Snapshot Best Practices:
Org Merges

Snapshot has powerful metadata deployment and data migration tools that enable administrators to merge complex Salesforce orgs. The need to perform an org merge is usually a special situation and not something that is done on a regular basis. These projects are often mission critical and may involve contractual commitments or other external deadlines. Because of this, Metazoa also offers professional services to perform this work on behalf of customers. Either way, an org merge is a complex undertaking that requires careful planning, business analysis, and project management to ensure successful completion.

This whitepaper discusses some of the best practices when using the Snapshot toolset to perform an org merge. We will cover business analysis, project management, synchronizing settings, removing technical debt, metadata deployments, data migration, and final steps. First off, let’s discuss the definition of an org merge and the benefits of performing this transformation.

Org Merges

An org merge is when two or more source Salesforce orgs are merged into a single destination org. This situation usually happens when a company buys another company and needs to consolidate the two Salesforce orgs. An org merge might also happen when two corporate departments are combined.

An org merge requires careful planning and business analysis to decide how the new destination org should function. What is the purpose of the merge? What are the business requirements for the destination org? Here are some common business reasons for org merges:

- Centralized view of reports and dashboards
- Upsell and cross sell visibility and productivity
- Standardized processes for sales and service
- Better team collaboration and content
- Reduced cost of org administration

Org merges usually require more planning and business analysis than org clones and splits. If you are interested in the best practices for org clones and splits, that topic is discussed in another whitepaper available on the Metazoa website.

Strategy

The administrative team needs to designate one of the orgs to be merged as the destination org. This destination org will survive the merge process and ultimately become the new home for all the users, data, and customizations. The destination org that is selected might have the most users, the best data, or the most desirable features. The other orgs will serve as the source of users, data, and customizations for the merge.

One common scenario is that the destination org belongs to a larger company and the source org belongs to a smaller company that is being acquired. In this case, the users and data from the smaller entity are basically just incorporated into the larger destination org. Many of the customizations from the source org are lost in the process.

Another scenario is that two orgs of equal importance are being merged. This could happen if a large company chose to consolidate their Sales and Marketing divisions. The goal is to have all business processes functioning as they are today, but in one single consolidated instance. In this case, Profiles, Record Types, Picklists, and other important customizations need to be merged.

Project Management

A high degree of coordination is required between the team performing the org merge and the existing administrative staff. The simplest scenario is when the migration team has complete control of both the source and destination org. This is not always possible, especially for the destination org. One strategy is to use a full copy sandbox for the destination org. At the end of the project this sandbox can be deployed to production. Another trick is to merge all the customizations into a brand-new destination org. A side benefit of this is that technical debt can be eliminated when the data and metadata are copied over.

The source org might also be in active use. This can complicate the situation, especially if metadata and data assets in the source org are being modified during the merge process. The best practice here is to agree on outage windows where the migration team can have exclusive access to the required orgs. This usually requires a two-phase migration: phase one moves all the metadata and data, and phase two checks for new differences before final delivery.

Any org merge will need an administrative champion who can answer questions about the corporate requirements for a successful transition. For example, what users will be moving to the destination org? What data records should be migrated? Which business processes do we need to keep, and which ones can be deprecated? The administrative champion should also coordinate outage windows, org validation, user acceptance testing, user training, and final cutover to the new destination org.

Process Review

Org merges often require that business processes from multiple orgs are combined. Some of these business processes might be conflicting, others might be redundant, and a few might be unnecessary. A useful exercise is to identify the custom objects and metadata assets that are associated with each business process. Don’t forget about managed packages. Each managed package is a great big business process that must be adopted or deprecated. Next, prioritize each process to narrow the scope of the metadata deployment and data migration. Here are some process categories we recommend.

- Process is not required
- Process is already on destination
- Process conflicts with destination
- Process must be migrated to destination
- Process must be merged with destination

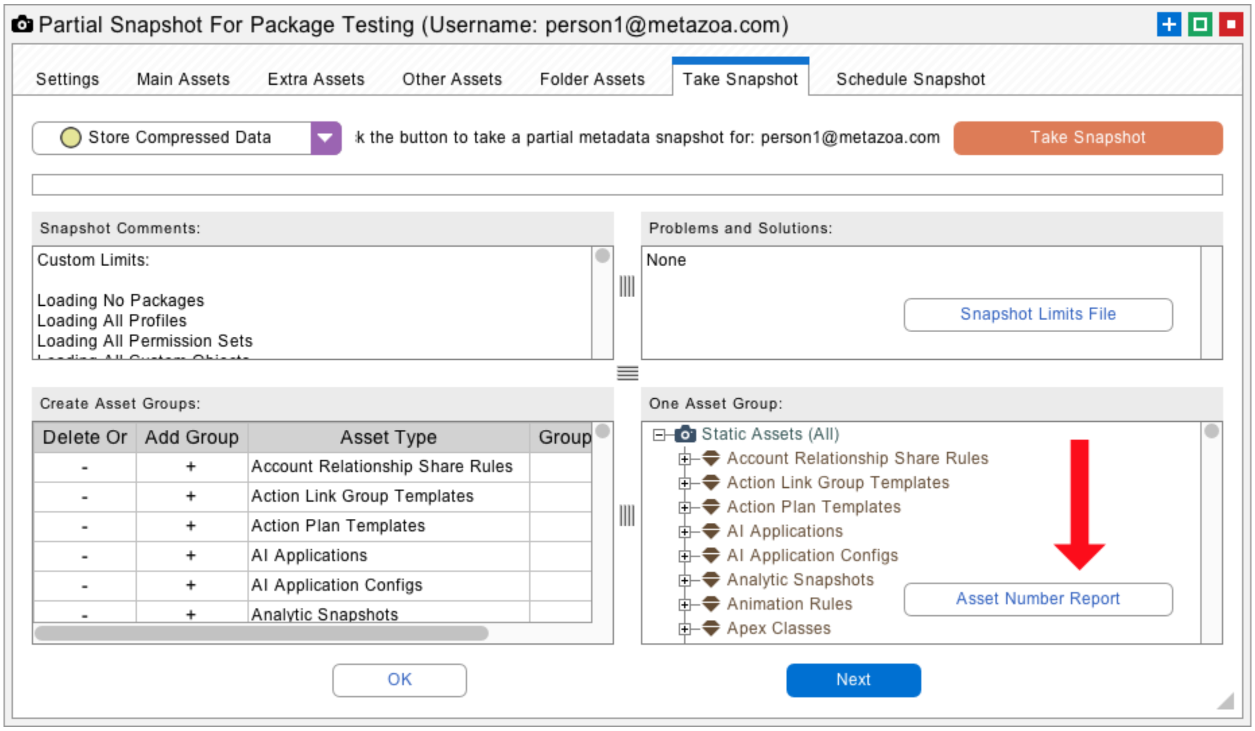
Inevitable Differences

If there are conflicting business processes that cannot be merged, then difficult decisions must be made. Which business process should be followed? There are also likely going to be differences introduced when different types of Salesforce orgs are being merged. For example, older Salesforce orgs can start up in Classic mode, but all new orgs will automatically start up with the Lightning user experience. SControls can no longer be created, and they will need to be replaced with Visualforce pages. Workflows will also be deprecated, and they should probably be replaced with Flows. Portals can no longer be migrated, so they will need to be replaced with Communities. Because of these differences, user training and adoption are an important part of any org merge project. The transition is probably going to be the most difficult for users moving from the source org to the new destination.

Sanity Check

Let’s look at the big picture. First, the migration team will need to take snapshots of the metadata from the source org and then deploy these customizations into the destination org. Then they will need to create datasets of various records in the source org and migrate this information to the destination as well. Before work begins, the migration team needs to conduct a sanity check of the required team size, processing power and hard disk storage required to get the job done.

The number of metadata assets in the source org can be estimated with the Partial Snapshot interface. Enter your org credentials and navigate to the Take Snapshot tab, and you will see the Asset Number Report button at below right. This report will count all the metadata assets in the source org, including reports, dashboards, and packages.



Partial Snapshot For Package Testing (Username: person1@metazoa.com)

○ Store Compressed Data

k the button to take a partial metadata snapshot for: person1@metazoa.com

Snapshot Comments:

Custom Limits:

Loading No Packages
Loading All Profiles
Loading All Permission Sets

Problems and Solutions:

None

Create Asset Groups:

Delete Or	Add Group	Asset Type	Group
-	+	Account Relationship Share Rules	
-	+	Action Link Group Templates	
-	+	Action Plan Templates	
-	+	AI Applications	
-	+	AI Application Configs	
-	+	Analytic Snapshots	

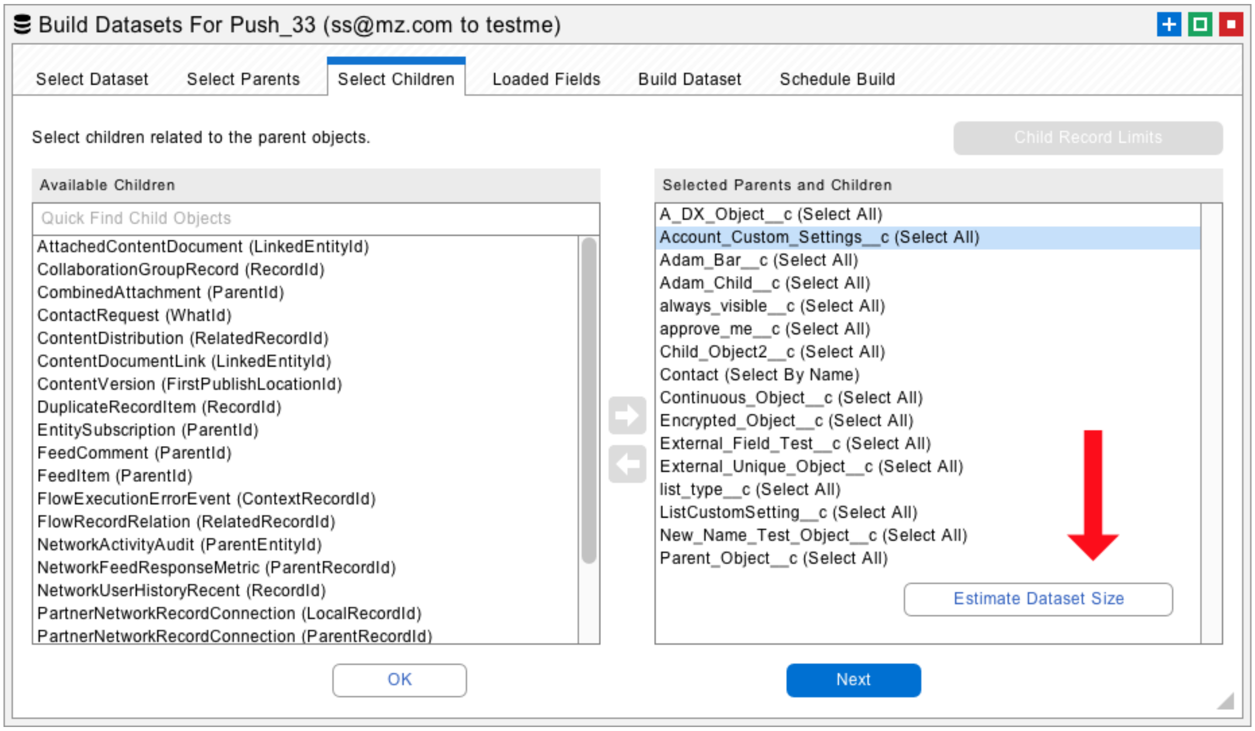
One Asset Group:

- Static Assets (All)
  - Account Relationship Share Rules
  - Action Link Group Templates
  - Action Plan Templates
  - AI Applications
  - AI Application Configs
  - Analytic Snapshots
  - Animation Rules
  - Apex Classes

There is a limit of 10,000 metadata assets in a single download. If the source org falls inside this limit, then you can use the Full Snapshot interface to capture all the metadata. If the source org has a gigantic number of assets, then the Partial Snapshot interface can be used to take multiple snapshots and stitch them together into a complete picture. You can also grab individual pieces of the source metadata and deploy each one separately. The controls in the lower left of the Partial Snapshot interface define the groups of metadata to download, see the help page on this interface for more information.

Next, you need to get an estimate of the number of data records that must be transferred. The Storage Usage in Salesforce Setup can give you some of this information, but if you are not migrating all objects and fields, then what you really need are statistics on the records actually being moved. An easy way to get this information is from the Build Datasets interface.

Select all the parent and child objects that must be migrated and navigate to the Select Children tab. The Estimate Dataset Size button at the bottom right will create a report that gives you an accurate count of the number of records that must be moved and an estimate of the total size of the dataset. You probably don't want to build a single dataset with all the records that need to be moved. You can move the data in sections. But this report will give you a bird's eye view of the size of the job.



Select children related to the parent objects.

Available Children:

AttachedContentDocument (LinkedEntityId)
CollaborationGroupRecord (RecordId)
CombinedAttachment (ParentId)
ContactRequest (WhatId)
ContentDistribution (RelatedRecordId)
ContentDocumentLink (LinkedEntityId)
ContentVersion (FirstPublishLocationId)
DuplicateRecordItem (RecordId)
EntitySubscription (ParentId)
FeedComment (ParentId)
FeedItem (ParentId)
FlowExecutionErrorEvent (ContextRecordId)
FlowRecordRelation (RelatedRecordId)
NetworkActivityAudit (ParentEntityId)
NetworkFeedResponseMetric (ParentRecordId)
NetworkUserHistoryRecent (RecordId)
PartnerNetworkRecordConnection (LocalRecordId)
PartnerNetworkRecordConnection (ParentRecordId)

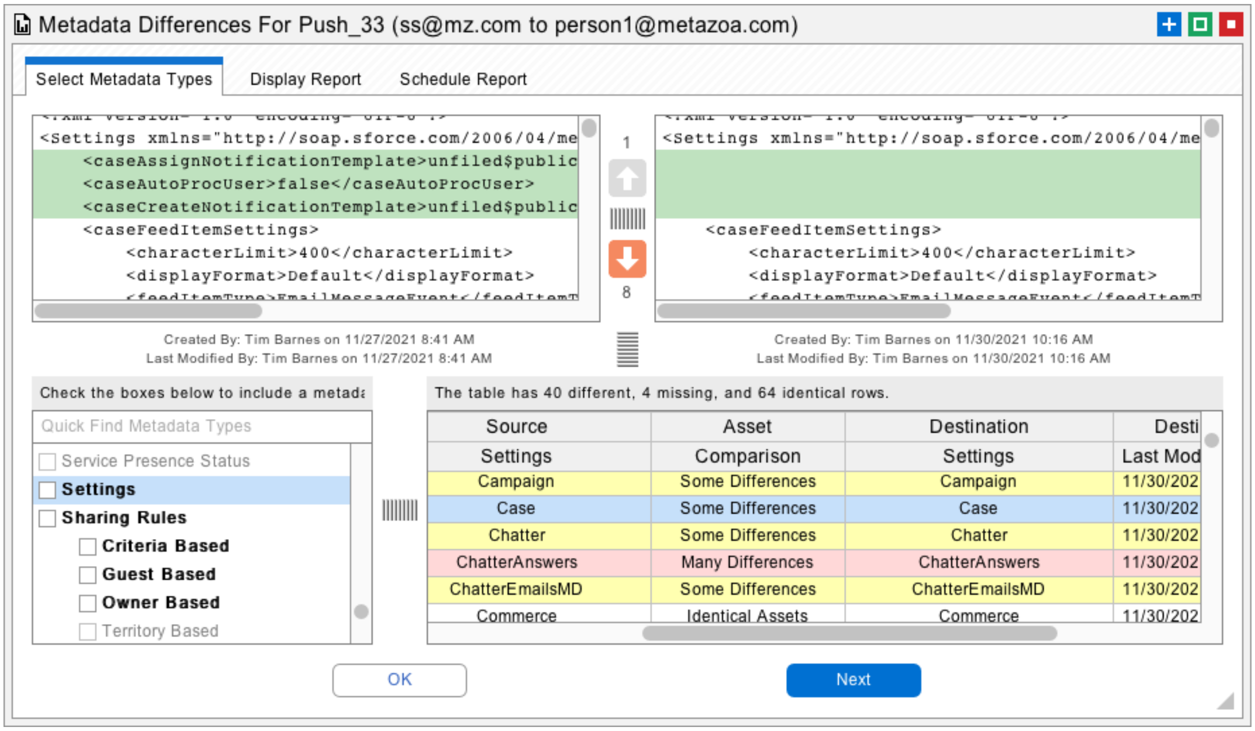
Selected Parents and Children:

A_DX_Object__c (Select All)
Account_Custom_Settings__c (Select All)
Adam_Bar__c (Select All)
Adam_Child__c (Select All)
always_visible__c (Select All)
approve_me__c (Select All)
Child_Object2__c (Select All)
Contact (Select By Name)
Continuous_Object__c (Select All)
Encrypted_Object__c (Select All)
External_Field_Test__c (Select All)
External_Unique_Object__c (Select All)
list_type__c (Select All)
ListCustomSetting__c (Select All)
New_Name_Test_Object__c (Select All)
Parent_Object__c (Select All)

Now it's time for the sanity check. Orgs with less than 10,000 metadata assets and 1 million records can usually be migrated by a small team. They will need modern laptop computers, and a fast Internet connection. Since millions of files are involved, a fast solid-state hard drive with many GB of storage is helpful. But there are also situations where org merges have millions of metadata assets and hundreds of millions of data records. In this situation, we recommend using Windows or Macintosh cloud servers. They can be outfitted with huge amounts of processing power, hard disk storage, and Internet connectivity. They can be controlled remotely from a laptop functioning as a terminal. You can run multiple instances of the Metazoa Player on each cloud server. Obviously, the team members will have to coordinate their efforts and migrate the metadata and data records. We discuss some of the best practices to organize massive migrations into logical sections, below.

Synchronize Environments

Before trying to deploy metadata or migrate data, you should do everything possible to make sure that the source and destination org environments are as similar as possible. If the environment is the same on both orgs, many metadata deployments and data migrations that would otherwise be impossible will work just fine. You probably want to make these environmental changes to the source org. If the source org is in active use, then perhaps the changes can be made to a full copy sandbox of the source org. Use the sandbox for metadata deployments and data migrations to the destination org.



Metadata Differences For Push_33 (ss@mz.com to person1@metazoa.com)

```
<Settings xmlns="http://soap.sforce.com/2006/04/me
    <caseAssignNotificationTemplate>unfiled$public
    <caseAutoProcUser>false</caseAutoProcUser>
    <caseCreateNotificationTemplate>unfiled$public
    <caseFeedItemSettings>
        <characterLimit>400</characterLimit>
        <displayFormat>Default</displayFormat>
```

Created By: Tim Barnes on 11/27/2021 8:41 AM
Last Modified By: Tim Barnes on 11/27/2021 8:41 AM

```
<Settings xmlns="http://soap.sforce.com/2006/04/me



    <caseFeedItemSettings>
        <characterLimit>400</characterLimit>
        <displayFormat>Default</displayFormat>
```

Created By: Tim Barnes on 11/30/2021 10:16 AM
Last Modified By: Tim Barnes on 11/30/2021 10:16 AM

Check the boxes below to include a metad

Quick Find Metadata Types

- □ Service Presence Status
- □ **Settings**
- □ **Sharing Rules**
  - □ **Criteria Based**
  - □ **Guest Based**
  - □ **Owner Based**
  - □ Territory Based

The table has 40 different, 4 missing, and 64 identical rows.

Source	Asset	Destination	Desti
Settings	Comparison	Settings	Last Mod
Campaign	Some Differences	Campaign	11/30/202
Case	Some Differences	Case	11/30/202
Chatter	Some Differences	Chatter	11/30/202
ChatterAnswers	Many Differences	ChatterAnswers	11/30/202
ChatterEmailsMD	Some Differences	ChatterEmailsMD	11/30/202
Commerce	Identical Assets	Commerce	11/30/202

First, take a snapshot of the Settings metadata type on the source and destination orgs. The Settings control many aspects of the org environment. For example, you can choose to automate Flows on deployment, or declare the org to be multi-currency. Then use the Metadata Differences Report to drill down into the differences. You can deploy the Settings metadata from the destination org into the source org. There may be deployment errors, which will point out other differences between the orgs. You might need to purchase some additional Salesforce product, or you might need to make manual changes with the Setup Menu. We estimate that 80% of the environmental differences between orgs can be rectified by deploying the metadata Settings from the destination into the source org.

All of the **custom** Objects, Tabs, Applications, and other assets that were created on the source org can be transferred to the destination org. But the **standard** Objects, Tabs, Applications, and User Permissions that are on the source org cannot be changed. Look at any differences between these asset types. The Compare Profiles interface is an easy place to do this. This information can be used to discover any remaining environmental differences. Also be aware that Snapshots metadata deployment tools can Remove Bad References to **standard** objects during the deployment process if necessary.

Technical Debt

It doesn't make any sense to transfer a bunch of technical debt from the source org into the destination org. A complete technical debt analysis is a great first step to discover the assets that should and should not be migrated. Snapshot has amazing reports that help with technical debt analysis, including the ability to identify and merge similar profiles, remove connections to inactive users, optimize license expense, delete forgotten metadata, and retire unused fields and picklists. Depending on the situation, you might want to remove technical debt from the source org or use this knowledge to control what is migrated to the destination.

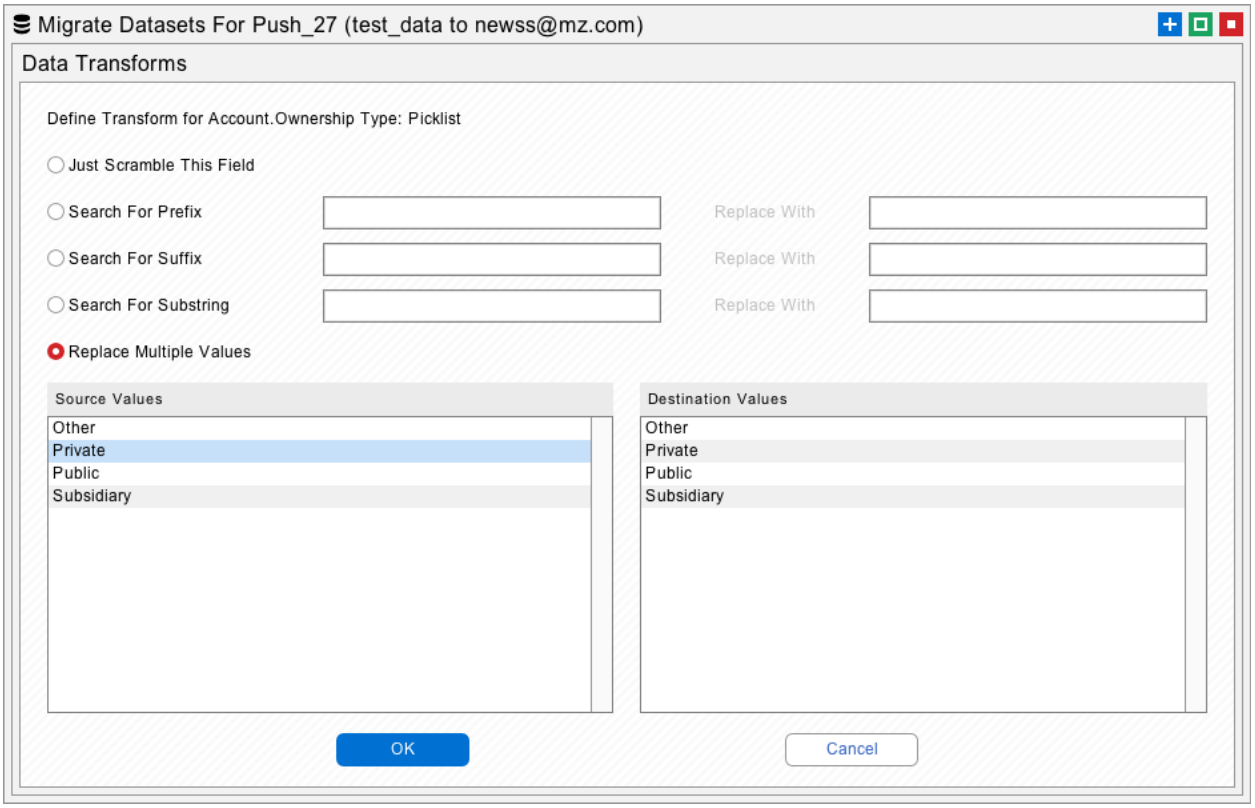
When it comes to data migration, make sure that Contacts, Leads, and Accounts in the source and destination org have been deduped if possible. This makes polymorphic references in Tasks and Events more likely to match by name. Converted Leads often do not need to be migrated, because that information has already been captured by Opportunities and Contacts. Attachments are being deprecated in favor of ContentDocuments, so you might want to convert them before migration to the destination org.

Another huge issue is how inactive users are handled. Snapshot has powerful tools to delete all the connections to inactive users in the source org. Once this is done, only the active users should be migrated to the destination. Sometimes administrators don't want to clean up the inactive users if the source org is in active use. That might cause unnecessary disruptions. The best practice is to clean up the inactive users in a sandbox org and deploy the metadata from there. Both the metadata deployment tools and data migration tools in Snapshot can transform usernames as necessary during migration.

Transformation Options

The metadata deployment tools have some great options that can help with org merges. The “Apply Data Transforms” button will remap any file name, folder name, package name, or element value when the metadata is deployed. This is especially useful for remapping usernames between the source and destination org. The “Remove Bad References” button scans Profiles, Permission Sets, Page Layouts, Custom Objects, and Reports. Any references to assets that are not available in the destination org or included in the current Job List are automatically removed. This is especially useful for migrating **standard** assets and user permissions that cannot be changed.

The data migration tools also have options to translate datasets as they are copied to the destination org. In the Migrate Datasets interface, navigate to the Scramble Fields tab. Select a field at right, and click the Data Transforms button at the bottom of the list. You can change field values by prefix, suffix, substring, and value. Also, in the Manage Datasets interface, the Remap button brings up an interface to remap all of the field names in the dataset. This is extremely useful when merging custom objects and fields with different names.


Migrate Datasets For Push_27 (test_data to newss@mz.com)

Data Transforms

Define Transform for Account.Ownership Type: Picklist

○ Just Scramble This Field

○ Search For Prefix Replace With

○ Search For Suffix Replace With

○ Search For Substring Replace With

● Replace Multiple Values

Source Values	Destination Values
Other	Other
Private	Private
Public	Public
Subsidiary	Subsidiary

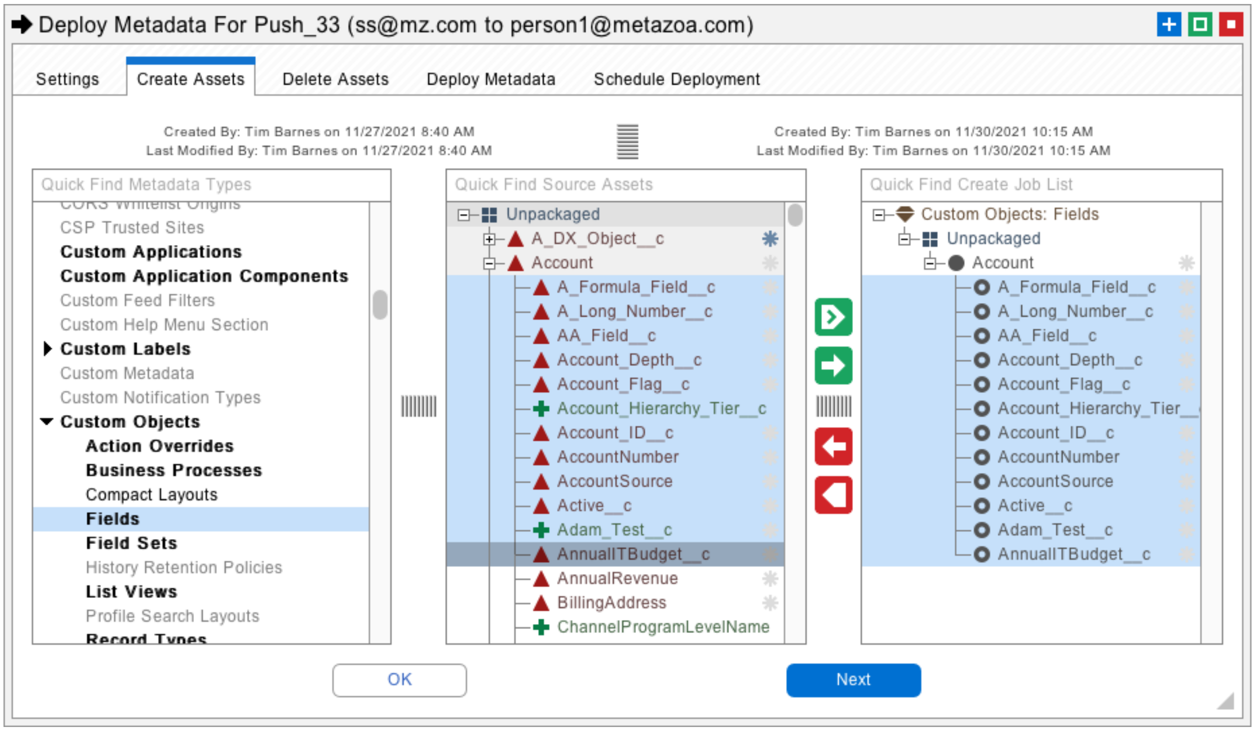
If the number of transformations becomes extreme or involves XML schema, you can always save assets from the source org in a Salesforce DX Developer Project. This allows editing of the source files by hand. Almost any transformation can be designed and then deployed to the destination. This approach comes in handy for metadata assets like Global Value Sets that require the addition and deletion of elements. Using Developer Projects is an alternative to the sandbox approach for the source org.

Metadata Deployment

There are over 250 different metadata types in Salesforce that define most of the customizations in an org. There are also 1200 different kinds of dependencies between these metadata types. Each dependency is a reference that connects two assets. When an asset is migrated, there can be validation problems or missing dependencies. Snapshot has an excellent Impact Analysis Report for understanding dependencies, and our deployment engine has features that simplify the deployment process. But deploying all the assets from one org into another org can be difficult because of complex circular dependencies and environmental differences.

Each source asset that is migrated will need to find all dependencies on the destination org or in the current job list. This is a crucial issue for org merges. For example, users depend on named Profiles. Is the plan to remap source users to new destination Profiles? Or is the plan to migrate the source profiles over and then match the users? If a company is buying a smaller company, then maybe the users will simply be remapped to existing Profiles. If Sales and Marketing are being consolidated, then both users and Profiles might need to be migrated to the destination org. This issue will affect metadata types like Groups and Roles as well. Can the source users be remapped to destination Groups? Or are new Groups required? What about the Role hierarchy?

When merging Custom Objects, a major sticking point will be Picklist values. Global Value Sets and Standard Value Sets should be deployed first. What picklists and Record Types are needed on the destination org? You will need to remap the source picklist values to the new destination, especially for Restricted Picklists. Custom Fields should be migrated next, after which data can be migrated.



Deploy Metadata For Push_33 (ss@mz.com to person1@metazoa.com)

Created By: Tim Barnes on 11/27/2021 8:40 AM
Last Modified By: Tim Barnes on 11/27/2021 8:40 AM

Created By: Tim Barnes on 11/30/2021 10:15 AM
Last Modified By: Tim Barnes on 11/30/2021 10:15 AM

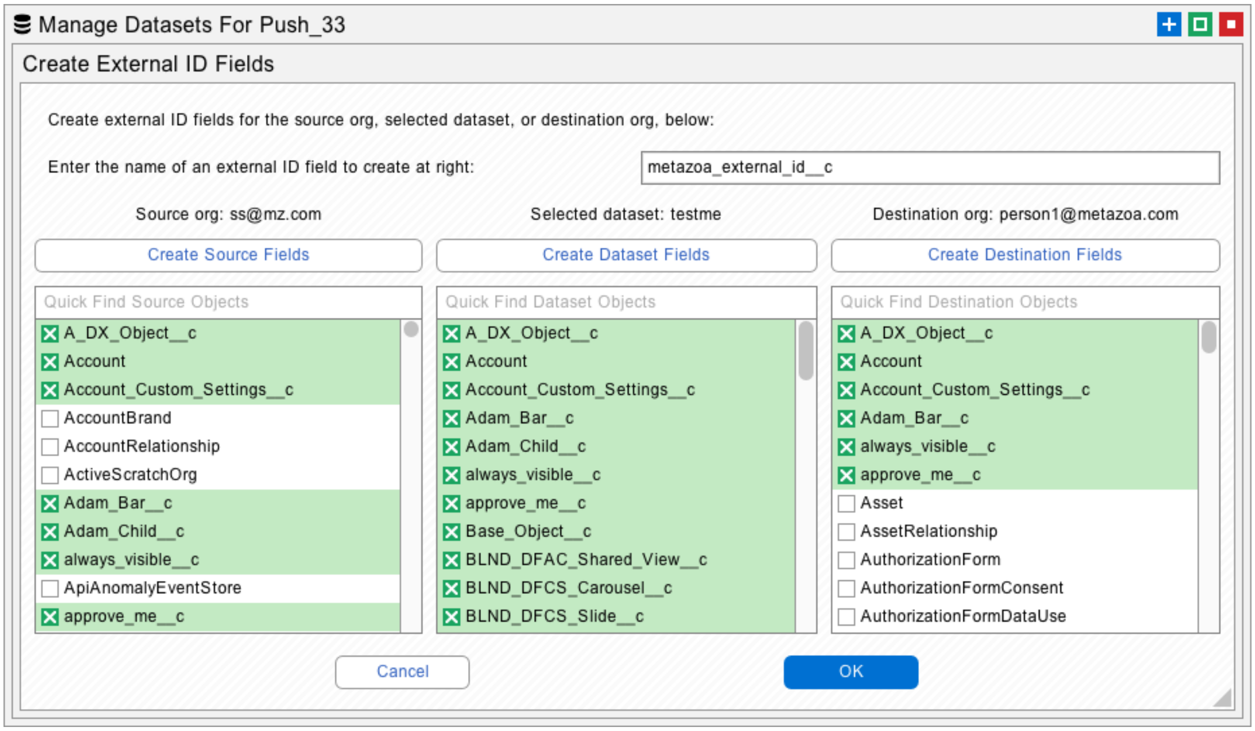
The difference between metadata and data is rather blurry. Many important junction objects are actually saved as data, not metadata. Examples of this include Group Members, Campaign Members, Permission Set Assignments, and Account, Opportunity, and Case Team Members. Once your metadata has been deployed, it’s time to focus on data migration.

Data Migration

The data migration process has some similarities to metadata deployment. There are likely to be multiple migrations, and various errors along the way that need to be addressed. Because of this, the best practice is to use external IDs to match destination data records with source records. Here is how this works. The external ID is a custom field on the source object that holds a unique external value. Often this value is just the source record ID itself. When these records are migrated to the destination, each destination record is permanently tagged with this value. Now when you match the records by external ID, the same record will be matched every time, even after repeated migrations.

Snapshot has introduced a killer feature that vastly simplifies the process of creating external IDs. The Manage Dataset interface has a new option to 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.

But 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.


Create Source Fields
Create Dataset Fields
Create Destination Fields
Quick Find Source Objects
Quick Find Dataset Objects
Quick Find Destination Objects
Cancel
OK

Manage Datasets For Push_33

Create External ID Fields

Create external ID fields for the source org, selected dataset, or destination org, below:

Enter the name of an external ID field to create at right: metazoa_external_id__c

Source org: ss@mz.com

☒ A_DX_Object__c
☒ Account
☒ Account_Custom_Settings__c
□ AccountBrand
□ AccountRelationship
□ ActiveScratchOrg
☒ Adam_Bar__c
☒ Adam_Child__c
☒ always_visible__c
□ ApiAnomalyEventStore
☒ approve_me__c

Selected dataset: testme

☒ A_DX_Object__c
☒ Account
☒ Account_Custom_Settings__c
☒ Adam_Bar__c
☒ Adam_Child__c
☒ always_visible__c
☒ approve_me__c
☒ Base_Object__c
☒ BLND_DFAC_Shared_View__c
☒ BLND_DFCS_Carousel__c
☒ BLND_DFCS_Slide__c

Destination org: person1@metazoa.com

☒ A_DX_Object__c
☒ Account
☒ Account_Custom_Settings__c
☒ Adam_Bar__c
☒ always_visible__c
☒ approve_me__c
□ Asset
□ AssetRelationship
□ AuthorizationForm
□ AuthorizationFormConsent
□ AuthorizationFormDataUse

Snapshot has very powerful tools to select source records for migration. You can select parent objects with a filter interface, match them by name, or manually enter a SOQL filter. Then you can select any number of child objects that are related to the parent. This user interface makes it easy to build up datasets that move the desired records.

However, when you get into the realm of moving millions of records, there are other strategies that might work better. For example, if you tag Accounts, Contacts, and Campaigns with an external ID then you could move them one at a time. First all Accounts, then all Contacts, then all Campaigns. Dividing the job up like this reduces the total number of objects and can be conducted on multiple machines.

Next, you could move all CampaignMembers. Each CampaignMember must have the correct reference to an Account, Contact, and Campaign before it can be created. The use of external IDs guarantees that each CampaignMember will be connected correctly to the parent object. The only trick here is moving the objects in the right order with the junction objects like CampaignMember in last place.

Final Steps

The real action starts when the new destination org is tested for validation. The testing might be conducted by multiple departments or different administrators who are familiar with the source org. If there are substantial changes in the destination org, say because new business processes have been introduced, then there will be user acceptance testing as well. Ultimately new users must be trained to adopt the destination org, just like any other Salesforce onboarding experience.

Conclusion

The Snapshot toolset is purpose built to solve difficult problems like org merges. The elimination of technical debt is a bonus along the way. The other option is to hire a large team of developers or consultants to do the work by hand. This approach is guaranteed to be slow, expensive, and error prone. If you need help with a mission critical project like this, please reach out and let us know how to provide support. We offer consulting on a project basis, or you can have our Professional Services Team handle everything for you.

Bill Appleton
CTO Metazoa
bill@metazoa.com
@bill_appleton