Mapping Documentation

This document describes how mappings are defined and used in DataMigratePro.

Mapping overview

Mappings describe how source tables and fields are transformed into destination fields. They are stored in the mapping database and can be generated or edited through the configuration workflows.

Calculation field

The **Calculation** column allows SQL expressions to transform source data. Expressions are injected into generated SQL and can reference the source table alias `T` and join aliases (`T18`, `T42`, etc.).

Placeholders

The following placeholders are supported in **Calculation** expressions:

- `{Company}` → replaced with the source company name.
- `{FieldName}` → replaced with the current **Source Field Name**, including brackets.

`{Company}` can also be used in **Join Clause** expressions to limit results in company-spanning tables (tables that contain data for multiple companies). This ensures joins only pick rows for the active source company.

Example (remove domain + \ from a user id field):

```
CASE WHEN CHARINDEX('\\',T.{FieldName})>0 THEN SUBSTRING(T.{FieldName}, CHARINDEX('\\',T.
{FieldName})+1,100) ELSE T.{FieldName} END
```

If **Source Field Name** is `User ID`, the generated SQL becomes:

```
CASE WHEN CHARINDEX('\\',T.[User ID])>0 THEN SUBSTRING(T.[User ID], CHARINDEX('\\',T.[User ID])+1,100)
ELSE T.[User ID] END
```

If **Calculation** is empty, the field is selected as `T.[User ID]`.

Source Field Name

The **Source Field Name** column must match the field name in the source table. It is used for:

- direct field selection when no calculation is provided
- `{FieldName}` placeholder expansion

Where Clause

The **Where Clause** column allows row-level filtering in generated SQL. Use it to migrate only records that match specific conditions.

The expression is injected into the source query and can reference the source table alias `T` and join aliases (`T18`, `T42`, etc.).

Typical usage

- limit migrated records to active or relevant data
- exclude invalid or placeholder values
- combine with **Join Clause** for company-specific filtering

Example:

```
T.[Notify] = 1 AND ISNULL(T.[To User ID], '') <> ''
```

Generated SQL (excerpt):

```
FROM [Demo Database NAV (6-0)].dbo.[Record Link] T
WHERE T.[Notify] = 1 AND ISNULL(T.[To User ID], '') <> ''
```

Example (Item table, option field `Status`):

```
T.[Status]<>3
```

In this example, option value `3` means **Inactive**, so the filter excludes inactive items.

Generated SQL (excerpt):

```
FROM [Demo Database NAV (6-0)].dbo.[Item] T
WHERE T.[Status]<>3
```

Example: {Company} in Join Clause + Calculation

The following mapping rows demonstrate how `{Company}` is used in the **Join Clause** to filter company-spanning tables, how **Calculation** can concatenate fields, and how **Where Clause** can restrict migrated records.

Configuration (excerpt):

Row 1 (URL1)

Column	Value
Source Table No.	2000000068
Source Table Name	Record Link
Source Field No.	3
Source Field Name	URL1
Source Field Type	Text
Destination Table No.	2000000068
Destination Table Name	Record Link
Destination Field No.	3
Destination Field Name	URL1
Destination Field Type	Text2048
Skip	No
Primary Key	No
Joined Table	0
Join Clause	
Where Clause	T.[Notify] = 1
Calculation	T.[URL 1]+T.[URL 2]+T.[URL 3]

Row 2 (Company)

Column	Value
Source Table No.	2000000068
Source Table Name	Record Link
Source Field No.	12
Source Field Name	Company
Source Field Type	Text
Destination Table No.	2000000068

Column	Value
Destination Table Name	Record Link
Destination Field No.	12
Destination Field Name	Company
Destination Field Type	Text30
Skip	No
Primary Key	No
Joined Table	2000000006
Join Clause	T2000000006.[Name]='{Company}'
Where Clause	T.[Notify] = 1
Calculation	

Generated SQL (excerpt):

```
;WITH T AS (
    SELECT CAST(SUBSTRING(CAST(T.[timestamp] AS VARBINARY(8000)), 1, 8) AS BIGINT) AS [timestamp],
           ROW_NUMBER() OVER(ORDER BY T.[Link ID]) [RowNumber],
           T.[Link ID] AS [Link ID],
           T.[Record ID] AS [Record ID],
           T.[URL 1]+T.[URL 2]+T.[URL 3][URL1],
           T.[Description] AS [Description],
           T.[Type] AS [Type],
           T.[Note] AS [Note],
           LEFT(CONVERT(varchar(50), CASE WHEN ISDATE(T.[Created]) = 1 AND T.[Created] NOT IN ('1753-
01-01', '1900-01-01') THEN CAST(T.[Created] AS datetime) ELSE NULL END, 127), 19) AS [Created],
           T.[User ID] AS [User ID],
           T.[Company] AS [Company],
           T.[Notify] AS [Notify],
           T.[To User ID] AS [To User ID]
    FROM [Demo Database NAV (6-0)].dbo.[Record Link] T
    JOIN [Demo Database NAV (6-0)].dbo.[Company] T2000000006 WITH (NOLOCK) ON T2000000006.
[Name]='CRONUS AG'
    WHERE T.[Notify] = 1
)
SELECT T.*, 0 [Type_of_Change]
FROM T
WHERE T.[RowNumber] BETWEEN 1 AND 10000000
```

How it comes together:

- The **Join Clause** uses `{Company}` so the join is scoped to the active source company. At runtime, `{Company}` is replaced with the source company name (here: `CRONUS AG`).
- The **Joined Table** value `2000000006` results in a join alias `T2000000006`, which is referenced in the join clause.
- The **Where Clause** filters the source rows before final paging/selection, so only matching records are migrated.
- The **Calculation** concatenates three source fields into one destination field (`URL1`).

Value mapping

Value mapping replaces a source value with a destination value based on a mapping code. It uses the **BC Value Mapping** and **BC Value Mapping Value** tables and is activated per field via the **Value Mapping Code** column in **BC Migration Mapping**.

Example: Language Code (table 18 / Customer)

Configuration (excerpt):

Table	Column	Value
BC Migration Mapping	Source Table No_	18
BC Migration Mapping	Source Field Name	Language Code
BC Migration Mapping	Value Mapping Code	LANGUAGE

Table	Column	Value
BC Value Mapping	Code	LANGUAGE
BC Value Mapping Value	Value Mapping Code	LANGUAGE
BC Value Mapping Value	Source Value	ENU
BC Value Mapping Value	Destination Value	ENG

Generated SQL (excerpt):

```
;WITH T AS (
    SELECT CAST(SUBSTRING(CAST(T.[timestamp] AS VARBINARY(8000)), 1, 8) AS BIGINT) AS [timestamp],
           ROW_NUMBER() OVER(ORDER BY T.[No_]) [RowNumber],
           T.[No_] AS [No_],
           T.[Name] AS [Name],
           VM1.[Destination Value] AS [Language Code]
    FROM [Demo Database NAV (6-0)].dbo.[CRONUS AG$Customer] T
    LEFT JOIN [Demo Database NAV (6-0)].dbo.[BC Value Mapping Value] VM1 ON VM1.[Value Mapping Code] =
'LANGUAGE' AND VM1.[Source Value] = T.[Language Code]
)
SELECT T.*, 0 [Type_of_Change]
FROM T
```

How it comes together:

- The **Value Mapping Code** on the mapping row causes the query generator to add a `LEFT JOIN` to **BC Value Mapping Value**.
- The field projection switches from `T.[Language Code]` to `VM1.[Destination Value]` and keeps the original destination field name.
- If the value mapping tables are missing or empty, the join is not added and the source value is selected directly.

Example: Dimension mapping (Dimension Code present)

If the **BC Value Mapping** record has a **Dimension Code**, the mapping is applied in the dimension CTE instead of via a join.

Configuration (excerpt):

Table	Column	Value
BC Migration Dimension	Table ID	21
BC Migration Dimension	Dimension Code	DEPARTMENT
BC Value Mapping	Code	DEPT
BC Value Mapping	Dimension Code	DEPARTMENT
BC Value Mapping Value	Value Mapping Code	DEPT
BC Value Mapping Value	Source Value	SALES
BC Value Mapping Value	Destination Value	SAL

Generated SQL (excerpt):

```
;WITH T AS (
    SELECT ...
), D AS
(
   SELECT T.[Entry No_],
          MAX(COALESCE(CASE WHEN [Dimension Code]='DEPARTMENT' THEN
              (SELECT TOP 1 VMV.[Destination Value]
               FROM [Demo Database NAV (6-0)].dbo.[BC Value Mapping Value] VMV
               WHERE VMV.[Value Mapping Code]='DEPT'
                 AND VMV.[Source Value]=T.[Dimension Value Code])
          ELSE '' END,'')) [DimensionValue(DEPARTMENT)]
     FROM [Demo Database NAV (6-0)].dbo.[CRONUS AG$Dimension Set Entry] T
    WHERE [Table ID]=21
 GROUP BY T.[Entry No_]
)
```

```
SELECT T.*, D.[DimensionValue(DEPARTMENT)]
FROM T
LEFT JOIN D ON T.[Entry No_]=D.[Entry No_]
```

How it comes together:

- The **Dimension Code** on **BC Value Mapping** links the mapping to a dimension.
- The generator injects a subquery into the dimension CTE to translate dimension values.
- If no mapping exists for a specific source value, the expression yields an empty string for that dimension value.

Why Table No. is configurable

Table No. scopes a value mapping to a specific source table. This avoids accidental reuse of a mapping code across tables that use the same values but require different target values.

Example: Same code, different targets per table

Configuration (excerpt):

Table	Column	Value
BC Value Mapping	Code	STATUS
BC Value Mapping	Table No.	18
BC Value Mapping Value	Source Value	B
BC Value Mapping Value	Destination Value	Blocked
BC Value Mapping	Code	STATUS
BC Value Mapping	Table No.	23
BC Value Mapping Value	Source Value	B
BC Value Mapping Value	Destination Value	Inactive

How it comes together:

- Table 18 (Customer) uses `STATUS` to translate `B` → `Blocked`.
- Table 23 (Vendor) can use the same `STATUS` code but translate `B` → `Inactive`.
- Without **Table No.**, both tables would share the same mapping and the translation could be wrong for one of them.