Power Query (M): Basics of Enumerations

There are 30 documented enumerations in M

```
= Web.Page(Web.Contents("https://learn.microsoft.com/en-us/powerquery-m/enumerations")){0}[Data]
```

	Name	Description
1	AccessControlKind.Type	Specifies the kind of access control.
2	BinaryEncoding.Type	Specifies the type of binary encoding.
3	BinaryOccurrence.Type	Specifies how many times the item is expected to appear in the group.
4	BufferMode.Type	Describes the type of buffering to be performed.
5	ByteOrder.Type	Specifies the byte order.
6	Compression.Type	Specifies the type of compression.
7	CsvStyle.Type	Specifies the significance of quotes in a CSV document.
8	Day.Type	Represents the day of the week.
9	ExtraValues.Type	Specifies the expected action for extra values in a row that contains co...
10	GroupKind.Type	Specifies the kind of grouping.
11	JoinAlgorithm.Type	Specifies the join algorithm to be used in the join operation.
12	JoinKind.Type	Specifies the kind of join operation.
13	JoinSide.Type	Specifies the left or right table of a join.
14	LimitClauseKind.Type	Indicates the features that the specific SQL dialect supports.
15	MissingField.Type	Specifies the expected action for missing values in a row that contains ...
16	Occurrence.Type	Specifies the occurrence of an element in a sequence.
17	ODataOmitValues.Type	Specifies the kinds of values an OData service can omit.
18	Order.Type	Specifies the direction of sorting.
19	PercentileMode.Type	Specifies the percentile mode type.
20	Precision.Type	Specifies the precision of comparison.
21	QuoteStyle.Type	Specifies the quote style.
22	RankKind.Type	Specifies the precise ranking method.
23	RelativePosition.Type	Indicates whether indexing should be done from the start or end of th...
24	RoundingMode.Type	Specifies rounding direction when there is a tie between the possible ...
25	SapBusinessWarehouseExecutionMode.Ty...	Specifies valid options for SAP Business Warehouse execution mode o...
26	SapHanaDistribution.Type	Specifies valid options for SAP HANA distribution option.
27	SapHanaRangeOperator.Type	Specifies a range operator for SAP HANA range input parameters.
28	TextEncoding.Type	Specifies the text encoding type.
29	TraceLevel.Type	Specifies the trace level.
30	WebMethod.Type	Specifies an HTTP method.

Enums are used to provide values to parameters in other functions

RelativePosition.Type

Article • 08/04/2022 • 2 minutes to read • 2 contributors

Feedback

Definition

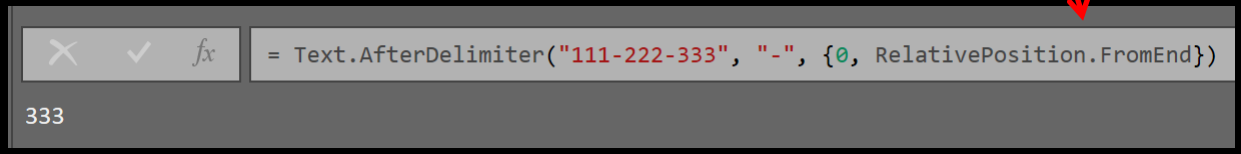
Indicates whether indexing should be done from the start or end of the input.

Allowed values

Name	Value	Description
RelativePosition.FromStart	0	Indicates indexing should be done from the start of the input.
RelativePosition.FromEnd	1	Indicates indexing should be done from the end of the input.

For example, in Text.AfterDelimiter, we can use the RelativePosition.Type enumeration to indicate that we want the delimiter index to begin from the end of the text.

This expression returns the text after the *0th-index-from-end* occurrence of "-"



Their values and names are interchangeable

When sharing your queries, prefer names so the code is more easily read and understood by others

Name	Value	Description
RelativePosition.FromStart	0	Indicates indexing should be done from the start of the input.
RelativePosition.FromEnd	1	Indicates indexing should be done from the end of the input.

```
= Text.AfterDelimiter("111-222-333", "-", {0, 1})
```

```
333
```

Wherever a named value from an enumeration is used, its equivalent 'Value' can also be used. So, these two expressions are equivalent

```
= Text.AfterDelimiter("111-222-333", "-", {0, RelativePosition.FromEnd})
```

```
333
```

We can find all the enumerations using a simple query on #shared

```
let
    Source = Record.ToTable(#shared),

    Filtered = Table.SelectRows(
                    Source,
                    each Text.EndsWith([Name],".Type") and
                        Type.Is( Value.Type([Value]) , type type )
                ),

    MetaDataRecord = Table.AddColumn(Filtered, "MetaDataRecord", each Value.Metadata([Value])),

    AllowedValues = Table.AddColumn(
        MetaDataRecord,
        "AllowedValues",
        each Value.Metadata([Value])[Documentation.AllowedValues]
    ),

    Result = Table.RemoveRowsWithErrors(AllowedValues)
in
    Result
```

Select rows where the name ends in .Type and are of type *type*

This includes all enums and other types such as type.any, type.text etc

Add a column containing a record of the type's metadata (this is the documentation)

If the type is an enum, the metadata has a field called Documentation.AlowedValues which contains a list of numbers allowed for this enum. If the type is not an enum, it doesn't have that field and this expression returns an error

So we can quickly inspect the documentation for the enumeration

	Name	Value	MetaDataRecord	AllowedValues
4	JoinKind.Type	Type	Record	List
5	MissingField.Type	Type	Record	List
6	GroupKind.Type	Type	Record	List
7	RoundingMode.Type	Type	Record	List

Specifies the kind of join operation.

Allowed values:

- `JoinKind.Inner`: A possible value for the optional `JoinKind` parameter in `Table.Join`. The table resulting from an inner join contains a row for each pair of rows from the specified tables that were determined to match based on the specified key columns.
- `JoinKind.LeftOuter`: A possible value for the optional `JoinKind` parameter in `Table.Join`. A left outer join ensures that all rows of the first table appear in the result.
- `JoinKind.RightOuter`: A possible value for the optional `JoinKind` parameter in `Table.Join`. A right outer join ensures that all rows of the second table appear in the result.
- `JoinKind.FullOuter`: A possible value for the optional `JoinKind` parameter in `Table.Join`. A full outer join ensures that all rows of both tables appear in the result. Rows that did not have a match in the other table are joined with a default row containing null values for all of its columns.
- `JoinKind.LeftAnti`: A possible value for the optional `JoinKind` parameter in `Table.Join`. A left anti join returns that all rows from the first table which do not have a match in the second table.
- `JoinKind.RightAnti`: A possible value for the optional `JoinKind` parameter in `Table.Join`. A right anti join returns that all rows from the second table which do not have a match in the first table.

Unfortunately, the AllowedValues list only shows the value and not the name

	Name	Value	MetaDataRecord	AllowedValues
4	JoinKind.Type	Type	Record	List
5	MissingField.Type	Type	Record	List
6	GroupKind.Type	Type	Record	List

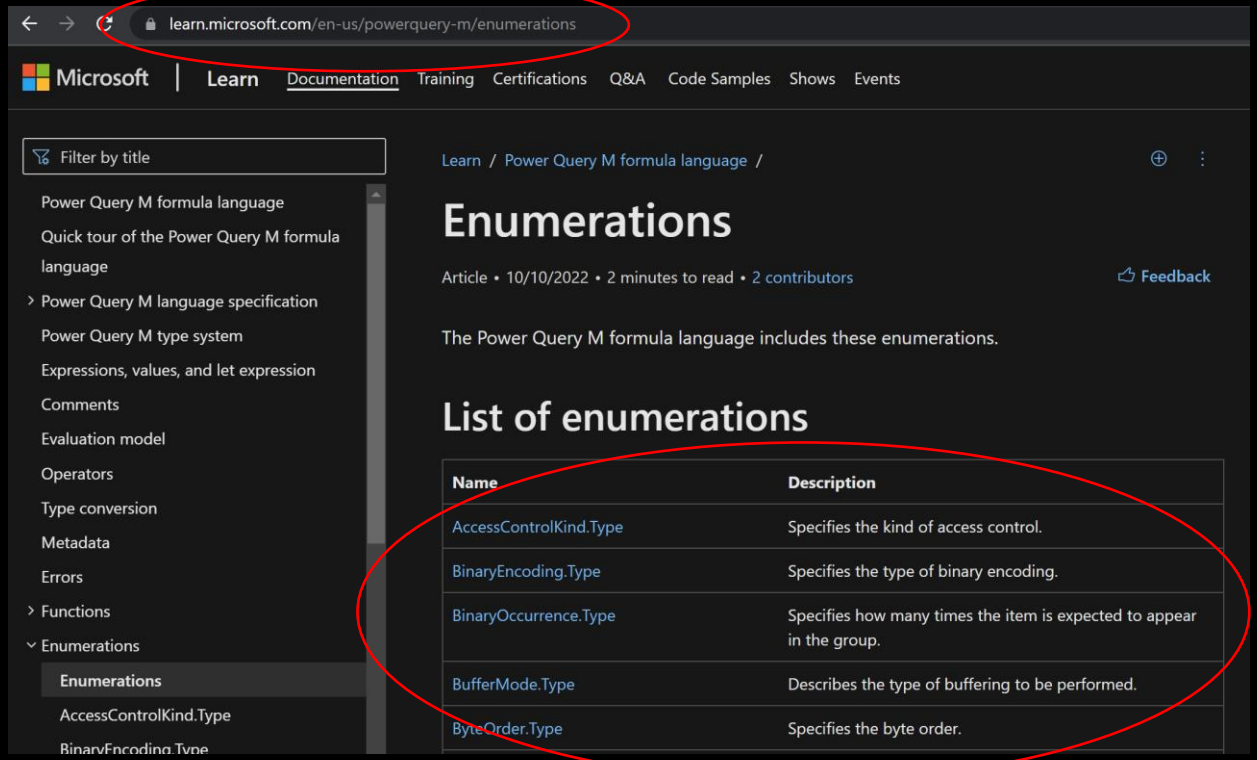
List
0
1
2
3
4
5

As an alternative, we can query the web documentation directly and get all the enum values at once

```
let
    URLPrefix = "https://learn.microsoft.com/en-us/powerquery-m/",

    MainPage = URLPrefix & "enumerations",

    Source = Web.Page(Web.Contents(MainPage)){0}[Data],

    EmptyTable = #table({"Name", "Value", "Description", "Enum", "URL"},{}),

    GetAllowedTables
        = List.Accumulate(
            Source[Name],
            EmptyTable,
            (output,enum) =>
                let
                    URL = URLPrefix & Text.Replace(enum, ".", "-"),
                    WebTable = Web.Page(Web.Contents(URL)){0}[Data],
                    AddEnumName = Table.AddColumn(WebTable, "Enum", each enum),
                    AddEnumURL = Table.AddColumn(AddEnumName, "URL", each URL)
                in
                    Table.Combine({output,AddEnumURL})
        )
in
    GetAllowedTables
```

There's a main page that has a table with a list of all enumerations

```
let
    URLPrefix = "https://learn.microsoft.com/en-us/powerquery-m/",

    MainPage = URLPrefix & "enumerations",

    Source = Web.Page(Web.Contents(MainPage)){0}[Data],
```



And each enum has a page of its own

```
8       EmptyTable = #table({"Name", "Value", "Description", "Enum", "URL"},{}),
9
10      GetAllowedTables
11          = List.Accumulate(
12              Source[Name],
13              EmptyTable,
14              (output,enum) =>
15                  let
16                      URL = URLPrefix & Text.Replace(enum, ".", "-"),
17                      WebTable = Web.Page(Web.Contents(URL)){0}[Data],
```

Each name is from the table on the main page. They have the form Enum.Type

Each URL for an enum ends with text of the form enum-type, so we modify the Enum name, then retrieve the first table from the web page. That table contains three columns:

1) Name
2) Value
3) Description

List.Accumulate iterates through a list and acts on each item in turn, much like REDUCE in Excel

We build a table for each enum and stack them on top of each other

```
18                  AddEnumName = Table.AddColumn(WebTable, "Enum", each enum),
19                  AddEnumURL = Table.AddColumn(AddEnumName, "URL", each URL)
20              in
21                  Table.Combine({output,AddEnumURL})
22          )
23  in
24      GetAllowedTables
```

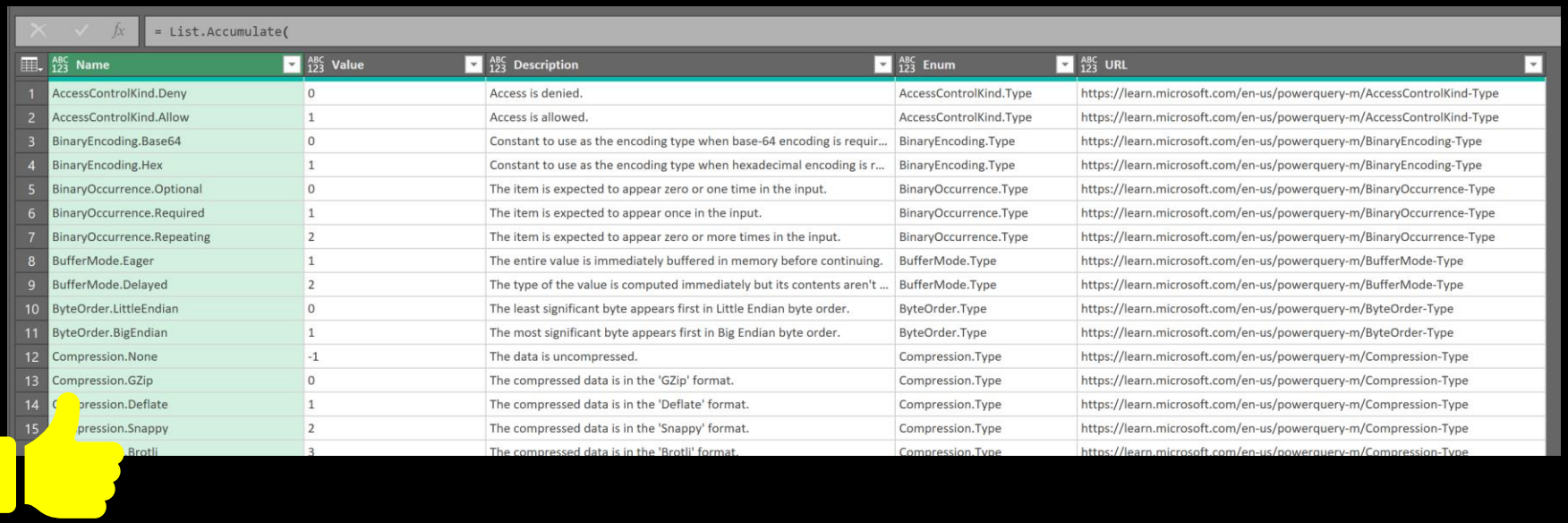
We can then add to the table a column for the enum name and for the URL to the page

Using Table.Combine in List.Accumulate, we can combine each successive iteration with the result of the previous iteration.

On the first iteration, this means we're adding the table retrieved from the first URL to the empty table.

On the second iteration, this means we're adding the table retrieved from the second URL to the table produced by the first iteration

The result is a table of all enumerations and their allowed values



```
= List.Accumulate(
```

	Name	Value	Description	Enum	URL
1	AccessControlKind.Deny	0	Access is denied.	AccessControlKind.Type	https://learn.microsoft.com/en-us/powerquery-m/AccessControlKind-Type
2	AccessControlKind.Allow	1	Access is allowed.	AccessControlKind.Type	https://learn.microsoft.com/en-us/powerquery-m/AccessControlKind-Type
3	BinaryEncoding.Base64	0	Constant to use as the encoding type when base-64 encoding is requir...	BinaryEncoding.Type	https://learn.microsoft.com/en-us/powerquery-m/BinaryEncoding-Type
4	BinaryEncoding.Hex	1	Constant to use as the encoding type when hexadecimal encoding is r...	BinaryEncoding.Type	https://learn.microsoft.com/en-us/powerquery-m/BinaryEncoding-Type
5	BinaryOccurrence.Optional	0	The item is expected to appear zero or one time in the input.	BinaryOccurrence.Type	https://learn.microsoft.com/en-us/powerquery-m/BinaryOccurrence-Type
6	BinaryOccurrence.Required	1	The item is expected to appear once in the input.	BinaryOccurrence.Type	https://learn.microsoft.com/en-us/powerquery-m/BinaryOccurrence-Type
7	BinaryOccurrence.Repeating	2	The item is expected to appear zero or more times in the input.	BinaryOccurrence.Type	https://learn.microsoft.com/en-us/powerquery-m/BinaryOccurrence-Type
8	BufferMode.Eager	1	The entire value is immediately buffered in memory before continuing.	BufferMode.Type	https://learn.microsoft.com/en-us/powerquery-m/BufferMode-Type
9	BufferMode.Delayed	2	The type of the value is computed immediately but its contents aren't ...	BufferMode.Type	https://learn.microsoft.com/en-us/powerquery-m/BufferMode-Type
10	ByteOrder.LittleEndian	0	The least significant byte appears first in Little Endian byte order.	ByteOrder.Type	https://learn.microsoft.com/en-us/powerquery-m/ByteOrder-Type
11	ByteOrder.BigEndian	1	The most significant byte appears first in Big Endian byte order.	ByteOrder.Type	https://learn.microsoft.com/en-us/powerquery-m/ByteOrder-Type
12	Compression.None	-1	The data is uncompressed.	Compression.Type	https://learn.microsoft.com/en-us/powerquery-m/Compression-Type
13	Compression.GZip	0	The compressed data is in the 'GZip' format.	Compression.Type	https://learn.microsoft.com/en-us/powerquery-m/Compression-Type
14	C pression.Deflate	1	The compressed data is in the 'Deflate' format.	Compression.Type	https://learn.microsoft.com/en-us/powerquery-m/Compression-Type
15	pression.Snappy	2	The compressed data is in the 'Snappy' format.	Compression.Type	https://learn.microsoft.com/en-us/powerquery-m/Compression-Type
	Brotli	3	The compressed data is in the 'Brotli' format.	Compression.Type	https://learn.microsoft.com/en-us/powerquery-m/Compression-Type