Question
[John Murray](#) · Mar 15, 2016

Managing source code across Caché and Ensemble version upgrades

One aspect of source code management is how to deal with the almost inevitable need to upgrade your Caché or Ensemble platform to a newer InterSystems release.

It's long been my experience that InterSystems does a very good job of maintaining backward compatibility. Code that works correctly on, say, 2012.1 is very likely to work correctly on, say, 2015.2 without any modification.

I'd guess that few sites are in a position to take a "big bang" approach and upgrade all of their instances (e.g. dev, test, live, DR) at the same time. So how do we deal with having different instances running different versions?

If we move code between instances by exporting and importing XML files, we can take advantage of this undertaking from the InterSystems documentation:

> Unless specified in the upgrade notes for a specific release, exported XML files can be imported into later releases.
>
> The reverse is NOT true. Later versions could be using new features not present in earlier versions, and could have a different class dictionary version number which alters how these classes are stored internally that cannot be undone when importing to a previous version.

In other words, if our code originates in instance A, then moves to B, then finally to C, we should upgrade C first, followed by B, and finally A.

If our code flow is more complex, our upgrade strategy needs to be more sophisticated. Where code can move freely between A, B and C perhaps we need to add a fourth instance D and only allow code to flow into D, never out. Then we can upgrade D to a newer InterSystems release and test our application there. Once verified on D, we might upgrade A, B and C all at the same time. Or spread those upgrades across a relatively short period, disallowing "reverse flow" of our code during the interim period.

Do you have any strategies or experiences to share on this topic?

[#Caché](#) [#Ensemble](#)