

Глава 23. Таблицы динамических параметров в Power Query

Это продолжение перевода книги Кен Пульс и Мигель Эскобар. Язык M для Power Query. Главы не являются независимыми, поэтому рекомендую читать последовательно.

[Предыдущая глава](#) [Содержание](#) [Следующая глава](#)

В [главе 21](#) мы рассмотрели, как пользовательские функции применяются для предварительной обработки наборов данных перед объединением файлов. В этой главе мы добавим «вишенку на торт». Вспомните пример из [предыдущей](#) главы, по извлечению расписаний из папки. Допустим, вы сохранили файл консолидации в H:\Payroll, и в течение нескольких месяцев хранили расписания во вложенной папке H:\Payroll\Timesheets. Если вы передадите свое решение кому-то другому, на новом компьютере название папки, скорее всего, будет иным. Нельзя ли настроить решение так, чтобы исходный путь не играл роли, а было лишь важно, что данные хранятся во вложенной папке ...Timesheets относительно файла консолидации?

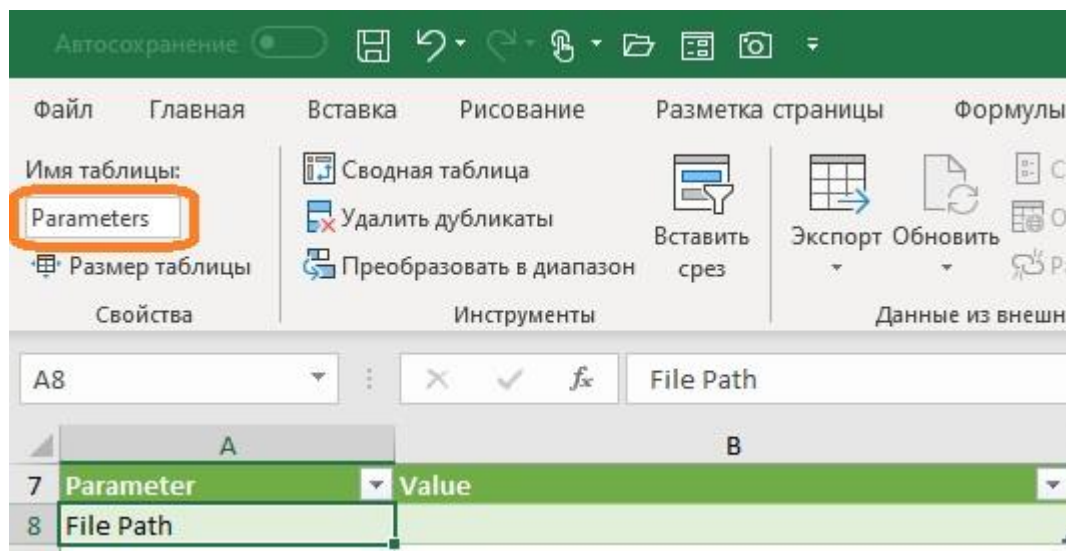


Рис. 23.1. Таблица параметров

В настоящее время в языке M нет функции, позволяющей определить путь к используемой книге. Чтобы реализовать описанный выше сценарий, нужно:

- создать таблицу параметров в Excel,
- создать функцию Power Query для извлечения значений из таблицы,
- модифицировать существующие запросы для вызова функции.

Откройте файл *Parameter Tables.xlsx*. Он является продолжением файла расписаний, с которым вы работали несколько последних глав. В настоящее время все пути к файлам с исходными данными жестко закодированы.

Создание таблицы параметров

На листе *Info* создайте заготовку таблицы, как показано на рис. 23.1. Следующие названия должны быть такими же, как на рисунке: заголовок первого столбца *Parameter*, заголовок второго столбца *Value*, имя таблицы *Parameters*. Это позволит использовать заготовленный текст функции.

Теперь вам каким-либо способом нужно указать путь к папке *Source Files* на вашем ПК. Это может быть текст или значение, возвращаемое формулой в Excel. Мы предложим вам решение на основе функции =ЯЧЕЙКА("имяфайла"). Введите следующую формулу в ячейку B8:

```
=ЛЕВСИМВ(ЯЧЕЙКА("имяфайла");НАЙТИ("[";ЯЧЕЙКА("имяфайла");1)-1)&"Source Files\"
```

Если вы еще не сохранили файл, эта функция вернет ошибку, поскольку Excel не может определить, где находится книга. Сохранение файла решит эту проблему:

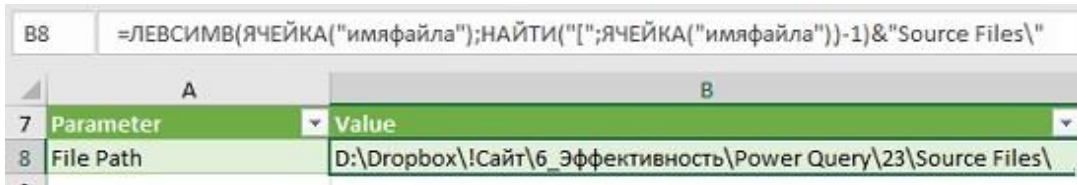


Рис. 23.2. Динамически возвращаемый путь к папке *Source Files* на основе формулы Excel

Несколько слов о том, как работает формула. Фрагмент ЯЧЕЙКА("имяфайла") возвращает путь к файлу, имя файла и имя активного листа – D:\Dropbox\!Сайт\6_Эффективность\Power Query\23\[Parameter Tables.xlsx]Info. Функция НАЙТИ() ищет первое вхождение символа [в пути к файлу. Функция ЛЕВСИМВ() возвращает текст – имя файла до символа [. Поскольку сам символ [нужно исключить, уменьшаем число извлекаемых символов на единицу. К извлеченному тексту с помощью знака конкатенации & добавляем Source Files\

Функция *fnGetParameter*

Теперь нужно предоставить Power Query возможность считывания значения в ячейке B8. Это можно сделать с помощью пользовательской функции:

```
(ParameterName as text) =>
let
    ParamSource = Excel.CurrentWorkbook()
        { [Name="Parameters"] } [Content],
    ParamRow = Table.SelectRows(ParamSource,
        each ([Parameter]=ParameterName)),
    Value=
        if Table.IsEmpty(ParamRow)=true
        then null
        else Record.Field(ParamRow{0}, "Value")
in
    Value
```

Функция подключается к таблице параметров в книге Excel, и извлекает путь к папке Source Files. Откройте файл *fnGetParameter.txt* и скопируйте его содержимое в буфер. В Excel перейдите по меню *Данные* → *Получить данные* → *Из других источников* → *Пустой запрос*. В редакторе Power Query перейдите на вкладку *Главная* → *Расширенный редактор*. Выделите все строки кода и нажмите Ctrl+V, чтобы вставить содержимое текстового файла из буфера обмена. Нажмите *Готово*. Измените имя функции на *fnGetParameter*. *Главная* → *Заккрыть и загрузить*. Напомню, что для функций используется единственный тип загрузки – *Только создать подключение*.

Вызов функции *fnGetParameter*

Итак, у вас есть актуальный путь к папке с исходными файлами. Этот путь обновляется при открытии на каждом новом ПК. Вам осталось модернизировать запрос. В файле Excel перейдите по меню *Данные* → *Запросы и подключения*. В области *Запросы и подключения* кликните правой кнопкой мыши на запросе *Timesheets* → *Изменить*. Запрос возвращает ошибку, так как путь к папке прописан жестко:

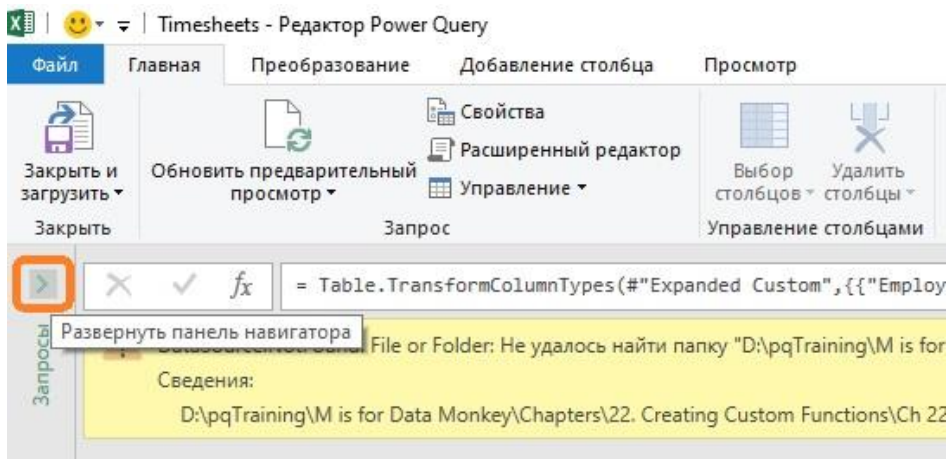


Рис. 23.3. Разверните панель навигатора

Кликните стрелку *Развернуть панель навигатора*. Навигатор позволяет выбрать любой из ваших запросов/функций, так что удобно переключаться между ними, внося изменения и проверяя эффекты, которые изменения оказывают на другие запросы (см. ниже рис. 23.5). Щелкните правой кнопкой мыши запрос *Timesheets* → *Расширенный редактор*. Вставьте строку кода сразу после *let*:

```
fullfilepath = fnGetParameter("File Path"),
```

Не забудьте про запятую в конце строки:

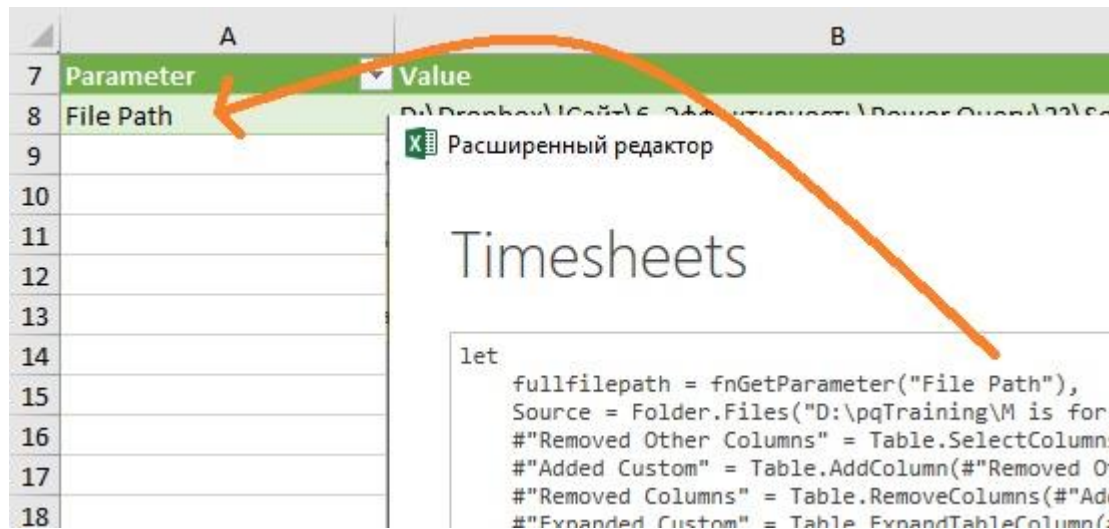


Рис. 23.4. Добавлена строка с вызовом функции *fnGetParameter*

Вы создали новую переменную *fullfilepath* для хранения значения из строки *File Path* таблицы Excel. Кстати, это делать не надо. Вы можете использовать вызов функции *fnGetParameter* вместо переменной в следующей строке. Однако, создав отдельную строку, вы облегчите отладку запроса. Нажмите *Готово*. Выберите шаг *fullfilepath* в области ПРИМЕНЕННЫЕ ШАГИ. Вам легко проверить, что путь указан верно (не будь отдельного шага с определением переменной, увидеть путь в коде Power Query было бы невозможно):

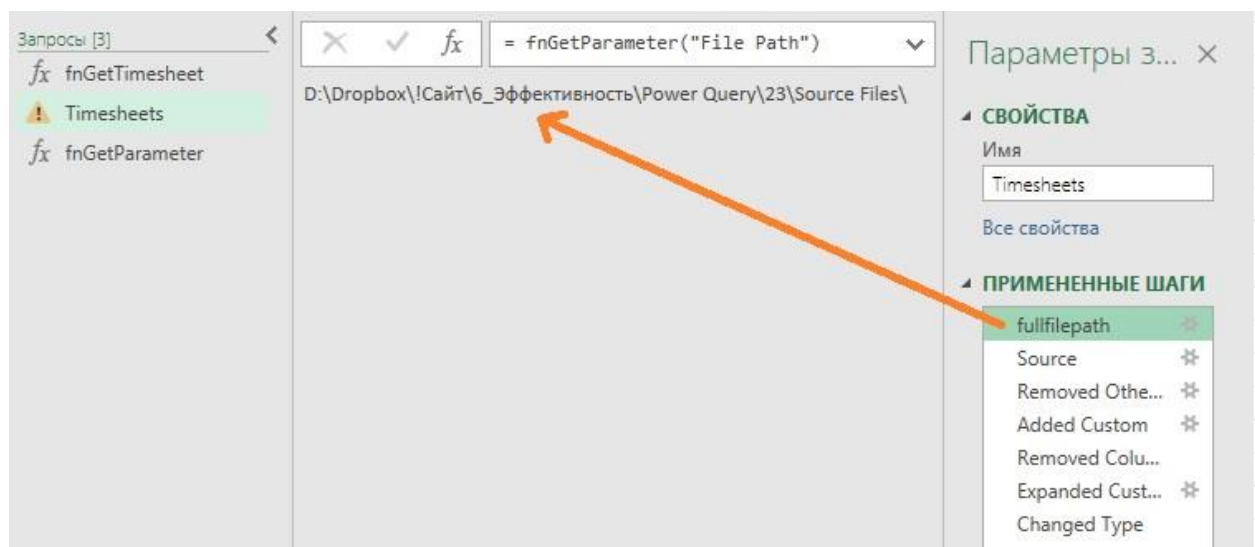


Рис. 23.5. Переменная *fullfilepath* правильно определяет путь к папке

Теперь, когда вы убедились, что функция возвращает правильный путь с помощью формулы Excel, вы можете подставить переменную вместо жестко заданного пути к файлу. Щелкните правой кнопкой мыши запрос *Timesheets* → *Расширенный редактор*. Вместо...

```
let
    fullfilepath = fnGetParameter("File Path"),
    Source = Folder.Files("D:\pqTraining\M is for Data Monkey\Chapters\22. Creating Custom Functions\Ch 22 Examples\Source Files"),
```

...введите

let

```
fullfilepath = fnGetParameter("File Path"),  
Source = Folder.Files(fullfilepath),
```

Нажмите *Готово*. Теперь запрос функционирует правильно:

	Employee	1.2 Exp...	1.2 Misc...	1.2 O...	O...	1.2 Re...	Work...
1	John Thompson	0	0	0	18:00:00	8,5	03.03.2015
2	John Thompson	0	0	0	18:00:00	8	04.03.2015
3	John Thompson	0	0	0	18:00:00	8,5	05.03.2015
4	John Thompson	0	0	0	18:00:00	8,5	06.03.2015
5	John Thompson	0	0	0	15:30:00	6,5	07.03.2015
6	John Thompson	0	0	0	18:00:00	8,5	10.03.2015
7	John Thompson	0	0	0	18:00:00	8,5	11.03.2015
8	John Thompson	0	0	0	18:00:00	6	12.03.2015
9	John Thompson	0	0	0	18:00:00	8,5	13.03.2015
10	John Thompson	0	0	0	18:00:00	8,5	14.03.2015
11	Bob Johnson	0	0	0	18:00:00	9,5	03.03.2015
12	Bob Johnson	0	0	0	18:00:00	8	04.03.2015
13	Bob Johnson	0	0	0	18:00:00	8,5	05.03.2015

Рис. 23.6. Модифицированный запрос «нашел» папку с файлами

Резюме

Ссылка на таблицу параметров дает нам огромную гибкость при построении решений. Это особенно важно, если вы разрабатываете решения для клиентов внутри или вне компании. Последнее, что вы захотите сделать, это отправить конечному пользователю инструкции по редактированию кода M))

Но сила таблиц параметров этим не ограничивается. Они будут также весьма полезны для решения следующих задач:

- Создание настольный календарь на основании даты в ячейке листа Excel.
- Создание фильтра для таблицы на основе значения в ячейке Excel.
- Выбор одной из нескольких таблиц Excel, которые следует загрузить в решение. Опять же на основе значения в ячейке Excel.

Возможно, это самая ценная идея в книге...