REVBLACK

CLAY TO CRM SETUP CHECKLIST

Connect Clay to HubSpot or Salesforce so it updates the records you already have and only creates the ones you don't.

WHO IT'S FOR

RevOps and marketing ops leads connecting Clay to HubSpot or Salesforce, or cleaning up after a sync that went sideways

WHAT YOU NEED

A Clay plan that includes CRM integrations, admin access to HubSpot or Salesforce, and a settled ICP with the signals you want Clay to watch

FULL GUIDE

revblack.com/guides/clay-hubspot-salesforce

How to use this checklist

Work through it in order. Each step starts with what you'll have when it's done, so you know what you're working toward. Tick each box as you go and write your decisions straight onto the page.

Leave auto-run off on every create and update action until step 4. Nothing should touch your real records before you've tested on a small batch.

0 **Before you start**
Line up the people and permissions that depend on someone else's time.

1 **Pick the right action**
Decide what each Clay table runs against your CRM, and which object it writes to.

2 **Decide which fields to write back**
Build the field map: each Clay column, the CRM field it writes to, and what happens to blanks.

3 **Stop duplicates before they start**
Put a lookup in front of every create action and pick a matching field for each object.

4 **Test on a small batch**
Run 20 rows, check them in the CRM, then turn on auto-run.

The example used throughout: marketing hands you a conference list of 400 companies, and 250 of them are already in your CRM. Set up right, Clay updates those 250 and creates 150. Set up wrong, you get 400 new companies, 250 of them duplicates.

HubSpot calls the top-level record a company and Salesforce calls it an account. This checklist says companies. The full guide has screenshots of each Clay setting and links to Clay's and HubSpot's own documentation.

BEFORE STEP 1

LINE THESE UP FIRST

Most of these depend on someone else's time, so start on them early.

A Whichever CRM you use

- [ ] Agree with your team on which CRM fields Clay is allowed to write to, and who signs off when a new field is needed.
- [ ] Pick a short list of records to enrich first. You'll test on these in step 4.
- [ ] Find someone who can create CRM fields, because step 2 usually turns up a few that don't exist yet.

Signs off on new fields _____

Can create CRM fields _____

Test records (list or view) _____

B If you use HubSpot

- [ ] Decide which objects Clay should be able to write to. When you connect, Clay asks for write access to contacts, companies, leads and deals by default, and you can switch off the ones it shouldn't touch.

CLAY CAN WRITE TO

☐ Contacts ☐ Companies ☐ Leads ☐ Deals

C If you use Salesforce

- [ ] Find out who will approve the Clay connection. They need **API Enabled** and **Connected App Access** on their profile or permission set, plus **Approve Uninstalled Connected Apps** if the approval throws an error.
- [ ] Make sure that person is on a standard user license. The approval won't go through from an **Integration User** or **API Only** license.
- [ ] Ask your admin which duplicate rules run on leads and contacts. Step 3 explains why they matter.

Connection approver _____

DUPLICATE RULES ON LEADS AND CONTACTS

STEP 1 OF 4

PICK THE RIGHT ACTION

The action decides whether a record gets updated or a new one gets made. For records you already have, update beats create. A create action makes a new record every time it runs, and auto-run is on by default.

> **When this step is done, you'll have** the action each Clay table will run against the CRM, and the object it writes to.

- [ ] Write down what each Clay table is for: enriching records you already have, or finding new ones.
- [ ] Pick the action that matches. Update or look up for records you already have. Create only where you mean to add records.
- [ ] Leave auto-run off on every create and update action for now. Step 4 is where you turn it on.
- [ ] On Salesforce, decide whether you're writing to contacts or leads, and write the decision down. We recommend contacts for most teams, because if leads stay, every new field has to be mapped on both objects.
- [ ] On Salesforce, if you're considering **Create or update record**, check that the object has an external ID field.
- [ ] Connect your CRM: **Add column**, then **Add enrichment**, pick the action, then **Account** and **Add account**. Or connect ahead of time under **Settings**, **Connections**, **Create**.

YOUR CLAY TABLES

Clay table	What it's for	Action	Object

HubSpot actions in Clay

- Create object
- Look up object
- Update object (needs the object ID)
- Create association
- Get associated objects

No single create-or-update action. Look up and update work as a pair.

Salesforce actions in Clay

- Create record
- Look up record
- Look up records with SOQL
- Update record (needs the record ID)
- Create or update record (needs an external ID)
- Convert lead

You're done when every table has one named action and one object, and no create action is set to run on its own.

STEP 2 OF 4

DECIDE WHICH FIELDS TO WRITE BACK

Every data point you push into the CRM becomes a field your reps either trust or learn to ignore. Do the field map with the people who use the data, and write back only what a rep would open the record to find.

> **When this step is done, you'll have** a field map listing each Clay column, the CRM field it writes to, and what happens to blanks.

- [ ] List the fields your team will use, tied to your ICP.
- [ ] For each one, name the CRM field it writes to, and create that field first if it doesn't exist. Map onto an existing field whenever the definition matches.
- [ ] Settle formats now, so "United States" and "US" don't end up in the same field.
- [ ] Mark any Clay column that came back inconsistent, and keep it in Clay until the output is steady.
- [ ] Check the **Ignore blank values** setting on every update action, on both platforms. On Salesforce, with it off, blanks overwrite existing text values.
- [ ] On Salesforce, confirm the values Clay sends to every picklist match exactly, including case, and use the API value. Multi-select values take semicolons with no spaces. Check whether each picklist is restricted.

YOUR FIELD MAP

Clay column	CRM field it writes to	Format	When Clay comes back blank

Example line: Employee count, writes to Number of Employees, whole number, leave the existing value.

You're done when every Clay column that writes to the CRM has a named field, a format and a decision on blanks.

STEP 3 OF 4

STOP DUPLICATES BEFORE THEY START

Look the record up first, update it if it exists, and create it only if it doesn't. Lookups are free to run, while every create or update counts as an action execution in Clay. HubSpot won't catch duplicates for you either, because it doesn't match companies created through an integration on domain.

> **When this step is done, you'll have** a lookup step in front of every create action, and a matching field written down for each object.

MATCHING FIELD FOR EACH OBJECT

Companies	
Contacts	
Other object	

For companies, that's usually the domain. For contacts, use email: two records can share a name, but they can't share an email.

1 Build three columns, in this order

- [ ] **Look up object** (**Look up record** on Salesforce). Search on your matching field.
- [ ] **Update object** (**Update record** on Salesforce). In the ID field, type / and pick the record ID from the lookup. Turn on **Add run condition** so it only runs when the lookup found a record.
- [ ] **Create object** (**Create record** on Salesforce), with the opposite run condition, so it only runs when the lookup came back empty.

2 Check the rest

- [ ] Clean up companies that are already in the CRM twice, or check them by hand. A lookup can return more than one match, and the update only takes one ID.
- [ ] If you're creating contacts, link each one to its company: a **Create association** column on HubSpot, or the contact's Account field filled with the account's record ID on Salesforce.
- [ ] Don't count on Clay's auto-dedupe. It only compares rows inside your table, never records already in your CRM.
- [ ] On Salesforce, list the duplicate rules running on the object you're writing to.
- [ ] On Salesforce, confirm the **Duplicate rule override** option is off on Clay's **Create record** action.

You're done when no create action in your tables can run without a lookup in front of it.

STEP 4 OF 4

TEST ON A SMALL BATCH

Mistakes in this setup don't announce themselves, so you want to find them on 20 records instead of all 400. With auto-run off, select the action's cells for those rows, right-click and run the selected cells, or use the play button on the column header to run the first 10 rows.

> **When this step is done, you'll have** a tested table with values landing in the fields you mapped, and auto-run turned on deliberately.

- [ ] Run the table on a small batch with auto-run still off.
- [ ] Open the written records in HubSpot or Salesforce and check every mapped field.
- [ ] Check one record that already had values and one that was mostly empty, since the blank-values setting behaves differently on each.
- [ ] On Salesforce, run the first pass against a sandbox if you have one: **Settings**, **Connections**, **Salesforce Test Env**.
- [ ] Fix the mapping, then re-run the same batch before you widen it.
- [ ] Turn auto-run on once a batch comes back clean.

TEST LOG

Run	Rows	Updated / created	What you found and fixed

Auto-run turned on (date) _____

You're done when a batch runs and nothing lands in a field you didn't intend.

Check that it worked

The setup worked when your record count moves only when you meant it to, and anyone on your team can explain where a Clay-written field came from.

- [ ] Company and contact counts match what you expected after the batch.
- [ ] Every CRM field Clay writes to is in your field map, and no extra field picked up values.
- [ ] None of the records your team had edited came back with a blank where a value used to be.

After launch, keep a dashboard on the fields Clay owns (fill rates and integration errors) next to your new company and contact counts.

When something goes wrong

PROBLEM	LIKELY CAUSE	WHAT TO DO
New companies appeared where you expected updates	A create action ran with auto-run on and no lookup in front of it	Turn auto-run off, add the lookup, and keep create for rows the lookup came back empty on
Salesforce rejects a create	A duplicate rule on that object is blocking it	Find the record that already exists and update it. Leave the **Duplicate rule override** option off in Clay
A value won't land in a Salesforce picklist	The value doesn't match an option exactly, or it's the display label instead of the API value	Send the API value, case included, with semicolons and no spaces between multi-select values
A field your reps had filled in came back blank	The **Ignore blank values** option is off on the update action	Turn it on, then restore the values from property history in HubSpot, or from field history in Salesforce if it's turned on for that field

REVBLACK

RATHER HAVE SOMEONE BUILD IT WITH YOU?

We set up Clay with HubSpot and Salesforce, starting with the field map and the lookup order.

Talk to a Clay expert at revblack.com/contact-us