# Portable Distributed Objects™ Release 3.0

*PDO.tiff ¬*

*Portable Distributed Objects seamlessly extends NEXTSTEP's industry leading object model to non-NEXTSTEP operating systems, providing an easy-to-use framework for building powerful client/server applications today.*

Using NEXTSTEP™ to build custom applications is a key factor in gaining a competitive advantage. Portable Distributed Objects™±PDO™±extends NEXTSTEP's dynamic object model and messaging architecture to industry-standard operating systems, scaling that advantage from the desktop to the data center.

PDO is the industry's first product to provide a heterogeneous client/server framework based on objects. NEXTSTEP applications consist of objects that encapsulate data and behavior, and communicate with each other through NEXTSTEP's messaging system. Objects are building blocks that can be executed on either the client or server hardware. PDO makes it possible to deploy objects on non-NEXTSTEP server machines. This architecture lends itself exceptionally well to a client/server environment; objects can be deployed anywhere across the enterprise, wherever most appropriate for the task.

The Enterprise Objects™ Framework supplies data access services that allow server objects to persist in industry-standard relational databases. The pairing of the Enterprise Objects Framework with PDO creates a distributed computing environment that provides an infinitely flexible choice of object deployment strategies. Persistent objects can be distributed across high-performance compute and database servers throughout the enterprise.

PDO extends the same seamless object framework that NEXTSTEP developers already use for local and distributed objects. NEXTSTEP developers don't need to learn new programming tools or techniques-PDO's tools and object frameworks are a subset of those in NEXTSTEP. The details of network communication can be ignored since PDO encapsulates low-level network protocols, making messaging a remote object as straightforward as messaging a local object.

PDO server objects transparently inter-communicate with other NEXTSTEP, OpenStep™ and PDO objects.

# FEATURES AND BENEFITS

**Operating system independence**
PDO is available on a variety of workstation and server class machines, guaranteeing deployment flexibility and independence from hardware and operating system vendors. PDO objects can be developed and deployed on HP-UX, SunOS, Solaris, and Digital UNIX operating systems.

**Transparent distribution**

PDO objects are location transparent; client applications message all objects identically, whether they are local, across the street, or across the country. Applications can be developed without concern for where objects will be deployed. As business needs change, the location of objects can change without any impact to existing applications.

**Three-tiered, scalable, client/server architecture**
Using PDO and the Enterprise Objects Framework developers can clearly separate the user interface, business objects and database components of their applications. Business objects can be reused by all applications and can be deployed on the most appropriate client, compute server, or database server, enabling scalability across the enterprise.

**Distributed execution**
PDO objects can be located (and moved) where they are needed and where they make best use of computing resources, while always appearing local to clients. PDO objects can be local to a process, distributed among processes on one machine, distributed among machines on the same local area network, and distributed among machines anywhere on the Internet.

**Dynamic object model**
Applications based on the PDO dynamic object model can easily accommodate new data types, new methods, and new objects at runtime without recoding, recompiling, or restarting. PDO objects are flexible and can adapt to changes that occur after their classes are compiled. Client applications need not be modified if the server object's underlying implementation is reworked.

**Object libraries**
PDO provides a useful set of NEXTSTEP objects for server applications on non-NEXTSTEP operating systems. Object libraries support object persistence, garbage collection, memory management, and string, array, and dictionary handling. Like all NEXTSTEP objects, their behavior can be modified or extended by subclassing their Objective C classes.

**Integrated compiler**
PDO's ObjectiveC++ compiler allows developers to write code that integrates Objective C, C++, and ANSI C in one application. Objective C and C++ statements can be combined in the same source file, and a C++ object can message an Objective C object, and vice versa. Developers do not have to reimplement C++ code that's already working.

**Graphical development tools**
Developers can build, debug, and maintain projects on a PDO server from a NEXTSTEP client. Running on the server, Portable BuildServer, NEXTSTEP-compatible make, the debugger, and the compiler suite can be executed using the client's graphical tool set.

**Persistent objects**
In conjunction with the Enterprise Objects Framework, developers can construct reusable business objects that combine business logic with persistent data stored in industry-standard relational databases. Business objects developed with the Enterprise Objects Framework are insulated from the semantics of how data is stored and retrieved. Changes to the database schema are easily accommodated with a simple remapping that propagates enterprise-wide to all objects using the schema. Applications can also be migrated to different databases without recoding any business logic.

**OpenStep interoperability**
PDO, NEXTSTEP, and OpenStep objects are based on the same Distributed Objects Framework and can seamlessly intercommunicate. With PDO, developers can today begin to develop OpenStep applications that can easily be deployed on OpenStep for Solaris, OpenStep for Digital UNIX, OpenStep for Windows, and all future OpenStep implementations.

**Legacy software integration**
PDO uses native linker and calling conventions allowing PDO applications to link in existing C libraries. A PDO server application can "serve" NEXTSTEP, OpenStep, and PDO clients as well as X Windows clients.

**Network integration**
PDO provides the NEXTSTEP networking software for hassle-free integration into a NEXTSTEP environment.
NetInfo™ is available, but not required, on PDO machines to simplify heterogeneous network management.

# PRODUCT DETAILS

**NEXTSTEP Object Frameworks**
Foundation Framework
Distributed Objects Framework
DOEventLoop
Other core classes

**Development Tools**
ObjectiveC++ Compiler(v2.5.8)
GDB Debugger(v4.13)
libg++ (v2.5.3)
GNU make (3.69)
Portable BuildServer

**Other Services**
Portable nmserver
Mach Emulation
NEXTSTEP's defaults system
On-line Documentation

**Supported Platforms**
HP-UX
SunOS
Solaris
Digital UNIX

## ORDERING INFORMATION

NeXT Telesales is now taking orders for Portable Distributed Objects. For additional product information or to order, please contact NeXT Telesales in North America at 1-800-848-NeXT. Visa and Mastercard accepted, plus shipping/handling and applicable state sales tax.

**For additional information, call 1-800-TRY-NeXT**

**NeXT Computer, Inc.**
*U.S. Headquarters*
900 Chesapeake Drive
Redwood City, CA94063

*European Offices*
Oskar-Messter-Str. 24
D- 85737 Ismaning
Germany
+49 (89) 996 5310

Tour CBC
8 rue Felix Pyat
92800 Puteaux La Defense
France
+33 (146) 93 27 82

Sommerville House
50a Bath Road
Hounslow, Middlesex TW3 3EE
United Kingdom
+44 (181) 565 0005

1993-95 NeXT Computer, Inc. All rights reserved. NeXT, the NeXT logo, NEXTSTEP, PDO, the PDO logo, Portable Distributed Objects, OpenStep and Project Builder are trademarks of NeXT Computer, Inc. UNIX is a registered trademark of UNIX Systems Laboratories. All other trademarks mentioned belong to their respective owners. NeXT will from time to time revise the specifications described herein, and reserves the right to make such changes without obligation to notify the purchaser.