

Podcast Script: Building Sugar

Welcome to this 60-second Sugar snippet. My name is Susie Williams. SugarCRM is building out and expanding here at our headquarters in Cupertino, California, so it's fitting that today's topic is about the Sugar build process.

We are often asked about how Sugar builds Sugar Open Source, Professional and Enterprise editions. The crux of the question actually gets to a misimpression out there in Sugarland that Sugar maintains three separate code branches, one each for each Sugar edition.

Au contraire!!

In fact, every Sugar edition stems from the exact same source code pool, or "trunk" as it's referred to by our engineers. The trunk source gives Sugar engineers access to every single line of Sugar code regardless of edition. On the other hand, Sugar Open Source developers out there in Sugarland have visibility only to the Sugar Open Source edition code.

Likewise, Sugar Professional licensees have visibility to Sugar Professional and Open Source code, but not Enterprise code. Sugar Enterprise licensees, perched at the top of the Sugar edition food chain, have visibility to Enterprise, Professional and Open Source code. But even they don't see everything that a Sugar engineer can see as we'll discover in a moment.

So, how does Sugar carve out a single source trunk to cover all editions? Through a patented process called "tagging", engineers will identify blocks of code designated for Professional and Enterprise edition functionality. Remaining source, the "untagged" source, will be found in the Open Source edition. Historically, about 80% of the Sugar trunk has been devoted to Sugar Open Source. The Sugar development build system reads in the entire source trunk, identifies and filters the PRO or PRO/ENT tags, and yields three separate edition ZIP files. It's also at this time that we insert the appropriate license headers and do about a million other things as well, including filtering out INTERNAL tags.

Internal tags? What are those? Remember before when we said that Sugar Enterprise code, even though it's at the top of the food chain, still differs from the trunk source? Well, the difference is the INTERNAL tags used by Sugar engineering to isolate code used for diagnostic purposes — code that never makes it into production Sugar code, regardless of edition.

One other nugget for you: have you ever noticed blank lines in your source code? Those are added by us when we strip out code that is not destined for that version. Putting in blank lines lets us relate line numbers in error messages between editions. This allows all of the Sugar developers around the world to share line numbers when discussing issues.

So there you have it. One trunk, three editions, and one very sophisticated build system.

And that's your 60-second Sugar snippet. Goodbye!