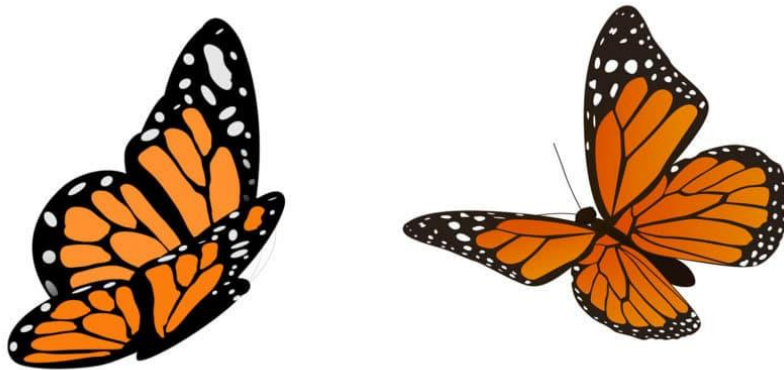# Snapshot Best Practices:
Relational Data Migration

Snapshot provides powerful tools for migrating connected sets of data records between Salesforce orgs. This capability is useful for backing up data, refreshing Sandboxes, merging orgs, and populating orgs with test data for application development. The Bulk API is used for all transactions to ensure that very large datasets can be moved efficiently. We have moved millions of records while testing the data migration tools in Snapshot.

When records are migrated between orgs, all of the internal relationships are preserved. External references in the dataset are also connected to matching objects on the destination. This whitepaper presents detailed information on how to build and migrate datasets. We also document the CSV format that Snapshot uses for importing datasets from other systems or creating them with a spreadsheet editor.

Source and Destination Orgs

When you select the Deployment Arrow between any two connected Snapshots, the Options Menu will display various commands. You can also right-click the Deployment Arrow to see the options in a popup menu. The third submenu will have options to Build, Migrate, and Manage Datasets. If you do not see these options, then perhaps the Deployment Arrow is connected to a Developer Project. They do not have any actual data and cannot be used as a source to build a dataset or as a destination for migration.
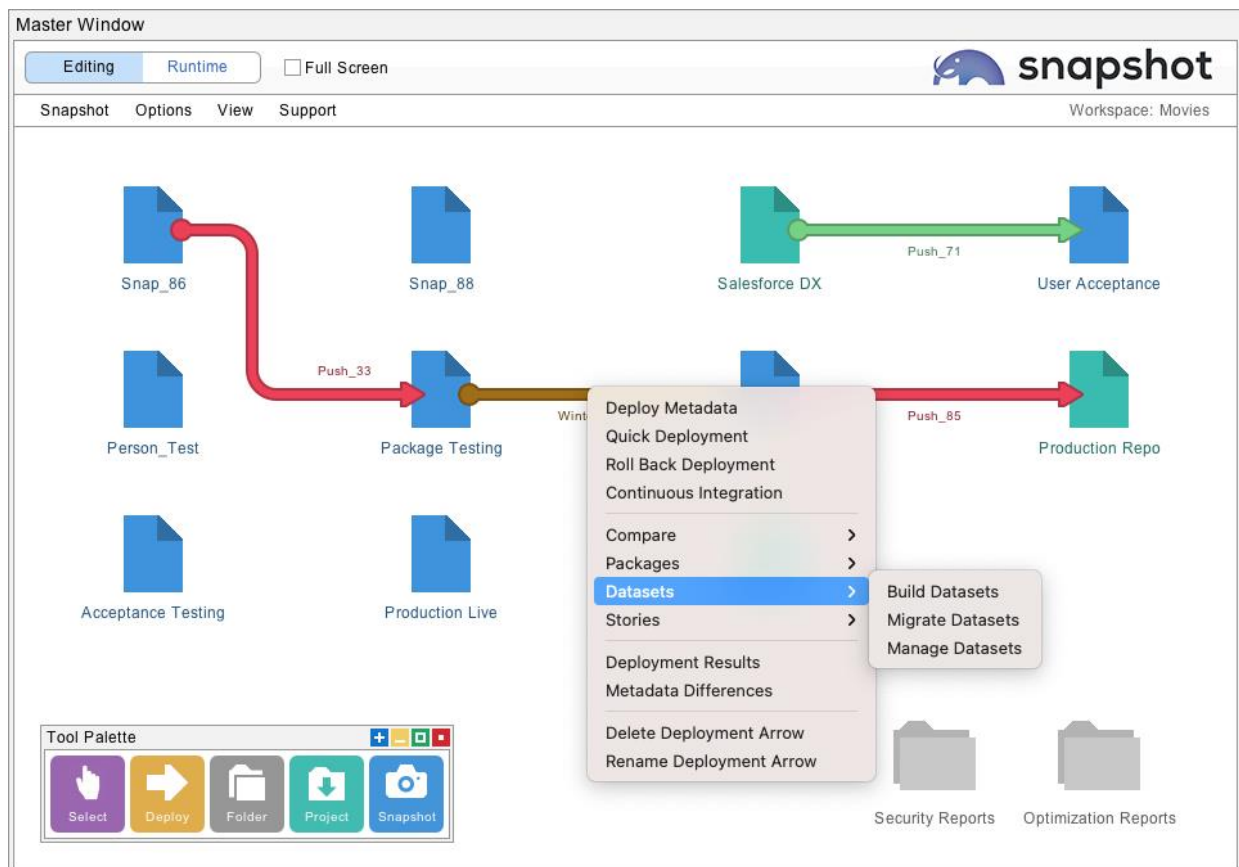
The option to Build Datasets will use the source Salesforce org to download multiple records in the form of XML files to your local machine. The option to Migrate Datasets will insert and update records from the selected dataset into the destination Salesforce org. The option to Manage Datasets can be used to create a dataset from CSV files, remap field names, remap field values, create external IDs, rename the dataset, duplicate the dataset, import CSV files, and export CSV files.

Build Datasets Dialog

The first tab of the Build Dataset dialog allows you to select the dataset and template that you would like to work with. If you are moving records associated with a managed package, then check out the Prebuilt Templates menu for help moving those records. For example, Salesforce CPQ, Health Cloud, and Government Cloud all have custom templates.
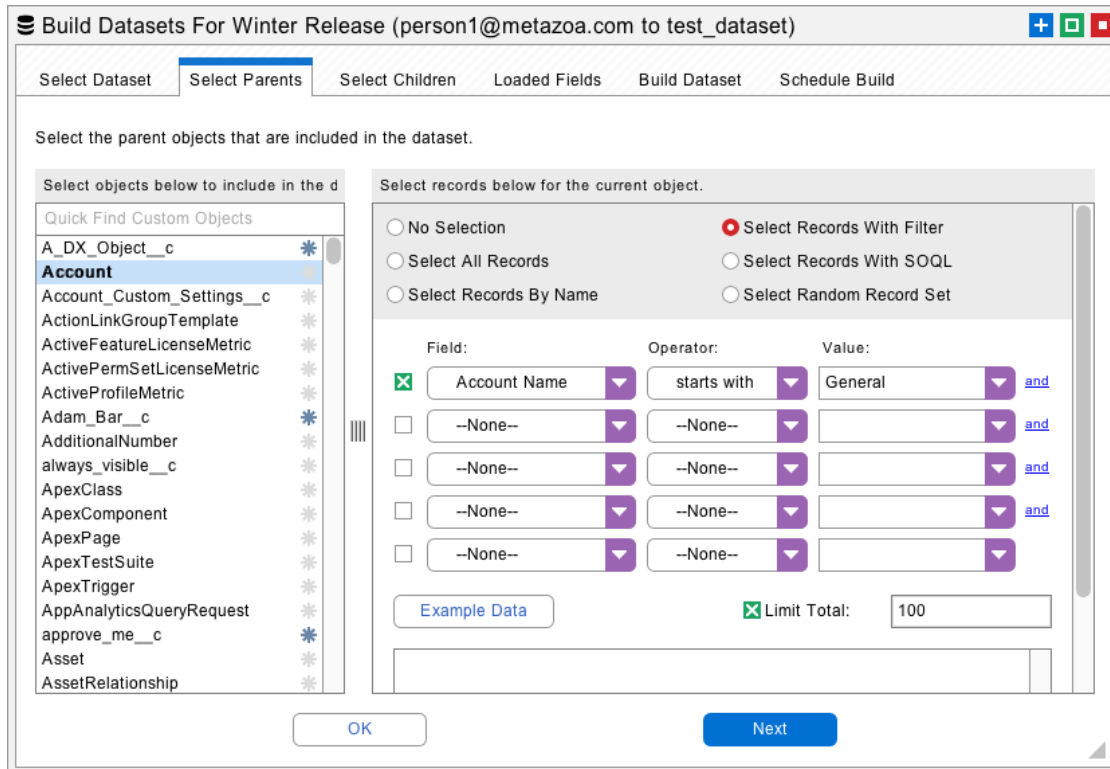
Selecting Parents and Children

The next tab allows you to select the parent records that you want to include in the dataset. These records are available on the source Salesforce org. You can select all records, a subset of records by name, or a subset of records using a complex filter. There is also an option to write a SOQL query by hand. The total number of downloaded records can be limited. This is useful for grabbing a random subset of records for acceptance testing or application development.

The next tab allows the selection of connected child objects for each parent object. When a dataset is created, the selected parent records are loaded first, followed by all the children connected to that parent. You can specify multiple child objects in a hierarchy. The relationship field used to associate each parent and child is shown in parenthesis. The internal relationships between parent and child are always preserved when the dataset is migrated.
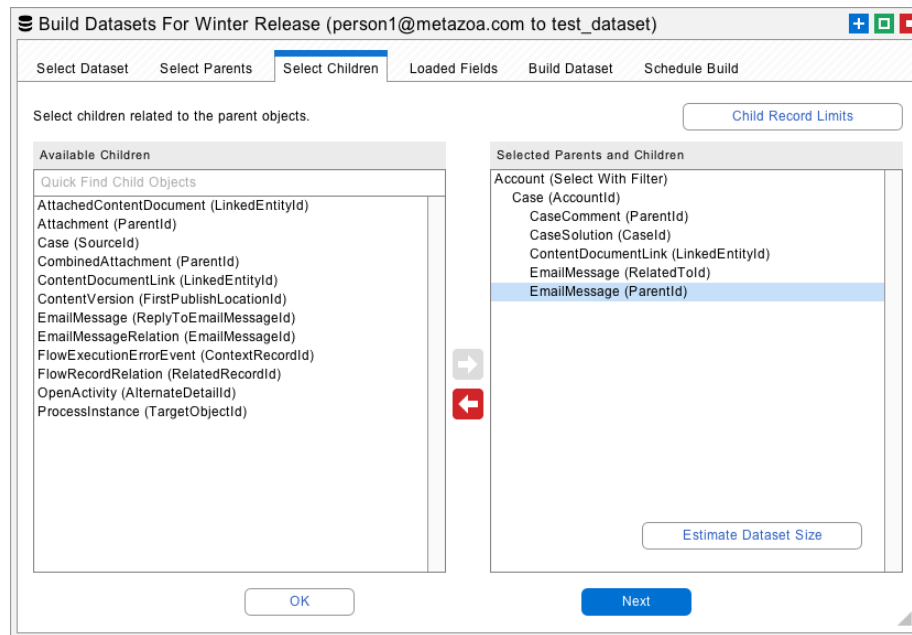
Notice the Child Record Limits button at upper right. This feature allows you to add a filter or limit the number of records for the selected child object. Also see the Estimate Dataset Size button at lower right. This option will calculate the estimated number of records and total size of the dataset before it is actually created.

Loaded Fields

After that, you can select fields to load for each parent and child object. The parent and child objects that were selected will be above the dividing line at left. You can choose fields that need to be loaded by moving them to the list at right. Snapshot will automatically figure out the best fields to load, so you may not need to do anything on this tab. Fields that cannot be created or updated on the destination do not usually need to be loaded. Removing unwanted fields makes your dataset smaller in size and easier to migrate.
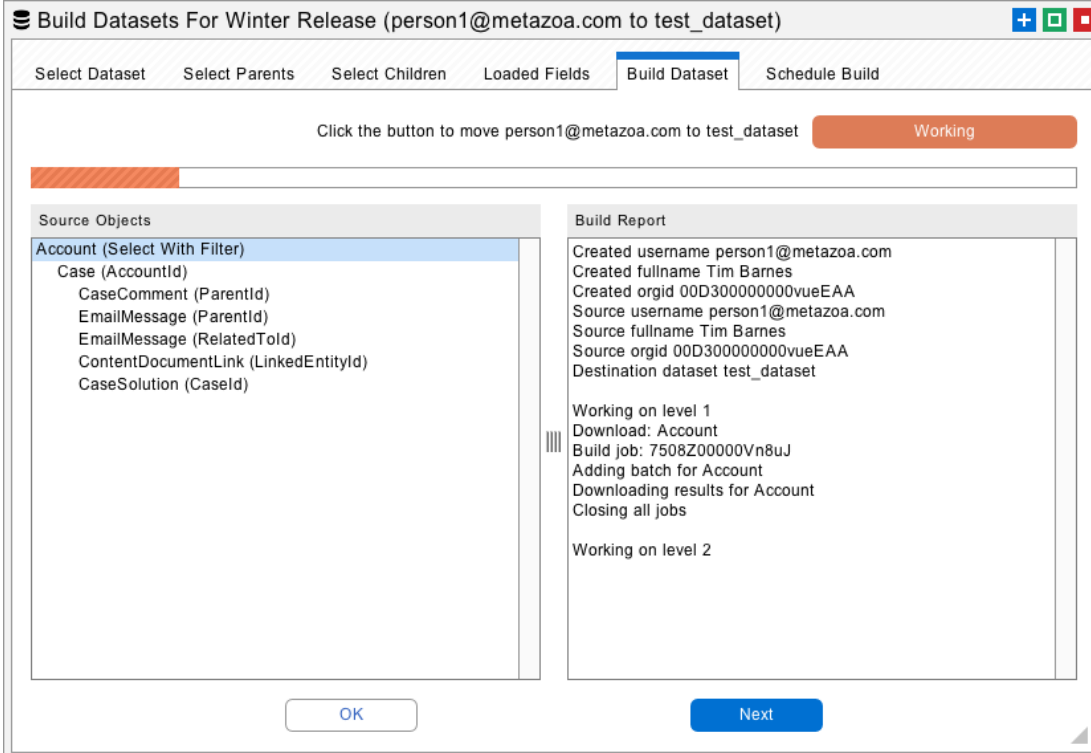

Snapshot also uses the loaded fields to identify matching records in the destination org during migration. If a source record matches a destination record, then the corresponding destination record is updated. If a source record does not match any destination record, then a new destination record is created.

Snapshot automatically selects the most common matching fields for you. However, you can also manually select matching fields. For example, if you want to match Accounts by Name and BillingCity, then be sure that both Name and BillingCity are loaded. You will be able to specify the exact list of matching fields that you want to use in the Migrate Datasets dialog.

Underneath the dividing line you will see "external references" to other objects in the destination org. For example, if you select Opportunity objects for migration then you will see an external reference to Campaigns, because the Opportunity object contains a CampaignId reference field. Snapshot will automatically connect external references to matching objects on the destination org when the dataset is migrated. You can manually specify additional fields needed for matching external references as well.

Build Datasets Button

The next tab allows you to enter the name of a new dataset and then click the Build Datasets button at right to start the download process. If you select an existing dataset name from the menu then that dataset will be replaced. All of the download results will be listed in the window pane at lower right.

The XML files with your dataset information will be saved in the "datasets" folder next to the "workspaces" folder in the Snapshot file system. Datasets are globally available for migration to any destination org. In other words, any source org can be used to build a dataset, and any dataset can be migrated to the destination org.

The last tab allows you to Build Datasets at a specific time in the future or as a recurring event. When a scheduled dataset is created, it will automatically replace the current dataset by name. Unlike Snapshots, datasets are not maintained in a time series.

Manage Datasets Dialog

The Manage Datasets dialog has some very helpful utilities for creating and transforming your datasets. From this interface you can create a dataset from CSV files, remap field names, remap field values, create external IDs, rename the dataset, duplicate the dataset, import CSV files, and export CSV files. The next section discusses the CSV file format that is used by the create, import, and export options. If you don't need that information right now, you might want to skip the next section. After that, we discuss all the options in the Manage Datasets interface.

CSV File Format

The CSV file format encodes all the field values as well as the internal and external relationships in the dataset. This format expects the first row of CSV data to be field names, followed by additional rows for each record. The columns must include the field name "Id" to specify the source record ID, and the field name "objtype" to specify the object type. Here is an example below of a CSV file with two Account records:

	A	B	C
1	Id	objtype	Name
2	0011W00002MtiupQAB	Account	Honeywell
3	0011W00002MtiuoQAB	Account	Accenture
4			
5			

The rest of the columns can be used for other Account fields, in this example we only have the Account Name.

Now let's look at another CSV file with Contact data. This file defines two Contacts and the Account that they are related to. The AccountId, AccountId.Name and AccountId.objtype fields contain information about the related Account. In this case, the Account Name can be selected as a matching field to find the Account. In our example, one Contact will connect to Accenture, and the other Contact will connect to Honeywell if these records exist in the destination org.

	A	B	C	D
1	**Id**	**objtype**	**FirstName**	**LastName**
2	0031W00002k3DeRQAU	Contact	Bob	Jones
3	0031W00002jHTsJQAW	Contact	Ann	Smith
4				

E	F	G
AccountId	**Account.Name**	**Account.objtype**
0011W00002MtiupQAB	Honeywell	Account
0011W00002MtiuoQAB	Accenture	Account

The required reference information is easy to include in the CSV file. For example, you could use the Salesforce SOQL Query below to capture all the information needed to create our example CSV data for Accounts and Contacts:
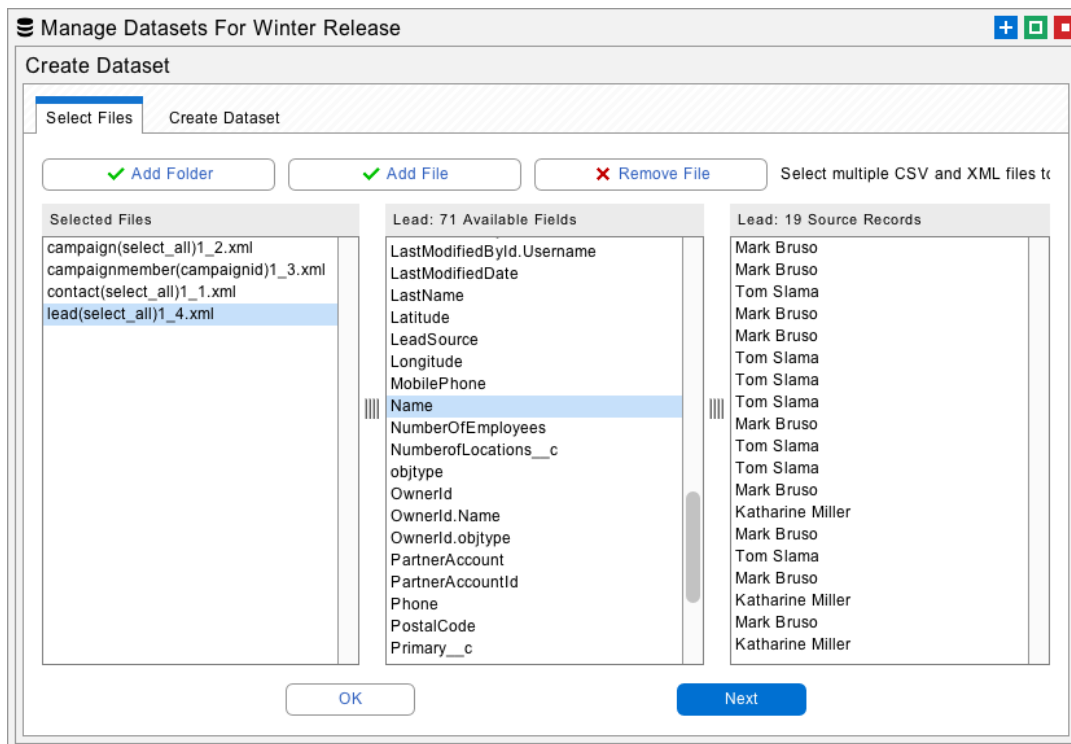
```
SELECT Id,Name FROM Account

SELECT Id,FirstName,LastName,AccountId,Account.Name FROM Contact
```

Create Datasets

Click the Create button on the Manage Datasets dialog to open an interface that allows you to import files and folders. This information will be used to create a new dataset. The files should be in the CSV file format discussed above. You can also import XML files. Simply go to the dataset folder and select any group of XML files from an existing dataset. This is useful if you want to migrate an existing dataset in sections.

The imported files are added to the list at left. You can right-click the items here and adjust their order. This is useful for making sure that parent objects are processed before child objects. In the previous example, you would want to make sure that Accounts are listed before Contacts. The imported fields and source records are visible in the lists at right.

To create a new dataset, go to the next tab, give your new dataset a name, and click the Create Dataset button. This will add the new dataset to the list on the first tab of the Manage Datasets dialog.
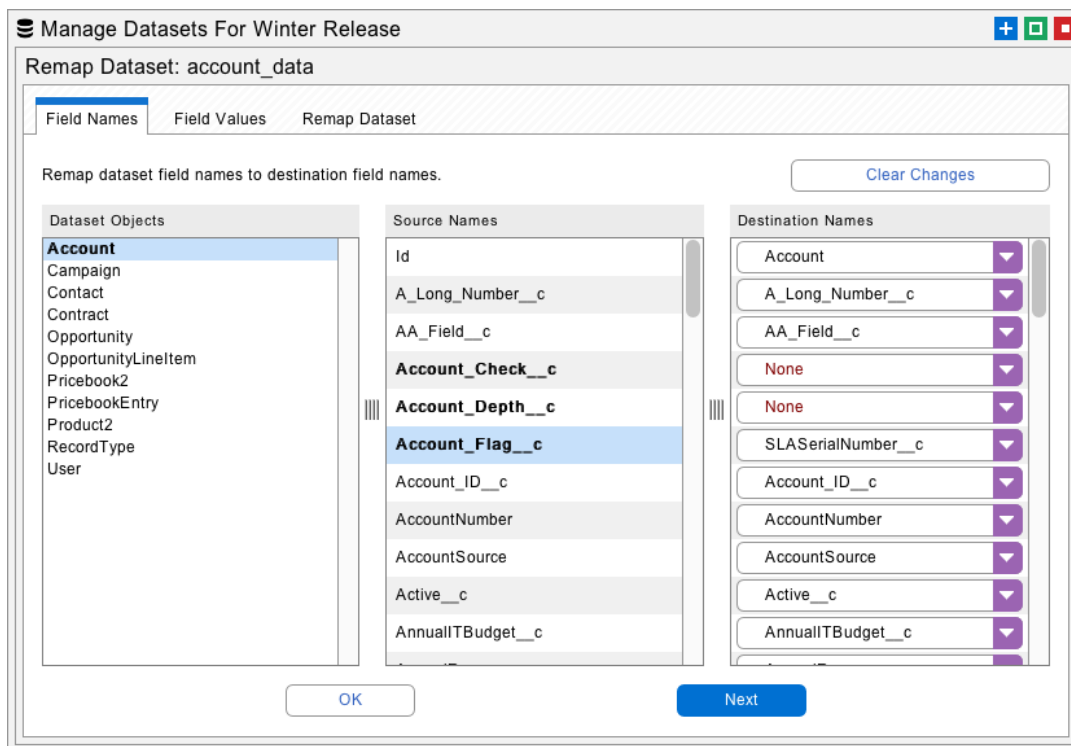
Remapping Field Names

Click the Remap button on the Manage Datasets dialog to open an interface that will remap field names and values. Remapping field names is useful when you want to migrate the data between fields with different names. For example, you might need to merge two different custom objects.

To use this feature, select the object to be remapped at left. Select the field name to remap in the center and select the destination field at right. The fields at right will be defined by the available objects and fields in the destination org. You might have to select an arrow that has the desired destination org for this to work as expected.

Proceed to the Remap Dataset tab and click the Remap Dataset button. This will transform all the field names in the selected dataset. Now the field names in this dataset will match the destination org.

Remapping Field Values

Click the Remap button on the Manage Datasets dialog to open an interface that will remap field names and values. Remapping field values is useful when there are username transformations, picklist differences, and email suffixes that need to be matched against the destination org.

To use this feature, select the object and field to be remapped at left. Select the source value in the center and type in a destination value at right. You can import and export all the destination values as text files. Right-click the destination for editing options.

Proceed to the Remap Dataset tab and click the Remap Dataset button. This will transform all the field values in the selected dataset. Now this dataset can be migrated to the destination and the new field values will match.
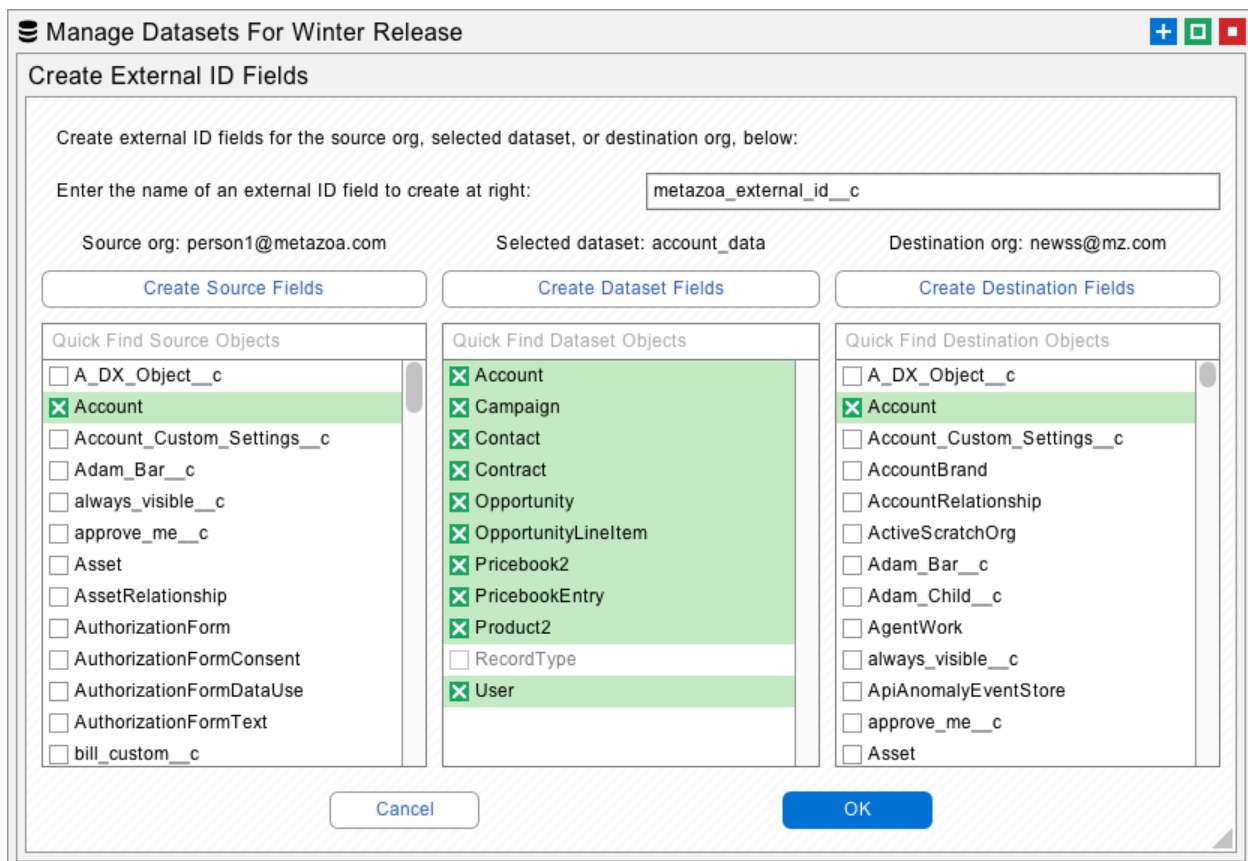
Create External IDs

Click the External button on the Manage Datasets dialog to open an interface that will create external IDs on the source org, the dataset itself, or the destination org. When you create an external ID on the source org, Snapshot creates a custom **formula** field that sets the field value equal to the object ID. When you create an external ID on the destination org, then Snapshot creates a custom **text** field that receives the source ID value. You can give the new custom field any name that you like. Field Level Security is automatically set for the System Administrator Profile and the field is hidden from everyone else.

Sometimes administrators don't want to create a bunch of external IDs in the source org. That might be disruptive if the org is in active use. In this case, Snapshot offers the capability to create an external ID on the dataset itself, as if the external ID was actually from the source org. This works exactly the same way when the destination records are tagged or matched with the source ID. The beauty of this is that you can fully exploit the power of migrating data with external IDs without actually having any external ID fields in the source org.

Import and Export Datasets

Click the Import button on the Manage Datasets dialog to open an interface that will import a dataset as a local file or from shared storage. Click the Export button to open an interface that will export a dataset as a local file or to shared storage. The local files will be saves as a zipped dataset folder. The shared storage uses Salesforce Content folders and documents that are saved in the Snapshot License Org. This is an easy way to archive datasets or share them with your administrative team.
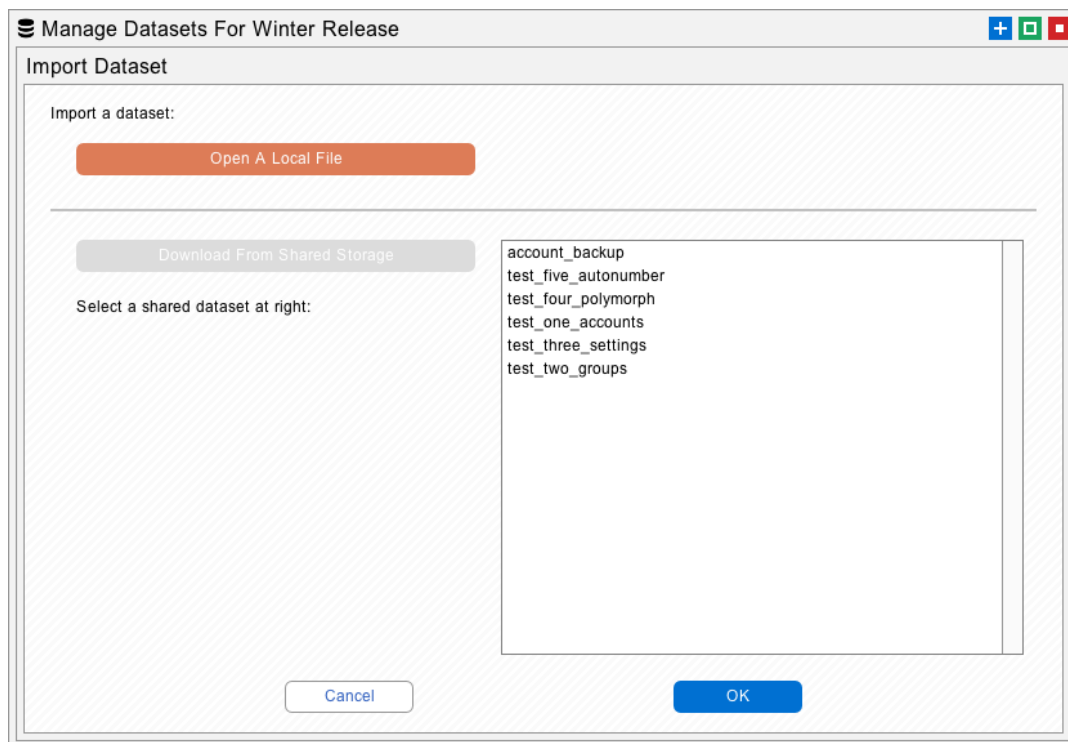
Migrate Datasets Dialog

After a dataset has been created, you are ready to migrate these records to a destination Salesforce org. Right-click a Deployment Arrow that is connected to the correct destination org and select the Migrate Datasets option to get started. The Migrate Datasets dialog allows you to select any of the global datasets from the list at left and see the objects and fields that are available in the list at right. The next four tabs provide options for matching fields, scrambling fields, deactivating assets, and finally migrating the selected dataset.
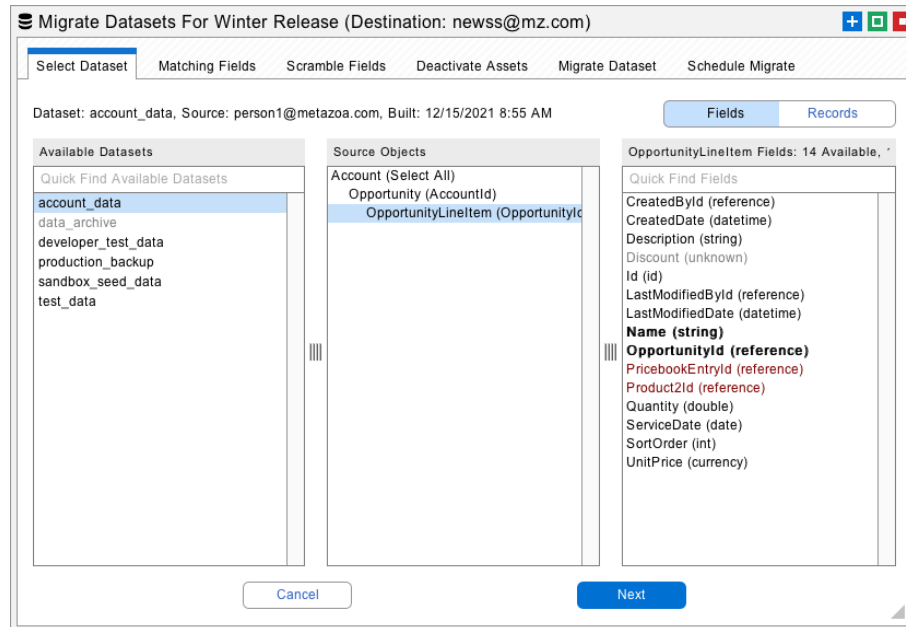
Here is a power user tip. You can right-click any of the objects in the middle list and export the data as an XML or CSV file. These files will be in the correct format for the Import Datasets dialog. For example, you could export a CSV file, edit the file as a spreadsheet, and then import your changes.
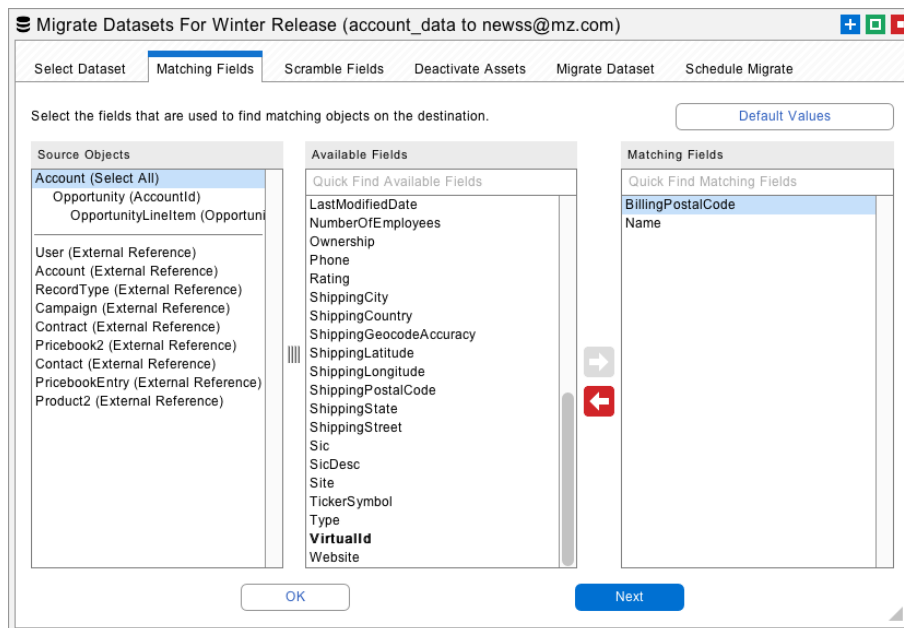
Matching Fields

Snapshot uses the loaded fields to identify matching records in the destination org during migration. If a source record matches a destination record, then the corresponding destination record is updated. If a source record does not match any destination record, then a new destination record is created.

Snapshot automatically selects common matching fields for you. However, you can also manually select matching fields. For example, if you want to match Accounts by Name and BillingCity, then be sure that both Name and BillingCity are loaded. The selected fields create a logical AND filter for matching destination records.

One powerful way to match destination objects is with External Ids. Other common matching fields include object names, email addresses, and usernames. These fields will automatically be available to select for matching. Some Salesforce sandboxes have the same Ids as production orgs. In that case you can simply use the Id field for matching destination objects.

Another way to match destination objects is with Virtual IDs. Use the Virtual Id to match hard to find objects that don't have a name. When these objects are created, Snapshot will remember the source ID and destination ID and always match them during future migrations.
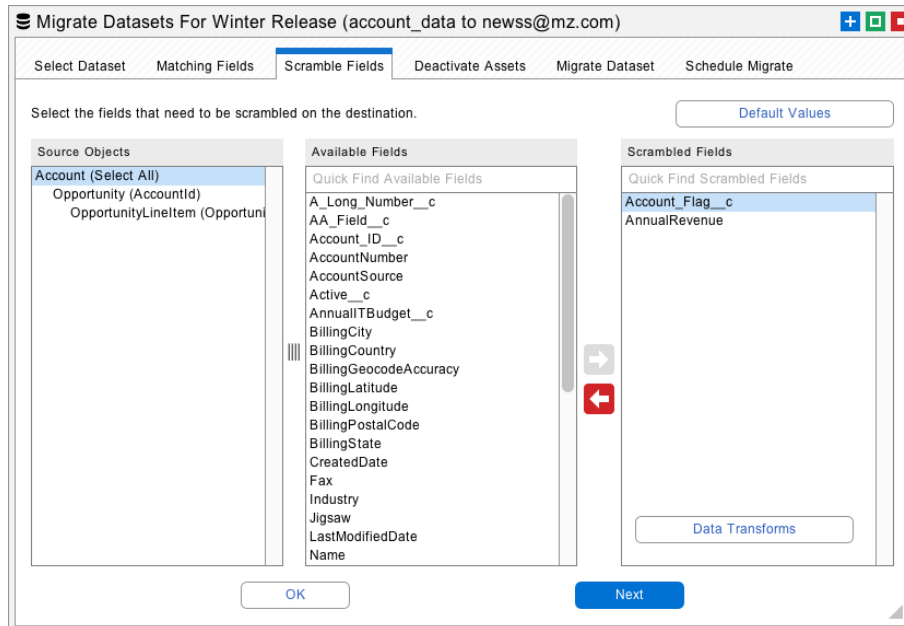
Underneath the dividing line you will see "external references" to other objects in the destination org. Snapshot will automatically connect external references to matching objects on the destination org when the dataset is migrated. You can manually specify additional fields needed for matching external references as well.

Scrambled Fields

Datasets are often used to move records into a Salesforce Sandbox or Developer Edition for testing or application development. In these situations, you may want to scramble data records that contain sensitive information. These fields might contain financial information, such as credit cards or bank accounts, or personal information, such as email addresses or Social Security numbers. The Scramble Fields tab provides an easy way to obscure fields on the destination org. Move the fields that you want to scramble over to the list at right.

Also notice the Data Transforms button at the lower right. Use this feature to transform field values on the fly as they are being written to the destination org. You can transform field values by prefix, suffix, or substring. You can also search and replace by multiple values. This is useful for remapping picklists.

Deactivate Assets

When a Salesforce record is inserted or updated, various Apex Triggers, Workflow Rules, and Validation Rules might be invoked. Apex Triggers perform custom actions before or after records are changed. Workflow Rules can also be invoked when records change, at which point they will perform automated actions. Validation Rules verify that the record data meets some kind of criteria before being inserted, updated, or deleted.

All of these automated behaviors can cause potentially undesirable effects during data migration. For example, thousands of emails might be sent out, or some records might not be updated. The Deactivate Assets tab provides an easy way to deactivate Apex Triggers, Workflow Rules, and Validation Rules in the destination org before data migration is attempted. After migration, the deactivated triggers and rules will be turned back on.

Migrate Datasets Button

The next tab has the main interface for migrating datasets to the destination org. First, make sure that the migration options are set correctly, these are discussed in more detail below. Then click the Migrate Datasets button to get started. All of the details of the migration will be written to the report at lower right. The last tab allows you to Migrate Datasets at a specific time in the future or as a recurring event.

Migration Options

There are various options for Data Migration:

- Upsert Records Only
- Delete Then Upsert
- Delete Records Only

The menu in the upper left gives you the option to delete the current dataset before migration, delete and then upsert, and upsert only. For production orgs we check twice that the administrator wants to delete the data. Be careful with this option.

- Stop After Error
- Continue After Error

If an error occurs, then Snapshot will either stop processing additional files with the Bulk API or continue. All errors are written to the log files. Common errors include too many duplicate records, email addresses in the wrong format, field data in the wrong format, etc.

- Don't Truncate Fields
- Allow Field Truncation

One common problem when moving data between different types of orgs is text strings that are too long for the destination field. Select this option to automatically truncate fields to the correct length or otherwise report an error.

- Log Migration Errors
- Log Errors and Success

All errors are written to the Log Files located in the "datasets" folder in the Snapshot file system. Optionally you can also log successful migrations. The log file contains the source and destination Ids as well as any error message.

- Serial Processing
- Parallel Processing

Parallel processing is faster but may cause database lock errors. Switch to serial processing if there are lots of lock errors reported.

The Upsert Batch Size field is used for specifying smaller batch sizes. There is a limit of 12 duplicate records per batch, so a smaller batch size may be helpful in avoiding this error.

Pro Tips

When you are building a dataset, you must select the parents and then the children in the object hierarchy. Try to only use parent objects and avoid using child objects if possible. Do not try to reconstruct the desired relationships in your data model when building the object hierarchy. All the relationships between the objects will be figured out by Monarch. Focus on simply getting the desired records into the dataset. Child objects should ONLY be used to limit selected records to the related parent. You do NOT need to reconstruct the relationships in your data model to connect everything properly.

If you build a complex object hierarchy that looks like your conceptual data model, then various problems can happen. First, child objects can end up in the dataset multiple times. There are situations where this might be needed, but when the dataset is migrated, each record will be migrated only once. Another consideration is that when a parent record fails to migrate because of some error that also prevents any child objects from being migrated, and their children, and so on.

When you are migrating a dataset, pay special attention to the matching fields that are selected. The matching fields must identify a unique record on the destination org. The first time a migration is run, matching destination record will be updated, and non-matching records will be created. If you run the migration again, you should only see updates, because every record on the destination will have a unique match. If you see more records being created that means you do not have reliable matching fields. This is why external IDs are highly recommended. They will create the record if needed and match forever after that. If for some reason you cannot use an External ID, then be absolutely certain that you are getting unique matching fields somehow.

Special Cases

Our Monarch product has been rewritten quite a few times over the last 15 years. Along the way we have added some special cases to handle difficult migration issues. Ironically, Custom Objects work great, and we handle every use case. The problems start when migrating Standard Objects that were developed back in the olden days. They often have special server-side rules and constraints that require special handling.

Polymorphic Relationships

Some lookup relationship fields are polymorphic. This means that a single relationship field can reference many different types of objects. For example, the Event object has a polymorphic relationship field called WhatID. This field can refer to dozens of different object types including Accounts, Contracts, and Opportunities.

Since many different types of objects can be referenced, there is no easy way to specify complex matching criteria for polymorphic relationship fields. By default, polymorphic relationship fields only reference the destination object by name. But some objects (like Accounts) can have duplicate names. Monarch handles this situation by first matching by name and then looking for External ID fields that match the source object ID.

Opportunity Pricebooks

Every Opportunity has some Opportunity Line Items, and each line item is associated with a Pricebook Entry. The Pricebook Entry is automatically generated to match each Product and Pricebook pair. However, the Opportunity also has a direct reference to a Pricebook. That Pricebook must be the same one that is indirectly referenced by the Opportunity Line Item via the Pricebook Entry.
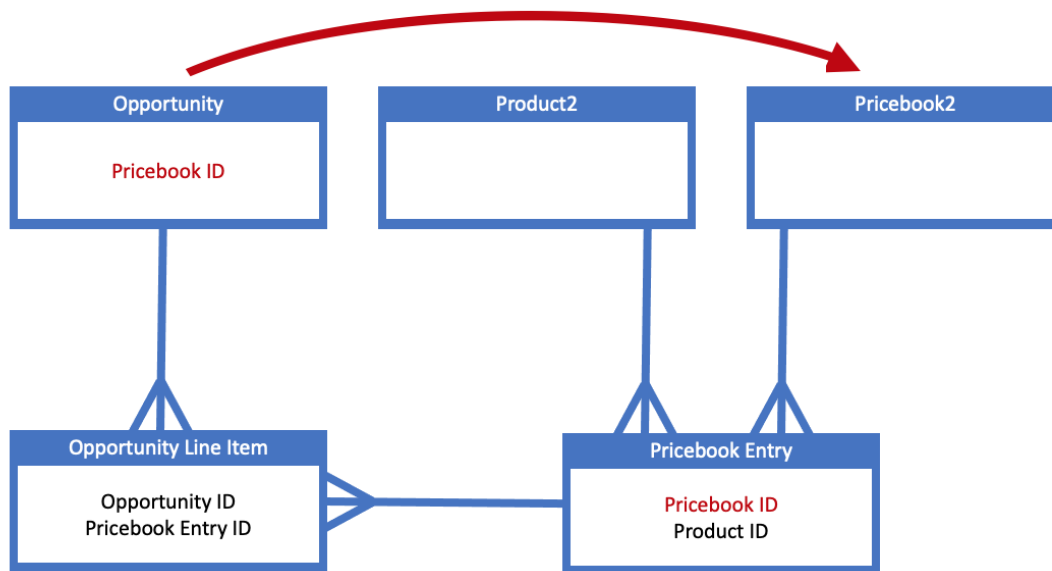
You can only create Opportunity Line Items that reference the same Pricebook that the Opportunity uses. This is tricky because the reference is indirect on both sides, so the correct Pricebook Entry must be selected. Monarch will do this automatically for you. Simply migrate these items in the hierarchy below:

```
Product2 (Select All)
Pricebook2 (Select All)
     PricebookEntry (Pricebook2Id)
```

Group Members

Here is another special case. Group Members are the junction object that connects a Group to another User or Group. They are not updateable, so Monarch will delete them and then create them as needed during migration. Be sure that the parent Group is migrated at the same time, the hierarchy should look like this:

```
Group (Select All)
     GroupMember (GroupId)
```

Custom Settings

There are two types of Custom Settings: List and Hierarchical. List custom settings must have a unique name, and the Setup Owner Id field must be equal to the Organization Id. Hierarchical settings must have unique Setup Owner Id that is equal to the Organization ID, a User ID, or a Profile ID. All of these constraints make Custom Settings hard to migrate. Monarch will handle the migration automagically, although there are some possible errors. For example, you cannot migrate a list Custom Setting to a hierarchical one, and vice versa.

Activities

Activities are made up of two objects - Tasks and Events. They can be migrated as children of the appropriate objects or as parent objects themselves. Snapshot is smart enough to figure out the relationships for standard objects like Account, Contacts, Opportunity etc. If you have a Custom Objects with activities, you are better off migrating the activities as children. Archived Tasks and Events are available when building a dataset. Filter by the IsArchived field if you do not want to include archived activities.

Email Messages

EmailMessages can be migrated as parent objects themselves. But you will want to drop 3 fields during the Loaded Fields step in Build Datasets - ActivityId, CreatedBy, and ValidatedFromAddress. That makes the user executing the migration the creator of all the records but the From and To fields will be correct.

Attachments and Content

Salesforce Classic stores files as Attachments. They can be migrated as children of related objects. For example, you might move Accounts as the parent with Attachments as the child. Salesforce Lightning stores files as Content Documents. First move all Content Versions to the destination. This migration will create the associated Content Documents automatically. There are some additional notes about moving Content Versions, below. Next, move the Content Document Link records as children of related objects. For example, you might move Accounts as the parent with Content Document Link as the child. This will connect each parent to the appropriate Content Document on the destination.

When moving Content Versions, we recommend that you filter on records where the IsMajorVersion flag is true and the ContentSize < 40,000,000. The minor versions cannot be moved or updated, and the size of any API transaction is limited to 50 MB. Also, do not load the FirstPublishLocationId and ContentModifiedDate fields. If you include these fields in the dataset existing Content Versions cannot be updated.

Account Contact Relations

When you turn on "Relate Contact to Multiple Accounts" under Account Settings, the AccountContactRelation object becomes available. This feature can reduce data duplication with Contacts that span Account boundaries. You must migrate Accounts and Contacts first, and then you can migrate AccountContactRelations as a child of Accounts or Contacts. Be aware that whenever a Contact that is related to an Account is created this automatically creates an AccountContactRelation.

Bulk API Limitations

The Bulk API won't move everything. A related problem is that some objects have Base64 fields that can be gigantic. For these special object Monarch switches to the SOAP Data API and handles the migration in a serial mode. This ensures that every type of object can be migrated. Here are the objects with Base64 fields:

- Attachment
- ContentNote
- ContentVersion
- Document
- EmailCapture
- EventLogFile
- MailmergeTemplate
- MobileApplicationDetail
- QuoteDocument
- Scontrol
- StaticResource

And here are the objects the Bulk API does not support:

- AcceptedEventRelation
- CaseStatus
- ContractStatus
- KnowledgeArticle
- KnowledgeArticleVersion
- KnowledgeArticleVersionHistory
- KnowledgeArticleViewStat
- KnowledgeArticleVoteStat
- LeadStatus
- OpportunityStage
- PartnerRole
- RecentlyViewed
- SolutionStatus
- TaskPriority
- UserRecordAccess
- ContentFolderItem
- DeclinedEventRelation
- EventWhoRelation
- TaskStatus
- TaskWhoRelation
- UndecidedEventRelation
- AccountUserTerritory2View
- DevelopmentPackageVersion
- InstalledPackageVersion

Conclusion

This whitepaper has discussed the best practices for data migration using Snapshot on the Salesforce platform. The Snapshot product from Metazoa provides a best-of-breed solution for continuous integration with a highly flexible toolset.

support@metazoa.com

1-833-METAZOA
1-833-638-2962

https://www.metazoa.com

Twitter: @metazoa4sf

Facebook: https://www.facebook.com/metazoa4sf

LinkedIn: https://www.linkedin.com/company/18493594/