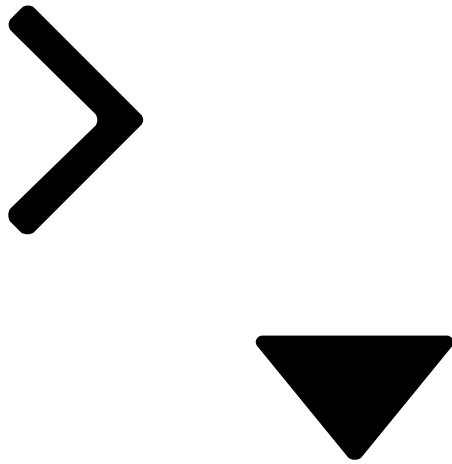


1. [Data Security](#)



2. [Create a Role Hierarchy](#)

Create a Role Hierarchy

Learning Objectives

After completing this unit, you'll be able to:

- Explain how a role hierarchy is different from an org chart.
- View and modify a role hierarchy.
- Create and assign roles to simplify access to records.

Create and Edit Roles

A role hierarchy works together with sharing settings to determine the levels of access users have to your Salesforce data. Users can access the data of all the users directly below them in the hierarchy.

Users who need to see a lot of data (such as the CEO, executives, or other management) often appear near the top of the hierarchy. But role hierarchies don't have to match your org chart. Each role in the hierarchy just represents a level of data access that a user or group of users needs.

- A manager always has access to the same data as his or her employees, regardless of the org-wide default settings.
- Users who tend to need access to the same types of records can be grouped together. We'll use these groups later when we talk about sharing rules.

Depending on your sharing settings, roles can control the level of visibility that users have into your Salesforce data. Users at any given role level can view, edit, and report on all data owned by or shared with users below them in the role hierarchy, unless your sharing model for an object specifies otherwise. Specifically, in the Organization-Wide Defaults related list, if the **Grant Access Using Hierarchies** option is disabled for a custom object, only the record owner and users granted access by the org-wide defaults receive access to the object's records.

Beyond setting the org-wide sharing defaults for each object, you can specify whether users have access to the data owned by or shared with their subordinates in the hierarchy. For example, the role hierarchy automatically grants record access to users above the record owner in the hierarchy. By default, the **Grant Access Using Hierarchies** option is enabled for all objects. It can only be changed for custom objects.

To control sharing access using hierarchies for any custom object, enter **Sharing Settings** in the Quick Find box and select **Sharing Settings**. In the Organization Wide Defaults section, click **Edit**. Deselect **Grant Access Using Hierarchies** if you want to prevent users from gaining automatic access to data owned by or shared with their subordinates in the hierarchies.

Define a Role Hierarchy

Implementing a role hierarchy in the platform is easy once you have an idea of what the hierarchy should look like. It's best to start with your company's org chart and then consolidate different job titles into single roles wherever possible.

For example, if a software development group has a staff software engineer and a junior software engineer, these positions can be consolidated into a single Software Engineer role in the hierarchy. Once that's done, you can get started defining the role hierarchy itself.

1. From Setup, use the Quick Find box to find **Roles**.



Tip

If you see an introductory splash page called Understanding Roles, click **Set Up Roles** at the bottom of the page to skip to the actual tool.

The default view for this page is the tree view, as indicated in the drop-down list on the far right side of the Role Hierarchy title bar. When creating a role hierarchy, it's probably easiest to stick with this or the list view, because they both make it easy to see how the roles all fit together in the hierarchy. The sorted list view is best if you know the name of a role that you want to find but aren't sure where it fits in the hierarchy, or if you don't want to click open all the tree nodes. For our purposes, we'll stick with the tree view.

When you first start defining a role hierarchy, the tree view displays a single placeholder node with the name of your org. From this point, we need to add the name of the role that is highest up in the hierarchy—in our case, the CEO.



Note

If you're building your app with a free Developer Edition or Trailhead Playground org, you may have a role hierarchy predefined as a sample. That's all right. You can still follow along and create some more roles.

2. Just under the company name, click **Add Role**.



Note

If the CEO role already exists, click **Edit**.

3. In the **Label** text box, enter **CEO**.

The **Role Name** text box autopopulates with CEO.

4. In the **This role reports to** text box, click the lookup icon  and click **Select** next to the name of your org.
By choosing the name of the org in the **This role reports to** text box, we're indicating that the CEO role is a top-level position in our role hierarchy and doesn't report to anyone.

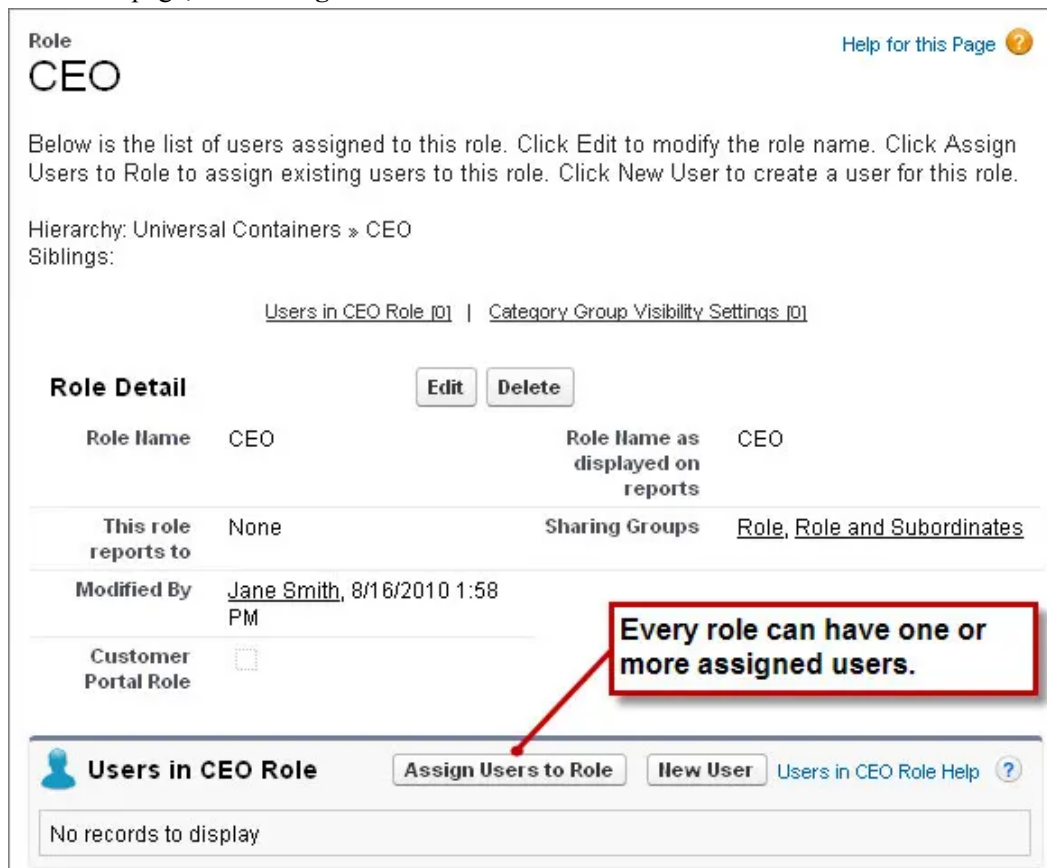
5. In the **Role Name as displayed on reports** text box, enter **CEO**.

This text is used in reports to indicate the name of a role. Since a long role name, like Vice President of Product Development, takes extra space in your report columns, we recommend using a short but readable abbreviation.

6. Leave any other options, such as **Opportunity Access**, set to their defaults, and save.

These access options don't have anything to do with our Recruiting app, and only appear if you have the org-wide defaults for a standard object set to a level more restrictive than **Public Read/Write**.

7. Now that you've created your first role, you can assign the appropriate user to it. Click **CEO**, and on the CEO role detail page, click **Assign Users to Role**.



Role
CEO

Below is the list of users assigned to this role. Click Edit to modify the role name. Click Assign Users to Role to assign existing users to this role. Click New User to create a user for this role.

Hierarchy: Universal Containers » CEO
Siblings:

[Users in CEO Role \[0\]](#) | [Category Group Visibility Settings \[0\]](#)

Role Detail [Edit](#) [Delete](#)

| | | | |
|-----------------------------|-------------------------------|--|---|
| Role Name | CEO | Role Name as displayed on reports | CEO |
| This role reports to | None | Sharing Groups | Role, Role and Subordinates |
| Modified By | Jane Smith, 8/16/2010 1:58 PM | | |
| Customer Portal Role | <input type="checkbox"/> | | |

Users in CEO Role [Assign Users to Role](#) [New User](#) [Users in CEO Role Help ?](#)

No records to display

8. In the **Available Users** drop-down list, select All Unassigned.

9. Choose a user from the list, and click **Add** to move her to the **Selected Users for CEO** list, then save.

If you return to the main Roles page from Setup, you can now see the new CEO role in the hierarchy. You can define the rest of the roles according to your role hierarchy diagram. There's no need to assign users to every role right away—you can do that later as you create the rest of your users and test out your app.

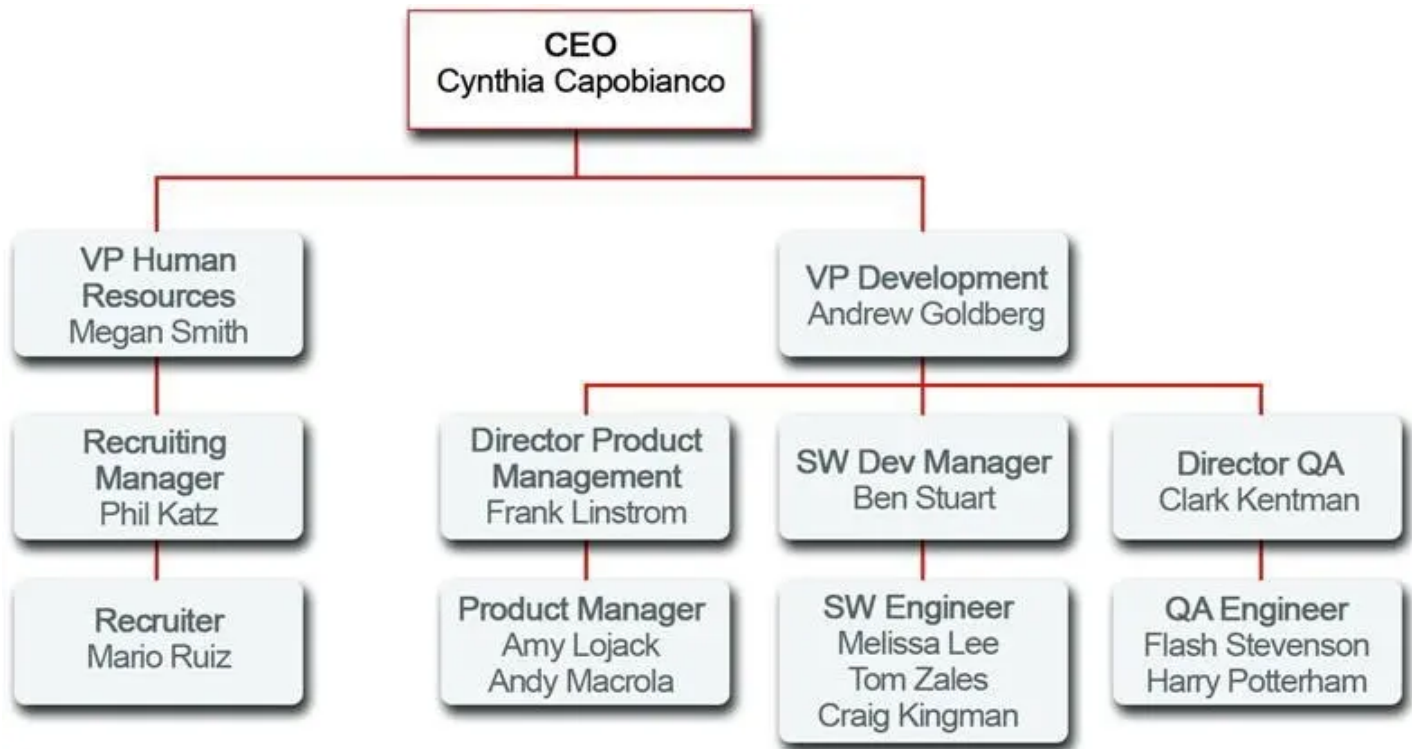


Tip

To speed up the process of adding a new role, click **Add Role** directly under the name of the role to which the new role should report. When you do this, the **This role reports to** text box is automatically filled in with the name of the appropriate role.

Role Hierarchy for the Recruiting App

Let's take a look at a branch of the role hierarchy for a fictional company that's using our Recruiting app. Remember, with the org-wide defaults we defined, hiring managers are allowed to view (but not create or update) all position, job posting, and employment website records, and to view and update other recruiting records they own. That doesn't make our app all that useful. However, once our role hierarchy is in place, our users can get to the data they need, and our app is off and running.



This role hierarchy automatically grants these kinds of record-level permissions:

- The CEO, Cynthia, can view and update every record that anyone else in the organization can view and update.
- The VP of Development, Andrew, can view and update any record that his managers or his managers' employees can view or update.
- The VP of Human Resources, Megan, can view and update any record that Phil, her recruiting manager, or Mario, Phil's recruiter, can view and update.
- The Recruiting Manager, Phil, can view and update any record that is owned by Mario, his recruiter.
- The Software Development manager, Ben, can view and update any record that is owned by Melissa, Tom, or Craig, his software engineers.
- The director of QA, Clark, can view and update any record that is owned by Flash or Harry, his QA engineers.

As you can see, the role hierarchy is a powerful way to open up data for people who need to see a lot of it!

With org-wide defaults and a role hierarchy in place, you're almost done with the record-level access permissions for the Recruiting app. All that's left is to share recruiting-related records between groups that appear in separate branches of the role hierarchy, and between peers in a single group. You can do both those things with a combination of sharing rules and manual sharing.

Resources

- [Controlling Access Using Hierarchies](#)
- [Security Implementation Guide](#)

Assessment Complete!

+500 points



Data Security

100%

Progress: 100%

Retake this Challenge

[View more modules](#)