

SFDX Data Move Utility 485 80

The Advanced Salesforce Data Migration Tool

Search the knowledge base



GET STARTED NOW

The SFDX Data Move Utility (SFDMU) is the top-notch modern salesforce data migration solution.

You will get your Scratch or Sandbox populated with quality records just in minutes.

- ✓ The SFDMU will help you to fill your Scratch / Dev / Sandbox / Prod org with records imported from another org or CSV files.
- ✓ It can perform all CRUD operations (Insert / Update / Upsert / Delete) also for multiple related sObjects.
- ✓ It's the comprehensive All-In-One data migration solution having a bulk of unique features have never provided before by any of the existing tools.
- ✓ This Plugin is approved and digitally signed by Salesforce.

[Get Started](#)



[View On GitHub](#)



[GUI Application](#)



Why do you need the SFDMU ?

Because the Plugin is equipped with an extended built-in functionality making migration of multiple related objects very handy:

- 👉 It supports direct Org-to-Org data migration without having to deal with annoying intermediate CSV files.
- 👉 Full support of CRUD operations: Insert / Update / Upsert / Delete.
- 👉 Supports import and export of CSV files.
- 👉 Supports multiple object sets allowing you to execute multiple migration jobs in one run.
- 👉 Supports simultaneous migration of multiple SObjects in one run..
- 👉 Handles complex circular references between multiple SObjects.
- 👉 Simple and user-friendly configuration just with a [single export.json file](#)
- 👉 Can be run from a command line, can be easily integrated into any DevOps pipeline.

- 👉 Supports data [migration preserving Record Type](#) for each record.
- 👉 Supports [fully customized Fields and Object Mapping](#).
- 👉 Has the [built-in ETL features](#), including [customized Values Mapping](#), [expression-based records transformation](#).
- 👉 Supports migration of **Person Accounts**, including **mixed account records of Busines and Person type**.
- 👉 Supports migration of [Salesforce Files](#) as well as old-fashioned [Notes](#), [Attachments](#), including [file binary data](#).
- 👉 Supports record owner assignment, including [option for a default owner assignment](#).
- 👉 Has the [built-in data anonymization feature](#) to anonymize a sensitive source data (e.g. coming from Production org).
- 👉 Supports binding of [polymorphic lookup fields](#) (e.g. FeedItem.ParentId).
- 👉 **Secured and safe.** All operations are performed on the client's machine, there is no cloud interaction, so all is completely safe.
- 👉 **Fast performance.** Processes only a selected subset of records and fields that need to be inserted or updated and does not touch others.
- 👉 The core functionality can be extended by using the [Custom Add-On Api](#). You can create your own Add-On for your special purposes.

SFDMU Overview

- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

Advanced features

- [Composite External ID keys](#)
- [Compound Fields](#)
- [Data Anonymization](#)
- [DeleteHierarchy operation](#)
- [DeleteSource operation](#)
- [Export of the Notes, Attachments and Files](#)
- [Field Multiselect](#)
- [Fields Mapping](#)
- [Handling of record OwnerId](#)
- [Handling of the RecordType](#)
- [Multiple object sets](#)
- [Target Records Filter](#)
- [Values Mapping](#)

lacking the real data critical for developing and debugging applications. Traditionally, loading data manually for each new org using tools like Salesforce Data Loader is inefficient, especially since data can be imported directly from production or another sandbox.

Often, the challenge is not just importing data but modifying specific records or fields within a complex data model with many dependencies. Additionally, maintaining a clean data model without adding external ID fields to SObjects requires creative solutions for updating records in your developer org.

A significant issue arises with the inconsistent IDs across environments when populating the Recordtypeid field. Currently, there's no effective tool for inserting and updating dependent objects based on a unique field used as an External ID. Typically, developers resort to using Excel to manage source tables, which can be cumbersome and error-prone for even a small number of objects.

The SFDX Data Move Utility (SFDMU) offers a viable solution for these challenges.

Watch the demo. The Plugin in action:

Salesforce DX Data Move Utility (SFDMU) - Command line ...



SFDMU OVERVIEW

[Key Features of the SFDMU](#)

[Get Started](#)

[Installation](#)

[Configuration](#)

[Running](#)

[Debugging](#)

[SFDMU GUI Application](#)

FULL DOCUMENTATION

[!\[\]\(fe3aebe81acea8d45108cd2768939da7_img.jpg\) Export.json File Specific](#)

[!\[\]\(626ce8ac21792b9405bfddfea8e0c96a_img.jpg\) Advanced features](#)

[!\[\]\(a8f9309f944226d1420f5fed22e2b6e6_img.jpg\) Add-On API](#)

[!\[\]\(248b91fcdac4810ffd15cf33fb6aec6f_img.jpg\) Reports](#)

EXAMPLES

Key Features of the SFDMU.

SFDMU (SFDX Data Move Utility) offers a comprehensive set of features designed to facilitate efficient and secure data migration between Salesforce organizations, as well as from and to CSV files:

- **Direct Org-to-Org Migration:** Eliminates the need for intermediate CSV files by supporting direct data transfer.
- **CRUD Operations:** Fully supports Create (Insert), Read (Update), Upsert, and Delete operations.
- **CSV File Handling:** Enables both import and export of data via CSV files.
- **Multiple Object Sets:** Allows the execution of multiple migration jobs in a single run, managing several SObjects simultaneously.
- **Complex Reference Handling:** Effectively manages circular references between multiple SObjects and self-referenced fields, such as `Account.ParentId`.
- **Configuration Ease:** Utilizes a single, user-friendly `export.json` file for configuration.
- **DevOps Integration:** Can be run from the command line, integrating seamlessly into any DevOps pipeline.
- **Advanced ID Handling:** Supports external ID fields of any type, including formula and auto-named fields, as well as composite external ID keys.
- **Record Type Preservation:** Migrates data while maintaining record types.

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

Get Started.

Table of Contents:

- [Overview](#)
- [Step 1. Install the SFDMU](#)
- [Step 2. Configure the Migration Job](#)
 - [Basic Configuration Example: Upserting Accounts](#)
 - [Advanced Configuration Example: Upserting Multiple Related SObjects](#)
 - [Important Notes](#)
- [Step 3. Run the Migration Job](#)
- [Demonstration Video](#)

Overview

The **SFDX Data Move Utility (SFDMU)** is a powerful and versatile tool designed to facilitate complex data migration and synchronization processes between Salesforce organizations. As a fully-featured, open-source solution, SFDMU offers significant

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

Installation.

Table of Contents:

- [Prerequisites.](#)
- [Installation.](#)
 - [Installation as a SFDX Plugin](#)
 - [Installation from Cloned Source Code \(for Debugging\)](#)

Prerequisites.

Before you install and use the SFDMU Plugin, ensure that the [SF CLI](#) is installed on your system.

If you wish to use the source code of the SFDMU Plugin, you must also install [Node.js](#) and [Git](#).

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [!\[\]\(5774573cf757c446bb08af21f46b2969_img.jpg\) Export.json File Specific](#)
- [!\[\]\(a502cb21d600ba28a5cdf414d68eef89_img.jpg\) Advanced features](#)
- [!\[\]\(b90ad4352d6e82333440a21dde15d657_img.jpg\) Add-On API](#)
- [!\[\]\(c887fe1bc1f2363e586d4073ecf6e4e9_img.jpg\) Reports](#)

EXAMPLES

Configuration.

Table of Contents:

- [How to Configure SFDMU](#)
- [Basic Configuration: Upserting Accounts](#)
- [Advanced Configuration: Upserting Multiple Related SObjects](#)

How to Configure SFDMU

SFDMU can be configured using a single JSON file named **export.json**. This file specifies the objects and fields to be migrated, focusing on fields that are updatable in the target metadata.

Basic Configuration: Upserting Accounts

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [!\[\]\(750841ae7100dc832cb0a4b3af4492f3_img.jpg\) Export.json File Specific](#)
- [!\[\]\(78e449f8a1164b81ecbd00cd97498e27_img.jpg\) Advanced features](#)
- [!\[\]\(9931ff4a747d4e6edc8cfe9a6d936949_img.jpg\) Add-On API](#)
- [!\[\]\(d06bd4b5386b5ebdee91452b0403e593_img.jpg\) Reports](#)

EXAMPLES

Running.

Table of Contents:

- [Simplest Way to Run the SFDMU](#)
- [Run the SFDMU as an SFDX Plugin](#)
- [Full Format of the CLI Command](#)
- [Available CLI Command flags](#)
 - [--sourceusername, -s:](#)
 - [--targetusername, -u:](#)
 - [--apiversion:](#)
 - [--canmodify, -c:](#)
 - [--filelog, -l:](#)
 - [--json:](#)
 - [--logfullquery:](#)
 - [--loglevel:](#)
 - [--noprompt, -n:](#)
 - [--nowarnings, -w:](#)
 - [--path, -p:](#)
 - [--verbose:](#)
 - [--concise:](#)
 - [--quiet:](#)
 - [--silent:](#)

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

Debugging.

How to Debug SFDMU's Source Code

Debugging the SFDMU plugin involves setting up the local environment with the source code from its GitHub repository.

Setup Instructions

1. Install the SFDMU Plugin from the cloned source code following [these instructions](#).
2. **Configure VSCode for Debugging:**
 - Open VSCode and go to "File" => "Preferences" => "Settings".
 - Enable "Node Debug" by setting "Auto Attach" to "On".
 - Turn "Use Preview Auto Attach" to "Off".
3. **Initiate Debug Session:**
 - In the source code directory, run the debug command:

SFDMU OVERVIEW

[Key Features of the SFDMU](#)

[Get Started](#)

[Installation](#)

[Configuration](#)

[Running](#)

[Debugging](#)

[SFDMU GUI Application](#)

FULL DOCUMENTATION

 [Export.json File Specific](#)

 [Advanced features](#)

 [Add-On API](#)

 [Reports](#)

SFDMU GUI Application.

👉 The Application GitHub repository can be found here: [SFDMU GUI App Application Repository](#)

Table of Contents:

[Overview](#)

[When is SFDMU GUI Application Useful?](#)

[Supported OS](#)

[Installation of the Release Version](#)

[General Information](#)

[Prerequisites](#)

[Installation of Latest Release](#)

[Step 1. Download](#)

[Step 2. Install](#)

[Installation of Specific Release](#)

[Step 1: Select the Release](#)

[Step 2: Select the Release Tag](#)

[Step 3: Download the Setup Files](#)

[Step 4: Install the Application](#)

[Running the Installed Application](#)

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

How do I adjust the batch size for a Bulk API Query?

Question

I'm unable to find an option to control the batch size for bulk select queries and would like to adjust this. Currently, I need to increase the node memory limit to 16GB to select 700,000 Accounts, and I'm not sure if this will be effective.

Answer

For the **Bulk API Query**, there is no technical option to adjust the batch size or perform memory optimization during runtime. Your options include [increasing the node memory size](#) or limiting the size of the fetched records with `LIMIT` or `WHERE` clauses.

For DML operations, the batch size for **Bulk API V1.0** can be controlled using the `bulkApiV1BatchSize` parameter. For more details, see the [Export.json File Specification](#).

Last updated on 23rd Apr 2024

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

Why doesn't Bulk API v1 Parallel Mode create batches asynchronously?

Question

I am using the "**concurrencyMode": "Parallel"** setting in my export.json, but I notice that batches are created sequentially, not in parallel:

```
1st batch created @ 15:33:20. Completed 15:33:39  
2nd batch created @ 15:33:40. Completed 15:33:57  
3rd batch created @ 15:33:58. Completed 15:34:15
```

I expected SFDMU to create batches simultaneously with "concurrencyMode": "Parallel".

Answer

In the current implementation, SFDMU does not create and send bulk batches in parallel threads. The "**concurrencyMode": "Parallel"**" setting instructs Salesforce Bulk API to

SFDMU OVERVIEW

[Key Features of the SFDMU](#)

[Get Started](#)

[Installation](#)

[Configuration](#)

[Running](#)

[Debugging](#)

[SFDMU GUI Application](#)

FULL DOCUMENTATION

 [Export.json File Specific](#)

 [Advanced features](#)

 [Add-On API](#)

 [Reports](#)

EXAMPLES

What are the limitations on data volume processing?

No, there are no special limitations for this.

Basically, the plugin can process any number of records.

However, it's important to be aware that SFDMU is designed to transfer only small to medium data volumes to seed sandboxes or scratch orgs with limited demo data needed for developers. It is highly NOT recommended to use SFDMU for transferring entire large orgs.

When dealing with large data volumes, you may face various issues due to the API limits of the Salesforce environments and Node.js memory issues. There are no affordable out-of-the-box solutions for these issues.

If you are facing such issues, try:

- Optimizing queries in the **export.json** file.
- For Node.js memory issues, see: [I have an JavaScript heap out of memory error. How to increase?](#)

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [!\[\]\(8ad79f5e68f2cb9a3d09e9f5525e1847_img.jpg\) Export.json File Specific](#)
- [!\[\]\(7dc9b5077f5dabe133f87fbb17be996a_img.jpg\) Advanced features](#)
- [!\[\]\(1e8978482d0425bc074087510f7709e3_img.jpg\) Add-On API](#)
- [!\[\]\(06d4914abce9eb315d111cdbcdeddc03_img.jpg\) Reports](#)

EXAMPLES

Can I include referenced fields, like Account.Name into the query string?

Question:

I want to avoid including the **Account** object in `export.json` but still migrate the **Name** of the **Account** referenced by a given **Contact**. For example, like this:

```
{  
    "query": "SELECT Id, Account.Name FROM Contact",  
    "operation": "Upsert",  
    "externalId": "Name"  
}
```

Can I do this?

Answer:

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

Is there a limitation on fields used in composite external IDs in SFDMU?

Question:

I am using the Composite External Id feature in SFDMU and specifying a large number of fields in the externalId fields chain for each SOQL query.

Are there any limitations on the number of external IDs that can be used in a composite external ID in these queries?

Answer:

While SFDMU itself does not impose a strict limit on the number of fields you can include in a composite external ID, the practical limitation comes from the Salesforce API.

Each field in the composite external ID chain increases the length of the SOQL query, which is sent as a GET request. Salesforce REST API has a maximum URI length limit, which could be exceeded if too many fields are used.

Therefore, it's advisable to carefully select only the necessary fields for your composite external IDs to avoid potential issues with URI length.

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

What do master and slave objects mean?

Question:

The [documentation](#) mentions setting **master=false** for child objects to select only those related to previously selected parent objects.

1. Can you explain how this works and whether multiple "master" objects can be defined?
2. Can I specify multiple "master" objects ?

Answer:

Regarding the **first question**, master/slave modes operate as follows in your export.json:

- By default, all sObjects are in **master** mode, executing the query as defined, including any WHERE or LIMIT clauses.
- For a **slave** object, the plugin removes the LIMIT but keeps the WHERE clause, adding conditions to select only related child records.

Consider this configuration example:

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

What encoding is supported for CSV files by the plugin?

Answer.

The **required** encoding of the source CSV files is **UTF-8**

The same is regarding all CSV files generated by the Plugin itself, they are generated using UTF-8.

Last updated on 18th Apr 2024

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

What encoding type should be used for the export.json file?

Answer.

Your **export.json** file should be **ONLY** in UTF-8 format. Other formats **are not supported**.

Last updated on 20th Apr 2024

SFDMU OVERVIEW

[Key Features of the SFDMU](#)

[Get Started](#)

[Installation](#)

[Configuration](#)

[Running](#)

[Debugging](#)

[SFDMU GUI Application](#)

FULL DOCUMENTATION

 [Export.json File Specific](#)

 [Advanced features](#)

 [Add-On API](#)

 [Reports](#)

EXAMPLES

Can CSV files from the standard Salesforce Data Loader be used with this plugin?

Answer.

Yes, you can. They are mostly compatible, except for the differences listed below.

Using SFDMU to import data exported using Data Loader:

- The Data Loader generates CSV files with column names spelled in upper case. You must rename the column names to match exactly the field API names, as they are case-sensitive.
- The header row cannot contain double quotes (""); ensure to remove them.
- Consider setting the UTF-8 format in the Salesforce Data Loader settings when exporting to a CSV file.

Using Data Loader to import data exported using SFDMU:

- The CSV file generated by SFDMU is fully compatible with the Salesforce Data

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

Can I override the default query for the User or Group objects?

Answer:

When you include any lookup field to the **User** or **Group** objects (for example, `OwnerId`), SFDMU will automatically add both **User** and **Group** objects to the job with a **ReadOnly** operation. SFDMU supports only reading records from these objects but does not allow updating or inserting them.

The default query for the **User** object included in the job is:

```
SELECT Id, Name FROM User
```

The default query for the **Group** object is:

```
SELECT Id, Name FROM Group WHERE Type = 'Queue'
```

You can override the default query for the **User** object by adding a dedicated section for

SFDMU OVERVIEW

[Key Features of the SFDMU](#)

[Get Started](#)

[Installation](#)

[Configuration](#)

[Running](#)

[Debugging](#)

[SFDMU GUI Application](#)

FULL DOCUMENTATION

 [Export.json File Specific](#)

 [Advanced features](#)

 [Add-On API](#)

 [Reports](#)

EXAMPLES

Can the plugin migrate Users or Groups?

Answer.

No, it's not possible.

Migration of **Users/Groups** is **NOT** supported.

The following objects are not supported for **Insert/Update/Upsert** operations:

- Profile
- User
- Group
- DandBCompany

However, these objects can still be used in **Readonly** operations.

Last updated on 18th Apr 2024

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

Does the plugin support exporting object metadata?

Answer.

No, the SFDMU can't create/update metadata information.

Export and import of metadata are not supported by SFDMU, as this tool is intended solely for data migration and not for metadata modification.

You must have identical object definitions in both the source and target orgs unless you map your sObject/field to a different one using the [Fields Mapping feature](#).

Last updated on 18th Apr 2024

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

Using the `all` multiselect keyword with the Product2 object.

Question:

I am trying to migrate **Product2** records from one sandbox to another using the `all` multiselect keyword:

```
SELECT all FROM Product2
```

However, I am encountering the following error. The field permissions (FLS) appear to be correctly configured:

```
[12:13:36.877] [ERROR] Error during execution of the command:  
Recommended_Retail_Price__c, ExternalDataSource.DeveloperName ^ ERROR at  
Row:1:Column:3200 Didn't understand relationship 'ExternalDataSource' in field  
path. If you are attempting to use a custom relationship, be sure to append '__r'  
to the custom relationship name. Please reference your WSDL or the describe call  
for the appropriate names.. [12:13:36.879] [ERROR] Execution of the command  
sfmu:run has been completed. Exit code 4 (COMMAND_EXECUTION_ERROR).
```

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

-  [Export.json File Specific](#)
-  [Advanced features](#)
-  [Add-On API](#)
-  [Reports](#)

EXAMPLES

Why are there differences in failures between simulation and non-simulation modes?

Question:

I assumed that CSV files generated in simulation mode would be the same as those in non-simulation mode during org-to-org migration. In simulation mode, I'm not observing any failures, but when using the same export configuration in non-simulation mode, I see many failures in the target CSV files.

Why does simulation mode not generate the same set of failures that occur in non-simulation mode?

Answer:

Simulation mode in org-to-org migration is designed solely to check the metadata and query the source organization without updating the target organization. Its primary function is to verify data integrity and identify potential configuration issues related to objects, fields, and relationships, and to assess the volume of data that will be updated in the target. The

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

Can the Document object be transferred with binary data?

Answer.

Document object is **NOT** currently supported by the [core:ExportFiles Add-On](#).

So this subject type can be transferred **without** binary data.

Last updated on 18th Apr 2024

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

Is it possible to use multiple CSV source files for the same object?

Question:

I'm using the Multiple Object Sets feature and would like to use a different Account.csv file as the source for each ObjectSet in my configuration.

I want to import records sequentially from Account1.csv, then Account2.csv, and so on. Is this possible?

Answer:

Yes, using multiple CSV sources is supported.

You can configure each Object Set to use either the same or different CSV files by setting the **useSeparatedCSVFiles** property.

For more detailed information, refer to the [Multiple Object Sets](#) documentation.

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

What types of fields can be excluded using the "excludedFields" property?

Answer:

Use **excludedFields** to omit specific fields from your SOQL query results.

- Excluding a field from a regular query:

```
{  
    "query": "SELECT Id, RecordTypeId, Name, IsPersonAccount FROM Account",  
    "excludedFields": ["IsPersonAccount"]  
}
```

This configuration excludes **IsPersonAccount** from processing.

- Excluding a field from a multiselect query:

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

Are field names and object names in a SOQL query case sensitive?

Answer.

- The **field names** used in an object's SOQL select statements are **case-insensitive**. Therefore, the query: "**SELECT id, NAme FROM Account**" should be correct.
- The **object names** used in SOQL are **case-sensitive**. Therefore, the query: "**SELECT id, Name FROM ACCOUNT**" may **not** be handled correctly.

Despite this, there are several exceptions where the case of SOQL fields is important:

- For [Fields Mapping](#) and [Data Anonymization](#), you need to specify fields and object names **exactly as they are case-sensitive**.
- When listing sObject field(s) in an **externalId** setting, it is crucial to ensure that the fields are spelled correctly, as this setting is case-sensitive.

Last updated on 18th Apr 2024

SFDMU OVERVIEW

[Key Features of the SFDMU](#)

[Get Started](#)

[Installation](#)

[Configuration](#)

[Running](#)

[Debugging](#)

[SFDMU GUI Application](#)

FULL DOCUMENTATION

 [Export.json File Specific](#)

 [Advanced features](#)

 [Add-On API](#)

 [Reports](#)

EXAMPLES

Can autonumber fields be used as External IDs?

Answer.

Yes, you can use autonumber fields as an External Id, but only for the **Insert** operation.

- For operations that might include record updates, autonumber fields **CANNOT** be used as External Ids. This is because autonumber values are generated independently in each org, meaning they will never overlap between the Source and the Target orgs.
- If you use autonumber fields for insertions, it essentially does not matter. The plugin [Can I omit the definition of External ID for an object?](#).

Last updated on 18th Apr 2024

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

Can I export fields from parent sObjects using a child object query?

Question:

I have a parent object `ParentObject__c` and want to retrieve all names of parent objects using a child query on `ChildObject__c`.

In my configuration:

- `ChildObject__c` is the child object.
- `ParentObject__c` is the parent.

I am trying to use this query, but it's not working:

```
SELECT Name, ParentObject__r.Name FROM ChildObject__c
```

What is wrong?

Answer:

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

-  [Export.json File Specific](#)
-  [Advanced features](#)
-  [Add-On API](#)
-  [Reports](#)

EXAMPLES

Is it necessary to always specify the parent object for a polymorphic field?

Answer.

Yes, you should always specify the parent object for polymorphic fields.

See the examples:

```
"query": "SELECT Id, ParentId$Case FROM FeedItem WHERE Parent.Type = 'Case'"  
"query": "SELECT Id, ParentId$Case, Name, Body FROM Attachment WHERE Parent.Type = 'Case'"
```

Typically, the plugin detects the referenced object type from the field metadata.

However, for polymorphic fields, the parent object type is not defined in the field metadata. Therefore, the parent object type cannot be detected automatically and must be explicitly specified in the query string.

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

Should the Id field always be included in my query?

Answer.

No, you don't need to explicitly include the Id field in your configuration; the plugin includes it automatically.

For example, the query **SELECT Name FROM Account** (which does not explicitly include the Id field) is still valid.

Last updated on 18th Apr 2024

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

Can I modify and use the plugin code for my own projects?

Answer.

Yes, you can.

The Plugin is distributed as open-source project.

You can use the source code of the Plugin in your projects in terms of the [License information](#)

Last updated on 18th Apr 2024

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

Can the plugin be used in production environments?

Answer.

Yes, you can use it for every type of environments.

For the details see the [License Agreement](#)

Last updated on 18th Apr 2024

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

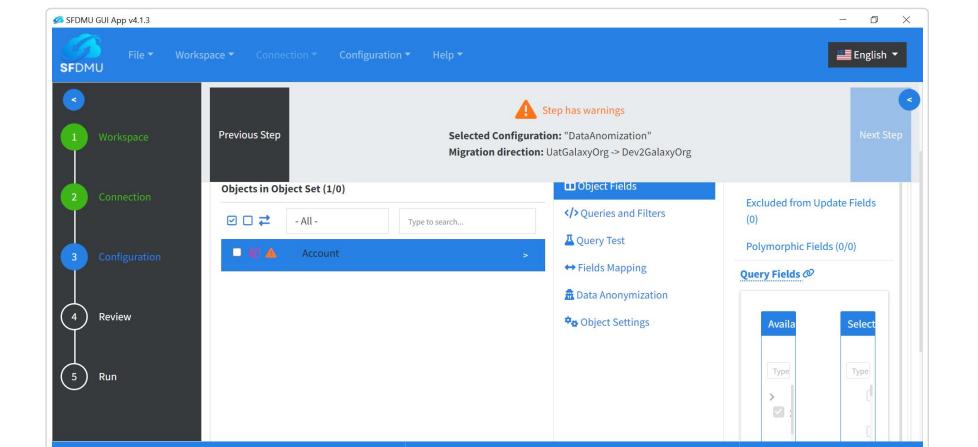
- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

How to address UI issues and field selection in the new SFDMU GUI?

Question:

I have switched to the new version of the SFDMU GUI, but I am unable to select object fields, the field selector is too narrow.



SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

Is SFDMU compatible with the latest SF CLI?

Question:

Our company uses SF as the standard CLI and we are not allowed to use SFDX as it is no longer supported. Do you have plans to move to the up-to-date SF tool? Do current users have both SF and SFDX installed? I encountered issues when both were installed on my personal machine.

Answer:

SFDMU is fully compatible with the latest SF CLI version. To install SFDMU, you can use the command: `sf plugins install sfmu`. To run SFDMU, use the following command format: `sf sfmu run --sourceusername source@org.org --targetusername target@org.org`.

This means you simply need to replace the `sfdx` command structure with `sf` followed by the relevant subcommands, allowing seamless integration with your current setup.

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

Why does SFDMU command fail in the force-app directory with SF CLI?

Question:

The source code of the Plugin has been linked in some directory, for example, SFDX-Data-Move-Utility-master .

I want to execute the `sf sfmu run` command inside the root folder of my `force-app` project or one of its subfolders like `force-app` , `force-app/main` , etc.

However, I encounter an error displayed as a `SyntaxError` related to import statements outside a module. The error message is as follows:

```
(node:22516) SyntaxError: Plugin: sfmu: Cannot use import statement outside a
module
module: @oclif/core@3.12.0
task: cacheCommand
plugin: sfmu
root: C:\Repository\SFDX-Data-Move-Utility-master
See more details with DEBUG=*
(Use `node --trace-warnings ...` to show where the warning was created)
```

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

Why doesn't FieldMapping rename CSV column headers in my configuration?

Question:

I am using FieldMapping to generate a CSV file with specific column headers for an external target system. My configuration is as follows:

```
{  
    "objects": [  
        {  
            "query": "SELECT Name from Account",  
            "externalId": "Name",  
            "operation": "Insert",  
            "useFieldMapping": true  
        }  
    ]  
}
```

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

Can all objects be defined as "slave" objects in the configuration?

Answer:

Yes, you can technically configure all objects in your system as "slaves". This setup necessitates an implicit external master or an external process that supplies the necessary IDs for filtering.

When all objects are set as "slaves" without an explicit "master," there is no inherent mechanism to specify which records to fetch. Slave objects rely on external IDs (such as customer or order IDs) provided by another process or query. These IDs are essential for filtering and determining which records the slave objects retrieve, typically based on the outputs from a master object's query.

Example `export.json` for an all-slave configuration:

```
{  
    "objects": [  
        {  
            "query": "SELECT Id FROM Account WHERE Id IN :providedAccountIds",  
            "externalId": "Name".
```

SFDMU OVERVIEW

[Key Features of the SFDMU](#)

[Get Started](#)

[Installation](#)

[Configuration](#)

[Running](#)

[Debugging](#)

[SFDMU GUI Application](#)

FULL DOCUMENTATION

 [Export.json File Specific](#)

 [Advanced features](#)

 [Add-On API](#)

 [Reports](#)

EXAMPLES

Can CSV files be used for CSV-to-org migration in simulation mode?

Answer.

No. CSV files cannot be used for CSV-to-org migration in simulation mode.

The simulation mode is intended only to verify metadata and query the source organization without updating the target organization. Its primary purpose is to ensure data integrity and identify potential configuration issues related to objects, fields, and relationships. It also helps assess the volume of data that will be updated in the target organization.

Since the source in this case is a CSV file, simulation mode is not applicable for CSV-to-org migrations.

Last updated on 18th Apr 2024

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

Can I omit the definition of External ID for an object?

Answer.

Yes. You can omit it, but only for the **Insert** operation.

The reason is that for insertion, the plugin always uses the `Id` as the External Id field, regardless of what you explicitly define in your `export.json`.

Last updated on 18th Apr 2024

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [!\[\]\(62bfc7c6857b3c70e5a98ef6c1ec8435_img.jpg\) Export.json File Specific](#)
- [!\[\]\(642894274b274b68e387930c13530312_img.jpg\) Advanced features](#)
- [!\[\]\(1911ace2d61940880c1b51dbab0a7bc7_img.jpg\) Add-On API](#)
- [!\[\]\(2ad57714d4bfb1bb732ec99566b3dab7_img.jpg\) Reports](#)

EXAMPLES

How to modify the include regex option in SFDMU configuration?

Question:

Based on the provided configuration snippet for SFDMU, how can we remove the include regex option for anonymizing data?

```
{
  "objects": [
    {
      "operation": "Upsert",
      "externalId": "FirstName;LastName;Email",
      "mockFields": [
        {
          "name": "Phone",
          "pattern": "phone",
          "excludedRegex": "",
          "includedRegex": ""
        },
        {
          "name": "Fax",
          "pattern": "phone"
        }
      ]
    }
  ]
}
```

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

How to prevent resetting checkboxes using SFDMU with multiselect keyword?

Question:

In SFDMU, I am using the multiselect keyword in my export.json for updating accounts, but the checkbox field (TEST_CHK_c) is being reset because it is not included in the source Account.csv file. How can I use the same CSV file with the multiselect keyword and still prevent the checkbox from resetting?

```
{  
    "objects": [  
        {  
            "query": "SELECT all FROM Account",  
            "operation": "Update",  
            "externalId": "Name"  
        }  
    ]  
}
```

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

What pre-validation options does SFDMU offer before inserting data?

Question:

When using SFDMU, it operates in a partial success mode, where valid records are inserted even if others fail. This can leave an Org in an unfinished state, with some records inserted and others not.

For such scenarios, a data validation or rollback feature is often requested. SFDMU's **simulationMode** only detects missing parent records (both lookup and master-detail), creating a 'MissingParentRecordsReport.csv' file, but it does not validate against validation rules or field uniqueness.

What pre-validation options does SFDMU offer?

Answer:

Since SFDMU does not fully support record rollback, there are two main pre-validation options:

1. **Simulation Mode:** By setting `sfmu.simulationMode = true`, you can preview which records

SFDMU OVERVIEW

[Key Features of the SFDMU](#)

[Get Started](#)

[Installation](#)

[Configuration](#)

[Running](#)

[Debugging](#)

[SFDMU GUI Application](#)

FULL DOCUMENTATION

 [Export.json File Specific](#)

 [Advanced features](#)

 [Add-On API](#)

 [Reports](#)

EXAMPLES

Can child records with a self-lookup field be inserted in a single job?

Question:

Can child records with a populated self-lookup field (i.e., a ParentId field pointing to the same sObject type for the Account object) be inserted in a single job, especially when the after insert trigger logic is dependent on whether the self-lookup field is populated?

Answer:

It is **impossible** to insert child records with self-referential parent lookup fields pre-populated since the target parent IDs may not be known in advance.

The plugin consistently performs two actions for self-referenced objects:

1. Inserts all records with the self-reference field set to null.
2. Updates the self-reference fields with the already existing parent values.

Last updated on 18th Apr 2024



SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
 - [Get Started](#)
 - [Installation](#)
 - [Configuration](#)
 - [Running](#)
 - [Debugging](#)
 - [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
 - [Advanced features](#)
 - [Add-On API](#)
 - [Reports](#)

EXAMPLES

How to combine multiple account extractions into one CSV with SFDMU?

Question:

I am trying to combine accounts from several extractions into one Account.csv file and subsequently fetch related contacts using another job in SFDMU.

I encounter issues with the accounts CSV being overwritten after each extraction.

Here is what my `export.json` file looks like:

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

How to handle checkbox fields during CSV imports in SFDMU?

Question:

When upserting Accounts from a CSV file, the process fails if the target organization has additional fields such as a Checkbox field (e.g., IsTest__c) that is not present in the CSV. The error "IsTest: value not of required type:" occurs because the checkbox values in the generated target/Account_insert_target.csv are empty (not boolean).

How can this issue be resolved?

Answer:

To resolve the issue with checkbox fields during imports using SFDMU, it is possible to insert an empty value instead of "false" for checkboxes, which effectively resets the checkbox in the target organization.

Here's a sample configuration for your export.json that specifically handles checkbox fields:

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

How to handle Master-Detail relationships with SFDMU?

Question:

I'm facing challenges with inserting Authorization Form & Authorization Form Text using SFDMU, likely due to their Master-Detail relationship.

```
{  
    "objects": [  
        {  
            "query": "SELECT Id, Name, RevisionNumber FROM AuthorizationForm",  
            "externalId": "Name",  
            "operation": "Upsert"  
        },  
        {  
            "query": "SELECT Name, AuthorizationFormId FROM  
AuthorizationFormText",  
            "externalId": "Id",  
            "operation": "Upsert"  
        }  
    ]  
}
```

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

How to resolve "Object Location is missing in the Target" error in SFDMU?

Question:

I am able to generate the configuration file using the SFDMU GUI app without any errors. However, when I attempt to migrate data from the object `Location`, I encounter the following error:

[ERROR] Mismatch org metadata error: Object Location is missing *in* the Target.

I have confirmed that the user has access to the `Location` object via a Permission Set or Profile and have completed the Field Service setup. What should I check next?

Answer:

The error you're encountering, where the "Location" object is reported as missing in the

SFDMU OVERVIEW

[Key Features of the SFDMU](#)

[Get Started](#)

[Installation](#)

[Configuration](#)

[Running](#)

[Debugging](#)

[SFDMU GUI Application](#)

FULL DOCUMENTATION

 [Export.json File Specific](#)

 [Advanced features](#)

 [Add-On API](#)

 [Reports](#)

EXAMPLES

What is the character limit for the WHERE clause in the export.json file?

Question:

I have an organization with thousands of records. Can I specify which ones to select by including all their IDs in the WHERE clause of my export.json? What are the limitations on the length of a SOQL statement in export.json?

Answer:

Yes, you can include a list of IDs in the WHERE clause using the IN operator.

- Remember to set **master = true** for your sObject or omit this parameter.
- The length of the WHERE clause is limited to 4000 characters, as per [Salesforce documentation](#).

Last updated on 18th Apr 2024

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

Why does adding person account records not respect specified record types?

Question:

I'm facing an issue while trying to insert person account records into my org with the Person Accounts feature enabled. I'm using the Record Type feature in SFDMU and specifying the RecordTypeid in the export.json and source CSV file.

However, when I run the SFDMU command, I receive a warning that says "[WARN] {Account} 500 missing parent lookup records were found. See MissingParentRecordsReport.csv for details." Consequently, accounts are created with the default record type, not as person accounts.

What could be causing this discrepancy?

Answer:

To load account records with their specified record types, it's essential to provide accurate

SFDMU OVERVIEW

[Key Features of the SFDMU](#)

[Get Started](#)

[Installation](#)

[Configuration](#)

[Running](#)

[Debugging](#)

[SFDMU GUI Application](#)

FULL DOCUMENTATION

[Export.json File Specific](#)

[Advanced features](#)

[Add-On API](#)

[Reports](#)

EXAMPLES

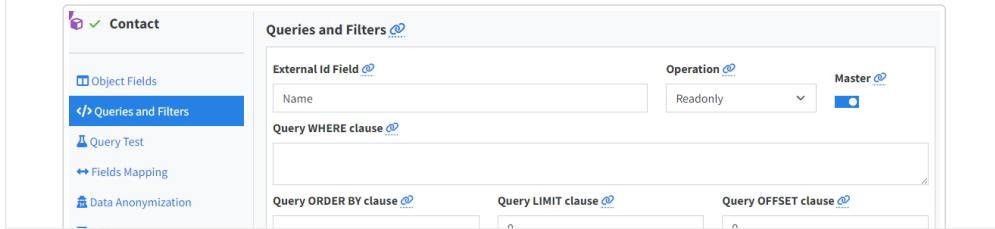
SFDMU GUI - I can't find where can I set the Operation for the object.

Question:

I'm struggling to find where I can switch the operation for the selected SObject. For example, how do I set it to Upsert?

Answer:

By default, every object is set to **Readonly** mode. You can update the **Operation**, **WHERE Clause**, and other main object parameters under the **Queries and Filters** tab as shown below:



SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [!\[\]\(99129bd9ab998abccdc52d2c8c2d0de5_img.jpg\) Export.json File Specific](#)
- [!\[\]\(82a5708b338d99f0a9d21d4022c6185e_img.jpg\) Advanced features](#)
- [!\[\]\(2b7a4d7406d610a7efb3ceb7a3818de5_img.jpg\) Add-On API](#)
- [!\[\]\(729618ca393055bd4c0fc4cf43450b47_img.jpg\) Reports](#)

EXAMPLES

SFDMU GUI - SF CLI is no longer detected.

Question:

I'm using macOS.

When I reach the configuration step and select **Detect Available SFDX Connections**, I get the error: "**Failed to execute the command... Check whether the CLI is installed.**"

Answer:

On macOS, you can't run the SFDMU GUI app directly from the **Applications** folder. Instead, you need to launch it from the Terminal, following the exact steps described here:

[Running the Installed Application](#)

Last updated on 9th Mar 2025

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

Notes and attachments migration issues.

Question:

I attempted to import attachments from the Account object using this query:

```
SELECT Id, ParentId, Title, IsPrivate, Body FROM Note WHERE Parent.Type = 'Account'
```

However, the Plugin failed to insert records and displayed this error:

```
[ERROR] Error during execution of the command:  
Name, ParentId, CreatedBy.Name, Parent.Name FROM Attachment  
^  
ERROR at Row:1:Column:136  
No such column 'Name' on entity 'Name'. If you are attempting to use a custom  
field, be sure to append the '__c' after the custom field name. Please reference  
your WSDL or the describe call for the appropriate names.  
Execution of the command sfmu:run has been finished. Exit code 4  
/COMMAND_EXECUTION_ERROR/
```

SFDMU OVERVIEW

[Key Features of the SFDMU](#)

[Get Started](#)

[Installation](#)

[Configuration](#)

[Running](#)

[Debugging](#)

[SFDMU GUI Application](#)

FULL DOCUMENTATION

[Export.json File Specific](#)

[Advanced features](#)

[Add-On API](#)

[Reports](#)

EXAMPLES

Why no records inserted?

Question

I am attempting to insert new records into a system. There are 46 records available in the source, but the process inserts 0 records, and I cannot determine the reason. Here is my configuration:

```
{  
    "operation": "Insert",  
    "externalId": "Name",  
    "query": "SELECT Id, Name FROM Commitment__c WHERE Deal__r.Name LIKE  
    'MigVax%'  
}
```

What could be the issue?

Answer

The issue likely stems from the use of the **Name** field as an external ID in your `Commitment__c` object, which might be an **auto-number** field. This configuration means

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

Problem of values mapping with regex.

Question:

I am attempting to modify the Contact Email before inserting or updating it in the target organization by replacing ".", "-", and "@" with "_". However, my regex doesn't seem to find any records to update. I even tried using JavaScript evaluation, but had no luck. My regex looks correct; what could be the issue?

Here is what my **ValueMapping.csv** looks like:

```
ObjectName,FieldName,RawValue,Value
Contact,Identite_digital_c,2,1
Contact,Email,/(.[-]+)/,_
```

Answer:

The issue stems from your regex expression not being enclosed in double quotes, which is necessary for it to be recognized as a single string value. The presence of a comma within

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

Warning about missing parent lookup records.

Question:

Why does the Plugin generate a warning about missing parent lookup records for {OBJECT_NAME} stating "{OBJECT_NAME} N missing parent lookup records were found. See MissingParentRecordsReport.csv file for the details," even when all lookup fields are populated for this object in the Target?

Answer:

This warning may occur if the Plugin attempts to insert a child record before the corresponding parent record is present in the Target, leading to a precautionary warning despite successful population of the lookup fields later.

If you confirm that all lookup fields are correctly populated, this warning can be safely ignored.

To avoid interruption during operation, set the `promptOnMissingParentObjects` configuration option to `false`. To suppress all such warnings, you can add the `--`

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [!\[\]\(336b9645836fc42ff00f28532341057f_img.jpg\) Export.json File Specific](#)
- [!\[\]\(58bc2079078eebb2856ad932af40b9a4_img.jpg\) Advanced features](#)
- [!\[\]\(5450cff214b609a4e91e5b7e6d7ae6c0_img.jpg\) Add-On API](#)
- [!\[\]\(48ab4ea60a88d426876cccaf59c40396_img.jpg\) Reports](#)

EXAMPLES

Why data anonymization is not working?

Question:

I'm having trouble with this configuration because the phone field value remains the same as the source, even though I've tried to anonymize it:

```
{  
    "objects": [  
        {  
            "operation": "Upsert",  
            "externalId": "FirstName;LastName;Email",  
            "mockFields": [  
                {  
                    "name": "Phone",  
                    "pattern": "phone",  
                    "excludedRegex": "",  
                    "includedRegex": ""  
                },  
                {  
                    "name": "Fax",  
                    "pattern": "phone"  
                }  
            ]  
        }  
    ]  
}
```

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

Why does the values mapping not work for me?

Question:

I'm attempting to import a CSV file using the following configuration:

```
{  
    "objects": [  
        {  
            "query": "SELECT Id, Name FROM Contact",  
            "operation": "Insert",  
            "externalId": "Name",  
            "useValuesMapping" : true  
        }  
    ]  
}
```

I have created a **ValueMapping.csv** file and enabled the feature with the **useValuesMapping=true** parameter, but it doesn't seem to work. What could be the issue?

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

Why user and record type lookups are not filled on insert?

Question:

I'm importing data from a CSV source into a custom object with a custom Lookup field to the User object, but the field isn't being filled when inserting new records. Why is this happening?

Answer:

The issue might be due to missing **RecordType.csv** and/or **UserAndGroup.csv** source files.

For successful import of lookup values, it's essential to provide **ALL** CSV source files, as the Plugin treats these files like a "virtual" Salesforce org containing all necessary records.

Last updated on 20th Apr 2024

SFDMU OVERVIEW

[Key Features of the SFDMU](#)

[Get Started](#)

[Installation](#)

[Configuration](#)

[Running](#)

[Debugging](#)

[SFDMU GUI Application](#)

FULL DOCUMENTATION

 [Export.json File Specific](#)

 [Advanced features](#)

 [Add-On API](#)

 [Reports](#)

EXAMPLES

GUI app installation failed or showing blank screen.

Symptom:

Users may encounter issues when installing and launching the SFDMU GUI App on the Electron platform. Common problems include error messages during installation or a blank window upon attempting to run the app.

Solution:

For Windows OS:

- Delete the SFDMU GUI app folder.
- Uninstall Electron.
- Install Windows Build Tools using `npm install --global windows-build-tools`.
- Reinstall Electron globally with `yarn global add electron`.
- Re-clone the SFDMU GUI App repository.
- Run `npm install` followed by `npm start`.

For MacOS, there is currently no working solution available for this issue.

SFDMU OVERVIEW

[Key Features of the SFDMU](#)

[Get Started](#)

[Installation](#)

[Configuration](#)

[Running](#)

[Debugging](#)

[SFDMU GUI Application](#)

FULL DOCUMENTATION

 [Export.json File Specific](#)

 [Advanced features](#)

 [Add-On API](#)

 [Reports](#)

EXAMPLES

Plugin cannot connect to the org.

Question

I can connect to other sandboxes without issues, but when trying to connect to Production, I receive the following error:

```
[12:15:22.262] Connecting to staging using SFDX force:org:display ...
[12:15:30.534] Successfully connected to staging.
[12:15:31.290] Connecting to prod using SFDX force:org:display ...
[12:15:33.342][ERROR] Command initialization error: Attempt to connect to prod failed. Please, try to refresh your local SFDX CLI connection.
```

The result of **sf force org display** shows the org is connected:

```
{
  "status": 0,
  "result": {
    "id": "00D1U000000xKhkUAF",
    "accessToken": "accessToken",
    "instanceUrl": "https://obscured-company-name.my.salesforce.com",
    "username": "obscured_username@company.com"
```

SFDMU OVERVIEW

[Key Features of the SFDMU](#)

[Get Started](#)

[Installation](#)

[Configuration](#)

[Running](#)

[Debugging](#)

[SFDMU GUI Application](#)

FULL DOCUMENTATION

[Export.json File Specific](#)

[Advanced features](#)

[Add-On API](#)

[Reports](#)

EXAMPLES

Plugin cannot locate export.json in working directory.

Question

I followed the setup instructions and created the following `export.json`:

```
{  
    "objects": [  
        {  
            "query": "SELECT Id, Name FROM Account",  
            "operation": "Upsert",  
            "externalId": "Name"  
        }  
    ]  
}
```

I saved it in `./scripts`, installed SFDMU successfully, and then ran:

```
$ sfdx sfdmu:run -sourceusername source -targetusername target -p ./scripts
```

SFDMU OVERVIEW

[Key Features of the SFDMU](#)

[Get Started](#)

[Installation](#)

[Configuration](#)

[Running](#)

[Debugging](#)

[SFDMU GUI Application](#)

FULL DOCUMENTATION

[Export.json File Specific](#)

[Advanced features](#)

[Add-On API](#)

[Reports](#)

EXAMPLES

Plugin error.

- **Question:**

I created a simple configuration using the "all" macro to include all fields from the Account object:

```
{  
    "objects": [  
        {  
            "query": "SELECT all FROM Account",  
            "operation": "Insert"  
        }  
    ]  
}
```

However, I received an error:

The SOQL query *for* sObject Account does not include any field that can be updated or inserted.

SFDMU OVERVIEW

[Key Features of the SFDMU](#)

[Get Started](#)

[Installation](#)

[Configuration](#)

[Running](#)

[Debugging](#)

[SFDMU GUI Application](#)

FULL DOCUMENTATION

 [Export.json File Specific](#)

 [Advanced features](#)

 [Add-On API](#)

 [Reports](#)

EXAMPLES

Unable to run plugin due to Node.js or SFDX-CLI issue.

Question

I am experiencing issues running the plugin, with an error pointing to a problem with Node.js or SFDX-CLI. How can I resolve this?

Answer

To troubleshoot issues with Node.js or SFDX-CLI, follow these steps:

For Node.js:

1. Open the command prompt (Windows) or Terminal (macOS).
2. Type `node -v` and press Enter to check the Node.js version. This confirms if Node.js is installed and operational.
3. If Node.js is not installed, download it from the [official Node.js website](#).
4. For further help, refer to the [Node.js documentation](#).

For SFDX-CLI:

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

Allow disabling SSL and hash checks for plugin linking.

Question

I encounter an error when trying to link a plugin to the CLI in a corporate network environment:

```
PS C:\Sandbox\SFDX-Data-Move-Utility> sfdx plugins:Link  
warning package-lock.json found. Your project contains lock files generated by  
tools other than Yarn. It is advised not to mix package managers in order to avoid  
resolution inconsistencies caused by unsynchronized lock files. To clear this  
warning, remove package-lock.json.
```

```
(node:5912) Warning: Setting the NODE_TLS_REJECT_UNAUTHORIZED environment variable  
to '0' makes TLS connections and HTTPS requests insecure by disabling certificate  
verification.
```

```
error An unexpected error occurred: "SSL Error:  
UNABLE_TO_GET_ISSUER_CERT_LOCALLY".
```

```
sfdx-cli: linking plugin sfmu... !
```

SFDMU OVERVIEW

[Key Features of the SFDMU](#)

[Get Started](#)

[Installation](#)

[Configuration](#)

[Running](#)

[Debugging](#)

[SFDMU GUI Application](#)

FULL DOCUMENTATION

 [Export.json File Specific](#)

 [Advanced features](#)

 [Add-On API](#)

 [Reports](#)

EXAMPLES

Authentication issues with SFDX orgs.

Question

When attempting to execute a migration job and connect to my SFDX orgs, I encounter the following error:

Command initialization error: Access token to the CAREFUL_PRODUCTION_CAREFUL has expired or the user has no access.

How can I resolve this?

Answer

This error can occur for several reasons:

- If you're using the `orgs` section in the `export.json`, your access token may have expired. Refer to the [Export.json File Specification](#) for details.

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

JavaScript heap out of memory error.

Question

I'm encountering a "JavaScript heap out of memory" error. How can I resolve this?

Answer

This error often occurs when processing large datasets, such as a CSV file larger than 100MB. To resolve this, you can increase the available memory for Node.js by adjusting the heap size. Here's how to set the maximum allowed memory size to 8GB:

For Windows OS:

```
set NODE_OPTIONS=--max-old-space-size=8192
```

For Mac/Linux OS:

SFDMU OVERVIEW

[Key Features of the SFDMU](#)

[Get Started](#)

[Installation](#)

[Configuration](#)

[Running](#)

[Debugging](#)

[SFDMU GUI Application](#)

FULL DOCUMENTATION

 [Export.json File Specific](#)

 [Advanced features](#)

 [Add-On API](#)

 [Reports](#)

EXAMPLES

Read ECONNRESET error during source retrieval.

Question

I'm attempting to fetch around 800k records for an sObject, but I encounter an error, and the command remains in progress without completing any actions. What could be the issue?

```
[Application__c] Retrieving the source data (ALL RECORDS) in progress...
{ Error: read ECONNRESET
  at TLSWrap.onStreamRead (internal/stream_base_commons.js:111:27) errno: 'ECONNRESET', code: 'ECONNRESET', syscall: 'read' }
Command in progress..... -
```

Answer

The `ECONNRESET` error typically indicates one of three issues:

1. Your internet connection was suddenly interrupted.
2. The remote server has stopped responding, possibly due to maintenance in the Salesforce environment.
3. The remote call timed out due to the large data size or a slow internet connection.
The request timeout for queries is set to 4 minutes, which can lead to various timeout

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

Unexpected behavior with default SFDX target org configuration.

Question

When I omit the `-u` CLI flag and rely on the currently set default SFDX target org, I expect the tool to handle it. However, the tool returns a message saying this has not been supplied. Does SFDMU support using the default SFDX org?

Answer

No, SFDMU does not support using the default SFDX org set via the `sfdx force:config:set defaultusername=<Alias_name>` command. When you omit the `-u` or `--targetusername`, the `--targetusername` does not inherit from the default org specified.

To ensure SFDMU connects to your desired target org, you must explicitly provide `-u` with the `sfdx sfmu:run` command. If you omit the `-u` flag, SFDMU will not return a runtime error, but will default to using **csvfile** as the target.

Last updated on 20th Apr 2024

SFDMU OVERVIEW

[Key Features of the SFDMU](#)

[Get Started](#)

[Installation](#)

[Configuration](#)

[Running](#)

[Debugging](#)

[SFDMU GUI Application](#)

FULL DOCUMENTATION

 [Export.json File Specific](#)

 [Advanced features](#)

 [Add-On API](#)

 [Reports](#)

EXAMPLES

Why do I get the error "Invalid record hashcode from bulk job"?

Question

When attempting to copy all Contact records, errors occur stating "Invalid record hashcode. Unable to find matching record from the response returned by the bulk job."

The `export.json` configuration is:

```
{  
    "operation": "Upsert",  
    "externalId": "Email",  
    "query": "SELECT Id, Name, Email, Platform_Role__c, LastName, FirstName,  
    Salutation, Secondary_Email__c, MailingCity FROM Contact"  
}
```

Console output indicates:

```
[WARN] [Batch#7502g000000rYc5AAE:Insert] {Contact} Incompleted.
```

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [!\[\]\(c226daf732c9ba9ed5234aad8ccfce22_img.jpg\) Export.json File Specific](#)
- [!\[\]\(0dd311ad4667333b761fafacdedcbc79_img.jpg\) Advanced features](#)
- [!\[\]\(f5ec85e103a7c043704f845af2296684_img.jpg\) Add-On API](#)
- [!\[\]\(acb2edb5444c3bb777eae09ee0759fd0_img.jpg\) Reports](#)

EXAMPLES

Why the deleting from the target always runs for me?

Question:

Why does the deletion process always run when I use the export file below, even though I've set `deleteOldData` to false?

```
{  
    "query": "Select Name from TestObject1__c",  
    "operation": "Insert",  
    "deleteOldData": "false",  
    "externalId": "Name"  
}
```

Answer:

The issue arises because the `deleteOldData` property is incorrectly set as a string rather than a boolean. Remove the quotation marks to correctly set it as a boolean value:

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [!\[\]\(e26b0f868e6bc4b00f6e748373cd71a5_img.jpg\) Export.json File Speci](#)
 - [Export.json File Overview](#)
 - [Script Object](#)
 - [ScriptObjectSet Object](#)
 - [ScriptObject Obiect](#)

Export.json File Overview.

Table of Contents:

- [Configuring SFDMU with the export.json File](#)
 - [Understanding the export.json File Structure](#)
 - [Notes on Configuration](#)
- [Export.json Object Model](#)
 - [Script Object](#)
 - [ScriptObjectSet Object](#)
 - [ScriptOrg Object](#)
 - [ScriptObject Object](#)
 - [MappingItem Object](#)
 - [MockField Object](#)
 - [AddonManifestDefinition Object](#)
- [Additional Notes](#)

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Spec](#)
 - [Export.json File Overview](#)
 - [Script Object](#)
 - [ScriptObjectSet Object](#)
 - [ScriptObject Object](#)

Script Object.

[**← Go Back to Export.json File Overview**](#)

Table of Contents:

- [Script Object Overview](#)
- [Script Object Structure](#)
- [Script Object Properties](#)
 - [afterAddons \(Array of AddonManifestDefinition\)](#)
 - [allOrNone \(Boolean\)](#)
 - [allowFieldTruncation \(Boolean\)](#)
 - [alwaysUseRestApiToUpdateRecords \(Boolean\)](#)
 - [apiVersion \(String in float format\)](#)
 - [beforeAddons \(Array of AddonManifestDefinition\)](#)
 - [binaryDataCache \(String\)](#)
 - [bulkApiV1BatchSize \(Integer\)](#)
 - [bulkApiVersion \(String\)](#)
 - [bulkThreshold \(Integer\)](#)
 - [queryBulkApiThreshold \(Integer\)](#)
 - [concurrencyMode \(String\)](#)
 - [createTargetCSVFiles \(Boolean\)](#)
 - [csvReadFileDelimiter \(String\)](#)

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Speci](#)
- [Export.json File Overview](#)
- [Script Object](#)
- [ScriptObjectSet Object](#)
- [ScriptObject Obiect](#)

ScriptObjectSet Object.

Table of Contents:

- [ScriptObjectSet Object Overview](#)
- [Usage of ScriptObjectSets](#)
- [Configuration Example](#)
- [ScriptObjectSet Object Properties](#)
- [objects \(Array of ScriptObject\)](#)

ScriptObjectSet Object Overview

The **ScriptObjectSet** object within the SFDMU's **export.json** file is a child of the [Script](#) object. It is specifically designed to segment a large migration job into smaller, more manageable sub-jobs.

Each **ScriptObjectSet** can contain a subset of **ScriptObjects**, which are executed

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Speci](#)
- [Export.json File Overview](#)
- [Script Object](#)
- [ScriptObjectSet Object](#)
- [ScriptObject Obiect](#)

ScriptObject Object.

Table of Contents:

- [ScriptObject Object Overview](#)
- [Usage of ScriptObjects](#)
- [ScriptObject Object Properties](#)
 - [afterAddons \(Array of AddonManifestDefinition\)](#)
 - [afterUpdateAddons \(Array of AddonManifestDefinition\)](#)
 - [beforeAddons \(Array of AddonManifestDefinition\)](#)
 - [beforeUpdateAddons \(Array of AddonManifestDefinition\)](#)
 - [bulkApiV1BatchSize \(Integer\)](#)
 - [deleteByHierarchy \(Boolean\)](#)
 - [deleteFromSource \(Boolean\)](#)
 - [deleteOldData \(Boolean\)](#)
 - [deleteQuery \(String\)](#)
 - [excluded \(Boolean\)](#)
 - [excludedFields \(Array of string\)](#)
 - [excludedFromUpdateFields \(Array of string\)](#)
 - [externalId \(String\)](#)
 - [fieldMapping \(Array of MappingItem\)](#)

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [!\[\]\(748ac6c7c535f6b82440890c72a94bd1_img.jpg\) Export.json File Speci](#)
 - [Export.json File Overview](#)
 - [Script Object](#)
 - [ScriptObjectSet Object](#)
 - [ScriptObject Object](#)

ScriptOrg Object.

[!\[\]\(b1e1517aa7f96066f815ccc937da9a22_img.jpg\) Go Back to Export.json File Overview](#)

Table of Contents:

- [ScriptOrg Object Overview](#)
- [Example Configuration with orgs Section](#)
- [ScriptOrg Object Properties](#)
 - [name \(String\)](#)
 - [instanceUrl \(String\)](#)
 - [accessToken \(String\)](#)

ScriptOrg Object Overview

The **ScriptOrg** object within the SFDMU's **export.json** file is a child of the **Script** object. It serves a crucial role in providing connection details for Salesforce organizations involved in the migration.

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [!\[\]\(483b0b1d5025bed5967b2601f9bf655b_img.jpg\) Export.json File Speci](#)
 - [Export.json File Overview](#)
 - [Script Object](#)
 - [ScriptObjectSet Object](#)
 - [ScriptObject Object](#)

MappingItem Object.

[!\[\]\(cb942b8cebb39e39ef3bb5d985720db2_img.jpg\) Go Back to Export.json File Overview](#)

Table of Contents:

- [MappingItem Object Overview](#)
- [Configuring Field Mappings](#)
- [MappingItem Object Properties](#)
 - [sourceField \(String\)](#)
 - [targetField \(String\)](#)
 - [targetObject \(String\)](#)

MappingItem Object Overview

The **MappingItem** object, a child of the [ScriptObject](#) in the SFDMU's **export.json** file, is critical for setting up mappings for both objects and fields between source and target Salesforce environments. It enables precise customization of data mappings, allowing specific fields from the source object to be mapped to corresponding fields in the target.

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [!\[\]\(84ddc717b750a361915d81e72ec2cd39_img.jpg\) Export.json File Speci](#)
 - [Export.json File Overview](#)
 - [Script Object](#)
 - [ScriptObjectSet Object](#)
 - [ScriptObject Object](#)

MockField Object.

[!\[\]\(d447a39d466d76c338e0c282e090b13b_img.jpg\) Go Back to Export.json File Overview](#)

Table of Contents:

- [MockField Object Overview](#)
- [Configuring Data Anonymization](#)
- [MockField Object Properties](#)
 - [name \(String\)](#)
 - [pattern \(String\)](#)
 - [excludedRegex \(String\)](#)
 - [includedRegex \(String\)](#)
 - [excludeNames \(Array of String\)](#)

MockField Object Overview

The **MockField** object within the SFDMU's **export.json** file is a child of the [ScriptObject](#). It plays a vital role in the setup of data anonymization for specific Salesforce objects

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [!\[\]\(f2dbbb2f98159c235612610d361f948a_img.jpg\) Export.json File Speci](#)
 - [Export.json File Overview](#)
 - [Script Object](#)
 - [ScriptObjectSet Object](#)
 - [ScriptObject Object](#)

AddonManifestDefinition Object.

[!\[\]\(898c6822068b3e877f28019387922ee6_img.jpg\) Go Back to Export.json File Overview](#)

Table of Contents:

- [AddonManifestDefinition Object Overview](#)
- [Configuring Add-Ons](#)
- [AddonManifestDefinition Object Properties](#)
 - [module \(String\)](#)
 - [command \(String\)](#)
 - [path \(String\)](#)
 - [description \(String\)](#)
 - [excluded \(Boolean\)](#)
 - [args \(Object\)](#)

AddonManifestDefinition Object Overview



SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
 - [Composite External ID keys](#)
 - [Compound Fields](#)
 - [Data Anonymization](#)

Composite External ID keys.

Table of Contents:

- [Overview](#)
- [Configuration Example](#)
- [How It Works](#)

Overview

Purpose: This feature enables the binding of source and target records by generating a "virtual" External ID field.

This virtual ID is formed by concatenating multiple field values, facilitating operations where unique identifiers are not present in the target object.

Use Case: Consider a `Description` object that lacks a unique field and is associated with two parent objects, `Article` and `Language`. Each `Article` can have multiple `Descriptions`, each in a different `Language`. Both `Article` and `Language` have unique

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

-  [Export.json File Specific](#)
-  [Advanced features](#)
 - [Composite External ID keys](#)
 - [Compound Fields](#)
 - [Data Anonymization](#)

Compound Fields.

Table of Contents:

- [Overview](#)
- [Configuration Example](#)
- [How It Works](#)

Overview

Purpose: This feature enhances the readability and conciseness of SOQL queries by allowing the direct inclusion of compound fields such as `BillingAddress`.

This approach simplifies query writing by abstracting the complexity of specifying each individual component of compound fields.

Related Features: This functionality complements features like the [Field Multiselect Feature](#), which also aim to streamline the creation of SOQL queries.

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
 - [Composite External ID keys](#)
 - [Compound Fields](#)
 - [Data Anonymization](#)

Data Anonymization.

Table of Contents:

- [Overview](#)
- [Feature Configuration](#)
- [Example Configuration](#)
- [Detailed MockField Object Configuration](#)
- [Most Common Anonymization Patterns](#)
 - [Address](#)
 - [Personal Information](#)
 - [Text](#)
 - [Internet](#)
 - [Numbers and Dates](#)
- [Special Anonymization Functions and Patterns](#)
 - [1. `c_seq_number\(prefix, from, step\)`](#)
 - [2. `c_seq_date\(from, step\)`](#)
 - [3. `c_set_value\(value\)`](#)
 - [4. `ids`](#)
 - [RAW_VALUE Keyword](#)
- [Special Regex Expressions](#)

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [!\[\]\(434698d565b0f445468b911a9e387cac_img.jpg\) Export.json File Specific](#)
- [!\[\]\(36215973be8f86ae3b501663ed3564fd_img.jpg\) Advanced features](#)
 - [Composite External ID keys](#)
 - [Compound Fields](#)
 - [Data Anonymization](#)

DeleteHierarchy operation.

Table of Contents:

- [Overview](#)
- [Feature Description](#)
- [Example Configuration for DeleteHierarchy](#)
- [Executing the Deletion](#)

Overview

Purpose: This feature facilitates the deletion of records from the target Salesforce organization, using configurations set in the `export.json` file.

It considers the relationships between objects to enable a hierarchical or relational context for record deletion.

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [!\[\]\(3d0afa5dbfa33662ea780baf91fb56d2_img.jpg\) Export.json File Specific](#)
- [!\[\]\(8660d1a8a5970d25f5e7f77bce115d25_img.jpg\) Advanced features](#)
 - [Composite External ID keys](#)
 - [Compound Fields](#)
 - [Data Anonymization](#)

DeleteSource operation.

Table of Contents:

- [Overview](#)
- [Feature Description](#)
- [Example of DeleteSource Configuration](#)
- [Executing the Deletion](#)

Overview

Purpose: This feature enables the deletion of records directly from the source Salesforce organization, based on a complex selection derived from the entire `export.json` configuration and considering the relationships between objects.

Feature Description

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [!\[\]\(7b0c38b5bc2931612ae54e67b61f7c8d_img.jpg\) Export.json File Specific](#)
- [!\[\]\(212d477d8b584b5320e1f47c84c6fa47_img.jpg\) Advanced features](#)
 - [Composite External ID keys](#)
 - [Compound Fields](#)
 - [Data Anonymization](#)

Export of the Notes, Attachments and Files.

Table of Contents:

- [Overview](#)
- [Exporting Notes and Attachments](#)
- [Exporting Salesforce Files](#)

Overview

Purpose: This guide provides an example configuration for including **Note** and **Attachment** objects in a Salesforce data migration, specifically tailored to export these items along with associated Account records, including the binary data of the Attachments.

Exporting Notes and Attachments

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [!\[\]\(1a4529165ff8133661aae05de9e3cec2_img.jpg\) Export.json File Specific](#)
- [!\[\]\(80555ebce1b49654e5f3b7f44f5d61f0_img.jpg\) Advanced features](#)
 - [Composite External ID keys](#)
 - [Compound Fields](#)
 - [Data Anonymization](#)

Field Multiselect.

Table of Contents:

- [Overview](#)
- [Feature Description](#)
- [Supported Multiselect Keywords](#)
- [Examples of Using Multiselect Keywords](#)

Overview

Purpose: This feature simplifies SOQL query construction by allowing the use of special keywords to automatically include multiple SObject fields based on specific criteria. It is designed to enhance efficiency and maintainability of queries involving large sets of fields.

Feature Description

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

-  [Export.json File Specific](#)
-  [Advanced features](#)
 - [Composite External ID keys](#)
 - [Compound Fields](#)
 - [Data Anonymization](#)

Fields Mapping.

Table of Contents:

- [Overview](#)
- [Example Scenario](#)
 - [Use case](#)
 - [Defining the Field Mapping in the Export Configuration](#)
- [Using an External CSV File for Field Mapping](#)
- [Mapping External Id Field](#)
- [Additional Notes](#)

Overview

Purpose: This feature simplifies data exports to different target data models by allowing the use of field mappings that accommodate differences in field names and structures between the source and target objects.

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
 - [Composite External ID keys](#)
 - [Compound Fields](#)
 - [Data Anonymization](#)

Handling of record ownerId.

Table of Contents:

- [Overview](#)
- [Basic Example: ownerId Assignment](#)
- [Advanced Example: Overriding the Default External ID for the User sObject](#)
- [Filtering User Records](#)
- [Support of Group sObject](#)

Overview

Purpose: This feature allows the assignment and reassignment of the `OwnerId` lookup values of the target records during data migration tasks.

Basic Example: ownerId Assignment

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
 - [Composite External ID keys](#)
 - [Compound Fields](#)
 - [Data Anonymization](#)

Handling of the RecordType.

Table of Contents:

- [Overview](#)
- [Use cases](#)
- [Full RecordType Object Definition](#)
 - [Configuration Details](#)
- [Key Considerations](#)

Overview

Purpose: This feature facilitates the seamless migration of RecordType attributes with records, ensuring that the RecordTypes in the target environment match those in the source.

It handles complexities such as multiple RecordTypes with the same DeveloperName, which can create confusion in the target environment.

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [!\[\]\(143b5c1391b4daa7cc6bf20707023003_img.jpg\) Export.json File Specific](#)
- [!\[\]\(4b447511c2ca94e452381a0ac521d489_img.jpg\) Advanced features](#)
 - [Composite External ID keys](#)
 - [Compound Fields](#)
 - [Data Anonymization](#)

Multiple object sets.

Table of Contents:

- [Overview](#)
- [Example Configuration Using ObjectSets](#)
- [Combined Configuration with Main objects Set](#)
- [Key Considerations](#)

Overview

Purpose: This feature is designed to enhance flexibility in operations by allowing the processing of multiple sub-jobs under one main job.

This is especially useful when multiple operations need to be performed on the same sObject, which is not possible using the standard "objects" array as each object can be specified only once.

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [!\[\]\(5e848a4fde6eee0004648df3a9cb9445_img.jpg\) Export.json File Specific](#)
- [!\[\]\(4bbf562e724f3deedfb62f3e3e252563_img.jpg\) Advanced features](#)
 - [Composite External ID keys](#)
 - [Compound Fields](#)
 - [Data Anonymization](#)

Polymorphic Lookup Fields.

Table of Contents:

- [Overview](#)
- [How to Specify the Parent Object](#)
- [Key Considerations](#)

Overview

Purpose: Polymorphic lookup fields can relate to more than one type of object. Specifying the exact type of SObject as the parent is crucial to accurately retrieve and migrate data involving such fields.

This feature enables the user to specify the parent object type for polymorphic lookup fields such as `ParentId` on `FeedItem`, and `WhatId` and `WhoId` on `Activities`.

[How to Specify the Parent Object](#)

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [!\[\]\(b9d5f6da280406118b88224ac49d17ae_img.jpg\) Export.json File Specific](#)
- [!\[\]\(2060b9955fbfc3deee2085df715b8838_img.jpg\) Advanced features](#)
 - [Composite External ID keys](#)
 - [Compound Fields](#)
 - [Data Anonymization](#)

Target Records Filter.

Table of Contents:

- [Overview](#)
- [Examples of Using Target Record Filter](#)
 - [Example 1: Exclude Records with Null Values](#)
 - [Example 2: Include Only Records with Null Values](#)
 - [Example 3: Include Records Based on Multiple Conditions](#)
- [Understanding the TargetRecordsFilter Syntax](#)

Overview

Purpose: This feature provides an additional method to filter out unwanted records **just before** they are sent to the Target, which is particularly useful when unable to apply the desired filters directly using a WHERE clause in SOQL queries.

Use case: In some scenarios, the data fetched from the source includes records that you

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

-  [Export.json File Specific](#)
-  [Advanced features](#)
 - [Composite External ID keys](#)
 - [Compound Fields](#)
 - [Data Anonymization](#)

Values Mapping.

Table of Contents:

- [Overview](#)
- [Configuration and Basic Transformation](#)
- [Advanced Transformations](#)
 - [Regex Replacement](#)
 - [JS Eval Replacement](#)
 - [RAW_VALUE Keyword](#)
- [Key Considerations](#)

Overview

Purpose: This feature facilitates the transformation of source field values based on predefined mappings before they are updated in the target system. This capability is crucial for scenarios where source data needs to be adjusted to fit target system requirements, such as data type compatibility, data integrity, or specific business rules.

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [!\[\]\(b652f4017f25c4395ec1e13f86187d2c_img.jpg\) Export.json File Specific](#)
- [!\[\]\(4d1d7a5b20f1892d258cbd9a34f67ee5_img.jpg\) Advanced features](#)
- [!\[\]\(4c5f726ba299f3ad98589569bf21b34f_img.jpg\) Add-On API
 - \[Add-On API Overview\]\(#\)
 - \[ExportFiles Core Add-\]\(#\)](#)

Add-On API Overview.

Table of Contents:

- [Overview](#)
- [Why Do You Need the Add-On API?](#)
- [Two Types of Add-On Modules](#)
- [Add-On Module Configuration Skeleton](#)
 - [Sample Configuration](#)
 - [Understanding the Configuration](#)
 - `module` `(String)`
 - `description` `(String)`
 - `args` `(Object)`
- [How to Create and Run Your Own Custom Add-On Module](#)
- [List of Currently Available Core SFDMU Add-On Modules](#)

Overview

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [!\[\]\(a8be76a217f8cde4376e86f8fde6fb68_img.jpg\) Export.json File Specific](#)
- [!\[\]\(9a67a6c43a64c782e3d624c6e52bb268_img.jpg\) Advanced features](#)
- [!\[\]\(88326c8a0f20c6342a66eeb902a69f06_img.jpg\) Add-On API](#)
- [!\[\]\(57d22dafefd7cc45aa11da4a6d292ee2_img.jpg\) Reports](#)

EXAMPLES

Basic examples.

Table of Contents:

- [Example 1. Handling circular references.](#)
- [Example 2. Transfer Accounts related to specific Contacts.](#)
- [Example 3. Limit Contacts to Specific Accounts and Vice Versa](#)
- [Example 4. Exporting Objects with Self-Reference Fields](#)
- [Example 5: Handling Geolocation Fields](#)

Example 1. Handling circular references.

One of the main advantages of SFDMU is that it enables us to migrate complex data models with circular references without any data loss.

Here is a simple example of how SFDMU can insert objects that have circular references between them.

This configuration can be used for various data migration directions: org-to-org, org-to-csv,

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

Advanced examples.

Table of Contents:

- [Example 1. Migration of Accounts and Contacts with AccountContactRelations.](#)
- [Example 2. Migration of FeedItems and FeedComments.](#)
- [Example 3. Locate and delete the same target records which are already deleted in the source.](#)
- [Example 4. Inserting Variation Attributes and Product Variations.](#)
- [Example 5. Upserting Variation Attributes and Product Variations.](#)

Example 1. Migration of Accounts and Contacts with AccountContactRelations. [↑](#)

```
{  
    "objects": [  
        {  
            "query": "SELECT Id, Name FROM Contact",  
            "type": "Account"  
        },  
        {  
            "query": "SELECT Id, Name FROM Contact",  
            "type": "Contact"  
        }  
    ]  
}
```

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [!\[\]\(cf2e48830f516781459b2c41f2f7564d_img.jpg\) Export.json File Specific](#)
- [!\[\]\(d0e318345d3aad0515af976cbe1d4b79_img.jpg\) Advanced features](#)
- [!\[\]\(cb70b0d868721fa999107e749a2ffe84_img.jpg\) Add-On API](#)
- [!\[\]\(acb476b05dc59a35543c3fd80ad17489_img.jpg\) Reports](#)

EXAMPLES

CSV import-export examples.

Table of Contents:

- [Example 1. Import AccountBrand.csv without providing Account.csv.](#)
- [Example 2. Update object having composite external Id key.](#)
- [Example 3. Import mixed Person Accounts and Business Accounts from the same Account.csv.](#)
- [Example 4. Resetting target field values using CSV files.](#)

Example 1. Import AccountBrand.csv without providing Account.csv.

When the linked Accounts is already exist in the target org, you can simple do:

```
{  
  "objectSets": [  
    {
```

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

File data transfer examples.

Table of Contents:

[Example 1. Migration of Feed Attachments along with Feed Item or Feed Comment.](#)

Example 1. Migration of Feed Attachments along with Feed Item or Feed Comment. [↑](#)

Below is example of import (**Insert**) chatter attachments with chatter feed items using the **core:ExportFiles** Add-On module.

The SFDMU will automatically detect and maintain relationship between the parent records and attachments.

```
{  
    "objects": [  
        {  
            "operation": "Readonly",
```

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [!\[\]\(7703a12e9ae6db41fdad3baa80e2e9fa_img.jpg\) Export.json File Specific](#)
- [!\[\]\(512f63fc808fd4c06338bdd0937d22c9_img.jpg\) Advanced features](#)
- [!\[\]\(4fec3ce9d2e53c51bd3fef27d0a19891_img.jpg\) Add-On API](#)
- [!\[\]\(555d0a599fbee2067a9799100e4fc91d_img.jpg\) Reports](#)

EXAMPLES

Record transformation examples.

Table of Contents:

- [Example 1. Assign a default User \(e.g. OwnerId lookup\) to target record.
Using the core:RecordsTransform module.
Using the ValueMapping.](#)
- [Example 2. Clear all text fields in an sObject.](#)
- [Example 3. Assign constant value to specific field.](#)
- [Example 4: Reset All Checkbox Field Values to False](#)
- [Example 5. Populate Text Field in Child Object with Parent Record ID \(Parent-Child Relationship\)](#)
- [Example 6. Populate Text Field with Values Taken from Both Source and Target Orgs](#)

Example 1. Assign a default User (e.g. OwnerId lookup) to target record.

The configurations below assign default user Id to OwnerId lookup if the given user does

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [!\[\]\(cabaa2f1621263ebd47b0c67bd921221_img.jpg\) Export.json File Specific](#)
- [!\[\]\(4c9a401cd2734350331b4e4a5c577f9a_img.jpg\) Advanced features](#)
- [!\[\]\(6e5d8b6354949196c5ded5db099e4d04_img.jpg\) Add-On API
 - \[Add-On API Overview\]\(#\)
 - \[ExportFiles Core Add-\]\(#\)](#)

Add-On API Overview.

Table of Contents:

- [Overview](#)
- [Why Do You Need the Add-On API?](#)
- [Two Types of Add-On Modules](#)
- [Add-On Module Configuration Skeleton](#)
 - [Sample Configuration](#)
 - [Understanding the Configuration](#)
 - `module_(String)`
 - `description_(String)`
 - `args_(Object)`
- [How to Create and Run Your Own Custom Add-On Module](#)
- [List of Currently Available Core SFDMU Add-On Modules](#)

Overview

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [!\[\]\(deb0db8a4337aac54a92aa1d2615fa0b_img.jpg\) Export.json File Specific](#)
- [!\[\]\(792c8873299a696d5ea6b0e022de9aa7_img.jpg\) Advanced features](#)
- [!\[\]\(5244864bead43941176199a51b3f592d_img.jpg\) Add-On API](#)
 - [Add-On API Overview](#)
 - [ExportFiles Core Add-On Module](#)

ExportFiles Core Add-On Module.

Table of Contents:

- [Overview](#)
- [Minimal ExportFiles Module Setup](#)
- [Full ExportFiles Module Setup](#)
 - [deleteOldData \(Boolean\)](#)
 - [operation \(String\)](#)
 - [externalId \(String\)](#)
 - [sourceWhere \(String\)](#)
 - [targetWhere \(String\)](#)
 - [contentDocumentLinkOrderBy \(String\)](#)
 - [maxChunkSize \(Number\)](#)
 - [maxFileSize \(Number\)](#)
- [Additional Notes](#)

Overview ↑

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [!\[\]\(64938096b7968e9032f68f41d665b89e_img.jpg\) Export.json File Specific](#)
- [!\[\]\(55b068b431b5319fade9b43f80b99ad1_img.jpg\) Advanced features](#)
- [!\[\]\(3cb8aa82d02d6c9488caf1ae9590dc61_img.jpg\) Add-On API](#)
 - [Add-On API Overview](#)
 - [ExportFiles Core Add-On Module](#)

Records Transform Core Add-On Module.

Table of Contents:

- [Overview](#)
- [Sample of Record Transformation](#)
 - [Configuration File](#)
 - [Object Model](#)
 - [Requirements](#)
 - [Solution](#)
 - [Step 1: Define the Source Fields](#)
 - [Step 2 Setup lookup relationships.](#)
 - [Step 3 Define the Target Fields and Transformation Expressions](#)

Full Specification of args Properties

- [fields \(Array of Field objects\)](#)
 - [Field.alias \(String\)](#)
 - [Field.sourceObject \(String\)](#)
 - [Field.sourceField \(String\)](#)
 - [Field.includeFields \(Array of strings\)](#)
 - [Field.valueSource \(String\)](#)

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [!\[\]\(862f54ad282fb94939fdbdda96030568_img.jpg\) Export.json File Specific](#)
- [!\[\]\(aac02eaa26a9cdf6ca2f46641e6d657b_img.jpg\) Advanced features](#)
- [!\[\]\(e0d03d7116796abcbe06e48d7f025123_img.jpg\) Add-On API
 - \[Add-On API Overview\]\(#\)
 - \[ExportFiles Core Add-On Module\]\(#\)](#)

RecordsFilter Core Add-On Module.

Table of Contents:

- [Overview](#)
- [Example of Module Setup](#)
- [RecordsFilter Module Setup](#)
 - [filterType \(String\)](#)
 - [settings \(Object\)](#)
- [Available Filters](#)
- [Additional Notes](#)

Overview

The `core:RecordsFilter` module enhances data transfer capabilities by offering advanced filtering options for source records before they are uploaded to the target system.

It enables precise selection and management of data, ensuring that only relevant records

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [!\[\]\(17ce7f92c2822d00d43cd5623e780c7c_img.jpg\) Export.json File Specific](#)
- [!\[\]\(66a645b84eac8a4c85174aa3d60ece4e_img.jpg\) Advanced features](#)
- [!\[\]\(5665ef02e236297aefeda2ae346da2e5_img.jpg\) Add-On API
 - \[Add-On API Overview\]\(#\)
 - \[ExportFiles Core Add-On Module\]\(#\)](#)

RecordsFilter - Bad Words Filter.

Table of Contents:

- [Overview](#)
- [BadWords Filter Setup](#)
 - [badwordsFile \(String\)](#)
 - [detectFields \(String\[\]\)](#)
 - [highlightWords \(Boolean\)](#)
 - [outputMatches \(Boolean\)](#)
- [Additional Notes](#)

Overview

The **BadWords** filter scans the object's fields listed in the **detectFields** for unwanted words, then can output the matched words to the console or even modify the source field value to highlight these words, then upload the modified value to the target (or write to the CSV file).

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

-  [Export.json File Specific](#)
-  [Advanced features](#)
-  [Add-On API](#)
 - [Add-On API Overview](#)
 - [ExportFiles Core Add-On Module](#)

Supported Add-On Api events.

Table of Contents:

- [List of Supported Events Available for Triggering Add-On Modules](#)
 - [OnBefore](#)
 - [OnAfter](#)
 - [OnDataRetrieved](#)
 - [Object.OnBefore](#)
 - [Object.OnAfter](#)
 - [Object.OnFilterRecords](#)
 - [Object.OnBeforeUpdate](#)
 - [Object.OnAfterUpdate](#)

[The Add-On Event and Script Execution Lifecycle](#)

List of Supported Events Available for Triggering Add-On Modules [↑](#)

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [!\[\]\(981236151edba252333289b82e8f89e4_img.jpg\) Export.json File Specific](#)
- [!\[\]\(409aafb48e949a2b998357b94a5f8ec7_img.jpg\) Advanced features](#)
- [!\[\]\(d4b400de1cf655bcd9513cfe346f044d_img.jpg\) Add-On API](#)
 - [Add-On API Overview](#)
 - [ExportFiles Core Add-On Module](#)

Custom Add-On Api.

Table of Contents:

- [Overview](#)
- [Example of a Custom SFDMU Add-On Module](#)
- [How to Try This Module](#)
 - [Step 1: Install](#)
 - [Step 2: Compile](#)
 - [Step 3: Create Root Directory and Copy Necessary Files](#)
 - [Step 4: Create New export.json File](#)
 - [Step 5: Create New Custom Fields](#)
 - [Step 6: Create New Account Record](#)
 - [Step 7: Launch the Migration Job](#)
 - [Step 8: Check the Results](#)

- [Full Specification of the Add-On Event Declaration](#)
- [List of Supported Add-On Events and the Execution Lifecycle](#)
- [How to Create and Run Your Own Custom SFDMU Add-On Module](#)
 - [Step 1: Create Module Directory](#)
 - [Step 2: Compile](#)
 - [Step 3: Copy Files to the Module Directory](#)
 - [Step 4: Create New export.json File](#)

The SFDMU Custom Add-On API

The Custom SFDMU Add-On Object Reference.

The [Custom SFDMU Add-On Module Api](#) allows you to build your own SFDMU Add-On extensions allowing you to extend the SFDMU with unlimited new custom features which may be required by your business.

Here you can find the detailed documentation about the object model used by the **Custom Add-On Api**.

See also:

[Introduction to the SFDMU Add-On Api Engine](#)

[The Export Files Core Add-On Module](#)

[The Custom SFDMU Add-On Module Api](#)

[Full export.json format](#)

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)
- [Identifying Issues in CSV Files with the Plugin.](#)

Identifying Issues in CSV Files with the Plugin.

When setting CSV files as a data source (`--sourceusername csvfile`), all detected issues within the source CSV files are documented in the **CSVIssuesReport.csv** file.

This file is located in the same directory as the `export.json` file.

Notes:

- Ensure that the flag `importCSVFilesAsIs = false` is either unset or explicitly set to false. Setting this flag to true disables the automatic correction of "broken" CSV files and prevents the generation of the **CSVIssuesReport.csv**.
- For projects involving multiple Object Sets, beginning with Object Set #2, the **CSVIssuesReport.csv** file for each set is located in a specific subdirectory. For instance, for Object Set #2, find the report at `reports/object-set-2/CSVIssuesReport.csv`, relative to where the `export.json` file sits.

Example of working folder structure:

```
| - export.json
```

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)
 - [Identifying Issues in CSV Files](#)

Handling Missing Parent Records in Plugin Migrations.

Occasionally, the Plugin may encounter issues binding child lookup records to their parent records using the parent's ExternalId key, as defined in the `export.json` file. This issue can cause the migration job to fail.

To assist you in analyzing why your migration job failed, the Plugin generates a report for all such 'broken' records in a **MissingParentRecordsReport.csv** file.

This file is located in the same directory as the `export.json` file.

Notes:

- For projects involving multiple Object Sets, starting from Object Set #2, the `MissingParentRecordsReport.csv` file can be found in the corresponding subdirectory for each set. For example, for Object Set #2, the report will be located at **reports/object-set-2/MissingParentRecordsReport.csv**, relative to where the `export.json` file sits.

Example of working folder structure:

SFDMU OVERVIEW

[Key Features of the SFDMU](#)

[Get Started](#)

[Installation](#)

[Configuration](#)

[Running](#)

[Debugging](#)

[SFDMU GUI Application](#)

FULL DOCUMENTATION

 [Export.json File Specific](#)

 [Advanced features](#)

 [Add-On API](#)

 [Reports](#)

[Identifying Issues in CSV Files](#)

Target CSV Files in SFDMU.

When the setting **createTargetCSVFiles = true** is enabled or left unset, the Plugin generates CSV files that include all records as they were sent to the Salesforce API for processing.

Each file is named following a specific pattern, for example, `TestObject_Description__c_delete_target.csv`, indicating the **object type** and **operation** type.

Conversely, setting **createTargetCSVFiles = false** disables the generation of these target report files.

Notes:

- The Plugin incorporates an additional **Errors** column in each CSV file. This column is left blank for records that have been successfully processed and contains an error message for any records that failed.
- All target files are stored in the **target** subdirectory, which is located in the same directory as the `export.json` file.
- For the main object set (object set index = 1), CSV files are always written to the main **target** directory.
- For scripts containing multiple object sets, starting from Object Set #2, SFDMU

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

-  [Export.json File Specific](#)
-  [Advanced features](#)
-  [Add-On API](#)
-  [Reports](#)
- [Identifying Issues in CSV Files with the SFDMU](#)

Source (Working Copy) CSV Files in SFDMU.

When CSV files are set as a data source, SFDMU analyzes and attempts to repair the provided raw source CSV files. After processing, the final versions are stored in a special directory. These processed files become the working copies, which are then used to import data into the target organization.

You can examine these files in the **source** subdirectory, located in the same directory as the `export.json` file, to understand how SFDMU processes your CSVs.

Notes:

- Each file is appended with the postfix **_source**, for example, **Account_source.csv**.
- If multiple Object Sets are used, the sources for the first Object Set are placed in the main **source** subdirectory. The sources for subsequent Object Sets, starting from Object Set #2, are stored in their respective directories, for example, **source/object-set-2/Account_source.csv**, relative to where the `export.json` file sits.

Example of working folder structure:

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

-  [Export.json File Specific](#)
-  [Advanced features](#)
-  [Add-On API](#)
-  [Reports](#)
- [Identifying Issues in CSV Files during Migrations](#)

Log File Management in Plugin Migrations.

The Plugin captures events triggered during the execution of a migration job in a **.log** file.

Each migration job generates its own log file to ensure detailed tracking of events.

The name of each log file follows the pattern **yyyy-MM-dd__HH_mm_ss.log**, which helps in identifying the exact time of the events logged.

All log files are systematically stored in the **logs** subdirectory, located in the same directory as the `export.json` file.

Example of working folder structure:

```
|- export.json
|- logs/
    |- 2024-04-24__02_17_22.log
    |- 2024-04-25__15_43_07.log
    |- ...
```

SFDMU OVERVIEW

[Key Features of the SFDMU](#)

[Get Started](#)

[Installation](#)

[Configuration](#)

[Running](#)

[Debugging](#)

[SFDMU GUI Application](#)

FULL DOCUMENTATION

[!\[\]\(6e5da4e84f9166ffba1fb1ee25bf9646_img.jpg\) Export.json File Specific](#)

[!\[\]\(71df6973b13776ad46bc52cd21c3ed43_img.jpg\) Advanced features](#)

[!\[\]\(eee8af8ae516448946c14152f2f222c3_img.jpg\) Add-On API](#)

[!\[\]\(2203b124cc5351648033e92a1c41e89a_img.jpg\) Reports](#)

EXAMPLES

Support policy.

Dear Friends,

We're thrilled that you've chosen **SFDMU** and hope it greatly enhances your work!

IMPORTANT:

THIS IS NOT AN OFFICIAL SALESFORCE PROJECT.

IT IS DEVELOPED AS A FREE OPEN-SOURCE CONTRIBUTION TO THE SALESFORCE COMMUNITY.

WHILE SALESFORCE HOLDS THE COPYRIGHTS, IT IS NOT INVOLVED IN THE DEVELOPMENT OR SUPPORT OF THIS PROJECT.

For assistance, please adhere to the following guidelines:

- **Important:** We offer NO guarantees of support and handle issues solely on our own time, without a service level agreement (SLA). While we aim to address issues promptly, please be aware that responses may often be delayed. Your

SFDMU OVERVIEW

- [Key Features of the SFDMU](#)
- [Get Started](#)
- [Installation](#)
- [Configuration](#)
- [Running](#)
- [Debugging](#)
- [SFDMU GUI Application](#)

FULL DOCUMENTATION

- [Export.json File Specific](#)
- [Advanced features](#)
- [Add-On API](#)
- [Reports](#)

EXAMPLES

Code contribution policy.

Thank You for Your Interest in Contributing to SFDMU!

We greatly appreciate your willingness to contribute code to SFDMU. To streamline the process and make your contributions as efficient as possible, please follow these guidelines:

- **Check Existing Issues:** Before starting work, review our issue tracker to see if your bug or suggestion has already been reported or discussed. If you're unsure about whether your changes will be accepted, please contact us beforehand to avoid unnecessary effort.
- **Scope of Changes:** We primarily accept minor, cosmetic modifications to the core SFDMU code. For more complex enhancements that might benefit other users, please share your ideas with us first. Depending on the proposal, we may update the SFDMU Core or develop a new Core Add-On Module using our [Add-On API Engine](#).
- **Custom Add-Ons:** If you need a specific feature for your own business, consider developing a [Custom Add-On Module](#) and maintaining it locally without submitting a

SFDX-Data-Move-Utility / LICENSE.txt



forcedotcom/SFDX-Data-Move-Utility is licensed under the

BSD 3-Clause "New" or "Revised" License

A permissive license similar to the BSD 2-Clause License, but with a 3rd clause that prohibits others from using the name of the copyright holder or its contributors to promote derived products without written consent.

Permissions

- ✓ Commercial use
- ✓ Modification
- ✓ Distribution
- ✓ Private use

Limitations

- ✗ Liability
- ✗ Warranty

Conditions

- ⓘ License and copyright notice

This is not legal advice. [Learn more about repository licenses](#)



hknokh updates

c408c72 · 6 years ago



12 lines (7 loc) ...

Code

Blame

Raw



```
1 Copyright (c) 2020, Salesforce.com, Inc.
2 All rights reserved.
3
4 Redistribution and use in source and binary forms, with or without modification,
5
6 * Redistributions of source code must retain the above copyright notice, this li
7
8 * Redistributions in binary form must reproduce the above copyright notice, this
9
10 * Neither the name of Salesforce.com nor the names of its contributors may be us
11
12 THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
```

