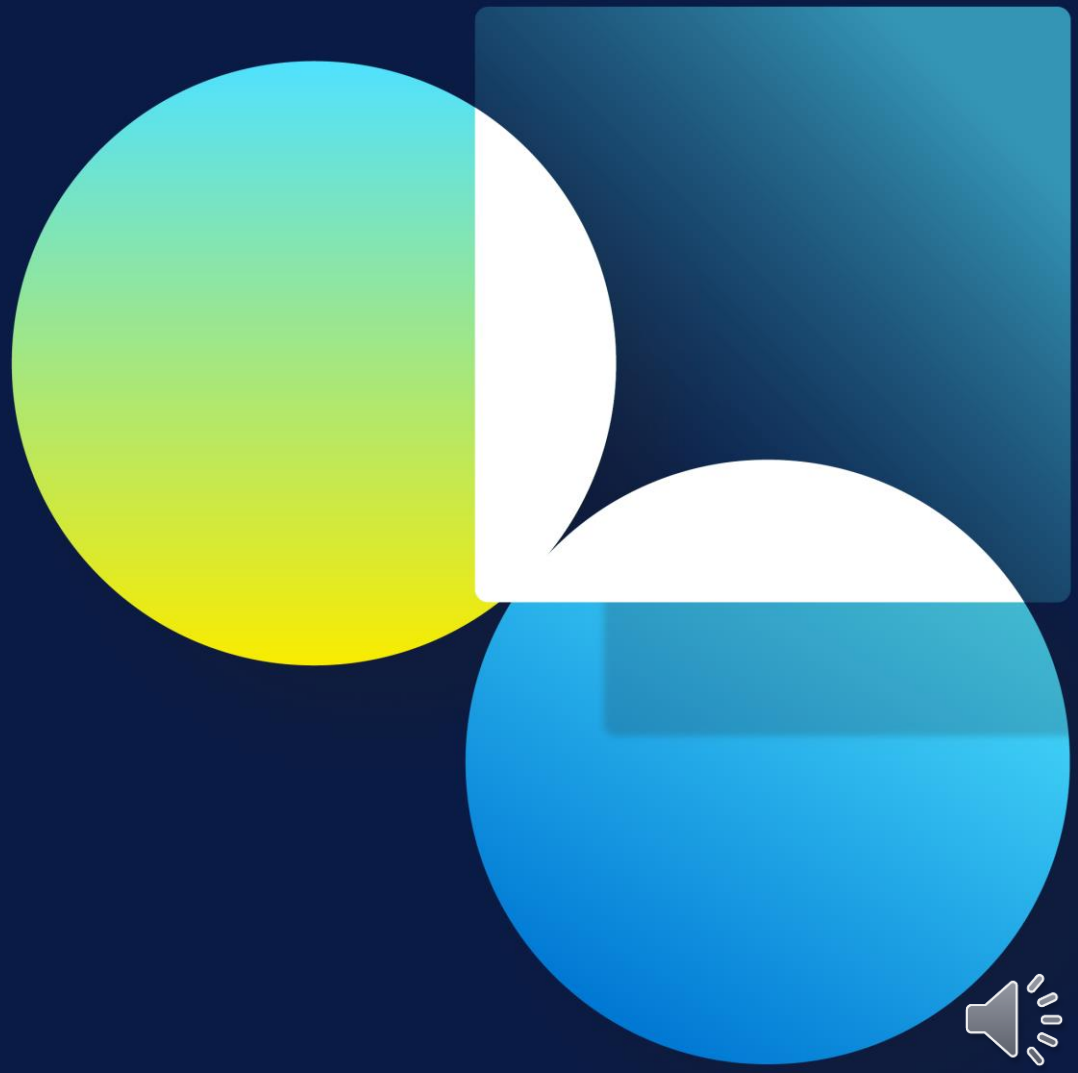
Microsoft Dynamics 365

Intelligent order management - Power Query Transformations

Dynamics 365 FastTrack
Architecture Insights Series

Steven Koppens
Senior FastTrack Solution Architect



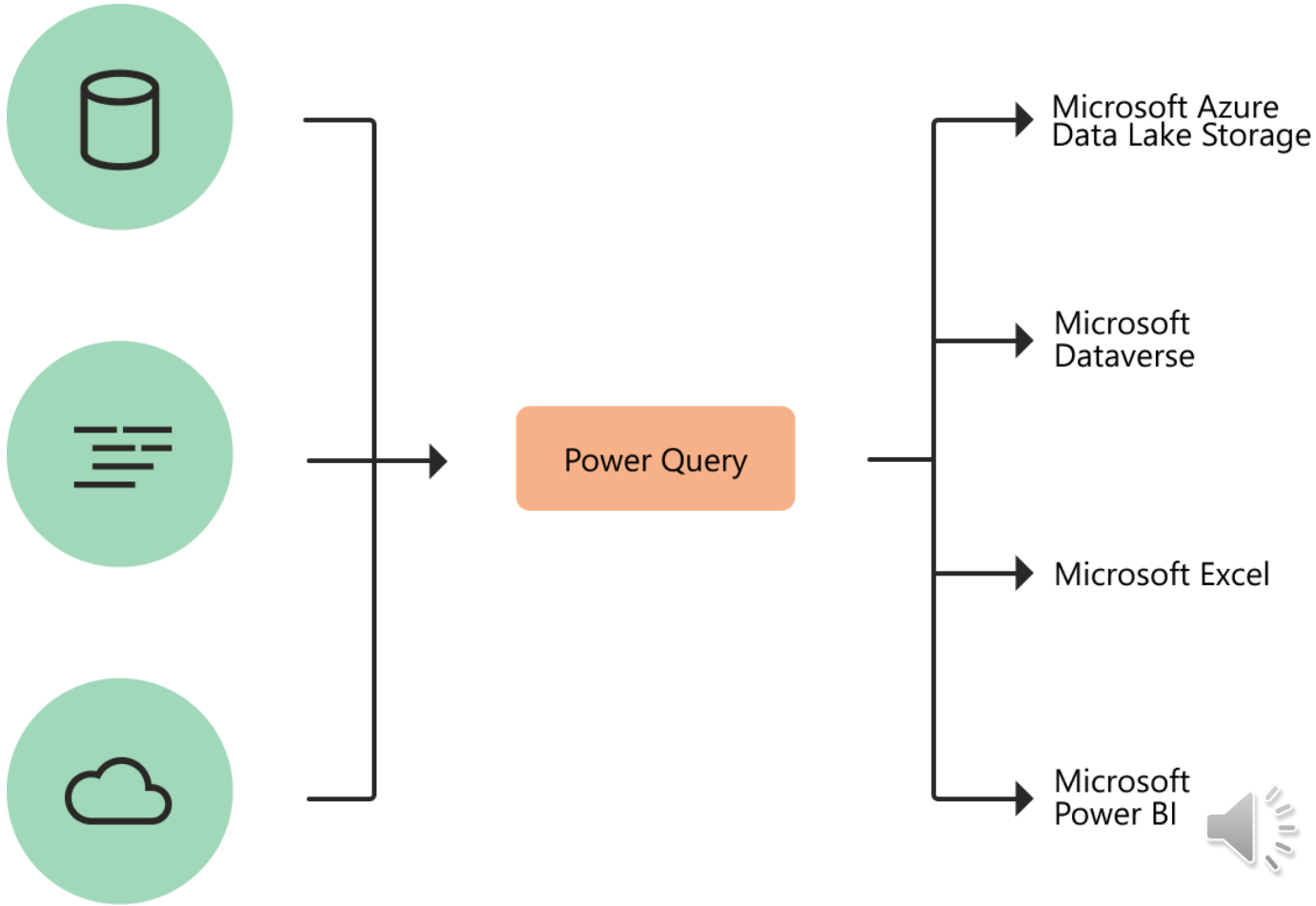
Agenda

- Power Query Overview
- Power Query and IOM
- Scenarios
- Debugging
- Resources

Power Query Overview

Power Query Overview

- Data transformation and data preparation engine
- Dataflows



Power Query Overview – M formula language

Power Query M

```
let
    Source = Exchange.Contents("xyz@contoso.com"),
    Mail1 = Source{[Name="Mail"]}[Data],
    #"Expanded Sender" = Table.ExpandRecordColumn(Mail1, "Sender", {"Name"}, {"Name"}),
    #"Filtered Rows" = Table.SelectRows(#"Expanded Sender", each ([HasAttachments] = true)),
    #"Filtered Rows1" = Table.SelectRows(#"Filtered Rows", each ([Subject] = "sample files for email PQ test"
    #"Removed Other Columns" = Table.SelectColumns(#"Filtered Rows1",{"Attachments"}),
    #"Expanded Attachments" = Table.ExpandTableColumn(#"Removed Other Columns", "Attachments", {"Name", "Attac
    #"Filtered Hidden Files1" = Table.SelectRows(#"Expanded Attachments", each [Attributes]?[Hidden]? <> true
    #"Invoke Custom Function1" = Table.AddColumn(#"Filtered Hidden Files1", "Transform File from Mail", each #
    #"Removed Other Columns1" = Table.SelectColumns(#"Invoke Custom Function1", {"Transform File from Mail"}),
    #"Expanded Table Column1" = Table.ExpandTableColumn(#"Removed Other Columns1", "Transform File from Mail",
    #"Changed Type" = Table.TransformColumnTypes(#"Expanded Table Column1",{{"Column1", type text}, {"Column2"
in
    #"Changed Type"
```

Power Query and IOM

Looking at Power Query and IOM

- ImportMappingKey

```
[
    entityname1 = {
        [
            // Lookup #1
        ],
        [
            // Lookup #2
        ],
        ...
    },
    entityname2 = {
        // Lookups for entityname2
    },
    ...
]
```

Looking at Power Query and IOM

- Entity names:
  - account
  - salesorder
  - currency

```
shared ImportMappingKey = [
    account = {
        [
            ExternalRecordKey = [IOMAccountMapping =
"unknown@email.com"],
            SelectedFields = { "accountid"}
        ],
        [
            ExternalRecordKey = [IOMAccountMapping =
Source[billingAddress][email]],
            SelectedFields = { "accountid"}
        ]
},
```

Looking at Power Query and IOM

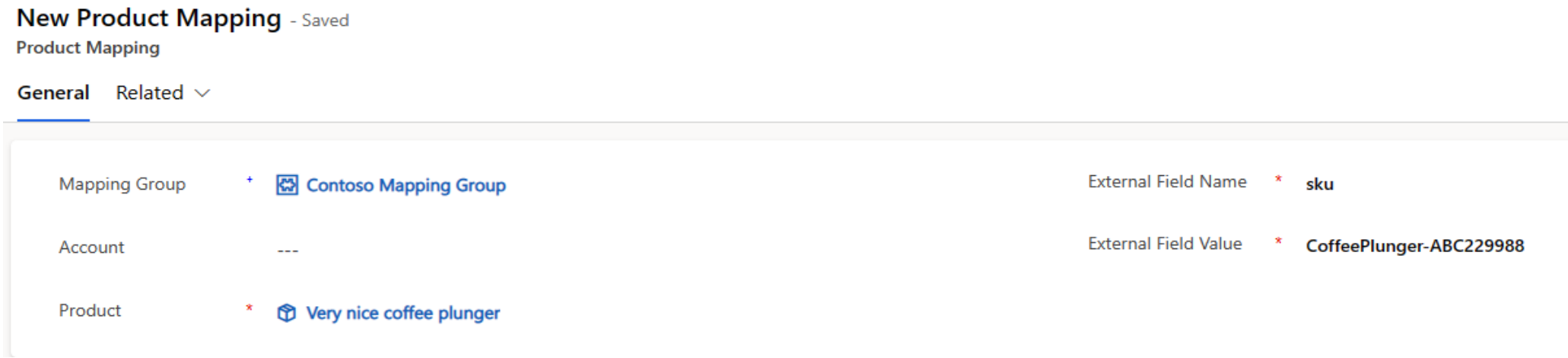
- Lookup Format
  - External to Internal

Example of an external record

```
{
    "productid": “CoffeePlunger-ABC229988",
    "name": “Very nice coffee plunger",
    "version": 4,
    "description": “Nice description."
}
```

Example of using 1 value as a key

```
[
    product = {
        [
            ExternalRecordKey = [sku = "CoffeePlunger-ABC229988"]
        ]
    }
]
```



Looking at Power Query and IOM

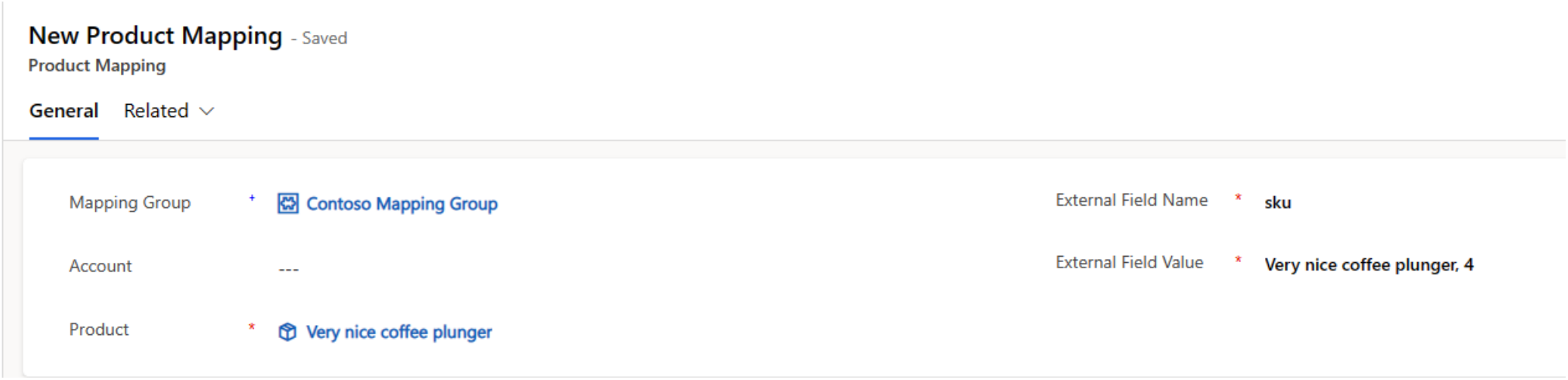
- Lookup Format
  - External to Internal

Example of an external record

```
{
    "productid": “CoffeePlunger-ABC229988",
    "name": “Very nice coffee plunger",
    "version": 4,
    "description": “Nice description."
}
```

Example of using 2 values as a key

```
[
    product = {
        [
            ExternalRecordKey = [
              sku = "Very nice coffee plunger", 4]
        ]
    }
]
```



Looking at Power Query and IOM

- SelectedFields

```
shared ImportMappingKey = [
    account = {
        [
            ExternalRecordKey = [IOMAccountMapping =
"unknown@email.com"],
            SelectedFields = { "accountid"}
        ],
        [
            ExternalRecordKey = [IOMAccountMapping =
Source[billingAddress][email]],
            SelectedFields = { "accountid"}
        ]
},
```

Looking at Power Query and IOM

- TransformSourceData

```
shared TransformSourceData =
    let
        leadtime = 5,
        useWriteInProductEnabled = Source[useWriteInProductEnabled],
        isDualWriteEnabled = Source[isDualWriteEnabled],
        orderProducts = Source[orderdetails],
        billingAddress = Source[billingAddress],
        shippingAddress = Source[shippingAddress],
        account =IOM.MapRecord(IOM.MappingTables[account],
[IOMAccountMapping = Source[billingAddress][email]]),
```

Looking at Power Query and IOM

- IOM.MappingTables
- IOM.MapRecord

```
IOM.MappingTables[Entity]
IOM.MapRecord(EntityTable, MapKey)
```

```
shared TransformSourceData = let
        account = IOM.MapRecord(IOM.MappingTables[account], [IOMAccountMapping =
Source[billingAddress][email]])
in
        account[accountid];
```

Looking at Power Query and IOM

- Lookup Format
  - Internal to External

```
shared ImportMappingKey =
    [
        salesorder = {
            [
                InternalRecordId = Source[salesorderid]
            ]
        }
    ];
```

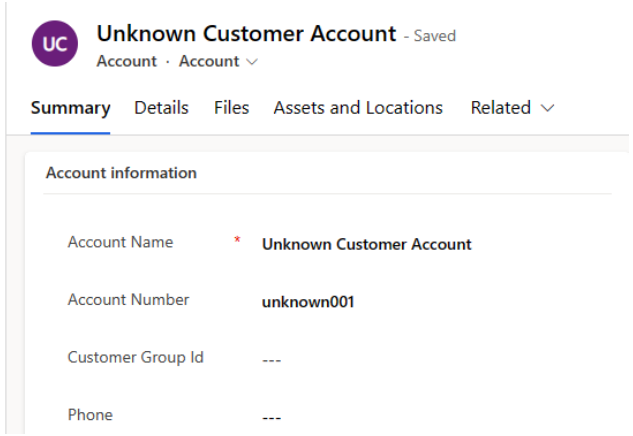
```
shared TransformSourceData = let
        externalOrder = IOM.MapRecord(IOM.MappingTables[salesorder],
[InternalRecordId = Source[salesorderid]])
in
    externalOrder[orderID];
```

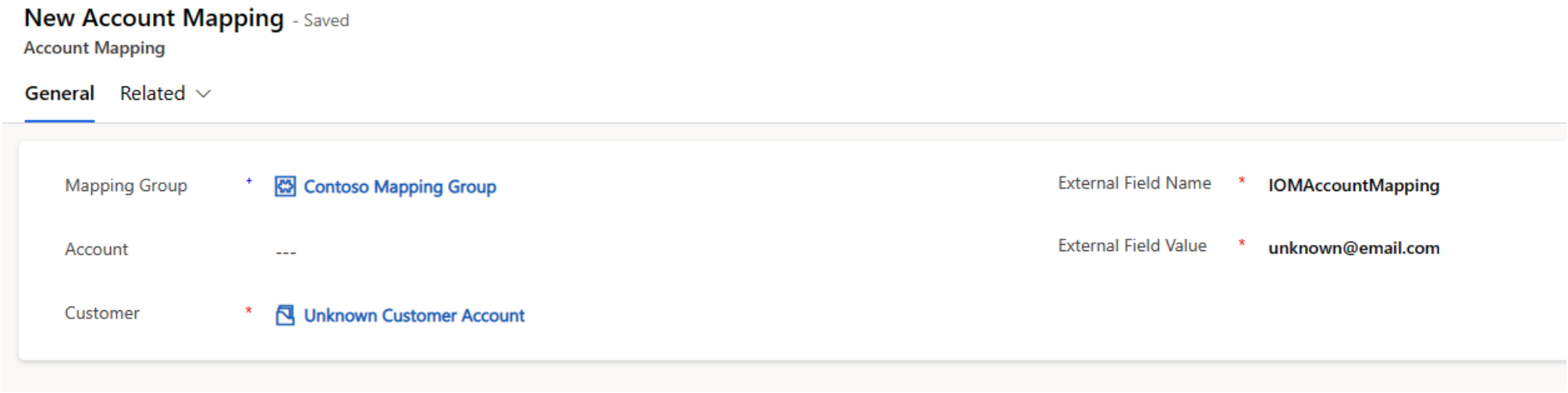
Scenarios

Scenario 1
Missing or incorrect customer identifier

Data in the message is missing or incorrect

- A customer submits an order with an incorrect e-mail





Data in the message is missing or incorrect

- A customer submits an order with an incorrect e-mail

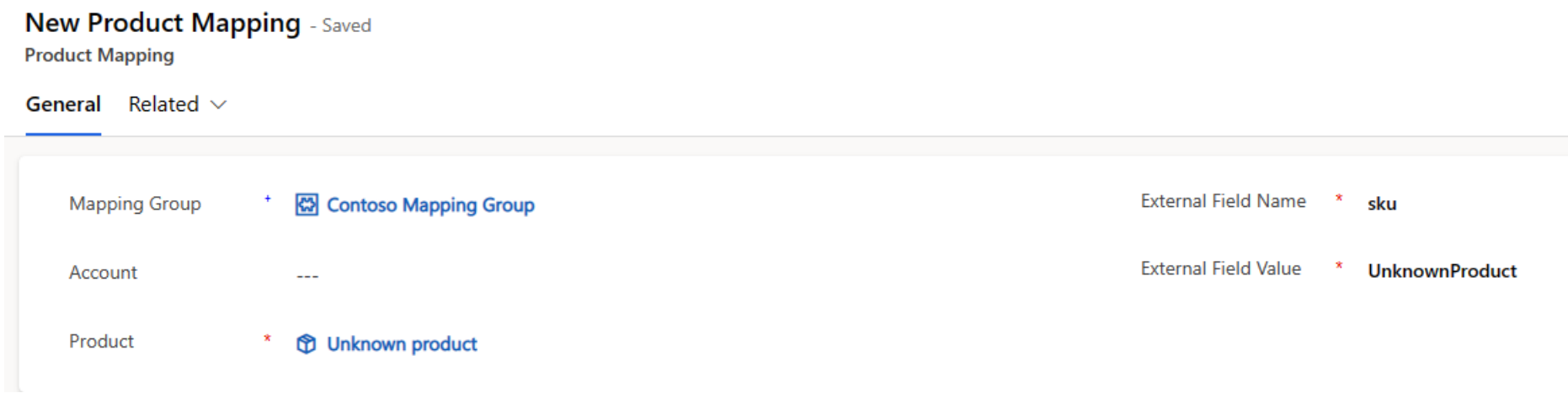
Sample Transformation

```
shared ImportMappingKey = [
account = {
        [
              ExternalRecordKey =  [IOMAccountMapping = "unknown@email.com"],
              SelectedFields = {"accountid"}
        ],
        [
            ExternalRecordKey = [IOMAccountMapping = Source[billingAddress][email]],
            SelectedFields = {"accountid"}
        ]
},

…

account =IOM.MapRecord(IOM.MappingTables[account], [IOMAccountMapping = Source[billingAddress][email]]),
resolvedAccount =
            if Record.FieldCount(account) > 0
            then account
            else IOM.MapRecord(IOM.MappingTables[account], [IOMAccountMapping = "unknown@email.com"]),

…

#"customerid_account@odata.bind" = "/accounts(" & Text.From(resolvedAccount[accountid]) & ")",
```

Scenario 2
Missing or incorrect product identifier

Data in the message is missing or incorrect

- A customer submits a message with an incorrect product ID



Data in the message is missing or incorrect

- A customer submits a message with an incorrect product ID

```
product = List.InsertRange(List.Distinct(List.Transform(Source[orderdetails], each
[
        ExternalRecordKey = [sku = _[sku]],
        SelectedFields = { "productid"}
])), 0, {
[
            ExternalRecordKey = [sku = "UnknownProduct"],
            SelectedFields = { "productid"}
]
        }
),
```

Data in the message is missing or incorrect

- A customer submits a message with an incorrect product ID

```
shared MapRecordOrDefault = (mappingTable as table, mappedRecordKey as record,
defaultKey as record) =>
    let
        mappedRecord = IOM.MapRecord(mappingTable, mappedRecordKey),
        resolvedRecord =
            if Record.FieldCount(mappedRecord) > 0
            then mappedRecord
            else IOM.MapRecord(mappingTable, defaultKey)
    in
        resolvedRecord;
```

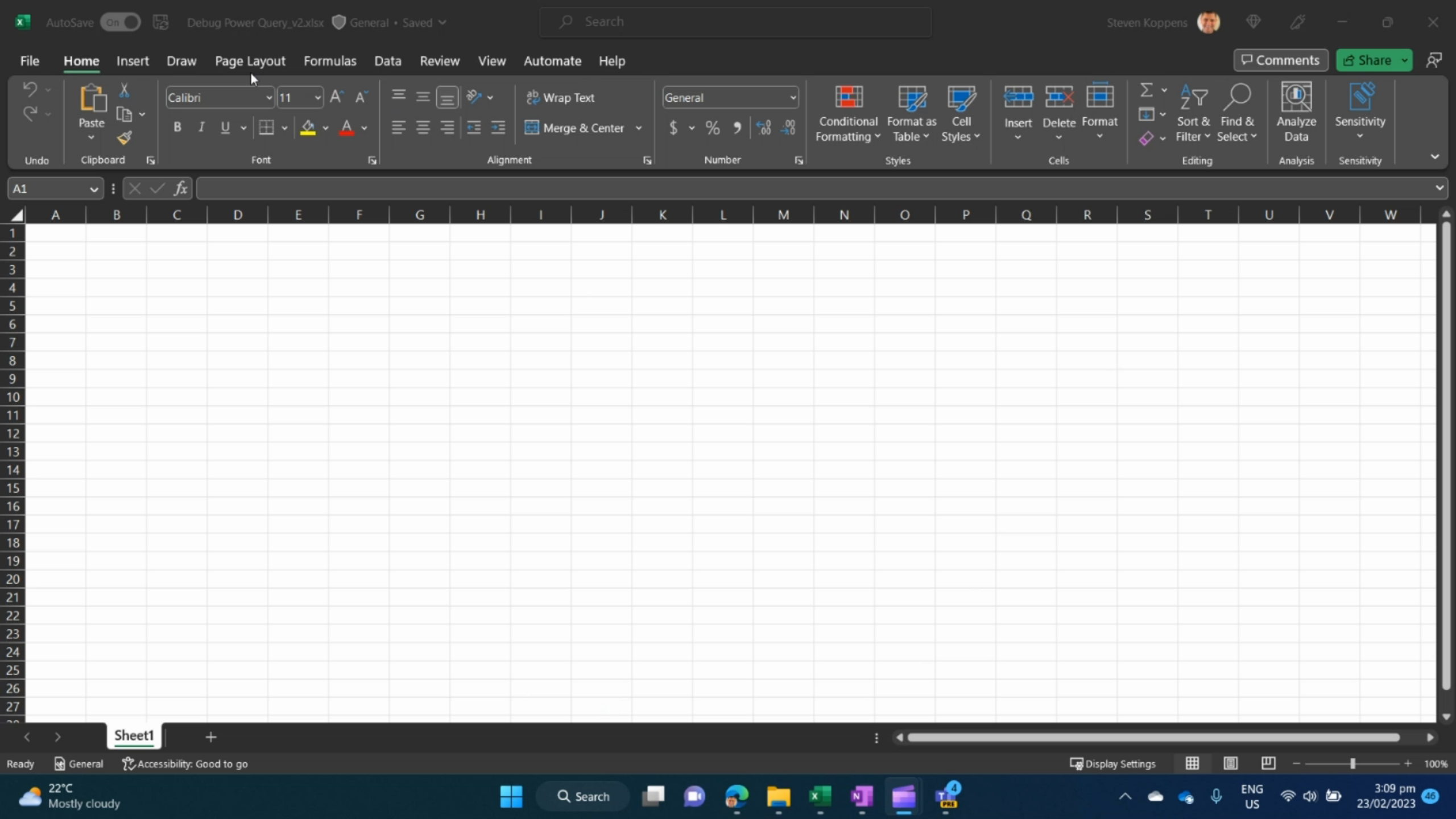
Data in the message is missing or incorrect

- A customer submits a message with an incorrect product ID

```
Record.ToTable
(
let
    productOrDefault = MapRecordOrDefault(IOM.MappingTables[product], [sku = Record.FieldOrDefault(_, "sku",
"UnknownProduct")], [sku = "UnknownProduct"])
in
[
     ispriceoverridden = true,
     #"productid@odata.bind" = try "/products(" & productOrDefault[productid] & ")" otherwise null,
```

Debugging

Scenario 1
Using Power Query Editor in Excel


AutoSave On
Debug Power Query_v2.xlsx
General • Saved
Search
Steven Koppens
File Home Insert Draw Page Layout Formulas Data Review View Automate Help
Comments
Share
Undo
Paste
Clipboard
Calibri
11
Font
Wrap Text
Merge & Center
Alignment
General
Number
Conditional Formatting
Format as Table
Cell Styles
Styles
Insert
Delete
Format
Cells
Sort & Filter
Find & Select
Editing
Analyze Data
Analysis
Sensitivity
Sensitivity
A1
Sheet1
Ready
General
Accessibility: Good to go
Display Settings
100%
22°C
Mostly cloudy
Search
ENG
US
3:09 pm
23/02/2023

Scenario 2
Using IOM Provider Transformer

Objects

Search

- Action Type Input B... (72)
- Adaptive Card Confi... (2)
- AI models (5)
- App profile (3)
- Application Extension (4)
- Application ribbons (1)
- Application Tab Tem... (8)
- Apps (12)
- Article templates (4)
- Business Event Def... (262)
- Capacity Profile (1)
- Catalog (24)
- Catalog Assignment (249)
- CatalogEventStatusC... (7)
- Channel Integration ... (1)
- Chatbot subcompo... (12)
- Chatbots (0)
- Choices (321)
- **Cloud flows (61)**

New | Publish all customizations | ...

Search

Default Solution > **Cloud flows**

Display name	Name	Type	Managed	Last Modif...	Owner	Status
(Deprecated)IOMActionEventUpdateHanc	(Deprecated)IO...	Cloud Flow	Yes	1 week ago	SA Solutions Arc...	On
Flow to create an IOM order when HTTP r	Flow to create an...	Cloud Flow	No	1 month ago	Steven Koppens	On
IOM Create Fulfillment Order Mapping	IOM Create Fulfil...	Cloud Flow	Yes	2 months ago	SA Solutions Arc...	On
IOM Create Fulfillment Order Product Ma	IOM Create Fulfil...	Cloud Flow	Yes	2 months ago	SA Solutions Arc...	On
IOM Create Order Mapping	IOM Create Orde...	Cloud Flow	Yes	2 months ago	SA Solutions Arc...	On
IOM Get External Mapping	IOM Get External...	Cloud Flow	Yes	2 months ago	SA Solutions Arc...	On
IOM Get Internal Mapping	IOM Get Internal...	Cloud Flow	Yes	2 months ago	SA Solutions Arc...	On
IOM Provider Transformer	IOM Provider Tra...	Cloud Flow	Yes	1 month ago	SA Solutions Arc...	On
IOM Raise Business Event	IOM Raise Busin...	Cloud Flow	Yes	2 months ago	SA Solutions Arc...	On
IOM Raise Lines And Aggregated Order E	IOM Raise Lines ...	Cloud Flow	Yes	1 month ago	SA Solutions Arc...	On
IOM Raise Sales Order Lines Business Eve	IOM Raise Sales ...	Cloud Flow	Yes	2 months ago	SA Solutions Arc...	On
IOM Sales Order Creation	IOM Sales Order ...	Cloud Flow	Yes	1 week ago	SA Solutions Arc...	On

Resources

Resources

- Power Query Online: Intelligent Order Management leverages the capabilities of Power Query and the M query language to support transformation in the data pipeline for providers. To learn more, see the Power Query documentation.
- More information: What are dataflows?
- More information: Power Query M formula language
  - M Language Functions - PowerQuery M | Microsoft Learn
  - List.InsertRange - PowerQuery M | Microsoft Learn

Thank you

Microsoft Dynamics 365

© Copyright Microsoft Corporation. All rights reserved.