

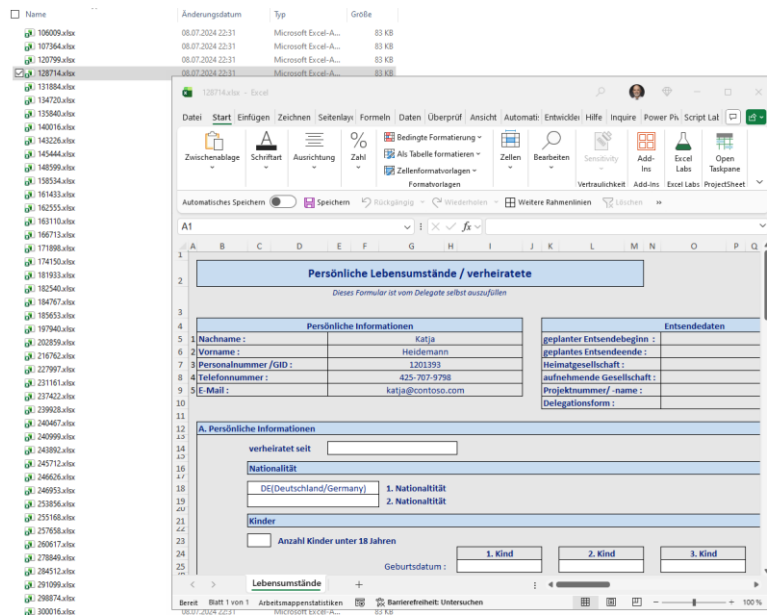
Buggen und Debuggen in Power Query

Excelstammtisch vom 17.11.2025

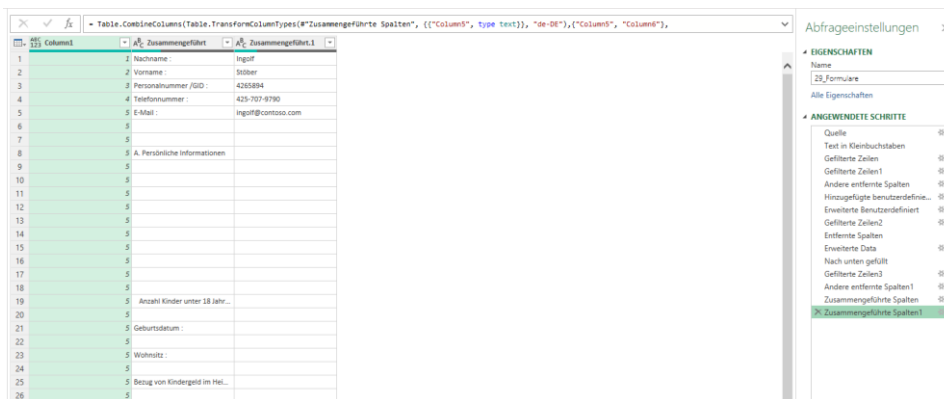
(zusammengefasst von René Martin)

1. Plötzlich ein Fehler ...

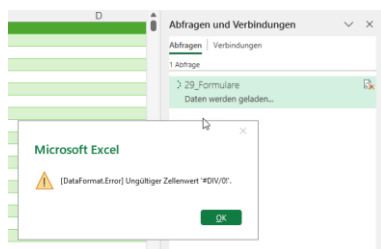
In einem Ordner befinden sich eine große Anzahl an Excelmappen. Es handelt sich um Formulare mit Informationen (hier: zu bestimmten Personen)



Mit Power Query wird auf den Ordner zugegriffen, die Daten werden extrahiert und transformiert:

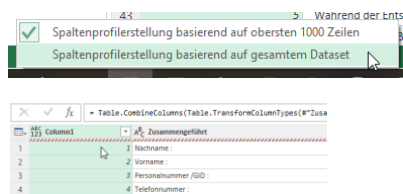


Und nach Excel geladen, wo ein Fehler gemeldet wird:



In Power Query wird dieser allerdings nicht angezeigt. Wo steckt er?

Die Antwort: In Power Query werden nur die ersten 1.000 Zeilen angezeigt – der Fehler steckt jedoch weiter unten ...



Dann wird der Fehler auch angezeigt.

2. Fehler wird nicht angezeigt – doch!

Ganz anders verhält es sich bei folgendem Problem.

Jede Woche erhält man einen Download aus SAP. In dieser Excelmappe müssen Spalten und Zeilen gelöscht werden, der Datentyp der Zahlen muss Zahl und nicht Text sein.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	Status:			offen						vorerfasst			ausgegl.							
2	fälligkeit:			überfällig						fällig			nicht fäll.							
3																				
4	Kreditort							704324												
5	Buchungskreis							140												
6																				
7	Name						ABC													
8	Ort						Schaffhausen													
9																				
10																				
11			St	Zuordnung		Rfrnz			Art	St	Belegdatum	Buch.dat.		Fa	Betrag in l Währ	Betr. in H Währ	Text			
12																				
13			812500	9409332		9409332			KW	VO	29.05.2025	15.06.2025			6.761,73	EUR	6.761,73	EUR	RECHNUNG ABC 812500	
14			811879	811879		811879			KC	VO	01.06.2025	30.06.2025			123,33	EUR	123,33	EUR	ABC Rep.	
15			811875	811875		811875			KC	VO	01.06.2025	30.06.2025			202,60	EUR	202,60	EUR	ABC Rep.	
16			811874	811874		811874			KC	VO	01.06.2025	30.06.2025			505,84	EUR	505,84	EUR	ABC Rep.	
17			811890	811890		811890			KC	VO	01.06.2025	30.06.2025			479,22	EUR	479,22	EUR	ABC Rep.	
18			811867	811867		811867			KC	VO	01.06.2025	30.06.2025			418,05	EUR	418,05	EUR	ABC Rep.	
19			101768	101768		101768			KC	VO	01.06.2025	30.06.2025			54.984,00	EUR	54.984,00	EUR	WM-Tickets Coop	
20			811868	811868		811868			KC	VO	01.06.2025	30.06.2025			116,88	EUR	116,88	EUR	ABC Rep.	
21			811893	811893		811893			KC	VO	01.06.2025	30.06.2025			286,36	EUR	286,36	EUR	ABC Rep.	
22			811917	811917		811917			KC	VO	01.06.2025	30.06.2025			253,25	EUR	253,25	EUR	ABC Rep.	
23			811918	811918		811918			KC	VO	01.06.2025	30.06.2025			171,43	EUR	171,43	EUR	ABC Rep.	
24			811920	811920		811920			KC	VO	01.06.2025	30.06.2025			253,25	EUR	253,25	EUR	ABC Rep.	
25			811934	811934		811934			KC	VO	01.06.2025	30.06.2025			405,19	EUR	405,19	EUR	ABC Rep.	
26			811956	811956		811956			KC	VO	01.06.2025	30.06.2025			279,74	EUR	279,74	EUR	ABC Rep.	
27			811892	811892		811892			KC	VO	01.06.2025	30.06.2025			93,50	EUR	93,50	EUR	ABC Rep.	
28			811869	811869		811869			KC	VO	01.06.2025	30.06.2025			284,42	EUR	284,42	EUR	ABC Rep.	
29			811870	811870		811870			KC	VO	01.06.2025	30.06.2025			307,79	EUR	307,79	EUR	ABC Rep.	
30			811872	811872		811872			KC	VO	01.06.2025	30.06.2025			214,28	EUR	214,28	EUR	ABC Rep.	
31			811880	811880		811880			KC	VO	01.06.2025	30.06.2025			401,30	EUR	401,30	EUR	ABC Rep.	
32			811873	811873		811873			KC	VO	01.06.2025	30.06.2025			54,55	EUR	54,55	EUR	ABC Rep.	
33			811882	811882		811882			KC	VO	01.06.2025	30.06.2025			208,44	EUR	208,44	EUR	ABC Rep.	
34			811891	811891		811891			KC	VO	01.06.2025	30.06.2025			299,99	EUR	299,99	EUR	ABC Rep.	
35			811889	811889		811889			KC	VO	01.06.2025	30.06.2025			227,92	EUR	227,92	EUR	ABC Rep.	
36			811886	9409159		9409159			KW	VO	01.06.2025	12.06.2025			55,72	EUR	55,72	EUR	RECHNUNG ABC 811886	
37			811884	9409158		9409158			KW	VO	01.06.2025	12.06.2025			45,45	EUR	45,45	EUR	RECHNUNG ABC 811884	
38			811883	9409157		9409157			KW	VO	01.06.2025	12.06.2025			113,64	EUR	113,64	EUR	RECHNUNG ABC 811883	
39			811878	9409156		9409156			KW	VO	01.06.2025	12.06.2025			553,89	EUR	553,89	EUR	RECHNUNG ABC 811878	
40			811876	9409155		9409155			KW	VO	01.06.2025	12.06.2025			107,15	EUR	107,15	EUR	RECHNUNG ABC 811876	
41			811871	9409154		9409154			KW	VO	01.06.2025	12.06.2025			438,35	EUR	438,35	EUR	RECHNUNG ABC 811871	
42			811901	811901		811901			KW	VO	01.06.2025	12.06.2025			1010,41	EUR	1010,41	EUR	RECHNUNG ABC 811901	

Ein klarer Fall für Power Query.

Die Zeilen und Spalten werden entfernt:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	Zuordnung																			
2																				
3																				
4																				
5																				
6																				
7																				
8																				
9																				
10																				
11																				
12																				
13																				
14																				
15																				
16																				
17																				
18																				
19																				
20																				
21																				
22																				
23																				

Die erste Zeile wird als Überschrift verwendet:

= Table.SelectRows("#Geänderter Typ", each ([Zuordnung] <> null))				
ABC 123	Zuordnung	Belegdatum	1.2 Betrag in BW	A ^B _C Text
7	101768	01.06.2025	34984	WM-Tickets Coop
8	811868	01.06.2025	416,88	ABC Rep.
9	811893	01.06.2025	286,36	ABC Rep.
10	811917	01.06.2025	253,25	ABC Rep.
11	811918	01.06.2025	171,43	ABC Rep.
12	811920	01.06.2025	253,25	ABC Rep.
13	811934	01.06.2025	405,19	ABC Rep.
14	811956	01.06.2025	279,74	ABC Rep.
15	811892	01.06.2025	93,5	ABC Rep.

Der Datentyp der Spalte „Zuordnung“ wird in ganze Zahl umgewandelt:

1 ² 3	Zuordnung	Bele
1	9409332	
2	811879	
3	811875	
4	811874	

Zufällig entdeckt man, dass in Zeile 54 ein Fehler steckt. Dort befand sich Text, der nicht in eine Zahl konvertiert werden konnte:

53	812341	06.06.2025
54	Error	07.06.2025
55	9409329	07.06.2025
56	812366	07.06.2025
57	812463	07.06.2025
58	812384	07.06.2025

DataFormat.Error: Die Konvertierung in "Number" war nicht möglich.
Details: ROLLWA

Aber: Warum zeigt das Power Query nicht an?

= Table.TransformColumn				
1 ² 3	Zuordnung	Belegdatum		
	<ul style="list-style-type: none"> Gültig 100 % Fehler 0 % Leer 0 % 	<ul style="list-style-type: none"> Gültig Fehler Leer 		
	<div> <div>Zuordnung</div> <div>560 (100 %)</div> <div>Gültig</div> </div> <div> <div>0 (0 %)</div> <div>Fehler</div> </div> <div> <div>0 (0 %)</div> <div>Leer</div> </div>			
7	101768			

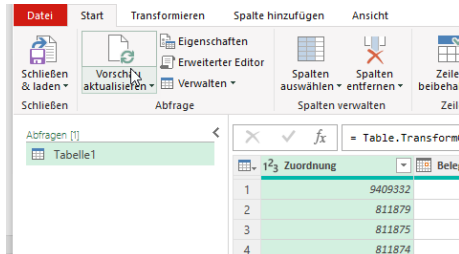
Der Grund: Durch das Verschieben der ersten Zeile zur Überschrift wurde automatisch der Datentyp any zugewiesen:

= Table.TransformColumnTypes("#Höher gestufte Header",{{"Zuordnung", type any}, {"Belegdatum"				
ABC 123	Zuordnung	Belegdatum	1.2 Betrag in BW	A ^B _C Text
1	9409332	29.05.2025	6761,73	RECHNUNG ABC 812500
2	811879	01.06.2025	212,33	ABC Rep.
3	811875	01.06.2025	202,6	ABC Rep.
4	811874	01.06.2025	305,84	ABC Rep.

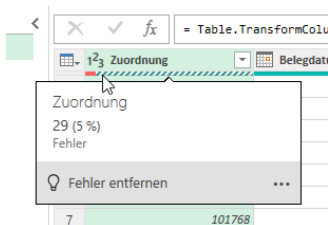
Wird er nun manuell geändert, holt sich Power Query aus dem Cache noch den „alten“ Datentyp „any“:

	Zuordnung	Belegdatum	1.2 Betrag in BW	Text
1	9409332	29.05.2025	6761,73	RECHNUNG ABC 812500
2	811879	01.06.2025	212,33	ABC Rep.
3	811875	01.06.2025	202,6	ABC Rep.
4	811874	01.06.2025	305,84	ABC Rep.
5	811890	01.06.2025	479,22	ABC Rep.

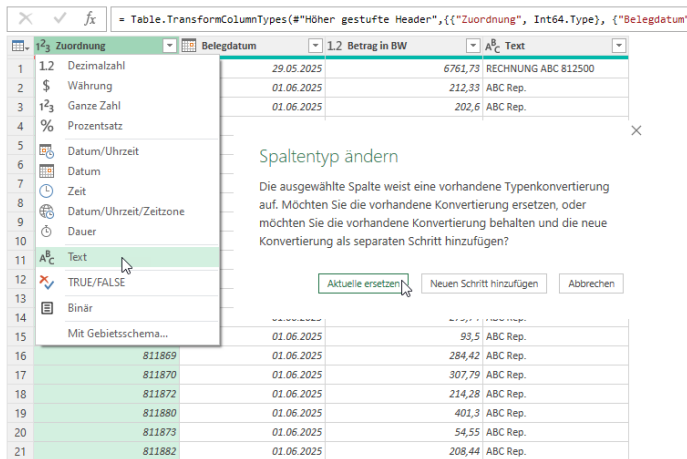
Das Perfide an dieser Aktion: Auch Aktualisieren nützt nichts!



Lediglich das Wechseln zu einem anderen Schritt und wieder zurück – JETZT wird der Fehler angezeigt!



Umgekehrt ebenso: Wird der Datentyp durch den Typ „Text“ ersetzt:



Wird weiterhin der alte, jetzt aber nicht mehr vorhandene Fehler angezeigt.

	Zuordnung	Belegdatum
1	9409332	
2	811879	
3	811875	
4	811874	
5	811890	
6	811867	

Erst das Wechseln zu einem anderen Schritt und wieder zurück beendet den Spuk!

Perfide!

3.... try ... otherwise ...

Angenommen in einer Tabelle befinden sich Fehler oder können fehlerhafte Daten enthalten:

=XVERWEIS(C19;Kurs!\$A\$2:\$A\$11;Kurs!\$B\$2:\$B\$11)				
A	B	C	D	E
Artikel	Preis	Währung	Kurs	
1 Humpfendübel	2,19	EUR	1	
2 Schnalzzange	5,00	CHF	0,93	
3 Breklopf	80,00	RUB	90,36	
4 Blechlochscharter	500,00	JPY	162,01	
5 Streckkreide	1,25	GBP	0,84	
6 Krählochscher	4,00	CHF	0,93	
7 Steckscheibenkrepp	0,69	EUR	1	
8 Zwingstreckendoppler	0,89	USD	1,14	
9 Schwarzwandkupplung	1,00	RUB	90,36	
10 Blendhammer	400,00	JPY	162,01	
12 Orkandhacke	0,50	GBP	0,84	
13 Meisterkottverbindung	2,00	GBP	0,84	
14 Lattengitterfang	0,40	EUR	1	
15 Beutelrattenklemme	2,19	EUR	1	
16 Handwieselpumpe	5,00	CHF	0,93	
17 Gröhlzahnkurbel	80,00	RUB	90,36	
18 Heftschneldendichtung	500,00	JPY	162,01	
19 Rotholzschaltung	1,25	ZAR	#NV	
20 Tragbares Loch	4,00	CHF	0,93	
21 Ayurvedische Würfel	0,69	EUR	1	
22 Albzahnseilensneider	0,89	USD	1,14	
23 Humpfendübel	1,00	RUB	90,36	
24 Schnalzzange	400,00	JPY	162,01	
25 Breklopf	0,50	GBP	0,84	
26 Blechlochscharter	2,00	GBP	0,84	
27 Streckkreide	0,40	EUR	1	

Man kann die Weiterverarbeitung mit try ... otherwise abfangen. Also statt:

```
= Table.AddColumn("#Höher gestufte Header",
    "Währung in EURO", each [Preis] / [Kurs])
```

= Table.AddColumn("#Höher gestufte Header", "Währung in EURO", each [Preis] / [Kurs])				
Artikel	Preis	Währung	Kurs	Währung in EURO
1 Humpfendübel	2,19	EUR	1	2,19
2 Schnalzzange	5	CHF	0,93	5,376344086
3 Breklopf	80	RUB	90,36	0,885347499
4 Blechlochscharter	500	JPY	162,01	3,086229245
5 Streckkreide	1,25	GBP	0,84	1,488095238
6 Krählochscher	4	CHF	0,93	4,301075269
7 Steckscheibenkrepp	0,69	EUR	1	0,69
8 Zwingstreckendoppler	0,89	USD	1,14	0,780701754
9 Schwarzwandkupplung	1	RUB	90,36	0,011066844
10 Blendhammer	400	JPY	162,01	2,468983396
11 Orkandhacke	0,5	GBP	0,84	0,595238095
12 Meisterkottverbindung	2	GBP	0,84	2,380952381
13 Lattengitterfang	0,4	EUR	1	0,4
14 Beutelrattenklemme	2,19	EUR	1	2,19
15 Handwieselpumpe	5	CHF	0,93	5,376344086
16 Gröhlzahnkurbel	80	RUB	90,36	0,885347499
17 Heftschneldendichtung	500	JPY	162,01	3,086229245
18 Rotholzschaltung	1,25	ZAR	Error	Error
19 Tragbares Loch	4	CHF	0,93	4,301075269
20 Ayurvedische Würfel	0,69	EUR	1	0,69
21 Albzahnseilensneider	0,89	USD	1,14	0,780701754
22 Humpfendübel	1	RUB	90,36	0,011066844
23 Schnalzzange	400	JPY	162,01	2,468983396
24 Breklopf	0,5	GBP	0,84	0,595238095
25 Blechlochscharter	2	GBP	0,84	2,380952381
26 Streckkreide	0,4	EUR	1	0,4

Kann man schreiben:

```
= Table.AddColumn("#Höher gestufte Header",
    "Währung in EURO",
    each try [Preis] / [Kurs] otherwise 0)
```

Table.AddColumn("#Höher gestufte Header", "Währung in EURO", each try [Preis] / [Kurs] otherwise 0)

ABC 123	Artikel	ABC 123	Preis	ABC 123	Währung	ABC 123	Kurs	ABC 123	Währung in EURO
	1		Humpfendübel		2,19		EUR		2,19
	2		Schnalzzange		5		CHF		5,376344086
	3		Breiklopf		80		RUB		0,885347499
	4		Blechlochscherer		500		JPY		3,086229245
	5		Streckkreide		1,25		GBP		1,488095238
	6		Krählöscher		4		CHF		4,301075269
	7		Steckscheibenkrepp		0,69		EUR		0,69
	8		Zwingstreckendoppler		0,89		USD		0,780701754
	9		Schwarzwandkupplung		1		RUB		0,011066844
	10		Blendhammer		400		JPY		2,468983396
	11		Orkandhacke		0,5		GBP		0,595238095
	12		Meisterkottverbindung		2		GBP		2,380952381
	13		Lattengitterfang		0,4		EUR		0,4
	14		Beuterrattenklemme		2,19		EUR		2,19
	15		Handwieselpumpe		5		CHF		5,376344086
	16		Größtstahnkurbel		80		RUB		0,885347499
	17		Heftschneidendichtung		500		JPY		3,086229245
	18		Rotholzschaltung		1,25		ZAR	Error	0
	19		Tragbares Loch		4		CHF		4,301075269
	20		Ayurvedische Würfel		0,69		EUR		0,69
	21		Altbahnseitenschneider		0,89		USD		0,780701754

Den try-Befehl kann man auf fast alle Befehle anwenden, beispielsweise: der Datentyp soll geändert werden:

```
= Table.AddColumn(Benutzerdefiniert1, "Benutzerdefiniert", each try  
Decimal.From([Kurs]) otherwise 0 )
```

Oder:

```
= Table.TransformColumns("#Hinzugefügte benutzerdefinierte Spalte", {{  
"Kurs", each try Decimal.From(_) otherwise 0}})
```

4. Benutzereingaben abfangen

Und diesen Befehl kann man auch verwenden, um Benutzereingaben abzufangen.

Beispiel

In einem Ordner befinden sich mehrere Formulare:

Name	Änderungsdatum
Anders.xlsx	04.11.2025 20:10
Bauer.xlsx	04.11.2025 20:10
Huber.xlsx	04.11.2025 20:10
Maier.xlsx	04.11.2025 20:11
Martin.xlsx	04.11.2025 20:38
Schmidt.xlsx	04.11.2025 20:11
Schulz.xlsx	04.11.2025 20:13
Wagner.xlsx	04.11.2025 20:13
Zeyer.xlsx	04.11.2025 20:12

Sie werden geöffnet und weiterverarbeitet:

Kurztext	Anders	Bauer	Huber	Maier	Martin	Schmidt	Schulz	Wagner	Zeyer
Inhaltsverzeichnis									
Allgemeine Beschreibung									
Leistungsverzeichnis									
Positionen									
Arbeitskräfte für Baumpflegemaßnahmen									
Mitarbeiter*in Baumpflege mit Zertifikat Seilklettertechnik Stufe B (SKT-B)	4	7	25	8	8	75	58	89	61
Zuschlag zur Vorposition für Nacharbeit	32	78	6	59	96	81	4	23	34
Mitarbeiter*in Baumpflege mit Zertifikat Seilklettertechnik Stufe A (SKT-A)	14	82	84	2	8	97	54	23	51
Zuschlag zur Vorposition für Nacharbeit	33	38	93	48	38	90	32	50	31
Mitarbeiter*in Baumpflege mit Zertifikat AS Baum 2	3	3	14	35	85	82	80	11	95
Zuschlag zur Vorposition für Nacharbeit	4	20	46	11	39	39	97	77	39
Mitarbeiter*in Baumpflege mit Zertifikat AS Baum 1	27	12	1	54	96	31	41	96	63
Zuschlag zur Vorposition für Nacharbeit	44	89	64	20	4	87	37	16	81
Mitarbeiter*in Baumpflege angelernte Hilfskraft	16	12	87	35	79	47	82	91	20
Zuschlag zur Vorposition für Nacharbeit	7	12	25	78	55	22	4	68	42
Arbeitskräfte für Baumpflegemaßnahmen									
Transportfahrzeuge, Maschinen und Geräte	3	25	67	21	98	77	91	39	74
Transportfahrzeug zul. Ges. bis 3.500 kg	54	51	82	5	86	52	96	64	37
Transportfahrzeug zul. Ges. bis 18.000 kg inkl. Ladekran	69	23	9	1	8	81	52	75	10
Falkran inkl. Bedienung, Reichweite ca. 30 m	97	99	59	5	24	49	40	24	88
Fallgreifer inkl. Zugfahrzeug und Bedienung, Reichweite ca. 20 m	89	97	40	23	94	64	33	17	42
Motorsäge Schnittlänge von 20 cm bis 30 cm	61	96	41	91	81	44	63	8	67
Motorsäge Schnittlänge von 30 cm bis 50 cm	28	26	75	65	5	74	59	91	8
Häcksler Material über 16 cm bis 22 cm Durchmesser	53	29	20	73	12	73	95	65	94
Häcksler Material über 22 cm Durchmesser	67	77	69	9	59	99	80	41	87
handgeführtes Laubblassgerät	26	76	16	33	79	72	95	55	71
Transportfahrzeuge, Maschinen und Geräte									
Verkehrsrechtliche Anordnung									
Einholen verkehrsrechtliche Anordnung	26	72	14	1	41	37	11	96	52
Verkehrsrechtliche Anordnung									
Einbau von Kronensicherungen									
Dynamisches Kronensicherungssystem Mindestsystembruchlast 4,0 to	44	81	24	50	70	34	86	26	30
Dynamisches Kronensicherungssystem Mindestsystembruchlast 8,0 to	72	71	9	35	44	31	20	77	18
Trag-/ Halteverbindung Mindestsystembruchlast 4,0 to	53	68	90	67	70	66	76	4	3
Trag-/ Halteverbindung Mindestsystembruchlast 8,0 to	63	87	75	57	97	66	45	14	74
Einbau von Kronensicherungen									

Dafür wird eine Zeile für den Pfad zur Verfügung gestellt:

Pfad	:	X	✓	f _x	C:\Users\RenéMartin\Documents\Excel\Excelstammtisch
1	Bitte geben Sie den Speicherpfad an, in dem sich die Vorlagen befinden:				
2	C:\Users\RenéMartin\Documents\Excel\Excelstammtisch\regi - Renes_Excel_Gimmicks				
3					
4					

Der Pfad wird eingelesen:

X	✓	f _x	= Excel.CurrentWorkbook()[[Name="Pfad"]][Content]{0}[Column1]
C:\Users\RenéMartin\Documents\Excel\Excelstammtisch\regi - Renes_Excel_Gimmicks\Excelgimmicks37_PQ_Fehler\04_Vorlagen			

Ein Befehl, beispielsweise `Table.RowCount` wird auf die Liste der Dateinamen angewendet und mit `try` überprüft:

X	✓	f _x	= try Table.RowCount(Folder.Files(Dateipfad))
HasError TRUE			
Error Record			

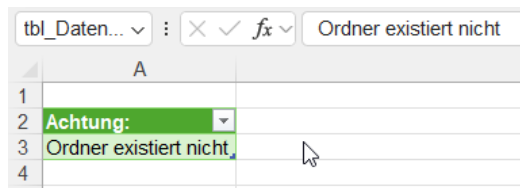
Das Ergebnis ist ein Datensatz mit dem Befehl `HasError`. Der Befehl `Error` ist selbst wieder ein Datensatz, auf dessen Informationen zugegriffen werden kann:

X	✓	f _x	= try Table.RowCount(Folder.Files(Dateipfad))
HasError TRUE			
Error Record			
Reason DataSource.NotFound			
Message File or Folder: Der Ordner "C:\Users\RenéMartin\Documents\Excel\Excelstammtisch\regi - Renes_Excel_Gimmicks\Excelgimmicks37_PQ_Fehler\04_Vorlagen\" wurde nicht gefunden.			
Detail C:\Users\RenéMartin\Documents\Excel\Excelstammtisch\regi - Renes_Excel_Gimmicks\Excelgimmicks37_PQ_Fehler\04_Vorlagen			
Message.Format #[0]: #[1]			
Message.Parameters List			

Es genügt jedoch `HasError`. Damit kann eine Weiche mit einem `If`-Befehl eingebaut werden:

X	✓	f _x	= if Logical.From(Quelle[HasError]) then Ergebnis_FalscherPfad else tbl_Zusammenfassung
Achtung:			
1 Ordner existiert nicht			

Entweder der Ordner existiert – dann wird die Abfrage (hier) tbl_Zusammenfassung aufgerufen. Oder der Pfad existiert nicht – dann wird Ergebnis_FalscherPfad angezeigt. Es handelt sich hierbei um eine kleine Tabelle, welche die Information liefert, dass der Pfad nicht gefunden wurde:



The screenshot shows an Excel spreadsheet with a table named 'tbl_Daten...'. The table has a single column 'A'. In row 2, there is a green cell with a dropdown arrow and the text 'Achtung:'. In row 3, the dropdown menu is open, showing the text 'Ordner existiert nicht'. The status bar at the bottom of the spreadsheet also displays 'Ordner existiert nicht'.

	A
1	
2	Achtung: ▼
3	Ordner existiert nicht
4	

Und warum das Ganze?

Nun: Power Query kennt keine Meldungsfenster. Aber mit dieser Lösung könnte man dem Anwender Informationen über die Auswertung zur Verfügung stellen.