

Bankmaster, ERP Interface

Objective:

This system was developed for a large Bank in South Africa. The bank uses a core banking product which runs on a mainframe system. All the transactions of the day are written to a flat file at the end of day. The bank has another ERP system which captures these data and reconciles the balances. There was a need for an interface which will take the data from the source system and transform it and present it to the destination system and the destination system does not have to do any transformation. The way the bank accounts are represented in the Core Banking system is different from the way the accounts are represented in the ERP system. So the need was to develop an interface system which understands the formats of the source and destination system and facilitate seamless transition of the data from one system to another.

Process followed:

To achieve this objective the most important factor was to understand the way the core banking system and the ERP systems operate and the way they represent the various account types. This is the basis of the main business logic which is applied in the transformation of the accounts from one system to another. The functional consultants were engaged to understand and map this to some business rules. The system was built in such a way that it could pick the files from the source system at a scheduled time, apply the transformation and produce an output file in the format which the destination system understands. Then the output file is pushed into the destination system at another scheduled time. So scheduler was built into the interface system as well.

Key Benefits

- ✓ This interface ensured that there was no modification done on the core banking system or the ERP system for the integration, which would have impacted other systems running in the bank.
- ✓ These tools helped the bank to reduce the complexities around the various systems that were being used earlier to transform the data.
- ✓ This system has an inbuilt scheduler, so it requires no human intervention and the total process could be automated.

Technologies Used

- ✓ Java was used as the platform for building this tool.
- ✓ XML was used to define the source and destination systems.
- ✓ Multithreading was used extensively to help the system process parallel requests. This improved the system performance.
- ✓ Java Reflection was used to build the generic capability into the system.
- ✓ Quartz Scheduler was used for the scheduling of jobs.