

*A couple of points before you begin*

**Author: Bill Ennis Introduction**

Oracle 9i introduced a new feature named Merge that excites anyone that has had to work on a database load process that needed to deal with the addition of new records and modification of existing records. Before Oracle 9i this type of resulted in developing code in PL/SQL, Pro-C, OCI, or some other Oracle database API that first checked for the existence of the record via a unique key lookup. While this is not terribly difficult the lines of code do start to add up after a while. The Merge feature offers the ability to combine the new data with the target table in one SQL statement. You could load the new data into a transient type table with SQL Loader and then execute an SQL statement without writing any code!

## Issues to Consider

While this feature definitely has some attractiveness to it there are a couple of things you need to consider before jumping on the Merge bandwagon. The first thing you must remember is the Merge is an SQL statement. SQL statements are an atomic unit with respect to the database transaction that they are a part of. The statement either succeeds or fails as a unit. What complications can this bring? Well, if you run across a constraint violation on any one row nothing, nada, zilch... will get loaded. Also, if the data set you are working on is extremely large you could run out of rollback or undo space. There is one other thing I have noticed while using the Merge feature that I would like to point out. The structure of a merge statement includes an insert leg, an update leg, and also a leg that contains the logic that determines if a row will be considered an insert or an update. Well, again all rows pertaining to a merge are part of the same transaction. The insert/update determination is made by Merge checking to see if the rows existed at the start of the Merge statement. This will cause complications if you have a duplicate (as determined by your Merge key) tuple in the new set. As a simple example lets assume we are attempting to use Merge on a client list and that the Merge key will be on a unique value named the client id. Let us also assume that the source of our data has mistakenly published a client id twice and that this client is new. If we have defined an integrity constraint on the client id and we try and insert it twice guess what will

happen? The constraint check will fail the second time around, the Merge statement will fail, and no records will have been Merged.

## Summary

Am I suggesting that you never use the new Merge statement? Absolutely not! What I am suggesting is that you use Merge with the appropriate caution. I have decided to use Merge on data only after it has gone through a thorough scrubbing process and when I can assure that the issues that I have mentioned have already been addressed.