

Design sales territories with Territory Management 2.0.

Table of Contents

Why designing sales territories?

Understand Territory Management 2.0 basics

Territories for named accounts with subsidiaries

The upload sequence

How to update these territories?

Advanced Settings for Sales Operations

Why designing sales territories?

There are many ways to design sales territories within Salesforce. In small companies, sales representatives have a list of accounts they own, and own the opportunities related to these accounts. This list can be updated by the sales director each quarter. Ownership works great if your company targets small accounts, and if you have 1 sales hunter for each account.

This model doesn't work as the sales team grows. What if your accounts have dozens of subsidiaries? What if account based marketing managers, sales development representatives, pre-sales engineers and customer success managers work on the same accounts? Ownership model won't work in this case.

Usually, when a company reaches this level of complexity, each department manage their territory in Excel files, and on the Salesforce side, the Organization Wide Defaults are set to "Public Read Write" for all main objects such as accounts and opportunities. This isn't a best practice at all. Salesforce users should only have access to the objects / records they need to be able to work. They could make a mistake and update the wrong account / opportunity. They could see a record they're not supposed to see. Moreover, these Excel files aren't a single source of trust, and because each department worked in silo and designed their own territory strategy, the sales team isn't aligned with other teams. Territories and teams should be visible in Salesforce, and part of the selling strategy.

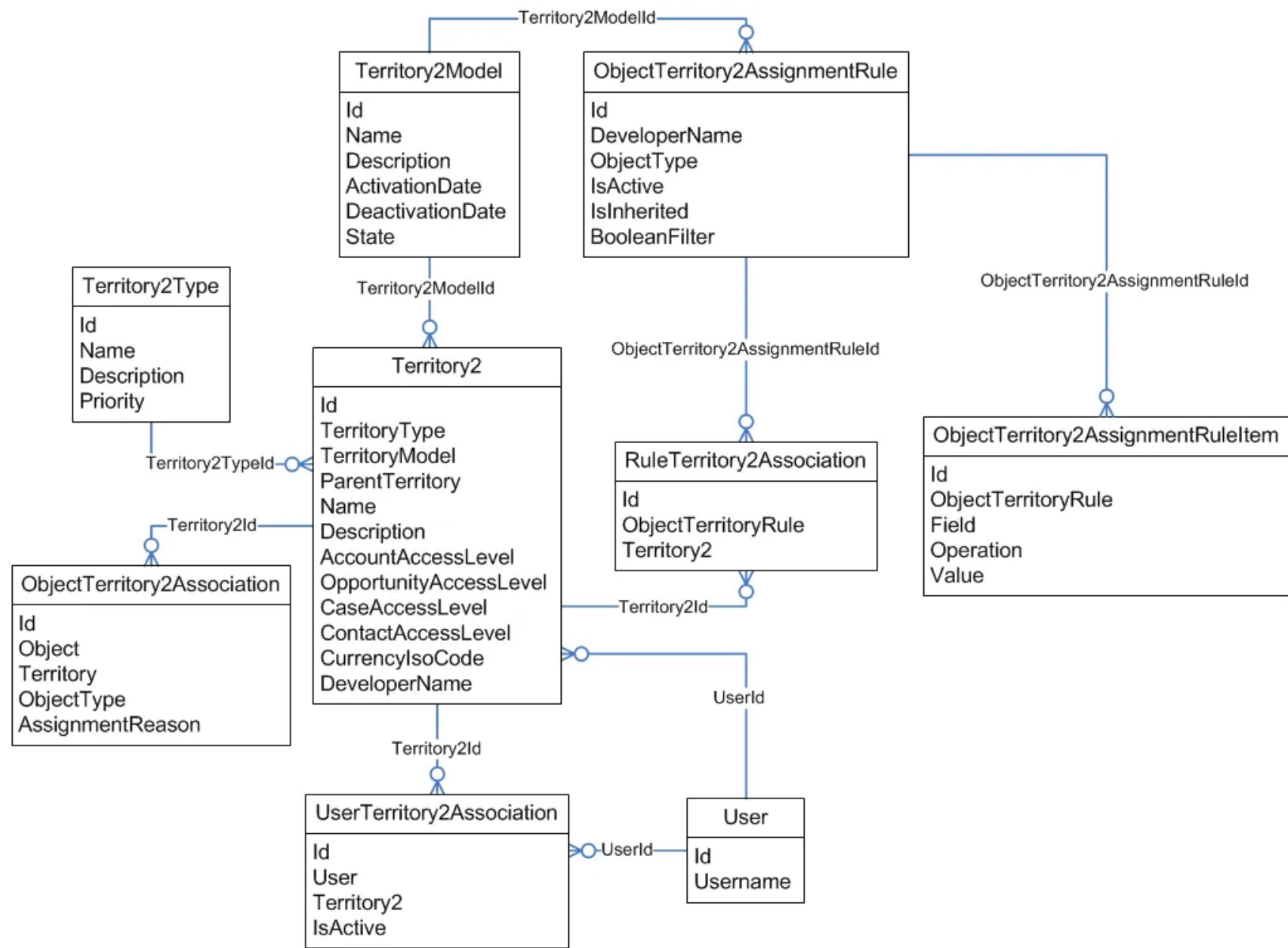
In this article, I'll explain how to efficiently design your sales territories with Salesforce Territory Management 2.0. The documentation on this topic gives an overview of standard functionalities, but doesn't help much if you company has a huge list of named accounts with subsidiaries.

How do you grant access to subsidiaries automatically? How do you maintain your territories?

Understand Territory Management 2.0 basics

First, I suggest you take a look at this [Trailhead Trail](#) to learn the basics.

Once you've understood the general functionalities, it's interesting to take a glance at the data model behind it. Here's the entity relationship diagrams (ERDs) for [Territory Management 2.0 Objects](#).



As you can see, territories open access right for main Sales Cloud objects such as Accounts, Opportunities, Contacts and cases. Thus, the first step is to update your Organization Wide Default to, at least, “Public Read Only” for account and opportunities (and contacts / cases if needed).

Territories for named accounts with subsidiaries

If your company targets huge accounts with hundreds of subsidiaries, and if there are more than one user that works on an account, I suggest you create territories similar to this one with the Data Loader – creating these territories through the Salesforce UI can take ages.

First, you have to create a custom field on the account object. We’ll use this field to create 1 territory per named account. This formula field, usually called “Ultimate Account ID”, returns the Salesforce of the account at the top of the account hierarchy. As you can see, this formula works as long as your hierarchy doesn’t have more than 6 hierarchy levels, which is enough for most companies.

```

1 BLANKVALUE( Parent.Parent.Parent.Parent.Parent.Parent.Id,
2 BLANKVALUE( Parent.Parent.Parent.Parent.Parent.Id,
3 BLANKVALUE( Parent.Parent.Parent.Parent.Id,
4 BLANKVALUE( Parent.Parent.Parent.Id,
5 BLANKVALUE( Parent.Parent.Id,
6 BLANKVALUE( Parent.Id,
7 Id))))))
  
```

There are no easy way to block a user from adding a 7th level of hierarchy, even with Apex Triggers. However, you can create another custom field that returns the account hierarchy level, and create a report that checks if you have any account for which this formula returns the NULL value. If so, you can alert the account owner to work on his account hierarchy. I named this formula fields : Account Hierarchy Level. In my company, 5 is the maximum.

```
1 IF( ISBLANK(Parent.Id) = TRUE, 1,
2 IF( ISBLANK(Parent.Parent.Id) = TRUE, 2,
3 IF( ISBLANK(Parent.Parent.Parent.Id) = TRUE, 3,
4 IF( ISBLANK(Parent.Parent.Parent.Parent.Id) = TRUE, 4,
5 IF( ISBLANK(Parent.Parent.Parent.Parent.Parent.Id) = TRUE, 5,NULL))))))
```

Here’s how I created a territory for each named account, utilizing the Ultimate Account ID formula field. I created one territory per named account, updated the territory access levels and added a team for this territory, with territory roles.

Territory

EIFFAGE SA

Hierarchy: [Finalcad Territory Model 2018](#) » [Standard - Strategic](#) » EIFFAGE SA

Assigned Users (7) | Manually Assigned Accounts (0) | Inherited Assignment Rules (0) | Assignment Rules Assigned to This Territory (1) | Child Territories (0) | Associated Groups (1)

Territory Detail

EditDeleteCloneView Accounts

LabelEIFFAGE SA

Territory TypeHigh Priority

Parent TerritoryStandard - Strategic

Description

Forecast Manager

Territory NameEIFFAGECONSTRUCTIONFRANCE

Territory ModelFinalcad Territory Model 2018

▼ System Information

Last Modified ByNathan Prats1/5/2018 10:33 AM

Last Modified Date1/5/2018 10:33 AM

▼ Territory Access Levels

Account Access LevelUsers in this territory can view and edit accounts assigned to this territory.

Opportunity Access LevelUsers in this territory can view and edit all opportunities associated with accounts in the territory, regardless of who owns the opportunities.

Case Access LevelUsers in this territory can view and edit all cases associated with accounts in the territory, regardless of who owns the cases.

Assigned Users

Manage Users

Action	Full Name	Username	Email	Active	Role in Territory	Is Forecast Manager
Edit Remove				✓	CSM	<input type="checkbox"/>
Edit Remove				✓	CSM	<input type="checkbox"/>
Edit Remove				✓	CSM	<input type="checkbox"/>
Edit Remove				✓	CSM	<input type="checkbox"/>
Edit Remove				✓	ABM	<input type="checkbox"/>
Edit Remove				✓	CSM	<input type="checkbox"/>
Edit Remove				✓	GAM	<input type="checkbox"/>

Manually Assigned Accounts

Add Accounts

No records to display

Inherited Assignment Rules

No records to display

Assignment Rules Assigned to This Territory

NewRun RulesAssign Rules

Action	Rule Name	Active	Apply to Child Territories	Modified Date
Edit Remove	EIFFAGE SA	✓	<input type="checkbox"/>	1/5/2018 10:34 AM

Child Territories

Create Territory

No records to display

Associated Groups

This territory is defined by this rule here below, based on the “Ultimate Account ID”, which is the account at the top of the account hierarchy. It means that for this account and all its subsidiaries, the 7 users listed above will have read / write access to accounts / opportunities and contacts.

Object Territory Assignment Rule

EIFFAGE SA

< Back to List: Object Territory Assignment Rules

Assigned Territories [1]

Object Territory Assignment Rule Detail

EditDeleteClone

Rule Details

Rule Name

EIFFAGE SA

Territory Model

Finalcad Territory Model 2018

Object

Account

Active

☒

i

Unique Name

EIFFAGE_CONSTRUCTION_FRANCE

Rule Criteria

Account: Ultimate Account ID EQUALS 001w000001Zpeqw

Assigned Territories

Action	Territory	Parent Territory	Apply to Child Territories
Remove	EIFFAGE SA	Standard - Strategic	<input type="checkbox"/>

< Back To Top

Always show me [fewer](#) / [more](#) records per related list

The upload sequence

In some companies, you have hundreds of named accounts. In this case, you want to use the data loader to avoid any manual work.

You already have:

- 1. **User** records

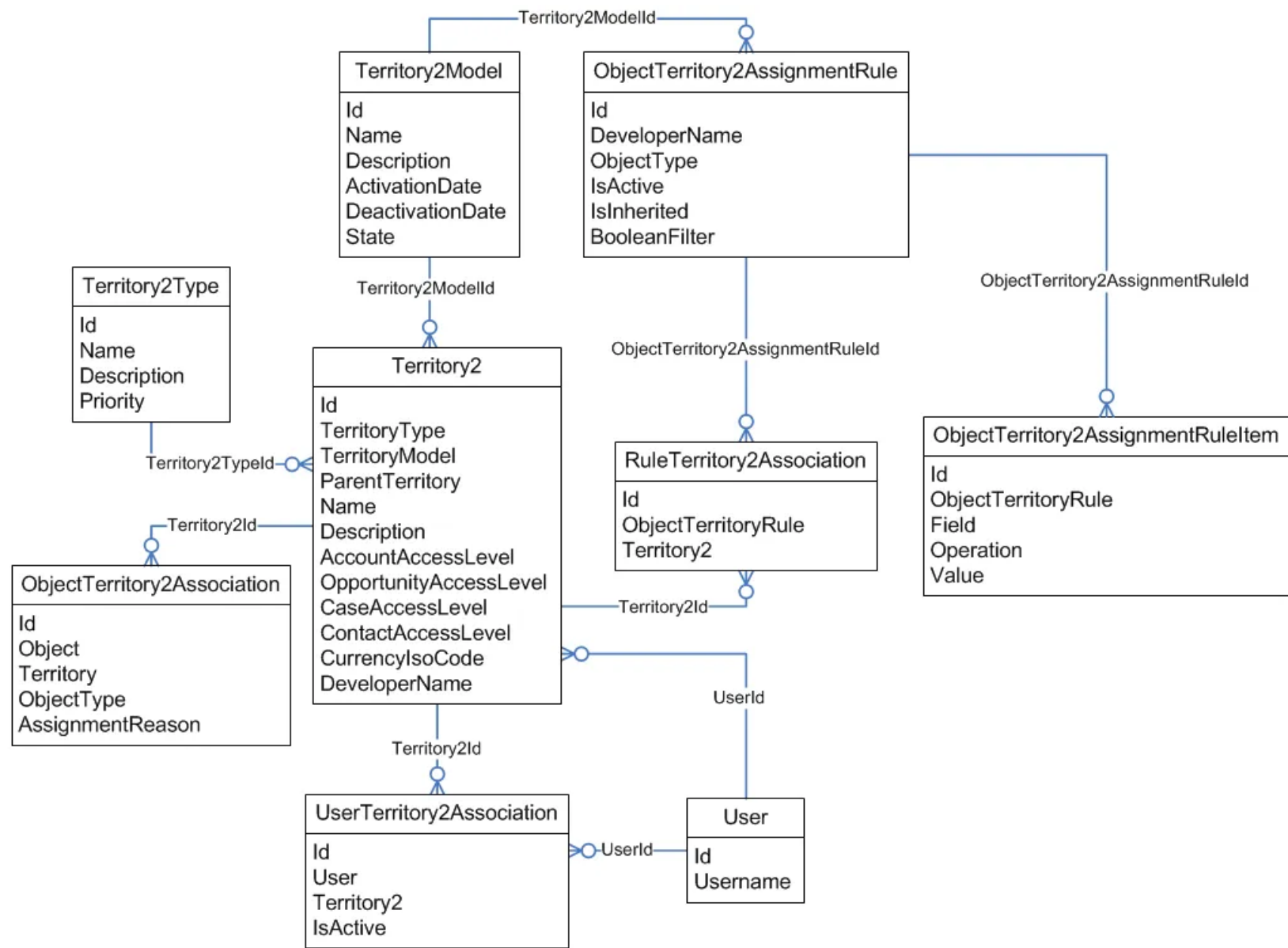
Manually, create

- 1. **Territory2Model** records, usually 1 per fiscal year
- 2. **Territory2Type** records

Then with the Data Loader, upload

- 1. **ObjectTerritory2AssignmentRule** records
- 2. **Territory2** records
- 3. **ObjectTerritory2AssignmentRuleItem** records
- 4. **RuleTerritory2Association** records
- 5. **UserTerritory2Association** records

The **ObjectTerritory2Association** is the table that associate automatically an account with one or multiple territories. It's updated when you "Run Assignment Rules".



How to update these territories?

The most common question will be “How to I update these territories”, as managers used to update their territories in their Excel files. There is a permission that will allow a user to update territories. If needed, I recommend creating a permission set named “Manage Territories”, and add the “Manage Territories” permission under “App Permission”. However, I recommend the following process to update territories:

If you want to update the teams:

- Extract the UserTerritory2Association table as a CSV
 - From the Data Loader or Workbench, run this query
 - “Select id,UserId,User.Name,Territory2Id, Territory2.Name,RoleInTerritory2 from UserTerritory2Association”
- Send the CSV to the manager who needs to update the territories
- Update the UserTerritory2Association table according to the modifications done
 - Use the Data Loader for example.

We can update most settings with similar processes to add strategic accounts to the list. It doesn’t have to be a manual process at all.

Advanced Settings for Sales Operations

As most named account territories rules are based on the account at the top of the hierarchy, we don’t want our Salesforce users to accidentally add a parent account on the account we used to define the territory. For this reason, I’ve added a validation rule on the account object, which prevent a user from adding a parent account on an account where the checkbox “Block Ultimate Account” is selected.

Obviously, we must check the “Block Ultimate Account” checkbox for all strategic accounts.

Account Validation Rule

[Back to Account Validation Rules](#)

Validation Rule Detail				<div>EditClone</div>	
Rule Name	Block_Ultimate_Account			Active	<input checked="" type="checkbox"/>
Error Condition Formula	AND(Block_Ultimate_Account__c = TRUE, NOT(ISBLANK(ParentId)))				
Error Message	You can't add a parent account to an ultimate account when the checkbox "Block Ultimate Account" is checked.			Error Location	Parent Account
Description					
Created By	Nathan Prats, 5/14/2018 8:59 AM			Modified By	Nathan Prats, 7/19/2018 11:00 AM
				<div>EditClone</div>	

I also created a script to make sure that the owner of an account is the owner of all children accounts, as long as the “IsExcludedFromRealign” checkbox field is unchecked. It’s a good way to avoid unnecessary chats between sales reps and sales operations regarding account reassignments. It runs daily.

Also, the field level security settings for “IsExcludedFromRealign” and “Block Ultimate Account” fields must be “READ ONLY” for all profiles except Sales Operations profiles.

Here’s the scheduled apex class:

```
1 global class ReassignAccounts implements Schedulable {
2 global void execute(SchedulableContext ctx) {
3
4 Integer count = 1;
5
6 do {
7 List<Account> AccList = [SELECT Id,OwnerId,Parent.OwnerId
8 FROM Account
9 WHERE Account_Owner_Parent_Owner__c = TRUE
10 AND IsExcludedFromRealign = FALSE
11 AND IsDeleted = FALSE];
12 for(Account Acc : AccList){
13 Acc.OwnerId = Acc.Parent.OwnerId ;
14 }
15 update AccList;
16 System.debug(count);
17 count++;
18 } while (count < 5); }
19 }
```

Here’s the class test:

```
1  @isTest
2  private class ReassignAccountsTest {
3  @testSetup
4  static void setupTestData(){
5  test.startTest();
6  Account acc = new Account(Name = 'Name123');
7  Insert acc;
8  test.stopTest();
9  }
10
11 static testMethod void test_execute_UseCase1(){
12 List<Account> acc = [SELECT Id,Name from Account];
13 System.assertEquals(true,acc.size()>0);
14 ReassignAccounts obj01 = new ReassignAccounts ();
15 String sch = '0 00 1 3 * ?';
16 system.schedule('Test', sch, obj01);
17 }
18 }
```