

EOF-Using EOF-Multiple Stacks or Instances

Overview

Anjo Krank

This is what the ERXObjectStoreCoordinatorPool does. You need to create each EC via the ERXEC factory to use it. Basically it assigns each session an OSC from a pool (more info is in the docs and the code). This does not help you when you have long-running statements going on, as there still sessions blocked - albeit not so many.

If you have long-running stuff, then you should create an OSC especially for them and use a long-running page.

Synchronizing Multiple EOF Stacks

The problem is synchronizing multiple EOF stacks is essentially the same as synchronizing multiple instances of your WebObjects application. Typically the same tricks apply to both.

WireHose

- [Wirehose Change Notification](#)
- [ERChangeNotificationJMS](#)
- [ERXObjectStoreCoordinatorSynchronizer](#)

Christian Pekeler

I prefer architectures where the database is the private store of one application (or a set of related applications built using the same technology - WO/EOF in our case). So if you need a different kind of application to change data, let it go through the application that owns the DB instead of giving it direct access to the DB.

Just recently I was involved in a project that heavily relied on DB triggers. We created a new WO application called CheckInProcessor. It listened on a port for new data (using a proprietary protocol). The external app that used to write new data directly to the DB was changed to send the data to the CheckInProcessor. The CheckInProcessor writes that new data to the DB, performs tasks equivalent to the old triggers, and notified the other WO apps of any changes (using WONDER's EOUpdateNotification framework).