Release 3.3 Copyright ©1995 by NeXT Computer, Inc. All Rights Reserved.

Introduction

This book presents the Database Kit™, an object-oriented kit that helps you create client applications that communicate with industry-standard database servers such as those supplied by Oracle Corporation and Sybase, Inc. The Database Kit is just one of the many object-oriented kits included in NEXTSTEP™, but for many individual and corporate programmers it's the kit that promises the greatest productivity gains.

The Database Kit and its associated development tools let you rapidly create applications that communicate with commercial database servers. Since you build the application by selecting and interconnecting pre-designed programmatic objects, your programming burden is greatly reduced. In fact, you can create a simple database browser without any programming at all. Furthermore, the application you build is not tied to a particular brand of database server. Through the use of server-specific *adaptors* (more on this in the next chapter), the Database Kit lets your application work with any of a number of database serversÐwithout requiring you to change a line of code.

Superficially, the Database Kit may seem to provide the same benefits as competing 4GL tools. Although the Database Kit and 4GL tools attempt to solve the same kinds of problems, the approaches they takeÐand the consequences to you, the programmerÐare entirely different. NEXTSTEP and the Database Kit provide a much more flexible and extensible environment for building distributed client/server database applications than any 4GL product available today.

The primary advantages of the Database Kit derive from its object-oriented architecture. Object orientation makes programs more maintainable since the interfaces between the self-contained objects are well-defined. It's clear how your pieces fit with those provided by the Database Kit. By using the pre-designed and pre-tested objects defined in the Database Kit and other NEXTSTEP kitsÐor purchased from third-party object vendorsÐyou need to write far less code, making it easier to create reliable applications quickly.

NEXTSTEP objects bring more than reliability to an application. The years of development effort that has gone into creating and refining the NEXTSTEP kits endow your application with sophisticated functionality with little effort on your part. By virtue of being part of NEXTSTEP, your application will be integrated into an award-winning user environment in which even simple applications support printing, faxing, sound and image handling, cutting and pasting of various data types between applications, and access to system services such as electronic mail.

This book describes the architecture of the Database Kit and helps you get started using its various components. It describes software that's included with NEXTSTEP releases 3.0 through 3.2 and is part of a collection of manuals called the *NEXTSTEP Developer's Library*.

How this Manual is Organized

This manual is divided into ten chapters and one appendix:

- Chapter 1, ªOverview,º introduces the Database Kit and the associated components of the development environmentÐDBModeler, Interface Builder, and the Database Kit library. It also describes how a database adaptor links your application to a commercial database server.

- Chapter 2, ªEntity-Relationship Modeling,º describes the theoretical basis for the Database Kit's representation of data stored in the database server.

- Chapter 3, ªDatabase Models,º shows how, using the DBModeler application and the theory presented in Chapter 2, you create data models that your application can use to map server data to objects in your application.
- Chapter 4, ªCreating a Database Project,º walks you through the creation of a Database Kit application. In the process, Project Builder, Interface Builder, and several of the principal classes in the Database Kit are introduced.
- Chapter 5, ªDatabase Kit Classes and Protocols,º presents the major divisions of the Database Kit and details the classes that compose each part.
- Chapter 6, ªModel Emulationº examines the DBDatabase class and the DBProperties and DBEntities protocols as they define modeling objects.
- Chapter 7, ªConnecting to the Serverº briefly looks at the DBDatabase methods through which you connect your application to a database server.
- Chapter 8, ªData Storage, Retrieval, and Manipulationº explains the basic techniques of configuring data-storage objects, retrieving data from the server, manipulating data, and sending modified values back to the server.
- Chapter 9, ªAdvanced Record List Techniquesº pays attention to the finer points of data storage. In this chapter, data-storage objects are fine-tuned, and techniques such as joining tables and creating master-detail tables are demonstrated.
- Chapter 10, ªFetching and Saving Dataº looks at the ways that you can fine-tune the data-fetching mechanism, and explains how to validate modified data.
- Appendix A, ªAdaptor Default Parameters,º lists the default parameters that a user can set to alter certain aspects of the Oracle and Sybase adaptors.

Conventions

Where this manual discusses methods, functions, and other programming elements, it makes special use of bold and italic fonts. **Bold** denotes words or characters that are to be taken literally (typed as they appear). *Italic* denotes words that represent something else or can be varied. For example, the syntax:

```
Entities =
(
    ( entity1 ) ,
    ( entity2 ) ,
    ( entity3 ) ,
    ... ,
    ( entityN )
) ;
```

means that all elements in bold are to be taken literally, but that you can choose the name for the individual entities. *Italic* is also used to mark terms that are defined in the glossary.

As explained in Chapter 2, the names of entity, attribute, and relationship modeling objects are presented in a distinctive font, as Entity, attribute, and relationship.