The Salesforce Migration Playbook

By Capstorm

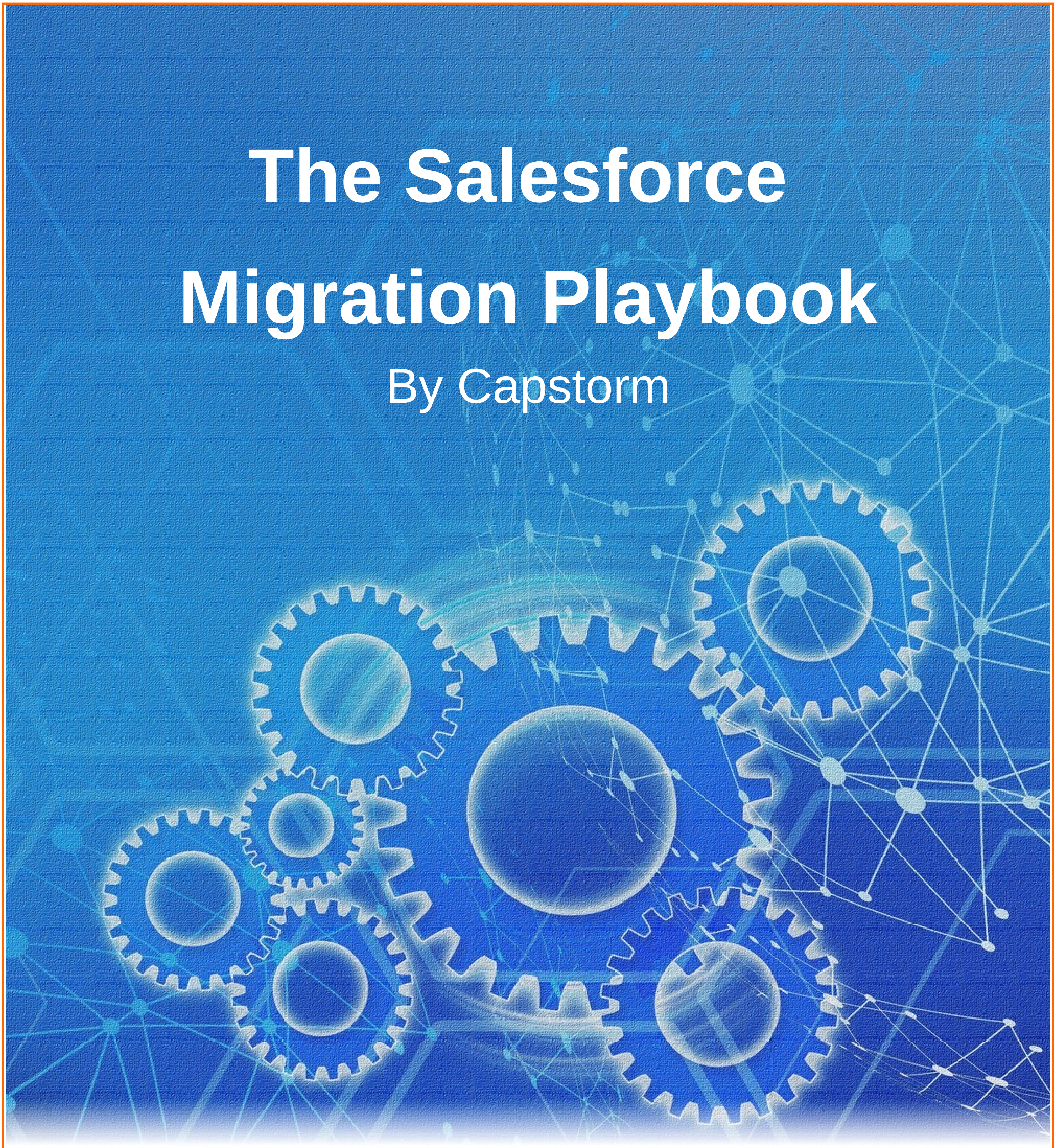


Table of Contents

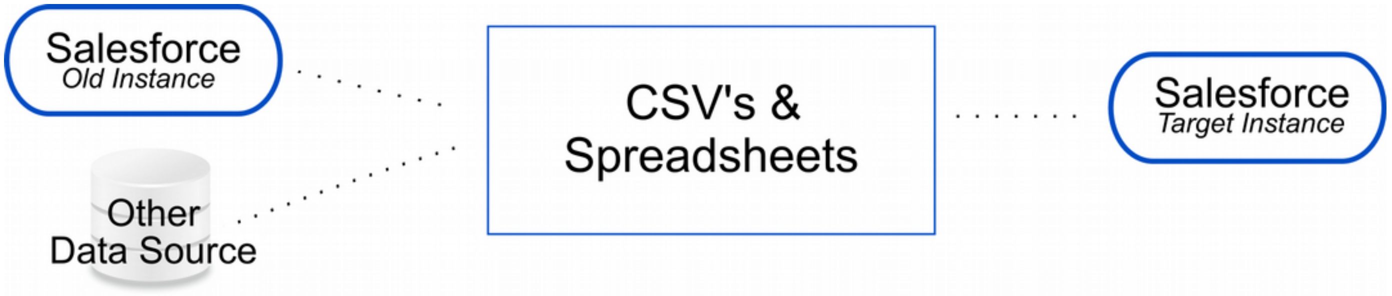
CAPSTORM

Migrating Data The Slow Way

The accepted “standard” way to perform a Salesforce migration involves a handful of tools that are intended to simplify the difficult tasks of exporting and importing records to Salesforce. For example, Salesforce provides built-in data export and import tools, however, the native tools come with limitations. The content and format of the export is not flexible (CSVs) and the timing of exports is severely limited. In addition, data import is a complex, multi-stage process. Migrating data into Salesforce with Salesforce’s built-in tools presents a number of challenges:

- Data import is cumbersome and relies on CSV data maintained in complicated spreadsheets.
- Record relationships are difficult to import and require multiple steps with manual data patching. The sheer number of steps increase the likelihood of human error stalling the project.
- Native data export can take a lengthy amount of time and can only be performed once a week (or even once a month!).
- The Salesforce data export is incomplete. For example: Attachments, Files, and Knowledge Articles are not included.

Slow migrations generally follow a process like this:

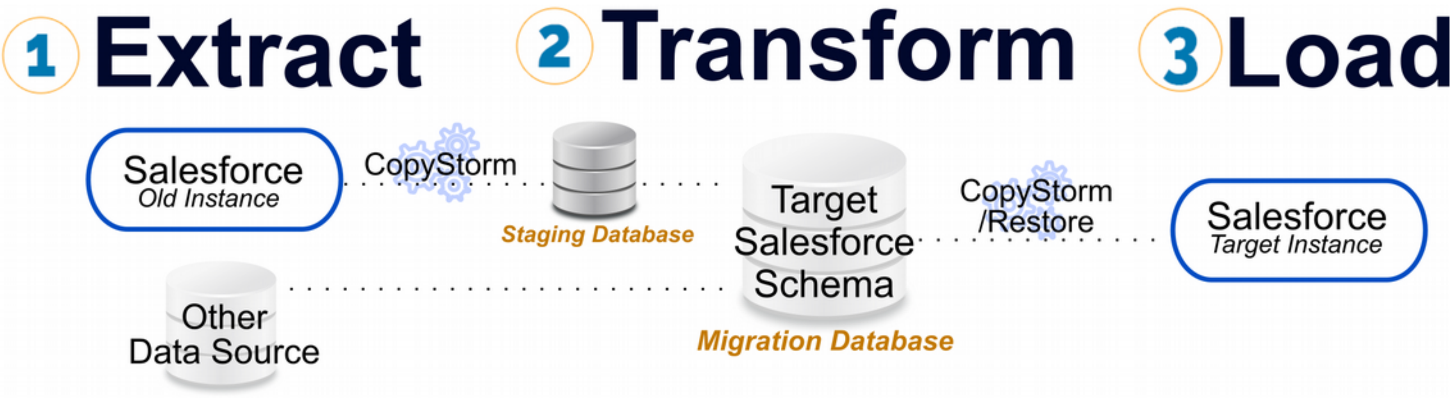


Migrating Data The Fast Way

There is an easier and faster way to migrate data into Salesforce! Migrations can be accomplished using third party applications specifically designed to move connected hierarchies of Salesforce data. Data is extracted to a local database, transformed into the format required by the Migration database, and loaded into the target Salesforce. Follow these steps to migrate data to Salesforce from another Salesforce instance or from a wholly independent third party data source:

1. **Extract** the **data** being migrated – from either a source Salesforce or any other data repository.
   - CopyStorm downloads and incrementally updates a local database containing the source Salesforce data and schema.
2. **Extract** the **schema** of the target Salesforce.
   - CopyStorm extracts the target Salesforce schema into a local relational database.
3. **Transform** data into the target Salesforce schema.
   - Standard database migration tools move extracted data into the target Salesforce schema.
4. **Load** data into the target Salesforce.
   - CopyStorm/Restore loads connected data and metadata into the target Salesforce.

An easier, faster method for Salesforce migration:



High-Level Migration Overview

A number of high-level concepts should be discussed prior to a more detailed discussion of the ETL migration process. Migrating data to Salesforce is a daunting task that is complicated by both the limitations of Salesforce's API and a need for solutions that allow Salesforce users to continue to using Salesforce mid-migration. Every data migration also needs to take into account the instability that comes from uploading data to the Salesforce via an Internet connection. The process within this document addresses these issues, allowing you to migrate connected data to Salesforce from anywhere:

1. From Salesforce into another Salesforce instance.
2. From a third party data source into Salesforce.

Capstorm’s recommended migration process is modeled after the well-tested, industry standard Extract-Transform-Load (ETL) methodology. The Capstorm ETL process allows Salesforce integrations to be performed using Salesforce-specific data download and upload applications - turning Salesforce data migration projects into much simpler relational database migrations. To migrate data into a live Salesforce instance using the ETL process, every migration component must be able to provide these two critical features:

1. Each piece of the migration process must be incremental.
    - Both the source environment (whether Salesforce or another third party) and the target Salesforce environment are live systems – salespeople won’t stop adding data to make room for a time-consuming migration! Two types of data import are crucial:
        - A mass import of records and their corresponding relationships.
        - Smaller imports of added or modified records – associated with previously updated data.
    - All communication with Salesforce is performed via an Internet connection – incremental loading minimizes the impact of any network instability or downtime encountered during the download or upload.
2. Each process component must be automated.
    - Automation allows for sandbox testing, which is crucial to avoid data corruption in production.
    - Non-automated solutions, such as manually updating CSVs or other data files, are cumbersome processes and too error-prone to provide reliable and consistent results. Nobody should have to do manual record ID patching!

Most Salesforce import and export tools do not provide these key features and are limited to providing CSV formatted exports every week or month. This is not frequent or flexible enough to drive an iterative data migration. The process described in this document provides these critical features by using Capstorm’s applications in order to export source Salesforce data to a local database and import data from a local database into a target Salesforce instance.

Incremental Salesforce Data Extract and Load

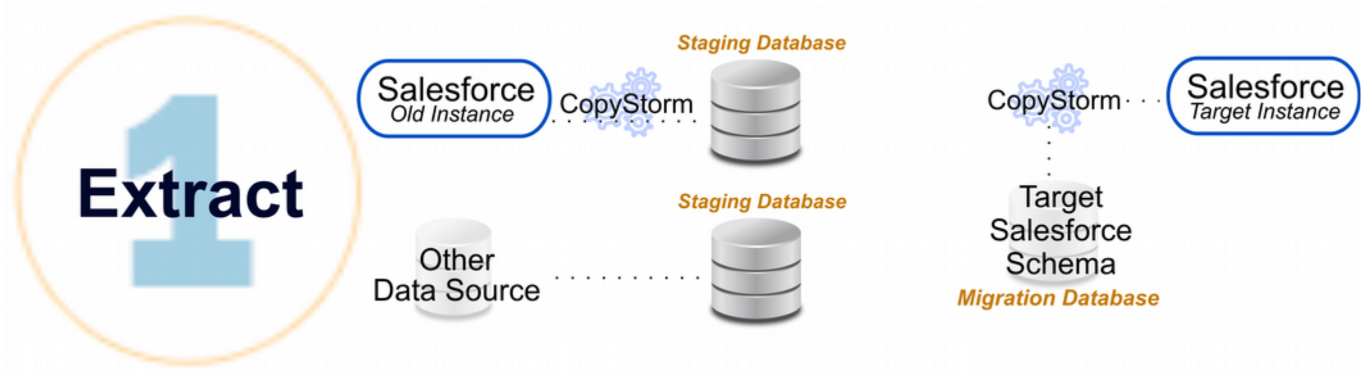
Incremental Salesforce data and schema extraction is performed by the CopyStorm application from Capstorm – replicating the Salesforce database table for table, field for field, to a local relational database.

- CopyStorm’s incremental extraction process keeps the source Salesforce’s data export in sync with the Salesforce environment – including any metadata or field changes that may occur mid-migration.
- Data and metadata extraction is fully automated, updating the data extract on the schedule required by the migration. Update the CopyStorm database up to every 5-10 minutes to keep the database in close time stamp proximity with Salesforce, synchronizing Salesforce data changes to the database.

Incremental Salesforce data uploading is performed by the CopyStorm/Restore application from Capstorm – creating or updating connected data within the target Salesforce instance using data stored in a local relational database.

- CopyStorm/Restore handles the entire Salesforce data upload, including the upload of more complex objects such as Attachments, Content Versions, even Salesforce metadata and Chatter!
- Relationships between uploaded data are preserved – even if the related records are uploaded at different times during the course of the ETL project.
- The upload process is fully automated. CopyStorm/Restore handles all relationship uploading, providing a reliable and consistent way to add and update data in Salesforce.

The remainder of this document provides a detailed view of each step in the Extract-Transform-Load process, as well as best practices to follow when migrating data into Salesforce.



Step 1: Extract Data Into A Staging Environment

The Staging Environment is a holding area for data being migrated to the target Salesforce – prior to being transformed into the target Salesforce's schema. The Staging Environment typically consists of databases, CSV files, or other sets of formatted data.

The schema of the target Salesforce environment should also be extracted. Extracting the target Salesforce schema to the Migration database is crucial because the Migration database allows off-the-shelf database transformation tools to be used when copying data into the format required to load into the target Salesforce.

Loading data into the Staging Environment must be a repeatable, incremental process. This allows data in the source environment to continue changing throughout the development of the ETL environment and initial data migration. Similarly, the target Salesforce may change during the ETL process, so it is necessary to have a way to update the Migration database's schema to keep in step with Salesforce changes.

Extract Salesforce Data

Migrating Salesforce data from one Salesforce environment to another requires that all data being migrated be exported from the source Salesforce instance to a Staging database. Exporting to a Staging database allows the remainder of the Extract-Transform-Load process to be performed using widely available relational database tools, eliminating the complexities of working with the Salesforce API. Exporting and incrementally updating the Staging database also allows users to continue working in the source Salesforce instance during the migration. The Staging database is periodically updated to include all new and modified records. This keeps the data export identical to the source Salesforce – allowing you to incrementally move changed data throughout the course of the migration project.

After creating the Staging database with CopyStorm, the database can be:

1. Queried directly by off-the-shelf database transformation tools.
   - The Staging database contains every Salesforce object (including Attachments, Salesforce Files, Chatter, and Knowledge Articles). This includes a table for every object type with a column for each of the object's fields.
2. Incrementally updated with new and modified records throughout the ETL process – the Staging database can be updated as often as every 5-10 minutes.
3. Updated as schema changes occur in the source Salesforce environment.
   - CopyStorm automatically updates the corresponding database table after a new custom field is added in Salesforce or creates a new database table when a new custom object is added.
   - New fields and tables are automatically populated with new and modified data whenever a new field or table is added.

For more information detailing how to create a database-based Salesforce backup, please see:

- https://learn.capstorm.com/copystorm/

Extract Non Salesforce Data

Migrating data from non Salesforce / third-party applications to Salesforce require that all data being migrated be exported to the Staging environment. Due to the varying nature of third-party applications, the data load into the Staging Environment can be done in a number of ways. The export approach will depend on both the form the data takes and where it is hosted. For example:

1. Data hosted in a third-party application may be directly accessible via a database connection.
    - In this case it may be feasible to directly query the source system when transforming data – but it is usually preferable to replicate the source database into the Staging Environment.
2. Data kept in shared spreadsheets can be made accessible for the migration process.
3. Data hosted in a third-party cloud-based application can be backed up in a format accessible by the transform tool.

In all cases the end goal is to provide the transform tool with data access in a usable format so that data can be translated into the target Salesforce schema.

Create the Migration Database

Loading data into Salesforce using CopyStorm/Restore requires that the data be transformed to match the target Salesforce schema. This can be done by extracting the target Salesforce schema to a local relational database then populating the schema with data extracted from the source environment(s). After populating the Migration database, CopyStorm/Restore can be used to load data and relationships from the database into Salesforce. A CopyStorm/Restore data load will maintain relationships between individual records and allows the load to be expanded later to include additional data.

To create a Migration database using CopyStorm, follow these instructions:

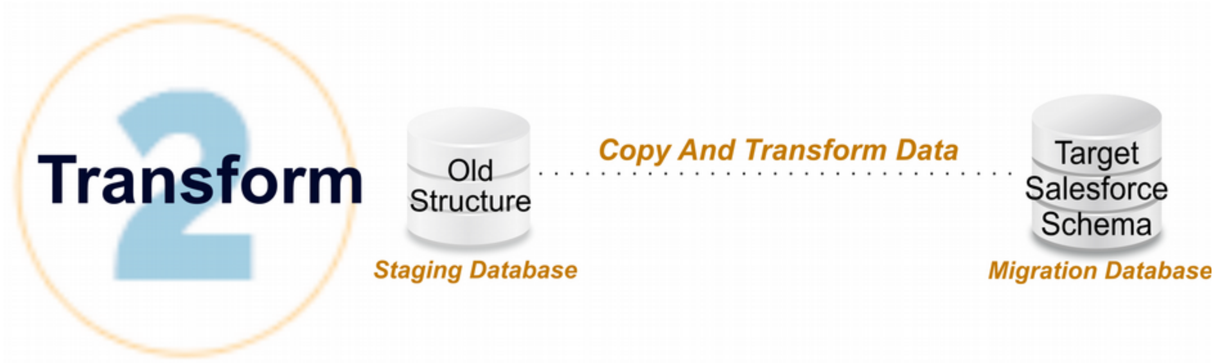
- https://learn.capstorm.com/migration-schema-backup

The Migration database's structure mirrors the structure of the target Salesforce, including:

- A database table for every Object (standard and custom) in the target Salesforce with a database column for each field in the Object.
- Metadata tables describing the target Salesforce schema that provide a way to identify field and table differences between source and target Salesforce environments.

CopyStorm can also keep the schema in sync with the target Salesforce. For example:

- If a field required for the migration is added to the target Salesforce, a new database column is added to the corresponding table. The CopyStorm/Restore data load process will automatically pick up the change.
- If a field data type is changed in the target Salesforce, CopyStorm will modify the corresponding column definition in the Migration database to match the new data type. The CopyStorm/Restore data load process will automatically pick up the change.



Step 2: Copy Data Into the Migration Database

The purpose of the transform process is to:

1. Copy data extracted from the source Salesforce instances or third-party data sources into the schema required by the target Salesforce instance.
2. Manage data changes required to transform data into the format needed by the target Salesforce instance.

Copying data from the Staging database(s) into the Migration database is as simple as any other database migration project – that is to say, complicated. Fortunately the use of Staging databases and a Migration database turns the problem of Salesforce data transformation into a standard relational database transformation – allowing the use of familiar database-oriented transform tools and eliminating the additional complexity of working with the Salesforce API.

There are a number of Salesforce-specific items to handle when copying data to the Migration database:

1. Transform related data so that the import will preserve relationships.
2. Transform data that is related to preexisting Salesforce records so that the records will be related after the data import.

Preserve Relationships

Loading connected data into Salesforce can be tricky, requiring that record references be preserved when copying data into the Migration database. CopyStorm/Restore identifies references between records by comparing the values of Ids to the values of reference fields, requiring that Id values be consistent in the Migration database. When migrating data from one Salesforce instance to another, the Salesforce Ids exported from the source Salesforce instance can typically be used directly. This will only cause issues if any of the Ids are already present in the target Salesforce. When importing data from a third-party system, creating synthetic Ids as part of the Transform process will cause the data to be uploaded with all relationships preserved.

For example, if an Account record is created in the target Salesforce schema database with a generated Id of “001ZZ0000000000001” and a Contact is created with an AccountId value of “001ZZ0000000000001”, CopyStorm/Restore will:

1. Upload the Account, recording the new Salesforce Id of the “001ZZ0000000000001” Account record.
2. When uploading Contacts, take any AccountId with the value of “001ZZ0000000000001” and substitute the Salesforce Id of the new record.

This is the behavior followed for all reference fields, ensuring that related data remains connected after uploading.

This process is fully incremental and allows for:

1. Data loading to be paused and resumed at any time.
2. Recovery after network failure.
3. New records to be added in later restore jobs, preserving relationships with previously uploaded data.

Creating Salesforce Ids

If the migration requires the creation of synthetic Salesforce Ids, there are a few important rules to follow:

1. Ensure that each Id starts with the corresponding table’s three-digit prefix.
    - For example, “Account” Ids always start with “001”.
    - This is required for CopyStorm/Restore to find related records in polymorphic tables – tables with reference fields that could point to more than one table. Examples of this behavior are “Note” and “Attachment”. They both contain a field named “ParentId” that can point to a large number of objects.
2. Ensure that each record has a unique Id that is not present in the target Salesforce instance.
    - For example an “Account” could have a generated Id of “001ZZ0000000000001”.
3. Ensure that all record references use the generated Id.
    - CopyStorm/Restore will compare reference fields with the Id column to find matches – just as if they were Salesforce Ids in a CopyStorm backup.
    - Consistent reference values will cause CopyStorm/Restore to automatically create relationships in the target Salesforce when uploading data.

Handle Preexisting Records and Avoid Duplicates

Many data migrations require that a portion of the data import reference data already in the target Salesforce. This may require you to:

1. Skip duplicate data.
    - Preexisting data in the target Salesforce may match data in the import. For example, a Contact being imported may match a Contact in the target Salesforce – with the same email address or name.
    - Accounts may have names matching Accounts in the target Salesforce. These may need to be merged to prevent duplicate Accounts from being created.
2. Merge child data for duplicate records.
    - Duplicate records may have child data that should be merged into the target Salesforce. For example, Accounts with Names already in the target Salesforce may be duplicates, so all related Opportunities, Contacts, etc… should be added to the matching Account.
    - This will merge data into a preexisting Account, but will not modify any other Account information in the target.
3. Update User references to ensure that record ownership is preserved.
    - The User login “myName@oldCompany.org” in the Migration database may correspond to the User login “myName@newCompany.com” in the target Salesforce.
    - Salesforce’s globally-unique rules for User logins make this process very common during migrations.

The following sections provide ways to handle these situations by using migration features built into CopyStorm/Restore.

Merge With Preexisting Data

CopyStorm/Restore is very flexible when setting up data merge rules. When a record is identified as already present in the target Salesforce it will be updated, otherwise it will be inserted. The Salesforce Id column is generally used to identify records to update. However, any column or combination of columns can be used. For example:

1. Contacts with the same Email could be considered to be duplicates.
2. Accounts with the same Name in the same City could be considered to be duplicates.

CopyStorm/Restore can match on any combination of fields when determining whether a record should be inserted or updated – including External Ids and record Names. If there is a matching record in the target Salesforce, the matching record will be updated. If there is no matching record, a new record will be created. This can minimize complexity in data transformation while ensuring that data is imported to the appropriate records.

For more information about this feature, please see:

- https://learn.capstorm.com/migration-record-matching

Map Data References

System tables, such as the User or Role table, will rarely be migrated directly from one Salesforce to another. However, being able to map references to these tables is often needed to preserve data integrity. CopyStorm/Restore maintains these references using a map of values in the Migration database to values in the target Salesforce.

For example, two Salesforce instances are being merged and users of the old Salesforce have access to both instances for a period of time. New Users are created in the new Salesforce, given new logins, and begin entering new data into the new Salesforce. Employees login to the old Salesforce to reference and update Opportunities, Cases, and other data until the data is moved to the new Salesforce.

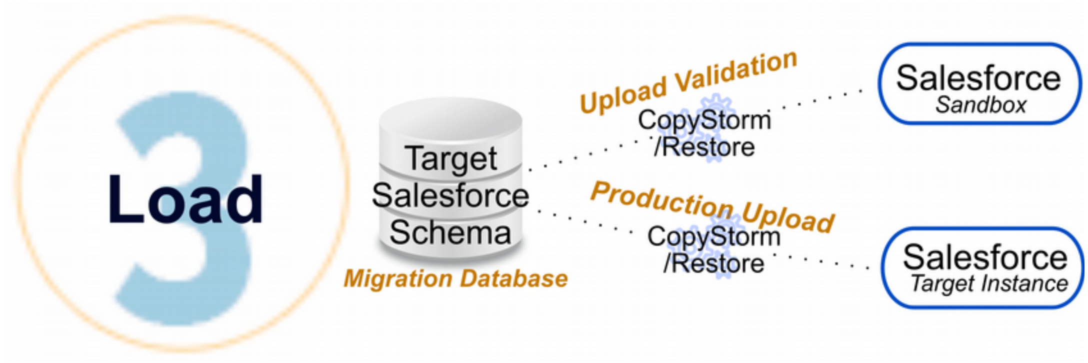
In this case, any data under the Users should be migrated so that the data is under the corresponding User in the new Salesforce.

CopyStorm/Restore provides this functionality by using a mapping of old User value to new User value – typically login Id or Email. Beyond User changes, this behavior is frequently required for Roles and Groups. Any reference can be mapped with CopyStorm/Restore, allowing very flexible import rules and data merging. Providing the data mapping allows CopyStorm/Restore to automatically upload complicated reference changes such as:

- Moving all data under the User Id “myname@oldorg.com” to be under the User in the target Salesforce with the User Id “myname@neworg.com” – modifying all User references, including Record Owner and Created By!
- Updating all Role references to the Role named “SW Dev Manager” in the old Salesforce to the Role named “Software Manager” in the new Salesforce.
- And any other data mapping provided – any and all references can be mapped if needed!

For more information, please see:

- https://learn.capstorm.com/migration-mapping



Step 3: Load Connected Data Into Salesforce

The final step of the ETL process is to import data from the migration database into the target Salesforce.

This is done by using CopyStorm/Restore to directly import records from the Migration database into the target Salesforce. This includes:

1. Automatically importing connected data (no manual Id mapping or CSVs!).
2. Incrementally importing new and modified data by:
   - Uploading changes for records that are modified between uploads.
   - Maintaining record relationships automatically – between any records that have been uploaded, even if they were migrated at wildly different points in the ETL lifecycle.
3. Consistently validating the restore configuration against a sandbox. This allows any data restoration to be tested in a sandbox prior to being used to upload data to the production Salesforce environment.
4. Automatically recovering from errors communicating with Salesforce – it can even be paused mid-restore and resumed later!

To load data into Salesforce using CopyStorm/Restore:

1. For a full data load, identify the records to upload – for example:
   - Select all Accounts, Contacts, and Leads by clicking a few checkboxes.
   - This is all that is needed – just select the tables. Data will be uploaded in the correct order and all relationships will be created in the target Salesforce.
2. For a partial load, select related data using a point-and-click UI. Click a few checkboxes to upload:
   - All child Contacts, Opportunities, Attachments, or other data related to the selected top-level records.
   - The Parent Account, Primary Contact, or other data referenced by the selected top-level records.
   - Nest any other related data – to any relationship depth. CopyStorm/Restore can handle any level of complexity or relationship depth when uploading connected data.
   - No matter how complicated the relationships are between records, CopyStorm/Restore will figure out the correct order to upload and ensure that record relationships are created in the target Salesforce.

After identifying data to upload, press the “Start Restore” button. This will do all necessary Salesforce data loading and relationship restoration.

The data upload can be paused and resumed at any time – CopyStorm/Restore will pick up where it left off. Record selection can be updated at any time – either to upload changes to previously uploaded records or to add new data to a finished or partial upload.

For more information about loading data with CopyStorm/Restore, please see:

- https://learn.capstorm.com/copystorm-restore/

Best Practices for Salesforce Data Import

The majority of Salesforce data can be loaded as normal relational data, however there are several caveats to the data loading process. For example, migrating users and importing security information typically involves a greater level of review and validation than a bulk data load. Validation rules, triggers, and workflows may also cause issues when uploading data, and managed package data will require special consideration. This section provides a number of useful strategies when dealing with corner cases specific to the Salesforce platform.

Migrate User Security Information

User account and security role migration is likely to require manual review and hand-editing of the security schema to ensure compatibility with security rules in the target Salesforce. Migrating is likely to include one or more of the following:

1. Reassigning new or migrated Users to parallel roles within the target Salesforce.
2. Creating roles for Users whose role does not exist in the target Salesforce.
3. Pruning and consolidating roles to ensure that duplicate or functionally-identical User roles are not created in the target Salesforce.

A manual review of roles will identify duplicate or functionally-identical roles. These differences are difficult to identify using an automated procedure. Salesforce security administrators for the target organization will also be interested in reviewing any changes to the security layout of their organization.

Migrate Metadata

Metadata migration is a common task that must be performed prior to importing records. For example, changing the edition of a Salesforce instance may require that all metadata in the source Salesforce to be migrated before moving any data! There are a number of ways to import metadata elements into the target Salesforce:

1. CopyStorm/Restore's metadata migration features.
   - When migrating metadata from one Salesforce instance to another, CopyStorm/Restore's point-and-click GUI provides a streamlined way to import tables and field definitions, as well as other metadata elements, into the target Salesforce.
2. Salesforce's built-in metadata deployment tools.
   - If the development organization is already familiar with deploying metadata via Salesforce's built-in Change Sets or ANT scripts, existing deployment practices can be used to migrate metadata elements.
   - CopyStorm/Restore can speed up this process by creating the required directory structure without manually updating metadata files.
3. Creating new features in a sandbox and deploying changes to production.
   - If migrating data from an independent third-party system, this may be the simplest way to create new tables and fields.
   - CopyStorm and CopyStorm/Restore can be used to migrate the changes to production – back up the sandbox's metadata with CopyStorm then use CopyStorm/Restore to push the changes to production.

Metadata changes need to be replicated to the ETL environment before being used for data transformation. The same CopyStorm configuration that originally created the target Salesforce schema database can be used to incrementally update the database with any new or modified tables and fields in the target Salesforce environment:

- https://learn.capstorm.com/migration-schema-backup

Address Validation Rules, Triggers, and Workflows

It is best practice to ensure that all migrated data matches the validation rules in the target Salesforce. However, this is not always possible. If data does not match all of the validation rules, a common case when importing legacy data, CopyStorm/Restore can automatically disable the problem rules prior to beginning the upload then automatically re-enable them after the upload completes. Validation rules are global, so this will disable the rules for the entire Salesforce environment while the upload is running.

CopyStorm/Restore can also disable both triggers and workflows while performing an upload, but this will disable them for everyone using Salesforce. Unlike validation rules, which will reject invalid data but are unlikely to impact other business processes, triggers and workflows often contain runtime logic that can cause wider problems if they are not handled properly.

For example:

- A trigger that sends the VP of Sales an email any time an Opportunity with a value more than $100,000 is inserted.
  - If not disabled while loading data, this can cause 100s of emails to land in the VP’s inbox – embarassing!
- A trigger that creates data in a second table when records are inserted.
  - If disabled while loading data the second table will never be populated, causing problems later when using the table.

It’s important to review any code that may be fired by a data import and ensure that any triggers left enabled during the import will not cause problems.

To minimize the amount of time validation rules, triggers, and workflows are disabled in the target Salesforce, CopyStorm/Restore can disable and re-enable them as part of the data load process. CopyStorm/Restore disables the selected items immediately prior to loading data, then re-enables them immediately after the data load is complete or paused:

- https://learn.capstorm.com/migration-disable

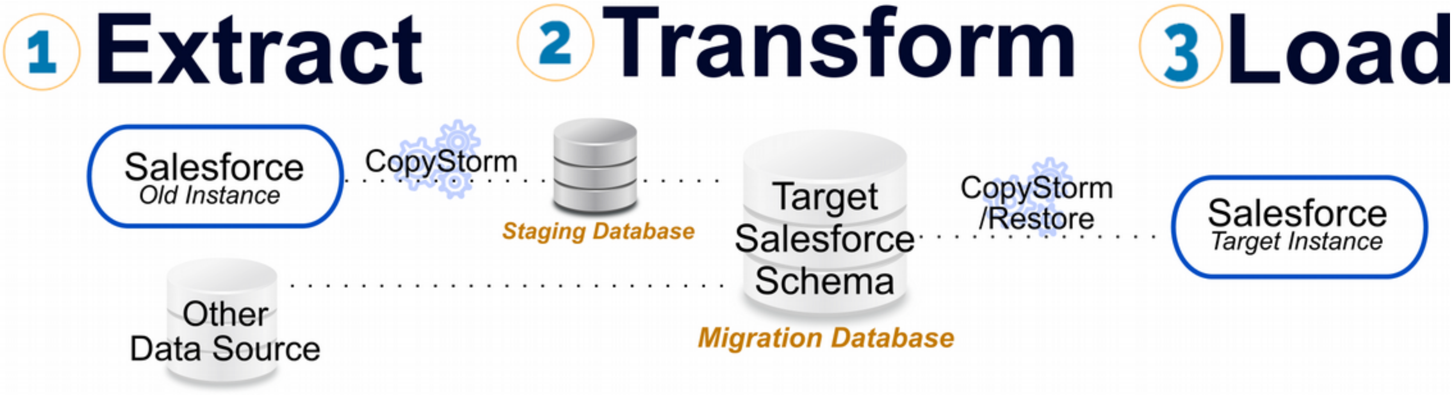
Migrate Managed Package Data

CopyStorm/Restore can upload data to managed packages – just like any other data. However, managed packages come with a unique restriction impacting data creation:

- Triggers and workflows in managed packages cannot be disabled via the Salesforce API.

This poses a tricky problem for data migrations: importing data will kick off all automation built into the package. Unfortunately, there is no one-size-fits-all answer to this problem. Some managed packages contain the ability to disable their automation, others do not. Some packages provide detailed documentation on their behavior – allowing it to be understood and planned for as part of a migration – and others do not.

Prior to importing managed package data, review the behavior included in the package and be sure to understand the implications of any automation that cannot be disabled.



Salesforce Data Migration – Quick Reference

A complex Salesforce data and metadata migration project can be turned into a much simpler database transformation project by following this process:

1. **Extract** the **data** being migrated – from either a source Salesforce or any other data repository.
   - Create and incrementally update a Salesforce backup database with CopyStorm.
2. **Extract** the **schema** of the target Salesforce.
   - Extract the target Salesforce schema into the Migration database with CopyStorm.
3. **Transform** data into the target Salesforce schema.
   - Copy data into the Migration database using standard database tools.
4. **Load** data into the target Salesforce.
   - Load connected data and metadata into the target Salesforce with CopyStorm/Restore.

Every part of the process is incremental and repeatable – allowing all steps to be thoroughly tested before uploading data into the the target production Salesforce instance.

Get CopyStorm and CopyStorm/Restore For Your Project

To download and try CopyStorm and CopyStorm/Restore:

- Start a free 30 day trial of CopyStorm today by going to https://www.capstorm.com/download
- Start a free 30 day trial of CopyStorm/Restore by contacting sales@capstorm.com or by calling +1 314.403.2143

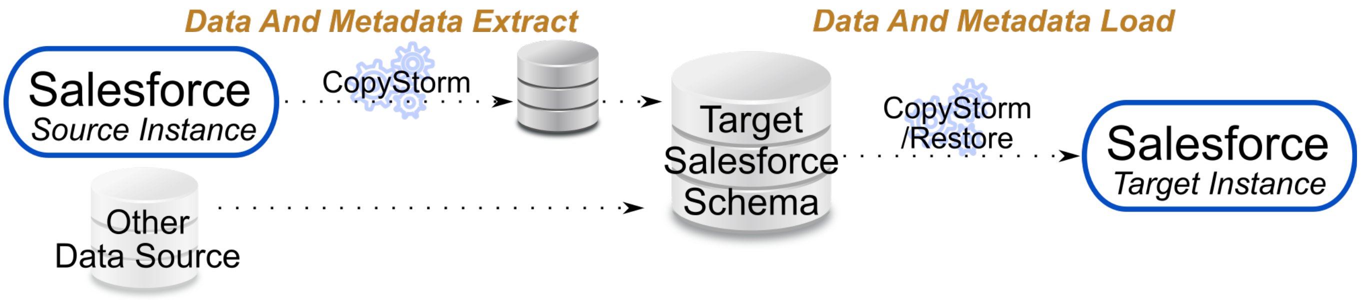
Additional product documentation, best practices, and installation instructions are available at:

- https://learn.capstorm.com

CAPSTORM

Salesforce Data Solutions

Migration



Data And Metadata Extract

Data And Metadata Load

Salesforce
Source Instance

CopyStorm

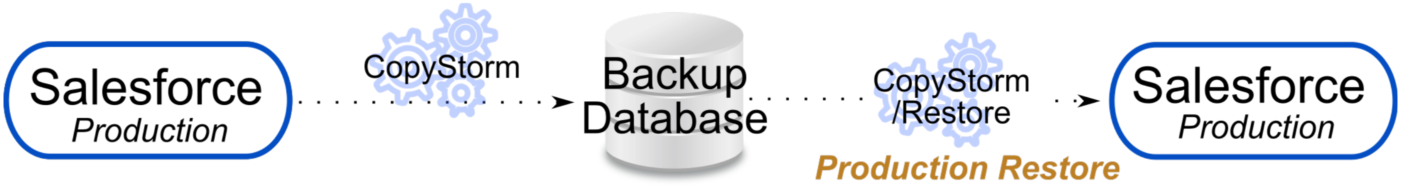
Other
Data Source

Target
Salesforce
Schema

CopyStorm
/Restore

Salesforce
Target Instance

Disaster Recovery



Salesforce
Production

CopyStorm

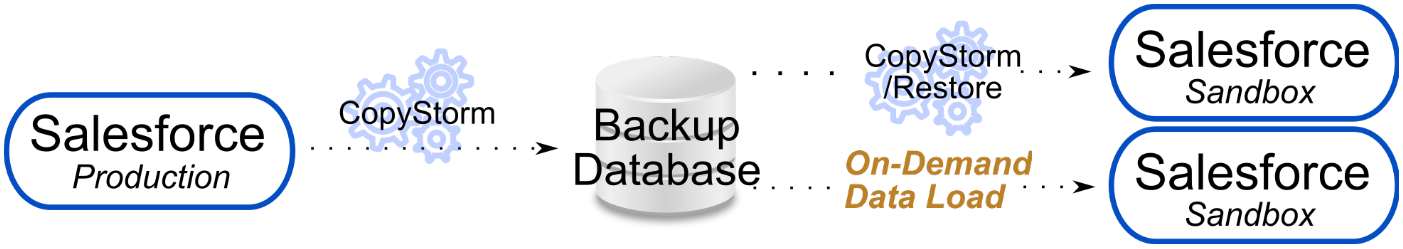
Backup
Database

CopyStorm
/Restore

Salesforce
Production

Production Restore

Sandbox Seeding



Salesforce
Production

CopyStorm

Backup
Database

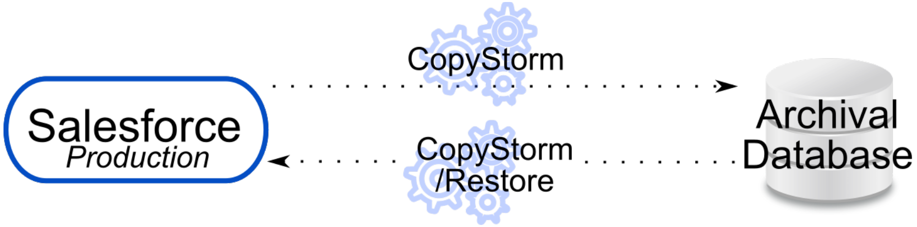
CopyStorm
/Restore

Salesforce
Sandbox

On-Demand
Data Load

Salesforce
Sandbox

Data Archival



Salesforce
Production

CopyStorm

CopyStorm
/Restore

Archival
Database

Keep deleted and archived records forever - with no data limits!
If needed, restore archived data to Salesforce.

CAPSTORM

info@capstorm.com | www.capstorm.com | +1 314.403.2143