NEXTSTEP Release 3

A Preamble for Developers

NEXTSTEP Release 3.0 contains new tools and software kits that assist in application development. In addition, some of the familiar tools and kits have been extended or modified, and many of the files that define the NEXTSTEP softwareÐsuch as header files, applications, and librariesÐhave been reorganized.

This document outlines the additions, modifications, and reorganizations provided by Release 3.0 that are of interest to the application developer. These features are presented in three sections: general concepts and compatibility with 2.0, tools (including Objective C), and software kits.

The document only highlights these new features; for more information on a specific topic, look in the appropriate release note file. The release notes, stored in **/NextLibrary/Documentation/NextDev/ReleaseNotes**, provide detailed, topical examinations of the new features, bugs, and incompatibilities presented by the 3.0 software. The **ReleaseNotes** directory is indexed for access through Digital Librarian.

Since there are relatively few changes for NEXTSTEP Developer Release 3.3, the fixed and existing problems are listed here instead of individual product Release Notes.

Additions and Changes for Release 3.3

This section summarizes the most important differences between the 3.3 and 3.2 releases. For more detailed information on changes to a specific component of the release, see the corresponding release note.

New Features

Release 3.3 has a few new features, as listed below.

Platforms

Release 3.3 now runs on SPARCstation 5, 10, and 20 models. It also supports the HP 9000 series 712, 715, 725, 735, and 755 computers; Intel; and NeXT computers.

Features not Available

The following features are not available in NEXTSTEP Developer Release 3.3 for SPARC stations:

- If a peripheral isn't supported by Sun on the SPARCstation 5, 10, and 20 models, NEXTSTEP won't support it either.
- Music and Music Kit (disabled)
- DSP support (68k only)

The following features are not available in NEXTSTEP Developer Release 3.3 for PA-RISC systems:

- EISA bus card support
- If a peripheral isn't supported by HP on the HP 9000 series 712, 715, 725, 735, or 755, NEXTSTEP won't support it either.
- Music and Music Kit (disabled)
- DSP support (68k only)

Differences between NEXTSTEP for SPARC stations and Other Platforms

These are some of the hardware and software architectural differences between NEXTSTEP Release 3.3 for SPARC stations and NEXTSTEP on other platforms:

- SPARC station hardware differences
- SPARC systems are ªBig Endian,º so the bytes in the word are the same as on a 68k implementation but swapped from an Intel implementation
- Data type byte alignment

  The SPARC architecture requires that data has ªnatural alignment.º This means that all the basic data types (**int**s, **float**s, **double**s, etc.) must live on byte boundaries that are an even multiple of the size of the data type. In other words, an **int** (which is 4 bytes in length) must reside in memory on a 4 byte boundary,and a **double** must reside on an 8 byte boundary.

  The compiler and memory allocation routines guarantee that this is always the case. In most cases, therefore, you won't have to worry about this restriction. Problems can arise, however, when assumptions are made about how data is laid out in memory. For example, if a **struct** from an architecture that did not have natural alignment were written to

disk, a SPARC system might not be able to read this correctly. If you need to write structures and data out to a file, you should use a machine-independent file format, such as Typed Streams.

- Naturally aligned memory addresses are required, so the kernel will take traps on non-aligned memory references and send a SIGBUS signal to the processÐwhich by default terminates it.

Differences between NEXTSTEP for PA-RISC Computers and Other Platforms

These are some of the hardware and software architectural differences between NEXTSTEP Release 3.3 for PA-RISC and NEXTSTEP on other platforms:

- PA-RISC Computer hardware differences
- HP systems are ªBig Endian,º so the bytes in the word are the same as on a 68k implementation but swapped from an Intel implementation
- Stack grows in the opposite direction from Intel and 68k implementations
- Data type byte alignment

  The PA-RISC architecture requires that data has ªnatural alignment.º This means that all the basic data types (**int**s, **float**s, **double**s, etc.) must live on byte boundaries that are an even multiple of the size of the data type. In other words, an **int** (which is 4 bytes in length) must reside in memory on a 4 byte boundary, and a **double** must reside on an 8 byte boundary.

  The compiler and memory allocation routines guarantee that this is always the case. In most cases, therefore, you won't have to worry about this restriction. Problems can arise, however, when assumptions are made about how data is laid out in memory. For example, if a **struct** from an architecture that did not have natural alignment were written to disk, an HP system might not be able to read this correctly. If you need to write structures and data out to a file, you should use a machine-independent file format, such as Typed Streams.

- Although naturally aligned memory addresses are required, the kernel will take traps on non-aligned memory references and fix them up. However, this adversely affects performance. Therefore, **gdb** has been enhanced to turn on and detect alignment traps by the following commands:

```
(gdb) set alignment-traps on
(gdb) show alignment-traps
```

- On PA-RISC systems, **/dev/sd*n*a** is the *n*th SCSI disk device where the SCSI devices are *reverse* ordered by SCSI ID. On other NEXTSTEP platforms, **/dev/sd*n*a** is the *n*th SCSI disk device where the SCSI devices are *forward* ordered by SCSI ID.

- The order in which function arguments are evaluated is not specified, so the statement

```
printf("%d %d\n",++n,power(2,n));   /* WRONG */
```

  can produce different results with different compilers, depending upon whether *n* is incremented before power is called.

- Significant software changes were made in adapting NEXTSTEP to PA-RISC systems for the Compiler (see the Compiler and Compiler Tools Release Notes).

Foundation

The Foundation Kit (or, simply, Foundation) defines a base layer of classes that is OpenStep compliant. OpenStep is an operating system independent, object-oriented application layer, based on NeXT's object technology. Foundation provides another primitive class and capabilities similar to the Common classes (List, Hash, Storage, and so on). It provides base classes for things like strings, values, collections and storage. Foundation also introduces paradigms and

mechanisms that enrich the object-oriented development process, especially a new way to deallocate objects. Foundation improves the persistence and distribution of objects within an object system. The *Foundation Reference* is available on-line in **/NextLibrary/Documentation/NextDev/Foundation**.

Enterprise Object Framework

The Enterprise Object Framework provides a new way to develop object-oriented database applications. It provides tools for defining an object model and mapping it to a data model, which allows you to create objects that encapsulate both data and methods for operating on that dataÐand these objects can persist in a relational database accessible via the Framework's data services. You should use the Enterprise Objects FrameworkÐnot the Database KitÐto develop database applications.

Interface Builder

The interface has been improved with new class and connection inspectors and dynamic palette loading.

Compiler

The new version of the compiler is based on gcc 2.5.8 and improves support for C++. It also includes **libg++** support.

Documentation

The *NEXTSTEP Developer's Library* has been augmented in these ways:

- The *Working With Interface Builder* manual has been added. It can be found in **/NextLibrary/Documentation/NextDev/DevTools/NewInterfaceBuilder** and is also available in printed form.

- The *Foundation Reference* has been added and is available on-line in **/NextLibrary/Documentation/NextDev/Foundation**.

- The documentation for the Driver Kit has been expanded and enhanced. It can be found on-line in **/NextLibrary/Documentation/NextDev/OperatingSystem/Part3_DriverKit**.

Problems Fixed

All Platforms

Reference: 20051

Problem: Distributed Objects don't return doubles or structures.

Description: This problem has been fixed, with the exception of some structures 4 bytes or less in size. Returning structures by value works with several caveats:

1. On Intel-based systems (client or server), structures cannot be returned by value. If the structure is increased to greater than 8 bytes, the server will dump core; otherwise, an (apparently) zero-initialized structure will be returned.

2. On Motorola-based systems, structures can be returned by value, but they must be greater than 8 bytes.

3. On PA-RISC systems, 8 byte structures can be returned by value.

Reference: 40637

Problem: Bundle extensions set in Project Builder don't get written to the **Makefile** on a PA-RISC system.

Description: Bundle Extensions are not written to the **Makefile** from the attributes section of Project Builder. The BUNDLE_EXTENSION entry in **Makefile** for Project Builder never exists on PA-RISC systems.

Reference: 45148

Problem: The QuickRenderMan ªconstantº shader doesn't work for PA-RISC systems.

Description: Using the ªconstantº surface shader in QuickRenderMan (qrman) kills the Window Server on PA-RISC systems.

Reference: 45675

Problem: QuickRenderMan (Qrman) can crash the Window Server on PA-RISC systems.

Description: Qrman will kill the Window Server if it tries to render into a 24 bit RGB window with alpha (bps=8, spp=4,alpha=YES). This occurs on HP 9000 systems configured with 24 bit video.

Reference: 41611

Problem: Project Builder compiler options don't carry over to **Makefiles** on PA-RISC systems.

Description: If you set compiler flags on Project Builder's Options panel, **makefile**s don't use these options.

Reference: 47094

Problem: Use of the **-posix** switch leads to unresolved external references.

Description: If you use the **localeconv()** or **setlocale()** functions from the **/usr/lib/libposix** library and compile with the **-posix** flag, you get unresolved external references.

PA-RISC

Reference: 39237, 48285

Problem: Backtraces display only the top frame on PA-RISC systems.

Description: If a breakpoint is hit as the result of a **forward::** message, **gdb** generally won't display the backtrace past the top frame.

Reference: 41097

Problem: **ProcessMonitor** can't look at Malloc information of applications on PA-RISC systems.

Description: When you select a process in **ProcessMonitor** and change the Control Inspector panel's pop-up list to Malloc, the application hangs.

Reference: 43881

Problem: On PA-RISC systems, data breakpoints may fail to be hit.

Description: **gdb** will sometimes miss data breakpoints on PA-RISC systems.

Reference: 45140

Problem: The **gdb printf** command does not work properly.

Description: Arguments passed to the **gdb printf** command are incorrectly printed.

Reference: 45835

Problem: Mute doesn't work for PA-RISC systems.

Description: You can't mute the speaker on PA-RISC systems with the ± **setSpeakerMute:** method in NXSoundDevice.

Reference: 48758

Problem: **gdb**'s info register command mixes up registers on PA-RISC systems.

Description: The **gdb** info register command prints the **sp** value where the **fp** value is supposed to be.

Known Problems

These problems exist in NEXTSTEP Developer Release 3.3:

All Platforms

Reference: 43098

Problem: Reference counting does not work properly in DO.

Description: Reference counting does not work as documented in DO. For instance, references are added each time a server object is passed across a connection (server side) and added to the remote proxy object only when it is created on the client side.

Workaround: None.

Reference: 48978

Problem: Returning a structure by value causes a core dump in server on Intel-based systems.

Description: When a DO server or client attempts to return a structure by value, it dumps core on Intel-based systems for Releases 3.1, 3.2, and 3.3. It works correctly on both Motorola and PA-RISC platforms.

Workaround: None.

Reference: 49064

Problem: DO can't transfer long doubles.

Description: Transfer of long doubles (by value) doesn't work for any architecture.

Workaround: None.

Reference: None

Problem: Icons created as combination 2-bit/12-bit TIFFs don't dither well.

Description: 24-bit icons dither significantly better in the 8-bit color environment than 12-bit icons since they are essentially pre-dithered.

Workaround: Create icons as combination 2-bit/24-bit TIFF images, rather than as 2-bit/12-bit images.

Reference: None

Problem: Emacs in NEXTSTEP Developer Release 3.2 doesn't work with User Release 3.3.

Description: The Emacs software included with NEXTSTEP Developer Release 3.2 doesn't work with Release 3.3.

Workaround: To use Emacs, you must install the **Emacs.pkg** file that comes on the Release 3.3 *NEXTSTEP* CD-ROM *after* you install NEXTSTEP Developer Release 3.3. See ªWhat's on the NEXTSTEP CD-ROM?º in the 3.3 Release Notes.

PA-RISC

Reference: 41593

Problem: When 12 bit window depth objects are created by **IconBuilder**, they change colors as they move on HP 9000 series 712 systems.

Description: If you set 12 bit color depth and create an object with **IconBuilder**, when you select part of the object and move it around, both it and the remaining object change colors.

Workaround: None.

Reference: 43683

Problem: **gdb** displays wrong values for register float variables.

Description: **gdb** consistently misreports the values of register float variables.

Workaround: None.

Reference: 44967

Problem: Using the sound functions **snddriver_*()** can cause a system panic.

Description: If you mix Sound Driver functions with Sound Kit functions, that results in mixing 22 kHz and 44.1 kHz sounds, which leads to a kernel alignment trap. This can occur if you adjust the volume while running an application that uses **snddriver_*()** functions because Preferences emits a system beep to indicate the setting. Current NEXTSTEP software does not use **snddriver_*()** functions.

Workaround: Use Sound Kit functions instead of **snddriver_*()** functions. If you do use **snddriver_*()** functions, do not mix two sounds with different sampling rates. Also make certain that buffers passed to these functions are properly aligned.

Reference: 46864

Problem: **ebadexec** warns that it doesn't know about PA-RISC executable files.

Description: If you type **ebadexec /bin/ebadexec**, you get the warning

```
ebadexec: warning executable file: /bin/ebadexec has cputype and cpusubtype unknown to ebadexec and
the entry point can't be checked to see if it is in a segment
ebadexec: file: /bin/ebadexec appears to be executable
```

Workaround: None.

Reference: None

Problem: Precompiled headers produced on a non-PA-RISC system don't work with the compiler and HeaderViewer on PA-RISC systems. Similarly, precompiled headers produced on a PA-RISC system won't work on a non-PA-RISC system.

Description: PA-RISC systems have a precomp version number that is incompatible with 3.2 Intel and Motorola systems. This makes the precompiled headers produced on a PA-RISC system incompatible with the existing compiler and HeaderViewer on Intel and Motorola systems.

Workaround: Build precompiled headers only on the system type you will be using them on.

Additions and Changes for Release 3.2

This section summarizes the most important differences between the 3.2 and 3.1 releases. For more detailed information on changes to a specific component of the release, see the corresponding release note.

New Features

Release 3.2 has a handful of new features, as listed below.

GNU Sources

The source code for the compiler is now distributed as part of NEXTSTEP. To place this source on your system, install the developer package **GNUSource.pkg**.

Documentation

The NEXTSTEP Developer's Library has been augmented in these ways:

- The *NEXTSTEP Database Kit* manual has been added. It can be found in **/NextLibrary/Documentation/NextDev/Concepts/DatabaseKit**.
- Header Viewer documentation can be found in **/NextLibrary/Documentation/NextDev/DevTools/ApA_HeaderViewer**.

- The Driver Kit documentation has been expanded and improved. It can be found in **/NextLibrary/Documentation/NextDev/OperatingSystem/Part3_DriverKit**.

Additions and Changes for Release 3.1

This section summarizes the most important differences between the 3.1 and 3.0 releases. For more detailed information on changes to a specific component of the release, see the corresponding release note.

New Features

Since Release 3.1 is primarily a maintenance release, there are few new features.

Device Driver Kit

This new kit provides an object-oriented interface to device drivers. See **/NextLibrary/Documentation/NextDev/ReleaseNotes/DriverKit.rtf** for more information.

Documentation

New documentation has been added to the on-line NEXTSTEP Developer's Library:

- *NEXTSTEP Object Oriented Programming and the Objective C Language* is in **/NextLibrary/Documentation/NextDev/Concepts/ObjectiveC.**

- The *NEXTSTEP Programming Interface Summary* manual is now available on line in **/NextLibrary/Documentation/NextDev/Summaries.** This manual was previously available only in printed form.
- Documentation for the new Device Driver Kit is available in **/NextLibrary/Documentation/NextDev/OperatingSystem/Part3_DriverKit**.

Changes

The Phone Kit has been removed from the release.

General Concepts

Application Compatibility

The 3.0 release is binary-compatible: Executable files that were created with Release 2 software and tools will launch, run, and behave as usual. However, 3.0 isn't source code-compatibleÐa handful of constants, functions, and methods that were present in Release 2 software kits have been removed. Because of this, recompiling Release 2 source code isn't guaranteed to work; you may have to modify your source code to successfully build existing applications.

Applications that you build with the 3.0 software won't run on Release 2 or older releases.

Warning: An exception to the binary-compatible rule is the Release 2 version of IconÐthis application will *not* run in the 3.0 release. You should note that Icon has been replaced, in the 3.0 release, by IconBuilder.

The NextDeveloper Directory

Most of the resources that you need to build and compile an application (with the notable exception of libraries) have been moved into the **/NextDeveloper** directory.

Header Files

All the header files that are provided by NEXTSTEPÐfiles that traditionally are placed in **/usr/include**Ðare now located in subdirectories of **/NextDeveloper/Headers**. Beneath the **/NextDeveloper/Headers** directory, the header files are organized much as they were when they lived in **/usr/include**. For example, the header files that define the API for the Application Kit are now in **/NextDeveloper/Headers/appkit**. An exception to this is the group of standard C language header files. These have been divided between two subdirectories, **ansi** and **bsd**.

The compiler knows about the header file move, so you don't have to change your applications' **`#import`** statements. For example, a statement such as

```
#import <sys/stat.h>
```

automatically finds and imports the file **/NextDeveloper/Headers/bsd/sys/stat.h**.

To make the change easier still, **/usr/include** hasn't gone away: It's now a symbolic link to **/NextDeveloper/Headers**.

Makefiles and Palettes

The new **Makefiles** and **Palettes** subdirectories of **/NextDeveloper** contain the standard makefiles and kit-specific palettes used by Project Builder (see below) and Interface Builder. The makefiles used to be in **/usr/lib/nib**; kit-specific palettes are new in Release 3.0.

Localization

A new concept in application portability is *localization*, whereby the text that an application presents is translated into different languages. To support the notion of localization, NEXTSTEP stores the interface data for an application in separate subdirectories within the application's ª.appº file package. An application, when launched, can choose to read in just those languages that the user has specified.

As described in the section ªCommon Classes,º below, the new NXBundle class assists in finding and reading an application's localized directories (as well as other application resources).

Application API

The 3.0 versions of Workspace Manager, Preferences, and Interface Builder present public API, allowing you to create modules that augment these applications. This API can be found in the **/NextDeveloper/Headers/apps** directory. Documentation for this API is in **/NextLibrary/Documentation/NextDev/GeneralRef**.

Tools

The following sections describe the new or modified applications, languages, and programs that help you build your own applications.

Project Builder

Executable: /NextDeveloper/Apps/ProjectBuilder.app

Documentation: /NextLibrary/Documentation/NextDev/DevTools/02_ProjectBuilder

Release notes: /NextLibrary/Documentation/NextDev/ReleaseNotes/ProjectBuilder.rtf

The Project Builder application is the most important addition to the suite of NEXTSTEP application-building tools. It assumes and expands on the role previously taken by Interface Builder's project management facilities. From Project Builder, you can specify the type of application you want to build and the files that it involves. Project Builder automatically creates language directories and provides a default interface file that can be opened by Interface Builder.

In addition to organizing the resources your application needs, Project Builder lets you build, run, and debug your application. It displays compiling and linking error messages in a window, and lets you step through your source code using Edit.

Project Builder is the central tool in creating an application. You begin the creation process by specifying a new project in Project Builder, and then examine the project's componentsÐits interface, source code, images, sounds, and so onÐby double-clicking in Project Builder's file viewer.

A project is described by a *project file*, which Project Builder always names **PB.project**. Project Builder can also read old-style Interface Builder project files (the familiar **IB.proj** files).

Interface Builder
F0.tiff ,

Executable: /NextDeveloper/Apps/InterfaceBuilder.app

Documentation: /NextLibrary/Documentation/NextDev/DevTools/03_InterfaceBuilder

Release notes: /NextLibrary/Documentation/NextDev/ReleaseNotes/InterfaceBuilder.rtf

As mentioned above, project management is no longer a task of Interface Builder. The menu items and panels associated with projects have been removed, and double-clicking on a project file launches Project Builder, rather than Interface Builder.

The format of the Interface Builder file (extension ª.nibº) has changed: In 3.0, nib files are no longer filesÐthey're file packages. However, Interface Builder can read nib files that were created with the 2.0 version of the application.

The palette format has changed as well. Palettes created in 2.0 aren't recognized by the 3.0 Interface Builder.

DBModeler

Executable: /NextDeveloper/Apps/DBModeler.app

Documentation: /NextLibrary/Documentation/NextDev/DevTools/07_DBModeler

Release notes: (none)

DBModeler is a new application that lets you create a representation, or ªmodel,º of an external database (a ªsourceº). This model is stored in a file package (extension ª.dbmodelº) that can be read into your application through objects defined in the Database Kit (described in a later section).

The model that you create isn't simply an image of the database source. Through the DBModeler application, you can create new columns of data, rename existing columns, restrict the amount or type of information that the model can access, and so on.

Icon Builder

Executable: /NextDeveloper/Apps/IconBuilder.app

Documentation: /NextLibrary/Documentation/NextDev/DevTools/06_IconBuilder

Release notes: (none)

Icon Builder is a TIFF-file creation application. It can be used to create any size or type of TIFF file; however, it's designed to be used to create icons for your applications. IconBuilder replaces the 2.0 application Icon (which, as pointed out above, will not run in the 3.0 release).

Objective C

Documentation: /NextLibrary/Documentation/NextDev/Concepts/ObjectiveC

Release notes: /NextLibrary/Documentation/NextDev/ReleaseNotes/ObjC.rtf
/NextLibrary/Documentation/NextDev/ReleaseNotes/Compiler.rtf

New to Objective C is the concept of *protocols*. A protocol is a named set of methods that's not associated with any one class in particular. Protocols let you reuse object interfaces, just as inheritance lets you reuse method implementations.

You declare a protocol through the **@protocol** specifier, giving the protocol a name and a set of methods terminated by **@end**:

```
@protocol StringParser
- convertToUpper;
- convertToLower;
@end
```

A protocol looks a lot like a class, but with two important differences: It can't have instance variables, and it doesn't define implementations for its methods. Implementing a protocol's methods is the responsibility of classes that *adopt* the protocol. A class adopts a protocol by including the protocol's name, in angle brackets, on the interface declaration line:

```
@interface MyDictionary : Object < StringParser >
```

Here, **MyDictionary** is declared to adopt **StringParser** and so is obliged to implement the methods that **StringParser** declares.

The 3.0 release of Objective C includes some other syntactical additions:

- Five new reserved words, **in**, **out**, **inout**, **oneway**, and **bycopy**, can be used to modify the data types of arguments to protocol methods. They enable special processing when sending a message to a *distributed object* (see ªDistributed Objects,º below, and the Distributed Objects release notes). Note that these new reserved words apply *only* to protocol methods.
- The @**class** specifier lets you refer to a class without importing its header file. This ªforward referenceº tells the compiler to allow the class to be used as a data typeÐfor example, as the type of an instance variable, or the return type of a method. However, it doesn't allow references to the class's methods (for which you still need to import the header file).
- The @**private**, @**protected**, and @**public** declarations are used to control access to instance variables. A private instance variable can only be accessed in the methods of the class that declares the variable. A protected variable can be accessed in the methods of the declaring class and any of its subclasses. Finally, a public instance variable can be used in any code that accesses an object that owns the variable.

Compiler and Preprocessor

Documentation: /NextLibrary/Documentation/NextDev/DevTools/11_Compiler
/NextLibrary/Documentation/NextDev/DevTools/12_Preprocessor

Release notes: /NextLibrary/Documentation/NextDev/ReleaseNotes/Compiler.rtf
/NextLibrary/Documentation/NextDev/ReleaseNotes/PrecompiledHeaders.rtf
/NextLibrary/Documentation/NextDev/ReleaseNotes/Preprocessor.rtf

The 3.0 compiler and preprocessor have been improved to work faster and more intelligently. Both have new command line switches that increase optimization and control the error and warning messages that they produce.

An important addition to the compilation process is *precompiled headers*, header files that have been preprocessed to make compiling your application much more efficient. Precompiled headers are created for entire directories of normal ª.hº files; for example, the Application Kit provides a single **appkit.p** file in the **appkit** directory.

To import a precompiled header, you specify the ª.hº version of the file:

```
#import <appkit/appkit.h>
```

The compiler will look for the corresponding precompiled header. If you're importing a precompiled header, you shouldn't import individual header files from the same directory (it isn't an error to do so, it's simply unnecessary and inefficient). Note that not all header file directories offer precompiled headers.

gdb

Documentation: /NextLibrary/Documentation/NextDev/DevTools/13_Debugger

Release notes: /NextLibrary/Documentation/NextDev/ReleaseNotes/Debugger.rtf

The connection between the gdb debugger and Edit has been tightened: gdb's **view** command adds a **Gdb...** menu item to Edit that, when clicked, brings up a panel that controls gdb's activity. You can run, step, breakpoint, and end the debugged program by clicking buttons in the panel. Through the panel you can bring up a browser that displays the contents of the program's stack frames.

Kits and Other Software

NEXTSTEP 3.0 offers a number of new software kits. In addition, some of the veteran kits contain many new features.

Database Kit

Documentation:
/NextLibrary/Documentation/NextDev/Concepts/DatabaseKit
/NextLibrary/Documentation/NextDev/GeneralRef/04_DatabaseKit

Release notes: /NextLibrary/Documentation/NextDev/ReleaseNotes/DBKit.rtf

The Database Kit is a new software kit that allows developers to create data access applications. It's designed to be flexible and abstract: The kit assumes nothing about the source of data, and a single database application can work with more than one source at a time.

There are three layers to the Database Kit:

- At the highest level, the Database Kit provides user interface objects that can be incorporated into Interface Builder through the Database Kit palette (**/NextDeveloper/Palettes/DatabaseKit.palette**).
- The data access layer provides objects that you use in your application to access and manipulate data that has been read from a database source. These objects don't affect the source directlyÐthey simply play with data that has been

copied from the source. It's the access layer objects that the interface objects communicate with.

- For each database source, you must have an adaptor. The adaptor translates the source's data representation into access layer objects. It also allow the access layer to talk to the source by translating access object commands into the source's native language. The DBModeler aids in creating an interface between a database source and the access layer. The 3.0 release includes adaptors for Sybase and Oracle databases.

Indexing Kit

Documentation: /NextLibrary/Documentation/NextDev/GeneralRef/07_IndexingKit

Release notes: /NextLibrary/Documentation/NextDev/ReleaseNotes/Indexing.rtf

The Indexing Kit is another new kit; it provides a fast and efficient mechanism for organizing and retrieving data. In that they both handle data, the Indexing Kit and Database Kit are similar; the difference between them lies in where the data is stored. The Database Kit depends on an external source of data that has its own data storage and manipulation language and a data server. The Indexing Kit defines objects that store data locally. The Digital Librarian application and the Help system both use the Indexing Kit to store and look-up information.

3D Graphics Kit

Documentation: /NextLibrary/Documentation/NextDev/GeneralRef/17_3DKit

Release notes: /NextLibrary/Documentation/NextDev/ReleaseNotes/3DKit.rtf

The 3D Graphics Kit defines objects that describe, manipulate, display, and print photo-realistic three-dimensional images. The kit is based on, and assumes some knowledge of, the industry standard RenderMan language (as documented in *The RenderMan Companion*, Steve Upstill, Addison-Wesley, 1990).

The images you build through the 3D Kit can be displayed in the same window with conventional two-dimensional images created by Application Kit objects. The 3D Kit uses a different description language to encode image data, but the mechanism by which the data is displayed is the same as the Application Kit.

Distributed Objects

Documentation: /NextLibrary/Documentation/NextDev/GeneralRef/06_DistributedObjects

Release notes: /NextLibrary/Documentation/NextDev/ReleaseNotes/DistributedObjects.rtf

Distributed Objects is a new facility that helps you create client-server programs. It takes the form of two new classes, NXConnection and NXProxy. These classes, which aren't assigned to a specific software kit, allow an application (the server) to provide objects that can be shared among any number of other applications (the clients). Your application can send a message to a distributed object and thereby affect any other application that shares the object.

Common Classes

Documentation: /NextLibrary/Documentation/NextDev/GeneralRef/03_Common

Release notes: (none)

The NXBundle class is new in the 3.0 release. An NXBundle object corresponds to a resource-providing directory (or ªbundleº). It provides methods that let you distinguish between parallel resources, and lets you retrieve specific resources from the directory.

Application Kit

Documentation: /NextLibrary/Documentation/NextDev/GeneralRef/02_AppKit

Release notes: /NextLibrary/Documentation/NextDev/ReleaseNotes/AppKit.rtf
/NextLibrary/Documentation/NextDev/ReleaseNotes/ObjectLinks.rtf

The 3.0 Application Kit boasts a number of new or improved features:

- An improved image-dragging scheme, whereby data (represented by an icon) can be dragged between windows.
- An interface to the Help system.
- A mechanism called Object Links that allows different applications to share dataÐif the data is modified by one application, the change is automatically seen in the other application.
- A new class, NXPrinter, that represents printer devices and their capabilities.
- Support for RTFD (*Rich Text Format Directory*) in the Text class.

- New classes and protocols that let you incorporate a spell-checking system into your application.
- Better support for application-provided services.

Sound

Documentation: /NextLibrary/Documentation/NextDev/GeneralRef/16_SoundKit

Release notes: /NextLibrary/Documentation/NextDev/ReleaseNotes/SoundKit.rtf
/NextLibrary/Documentation/NextDev/ReleaseNotes/Sound.rtf
/NextLibrary/Documentation/NextDev/ReleaseNotes/SndCmd.rtf

The Sound Kit provides six new classes that provide access to the sound driver for simple recording and playback. These classes replace the old sound driver functions that recorded and played back sound data.

New sound functions that convert sound data to and from the *Audio Transform Compression* (ATC) format have been added for 3.0. The ATC format can shrink a soundfile to a tenth of its original size (or even smaller) without affecting sound quality in most listening situations.

Music and the DSP

Documentation: (none)

Release notes: /NextLibrary/Documentation/NextDev/ReleaseNotes/DSP.rtf

The Music Kit and other DSP development tools, with the exception of the DSP assembler, linker, and librarian, and basic MIDI support, have been unbundled in Release 3.0. These tools are available from the Center for Computer Research in Music and Acoustics at Stanford University. See the release notes for details.