
Subject: Re: Optional serialization techniques
Posted by [chickenk](#) on Fri, 25 Feb 2011 10:16:46 GMT
[View Forum Message](#) <> [Reply to Message](#)

Maybe what you search for is some kind of migration procedure so that your 1.0 serialized objects could be automatically migrated to 1.1 format before loading.

The migration layer is still to be done, but that would mean no code change in your config loader, just make it deserialize 1.1 objects. the additional code would be isolated.

If you want to keep your original 1.0 objects, you can save the 1.1 migrated file in the same place with a filename change so that both are available then. And when you detect a 1.0 version, you try to find or generate the 1.1 version.

that would allow an automatic upgrade of saved files for your users. Then, the migration code would simply include the default values for a specific migration.

A simple migration class would only add the required changes between 2 specific version. Then you can chain the migrations if several versions appeared.

For example, a user has 1.0 format and upgrades it software where the 1.2 version of the data format is used. Before loading, the 1.0 version is detected, and then:

1.0 --> migrate_100_to_110 --> 1.1 --> migrate_110_to_120 --> 1.2

This migration system is used by some existing software such as Ruby on Rails (which gave me this idea for you).

Not sure it can apply, but that could be something to dig into, and maybe a small bazaar package in sight?