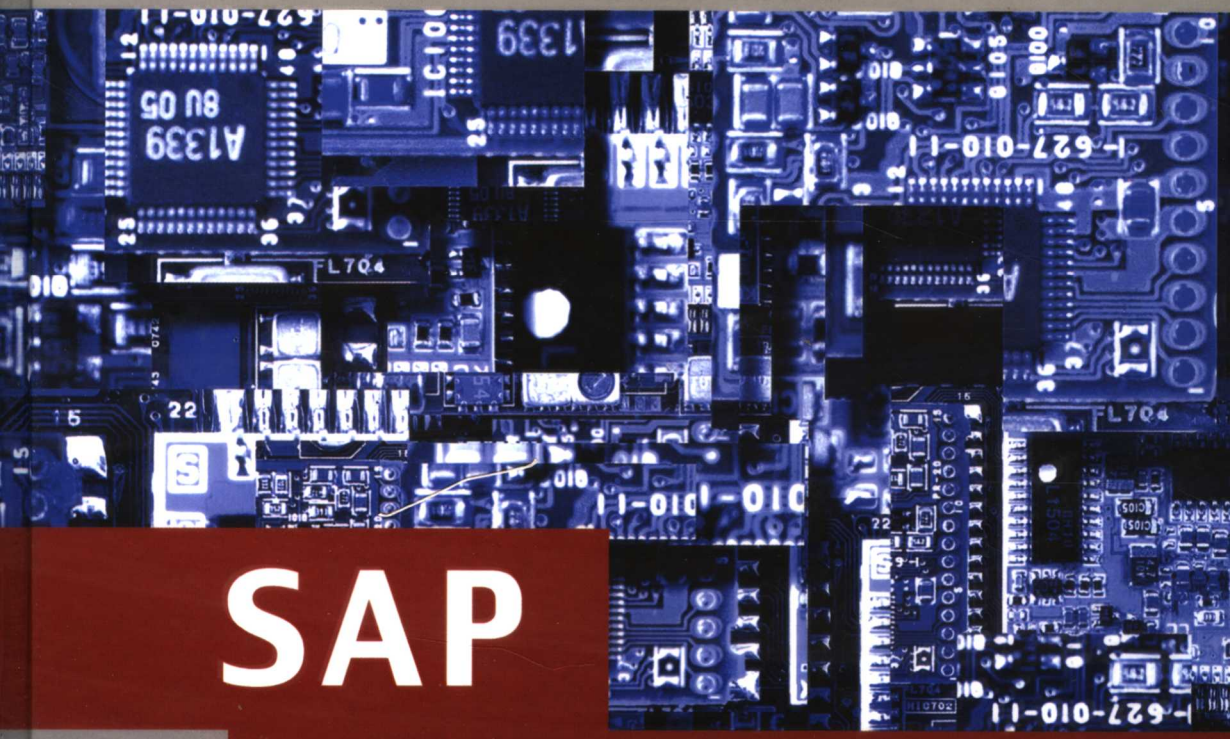
SAP NetWeaver 精要丛书

卡尔·凯斯勒(Karl Kessler)
彼得·蒂勒特(Peter Tillert)
帕纳尤特·多布林科(Panayot Dobrikov)
等著



SAP 应用服务器的JAVA编程（影印版）

JAVA Programming with the SAP Web Application Server

東方出版社

SAP PRESS

SAP NetWeaver 精要丛书

SAP 应用服务器的JAVA编程

JAVA Programming with the SAP Web Application Server(影印版)

卡尔·凯斯勒(Karl Kessler)
彼得·蒂勒特(Peter Tillert)
帕纳尤特·多布林科(Panayot Dobrikov)
等著

東方出版社
SAP PRESS

著作权合同登记号
图字:01-2005-3504号

责任编辑：吴秋淑
封面设计：左　涛

图书在版编目(CIP)数据

SAP 应用服务器的JAVA编程/凯斯勒　等著；
北京:东方出版社，2005.6
ISBN 7-5060-2228-1
Ⅰ.S … Ⅱ.凯 … Ⅲ.①计算机②软件 Ⅵ. F97
中国版本图书馆CIP数据核字(2005)第025787号

SAP 应用服务器的JAVA编程（影印版）

卡尔·凯斯勒(Karl Kessler)
彼得·蒂勒特(Peter Tillert)
帕纳尤特·多布林科(Panayot Dobrikov)
等著

出版发行　東方出版社
地　　址　北京朝阳门内大街166号　　邮政编码　100086
电　　话　010—82665724（编辑部）　　010—82665462（发行部）
　　010—82679079（生产部）
网　　址　http://www.erptraining.cn
经　　销　新华书店
印　　刷　山东新华印刷厂临沂厂
开　　本　787×1092毫米　1/16　　版　　次　2005年6月第1版
印　　张　32.50　　印　　次　2005年6月第1次
字　　数　600000　　定　　价　80.00元

简明目录

Contents

Preface

SAP NetWeaver is based on the SAP Web Application Server—the technological basis of all modern SAP solutions. Without the SAP Web Application Server (SAP Web AS), no Enterprise Portal, Exchange Infrastructure, Business Information Warehouse, or SAP enterprise software would be able to run. SAP Web AS is the platform on which applications can be developed and operated. The success of SAP's software depends largely on the efficiency and robustness of SAP Web AS.

As we reflect over the past decades, SAP's implementation of the application server at the end of the 1980s and the beginning of the 1990s was revolutionary. SAP's competitors—in so far as they existed at the time—were building two-tier applications. Some competitors moved the entire business logic to the frontend, creating the extremely cost-intensive *fat client*. This was difficult to scale as numbers of users increased, since every user process was assigned to a shadow process in the database. Other SAP competitors moved the business logic to stored procedures running directly in the database, but soon learned that with stored procedures an application couldn't be run on a database platform without recoding the entire application.

SAP's solution to this problem was to insert a separate tier between the frontend and the database: the *application tier*. This was when the application server—as we know it today—was born—even outside the context of SAP.

The application server from the days of R/3 still had all the features of a modern application server. It abstracted data from the underlying operating system and database platform, and optimized access to database resources using a sophisticated transaction and update concept. It provided a platform-independent programming language (ABAP) that was tailored to meet the requirements of professional enterprise software.

Many ideas from the world of Java, such as generating bytecode interpreted by a virtual machine, have been state of the art in the ABAP world for a long time. Another advantage of Java was that the development and runtime environments were interlinked. There's probably no other system in which you can perform all development work without ever having to leave the system. As the system compiles itself incrementally after changes, you no longer have the long generation times that were standard for classic C and Java development.

The swift success of the R/3 System on the market led to the application server being widely distributed. At the same time, the Internet and the World Wide Web (WWW) took off at an incredible speed. Proprietary worlds were no longer in demand; instead, people wanted open standards that were adopted by the Internet community. And the primary standard was the Java programming language. Even though its core wasn't designed for business applications, it was popular because it was simple and platform-independent. Consequently, Java became quickly widespread (largely because of the Internet).

But not all concepts from those early years had any chance of success: Downloading entire applications on request from the Internet onto simple devices still remains an illusion. Endeavors to define a Java-based application server that different software providers could implement were taken more seriously, although it wasn't until well-known providers (including SAP) actually began to implement the J2EE standard that Java had any real opportunity for success in the field of enterprise software.

However, implementing the standard came at a price. You had to implement many functions that were not imperative for business applications. Conversely, the J2EE standard could not sufficiently define many requirements essential for a business application, primarily the user interface, which is fundamental for business applications because an application's acceptance by end users depends largely on the user interface. This is where SAP's many years of experience with enterprise software came into play. Even in the earliest versions of SAP's software, the user dialog was the focal point. This tradition is currently supported by Web Dynpro for SAP Web AS. Web Dynpro represents an application's user interface running in the SAP Enterprise Portal and, as such, is often the starting point for a development project. Powerful tools such as the SAP NetWeaver Developer Studio (NWDS) help you to design your applications as prototypes, while Web Dynpro supports a highly declarative, model-oriented approach.

In addition to the user interface, elements and issues such as business logic, service-orientation, persistence, scalability, and maintainability are vitally important for the success of an application. A comprehensive chapter is therefore dedicated to each of these issues, starting with **Chapter 1**, which focuses on SAP Web Application Server's positioning within SAP NetWeaver and introduces the most important SAP NetWeaver components.