

ELLYPUNK

Applied

WEEKLY DIGEST / T201

28 September - 2 October 2026



This week

28 September - 2 October 2026

Select a title to turn to its pages, or open the original online.

MONDAY / 28 September 2026

Power Query Parameters: Switch Sources Without Editing Steps

3

[Open original online](#)

TUESDAY / 29 September 2026

Python CSV Parsing: Keep Quoted Fields Together

8

[Open original online](#)

WEDNESDAY / 30 September 2026

Python JSON Input: Check Required Keys Before Use

13

[Open original online](#)

THURSDAY / 1 October 2026

Power Query Data Types: Prevent Refresh Errors

18

[Open original online](#)

FRIDAY / 2 October 2026

PowerShell Parameters: Validate Script Inputs Before Running

23

[Open original online](#)

Original artwork is reproduced in full, in publication order.

Copyright Ellypunk. Contains AI-generated artwork.

ellypunk.com

POWER QUERY PARAMETERS: SWITCH SOURCES WITHOUT EDITING STEPS

OBJECTIVE:

Change one source setting, then verify the same query steps still work.

Elly needs to point a Power Query Online dataflow from a test source to a production source without rewriting transformations.

Data Works Better Together

Plan
Build
Test
Improve
Together

AUTOMATE
LEARN
BUILD
GO FURTHER

ONE PARAMETER. ONE PIPELINE. NO CHANGES TO STEPS.

SOURCE PARAMETER (Conceptual)

DataSource

Test



Test Source



Production Source

UNCHANGED QUERY STEPS



Connect



Transform



Shape



Result

Same steps work
for test and production.

RESULT



Verified!
No step changes
needed.

Just change the
source parameter
and refresh. The same
query steps still work!

WHERE THIS RUNS

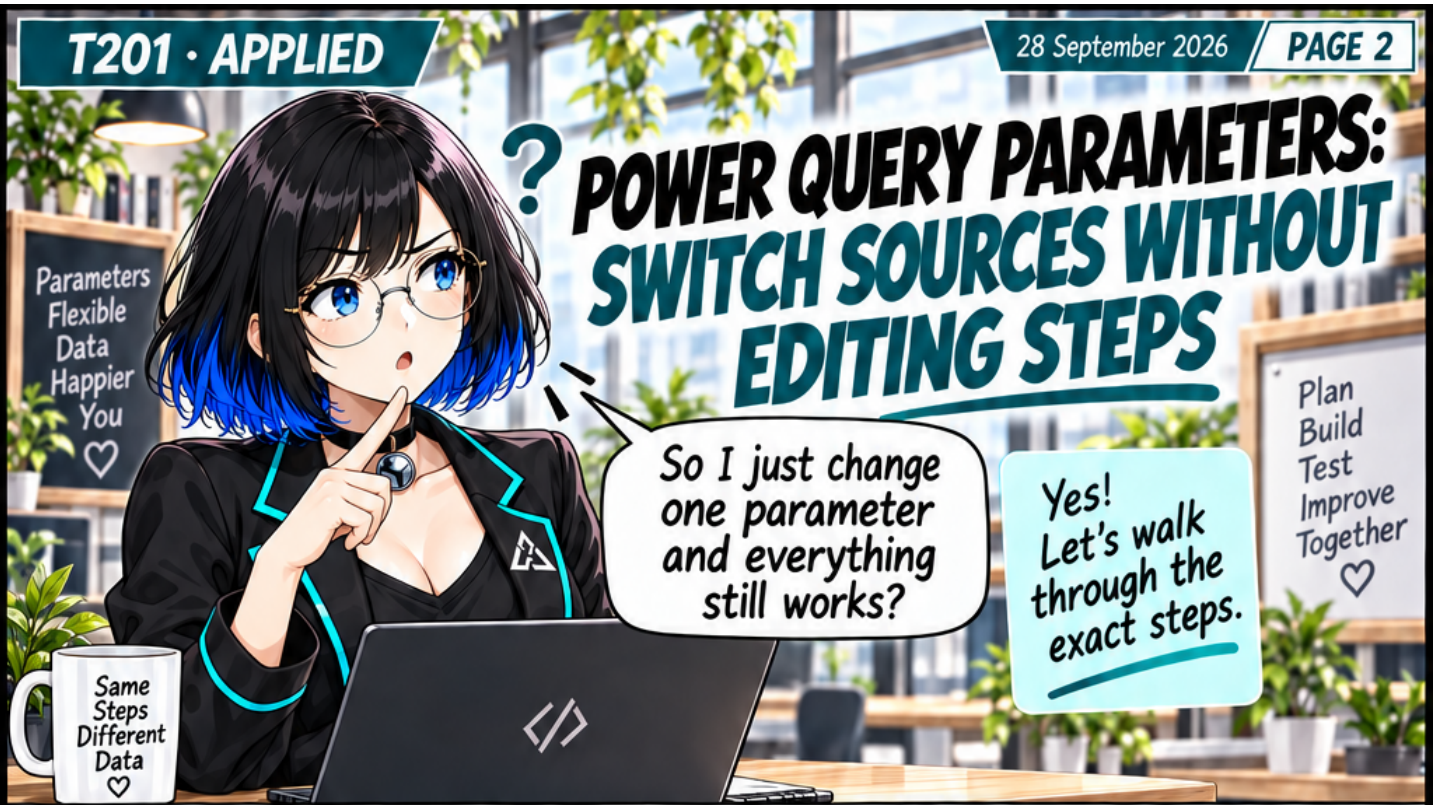


Power Query Online in a browser:
Linux / macOS / Windows
(with an eligible Microsoft service account
and source access).



IMPORTANT

Power BI Desktop is Windows-only.
This lesson uses Power Query Online.
No Desktop UI is shown.



1 Create a Text parameter for the server name.

In the editor, open **Manage Parameters** and create a Text parameter called **ServerName**.

Power Query

Home Transform Add column View

Manage parameters

Manage parameters

+ New parameter

Name:

Type:

Current value:

☒ Required

OK Cancel

2 Use the parameter in the SQL Server connector.

Use that parameter as the SQL Server connector's server value.

Power Query

Home Transform Add column View

Connect to data source

SQL Server

Import data from Microsoft SQL Server database.

Server:

Database (optional):

> Advanced options

OK Cancel

3 Keep the steps unchanged; test with the new source.

Keep the transformation steps unchanged. Test with the new source before relying on refresh.

Power Query

Home Transform Add column View

Queries [1]

Sales

Applied steps

- Source
- Navigation
- Removed Columns
- ABC Changed Type
- Filtered Rows
- Renamed Columns



Same steps work for test and production! Just change the parameter.



! IMPORTANT

A different server can need its own credentials or gateway.

Make sure you have the right permissions and, if required, configure a gateway for the new source.

WHERE THIS RUNS

Power Query Online in a browser: Linux / macOS / Windows (with an eligible Microsoft service account and source access).



No executable shell commands are required. This is done in the Power Query Online UI.

ESSENTIAL GUARDRAILS WHEN SWITCHING A SQL SERVER SOURCE IN POWER QUERY ONLINE

Small Checks Prevent Big Problems

It's not just a setting change... let's check a few important things first.

Change the source with confidence. Validate, don't assume.

Plan Build Test Improve Together

PLAN
CHECK
TEST
REFRESH
SUCCEED

Same Steps Safer Results

1 CONNECTION IDENTITY

Parameter stores a source value, not a password.

The new server needs suitable credentials and, for some private sources, a configured gateway.

Power Query

Connect to data source

SQL Server
Import data from Microsoft SQL Server database.

Server
fx ServerName

Database (optional)



CREDENTIALS

Use appropriate credentials for the new server.



GATEWAY

For some private sources, configure a gateway.

2 SCHEMA SHAPE

Check the new source has the columns and data types the unchanged steps expect.

Does this source have the same columns and data types? If not, the unchanged steps may fail.

Old source (expected)

Column name	Data type
SalesID	Whole number
OrderDate	Date
Customer	Text
Amount	Decimal number
Region	Text



New source (verify)

Column name	Data type
SalesID	Whole number
OrderDate	Date
Customer	Text
Amount	Decimal number
Region	Text



Names and data types should match what the existing steps expect.

3 REFRESH EVIDENCE

Preview first; then run the product's refresh and inspect errors.

Preview looks good? Now run a refresh and check for errors. If something fails, review the error details and fix the source or schema.



Preview
Check the data looks correct.



Run refresh
Use the product's refresh.



Inspect results
Review any errors and address them.

WHERE THIS RUNS



Power Query Online in a browser:
Linux / macOS / Windows
(with an eligible Microsoft service account)



WORKED EXAMPLE

SWITCH A SQL SERVER SOURCE IN POWER QUERY ONLINE

Same Parameter.
Same Steps.
Different Source.

Plan
Check
Refresh
Improve
Together

1 PROBLEM

Sales query points to a test SQL Server.



This query is still pointing to a test SQL Server.

Power Query

Connect to data source

SQL Server
Import data from Microsoft SQL Server database.

Server
test-server FICTIONAL EXAMPLE

Database (optional)
SalesDW

Server names are FICTIONAL examples, not literal network instructions.

2 ACTION

Change Text parameter ServerName from test-server to prod-server.



Update the ServerName parameter to prod-server.

Power Query

Parameters

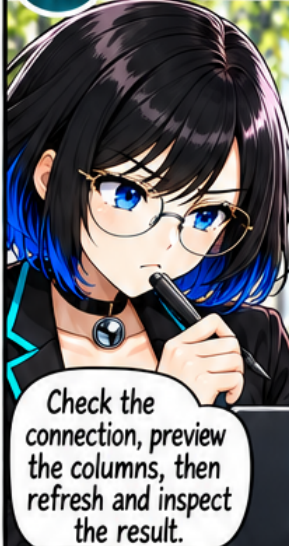
ABC ServerName
Text

prod-server FICTIONAL EXAMPLE

Server names are FICTIONAL examples, not literal network instructions.

3 CHECK

Confirm connection access, preview the expected columns, then refresh and inspect the result.



Check the connection, preview the columns, then refresh and inspect the result.

Power Query

Preview data

123	SalesID	OrderDate	ABC Customer	123	Amount
	1001	2026-09-01	Contoso		1250.00
	1002	2026-09-02	Fabrikam		948.50
	1003	2026-09-02	Northwind		430.00
	1004	2026-09-03	Adventure		2180.00
	1005	2026-09-03	Tailspin		760.25

Refresh

Review the data and check for any errors.

4 IMPROVEMENT

Record the selected source and refresh outcome.



Document the source and refresh outcome.

Source: prod-server
Refresh: Completed
Notes: Data looks correct

Better Data
Brighter Tomorrow

WHERE THIS RUNS



Power Query Online in a browser:
Linux / macOS / Windows
(with an eligible Microsoft service account).



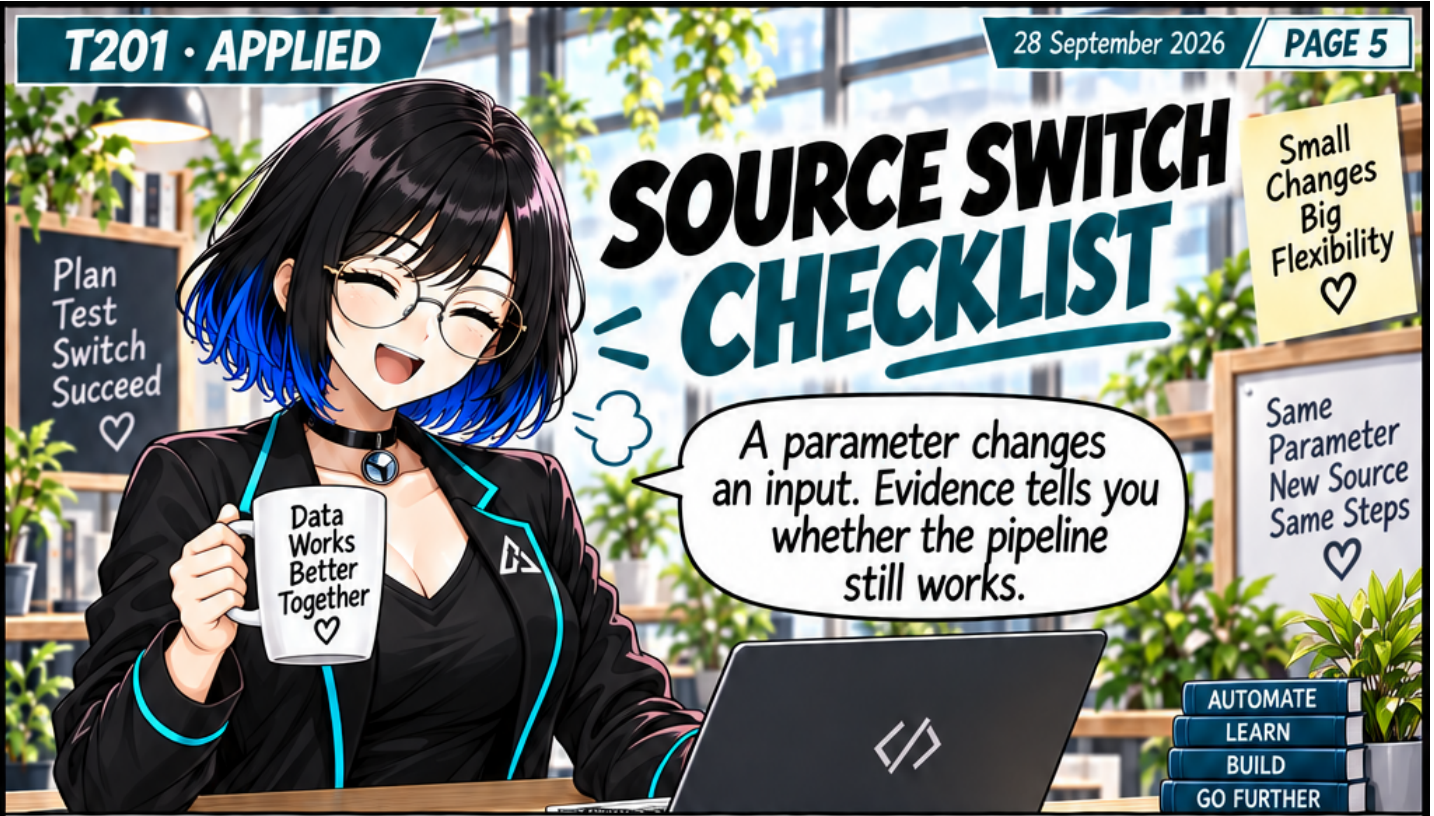
An on-premises SQL Server
may need a gateway.



IMPORTANT

Power BI Desktop is Windows-only.
This lesson uses Power Query Online.
No Desktop UI is shown.





1 Choose approved source.



Pick a source that's approved for your scenario.

DataSource

Production ▼

- Test Source
- Production Source
- Reporting Source

2 Confirm access and gateway.



Make sure you can connect and any gateway is ready.

-  Access confirmed 
-  Gateway available (if needed) 
-  Source reachable 

3 Preview expected columns and types.



Preview the data to confirm the columns and data types look as expected.

Preview data

ABC Customer	OrderDate	123 Amount
Contoso	2026-09-01	1250.00
Fabrikam	2026-09-02	948.50
Northwind	2026-09-02	430.00
Adventure	2026-09-03	2180.00
Tailspin	2026-09-03	760.25

4 Refresh and inspect the result.



Refresh and check the result to confirm everything looks good.



Refresh completed

Rows loaded: 5

The pipeline still works with the new source!

WHERE THIS RUNS



Power Query Online in a browser:
Linux / macOS / Windows,
with eligible account and source access.



Power BI Desktop is Windows-only.



PYTHON CSV PARSING: KEEP QUOTED FIELDS TOGETHER

OBJECTIVE:

Read a CSV row without
splitting a quoted comma.

name,city

"Mira, Jr.", Leeds

Mira, Jr.

Leeds

Two fields, not three.

A CSV parser knows when
a comma is inside quotes.

```

,
,
"Mira, Jr.",
,
,

```



Mira, Jr.	Leeds

QUOTES CHANGE THE COMMA

"Mira, Jr.", Leeds

Part of
the name

Field
separator



Do not use `text.split(",")` for CSV rows.

csv.DictReader
reads the header and
respects quoted fields.



Python 3 • Linux / macOS / Windows

THREE CSV READER RULES



1 · IMPORT



2 · OPEN



3 · READ





WORKED EXAMPLE

1 PROBLEM



2 ACTION



3 CHECK



4 IMPROVEMENT



CSV PARSING RECAP

RECAP:

A quoted comma stays inside one field.
DictReader uses the first row as column names.



CHECK THE RESULT

- ☐ Open with newline set to empty string
- ☐ Read with `csv.DictReader`
- ☐ Inspect name and city separately
- ☐ Check each row has expected fields

ELLY'S FINAL TIP

For "Mira, Jr.", Leeds,
confirm the name is Mira, Jr.
and the city is Leeds.

PYTHON JSON INPUT: CHECK REQUIRED KEYS BEFORE USE

Objective:

Decode external JSON,
then verify the shape
before reading fields.

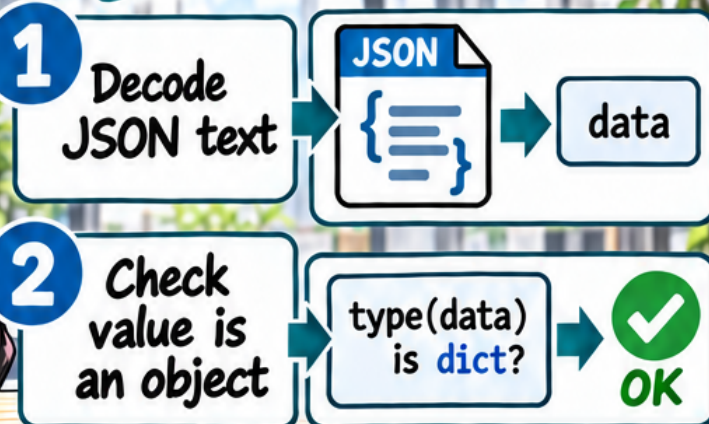


A valid JSON document can still
omit a key your code needs.

```
{"name": "Ari"}
```

❌ Missing: id

Two checks, in order!



First decode the JSON text. Then check the decoded value is an object. A JSON list is valid JSON, but it is not a field map.

✓ JSON object (OK)

```
{
  "name": "web01",
  "settings": {
    "logging": true
  }
}
```

type(data) is dict



✗ JSON list (valid JSON, not a field map)

```
[
  {"name": "web01"},
  {"name": "web02"}
]
```

type(data) is dict



Check required keys

Missing **id**? Reject the input with a clear error before reading it.



Then check types: this lesson expects **id** to be text and **name** to be text.



Work the input

1 PROBLEM

? **Problem:**
the JSON object
has name but
no id.

JSON

```
{  
  "name": "Ari"  
}
```

2 ACTION

Action:
compare its keys
with the required
pair: id and name.

✓ ✗

```
{  
  "id": ...  
  "name": ...  
}
```

```
{  
  "name": ...  
}
```

3 CHECK

Check:
id is missing.
Stop before
reading it.

✗

```
{  
  "name": "Ari"  
}
```

4 IMPROVEMENT

Improvement:
return a clear
validation error.

✗

```
{  
  "name": "Ari"  
}
```


JSON INPUT CHECKLIST

- 1  Decode the text.
- 2  Require an object.
- 3  Require id and name.
- 4  Check the expected types.
- 5  Reject malformed or incomplete input clearly.

JSON

```
{  
  ...  
}
```



FINAL TIP

Check key presence before reading data from an external JSON object.



? POWER QUERY DATA TYPES: PREVENT REFRESH ERRORS

OBJECTIVE:

Inspect incoming column types, then convert with the source locale.

1 THE PROBLEM

A CSV file contains dates as text. The value '22/10/2026' means 22 October 2026 (day/month/year) in the source system.

When Power Query automatically changes the column type using your default locale (month/day/year), it interprets the value as month 22, which is invalid, and returns an Error.

Sales.csv

OrderDate
22/10/2026
23/10/2026
24/10/2026

Date values in the file
(as text)

ERROR!

Power Query interprets '22/10/2026' as month 22 (MM/DD/YYYY) and returns an Error.

2 PREVIEW IN POWER QUERY

In Power Query Editor, you can see the effect after the automatic Changed Type step.

Source

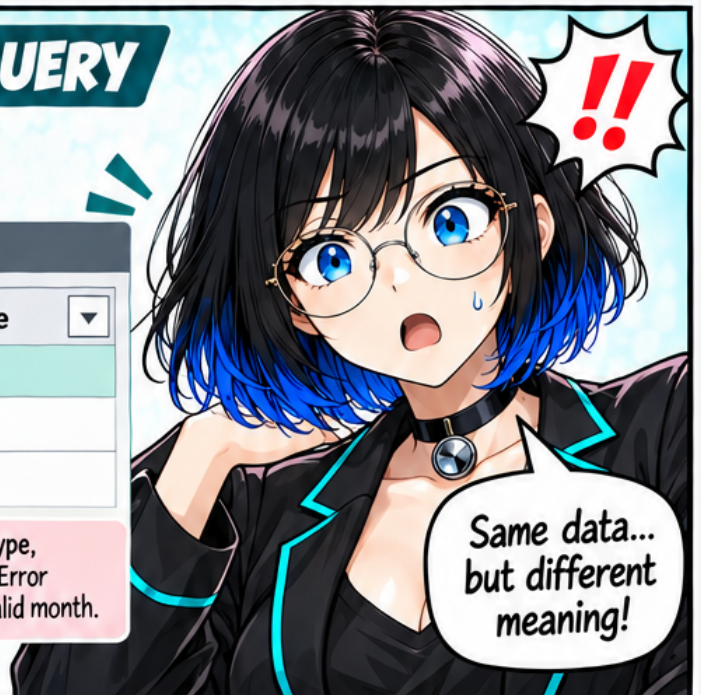
	^B _C OrderDate
1	22/10/2026
2	23/10/2026
3	24/10/2026

Values are text as imported
(from the CSV file).

Changed Type

	OrderDate
1	Error
2	Error
3	Error

After Changed Type,
the dates become Error
because 22 is not a valid month.



Same data...
but different
meaning!

T201 · APPLIED

01 October 2026 / PAGE 2



FIND THE TYPE STEP

CSV or text sources may add an automatic Changed Type step based on sampled values.

1 Open Applied Steps.

In Power Query Editor, open the Applied Steps pane to see the steps for your query.

Applied Steps

Source

Changed Type

Other Steps

2 Select Source, then Changed Type.

Click the Source step, then click the Changed Type step to see how the column type changes.

Applied Steps

Source

Changed Type

Other Steps

3 Compare column icon and values before and after the step.

Check the column icon and the data values to see what changed.

Before (Source)

A ^B C	OrderDate
	22/10/2026
	23/10/2026
	24/10/2026

Type: Text
(values as text)

After (Changed Type)

	OrderDate
	Error
	Error
	Error

Type: Date
(22/10/2026 becomes an Error in this locale)

CSV or text sources may add an automatic Changed Type step based on sampled values.

Review the Applied Steps and the column types to ensure the correct type is used for your data and locale.



 ELLYPUNK | ellypunk.com/training/power-query-data-types-prevent-refresh-errors | © 2026 Ellypunk / NeetOps

Issue 4 / original page 2 of 5

19 / 27

T201 · APPLIED

01 October 2026

PAGE 3

CONVERT WITH THE SOURCE LOCALE

1 REMOVE OR EDIT THE INCORRECT CHANGED TYPE STEP

In Power Query Editor, check the Applied Steps. Remove or edit the Changed Type step that is causing the error.

2 SELECT ORDERDATE, THEN CHANGE TYPE → USING LOCALE

Select the OrderDate column, then go to Transform → Data Type → Using Locale.

3 CHOOSE DATE AND ENGLISH (UNITED KINGDOM)

In the Change Type with Locale dialog, select Date and English (United Kingdom) for DD/MM/YYYY source text.

4 RESULT: DATE CONVERTS CORRECTLY

With the correct source locale, the date is parsed as a valid date.

Before (default locale)			After (using source locale)		
	Abc	OrderDate			OrderDate
1	22/10/2026	Error	1	22	October 2026
2	23/10/2026	Error	2	23	October 2026
3	24/10/2026	Error	3	24	October 2026

CAUTION

Use the source format, not the viewer's display locale. Date formats and display settings vary by region and can be different on your system.

T201 · APPLIED

01 October 2026 / PAGE 4



WORKED EXAMPLE

PREVENT REFRESH ERRORS

Problem: wrong locale causes date errors, and some values aren't dates at all.

1 THE SOURCE

A CSV file has dates stored as text.

	OrderDate
1	22/10/2026
2	23/10/2026
3	NA

The values are text, not dates.

2 ACTION

Set the column type to Date using English (United Kingdom).



Change column type

Type: Date
Locale: English (United Kingdom)

Use the correct locale, then check for errors.

3 CHECK

After the change, valid dates parse, but non-dates remain as errors.

	OrderDate
1	22 October 2026
2	23 October 2026
3	⚠ Error

These parse correctly as dates.

'NA' is not a date, so it remains an error.

4 IMPROVEMENT

Test a refresh on a new file. Verify the row count and error count.

- ✓ Refresh completes.
- ✓ Row count matches expectation.
- ✓ Review the number of errors.
- ✓ Decide how to handle errors (e.g. correct the source or apply an explicit replacement policy).

Don't silently replace all errors with an arbitrary date.



Correct data types and locale prevent refresh errors — but you still need to check for invalid values and handle them intentionally.

REFRESH CHECKLIST

Check the new-file refresh for type errors.

1 Inspect automatic Changed Type after import.

Column	Changed Type
OrderDate	 Date
Customer	A ^B C Text
Amount	1.2 Decimal Number



Confirm each column has the expected type after import.

2 Choose Date using the source locale.

Date 

Locale

English (United Kingdom) 



Set the Date type with the correct source locale (day/month/year or month/day/year).

3 Inspect error rows; correct invalid source values deliberately.

OrderDate	Status
22/10/2026	Valid
31/02/2026	Error
15/10/2026	Valid



Filter error rows, review the original values, and correct invalid source data deliberately.

4 Test a new-file refresh; compare row count and error count.



Rows: 1,246
Errors: 3

Rows: 1,246
Errors: 3



After refresh, check the row count and error count match (or are as expected).

RECAP:

Correct locale fixes interpretation; it does not turn NA into a date.

ELLY'S FINAL TIP

Keep one known day-first test row in your refresh check.



POWERSHELL PARAMETERS: VALIDATE INPUTS BEFORE RUNNING

OBJECTIVE:

Reject a bad input
before the script
does work.

1 SCRIPT INPUT

User provides
a parameter value.



2 VALIDATION

Check the input
against rules (type,
format, range, etc).



3 SAFE ACTION

If valid, the script
continues and does
the work.



If the input is invalid,
stop before doing any work.



Invalid input
is rejected.
No further action
takes place.

Reject invalid
input before any
work begins.

VALIDATION RUNS AT BINDING



PowerShell validates parameter input during the binding phase, before your script starts running.

1 CALL SCRIPT WITH A VALUE

Call script with a value.



2 VALIDATION CHECKS

PowerShell checks validation attributes.



3 ON FAILURE, NO EXECUTION

On failure, body does not run.



If validation fails, PowerShell rejects the input before any of your script code runs.



A default value is not checked as caller input.

Validation happens at binding, so invalid input is rejected before any work begins.



CHOOSE THE RIGHT VALIDATOR

1 [ValidateSet("Dev", "Prod")]

Allows only listed strings for the parameter value.

```
param(
    [ValidateSet("Dev", "Prod")]
    [string] $Environment
)
```

2 [ValidateRange(1, 10)]

Allows an integer 1 through 10.

```
param(
    [ValidateRange(1, 10)]
    [int] $RetryCount
)
```

3 INVALID VALUE IS REJECTED

```
param(
    [ValidateRange(1, 10)]
    [int] $RetryCount
)
```

script body would go here
Write-Host "Running..."

User provides an invalid value:

Caller input: RetryCount = 25



Rejected before the script body runs.



Declare validation attributes before the typed variable.

4 SAME SYNTAX ACROSS PLATFORMS

Use the same PowerShell 7 validation syntax on Linux / macOS / Windows when pwsh is installed.



Linux



macOS



Windows



Same PowerShell 7 syntax on all platforms.



WORKED EXAMPLE: ENVIRONMENT

SCRIPT

```
1 param(
2     [ValidateSet("Dev", "Prod")]
3     [string]$Environment
4 )
```

1 CALL WITH INVALID VALUE

PowerShell 7

Caller input: Environment = Test
Cannot validate argument on parameter 'Environment'.
The argument "Test" does not belong to the set "Dev, Prod" specified by the ValidateSet attribute. Supply an argument that is in the set and then try the command again.



The script body writes no output for the rejected value.

2 CALL WITH VALID VALUE

PowerShell 7

Caller input: Environment = Dev
Environment set to: Dev



The value is valid and the script continues.
The script body can run.

3 RESULT SUMMARY



KEY POINTS

- The ValidateSet attribute checks the input before the script body runs.
- Invalid values are rejected with a clear error.
- The script body does not run for rejected input.
- Valid values are accepted and the script can run.



SYNTAX

PowerShell 7 · Linux / macOS / Windows
(pwsh installed)



PARAMETER VALIDATION CHECKLIST

1 SET ALLOWED VALUES

Choose the allowed type or set.

- ✓ Dev
- ✓ Prod
- ✗ Other...



2 TEST INPUTS

Test one valid and one invalid input.

Dev



Test



3 CONFIRM REJECTION

Confirm invalid input stops before script body.



RECAP

`ValidateSet('Dev','Prod')` rejects 'Test'.
It accepts 'Dev'.



'Dev' ✓



'Test' ✗

ELLY'S FINAL TIP

Set a validation rule
on the parameter, then
test the rejection path.