom data connectors, 434
- Custom Date table, 363, 720
- **D**
- Dashboard/report sharing, 380  
    access levels, 528, 533  
    advantages, 532  
    content owner, 523  
    drawbacks, 532, 533  
    manage permissions, 528–532  
    sharing, 615  
    working, 523–528
- Data Analysis Expression (DAX), 11, 12, 43, 44, 49, 50  
    functions, 66  
    measures, 125, 331–336  
    username() function, 697
- Data changes  
    ModifiedDate/UpdatedDate column, 354
- Dataflows, 190  
    administrator's control, 141, 142
- Blank Query, 148, 149
- centralized data warehouse, 139, 140  
    vs. datasets, 224, 225

- data sources, 134, 144, 145
- data sources with different refresh schedules, 138, 139
- data storage, 131–133
- data transformation, 129, 130
- Date dimension, 137
- define new tables, 143, 144
- ETL layer, 222
- in Excel, 154, 155
- gateway setup, 147, 148
- multiple PBIX files, 138
- multiple Power BI files, 137
- in My Workspace, 140
- OData source, 145
- one Power Query table, 136
- online Power Query Editor, 149
- in organizational workspace, 141
- Power BI, 130, 131, 220, 221
- Power BI Desktop, 135, 136, 151–153
- Power BI Service, 140
- Power Query component, 219, 220
- Power Query Editor online, 146, 147, 150
- Power Query Online, 133, 134
- scheduled refresh, 150
- selecting tables for transformation, 146
- service-only (cloud-only) object, 130
- standard *vs.* analytical, 132, 133
- type, 135
  - in workspace, 143
- Data Hub, 160, 197, 202
- Data-level security
  - levels
    - column, 703
    - object, 704
    - page, 704
    - RLS, 683
- Data Lineage, 382
- Datamart components
  - Azure SQL database, 190, 193–197
  - dataflow, 190
  - dataset, 191, 192
  - Lineage view, 200–202
  - reports, 197–199
- Datamart Editor
  - create measures, 187
  - Data View, 183
  - features and capabilities, 182
  - Get Data and Transform
    - Data options, 183
    - incremental refresh, 187, 188
    - model diagram, 186, 187
    - New Query, 183–185
    - relationship editor, 186, 187
    - renaming, 190
    - row-level security, 189, 190
- in service, 190
- writing DAX measures, 187
- writing T-SQL queries, 185, 186
- Datamarts, 10, 11, 217–219, 225, 226, 228, 378
  - Azure SQL database, 171, 175, 197
  - BI Ecosystem, 173, 174
  - building, 170, 171
  - citizen data analyst, 173
  - components, 171, 172
  - creation, 178–180
  - database structure, 196
  - dataflows, 171, 226
  - vs.* datasets, 227
  - data warehouse, 172
  - developers, 173
  - Editor, 182–190
  - enterprises, 173
  - governance, 175
  - license, 175
  - loading data, 182
  - Power BI dataset, 171
  - Power Query Online, 179–182
  - Premium workspace, 175–177
  - SCD, 175
  - single unified UI, 174
  - SSMS, 170
  - web-based, 174
  - Web UI, 171, 175
- Data movement service, 418
- Datasets, 7, 8, 30, 217, 218, 223, 362
  - components, 159
  - DAX calculations and relationships, 221
  - vs.* dataflows, 224, 225
  - in Power BI Service, 159
  - and report, 157–159
  - settings, 351
- Data source
  - configuration, 408
  - connectors, 179
  - types, 33
  - users, 431, 432
- Data transformation, 5, 9, 41, 65–67, 75, 129, 130, 138, 145, 146, 155, 169, 173, 184, 209, 217, 222, 227, 228, 285, 360, 377, 628
- Dataverses, 130–132
- Data warehouse, 66, 75, 139, 140, 164, 170, 172, 173, 210, 218, 222, 225, 227, 228, 662
- Date dimension, 136, 137, 149
- DAX calculations
  - Power BI dataset, 359
  - reuse, 359
- Deployment pipelines, 379
  - assigning workspace, 675
  - comparing content, 676, 677
  - create pipelines, 674, 675

- Deployment pipelines (*cont.*)  
 data analytics team, 670  
 deploy process, 678, 679  
 end-user environment, 670  
 environment setup, 672  
 environments, implementation, 670  
 jack, 669  
 rules/connections, 680  
 TEST/LIVE environments, 673, 674  
 test user's environment, 671, 672
- Descriptive fields, 365
- Desktop tools, 26, 130, 451
- Developer API/REST API, 20, 21
- Dimensional model, 173, 211, 657
- DirectQuery, 67, 73, 84, 85, 277, 279, 285, 295, 298–300, 302, 303, 314, 316, 318, 319, 322, 326, 328, 330, 337  
 advantages, 119, 124  
 aggregations, 280  
 composite mode, 64, 65  
 create Power BI, 283, 284  
 data sources, 51, 52, 119  
 disadvantages, 119–121, 124  
 dual storage mode, 302, 306, 307  
 Import Data, 51  
 large scale dataset, 66, 67  
 limitations  
   DAX and modeling, 66  
   Power Query, 65, 66  
 no data tab, 54, 55  
 non-Microsoft sources, 127  
 no refresh need, 66  
 partition, 354, 357  
 performance  
   data source, 58, 59  
   Import Data, 58  
   maximum connections per data source, 62–64  
   query reduction, 60–62  
 Power BI, 51
- Sales Agg table, 302
- select tables, 53, 54
- source table, 293
- SQL Profiler, 55–57
- SQL Server, 52, 53
- storage mode, 296, 297
- table, 282, 301
- DirectQuery, Power BI Datasets  
 access building, source, 114, 115  
 creation, 103–106  
 data source credentials, 111, 112  
 feature, 101, 102  
 lineage view, 113  
*vs.* Live Connection, Power BI Datasets, 106  
 mechanism, 102
- multiple DirectQuery connections support, 107
- My Workspace, 102, 103
- publish to service, 108
- relationships between data sources, 107
- report sharing, 114
- “There is no gateway” error, 108–111
- Dual storage mode, 89, 95, 293  
 data visualization, 305  
 DirectQuery, 302, 304, 306, 307  
 DimCustomer, 308  
 DimDate table, 302, 303, 308  
 DimGeography, 308  
 DimProductSubcategory, 308  
 Import Data, 302  
 import table storage, 304  
 no queries, 303, 304  
 setting storage, 303  
 SQL Profiler tracks queries, 306  
 storage mode alert, 309  
 tables set, 309
- Dynamic Management Views (DMV), 515, 516
- Dynamic row-level security  
 assigning users, 698, 699  
 functions, 696, 697  
 Power BI desktop, 697  
 report/dashboard, sharing, 699, 700  
 sample data, 694, 695  
 sample report, 695, 696  
 testing security, 701–703

## ■ E

- EffectiveUserName, 81, 83
- Embedded Power BI  
 developer playground, 581, 582  
 implementation, 584, 585  
 ISV company, 579  
 licensing  
   pricing list, 582, 583  
   SKUs, turned off /on, 583  
   users or custom applications, 583  
 myths, 580  
 playground, 580  
 REST API, 584  
 sharing methods, 579
- Enterprise architecture, 215, 216
- Enterprise BI, 101, 102
- Entities, 101, 143, 147, 212, 213, 240, 658, 659
- Excel and Power BI integration, 241, 242  
 analyze in Excel  
   live, 252  
   row-level security, 253  
   security, 253  
   shared dataset, 253  
   visuals and fields, 253

analyze in Excel feature, 241  
 connection is live, 247  
 Excel features, 241  
 Excel workbook, 243  
 Excel workbook with a PivotTable, 243  
 explicit measures, 246  
 getting data from the Power BI dataset, 248–250  
 implicit measures, 244, 245  
 import Excel into Power BI Desktop, 256, 257  
 Power Platform dataflow, 251  
 publish data into Power BI, 254–256  
**E**xchangePowerShell, 503  
**E**xtensible Markup Language for Analysis (XMLA),  
 173, 192, 350, 460, 462, 517  
**E**xternal dataset, 163  
**E**xtract, transformation, and load (ETL) process,  
 139, 170

## F

**F**actInternetSales field, 355  
**F**ile cleanup  
 connect to model, 739  
 files open, 739, 740  
 removing unused fields, 741, 742  
 technical fields, hiding, 743–745  
 visualization information, 740, 741  
**F**iltering data, 343

## G

**G**ateways, 127  
 additional logging, 436  
 dataset setting, 435  
 data sources  
   gateway connection to dataset, 428, 429  
   on-premises, 415  
   Power BI Desktop, settings, 424  
   settings, 425, 427  
   type option, 426  
 execution flow, 417, 418  
 installation, 420  
   download, 419  
   Manage gateways, Power BI Service, 423  
   Power BI on-premises gateway status, 422  
   register gateway, 421  
   sign in, Power BI account, 419, 420  
   Tenant Administration, 423, 424  
 install, local domain, 415  
 logging system, 435  
 online/cloud-based data sources, 415  
 personal *vs.* standard on-premises modes, 416  
 Power BI cloud-based data analysis  
   technology, 415  
 requirements, gateway installation, 418

SQL Server database, 415  
**T**ypes  
 on-premises standard, 415–417  
 personal mode, 415, 416  
**G**itHub, 681  
**G**overnance, 169, 170, 175

## H

**H**elper tool  
 configuration, 710, 711  
 connect to model, 706, 707  
 documentation, 708–710  
 download/install, 705  
 exporting data, table  
   analyze in Excel, 717  
   any size/any table type, 716  
   connect to model, 712, 713  
   model analysis, 713–715  
   external tool, open, 705, 706  
   reduce file size, 717–722  
   visualization, 707, 708  
**H**ybrid dataset, 269, 274  
**H**ybrid tables, 339, 340, 349, 350, 357, 364  
**H**ybrid Table Setup  
 parameters, 348  
 PPU license, 347  
 PPU workspace, 348

## I

**I**mport data/scheduled refresh, 51, 74, 88, 94,  
 118, 302  
 advantages, 49, 118, 119, 124  
 aggregations  
   aggregated table, 328–330  
   DAX measures, 328, 331–336  
   Manage Aggregations window, 330  
 agility and performance, 125  
 combination data sources, 40, 41  
 compression engine, xVelocity, 118  
 configuration, 48  
 dataset  
   memory, 33  
   Power BI Service, 47  
 DAX, 43, 44  
 disadvantages, 49, 118, 119, 124  
 limitations, 50  
 load data  
   memory, 33  
   Power BI model, 35  
 Microsoft SQL Server Analysis Services, 38  
 navigator window, 34  
 next and last refresh time, dataset  
   properties, 48, 49

## ■ INDEX

Import data/scheduled refresh (*cont.*)

- Power BI, 50, 118
  - problems, 99
  - publish report
    - data source settings, 45
    - gateway configuration, 44–46
  - Select Excel, 33, 34
  - SQL Server database, 35
  - storage mode, 295
  - Task Manager, 38
  - transforming data, 42
  - xVelocity, 36
  - xVelocity compresses data, 36, 37
- Incremental Refresh, 187, 188, 354, 364
- configuration, 343
  - dataflows, 357
  - dataset, 339
  - data source language, 340
  - DateTime data type, 341
  - ETL processes, 356
  - FactInternetSales, 345
  - FactSales table, 340
  - parameters, 341, 342
  - partitions, 339
  - PBIX file, 340
  - PPU license, 340
  - RangeStart and RangeEnd parameters, 356
  - and Real-Time Data window, 344
  - settings, 340, 356
- Invoke\_RestMethod, 493

## ■ J

Jack, 669–671, 683

## ■ K

Key performance indicators (KPIs), 17, 18, 380

## ■ L

Layered architecture, 211, 213, 214, 659

Licensing, 20

- capacity-based (*see* Capacity-based licensing)
- categories, 437
- comparison table, 449
- options, 437
- Power BI Desktop, 438
- user-based (*see* User-based licensing)

Lineage view, 200–202

Live Connection, 161

- advantages, 124

- data sources, 69

- data sources creation, 78–80

- DAX queries, 72

*vs.* DirectQuery, 84, 121

Power Query transformations, 124

- relationship configuration, 123

- report-level measures, 124

disadvantages, 100, 125

EffectiveUserName, 81

Enterprise Solution, 126

OLAP/Tabular Engine, 121

performance, 73, 74

Power BI, 69

- report, 72

- service, 83, 84

Power Query transformations, 74, 75

report and gateway configuration, 78

report creation, 69–71

report-level measures, 76–78, 121, 125

Single Data Source, 74

Team Development, 126

UPN Mapping, 81–83

visualization, 121

## ■ M

Method selection, 125

Metrics, Power BI, 17, 18, 380

Microsoft SQL Server Database, 30

Microsoft Store, 23–27

Mobile applications, Power BI, 30, 92, 94, 234, 361, 439

Mobile-friendly Power BI reports, 234, 235

Modeling, composite model

- calculations and DAX, 96

- DirectQuery and Import Data, 97

- Power BI dataset, 96, 97

- relationships, 95, 96

M scripting language, 5, 6

Multi-layer architecture, 659

- challenges, using single PBIX file, 203, 204

- dataflows, 204, 227

- shared data sources, 206

Multiple layers, aggregations

- precedence setup, 321, 322

query

- data source, 327

- DirectQuery source table, 326–328

- first aggregated table, memory, 323, 324

- second aggregated table, memory, 324, 325

query path, 324

- second aggregated table, 326

second aggregation layer

- Group By settings, 318

- manage aggregations, 321

- sample relationship diagram, 319

- setting dual storage mode, 320

Multiple Power BI files, 137, 138, 204, 205, 359, 377

## N

Non-aggregated table, 333  
     data results, 336  
 Normal Power BI dataset, 263, 270, 278  
 Numeric fields, 244, 363

## O

Object-level security, 518, 683, 704  
 OData, 179, 180  
     source, 145, 201  
 Office 365 administrator, 439, 451, 737  
 On-premises gateway, 3, 18, 30, 415, 416, 422  
 On-premises standard gateway, 415–418

## P

Paginated reports, 15, 16, 229–231, 234, 383  
     best architecture practices, 235  
     mobile-friendly reports, 234  
     pixel-perfect printing, 229  
     *vs.* Power BI reports, 233, 234  
     in Power BI service, 229, 230, 240  
     printed output, 232, 233  
     printed version, 231, 232  
     with table visuals, 231  
 PBIGateway, 78  
 Power BI report (a \*.PBIX file), 157  
 PBIX download limitation, 340  
 PBIX file, 359  
 Power Automate, 265–267  
 Power BI, 363, 364  
     administrators, 141, 142  
     admin portal, 167, 178  
     advantages, 564  
     Analytical Engine (Tabular Engine), 7, 8  
     chained datasets, 228  
     citizen data analysts, 169  
     cloud-based technology, 4  
     components, 3, 4  
     creating app  
         audience, 557–559  
         content, 554–556  
         getting organization, 559–561  
         making changes, 561  
         setup, 552–554  
     content certification, 381  
 Dashboard, 8, 9  
 data connection, 117  
 dataflows, 9, 10, 130, 204, 217  
 Data Lineage view, 382  
 datamarts, 10, 11, 140, 170, 218, 219, 225, 226, 228, 379 (*See also* Datamarts)  
 data model, 363  
 data refresh, 564  
     datasets, 8, 130  
     DAX Language, 11, 12  
     Desktop, 4, 5  
     developer API/REST API, 20, 21  
     developers/users environment, 562–564  
     development and maintenance, 129  
     dev/user environments, 549  
     disadvantages, 565  
     ecosystem, 170  
     environment, 161  
     hosting reports, Power BI service, 12, 13  
     Import Data, 99  
     licensing, 374  
     Live Connection, 100  
     loads, 361  
     method, 117  
     metrics and scorecards, 17  
     mobile app, 13, 14  
     on-premises gateway, 18  
     paginated reports, 15, 16  
     Power Query, Data Transformation, 5  
     streaming dataset, 262–266  
     solution architecture, 116, 117  
     tenant administrator, 166  
     theme, 362  
     toolset, 369  
     web service page preview, 399  
     workspaces, 18, 19, 549–551  
 Power BI accounts  
     account type, 29  
     designation, 29  
     email, 28  
     functions, 27  
     Office 365 Portal, 28, 29  
     Trial Free, 27  
     Trial Pro accounts, 27  
 Power BI architecture, 164, 208  
     benefits, multi-layered architecture, 209  
         consistency and maintenance, 210  
         DAX calculations, 210  
         decoupling, 209  
         multi-developer environments, 210  
         tools, 210  
     chained datasets, 214, 215  
     datamarts, 210  
     layered architecture for dataflow, 211  
         layered architecture, 213, 214  
         staging database, 211, 212  
         transformation database, 212, 213  
     layers, 209  
     multi-layered architecture, 169, 170  
 Power BI dataset, 131, 135, 162, 171, 218, 339, 365  
     Data Hub, 160  
     endorsement, 165, 166  
     thin report, 160

- Power BI Desktop, 4, 5, 7, 70, 71, 130, 135, 136, 147, 151–153, 162, 173, 240, 343, 344, 349, 370, 383, 402–404, 438
- application, 370
  - apps, 375
  - auto-update, Microsoft Store App, 26
  - cloud-based hosting environment, 371
  - components and features, 374
  - dashboards, 380
  - development work, 27
  - endorsement, 167
  - features, 374
  - flexibility, installation, 27
  - Get Data interface, 144
  - hosting options, 370
  - Microsoft SQL Server Analysis Service, 158
  - Microsoft Store, 24, 25
  - organizations, 371
  - vs.* Power BI Desktop Report Server, 403, 404
  - report authoring tool, 24
  - reports, 161, 372
  - role creation, 189
  - as separate download link, 26
  - settings and configurations, 381
  - sharing content, 372
  - Task Manager, 158
  - task threads, 157
  - template apps, 375
  - users and authentications, 371
  - workspaces, 376
- Power BI Embedded, 20, 21, 439, 442, 443, 464, 492, 567, 575, 579–585, 587–589, 593–595, 599, 606, 608, 613, 621–624 (*See also* Embedded Power BI)
- Power BI Embedded licensing, 442–444, 583
- Power BI Free, 28, 234, 437, 439, 445, 548, 605, 617
- Power BI Helper, 30, 193, 491, 493, 502, 512, 518, 522, 705–745
- Power BI Premium, 20, 40, 49, 50, 118, 125, 175, 177, 340, 364, 413, 414, 442, 444–446, 533, 567, 607, 654, 656
- Power BI Premium licensing, 40, 444
- EM categories, 444
  - features, 446
  - per node, 444
  - share content, free users, 445
- Power BI Pro license, 131, 340, 439–442
- Power BI real-time dashboard, 268
- Power BI Report, 6, 7, 234, 364, 402, 406, 407
- Power BI Report Builder, 16, 17, 230, 234–240, 705
- Power BI Report Server, 14, 15, 30, 383, 384, 406, 410, 413, 414
- advantages, 412
  - Change Database, 391
  - configurations, 397
- databases, 392, 394
- server, 395
  - setup, 396
- disadvantages, 412
- environment, 401, 412
- features, 413
- gateway, 412
- installation process, 385, 388
- installation steps, 388
- network drive path, 409
- portal URL, 405
- and the Power BI Desktop Report, 384
- publish, 405
- Report Server, 396
- Scheduled Refresh feature, 411
- setting up, 385
- SQL Server Analysis Services, 413
- SQL Server database, 393
- SQL Server database engine, 387
- Temp database, 395
- Web Service URL, 397
- Power BI Service, 78, 84, 140, 193, 350, 373, 378, 382
- dataset size, 39
  - Manage Personal Storage, 38, 39
- PowerPivot model, 256, 257
- Power Platform Dataflows, 130, 133, 151, 154, 155, 251, 252
- Power Platform toolset, 265
- Power Query, 204, 217, 359
- Power query, data transformation
- Get Data and Transform, 5
  - M scripting language, 5, 6
  - Power BI Desktop, 5
  - Power Query Editor, 5
- Power Query Editor, 5, 42, 84, 96, 133, 134, 146, 147, 149, 150, 155, 181, 183, 184, 285, 344, 360, 506, 507, 685
- Power Query Editor online, 133, 134, 146, 147, 150, 155, 181, 184
- Power Query engine, 129, 155, 377
- Power query functions
- columns, 361
  - custom functions, 360
  - data transformation, 360
  - parameters, 360
  - Power BI dataset, 360
  - steps, 360
- Power Query online, 129, 133, 134, 179–182
- Power Query parameters, 340, 360, 361
- Power Query transformations, 41, 67, 74, 99, 124, 125, 133, 136, 339, 340, 360
- PowerShell, 512
- cmdlets
  - connect-PowerBIServiceAccount Cmdlet, 476
  - CSV file, workspace organization, 478, 479

- deployment pipeline solutions, workspace
  - ID, 479–482
  - install module, 474
  - ISE, 475
  - ISE, open, 472, 474
  - login Power BI account, 476
  - REST APIs, 483
  - workspace, 477
- easy-to-use tool, 471
- modules, 471, 472
- scripts, 265, 452
- Premium license, 154, 350, 442, 444, 445, 656, 681
- Premium Per User (PPU), 40, 175, 441, 442
- Premium workspaces, 175, 177, 349, 445, 540, 564, 618, 676
- Publish to Web**
  - confidentiality warning, 569
  - embed codes, central monitoring, 574, 575
  - embedded code, 567
  - open public link, web browser, 570
  - removing access, 572–574
  - report, 568
  - security issues, 570–572
  - share public data, 567
  - sharing dashboards, 575–578
- PubNub, 262, 270
  
- Q**
- Query reduction, 60–62
  
- R**
- RangeStart and RangeEnd parameters, 340–344, 348, 349, 356
- Real-time streaming datasets
  - Azure Stream Analytics, 270
  - DirectQuery, 277
  - hybrid dataset, 269
  - live connection, 274–276
  - vs.* Live/DirectQuery, 259, 260
  - Power Automate, 265–267
  - Power BI dashboard, 268
  - Power BI streaming dataset, 261–265
  - PubNub, 270
  - push dataset, 269
  - real-time report/dashboard, 276
  - salary survey, 260, 261
  - streaming dataset, 269
  - streaming scenario, 260
  - streaming services, 270
    - with Power BI REST API, 270
  - user's input, 260
  - writing calculations using Q&A, 270–274
- Reference and duplicate, 361
- Report Builder, 235–237
- Reporting Services Report Builder, 16, 235
- Report-level measures, 76–78, 100, 106, 121, 122, 124, 125, 710
- Report Server's page, 399
- Report Service setup, 398
- REST API, 20, 681
  - browser function, 488
  - calling API from .NET, 490, 491
  - custom web application, 492, 493
  - definition, 485
  - examples, 485
  - functions, 487
  - operation groups, 485, 486
  - Power Automate, 493
  - PowerShell command, 493
  - sample results, web browser, 489
  - service URL, workspace ID, 490
- Reusability, 169, 170
- Reusing components, 361
- Row-level security (RLS)
  - dynamic, 693
  - Jack, 683
  - sample data, 684
  - static, 684
  
- S**
- Sales Agg table, 289–302, 310, 314, 316, 321–323, 325, 331
- Second aggregation layer, 318–321
- Secure embed
  - advantages, 593, 594
  - features, 589
  - file menu, 589
  - licensing paths, 593
  - limitations, 594
  - methods, 587
  - Power BI embedded, 588
  - publish to web, 587
  - service, 589, 590
  - share data, 594, 595
  - view report, 590–593
- Self-service BI, 102
- Self-service users, 99, 101
- Server setup process, 400
- Setting up parameters, 342
- Shared dataset, 8
  - certified and promoted, 165–167
  - creation, 162
  - external dataset, 163
  - multiple reports, 159
  - multiple workspaces, 162
  - Power BI architecture, 164

- Shared dataset (*cont.*)  
 Power BI environment, 161  
 Sales.PBIX file, 163  
 thin reports, 159–161
- SharePoint Online  
 advantages, 608  
 definition, 597  
 disadvantages, 609  
 embed reports, 597, 600–603  
 Microsoft PowerPoint, 610, 611  
 Powe BI report  
 access page, 604, 605  
 managing permissions, 605–608  
 teams, 609, 610
- Sharing methods  
 basic sharing, dashboards/report, 614, 615  
 comparing method, 624  
 Power BI apps, 617–619  
 Power BI embedded, 622, 623  
 Publish to Web, 619, 620  
 secure embed, 623, 624  
 SharePoint Online page, 620–622  
 types, 613  
 workspaces, 615–617
- Slowly Changing Dimension (SCD), 175
- SQL Profiler, 55–57, 300, 301, 316  
 query tracks, 317  
 T-SQL query track, 301, 302
- SQL Server, 283  
 database, 78, 101, 138, 296, 301
- SQL Server Analysis Services (SSAS), 69, 70, 74, 514  
 data model, 75  
 dataset, 100  
 Multi-Dimensional, 84  
 Tabular, 7, 30, 69, 73, 78, 84  
 Tabular database, 81  
 Tabular instance, 69
- SQL Server Data Tools (SSDT), 514, 518
- SQL Server Enterprise Edition licensing, 447, 448
- SQL Server Management Studio (SSMS), 70, 170,  
 173, 193, 194, 350–352, 514, 515, 517, 518,  
 520, 521
- SSRS Report Builder, 235
- Staging database, 211, 657
- Standard *vs.* analytical dataflows, 132, 133
- Star-schema-designed model, 365
- Static row-level security  
 assigning users to roles, 690–692  
 build roles, 686, 688  
 creating report, 684, 685  
 desktop, testing roles, 688–690  
 example, 684  
 testing roles, 692, 693
- Storage modes, 94  
 combination DirectQuery and Import, 299–302
- DirectQuery, 296  
 Import Data, 295  
 Power BI tables, 293  
 properties, 293–295  
 Query, Multiple DirectQuery Tables, 297, 298  
 query sent to data source, 297, 299  
 table, 293
- Streaming dataset, 8, 259–263, 268–278, 441, 485,  
 492, 584
- Syncing slicers, 362
- **T**
- Tenant objects, documenting  
 configuring documentation output, 732–735  
 PBIX files, 722  
 scan service objects, 730–732  
 settings, 723–730
- Tenant, Power BI, 179
- Tenant settings  
 Admin Portal, 452, 453  
 audit/usage, 463  
 content pack, 459  
 custom visuals, 462  
 developer, 464  
 export/share  
 certifications, 459  
 copy/print/integrate data, 458  
 external users, 456  
 publish to web, 457  
 help/support, 453, 454  
 information protection, 455, 456  
 integration settings area, 460, 461  
 workspaces, 455
- Thin reports, 159–161, 206
- Transformation dataflows, 212, 213, 228, 657–659
- T-SQL queries, 173, 185, 186, 285
- **U**
- UPN Mapping, 81–83
- User-based licensing  
 options, 439  
 Power BI Free, 439  
 Power BI PPU, 441, 442  
 Power BI Pro  
 design, Power BI developer, 441  
 developer functions, service, 440, 441  
 share, 439  
 user subscription, Power BI, 439  
 workspaces, 441
- username() function, 696
- UserPrincipalName() function, 695–697, 702
- Users and access controls, gateways  
 data source users, 431, 432

- gateway admin, 431
  - Gateway Connection
    - creator, 431
    - creator with reshare, 431
  - Manage Gateway Installers, 429, 430
    - settings, 432–434
  - Users, Power BI
    - access control, 626
    - groups, 629
    - levels of training, 627
    - move between layers, 629
    - organization, 629
    - types, 625, 626
- V**
- Visualization, 6, 8, 9, 49, 51, 55–58, 61, 67, 69, 73, 78, 95, 99, 118–121, 125, 158, 163, 210, 215, 218, 222, 223, 240, 253, 272, 305, 307, 317, 362, 722, 740, 743
- W**
- Web portal setup, 401
  - Web Portal URL, 399, 400, 405
  - Web UI, 171, 173, 175, 186, 225
  - Workspaces, 18, 19
    - adding content, 541–543
    - advantages, 547
    - capacity settings, 654
    - creating
      - advanced settings, 539
      - login to service, 536
      - new workspace created, 540
      - premium capacity settings, 540
      - setting up general settings, 538
    - dev/user environment, 548
    - disadvantages, 547, 548
- edit/read-only access, 548
  - layers, 655
  - levels of access, 543
  - multiple developers, 654, 655
  - roles, 631, 632, 638, 640
    - admin, 637, 640
    - contributor, 633, 634, 639
    - individual accounts, 640
    - member, 634–637, 639
    - viewer, 632, 633, 638
  - Power BI content, 535, 536
  - separate environments, 656–660
  - separating developers, 652, 653
  - share content, 535
  - sharing developers/data analysts, 652
  - single development-sharing unit, 651
  - split load, 654
  - tenant administrator, 544, 546
  - version 1, 631
- X, Y, Z**
- XMLA Endpoint
    - Admin Portal, 519, 520
    - client tool, 520
    - connections, 518
    - dataset, 513, 514
    - definition, 517
    - Power BI dataset, SSAS, 516, 517
    - Power BI Helper, 522
    - sample connection, 521
    - SSAS, 514–516
    - tools, 518
    - URL, 519
  - xVelocity, 36
    - compresses data, 36, 37
    - compression engine, 118
    - DAX, 43