**TECHNICAL MEMORANDUM**

ASM EA Cookbook

|  |  |
| --- | --- |
| **PREPARED FOR:** | BES PRGFX Project |
| **PREPARED BY:** | D.F. Cautley |
| **COPIES:** |  |
| **DATE:** | February 12, 2010 |

# Getting Started

We assume you have EA up and running.

1. If you haven’t already, Clone the Google Code repository to some repository location on your local disk. This will get the baseline database and all other PRGFX code.
2. Alternately, if you are on the City network, you may:
   1. Create an empty EA project. The template doesn’t matter is it will be overwritten.
   2. Run Tools > Data Management > Project Transfer and transfer the Gold Copy SQL database to your local database. {need connection info for this server}

This won’t get you the rest of the code you need, so you will want to do step 1 in any case.

1. Make a copy of this file to some other working area. You don’t want to work directly in your controlled area, as you might inadvertently overwrite the baseline EA database.
2. Connect EA to this database and explore enough to be satisfied that it’s working.
3. The various XML export / import files will be a part of the cloned Hg repository.

At this point you should have an environment that looks something like figure 1.



Import / Export

Figure -- Overall Working Environment

# Making Changes

Figure -- Persistent Data Packages

Lets say you want to work on the persistent objects, which we used to call the database model.

In the V0.1 version, there are two packages… See figure 2. It’s likely that there will eventually be more but this will work for the example.

Here come the steps.

1. Using TortoiseHg, be sure that you have the latest version of the package XML file on your workstation. Do this by executing an ‘update to version’ command for the version you want.
2. In EA, with your copy of the project open, right click on the package you want to work on and select Import/Export > Import Package from XMI file. XMI files are particularly formatted XML.
3. The entire package in your EA model will be replaced with the most recent version from the master repository.
4. Consider if there are any other packages you need to update as well. If there are many you may consider downloading an entire baseline database file.
5. Make the changes you deem necessary.
6. When the changes are done, export your package to the working folder. TortoiseHg will now flag this file as changed.
7. Commit the modified file. This updates your local Hg repository.
8. Synchronize your local repository with the Google code repository:
   1. Pull any Google hosted changes. The ‘after pull’ action should be “update”.
   2. Resolve / merge the change sets, if any
   3. Push your version up to Google code.
9. If appropriate, notify other team members and/or the ASM lead of the changes.

These procedures mimic what EA does with its version control mechanism turned on; only we’re doing it by hand.