

## Business Process Automation and Modularity in Transactions

### Definition of business processes

In the context of this article, a business process is a series of stages thro which a transaction passes either within the same work group or across work groups. The process has a well defined begin and an end. The process consists of data and business logic which works w r t the data. At each stage a Transactor (role) participates in the process to alter the state of the data in the process. Even Approval or Rejection of the data in the process at a stage is considered as a change in state (though no explicit data change occurs in the process). The business logic could use the data present in the instance of the process and data in an associated database.

All such business processes are pre-defined in the system.

### Forking of a process by another process

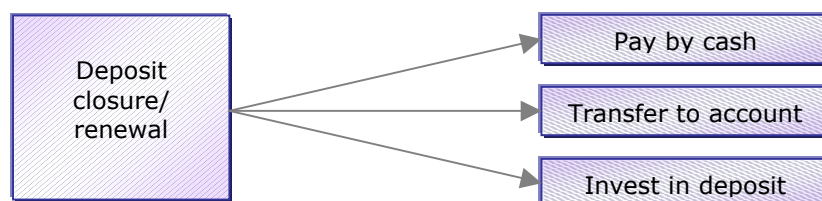
In Datanet's BPRO, we have a special capability called `forking`. Using this an instance of one process can initiate an instance of another process, pass or set up the data needed by that process and let that forked process flow according to its own business and routing logic. The forking process is called the Parent Process and the forked process is called the Child Process.

This enables excellent modularity and granularity control in defining processes. Combining them by use of forking makes it possible to build complex processes easily and quickly. Maintenance of the processes becomes easy because change in business rules are incorporated in the individual processes and automatically take effect when combined process is executed.

### Examples of forking

**1. Handling complexity:** This example is taken from a banking scenario. It pertains to Fixed Deposit maturity/ renewal. When a customers deposit account matures, he /she has the option of splitting the total amount by saying that one part be paid by cash, one part to be transferred to an existing savings account and the balance be invested in another fresh Deposit account.

Typically a bank has standard processes for paying cash, for transferring to another account and to open a deposit account. A new process is built for Deposit Account closure / renewal which will accept the customer's options and merely '**fork**' the three pre-defined standard processes, and transfer relevant data instead of creating a new complex process.



**2. Achieving productivity:** Consider a situation where data has to be entered for several instruments such as cheques and then each instrument's credit value has to flow separately either within the work group or across work groups. This independent flow is required because the fate of each instrument (cleared or bounced) is determined independently. Let us assume that such a 'carry credit value to a destination' is a well defined process. However during data entry we need to verify certain values and totals so that only if they tally, the details of each instrument should be entered and sent.

This is achieved by designing a main, non-flow enabled process which accepts summary or key data of each instrument, does the required validations and at the end of a batch of data, if no errors are found, starts 'forking' a series of 'carry credit' process instances, transfers relevant data, collects additional data directly on the new instances and lets these instances flow to their destinations.

Now the individual instances can be tracked and processed separately.

**3. Controlled Processes:** There are situations when a process has to be instantiated from another process, without allowing any operator or human intervention on that process. The main process may permit human operator interaction, upon completing which a controlled process is forked, the relevant data transferred and 'sent' to the next transactor automatically. Such a controlled process will also be a well defined business process.

*An example is the 'Reconciliation Reporting' transaction in multi branch banking transaction. When a inter branch funds transfer originates or terminates at a branch, an IBR reporting form is automatically forked with transaction details and sent to the reconciliation center. The data in such a transaction should not be edited by any transactor.*

### BPRO's features of forking

In BPRO forking is labeled as **fat** – **fork a transaction**. A process can fork either a single instance of another process or it can set up group forking. In group forking a group begin and end is indicated and BPRO manages and tracks these process instances as a group. This facility is used when a parent process forks several child processes, but they should not become available to the next transactor until all forms in a group are forked. This will be necessitated by the fact that the next transactor should make his/her decision only when all related processes have arrived in his/her 'in tray'.

The application developer or user can switch control between the parent and child processes. However only one child process instance can be 'open' along with the parent process.

Some of the related key BAPIs are:

Bpro\_fat  
 Bpro\_fat\_copy -- forks a transaction instance and copies specified fields from parent to child process  
 Bpro\_fat\_with\_message  
 Bpro\_get\_forked\_form\_nos  
 Bpro\_is\_form\_forked  
 Bpro\_switch\_form\_control .. etc.,

-- o o O o o --