CHAPTER 9

Advanced Java Programming in SAP
The final chapter in this book is devoted to advanced Java and JCo programming techniques that enhance existing applications and offer new functionality in the SAP environment. Although the JCo API can be used strictly for basic connectivity into the SAP system, it also offers extensive optimization and framework tools.

This chapter looks at the following:

- Optimizing with native JCo connection pools
- Building a JCo server
- Developing custom Java tags

Connection pooling in JCo is based on a standard design pattern that you would normally implement by hand. The first section details how you can add JCo's easy-to-use pooling mechanism into your existing application development. JCo also provides a server interface API that provides outbound connectivity from within the SAP system. In the second section, I detail this technique using a combined Java/ABAP tutorial. You have already used the JavaServer Page Standard Tag Library (JSTL) to simplify Web page development and to hide complex Java logic and are likely interested in building your own custom tags. The last section demonstrates a custom tag that provides default behavior in the material catalog JSP.

Introducing JCo Connection Pools

Chapter 6 demonstrated how you could use a shared context to let users access a common instance of the InterfaceCaller class. Limiting the number of InterfaceCaller instances created by the Java Virtual Machine (JVM) allowed you to optimize usage of Java server resources during times of peak traffic. Web applications that connect to SAP via JCo pose a similar threat when you do not restrain the number of JCo.Client instances used by these applications to an acceptable limit. Fortunately, the JCo API provides a connection pooling mechanism that allows your Java applications to better manage connections to the SAP application server.

If you plan to provide generic access to certain functions in SAP, using connection pools is an excellent way to increase performance. Each unique or named user connection your JCo application creates uses up resources in the R/3 system. Both foreground or SAP GUI users and background users such as those accessing the system via RFC share these resources. Allowing an unrestricted number of named user connections via JCo can have a negative impact on SAP GUI users who are simultaneously accessing the system. A connection pool allows you to optimize this resource usage by relying on a single generic user connection shared by multiple instances of your JCo application.