

Understanding CopyStorm/Restore

Copyright CAPSTORM, 2010-2015

This document provides information essential to understanding how CopyStorm/Restore restores records and complex relationships to a Salesforce database.

Table of Contents

Executive Summary.....	2
How Does CopyStorm/Restore Work?.....	2
The CopyStorm/Restore Tracking Database – the Most Important Topic.....	3
The CopyStorm/Restore Process.....	5
Common Issues.....	7

Executive Summary

To use CopyStorm/Restore effectively, understanding the basic approach used by the tool is essential knowledge. This document provides the core information that you will need without forcing a deep dive into CopyStorm/Restore internals.

The intended audience is people who will use CopyStorm/Restore on a regular basis.

How Does CopyStorm/Restore Work?

At its core, CopyStorm/Restore (CSRestore) is like a small army of data entry clerks that know:

- how to enter records in the proper order.
 - Example: Accounts should be restored before Contacts
- how to patch up record references.
 - Example: A Contact.ReportsTo field may need to be set after both contacts are created.
- how to take a coffee break and return to work later.
 - Example: The communication link to Salesforce goes down for a couple of hours and the data entry needs to resume where it stopped.
- how to accept additional data entry work from the boss.
 - Example: A decision is made to also restore Leads after most of the restore has completed.

Remember the saying 'Given a large number of monkeys each hitting keys on a typewriter for an infinite amount of time, they will eventually produce all the works of Shakespeare', then you are close to understanding CopyStorm/Restore – just replace monkeys with program threads, typewriters with internet connections, and Shakespeare with your Salesforce data.



Before getting to the details of how CopyStorm/Restore works, we must cover the most important topic – the CopyStorm/Restore Tracking Database.

The CopyStorm/Restore Tracking Database – the Most Important Topic

Understanding this section is the key to understanding how CopyStorm/Restore works.

CopyStorm/Restore has three basic jobs:

- Figure out which records need to be restored.
- Restore records to Salesforce
- Be able to resume a restore if something prevents a restore from completing. A number of things can cause a restore to fail to finish:
 - The link to Salesforce goes down.
 - The user cancels a restore while it is progress.
 - The user reboots his/her machine while a restore is in progress.
 - The system group decides to install patches and reboot while a restore is in progress.
 - The CopyStorm/Restore program runs out of memory and crashes.

CopyStorm/Restore accomplishes these three basic job by tracking the progress of a restore in a local relational database. Here is what you need to know about this “Tracking Database.”

- Tracking Databases do not require any administration. They are local H2 databases created by CopyStorm/Restore.
- By default all Tracking Databases are stored in the directory \$HOME/.capstorm/copyStormRestoreSets.
- By default, each CopyStorm/Restore configuration file creates a new Tracking Database the very first time a restore is started.
- The name of the current tracking database can be found on the CopyStorm/Restore “Global Parameters” tab. By default the name of the tracking database is a generated Globally Unique Identifier (GUID).
- You can name the Tracker Database anything name you want (as long as the name is a legal filename).

[Main](#) | [Global Parameters](#) | [Restore Set Editor](#) | [Restore Plan](#) | [Restore](#) | [License](#)

Global Restore Parameters

Set various parameters that control the default properties used when adding a table to property for context specific help.

Common Parameters

Default Salesforce User ?

Default Restore Type ? Insert Update

Tracker Database Name ?

Default Max # Per Update ?

Salesforce Writers ?

SOQL Log File ? ...

Truncate SOQL Log On Restore ?

Enable Trace Mode ?

Table Specific Parameters

Opportunity ?

OpportunityLineItem ?

Here is the key point: The Tracker Database retains the history/progress of a restore across each use of a CopyStorm/Restore configuration file. The means:

If you use a Tracker Database to restore records to “Sandbox A” then attempt to restore “Sandbox B”, no records will be written to “Sandbox B” because CopyStorm/Restore thinks they have already been restored.

The solution, in this case, is to tell CopyStorm/Restore to reset its tracker database (by clicking on the “Restart Restore” button on the Restore tab).

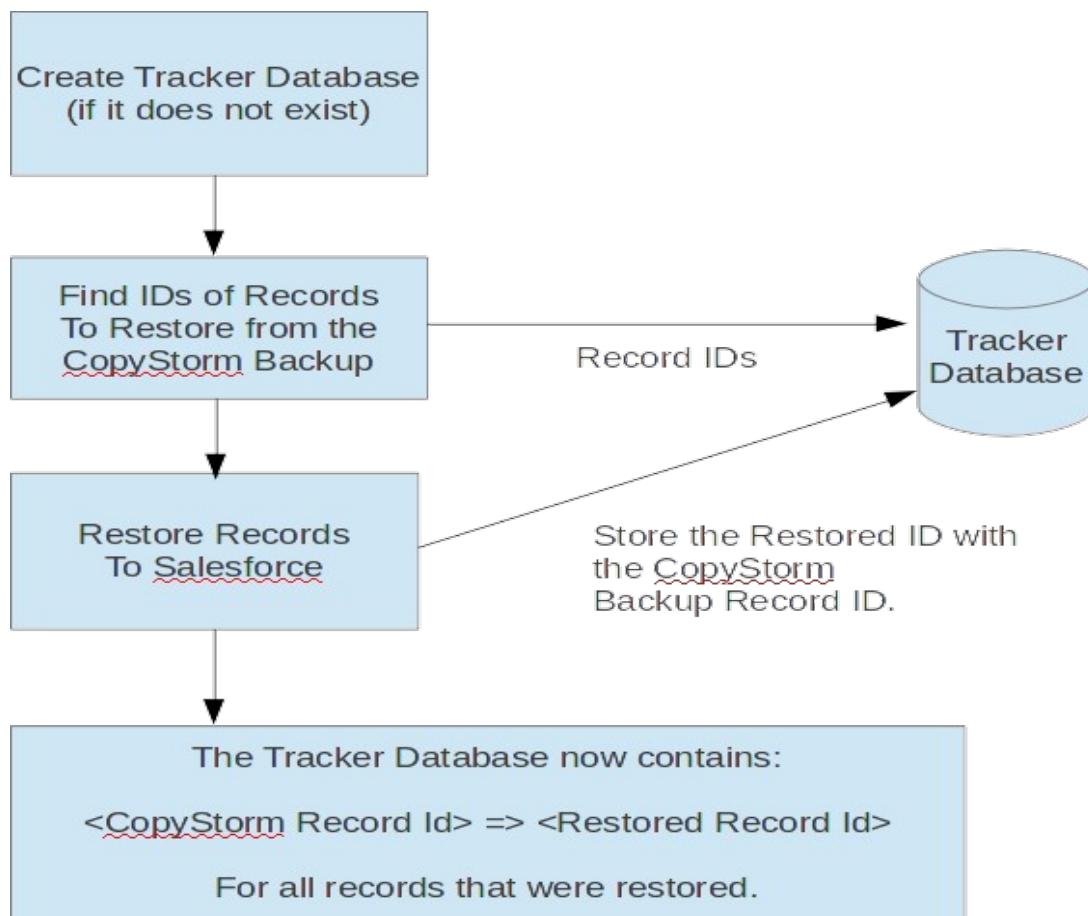
The content of a Tracker Database is fairly simple. For each record restored (or to be restored), a Tracker Database records:

- The unique ID of the record in the CopyStorm backup.
- The unique ID the record was restored to in the target Salesforce. This value is NULL until a record has been restored.

If this ID is the same as the ID in the CopyStorm backup, then the restore operation was an UPDATE. If this ID is different, then the restore operation was an INSERT.

Reality is a bit more complex than this description because CopyStorm/Restore also takes care of restoring infinite depth parent-child and circular relationships (not to mention limited child nodes and general child node filtering). If you want this level of understanding then read upon NP-Complete graph traversal problems and schedule a beer with the CAPSTORM CTO.

The following diagram illustrates the behavior of the Tracking Database that is key to understanding CopyStorm/Restore.



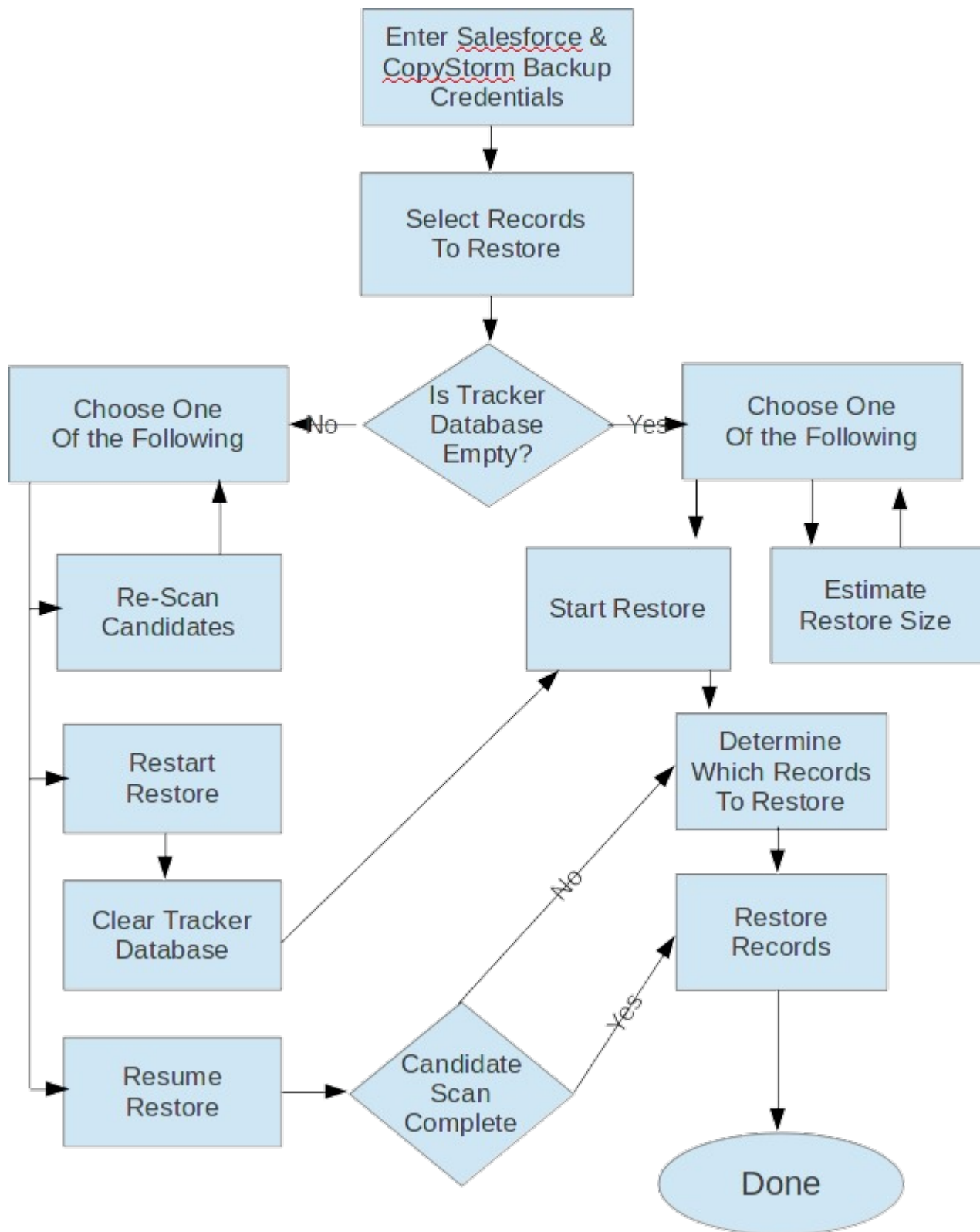
The CopyStorm/Restore Process

Now that you have a good understanding of the role played by a CopyStorm/Restore Tracking Database, the overall flow of CopyStorm/Restore can be explained.

There are four basic steps to performing a restore.

- Specify credentials for a CopyStorm backup and a Salesforce instance (the instance to be restored).
- Specify which records should be considered for a restore.
- If the tracking database is not empty then one of the following decisions must be made.
 - Continue the restore from where it left of the last time it ran.
 - Clear the tracking database and assume that none of the records in the backup have been restored. This is useful when a CopyStorm/Restore configuration file is used for a new empty sandbox.
- Run the restore.

More of the details are illustrated in the following diagram.



Naturally, a CopyStorm/Restore configuration can be save to a file and re-opened for later use. In this case the process starts at either of the “Choose One of the Following” boxes.

Common Issues

Since CopyStorm/Restore is just like 100s of data entry clerks, the same kind of issues sometimes arise.

- Triggers are executed on insert/update operations. Sometimes triggers need to be disabled during a restore.
- Validation rules are executed on insert/update operations. If the rules have changed over time, older data may not pass. Sometimes these rules need to be disabled during a restore.