

Programming transactions in Natural and 3GL

- Programming Adabas transactions in Natural
 - Programming 3GL Adabas transactions
-

Programming Adabas transactions in Natural

Here's a simple Adabas transaction:

1.	Modify data in Adabas database A
2.	Issue END TRANSACTION

This is indeed extremely simple. However, there is only one database that is modified so let's see what happens when multiple databases are involved:

1.	Modify data in Adabas database A
2.	Modify data in Adabas database B
3.	Issue END TRANSACTION

Programming 3GL Adabas transactions

Here's how the previous logic looks when it is implemented in 3GL:

1.	Modify data in Adabas database A
2.	Modify data in Adabas database B
3.	Issue ET command to Adabas database A
4.	Issue ET command to Adabas database B

The difference between programming transactions in Natural versus 3GL is that Natural makes it very simple (as it is supposed to do). In Natural you code a single `END TRANSACTION` statement no matter how many databases are modified. In 3GL programming you code specific `ET` commands for each of the modified databases.

This difference isn't difficult to understand, and appears to make sense. However, it highlights an extremely important area in transaction programming that we call *well-formed distributed transactions*.