Bulk-extracting Power Query M-code from multiple pbix files in Power BI

If you want to audit or analyse the M-code of multiple Power BI pbix-files at once, you start with either:

1. a from-folder query where you filter all files of interest or
2. a table with the full file-path-specification of the files to be analysed in “Column1”.

Then you add a column where you call the function that extracts the M-code:

Function to extract the M-code

```
(Filename as text) =>

let

// Unz-function from: https://querypower.com/2017/03/22/extracting-power-queries-in-m/
Unz =  (binaryZip,fileName) =>
let
//shorthand
    UInt32 = BinaryFormat.ByteOrder(BinaryFormat.UnsignedInteger32,ByteOrder.LittleEndian),
    UInt16 = BinaryFormat.ByteOrder(BinaryFormat.UnsignedInteger16,ByteOrder.LittleEndian),
//ZIP file header fixed size structure
    Header = BinaryFormat.Record([
                MiscHeader              = BinaryFormat.Binary(14),
                CompressedSize          = UInt32,
                UncompressedSize        = UInt32,
                FileNameLen             = UInt16,
                ExtraFieldLen           = UInt16]),
//ZIP file header dynamic size structure
    FileData = (h)=> BinaryFormat.Record([
                FileName                = BinaryFormat.Text(h[FileNameLen]),
                ExtraField              = BinaryFormat.Text(h[ExtraFieldLen]),
                UncompressedData        = BinaryFormat.Transform(
                        BinaryFormat.Binary(h[CompressedSize]),
                                (x) =>  try
                                        Binary.Buffer(Binary.Decompress(x, Compression.Deflate))
                                        otherwise null)]),
//Parsing the binary in search for PKZIP header signature
    ZipIterator = BinaryFormat.Choice(UInt32, (signature) => if signature <> 0x04034B50
                                            then BinaryFormat.Record([FileName=null])
                                            else BinaryFormat.Choice(Header,(z)=>FileData(z))),
    ZipFormat = BinaryFormat.List(ZipIterator),
    out = List.Select(ZipFormat(binaryZip), each _[FileName]=fileName)
in
    out{0}[UncompressedData],

    Source = Unz(Unz(File.Contents(Filename),"DataMashup"),"Formulas/Section1.m"),
    Custom1 = Lines.FromBinary(Source),
    #"Converted to Table" = Table.FromList(Custom1, Splitter.SplitByNothing(), null, null, ExtraValue
in
    #"Converted to Table"
```

MQueriesPBIX.pq hosted with ❤ by **GitHub**

This code is a variation of **Igors function which retrieves the code from an opened pbix-file**. So now you can apply it

For method 1 you call it like so (as it takes the full string for the file-path as its parameter):

MQueriesPBIX([Folder Path]&[Name])

And for method 2 like so:

MQueriesPBIX([Column1])

This returns a table with one row per code-line.

Function to identify query- & stepnames

The following function processes this further and adds columns with the query- & step-names for further analysis:

```
(PQTable as table) =>



let

    #"Added Index" = Table.AddIndexColumn(PQTable, "Index", 0, 1),

    #"Duplicated Column" = Table.DuplicateColumn(#"Added Index", "Column1", "Column1 - Copy"),

    #"Split Column by Delimiter" = Table.SplitColumn(#"Duplicated Column", "Column1 - Copy", Splitter

    #"Trimmed Text" = Table.TransformColumns(#"Split Column by Delimiter",{{"Part1", Text.Trim}}),

    QueryName = Table.AddColumn(#"Trimmed Text", "QueryName", each if Text.Start([Column1], 6) = "sha

    StepName = Table.AddColumn(QueryName, "Stepname", each if [Part1]="in" or [Part1]="let" or  Query

    #"Filled Down" = Table.FillDown(StepName,{"QueryName"})

in

    #"Filled Down"
```

MetaQueriesPBIX hosted with ♥ by **GitHub**

You call it within an added column again, with the name of the previously created column containing the code ("Code

MetaQueriesPBIX([Code])

This would be much easier, if we had a proper API like requested here: **https://ideas.powerbi.com/forums/265200-p**

That API would also enable us to bulk-retrieve other useful information from the file like everything about the DAX da

Enjoy & stay queryious 🙂

Share this:

Like this:

Like

Be the first to like this.

2017-10-15 · **COMMENTS 7**
Filed under: ***M***, ***Power BI***, ***Power Query***

Comments (7)

Maxim Zelensky

Neat as usual 🙂
Is the UnZip function the same as Mark White's one, or this is other variation?

Reply

Admin

Hi Maxim, yes, it's a variation 🙂

Reply

Sonali Tharwani

Thanks for sharing this! So useful. Have you had any success with extracting all the DAX from a file?

Reply

Admin

I didn't manage to extract the DAX from closed PBIX-files yet.
If your file is open, you can DMVs via DaxStudio or some M-code in the query editor to extract DAX-definitions to ext
Another option is to save your pbix as a template (pbit) or to migrate it to SSAS in Azure: This will produce a very nice
Please let me know if you need more Infos on any of the methods mentioned.
Cheers, Imke

Reply

Pingback: SSRS APIs, M Queries, Power BI Desktop and more | Guy in a Cube

didier terrien

Oh, that's great !
I cannot succeed to make Igor's solution to work. There is a problem with binary encoding.
Would you please adapt your solution to work with the opened PBIX file ?
Thanks a lot

Reply

Admin

Hi Didier,
you can use my function to access the currently opened PBIX as well.
Just remember that you will see the last saved version then.
Cheers, Imke

Reply

Leave a Reply

Enter your comment here...

Die Datenschutzbestimmungen finden Sie hier: / Please find the privacy policy here: https://wp.me/P6lgsG-Rz

← PREVIOUS POST