

Scratch Orgs: The Heart of Package Development

If you're new to Salesforce, or new to the Salesforce package development model, you might not be familiar with scratch orgs, or you might assume that they're just another type of sandbox. If either of these scenarios apply to you, you've come to the right blog! We're covering the basic concepts behind scratch orgs and package development so you can see how they go hand in hand.

The basics of package development

To fully understand scratch orgs, you must know the basics of package development.

With traditional development methodologies, as an application evolves and expands with new features, so do the challenges that developers face. The application inevitably becomes an increasingly complex web of metadata and code. Its architecture, and the interdependencies of the features within it, become blurry. These conditions impair code management, increase the likelihood of code conflicts, and require a great deal of effort to keep the train from going off the rails. This is not to disparage traditional development methods. Many awesome apps have arisen from them, and they still have a place in software development.

Package development, however, is a game changer. It's designed to mitigate, if not overcome, the obstacles to agility, continuous development, and mental health. It replaces the unwieldy cluster of metadata and code with a modular architecture that can be versioned over time. Each package contains only the metadata and code for a distinct unit of functionality. Modular architecture is much easier to manage, and makes it much easier to identify and track interdependencies. It also sets the stage for a much more collaborative environment that allows developers to work on different features simultaneously with less worry of stepping on each other's toes.

Another major advantage of package development is that it forms a beautiful synergy with source-driven development. In source-driven development, the source of truth lives in your version control system rather than a sandbox or production org. This eliminates confusion about where to find the latest metadata and code. Moreover, managing changes and code conflicts is much easier to do in a version control system than it is in an org.

The advantages of package development are clear. And at the heart of its modular, source-driven life cycle, what do we find? We find scratch orgs. Within them packages are born and bred.

Meet the scratch org

A scratch org is an environment for developing and testing packages. However, before we go into more detail, let's clear up a common misconception.

Scratch orgs ≠ sandboxes

This can't be understated. Do not confuse scratch orgs with sandboxes. They can't be used to replicate your production environment, so they do not replace the need for sandboxes. In package development, scratch orgs provide the development environments for you to work your magic, but you still need sandboxes for user acceptance testing and product staging.

What makes scratch orgs so special?

Scratch orgs are ideal environments for developing and modifying packages. Here's why:

- **They're all yours.** Scratch orgs are personal development environments for you to run free and develop modular packages. You get to develop and test new features in peace and quiet.
- **They're derived from the truth.** Scratch orgs are brought to life by pushing package source from the version control system that serves as your source of truth.
- **They're lightweight.** Unlike sandboxes, scratch orgs start out as blank slates, so you can spin them up in minutes. This promotes agility and creativity. Have a great idea for a package but you're not sure if it'll work? Just spin up a scratch org and try it out. If things don't work out, you can destroy the scratch org as quickly as you created it.
- **They're configurable.** You can configure scratch orgs to include the specific features, licenses, and permissions that are needed for the package that you're developing.
- **They're ephemeral.** The lifespan of a scratch org can't exceed 30 days, so it's usually good only for the length of a single agile sprint, which is typically between two to four weeks. The limited lifespan reinforces the idea of frequently synchronizing your scratch orgs with source control, keeping you true to the practice of source-driven development.
- **They have source tracking.** You can track all the changes that you make to a scratch org through the Salesforce CLI.

As you can see, scratch orgs have a special place in package development. By their very nature, scratch orgs encourage modularity, synchronization, and agility. They bring out the best in everyone.

Now that you know the basics of scratch orgs and package development, take advantage of our easy hands-on training in the following Trailhead modules:

[Package Development Model](#)

[Quick Start: Salesforce DX](#)

[App Development with Salesforce DX](#)