



# Power BI Enterprise Project Good and Best Practices

For designing scalable & manageable solutions



Microsoft®  
Most Valuable  
Professional



**Award Categories**

Data Platform

**First year awarded:**

2009

**Number of MVP Awards:**

10



[pturley@pragmaticworks.com](mailto:pturley@pragmaticworks.com)



[linkedin.com/in/pturley](https://www.linkedin.com/in/pturley)



[SqlServerBi.blog](http://SqlServerBi.blog)



[@Paul\\_Turley](https://twitter.com/Paul_Turley)

# Paul Turley

Principal Consultant, Pragmatic Works

**Pragmatic  
Works**

# Objective

Set of guidelines that address many aspects of new projects.

Recommendations are applicable in most use cases.

Living document with ongoing revisions with the goal to provide a comprehensive, best practices guide as the platform continues to mature and as experts continue to use it.

[sqlserverbi.blog/2019/08/24/power-bi-project-good-and-best-practices](https://sqlserverbi.blog/2019/08/24/power-bi-project-good-and-best-practices)

The screenshot shows a web browser displaying the article 'Power BI Project Good and Best Practices' by Paul Turley on the website sqlserverbi.blog. The article is dated August 24, 2019, and is marked as 'FEATURED'. Paul Turley is identified as a Power & Data Platform MVP, Principal Consultant, and Pragmatic Works. The article's preface states that it is a living document of guidelines and best practices for Microsoft Power BI projects, which are updated as the platform evolves and experts share their experiences. It mentions revisions from 8/25/2019 and 9/7/2019. The article's purpose is to provide a checklist of guidelines and recommended practices for managing Power BI projects. A sidebar on the left lists navigation options like Home, My Books, Video Tutorials, Presentations, Training & Consulting Services, Visualizations, About/Bio, and Paul's Bio. The right sidebar features a search bar, a world map showing 18,100 pageviews from August 21st to September 21st, and an email subscription form. The Pragmatic Works logo is visible in the bottom right corner of the page.

---

# This Session is:

- NOT a 100-level introduction to Power BI
- NOT a technical “how-to” deep dive but we will discuss some technical topics
- Guidelines for experienced Business Intelligence & Power BI practitioners
- Some good knowledge for less-experienced Power BI developers

---

# So Here's the ~~Problem~~: Opportunity:

- I want you to ask questions
- You need to understand some basics before much of this will make sense
- The topics covered in this session could be covered thoroughly in one or two weeks
- If the answer to your question isn't simple or within the context of this presentation, I may address it offline
- We can talk after the session

# Bad, Good and Best Practice

## Bad Practice

Better practice exists.  
Has negative outcome under certain conditions.  
Avoid making this mistake.

## Good Practice

A clear best practice may not exist.  
There may be multiple "good" options.  
Choice may be a matter of preference or opinion.

## Best Practice

Clearly the best choice.  
Technique or feature is well understood and accepted in the industry.  
Always do this for best outcome.

---

# Project Types

- **Formal projects** are scoped, funded, staffed and executed with the collaboration of a business champion and stakeholders; and IT Business Intelligence developers and data managers. These projects promote business and IT-governed datasets and certified reports.
- **Informal projects** are executed by business users and are considered ad hoc in nature. Datasets are generally not IT governed, and reports are typically not certified.
- **Hybrid projects** can be anything in-between. They might be a user-authored report using published, certified dataset used for self-service reporting. Informal, self-service datasets can be migrated to governed datasets in collaborative IT/business projects.

---

# IT-managed BI Solution Developers

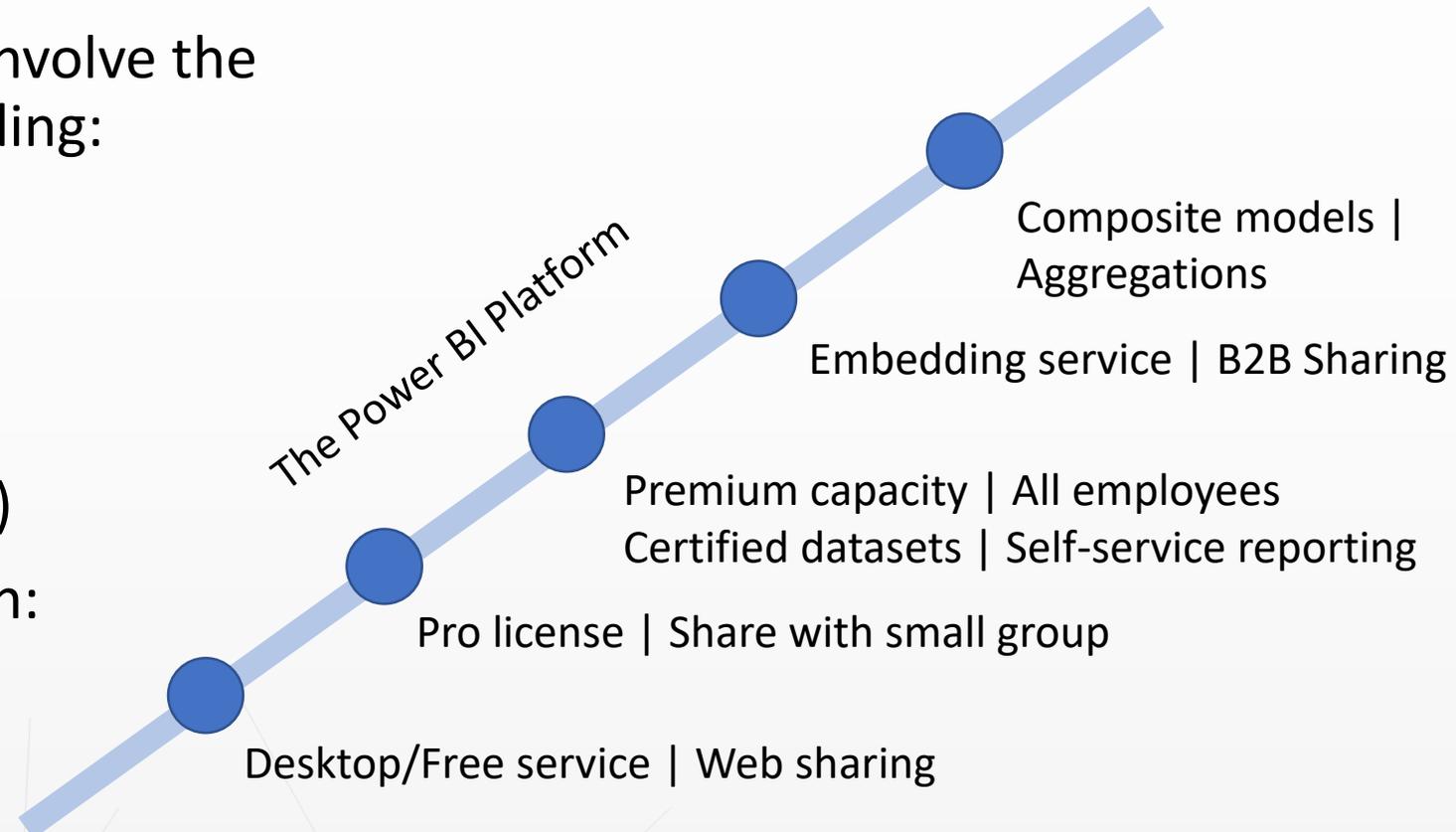
## Formal project teams might include:

- Business Systems Analysts
- BI Solution Architects
- Project Managers
- Data Transformation (ETL) Developers
- Enterprise Data Modelers
- Measure code experts (using DAX)
- Report Developers
- Operations & Deployment Engineers

# Solution Architecture

All Business Intelligence projects involve the same essential components including:

- Source Queries
- Data transformation steps
- Semantic data model
- Calculations (typically measures)
- Data visualizations that reside on:
  - reports
  - report pages
  - dashboards



# Managing Power BI Desktop Files

- **Store in a centrally managed network-assessable folder**  
The storage folder should support automatic backup and recovery in the case of storage loss.
- **Report and dataset developers must open files from the Windows file system**  
Files must either reside in or be synchronized with the Windows file system.
- Files containing imported data typically range in size from 100 to 600 MB. Any shared folder synchronization or disaster recovery system should be designed to effectively handle multiple files of this size.

## Options:

- OneDrive For Business (shared by team, with folder synchronization).
- SharePoint or SharePoint Online (with folder synchronization).
- GitHub and/or VSTS with local repository & folder synchronization. If used, Git must be configured for large file storage (LFS) if PBIX files are to be stored in the repository.

---

# Datasets and Reports

## Single PBIX File

## SSAS with Power BI

## Power BI Datasets

- Generally can have the same or more features and capabilities than SSAS data models.
- In Premium capacity, Power BI datasets can perform and scale as well as SSAS data models with up to 3 or 12 GB of compressed data, depending on the capacity level.
- SSAS & AAS requires additional licensing or usage fees. These are included with Power BI licensing when using Power BI datasets.
- Without Premium capacity, Power BI datasets in shared capacity may not perform as well as SSAS with more than 8-10 concurrent users, or during peak usage in the shared capacity service.
- SSAS or AAS ensures consistent capacity and performance without Premium dedicated capacity, so it might make more sense to use SSAS or AAS when Power BI Premium capacity isn't cost-effective or with very large data volumes that exceed Premium capacity limits.

# Version Control & Lifecycle Management

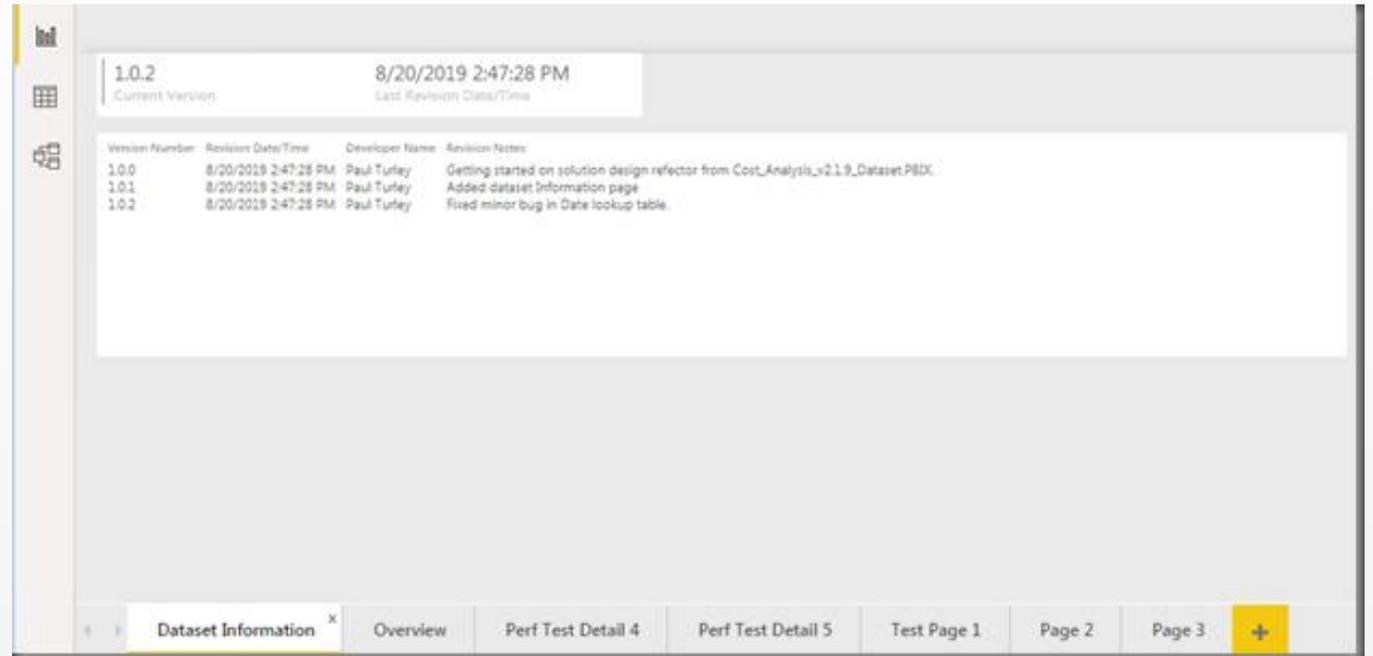
- At the present time, Power BI has no integrated version control mechanism. In addition, Power BI is incompatible with most version control systems that perform branching and differencing operations on code and script files that are common in application development projects. Power BI (PBIX and PBIT) report and dataset files are stored as binary archive files with internal checksum metadata. Change and differencing operations performed by most code and version management systems will corrupt these files and render them inoperable.
- **Reality:** Integrating changes made to multiple PBIX files is difficult to manage and challenging to perform technically. Even with differencing tools and effective version-control, just avoid multi-developer work on Power BI dataset files.
- Leverage Power BI's simplicity and don't over-engineer release management or version control. Until tooling exists to automate deployments, keep this as simple as possible. If deployment must be automated, use PowerShell to deploy datasets and reports to a workspace from a file system folder or staging workspace in the service.
- **File versions**
- Store the dataset and report PBIX files separately using a version number postfix for the file name in the following the format: **Major.Minor.Revision**. For example:

**Manufacturing Cost Analysis Dataset v1.5.3.PBIX**

# Version Control & Lifecycle Management

Add a **Version History** table to the data model by using the “Enter Data” feature in Power Query to create a table containing these columns:

- Version Number
- Revision Date/Time
- Developer Name
- Revision Notes



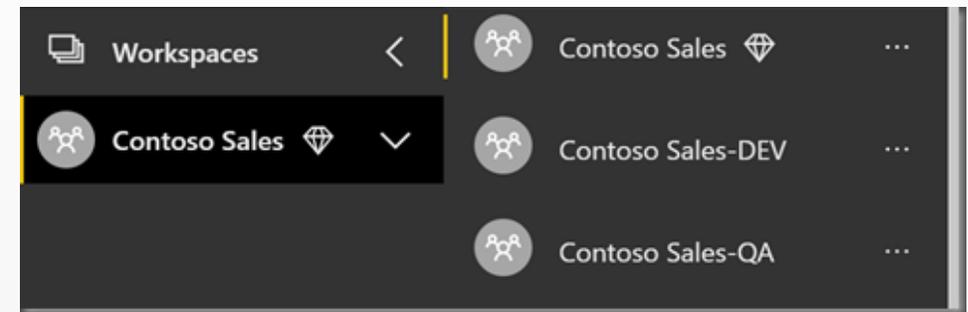
Version Number	Revision Date/Time	Developer Name	Revision Notes
1.0.0	8/20/2019 2:47:28 PM	Paul Turley	Getting started on solution design refactor from Cost_Analysis_v2.1.9_Dataset PBD.
1.0.1	8/20/2019 2:47:28 PM	Paul Turley	Added dataset Information page
1.0.2	8/20/2019 2:47:28 PM	Paul Turley	fixed minor bug in Date lookup table.

# Workspace and App Management

- Power BI platform currently lacks a formal process or mechanism
- Existing tools don't work (VSTS and Azure DevOps)
- Release management tools are being developed but in the early stages

For a formal project, create the following workspaces:

- **DEV Workspace** - Only development team members need Contributor access to this workspace. This workspace does not need to have Premium capacity; unless, developers need to unit test incremental refresh or other Premium features.
- **QA Workspace** - All testers must have View access for testing and Contributor access for report authoring. Should be in Premium capacity to test incremental refresh.
- **PROD Workspace** - Omit the "PROD" designation in the name. This workspace will be the name of the published app that users will see in their Apps, Home and Favorite pages so use a name that is simple and sensible. Must have Premium capacity to share the app with non-Pro licensed users.

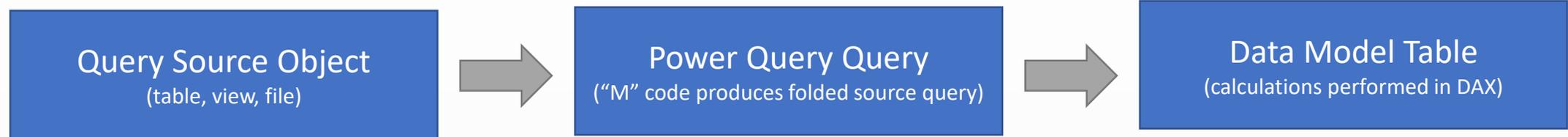


PowerShell script may be used to publish datasets and reports, and to change dataset bindings. It is possible to either publish to a production workspace or to effectively move assets from one workspace to another. This approach is discussed briefly in the [Power BI Enterprise Deployment Guide](#). Other approaches are discussed here: [Power BI release management](#)

# Dimensional Design

- **Build star schemas** – wherever possible, reshape data into fact a dimension tables with single key, one-to-many relationships from dimensions to fact. The art of dimensional modeling ranges from simple to complex. Refer to The Kimball Method for details: <https://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/>
- **Enforce dimension key uniqueness** – Just because a key value “should” be unique, there is no guarantee that it will be unless enforced at the data source. Perform grouping and duplicate reduction in the data source views or Power Query queries to guarantee uniqueness. Duplicate record count checks and other mechanisms can be applied to audit source data for integrity but do not allow the data model to violate these rules.
- **Avoid bi-directional filters and unnecessary bridging tables** – These data modelling patterns adversely affect performance.
- **Calculated columns** – Should be applied in Power Query and not in DAX calculated columns wherever possible. This maintains a consistent design pattern for maintainability.
- **Annotate code** – Use in-line comments and annotations in all code including SQL, M and DAX; to explain calculation logic and provide author and revision information.
- **Hide all fields not used directly by users.** These include: primary and foreign key columns, numeric columns used to create measures, and columns used to specify the sort order of other fields.
- **Set to Do Not Summarize** – Any non-hidden numeric columns that are not intended to roll-up or summarize values should be set to “Do Not Summarize” in the Modeling ribbon in Power BI Desktop. Columns set to summarize are indicated with a Sigma icon.

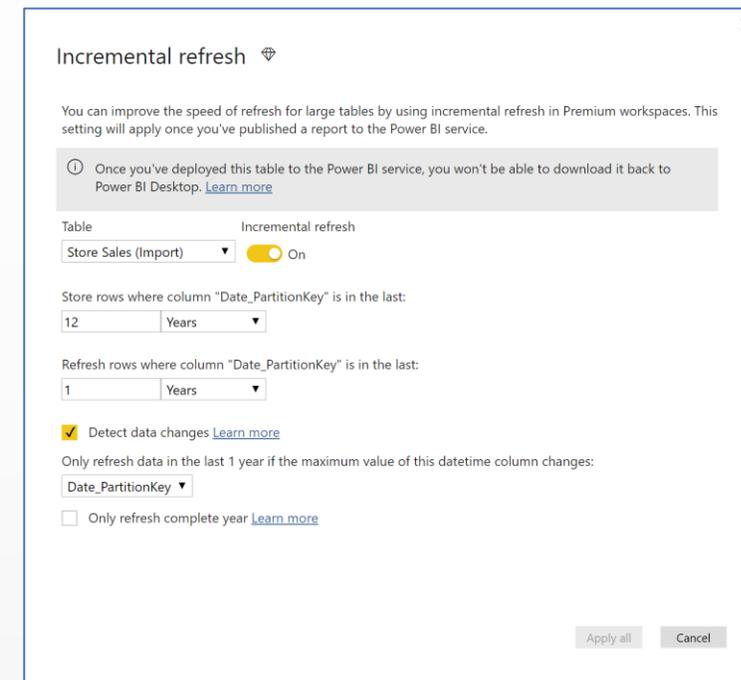
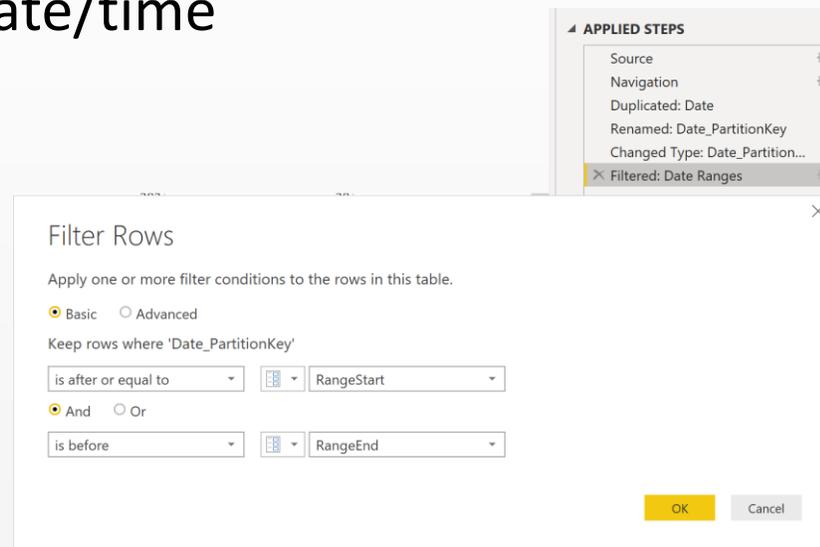
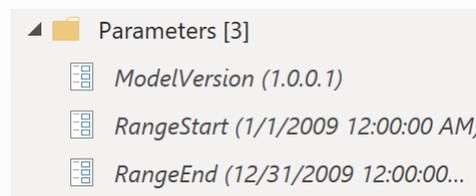
# Query Optimization



- Avoid using SQL statements in PQ queries. Use database views. Views and tables support query folding. SQL statements do not.
- Perform column transformations in view or PQ? – use maintains the query?
- Remove unnecessary columns & filter rows early in the query
- Consolidate field renaming, removing fields and data type changes
- Add custom columns in Power Query instead of calculated columns in DAX, where possible
- Use friendly field names for all fields that won't be hidden in the data model
- Rename steps and add annotations in M script

# Managing Dataset Size with Parameters

- Use parameters whether implementing incremental refresh or not
- **RangeStart & RangeEnd** parameters must be date/time type
- Apply range filter on date/time column in Power Query



\*Incremental Refresh is a Premium feature

# Implicit and Explicit Measures

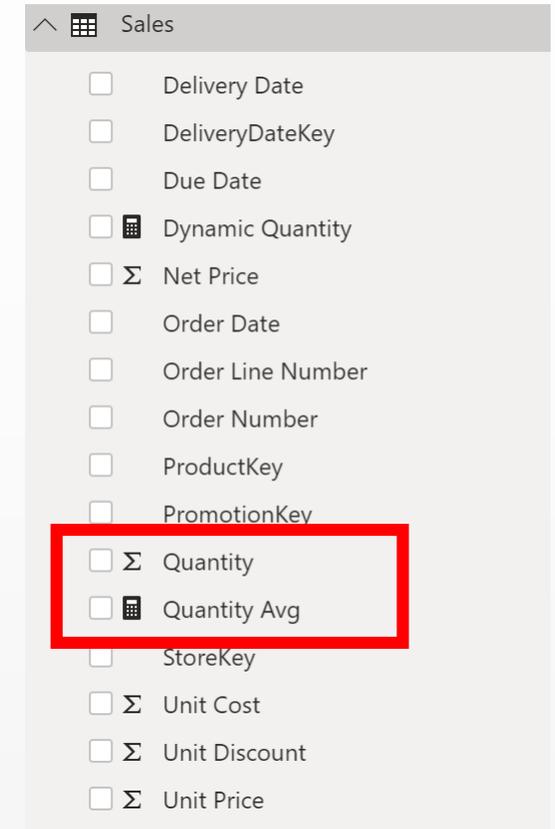
## Implicit measure

= numeric field with default summarization

## Explicit measure

= Defined using DAX expression

- Implicit measures are typically OK in informal projects
- Measures should be explicitly defined in formal data models
- Implicit measures don't work in some client tools



---

# Promote Self-service Reporting

## Non-governed Data

- Teach & support analyst users to use Power BI to acquire, mashup & model data
- “make mistakes, get messy”
  - Lilly Tomlin, Miss Frizzle
- Deploy to “user” designated workspaces
- User-authored solutions be used to prototype & pattern governed data models

## Governed Data

- Separate datasets from reports
- Publish to a secured & managed workspace
- Promote & Certify datasets
- Use dataflows for standardized common data models
- Enable users to connect to published datasets & create their own reports

---

# Operational & Paginated Reports

- Power BI is not a replacement for paginated, operational reporting
- For static, multi-page, printable reports; use SQL Server Reporting Services (SSRS) instead of Power BI
- SSRS (now called “Paginated Reports”) is integrated into the Power BI service with Premium capacity licensing and can be integrated with interactive Power BI reports and Power BI data datasets
- To a limited degree, some operational reports can be reproduced using Power BI reports and SSRS can be used, some a limited degree, to create interactive reports

# If Users Need Excel, Give them Excel

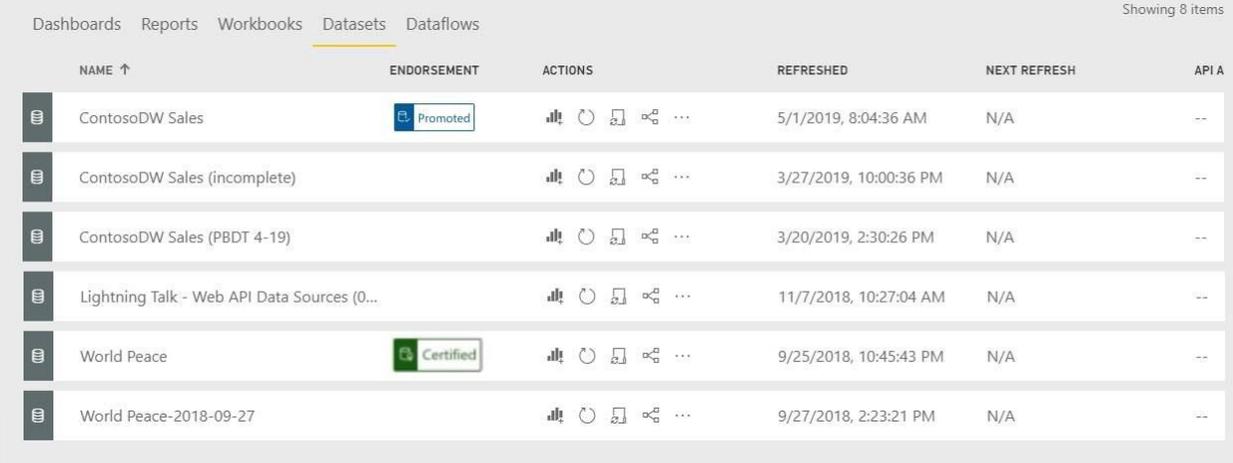
- Teach analyst users how to use Excel with Power BI
- Don't "export" ... "connect"
- "Analyze In Excel" allows Excel to connect, live, to a published Power BI dataset
- Available to Power BI Pro licensed users
- Now available to "free" licensed users in a Premium

The screenshot shows the Excel interface with a PivotTable. The PivotTable is set to show fields: Delivery Date. The data is summarized by Country (Row Labels) and Year (Column Labels). The PivotTable Fields task pane on the right shows the data source and field placement.

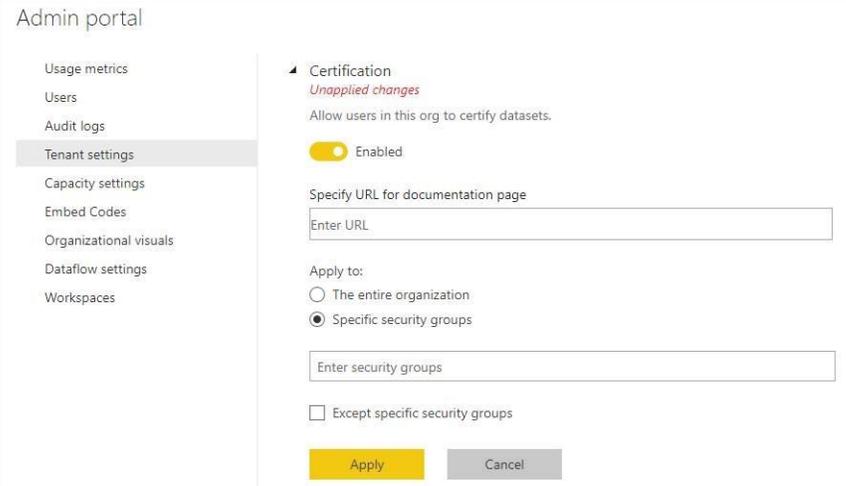
Row Labels	CY 2007	CY 2008	CY 2009	CY 2010				
Qty Sum	Quantity Avg	Qty Sum	Quantity Avg	Qty Sum	Quantity Avg	Qty Sum	Quantity Avg	
Canada	1829	1.42	1118	1.41	235	1.43	29	1.16
France	2513	1.41	2272	1.40	1440	1.40	13	1.18
Germany	6147	1.40	3131	1.39	2319	1.42	31	1.82
Greece			105	1.48	614	1.39		
Ireland					768	1.38		
Italy					781	1.38		
Poland					662	1.40	303	1.43
Portugal					686	1.41		
Russia					1582	1.40		
Spain					584	1.40		
Switzerland			21	1.31	1056	1.41		
the Netherlands			114	1.44	725	1.40		
United Kingdom	6260	1.40	4683	1.39	5192	1.40	11	1.00
United States	14062	1.40	14239	1.40	18209	1.40	574	1.46

# Certified & Shared Datasets

- Use Dataset endorsement & certification in the service
- Certification can be managed by security group
- Access to datasets can be restricted to certified datasets
- Organization defines certification policy & provides documentation



NAME ↑	ENDORSEMENT	ACTIONS	REFRESHED	NEXT REFRESH	API A
ContosoDW Sales	Promoted	  	5/1/2019, 8:04:36 AM	N/A	--
ContosoDW Sales (incomplete)		  	3/27/2019, 10:00:36 PM	N/A	--
ContosoDW Sales (PBDT 4-19)		  	3/20/2019, 2:30:26 PM	N/A	--
Lightning Talk - Web API Data Sources (0...		  	11/7/2018, 10:27:04 AM	N/A	--
World Peace	Certified	  	9/25/2018, 10:45:43 PM	N/A	--
World Peace-2018-09-27		  	9/27/2018, 2:23:21 PM	N/A	--



Admin portal

- Usage metrics
- Users
- Audit logs
- Tenant settings
- Capacity settings
- Embed Codes
- Organizational visuals
- Dataflow settings
- Workspaces

Certification

*Unapplied changes*

Allow users in this org to certify datasets.

Enabled

Specify URL for documentation page

Enter URL

Apply to:

The entire organization

Specific security groups

Enter security groups

Except specific security groups

Apply Cancel

---

# Enterprise Scale Options

In many ways, Power BI has now surpassed the capabilities of SQL Server Analysis Services. Microsoft are investing in the enterprise capabilities of the Power BI platform by enhancing Power BI Premium Capacity, adding Paginated Report and features to support massive scale specialized use cases. Consider the present and planned capabilities of the Power BI platform; before, choosing another data modeling tool such as SSAS.

## Resources:

<https://sqlserverbi.blog/2018/07/27/power-bi-for-grownups>

<https://sqlserverbi.blog/2018/12/13/data-model-options-for-power-bi-solutions>

# Project Preparation Checklist

## Solution Audience:

- Categorize the solution by identifying the author & user roles related to the project:
- Author role: Business Data Analyst
- Author role: Skilled Data Modeler, Analyst, Data Scientist
- Author role: IT BI Developer
- Users' role: Report/Dashboard Consumer
- Users' role: Self-service Report Author
- Users' role: Advanced Data Analyst

## Training and Usability Support:

- Develop & Document Support & training plan for users:
  - Usability training for read-only report/app users
  - Self-service reporting for Novice Report Authors & Data Analysts
- Identify the Solution Type for the project. This will guide other project management designs:
- Design single PBIX file for small group, departmental project authored by one developer for a limited group of users
- Design & deploy a separate dataset PBIX file – from report file(s) – when the dataset should be branded as a Certified dataset

- Design separate dataset and report PBIX files for formal projects with more than one dataset & report developer, to coordinate work
- Use SSAS/AAS as a data modeling option when those databases exist or where IT operations insist on management development and maintenance through integrated source control (e.g. Visual Studio Team Services & Azure DevOps)
- Identify the Project Type & related Solution Architecture:
  - Project type: Formal project
  - Project type: Informal project
  - Project type: Hybrid project
  - Architectural approach: Single PBIX
  - Architectural approach: Separate dataset and report PBIX
  - Architectural approach: Report PBIX connected to SSAS or AAS
- Understand DirectQuery model trade-offs and special use cases. Avoid if possible.
- Create storage locations and folder structure for Development file management:
  - Development file storage
  - Team member collaboration environment & processes
  - Folder synchronization

- Define File naming standards
- Decide on dataset and report names
- Define the Version Control & Lifecycle Management:
  - Postfix files with 3-part version number
  - Remove version number from published files in QA and PROD
  - Create Version History table in Power Query
  - Increment version numbers in data model
  - Backup PBIX files for archive
  - Create measures: Last Refresh Date/Time
  - Create measure: Current Version
  - Add data model info page to report
- Define your Release Management, DevOps & Automation strategy (if any – Might be OK to deploy files manually) (to automate or not to automate)
- Decide on Workspace and App Management, workspace & app name, etc.:
- Create PROD workspace (omit PRD from name), assign dedicated capacity if available
- Create QA workspace (post-fix name with QA), assign dedicated capacity
- Create DEV workspace (postfix name with DEV), dedicated capacity not required

## Assign licenses and access:

- Assign Pro licenses to all developers, admins and report author users (QA?)
- Assign Free licenses to all users if Premium/app deployment will be used
- Assign membership and access to workspaces

## Query Design:

- Create fact date range filter parameters: RangeStart & RangeEnd
- Filter large fact tables with range filters, consider incremental refresh policies if slow and/or over 800 MB compressed.
- Design source queries (T-SQL?) to reshape source data into conformed dimension & fact tables
- Create views in database for each dimension and fact
- Enforce key uniqueness to remove all duplicate keys from all dimension tables
- Query Date dim/lookup table at source if it exists
- If not available, generate Date dim/lookup table in Power Query

# Project Preparation Checklist

## Data Model Design:

- Avoid bi-directional relationship wherever possible
- Include code annotations & comment blocks in: views, M queries, measures
- Hide all key columns and other columns not used directly by users
- Use parameters to filter and reduce dataset PBIX size to ~400 MB or less
- Create incremental refresh policies for dataset larger than ~500 MB
- Use Premium capacity workspaces for datasets approaching 900 MB
- Use Premium capacity workspace when you need to share reports with non-Pro licensed user
- For non-Pro licensed user, convert QA (and then PRD) workspace to app and distribute the app to users via AD group membership
- Create explicit measures for all aggregate-able column values

- Hide all measure base numeric columns
- Set all non-aggregate-able numeric column default summarization to "Do Not Summarize"
- Format add whole numbers with thousand separators
- Format all currency & decimal measures to defined standard (perhaps 2 decimals, thousand separator)
- Format all ratios and percentage with appropriate decimal positions
- Use the DIVIDE function for all division operations to avoid div by zero
- Name all measures and user-visible fields with mixed-case, short, friendly sentence-like terms
- Assign measures to fact table where they logically belong – or to an empty measure group table.
- Measures that don't have an obvious home (fact) table should be assigned a measure group table.

- Precede measure group table names with "\_" so they sort to the top of the list. (display folders are not ready for use in Power BI Desktop)

## Project Discipline (as a team decide):

- When to increment the revision/fix number
- When to increment the minor version number
- Who owns the migration to QA process? How is it implemented?
- Who own migration from QA to PROD? How is it implemented?
- How is user/stakeholder sign-off obtained?
- How are data source requirements defined, presented & documented?
- Who owns & maintains the requirements log?
- How are measure & KPI calculation requirements documented & maintained?
- How are the report design, layout &

visualization requirements documented & maintained?

## Misc:

- What is the organization & project styling & branding standard?
- Does the organization have a standard brand image? Where is it documented? How owns that standard & can answer questions?
- Should report be designed for mobile consumption (with mobile layouts)?
- Utilize dataset endorsements to categorize Shared, Promoted and Certified datasets to manage data governance expectations for self-service report users



# Thank You



[pturley@pragmaticworks.com](mailto:pturley@pragmaticworks.com)



[linkedin.com/in/pturley](https://www.linkedin.com/in/pturley)



[SqlServerBi.blog](https://sqlserverbi.blog)



[@Paul\\_Turley](https://twitter.com/Paul_Turley)

---

Please connect with me using one of these mediums

**Pragmatic  
Works**