

## **YLA 10 Shots of DB2 Performance 3 Extra Tips**

This document is an amendment to the YLA podcast "10 Shots of DB2 Performance" dated 1/21/2009 on the DB2EXPERT.COM web site.

### **11 – Use RELEASE(DEALLOCATE) for Performance**

This BIND parameter allows for certain efficiencies in local applications. It holds table space and table level locks across commits. It retains information to support index lookaside and sequential detection across commits. It retains internal efficient code paths called SPROCS, UPROCS, and IPROCS. These features can significantly reduce CPU consumption and is great for high volume batch and online transactions. However, you should be aware that it is more difficult to get a drain on database objects with RELEASE(DEALLOCATE). So, if it is important to stop or REORG objects while they are being accessed this might not be the right option for you. Keep in mind it's a trade-off between performance and availability.

### **12 – Use a VDWQT of 0,40 to Trickle Buffer Writes**

We have a lot more memory these days and a lot bigger buffer pools. However, many times we don't change the default buffer pool parameters. With huge buffer pools normal defaults can represent a large amount of data. If you have highly active updates you could end up initiating large volumes of write I/O's at system checkpoints. Using this setting will allow DB2 to write 32 pages once 40 have been updated for an object. This can help balance write I/O and give you more even DASD response rates, especially when you don't have parallel access volume (PAV) support.

### **13 – Use Database Enforced RI versus Application Enforced for Non-Code Tables**

Database enforced RI works well as long as you have the proper indexes defined. When inserting or deleting data do you want to make several database calls to check relationships or one to allow the database to do the checking? In high volume environments each extra trip to the database engine means more money. You can significantly reduce the calls by using database enforced RI.