Worldwide Developers Conference

Introduction to OpenStep Application Framework

Jordan Dea-Mattson

Senior Evangelist
Rhapsody Evangelism

Introduction to Yellow Box Application Framework

Ali Ozer

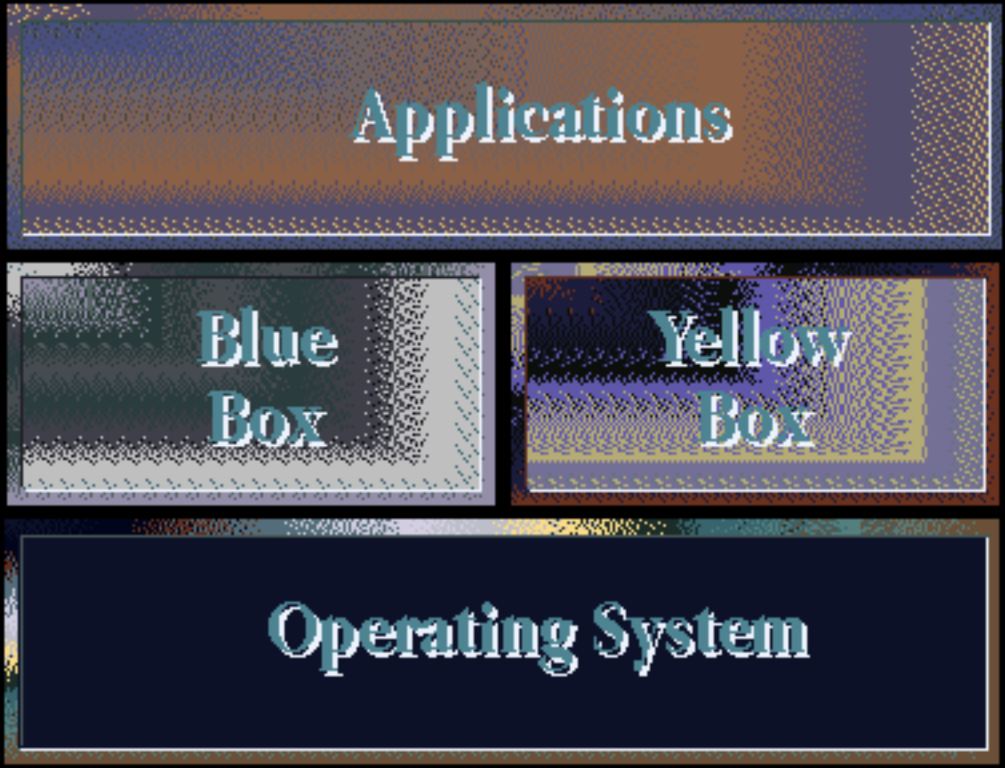
Manager, Yellow Box Application Frameworks

Yellow Box Application Framework

Where does it fit?

Yellow Box Application Framework

Where does it fit?



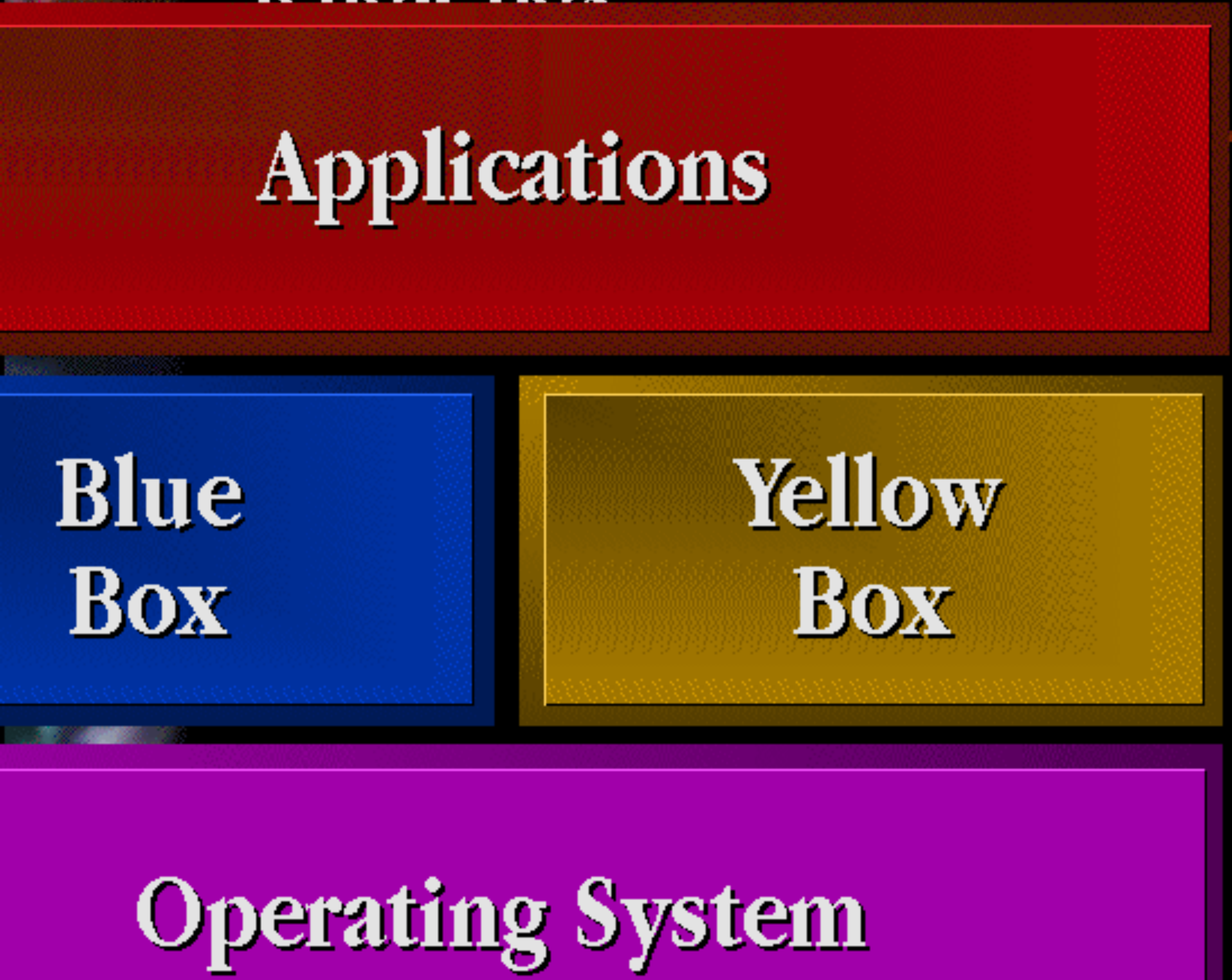
Applications

Blue Box

Yellow Box

Operating System

Yellow Box



Applications

Blue Box

Yellow Box

Operating System

Yellow Box Application Framework

Where does it fit?

Yellow Box

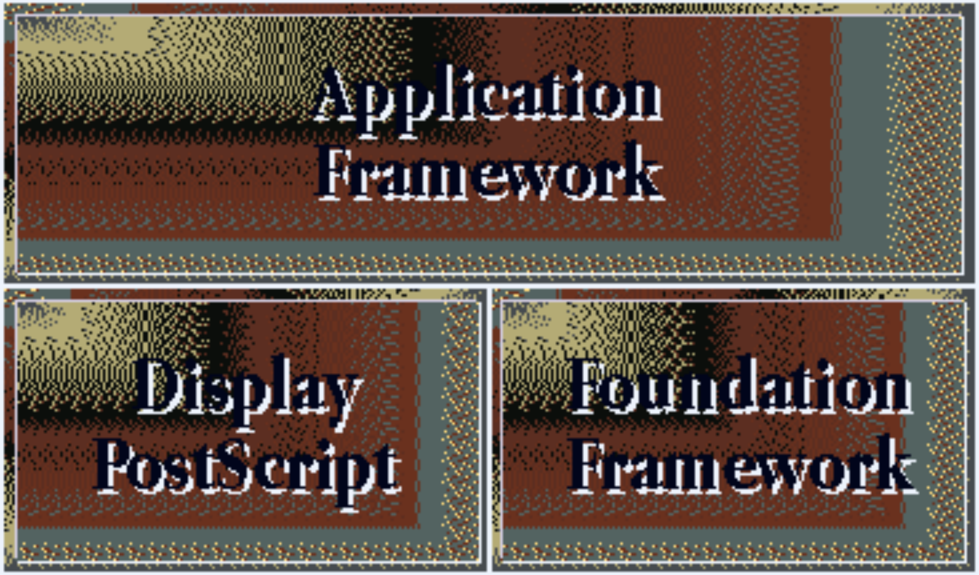
Yellow Box Application Framework

Where does it fit?

Yellow Box

Yellow Box Application Framework

Where does it fit?



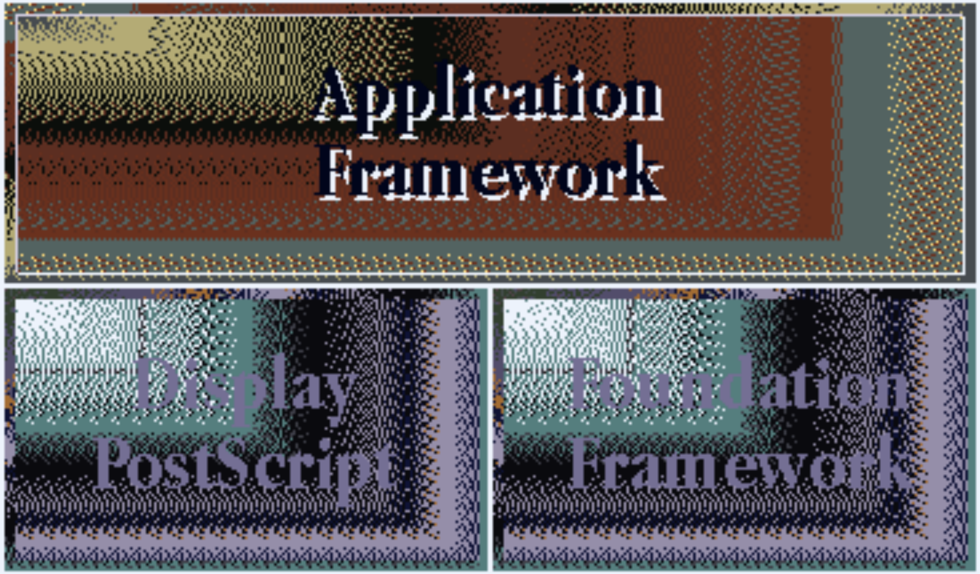
Application Framework

Display PostScript

Foundation Framework

Yellow Box
Application Framework

Where does it fit?



Yellow Box
Application Framework

What is it?

- A full-featured, object-oriented, platform-independent framework for building applications
- AKA “AppKit”

Application Framework

Topics

- Design principles
- Paradigms for extensibility
- Conventions for APIs
- Overview of classes
- What's there in 4.2
- What's coming up

Design Principles

Simple things simple, complex things possible

- Things should “just work”

Design Principles

Simple things simple, complex things possible

- Things should “just work”
- Factor common code out from applications

Design Principles

Simple things simple, complex things possible

- Things should “just work”
- Factor common code out from applications
- Provide extensible and reusable objects

Design Principles

Simple things simple, complex things possible

- Things should “just work”
- Factor common code out from applications
- Provide extensible and reusable objects
- Make the APIs easy to use

Design Principles

Simple things simple, complex things possible

- Things should “just work”
- Factor common code out from applications
- Provide extensible and reusable objects
- Make the APIs easy to use
- Get the layers right

Design Principles

Simple things simple, complex things possible

- Things should "just work"
- Factor common code out from applications
- Provide extensible and reusable objects
- Make the APIs easy to use
- Get the layers right
- Be pragmatic…

Design Principles

Simple things simple, complex things possible

- Things should "just work"
- Factor common code out from applications
- Provide extensible and reusable objects
- Make the APIs easy to use
- Get the layers right
- Be pragmatic…

Paradigms for Extensibility

- Subclassing
  - Basic way to extend most objects
  - Define primitives, override points

Paradigms for Extensibility

- Delegation
  - Allows one object to act on behalf of another
  - Extension without subclassing
  - Examples
    - NSWindow's *windowShouldClose:*
    - NSSavePanel's *panel:shouldShowFilename:*

Paradigms for Extensibility

- Notification
  - Allows broadcasting of events to any interested object
  - Objects register for notifications based on type and posting object
  - Examples
    - *NSWindowWillMiniaturize*, *NSWindowDidMiniaturize*
    - *NSApplicationWillTerminate*
  - Notifications are often available as delegate methods as well

Conventions for APIs

- Return autoreleased objects

Conventions for APIs

- Return autoreleased objects
- Reuse concepts wherever possible

Conventions for APIs

- Return autoreleased objects
- Reuse concepts wherever possible
- Choose clarity over brevity
  - No abbrevs

Conventions for APIs

- Return autoreleased objects
- Reuse concepts wherever possible
- Choose clarity over brevity
  - No abbrevs
- Make instance variables private

Conventions for APIs

- Return autoreleased objects
- Reuse concepts wherever possible
- Choose clarity over brevity
  - No abbrevs
- Make instance variables private
- Use exceptions for exceptional situations or programming errors

Conventions for APIs

- Return autoreleased objects
- Reuse concepts wherever possible
- Choose clarity over brevity
  - No abbrevs
- Make instance variables private
- Use exceptions for exceptional situations or programming errors

Overview

- Drawing
- Event handling
- User interface
- Standard panels
- Data types and abstractions
- Services
- Internationalization

Drawing

- NSView
  - Abstract class to handle drawing
  - Also provides event handling, dragging, printing/pagination, keyboard navigation, context sensitive menus, mouse tracking
  - Lives in hierarchies, one per window
  - Override *drawRect:* to customize drawing
  - Drawing often done buffered

Drawing

- NSView
  - Abstract class to handle drawing
  - Also provides event handling, dragging, printing/pagination, keyboard navigation, context sensitive menus, mouse tracking
  - Lives in hierarchies, one per window
  - Override *drawRect:* to customize drawing
  - Drawing often done buffered
- NSWindow

Drawing

- NSView
  - Abstract class to handle drawing
  - Also provides event handling, dragging, printing/pagination, keyboard navigation, context sensitive menus, mouse tracking
  - Lives in hierarchies, one per window
  - Override *drawRect:* to customize drawing
  - Drawing often done buffered
- NSWindow

Drawing

- AppKit abstractions, enough for most situations
  - NSCell *drawWithFrame:inView:*
  - NSImageRep, NSString, NSAttributedString *drawAtPoint:, drawInRect:*
  - NSImage *compositeToPoint:operation:*
  - NSBezierPath *stroke, fill, clip (New in DR)*
  - NSColor, NSFont *set*

Drawing

- AppKit abstractions, enough for most situations
  - NSCell *drawWithFrame:inView:*
  - NSImageRep, NSString, NSAttributedString *drawAtPoint:, drawInRect:*
  - NSImage *compositeToPoint:operation:*
  - NSBezierPath *stroke, fill, clip (New in DR)*
  - NSColor, NSFont *set*
- PostScript
  - *PSmoveto(), PSlineto()*

Drawing

- AppKit abstractions, enough for most situations
  - NSCell *drawWithFrame:inView:*
  - NSImageRep, NSString, NSAttributedString *drawAtPoint:, drawInRect:*
  - NSImage *compositeToPoint:operation:*
  - NSBezierPath *stroke, fill, clip (New in DR)*
  - NSColor, NSFont *set*
- PostScript
  - *PSmoveto(), PSlineto()*

Event Handling

- NSEvent, for low-level access
  - Look at the event stream or override *mouseDown:*, *keyDown:*, etc. in NSView

Event Handling

- NSEvent, for low-level access
  - Look at the event stream or override *mouseDown:*, *keyDown:*, etc. in NSView
- Target/Action
  - Simple but powerful model for handling events on UI widgets
  - UI widgets send their *target* an *action*, with themselves as the single argument
  - A *nil* target implies that the action goes to the first responder
    - *cut:*, *copy:*, *paste:*, *close:*, *save:*, ...

Event Handling

- NSEvent, for low-level access
  - Look at the event stream or override *mouseDown:*, *keyDown:*, etc. in NSView
- Target/Action
  - Simple but powerful model for handling events on UI widgets
  - UI widgets send their *target* an *action*, with themselves as the single argument
  - A *nil* target implies that the action goes to the first responder
    - *cut:*, *copy:*, *paste:*, *close:*, *save:*, …

User Interface

- Consistent with the platform

User Interface

- Consistent with the platform
- Created via Interface Builder, and stored in *nib* files as archived objects
  - The nib file also contains target/action connections and other relationships

User Interface

- Consistent with the platform
- Created via Interface Builder, and stored in *nib* files as archived objects
  - The nib file also contains target/action connections and other relationships
- Can have multiple nib files

User Interface

- Consistent with the platform
- Created via Interface Builder, and stored in *nib* files as archived objects
  - The nib file also contains target/action connections and other relationships
- Can have multiple nib files
- **As with all resources, nib files can be customized for different platforms**

  -
    ```
    MyApp.nib
    MyApp-macintosh.nib
    MyApp-windows.nib
    ```

User Interface

- Consistent with the platform
- Created via Interface Builder, and stored in *nib* files as archived objects
  - The nib file also contains target/action connections and other relationships
- Can have multiple nib files
- As with all resources, nib files can be customized for different platforms
  - `MyApp.nib`
    `MyApp-macintosh.nib`
    `MyApp-windows.nib`

User Interface Elements

- Simple widgets
  - NSControl, NSButton, NSSlider, NSTextField, NSColorWell, NSImageView, …

User Interface Elements

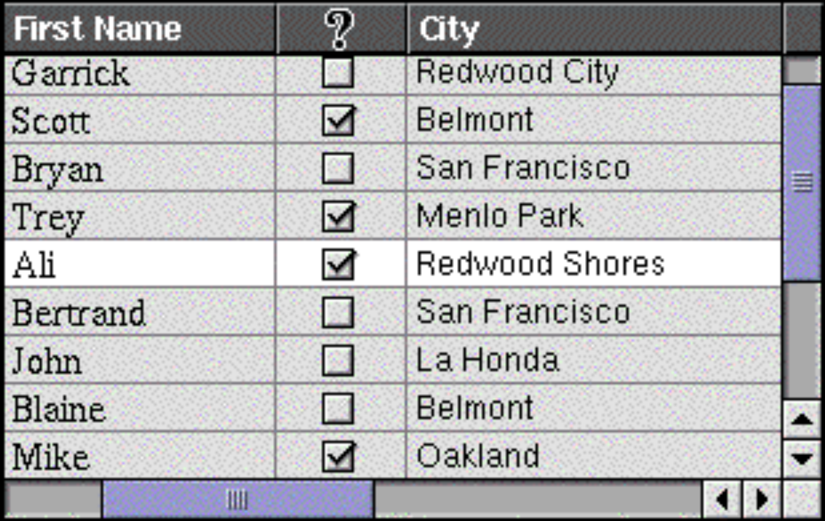
- Simple widgets
  - NSControl, NSButton, NSSlider, NSTextField, NSColorWell, NSImageView, …
- Complex widgets
  - NSMatrix, NSMenu, NSComboBox
  - NSTableView, NSBrowser, NSOutlineView
    - These display and edit complex data structures
    - They store no data; they talk to a *dataSource* to get the data lazily

User Interface Elements

- Simple widgets
  - NSControl, NSButton, NSSlider, NSTextField, NSColorWell, NSImageView, …
- Complex widgets
  - NSMatrix, NSMenu, NSComboBox
  - NSTableView, NSBrowser, NSOutlineView
    - These display and edit complex data structures
    - They store no data; they talk to a *dataSource* to get the data lazily

User Interface Elements

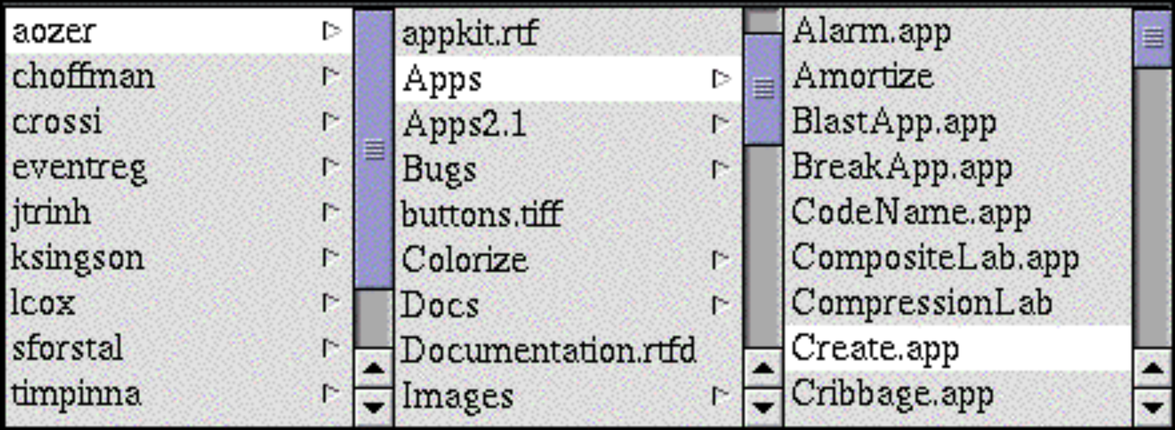
- **NSTableView**
  - Tables of data



First Name	?	City
Garrick	☐	Redwood City
Scott	☑	Belmont
Bryan	☐	San Francisco
Trey	☑	Menlo Park
Ali	☑	Redwood Shores
Bertrand	☐	San Francisco
John	☐	La Honda
Blaine	☐	Belmont
Mike	☑	Oakland

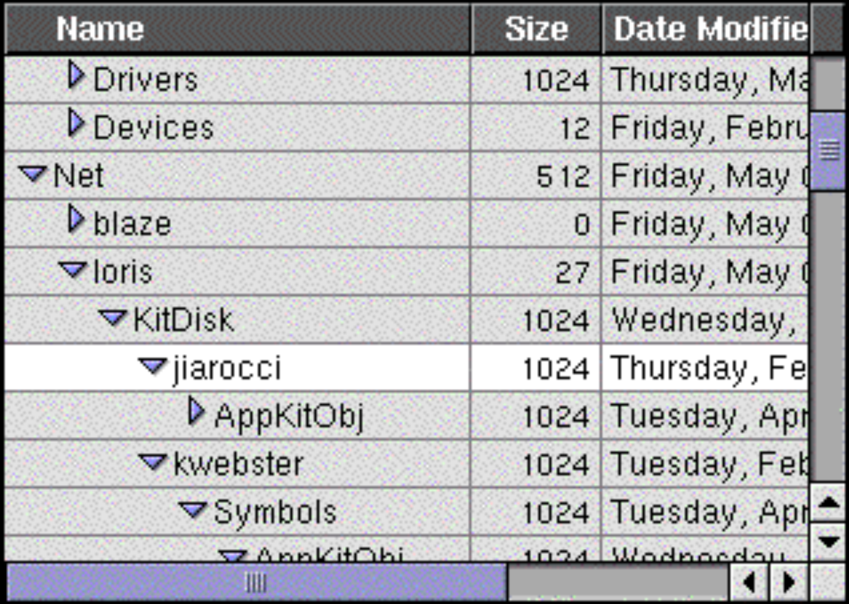
User Interface Elements

- NSBrowser
  - Hierarchical data, shown in columns



User Interface Elements

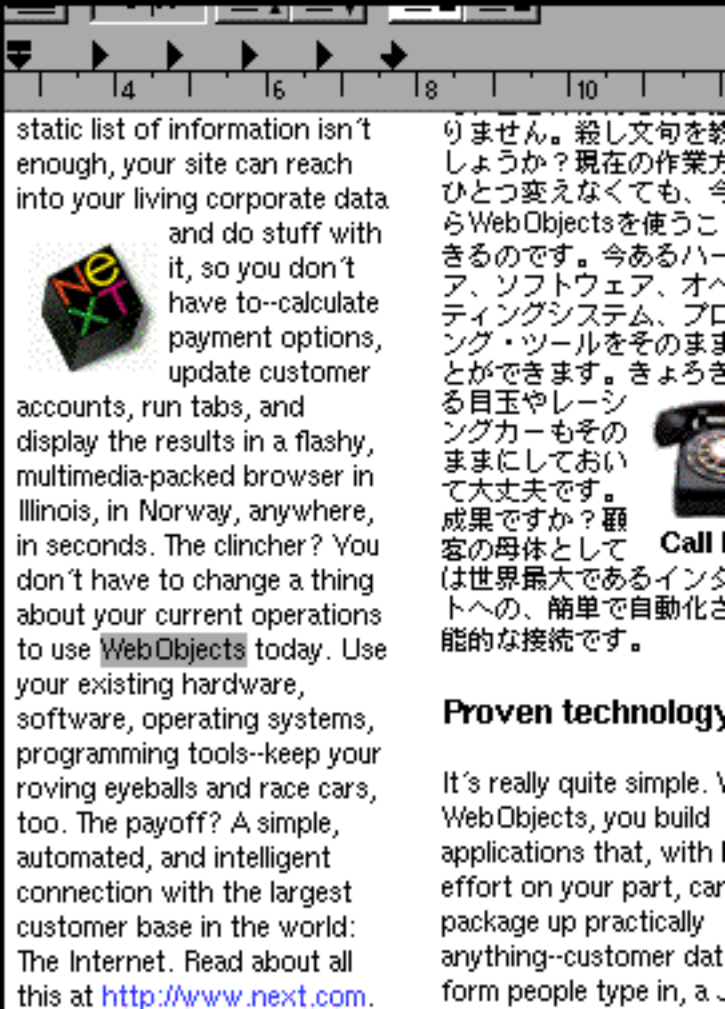
- NSOutlineView (New in DR)
  - Hierarchical data, shown hierarchically



Name	Size	Date Modifie
Drivers	1024	Thursday, Ma
Devices	12	Friday, Febru
Net	512	Friday, May (
blaze	0	Friday, May (
loris	27	Friday, May (
KitDisk	1024	Wednesday,
jiarocci	1024	Thursday, Fe
AppKitObj	1024	Tuesday, Apr
kwebster	1024	Tuesday, Feb
Symbols	1024	Tuesday, Apr
AppKitObj	1024	Wednesday

User Interface Elements

- NSTextView
  - Front-end to the text system
  - Powerful, extensible, international



static list of information isn´t enough, your site can reach into your living corporate data and do stuff with it, so you don´t have to--calculate payment options, update customer accounts, run tabs, and display the results in a flashy, multimedia-packed browser in Illinois, in Norway, anywhere, in seconds. The clincher? You don´t have to change a thing about your current operations to use WebObjects today. Use your existing hardware, software, operating systems, programming tools--keep your roving eyeballs and race cars, too. The payoff? A simple, automated, and intelligent connection with the largest customer base in the world: The Internet. Read about all this at http://www.next.com.

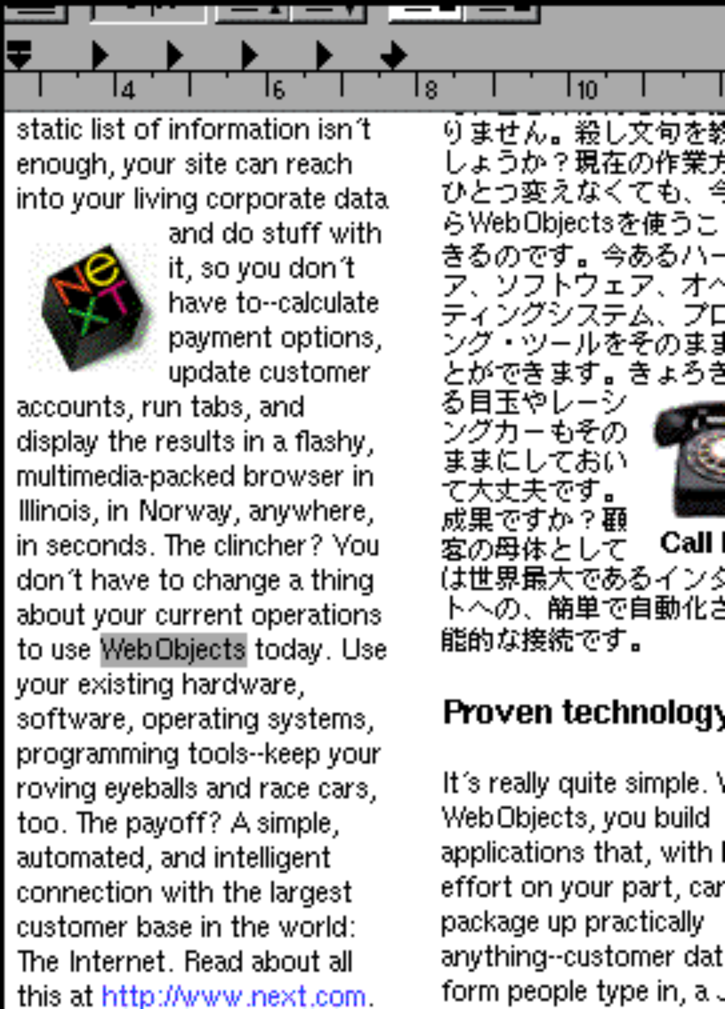
りません。殺し文句を教
しょうか？現在の作業方
ひとつ変えなくても、今
らWebObjectsを使うこ
きるのです。今あるハー
ア、ソフトウェア、オペ
ティングシステム、プロ
ング・ツールをそのまま
とができます。きょろき
る目玉やレーシ
ングカーもその
ままにしておい
て大丈夫です。
成果ですか？顧
客の母体として
は世界最大であるインタ
トへの、簡単で自動化さ
能的な接続です。

Proven technology

It´s really quite simple. W
WebObjects, you build
applications that, with l
effort on your part, can
package up practically
anything--customer dat
form people type in, a J

User Interface Elements

- NSTextView
  - Front-end to the text system
  - Powerful, extensible, international
- NSTextStorage
  - Back-end



static list of information isn't enough, your site can reach into your living corporate data and do stuff with it, so you don't have to--calculate payment options, update customer accounts, run tabs, and display the results in a flashy, multimedia-packed browser in Illinois, in Norway, anywhere, in seconds. The clincher? You don't have to change a thing about your current operations to use WebObjects today. Use your existing hardware, software, operating systems, programming tools--keep your roving eyeballs and race cars, too. The payoff? A simple, automated, and intelligent connection with the largest customer base in the world: The Internet. Read about all this at http://www.next.com.

りません。殺し文句を
しょうか？現在の作業
ひとつ変えなくても、
らWebObjectsを使うこ
きるのです。今あるハー
ア、ソフトウェア、オ
ティングシステム、プ
ング・ツールをそのま
とができます。きょろき
る目玉やレーシ
ングカーもその
ままにしておい
て大丈夫です。
成果ですか？顧
客の母体として
は世界最大であるイン
トへの、簡単で自動化
能的な接続です。

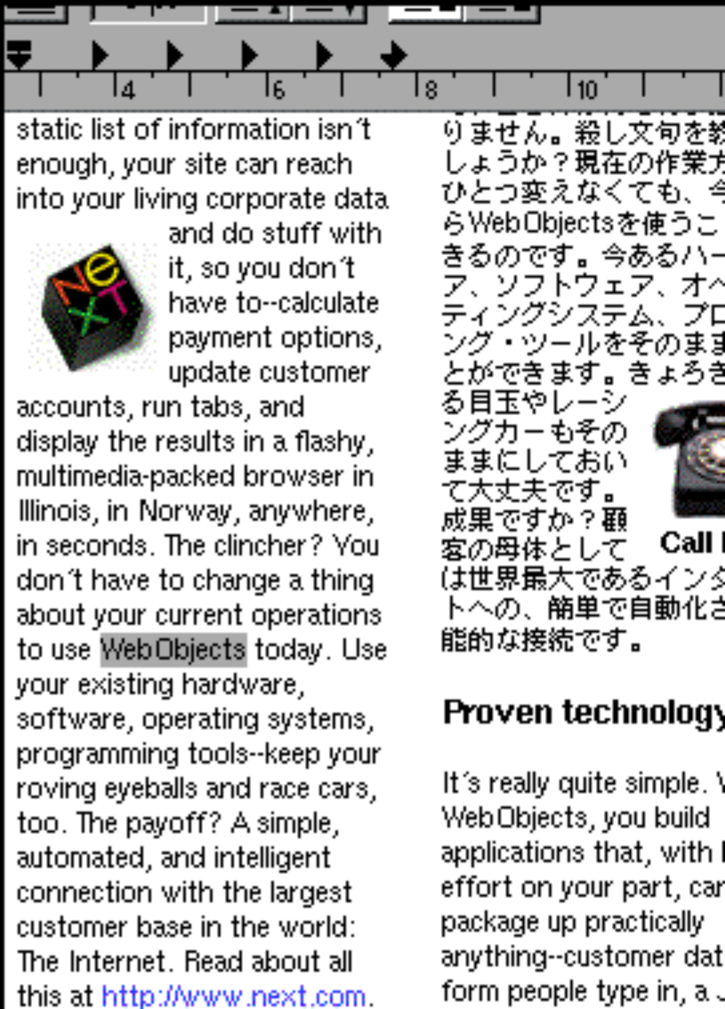
Call

Proven technolog

It's really quite simple.
WebObjects, you build
applications that, with
effort on your part, ca
package up practically
anything--customer dat
form people type in, a

User Interface Elements

- NSTextView
  - Front-end to the text system
  - Powerful, extensible, international
- NSTextStorage
  - Back-end
- NSLayoutManager



static list of information isn't enough, your site can reach into your living corporate data and do stuff with it, so you don't have to--calculate payment options, update customer accounts, run tabs, and display the results in a flashy, multimedia-packed browser in Illinois, in Norway, anywhere, in seconds. The clincher? You don't have to change a thing about your current operations to use WebObjects today. Use your existing hardware, software, operating systems, programming tools--keep your roving eyeballs and race cars, too. The payoff? A simple, automated, and intelligent connection with the largest customer base in the world: The Internet. Read about all this at http://www.next.com.

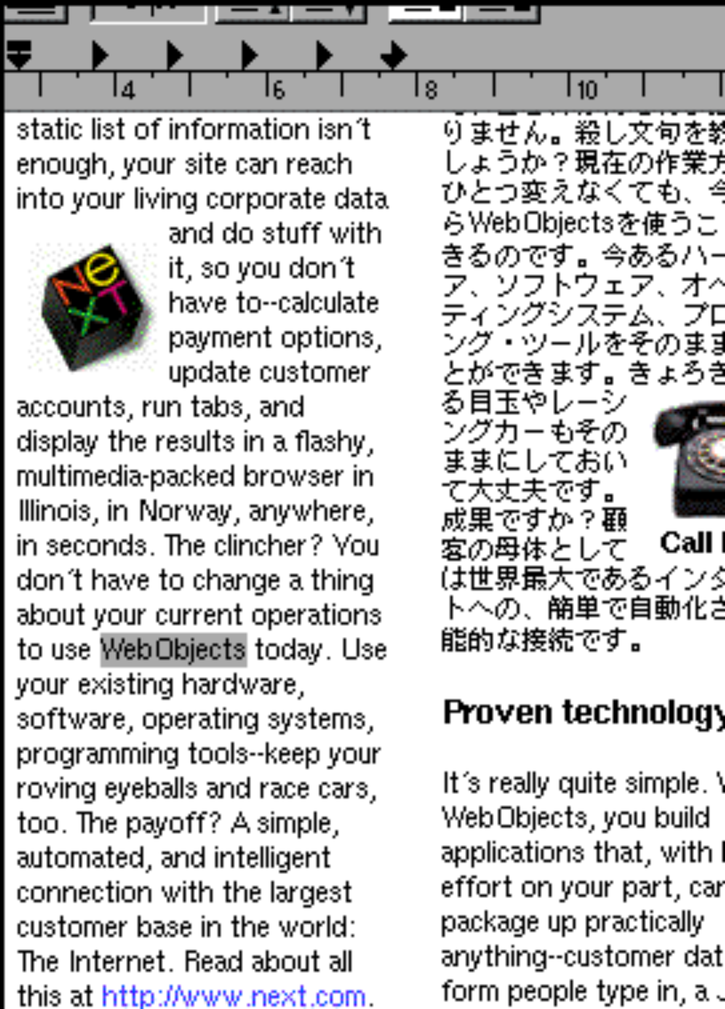
りません。殺し文句を
しょうか？現在の作業方
ひとつ変えなくても、今
らWebObjectsを使うこ
きるのです。今あるハー
ア、ソフトウェア、オペ
ティングシステム、プロ
ング・ツールをそのまま
とができます。きょろき
る目玉やレーシ
ングカーもその
ままにしておい
て大丈夫です。
成果ですか？顧
客の母体として
は世界最大であるインタ
トへの、簡単で自動化さ
能的な接続です。

Proven technology

It's really quite simple. W
WebObjects, you build
applications that, with l
effort on your part, car
package up practically
anything--customer dat
form people type in, a J

User Interface Elements

- NSTextView
  - Front-end to the text system
  - Powerful, extensible, international
- NSTextStorage
  - Back-end
- NSLayoutManager
- NSTextContainer



static list of information isn´t enough, your site can reach into your living corporate data and do stuff with it, so you don´t have to--calculate payment options, update customer accounts, run tabs, and display the results in a flashy, multimedia-packed browser in Illinois, in Norway, anywhere, in seconds. The clincher? You don´t have to change a thing about your current operations to use WebObjects today. Use your existing hardware, software, operating systems, programming tools--keep your roving eyeballs and race cars, too. The payoff? A simple, automated, and intelligent connection with the largest customer base in the world: The Internet. Read about all this at http://www.next.com.

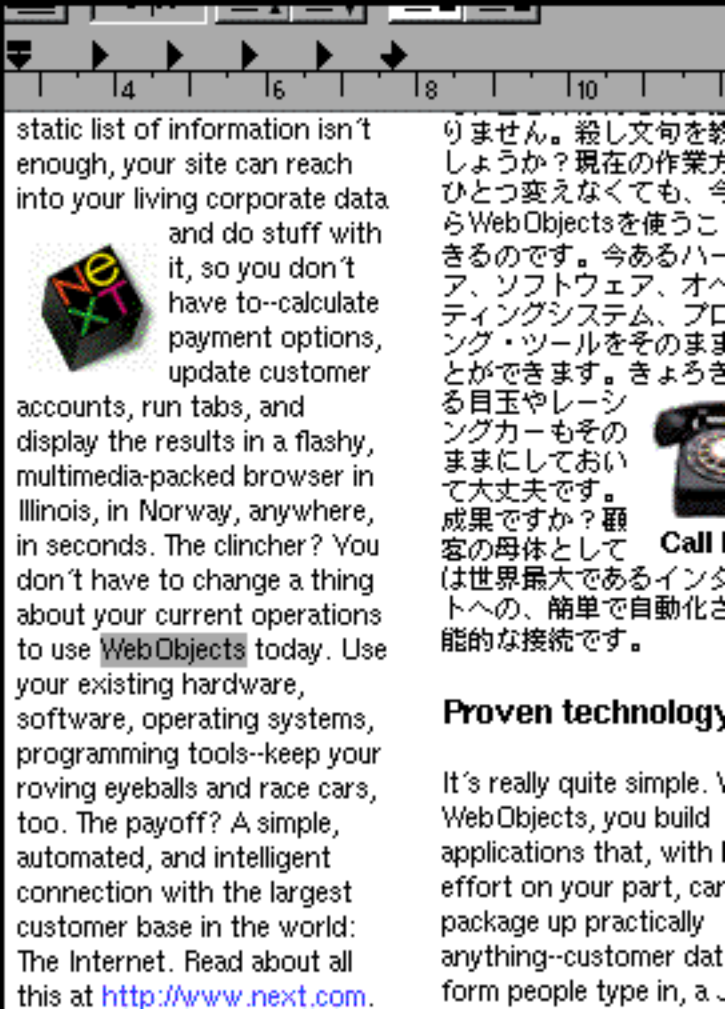
りません。殺し文句を
しょうか？現在の作業
ひとつ変えなくても、
らWebObjectsを使うこ
きるのです。今あるハー
ア、ソフトウェア、オ
ティングシステム、プ
ング・ツールをそのま
とができます。きょろ
る目玉やレーシ
ングカーもその
ままにしておい
て大丈夫です。
成果ですか？顧
客の母体として
は世界最大であるイン
トへの、簡単で自動化
能的な接続です。

Proven technolog

It´s really quite simple.
WebObjects, you build
applications that, with
effort on your part, ca
package up practically
anything--customer da
form people type in, a

User Interface Elements

- NSTextView
  - Front-end to the text system
  - Powerful, extensible, international
- NSTextStorage
  - Back-end
- NSLayoutManager
- NSTextContainer



static list of information isn´t enough, your site can reach into your living corporate data and do stuff with it, so you don´t have to--calculate payment options, update customer accounts, run tabs, and display the results in a flashy, multimedia-packed browser in Illinois, in Norway, anywhere, in seconds. The clincher? You don´t have to change a thing about your current operations to use WebObjects today. Use your existing hardware, software, operating systems, programming tools--keep your roving eyeballs and race cars, too. The payoff? A simple, automated, and intelligent connection with the largest customer base in the world: The Internet. Read about all this at http://www.next.com.

りません。殺し文句を教
しょうか？現在の作業方
ひとつ変えなくても、今
らWebObjectsを使うこ
きるのです。今あるハー
ア、ソフトウェア、オペ
ティングシステム、プロ
ング・ツールをそのまま
とができます。きょろき
る目玉やレーシ
ングカーもその
ままにしておい
て大丈夫です。
成果ですか？顧
客の母体として
は世界最大であるインタ
トへの、簡単で自動化さ
能的な接続です。

Call

Proven technology

It´s really quite simple. W
WebObjects, you build
applications that, with
effort on your part, can
package up practically
anything--customer dat
form people type in, a J

User Interface Elements

- View groupers
  - NSScrollView
  - NSSplitView
  - NSBox
  - NSTabView (New in DR)

User Interface Elements

- View groupers
  - NSScrollView
  - NSSplitView
  - NSBox
  - NSTabView (New in DR)
- Misc
  - NSRulerView
  - NSProgressIndicator (New in DR)

User Interface Elements

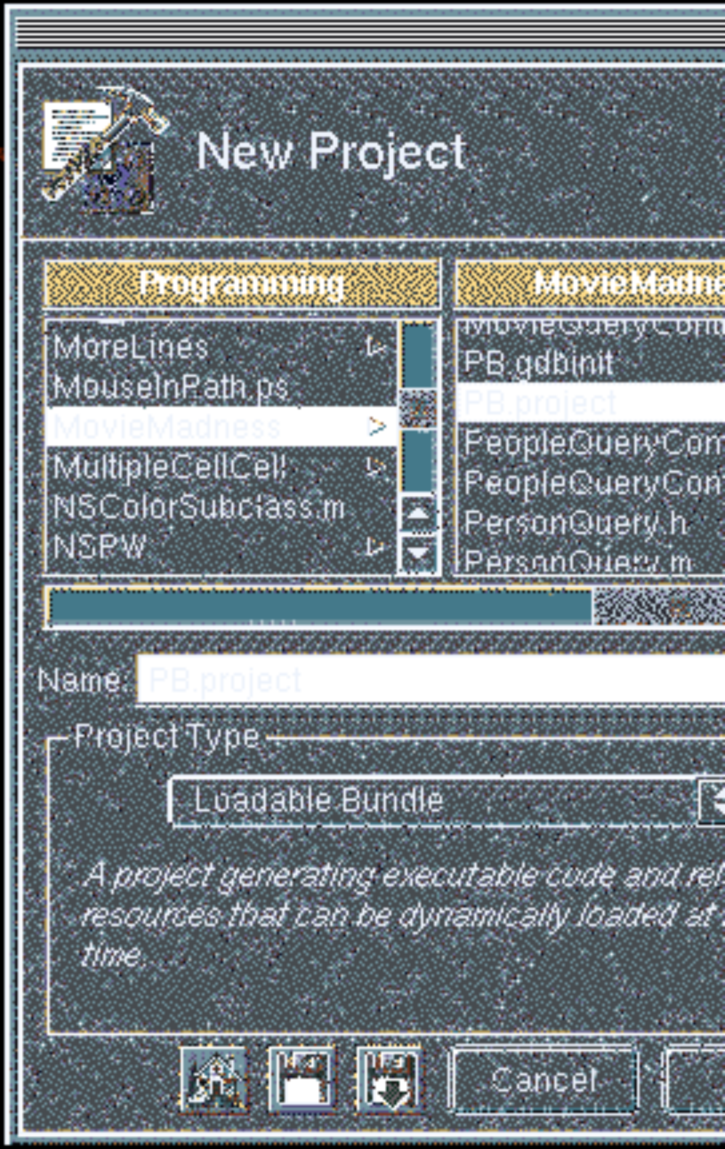
- View groupers
  - NSScrollView
  - NSSplitView
  - NSBox
  - NSTabView (New in DR)
- Misc
  - NSRulerView
  - NSProgressIndicator (New in DR)

Standard Panels

- Panels
  - Open/Save
  - Color
  - Font
  - Print, PageLayout
  - Fax
  - Spelling

Standard Panels

- Panels
  - Open/Save
  - Color
  - Font
  - Print, PageLayout
  - Fax
  - Spelling
- Accessory views allow extending the standard panels



New Project

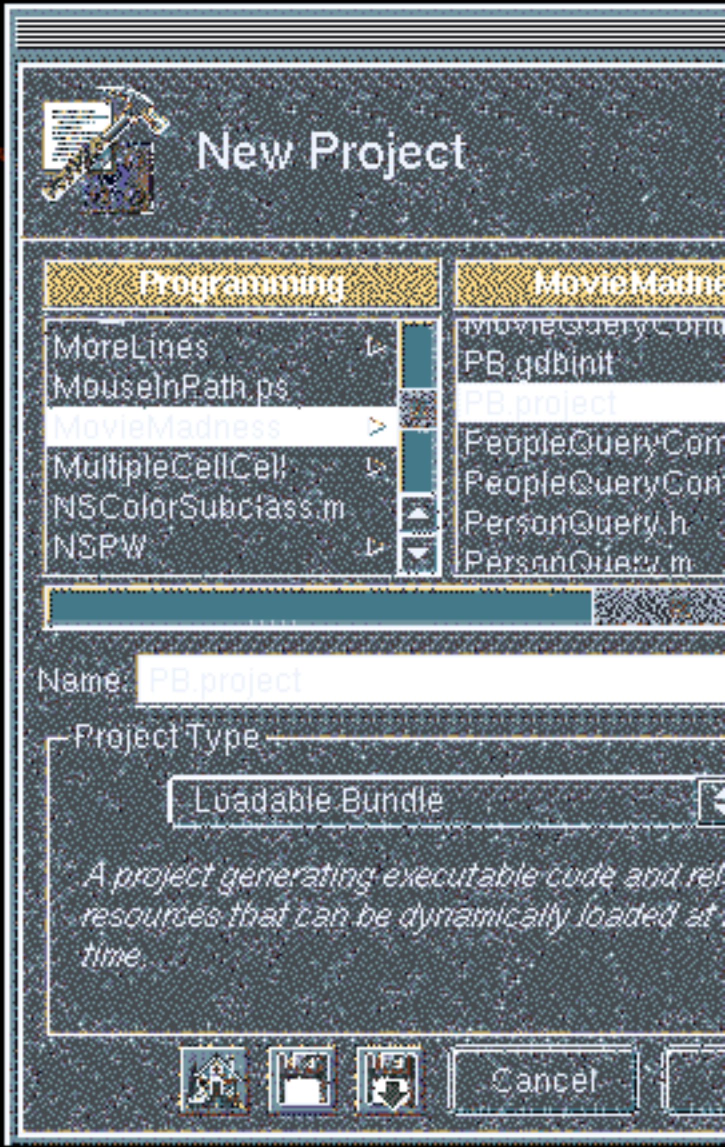
Name: PB.project

Project Type: Loadable Bundle

A project generating executable code and re resources that can be dynamically loaded at time.

Standard Panels

- Panels
  - Open/Save
  - Color
  - Font
  - Print, PageLayout
  - Fax
  - Spelling
- Accessory views allow extending the standard panels



Name: PB.project

Project Type: Loadable Bundle

A project generating executable code and re
resources that can be dynamically loaded at
time.

Data Type Abstractions

- NSImage, NSImageRep
  - EPS, TIFF, BMP, PICT, JPG, GIF

Data Type Abstractions

- NSImage, NSImageRep
  - EPS, TIFF, BMP, PICT, JPG, GIF
- NSString, NSAttributedString
  - Plain text in Unicode and other encodings, RTF, HTML

Data Type Abstractions

- NSImage, NSImageRep
  - EPS, TIFF, BMP, PICT, JPG, GIF
- NSString, NSAttributedString
  - Plain text in Unicode and other encodings, RTF, HTML
- NSFont
  - Type 1, TrueType

Data Type Abstractions

- **NSImage, NSImageRep**
  - EPS, TIFF, BMP, PICT, JPG, GIF
- **NSString, NSAttributedString**
  - Plain text in Unicode and other encodings, RTF, HTML
- **NSFont**
  - Type 1, TrueType
- **NSColor**
  - Various color spaces, patterns, ColorSync

Data Type Abstractions

- NSImage, NSImageRep
  - EPS, TIFF, BMP, PICT, JPG, GIF
- NSString, NSAttributedString
  - Plain text in Unicode and other encodings, RTF, HTML
- NSFont
  - Type 1, TrueType
- NSColor
  - Various color spaces, patterns, ColorSync

System Services

- NSApplication

System Services

- NSApplication
- NSWorkspace
  - Interact with the Workspace
  - Obtain information about applications, document types, icons

System Services

- NSApplication
- NSWorkspace
  - Interact with the Workspace
  - Obtain information about applications, document types, icons
- NSFileWrapper
  - Deal with documents stored as folders

System Services

- NSApplication
- NSWorkspace
  - Interact with the Workspace
  - Obtain information about applications, document types, icons
- NSFileWrapper
  - Deal with documents stored as folders
- NSPasteboard

System Services

- NSApplication
- NSWorkspace
  - Interact with the Workspace
  - Obtain information about applications, document types, icons
- NSFileWrapper
  - Deal with documents stored as folders
- NSPasteboard
- Drag and Drop

System Services

- NSApplication
- NSWorkspace
  - Interact with the Workspace
  - Obtain information about applications, document types, icons
- NSFileWrapper
  - Deal with documents stored as folders
- NSPasteboard
- Drag and Drop

System Services

- NSSpellChecker, NSSpellServer

System Services

- NSSpellChecker, NSSpellServer
- NSInputManager, NSInputServer

System Services

- NSSpellChecker, NSSpellServer
- NSInputManager, NSInputServer
- Formatting and Validation

System Services

- NSSpellChecker, NSSpellServer
- NSInputManager, NSInputServer
- Formatting and Validation
- Keyboard Navigation

System Services

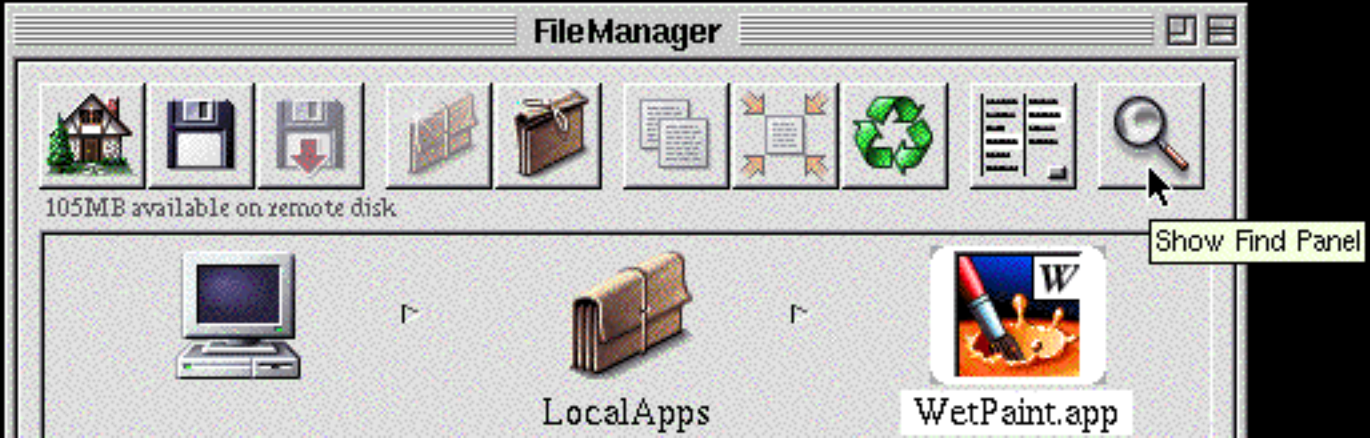
- NSSpellChecker, NSSpellServer
- NSInputManager, NSInputServer
- Formatting and Validation
- Keyboard Navigation
- NSKeyBindingManager (New in DR)
  - Map keyboard sequences to actions

System Services

- NSSpellChecker, NSSpellServer
- NSInputManager, NSInputServer
- Formatting and Validation
- Keyboard Navigation
- NSKeyBindingManager (New in DR)
  - Map keyboard sequences to actions

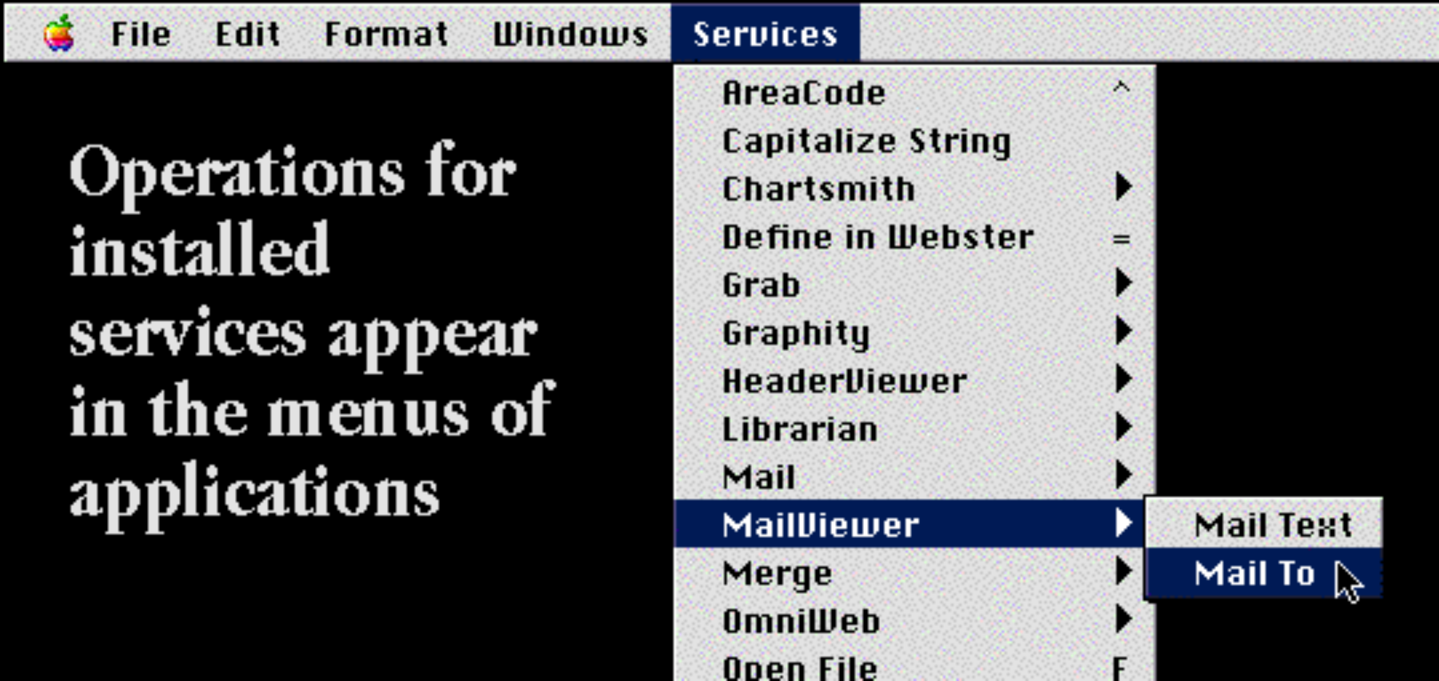
System Services

- Help
  - ToolTips
  - Context Help
  - Comprehensive Help



Interapplication Services

- Allows extending the system via operations on data



Operations for installed services appear in the menus of applications

Filter Services

- Convert files to formats recognized by the system
  - NSPasteboard *pasteboardByFilteringFile:* or *pasteboardByFilteringData:ofType:*
  - NSImage *initWithContentsOfFile:*, *imageFileTypes*
  - Soon on NSAttributedString

Internationalization

- Unicode everywhere

Internationalization

- Unicode everywhere
- International text object

Internationalization

- Unicode everywhere
- International text object
- Easily localizable

Internationalization

- Unicode everywhere
- International text object
- Easily localizable

What's There in 4.2

- Almost everything discussed so far

What's There in 4.2

- Almost everything discussed so far
- Tools
  - Interface Builder, Project Builder
  - FileMerge, Yap, Terminal
  - Performance Tools

What's There in 4.2

- Almost everything discussed so far
- Tools
  - Interface Builder, Project Builder
  - FileMerge, Yap, Terminal
  - Performance Tools
- Examples
  - /NextDeveloper/Examples/AppKit

What’s There in 4.2

- Almost everything discussed so far
- Tools
  - Interface Builder, Project Builder
  - FileMerge, Yap, Terminal
  - Performance Tools
- Examples
  - /NextDeveloper/Examples/AppKit
- Documentation
  - Class reference and release notes (accessible through PB)

What’s There in 4.2

- Almost everything discussed so far
- Tools
  - Interface Builder, Project Builder
  - FileMerge, Yap, Terminal
  - Performance Tools
- Examples
  - /NextDeveloper/Examples/AppKit
- Documentation
  - Class reference and release notes (accessible through PB)

Right Around the Corner...

- Macintosh UI

Right Around the Corner...

- Macintosh UI
- Undo

Right Around the Corner…

- Macintosh UI
- Undo
- New user interface classes

Right Around the Corner...

- Macintosh UI
- Undo
- New user interface classes
- API additions to existing classes

Right Around the Corner...

- Macintosh UI
- Undo
- New user interface classes
- API additions to existing classes
- Internet data types (HTML, JPG, GIF)

Right Around the Corner...

- Macintosh UI
- Undo
- New user interface classes
- API additions to existing classes
- Internet data types (HTML, JPG, GIF)

And Beyond…

- Java APIs

And Beyond…

- Java APIs
- Java Beans and (on Windows) ActiveX integration

And Beyond...

- Java APIs
- Java Beans and
  (on Windows) ActiveX integration
- Scriptability

And Beyond…

- Java APIs
- Java Beans and (on Windows) ActiveX integration
- Scriptability
- More sophisticated text support

And Beyond…

- Java APIs
- Java Beans and
  (on Windows) ActiveX integration
- Scriptability
- More sophisticated text support
- More multi-thread safety

And Beyond…

- Java APIs
- Java Beans and
  (on Windows) ActiveX integration
- Scriptability
- More sophisticated text support
- More multi-thread safety
- Advanced UI

And Beyond…

- Java APIs
- Java Beans and
  (on Windows) ActiveX integration
- Scriptability
- More sophisticated text support
- More multi-thread safety
- Advanced UI

Related Talks…

Talks you hopefully went to…

- *209* Rhapsody Text System and Localization
- *212* Imaging Under Rhapsody

Related Talks...

And talks you should go to...

- *208* Yellow Box Overview
  Today 3:10, Room B
- *216* Developing with IB & PB
  Tomorrow 10:50, Hall 1
- *405* Building Java-based Applications
  Tomorrow 3:10, Room A2
- *207* Intro to Foundation Framework
  Tomorrow 3:10, Room B
- *219* OpenStep: A View from the Trenches
  Tomorrow 3:10, Hall 1

Q&A

Worldwide Developers Conference