Transaction Processing

Many database applications, particularly those involving updating multiple rows at one time, are transaction oriented. For example, if money is transferred from a checking account to a savings account, this transaction involves updating the checking account table to reflect the withdrawal, and updating the savings account table to reflect the deposit. These updates should take place as a unit - either ***both*** of the updates occur, or ***neither*** occur. If only the checking account is updated without the savings account update occurring (or vice versa), the database is in an ***inconsistent*** state. In the real world, this could result in an account containing too much money, or an account indicating an overdrawn condition where, in fact, the account is not overdrawn.

Transaction processing prevents this situation by treating both the withdrawal and the deposit as part of the same transaction. Either both processes occur, or neither occur. In OpenInsight, the process is as follows:

1. Start transaction processing using the Transact subroutine.
2. Identify the tables to be used in the transaction, using the Control On routine.
3. Process the data changes. Instead of immediately committing the changes to the database, OpenInsight saves them to a temporary volume (the **TRANSACT** volume).
4. If no errors occur, commit the changes to the database. If errors occur, rollback the database to its state before the transaction was processed. In either case, use the Transact subroutine.
5. When tables do not require transaction processing, identify them and use the Control Off routine.

The next topic is a programming example that illustrates the process.

Sample Transaction Processing Subroutine