

## Delta Tables: Understanding the Two Patterns in Fabric Spark for creating Delta Tables

### 1. Save as a table (MANAGED table only)

#### Example

```
df.write.format("delta").saveAsTable("managed_products")
```

#### Result

- A managed table is created in the Lakehouse metastore
- Delta files are stored under the default managed table location

The screenshot displays the Microsoft Fabric workspace interface. On the left, the 'Explorer' pane shows the 'OneLake' hierarchy with 'El\_Lakehouse' expanded, containing 'Tables' and 'Files'. The 'managed\_products' table is highlighted under 'Tables'. The main workspace area shows a command execution history. The first command, executed in PySpark (Python), is `df.write.format("delta").saveAsTable("managed_products")`, which completed successfully. Below it, a Spark SQL command `DESCRIBE FORMATTED managed_products;` is shown, also successful. At the bottom, a 'Table view' for 'managed\_products' is displayed, showing 3 columns and 11 rows of metadata.

ABC_col_name	ABC_data_type
1 ProductID	int
2 ProductName	string
3 Category	string
4 ListPrice	double
5	
6 # Detailed Tab...	
7 Name	spark_catalog.chimcoblh92ajbp41musjbedo62or54l2monqcc5lmaq3felpma9b4c9ng.managed_products
8 Type	MANAGED
9 Location	abfss://05dd6b73-f930-4b8d-a2fb-34d884c23a8d@onelake.dfs.fabric.microsoft.com/d8b1dbae-b810-409f-ac22-44fda04b18b1/Tables/dbo/managed_products
10 Provider	delta

### 2. Save as a table and write Delta files to specified path (MANAGED table and Delta Files)

#### Example

```
df.write.format("delta").saveAsTable("external_products",  
path="abfs_path/external_products")
```

#### Result

- A managed table is created in the Lakehouse metastore
- Delta files are stored under the default managed table location
- Executes the write operation and writes Delta files to the path provided

Explorer

Data items Resources Connections

+ Add data items

OneLake

ELakehouse

Tables

dbo

external\_products

managed\_products

Files

external\_products

products

Other people in your organization may have access to notebooks and Spark job definitions in this workspace. Carefully review this item before running it.

```
1 df.write.format("delta").saveAsTable("external_products", path="abfss://05dd6b73-f930-4b8d-a2fb-34d884c23a8d@onelake.dfs.fabric.microsoft.com")
```

[4] ✓ 24 sec - Command executed in 24 sec 4 ms by Salim Lawal on 21/02/2026, 06:06:48 PySpark (Python)

Spark jobs (7 of 7 succeeded) Resources

```
1 %%sql
2 DESCRIBE FORMATTED external_products;
```

[5] ✓ 1 sec - Command executed in 2 sec 184 ms by Salim Lawal on 21/02/2026, 06:07:53 Spark SQL

Table view

Download Filter by keyword

ABC col_name	ABC data_type	ABC con
1 ProductID	int	NULL
2 ProductName	string	NULL
3 Category	string	NULL
4 ListPrice	double	NULL
5		
6 # Detailed Tab...		
7 Name	spark_catalog.chimcoblhdq2ajbp41rmusjbedo62or54l2monqcc5lmaq3felpma9b4c9ng.external_products	
8 Type	MANAGED	
9 Location	abfss://05dd6b73-f930-4b8d-a2fb-34d884c23a8d@onelake.dfs.fabric.microsoft.com/d8b1dbae-b810-409f-ac22-44fda04b18b1/Tables/dbo/external_prod...	
10 Provider	delta	

Inspect

Fabric does NOT allow external tables created via Spark. It ignores the `path=` argument in `saveAsTable()` and does not link the table to the folder created in the specified path.

Whereas, in **Databricks** it works. Databricks interprets this as:

- Create a metastore table
- Store its data on the custom path
- → Result, TYPE: external table

Eventhough the Microsoft Learn says external tables should be created this way.

### Create an external table

You can also create external tables, which may be stored somewhere other than the lakehouse, with the schema metadata stored in the lakehouse.

1. In the Explorer pane, in the ... menu for the **Files** folder, select **Copy ABFS path**. The ABFS path is the fully qualified path to the lakehouse Files folder.
2. In a new code cell, paste the ABFS path. Add the following code, using cut and paste to insert the `abfs_path` into the correct place in the code:

```
Code Copy
```

```
df.write.format("delta").saveAsTable("external_products", path="abfs_path/external_products")
```

3. The full path should look similar to this:

```
Code Copy
```

```
%%sql
DESCRIBE FORMATTED external_products;
```

4. Run the cell and in the results, view the Location property for the table. Widen the Data type column to see the full path and notice that the OneLake storage locations ends with `/Files/external_products`.

Why Fabric behaves differently?

Fabric Spark enforces:

- **Managed tables only** inside a Lakehouse
- No external tables is created via Spark
- No custom locations for Spark tables

This is because Fabric uses a **unified metastore** and tightly controls table locations for governance, lineage, and OneLake consistency.