

Box 1

Folder 10

**ParcPlace promotional material and
newsletters, 1988-1992**

Adele Goldberg papers

Smalltalk

X5774.2010, Box 1

102739393

THE COMPUTER HISTORY MUSEUM



1 027 3939 3

Objectworks\SmalltalkTM RELEASE 4

*The one object-oriented
system for developing
true color, graphic
applications that run
unchanged on UNIX, PC
and Macintosh workstations
under their standard
windowing systems.*



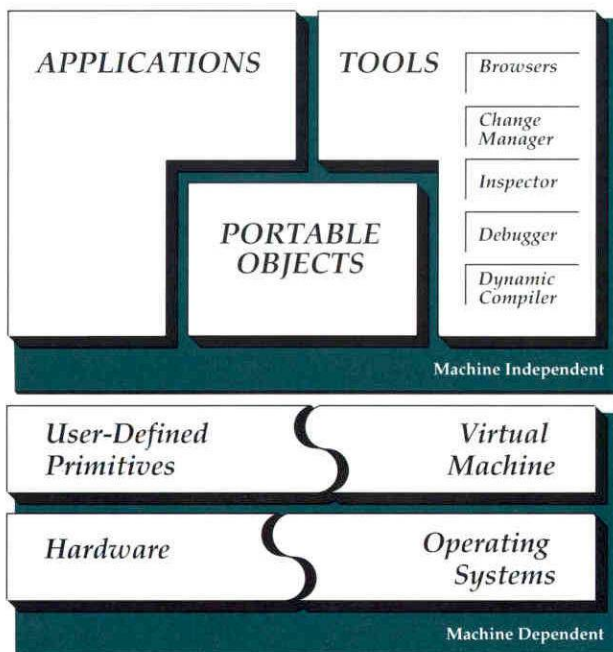
ParcPlace

Objectworks\Smalltalk

RELEASE 4, is an object-oriented system for the development and delivery of true color, graphic applications in a heterogeneous environment.

The system contains three crucial components which give software developers maximum productivity. These components provide the best environment for increased programmer productivity and rapid development of applications:

- *The Smalltalk language*
- *A set of integrated designed-for-objects development tools*
- *A mature class library of over 350 types of Portable Objects™*



Key Capabilities

INSTANT PORTABILITY

Objectworks\Smalltalk portability means instant portability - it's measured in minutes, not months. Our architecture is designed so that applications run without changes across platforms, operating systems, and windowing systems. The risks of developing in a particular environment are greatly reduced since you have the flexibility to move to new environments in order to meet market and technology requirements.

REUSABILITY

One of the great benefits of object-oriented languages is the ability to reuse code. Development productivity and quality increases by reusing existing components and existing designs.

Objectworks\Smalltalk's robust class library of Portable Objects provide the essential building blocks for application development. The complete source code for the entire system is included, allowing the developer to reuse every bit of code in the system - including all the development tools.

PRODUCTIVITY

Objectworks\Smalltalk is the best environment for rapid development. But you need more than that - the integrated environment makes maintenance faster, too. We use the platform windowing system without requiring the developer to learn the details of each windowing system. Our color implementation is device independent. Plus, the integrated tools provide superior access to your code - greatly improving productivity.

Smalltalk Language

SMALLTALK: A UNIFORM, DYNAMICALLY COMPILED LANGUAGE

Developers using Smalltalk benefit from the power of a uniform object-oriented language. Smalltalk's object-oriented characteristics include encapsulation, inheritance, polymorphism, data hiding, data abstraction, and dynamic binding. Developers also benefit from dynamic compilation, giving the performance of native machine code plus easier debugging.

DYNAMICALLY COMPILED LANGUAGE

Smalltalk stores programs in intermediate code (byte-codes), and then quickly compiles each procedure at run time into native machine code the first time that procedure is called. Thereafter, the procedure then executes at full compiled speed. We call this process dynamic compilation.

Dynamic compilation is the only compilation technology that offers all of the following benefits:

- Portability of applications (for example, within a heterogeneous network) without recompiling
- Very compact compiled code (typically a factor of 5 smaller than native machine code)
- Convenient support of high-level debuggers and performance analyzers
- The ability to quickly edit and compile code anytime and anywhere
- Full execution speed of native machine code

ADVANCED AUTOMATIC MEMORY MANAGEMENT

Automatic memory management intelligently controls memory requirements in the background, allowing the developer to concentrate on building the application rather than on writing and debugging heap management code, which is frequently error prone.

Objectworks\Smalltalk employs three types of garbage collectors in order to provide the most efficient memory management. Complete access to the growth and reclamation policies allows tuning of the memory-management policy to fit the needs of an application.

Integrated Development Tools

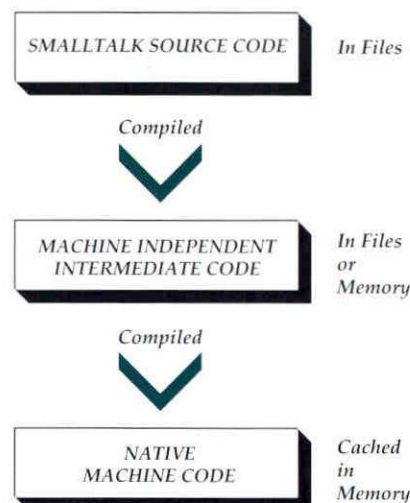
RICH, DESIGNED-FOR-OBJECTS TOOLS

An object-oriented programming language alone is not enough to insure the benefits of object-oriented technology. Integrated development tools are essential. These tools help the programmer to find, understand, observe and reuse code. Combined with the ability to modify and test code at anytime, the creation of new applications and the supportability of existing applications is dramatically accelerated.

INTEGRATED DEVELOPMENT TOOLS

Over fifteen integrated development tools have been designed to maximize programmer productivity.

- Browsers - to find, understand, and modify the class library. Types of browsers include System, Inheritance, and various Method (Sender, Implementor, Message) browsers
- Inspectors - to examine and modify the state of objects
- Dynamic Cross Referencing - to help the programmer understand code and objects
- Symbolic Debugger - to examine, observe, and modify code and objects related to executing code
- Change List Manager - to manage changes for group programming efforts
- Editor - a robust text editor complete with context sensitive spell checking
- Workspaces - scratch pads to test and explore code
- Application Delivery Optimizer - to assist in the creation of runtime applications



Class Library of Portable Objects

OVER 350 TYPES OF PORTABLE OBJECTS

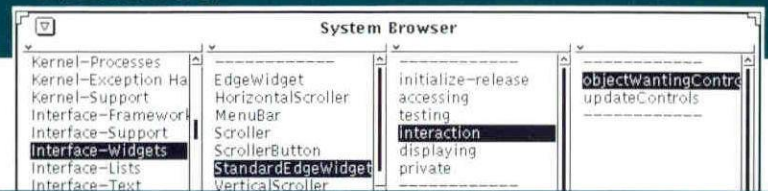
The industry standard, mature Smalltalk class library contains more than 350 types of Portable Objects and over 7400 methods. Since complete source code is included, you are able to modify, extend, and reuse every object; even tools like the editor can be included in your application. The high quality, tested code modules are available for immediate use as the essential building blocks for application development.

PORTABLE OBJECTS

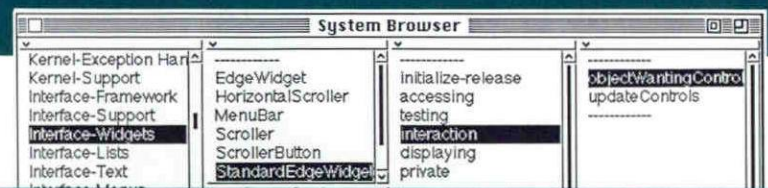
All of the classes and methods are tested and ready to use. Categories of Portable Objects include:

- Tools - all of the development tools are included
- Numeric Values - dates, times, integers, floating point numbers, fractions, and random numbers
- Collections - composite objects, including arrays, strings, and sets
- Graphics - Color and Visual Components
- Kernel - classes and methods used to interact with the platform-dependent part of the system, such as delays, semaphores, and exception handling
- Interface - GUI components and defaults such as widgets, menus, and text
- Operating System - interface to platform facilities such as files and directory

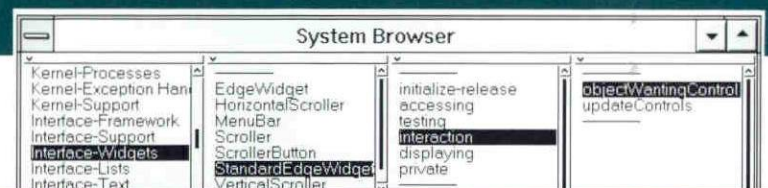
OPENLOOK



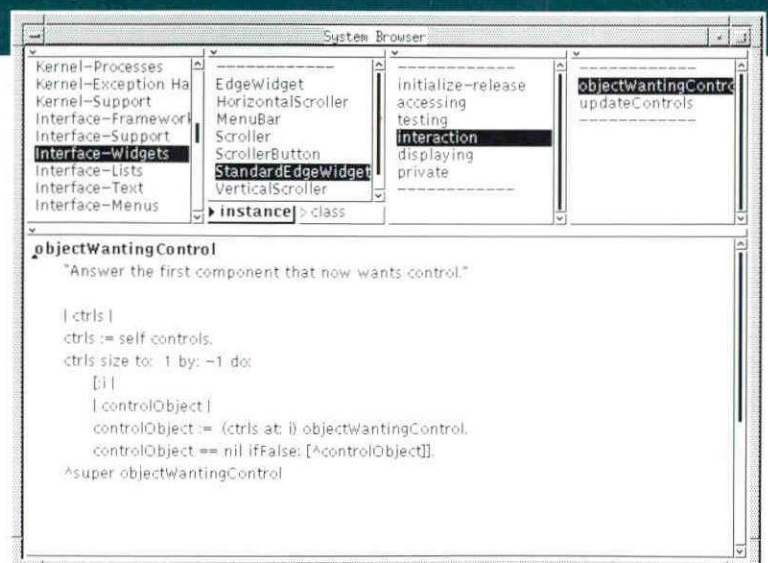
MACINTOSH



MS WINDOWS 3.0



MOTIF



Other Features

INTERFACING TO OTHER APPLICATIONS

Objectworks\Smalltalk environment is easily extended to interface with your existing software environment. User defined primitives, or UDPs, allow you to extend your Smalltalk application to access other routines, applications, and databases. Whether you call a mathematically intensive routine, or wish to access an existing "back end" application or database, Objectworks\Smalltalk leverages and complements the strengths of your existing applications.

Objectworks\Smalltalk also provides access and support to communication protocols; including access to BSD sockets, support for invoking UNIX programs as sub-processes such as csh and Bourne shells, and access to serial port devices including driver-level mode manipulation.

WINDOW SYSTEM INTEGRATION

Objectworks\Smalltalk integrates with the platform windowing system. Whether you're developing and/or running under Motif, OpenLook (or any other X Window System-based GUI), MS Windows 3.0 or the Macintosh, applications run without changes under your preferred windowing system. You have complete flexibility in choosing the windowing system - without spending the resources to learn each one.

STATE OF THE ART TRUE COLOR

True color is state of the art color technology. Color applications are easier to develop with true color, and our device independence gives applications total portability even among displays with different color resolution. Your application can take full advantage of the most advanced color displays yet also run on greyscale or monochrome displays - without changing a single line of code.

DELIVER RUNTIME APPLICATIONS

Customizable runtime applications are easily delivered in Objectworks\Smalltalk. The Application Delivery Optimizer is included for creating your runtime application - for stripping out unused classes and sealing off access to the compiler.

SPIM

SPIM, Smalltalk Portable Imaging Model, is a unique facility that leverages the platform's graphics facilities. Whether your hardware system has rich graphic facilities or less sophisticated ones, SPIM uses a "greatest common multiple" approach and compensates for the platform differences in order to make your application completely portable. It offers graphic support of filled/unfilled polygons, circular & elliptical arcs, wedges, images and lines (with styles).

NATIONAL LANGUAGE SUPPORT

Objectworks\Smalltalk works with a 16-bit character set, allowing support of national languages and complete compatibility with international printers, keyboards, and displays. The character set is optimized to 8 bits for more efficient handling of most European languages. Multiple national languages can be mixed throughout an application without modifying the underlying code, giving your applications and customers unprecedented flexibility. Even programs can be written in national language.

MODEL-VIEW-CONTROLLER (MVC)

Objectworks\Smalltalk's unique user interface architecture, Model-View-Controller (MVC), promotes development productivity and quality by increasing the potential for reuse and reducing the impact of changes. MVC cleanly separates the graphical application component (View), the data component (Model) and the interaction component (Controller). This enables you to "plug" together graphical applications.

Features

100% portable across supported platforms, operating systems, and windowing systems.

Mature class library of Portable Objects with complete source code

Dynamically compiled, uniform language

Device independent true color

Integration with standard windowing system

Delivery of runtime applications

Benefits

Instant portability is achieved in minutes, not months. Reduces the risk of developing a product for a particular environment since the product can move to new environments as technology and markets dictate.

Reuse of existing code gets programs up and running quickly, while increasing program modularity and improving program stability.

Allows you to exploit all the benefits of a uniform object-oriented language - plus giving you the ability to make changes quickly, simplifying development and maintenance, and the performance of native machine code.

Applications can be written for the most advanced color displays, yet can run on any display.

Users enjoy a familiar "look and feel". Developers' productivity increases since they don't need to learn the intricacies of each windowing system.

Sophisticated applications can be easily delivered to end users.

Platform Specification

<i>Platform</i>	<i>Minimum Memory</i>	<i>Disk</i>	<i>OS</i>	<i>Window System Requirements</i>
80386- and 80486-based IBM PS/2 and compatibles	4MB	10MB Free Space	MS-DOS 3.3	MS Windows 3.0
Macintosh II	4MB	10MB Free Space	System 6.0.4, Multifinder 6.0.4	None
Apollo Series 2500, 3500, 4500	8MB	10MB Free Space	Domain/OS 10.2	X Window System
DECstation	8MB	10MB Free Space	Ultrix 3.1/UWS 2.2D	X Window System
HP 9000 Series 300 and 400	8MB	10MB Free Space	HP-UX 7.0	X Window System
IBM RISC System/6000	8MB	10MB Free Space	AIX version 3	X Window System
Sun/3, Sun/4	8MB	10MB Free Space	SunOS 4.0 or 4.1	X Window System

EDUCATIONAL AND TECHNICAL SUPPORT SERVICES

ParcPlace Systems offers a variety of educational services on Objectworks\Smalltalk and Object-Oriented Methodology for developers and managers. Both introductory and intermediate Objectworks\Smalltalk classes are available for developers. The Methodology Workshop includes project management; analysis and design; and a technology seminar.

ParcPlace software support services, called Objectsupport, include a hotline service, a technical newsletter, an electronic bulletin board, and discounted software updates. Consulting services are also available.



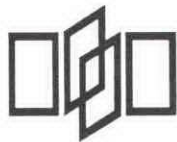
ParcPlace Systems

1550 Plymouth Street
Mountain View CA 94043
800.822.7880
415.691.6700 (within California)
Fax 415.691.6715

ParcPlace, Smalltalk-80 and Objectworks are registered trademarks and SPIM, Objectsupport and Portable Objects are trademarks of ParcPlace Systems, Inc. Other product names may be trademarks of their respective companies. Specifications subject to change without notice.
© 1990 ParcPlace Systems, Inc. All rights reserved. 9/90.

..... ***Objects Illustrated***

Applications & Services Catalog



ParcPlace

Objects Illustrated

Applications & Services Catalog

Objectworks for Smalltalk-80 Version 2.5

The descriptions were supplied by the applications developers themselves; copyright and trademark notices, or the lack thereof, are as submitted by the developers. We apologize for any errors or omissions.

Airline Marketing Tactician and BANKET are trademarks of BehavHeuristics, Inc. Apollo and DOMAIN are registered trademarks of Apollo Computers. IBM, IBM PC and IBM PC/AT are registered trademarks of International Business Machines. Draw-80 is a trademark of Knowledge Systems Corp. GemStone is a trademark of Servio Logic Corporation. Javelina is a trademark of the University of Illinois. Kyma is a trademark of Symbolic Sound Corporation. Macintosh and Macintosh II are trademarks of Apple Computer, Inc. PostScript is a registered trademark of Adobe Systems, Incorporated. R&D Analyst is a trademark of IDS Partners. Sun-3, Sun386i, Sun-4, and Sun Workstations are trademarks of Sun Microsystems, Inc. Tektronix 4310, 4400, 4405, 4406 is a trademark of Tektronix, Inc. UNIX is a registered trademark of AT&T. Xerox, Xerox 6085, 1186, and 1108, ANALYST, ASSISTANT, and HUMBLE are trademarks of Xerox Corporation. Objectworks and ParcPlace are registered trademarks of ParcPlace Systems, Inc. Objectsupport, ParcBench, ParcNotices, and Smalltalk-80 are trademarks of ParcPlace Systems, Inc. Other product names may be the trademarks of their respective companies. All specifications subject to change without notice.

© 1990 ParcPlace Systems, Inc. All rights reserved. 7/90

ParcPlace Systems, Inc.
1550 Plymouth Street
Mountain View, California 94043
(800) 822-7880

(415) 691-6700
(415) 691-6715 FAX



ParcPlace

Table of Contents

Applications and Tools for Objectworks for Smalltalk-80

<u>Page</u>	<u>Company Name</u>	<u>Product Name</u>
Accounting		
7	Georg Heeg	Financial Accounting System
CAD/CAM		
8	Ascent Logic	RDD System Designer
Database Interface Tools		
9	Georg Heeg	ISAM Toolbox
10	Servio Logic Corporation	Gemstone Smalltalk Interface
Development Environments and Tools		
11	Andreas Tonne	New Interface Framework
13	Clique Software	FrameWork
14	Clique Software	HyperDoc
15	Clique Software	Trace-Tool-Kit
16	Clique Software	VICK (Visual Construction Kit)
17	Georg Heeg	International Character Set Package
18	Georg Heeg	Terminal Emulation Package
19	Instantiations	Application Organizer 80
20	Knowledge Systems Corporation	Draw-80
21	MasPar Computer Corporation	MasPar Programming Environment

Table of Contents (continued)

<u>Page</u>	<u>Company Name</u>	<u>Product Name</u>
Development Environments and Tools (continued)		
22	Servio Logic Corporation	FACETS
24	Tigre Object Systems	Tigre Application Development Environment & Tigris Object Oriented Database Management System
Expert Systems		
25	Xerox Special Information Systems	Humble
Hypermedia		
26	Xerox Special Information Systems	Analyst
27	Xerox Special Information Systems	Assistant
Management Consulting		
28	Strategic Decisions Group	R&D Analyst
Music Applications		
29	Andreas Tonne	Trax
30	CERL Music Project	Javelina
31	Hyperscore User's Group	Doubletalk
31	Hyperscore User's Group	Hyperscore
32	Symbolic Sound Corporation	Kyma
Network Operations and Fault Management		
33	Avant-Garde	Net/Command
Office Automation		
34	Georg Heeg	Address and Letter System
35	Georg Heeg	Seminar Image and Textpage Package

Table of Contents (continued)

<u>Page</u>	<u>Company Name</u>	<u>Product Name</u>
Office Automation (continued)		
36	Taskon	Taskon/Exie
Office Automation -Analysis		
37	F B Informatik	MOBY
Resource Allocation		
38	BehavHeuristics, Inc.	Banket
Simulation Modeling		
39	Ciba Geigy and Georg Heeg	ProfiSEE
Consultants		
40	Instantiations, Inc.	
40	Knowledge Systems Corporation	
41	The McKenna Consulting Group	
41	The Object People Inc.	

Introduction

Welcome to the first edition of ParcPlace System's Applications & Services Catalog for Objectworks for Smalltalk-80. This Catalog profiles commercially available applications and services for Objectworks for Smalltalk-80. Listed applications include both stand alone applications and class libraries which require Objectworks for Smalltalk-80 development environment.

The applications span a range of industries from music to chemical pharmaceutical to management consulting to CAD/CAM. The class libraries were created to meet a variety of development needs including database interface, application development toolkits, office automation, hypermedia and expert systems.

ParcPlace Systems extends its appreciation to our value added partners for the software and services they provide.

Is there an application missing? Please let us know. To include a service in the catalog, please return the description (120 words or less) to ParcPlace Systems. To include an application in the catalog, please return the following information to ParcPlace Systems:

- a. Company name
- b. Product name
- c. Product description (120 words or less)
- d. Platforms
- e. Price
- f. Objectworks for Smalltalk-80 version
- g. Order information

Additionally, please answer the following questions regarding your application:

1. What is the size of the installed base? (number of installations or number of sites)
2. Is full source code included with the application?
3. What version of the application is currently shipping?
4. What application category is the product?
5. Is the application stand alone, or does it require Objectworks for Smalltalk-80?

Catherine Jaeger
Product Manager, Objectworks for Smalltalk-80
ParcPlace Systems Inc.
1550 Plymouth Street
Mountain View, CA 94043
Phone: (415) 691-6700
Fax: (415) 691-6715

Financial Accounting System

Georg Heeg

The Smalltalk-80 financial accounting system is designed to suit the needs of mid-sized companies. It provides two browsers on the accounting data: The journal view serves to edit the financial transactions. The account view is based on a set of numbered, named hierarchical accounts like provided by the Federal Republic of Germany IKR (Industriedontenrahmen). The hierarchy of accounts may be completely adapted.

- Input data is entered in German plain text and parsed by the system.
- Cross-referencing between items, bookings and journals.
- Profound error correction capabilities.
- Balancing and report writing tools are available.
- Data storage is based on Mr. Heeg's ISAM package
- The system is delivered with complete source code.

Platform availability:

Apple Macintosh
80386-based MS-DOS
Sun-3, Sun-4, or Sun386i
Hewlett-Packard HP 9000 Series 300
Apollo DOMAIN Series
DECstation 3100 and 5000
PCS Cadmus and Cadmus RCU

Pricing: 490,00 DM

For more information contact:

Mr. Georg Heeg
Smalltalk-80 Systems
Baroper Strasse 337
D-4600 Dortmund 50
Federal Republic of Germany
Phone: (x49-2 31)97500 10
Fax: (x49-2 31)97500 120
E-mail: georg@unido.(uucp,bitnet)

RDD System Designer

Ascent Logic

Ascent Logic Corporation develops and distributes products and services addressing System Design Automation. Requirements Driven Development (RDD) is Ascent's method that encompasses all aspects of requirements processing, system engineering and the capture of behavior. RDD's graphic language completely describes real, dynamic system behavior. The RDD System Designer is a tool for construction, management, display, and documentation of linked design objects that specify behavior. Documentation is produced with Mil-Std and custom reports. RDD provides distribution and merging of work, browsing and schemas with control. RDD has been applied to the modeling and specific of new and existing products and processes, in front of C.A.E. or C.A.S.E. RDD composes an interactive blueprint for architects of critical, complex, reliable systems.

Platform availability:

Apple Macintosh II
Apollo DN35xx and DN4xxx
Sun-3 and -4
DECstation 3100

Objectworks for Smalltalk-80 version: 2.3 for Macintosh and Apollo
2.5 for Sun and DECstation

For more information contact:

Ascent Logic Corporation
180 Rose Orchard Way, Suite 200
San Jose, California 95134
Phone: (408) 943-0630
Fax: (408) 943-0705

ISAM Toolbox

Georg Heeg

The ISAM toolbox implements a database manager for the well known ISAM (indexed sequential accessing method). It is written entirely in Objectworks for Smalltalk-80 and fits perfectly in its object-oriented programming paradigm. The DBM defines a collection of the ISAM items by mapping item keys into item records stored on disk. Several orderings on the records are defined by declaring different key classes, where the instances of a key class form an index.

- Full integration in Objectworks for Smalltalk-80
- No artificial limits on any sizes
- Special dense and fast storage of records
- Automatic crash recovery
- Easily extendable for user defined objects
- The system is delivered with complete source code

The records and keys used in the toolbox are composed from named, typed data fields. These fields support a fast and dense storage of the field's values in a binary format and assure database integrity through type checking. The sizes of these values are not fixed or limited.

Platform availability:

Apple Macintosh
80386-based MS-DOS
Sun-3, Sun-4, or Sun386i
Hewlett-Packard HP 9000 Series 300
Apollo DOMAIN Series
DECstation 3100 and 5000
PCS Cadmus and Cadmus RCU

Pricing: 90,00 DM

For more information contact:

Mr. Georg Heeg
Smalltalk-80 Systems
Baroper Strasse 337
D-4600 Dortmund 50
Federal Republic of Germany
Phone: (x49-2 31)97500 10
Fax: (x49-2 31)97500 120
E-mail: georg@unido.(uucp,bitnet)

The GemStone™ Smalltalk Interface (GSI)

Servio Logic Corporation

The GemStone™ Smalltalk Interface (GSI) provides a set of Smalltalk-80 classes that allow your Smalltalk-80 application to create and modify persistent objects in a GemStone Object-Oriented Data Management System. With the GSI, your Smalltalk-80 applications can share objects with other Smalltalk-80 or C applications on a variety of workstations. Concurrency control and transaction management are provided by a networked GemStone object server.

The GemStone Smalltalk Interface provides the following functions:

- sending database code to GemStone for execution
- sending messages to GemStone database objects
- translating objects between GemStone and Smalltalk
- examining and modifying the internal structure of GemStone database objects
- controlling GemStone sessions (logging in and out, managing transactions)
- recovering from errors

The GSI can be used in conjunction with the GemStone Object-oriented Data Management System to build comprehensive multiuser database applications. The GSI includes a Gemstone Browser which lets you examine and define database classes and methods, and a GemStone Workspace which enables you to execute database code interactively. Source code is not included.

Platform availability:

DECstation
Sun-3 and -4 workstations
Apple Macintosh
IBM PC and compatibles

Objectworks for Smalltalk-80 version: 2.5

For more information contact:

Servio Logic Corporation
1420 Harbor Bay Parkway, Suite 100
Alameda, California 94501-U.S.A.
Phone: (415) 748-6200

New Interface Framework

Andreas Tonne

This framework integrates new user-interface classes in to the existing Smalltalk-80 MVC-framework. This new interface supports directed update of dependencies through active variables that replace the traditional adapter protocols.

Active variables (values, constants) ease the implementation of complex MVC-structures by direct dependencies between components and speed up change propagation.

The new interface hierarchy includes modular classes for: buttons; sliders, gauges, scrollbars; listviews on text, paragraphs, display objects; and treeview on text and display objects.

Flexible dialog views implement non-disruptive views for file selectors (graphical), notifiers, and confirmers.

New menus support two dimensional arrangements of fields which contain arbitrary display objects. The fields may perform different actions like selections or Smalltalk-80 code depending on the state of modifier keys.

Platform availability:

Apple Macintosh
80386-based MS-DOS
Sun-3, Sun-4, or Sun386i
Hewlett-Packard HP 9000 Series 300
Apollo DOMAIN Series
DECstation 3100 and 5000

Pricing: 2990,00 DM workstation
1990,00 DM personal computer

For more information contact:

Mr. Georg Heeg
Smalltalk-80 Systems
Baroper Strasse 337
D-4600 Dortmund 50
Federal Republic of Germany
Phone: (x49-2 31)97500 10
Fax: (x49-2 31)97500 120
E-mail: georg@unido.(uucp,bitnet)

Net/Command *Avant Garde*

Net/Command is a powerful turnkey operations and fault management system for complex multi-vendor information systems and networks. *Net/Command* simplifies system and network operations by providing timely information and effective control at a single workstation. It provides an end-to-end view of network, host, and application performance, condition, and operations.

Features include unified surveillance, alert filtering and management, consol emulations, system databasing, color graphics and text reports, and system security.

Version 2.1 includes LAN and Host alarm management, real-time network maps, and threshold triggered and manually created alerts. Source code is not included.

Platform availability:

Sun 3/60, 3/80
Sun Sparc

Pricing: Starts at less than \$50,000

Objectworks for Smalltalk-80 version: 2.5

For more information contact:

Avant-Garde Computing
8000 Commerce Parkway
Mount Laurel, NJ 08054
Phone: (609) 778-7000
Fax: (609) 778-9580

Avant-Garde Computing, Europe
Clivemont Road
Maidenhead, Berkshire SL6 7BU
England
Phone: 44 (0) 628 771 902
Fax: 44 (0) 628 770 458

FrameWork *Clique Software*

FrameWork is a system to create object networks and graphical editors for these networks. Applications for which FrameWork can be used are music notation, production plants, hypermedia, or CAD. FrameWork is fully integrated in the Smalltalk-80 environment and provides many new classes with a well defined hierarchy. With this feature the user can develop and test prototypes for graphical applications quickly. With the shorter design-implementation-test-cycle, an application with networks will be produced essentially faster.

FrameWork provides:

- classes to represent and manage object networks
- classes to create object network editors

Platform availability:

Apple Macintosh
80386-based MS-DOS
Sun-3, Sun-4, or Sun386i
Hewlett-Packard HP 9000 Series 300
Apollo DOMAIN Series
DECstation 3100 and 5000

Pricing: 2990,00 DM workstation
1290,00 DM personal computer

For more information contact:

Mr. Georg Heeg
Smalltalk-80 Systems
Baroper Strasse 337
D-4600 Dortmund 50
Federal Republic of Germany
Phone: (x49-2 31)97500 10
Fax: (x49-2 31)97500 120
E-mail: georg@unido.(uucp,bitnet)

HyperDoc *Clique Software*

On-line documentation eases the work of programming systems and applications by providing a user guide integrated in the system. HyperDoc is a tool to create such documentation. The developer gets a tool to create text and additional graphics and store them together with keys. The user gets a tool to find the information quickly.

HyperDoc stores the information with keys to retrieve the information. By selecting a word or a text part, a new help is defined. For a help, window will open in which the user can write his help text and add pictures. All text styles from Smalltalk-80 are available.

- Automatic data storage: texts hold the textstyle and graphics will be compressed
- Different access possibilities to the keys
 - by selection in a text
 - by selection from a menu
- The keywords will be underlined in all texts automatically
- Any size for graphics by scrolling in a window
- On-line documentation creation
- Smalltalk-80 expression organization
- Text or graphic database

Platform availability:

Apple Macintosh
80386-based MS-DOS
Sun-3, Sun-4, or Sun386i
Hewlett-Packard HP 9000 Series 300
Apollo DOMAIN Series
DECstation 3100 and 5000

Pricing: 290,00 DM workstation
190,00 DM personal computer

For more information contact:

Mr. Georg Heeg
Smalltalk-80 Systems
Baroper Strasse 337
D-4600 Dortmund 50
Federal Republic of Germany
Phone: (x49-2 31)97500 10
Fax: (x49-2 31)97500 120
e-mail: georg@unido.(uucp,bitnet)

Trace-Tool-Kit

Clique Software

Trace tools help the user to analyze a program and to find bugs. In contrast to other debugging tools, Trace-Tool-Kit can visualize program execution. The user defines which parts of a program should be traced. Trace-Tool-Kit provides the main methods to create any tracer which analyzes the message passing from Smalltalk-80 applications.

Trace-Tool-Kit provides the tools and methods for the main functionality of a tracer. With a message oriented specification language, the user can trace his programs in text mode. Trace-Tool-Kit allows creation of any tracer.

The graphic tool kit provides tools for graphical specification and visualization of a trace. The user can specify the trace on a shown Object directly. The Smalltalk-80 tracer is an integration of the basic tool kit and the graphical tool kit into Objectworks for Smalltalk-80. The user can start the Smalltalk-80 tracer from the Smalltalk-80 tools (i.e. browser, debugger, inspector).

Platform availability:

Apple Macintosh
80386-based MS-DOS
Sun-3, Sun-4, or Sun386i
Hewlett-Packard HP 9000 Series 300
Apollo DOMAIN Series
DECstation 3100 and 5000

Pricing: 1490,00 DM workstation
1290,00 DM personal computer

For more information contact:

Smalltalk-80 Systems
Baroper Strasse 337
D-4600 Dortmund 50
Federal Republic of Germany
Phone: (x49-2 31)97500 10
Fax: (x49-2 31)97500 120
E-mail: georg@unido.(uucp,bitnet)

VICK (Visualization Construction Kit)

Clique Software

VICK is a Smalltalk-80 object visualization tool which supports knobs, sliders, bar charts, buttons, multiple choice menus, etc. to show objects to the user. These shown objects can be manipulated by using the components of the system. With VICK, users can create user interface components for different applications interactively.

VICK is fully integrated in the Smalltalk-80 environment and uses the features of the Smalltalk-80 system and VICK for its own user interface. Users can start VICK from other tools (i.e. browser, inspector). The components are integratable in other views according to the MVC-concept.

- Smalltalk-80 object visualization
- Simulation and program control
- User interface component creation by generate new messages automatically
- Automatic updating of manipulated objects
- Free size of all components and manipulation by mouse
- Attributes can be set directly (i.e. range, label, scale lettering, text style, etc.)

Platform availability:

Apple Macintosh
80386-based MS-DOS
Sun-3, Sun-4, or Sun386i
Hewlett-Packard HP 9000 Series 300
Apollo DOMAIN Series
DECstation 3100 and 5000

Pricing: 1490,00 DM workstation
1290,00 DM personal computer

For more information contact:

Mr. Georg Heeg
Smalltalk-80 Systems
Baroper Strasse 337
D-4600 Dortmund 50
Federal Republic of Germany
Phone: (x49-2 31)97500 10
Fax: (x49-2 31)97500 120
E-mail: georg@unido.(uucp,bitnet)

International Character Set Package

Georg Heeg

The International Character Set Package provides all characters of common alphabets in Western Europe. It can be added to the standard Smalltalk-80 programming system and especially to the Analyst.

The character encoding is compatible with the XEROX Character Code Standard. The string concept follows ISO 6937.

It provides a 256 character set including the well known ascii characters. It adds some special characters and 14 diacritical characters, which can be composed with any base character.

You can use the extended character set in the programming system and for typing documents in the Analyst. Every character can be created from a normal keyboard using a function key ("Compose Key") which can be assigned to any key on the keyboard.

- All Western European characters.
- Compatible with XEROX Character Code Standard.
- Reachable from normal keyboard.
- Postscript support.
- The system is delivered with the full source code.

Platform availability:

Apple Macintosh
80386-based MS-DOS
Sun-3, Sun-4, or Sun386i
DECstation 3100 and 5000
PCS Cadmus and Cadmus RCU

Pricing: 490,00 DM

For more information contact:

Mr. Georg Heeg
Smalltalk-80 Systems
Baroper Strasse 337
D-4600 Dortmund 50
Federal Republic of Germany
Phone: (x49-2 31)97500 10
Fax: (x49-2 31)97500 120
E-mail: georg@unido.(uucp,bitnet)

Terminal Emulation Package

Georg Heeg

The Smalltalk-80 Terminal Emulation Package provides emulations of the DEC VT100 and VT220 terminals, as well as the necessary framework to build emulations of other character-oriented terminals.

The pre-programmed terminal emulations (DEC VT100 and VT220) have the following features:

- full character attributes (bold, underlined, reversed, linking), double height and width characters, pseudo-graphics characters, function keys,
- 80 and 132 column mode,
- scrolling regions, smooth scrolling
- save/restore cursor,
- copy & paste between terminals windows and other Smalltalk-80 windows.

In addition, the VT220 emulation supports 8-bit character input and output and host access to the function keys.

Terminal windows can be connected either to serial lines, to local processes (via pseudo-ttys) and to servers on local and wide area networks (via socket connections). The class structure is extensible to accommodate different of terminal emulations and devices. The package is delivered as fully commented Smalltalk-80 source.

Platform availability:

Apple Macintosh
80386-based MS-DOS
Sun-3, Sun-4, or Sun386i
Hewlett-Packard HP 9000 Series 300
Apollo DOMAIN Series
DECstation 3100 and 5000
PCS Cadmus and Cadmus RCU

Pricing: 490,00 DM

For more information contact:

Mr. Georg Heeg
Smalltalk-80 Systems
Baroper Strasse 337
D-4600 Dortmund 50
Federal Republic of Germany
Phone: (x49-2 31)97500 10
Fax: (x49-2 31)97500 120
E-mail: georg@unido.(uucp,bitnet)

Application Organizer 80

Instantiations

Application Organizer 80 (Version 1.0) is a set of development tools giving programmers a major advantage in the management and efficient delivery of Objectworks for Smalltalk-80 applications. The set includes tools that help organize the elements of an application, manage changes to the standard image, and minimize application size and complexity. Application Organizer 80 includes:

- The Image Generator: allows you to designate required classes and methods, and generate an image containing only your application.
- The Package Browser: allows you to create, view and manipulate "packages" (the elements that comprise your application).
- The Package Organizer: allows you to organize changes within and among packages.
- The Inheritance Browser: enhances the standard Objectworks for Smalltalk-80 browser by providing the capability to directly view the inherited behavior of a class.
- Workspace Enhancements: eliminate the need for global variables or multiple inspectors when experimenting with, or testing, code.
- Complete source code is included.

Platform availability:

80386 MS-DOS

Apple Macintosh

Call for other platform availability. Available June, 1990.

Pricing: \$195. for single-user systems.

Objectworks for Smalltalk-80 version: 2.5

For more information contact:

Instantiations, Inc.

921 SW Washington Suite 312

Portland, Oregon 97205

Phone: (503) 242-0725

Fax: (503) 242-0729

Draw-80™
Knowledge Systems Corporation

Draw-80™ is an object-based graphics tool that provides most of the functions of the popular Apple Macintosh applications: MacDraw™ and MacDraft™. Draw-80 features include:

- Tool panels, pull-down menus, and on-line help provided.
- 70 patterns for lines, borders, and filled polygons, including a user defined pattern.
- 24 graphics definition tools providing graphics primitives.
- Text editing, font emphasis, coloring, justification control, and alignment features.
- User-definable line thickness and brush shape.
- Real-time scrolling of large documents.
- Copy, cut and paste operations using a buffer shared among all Draw-80 based applications.
- Pick and drag placement, reshaping, scaling, and rotation of graphic entities, with optional entry via keyboard. Accept/cancel options for transformations.
- Free angle rotation of all graphics entities and 90 degree rotation of text and Smalltalk forms.
- PostScript printing and Smalltalk-80 form output supported.

Smalltalk-80 classes provided in Draw-80 can be integrated into other Smalltalk-80 applications to support dynamic graphics. Draw-80 comes with a user manual detailing the many features of the graphics editor as well as information on how to use the Draw-80 classes in other Smalltalk-80 applications. Full source code is included.

Platform availability:

Apple Macintosh
Sun-3 and Sun-4
Hewlett-Packard HP 9000 Series 300
Tektronix 4300 and 4400 Series
IBM PC/AT

Pricing: \$499. for single user systems.

Objectworks for Smalltalk-80 version: 2.5

For more information contact:

Knowledge Systems Corporation
114 MacKenan Drive, Suite 100
Cary, NC 27511-U.S.A.
Phone: (919) 481-4000

MasPar Programming Environment
MasPar Computer Corporation

MasPar's MP-1 Family of data-parallel computers provides a new level of price/performance in the high-performance computing arena. Systems range in price from \$170,000 to \$810,000 with performance up to 26,000 MIPS and 1300 MFLOPS. The system has a small footprint and fits easily into an office or lab environment with no special requirements. The MP-1 incorporates a VAXstation™ 3520 running ULTRIX™. The VAXstation manages the user interface, program execution, and network communications for the system. MasPar offers a variety of language options including:

- MasPar Fortran which is based on Fortran 77 and allows access to the data-parallel system through Fortran 90 extensions,
- MasPar Application Language (MPL) which is based on K&R C and allows direct hardware control. MasPar Programming Environment (MPPE), a powerful graphical debugging system, has been built using Objectworks for Smalltalk-80. MPPE features an interactive, point-and-select window-based system that allows source-level debugging as well as graphical visualization of machine and data status.

Platform availability:

MP-1

For more information contact:

Krishnan Aghoramurthy
MasPar Computer Corporation
749 North Mary Avenue
Sunnyvale, California 94086
Phone: (408) 736-3300
Fax: (408) 736-9560

FACETS

Servio Logic Corporation

Servio has developed a toolkit called Facets for Objectworks for Smalltalk-80, consisting of four tools: Screen Painter, Menu Builder, Report Writer, and Help Author. Those tools can be used to build complete Smalltalk-80 applications, which can either run locally or be linked to GemStone. Facets' source code is included to give developers ever greater flexibility. All tools include a library of commonly-used blocks of code that developers can share and reuse.

Screen Painter

The Screen Printer allows you to dynamically construct enter/update forms that access data stored either locally, in GemStone, or in a relational database supported by GemStone.

Its features include:

- Flexible forms-based interface development tools
- Dictionary-driven data interface for easy form maintenance and creation
- Powerful, built-in data types including Currency, Date, Time, and Boolean
- Context-sensitive links to help systems generated by the Help Author for quick access to hypertext help

Menu Builder

The Menu Builder allows you to design easy-to-use front ends for your applications. Its features include:

- Horizontal and vertical menus for flexible interface
- Button-driven interface with mouse support for ease-of-use
- Easy access to forms, reports, and help systems built with Facets

Report Writer

The Report Writer allows you to design hard-copy reports. Its features include:

- Near-WYSIWYG report design mode
- Headers, footers, calculated fields, automatic totaling, and justification
- Landscape, portrait, and variable-columned printing for greater flexibility
- Dictionary-driven data interface for easy report maintenance and creation

Help Author

The Help Author allows you to design and build on-line, hypertext-based help systems. Its features include:

- Context-sensitive links from Screen Painter forms to any help system topic
- Hypertext links between topics that lead the user to related help information at the click of a mouse
- Help topic hierarchy that automatically organizes the help information
- Print function that allows you to create hard copy manuals from the hypertext information
- Customized hypertext topic paths, Pop-up notecards, bookmarks, and breadcrumbs

Platform availability:

All platforms supported by Objectworks for Smalltalk-80

Objectworks for Smalltalk-80 version: 2.5

For more information contact:

Servio Corporation
1420 Harbor Bay Parkway, Suite 100
Alameda, California 94501 U.S.A.
Phone: (415) 748-6200
Fax: (415) 748-6227

Tigre Application Development Environment
Tigris Object Oriented Database Management System
Tigre Object Systems

Tigre is a powerful, intuitive, and flexible application development environment for Objectworks for Smalltalk-80. Tigre provides an interactive interface construction system similar to those found in NextSTEP, or HyperCard, while incorporating the power of Objectworks for Smalltalk-80 for doing the actual programming behind the interface. Tigre is bundled with Tigris, our fully object-oriented multi-user database system.

Tigre allows the easy construction of new applications and interface-items, which can be wired together to form hypermedia or traditional applications. Tigre also allows the creation of sophisticated new user-interface items, such as animation views, enhanced text views, animated buttons, and other new views which can be easily incorporated into applications. Tigre provides an interactive screen editor for manipulating views, and dialogs are used for editing view parameters. To maximize programmer efficiency, running programs may be modified on the fly, with no penalties. Completely removing any kind of recompile time greatly enhances programmer efficiency and creativity. Further, the Tigre system itself can be easily customized by developers. Tigris is our fully object-oriented database, providing indexing, transaction logging, multi-user access, and error recovery. Tigris also incorporates a highly efficient Frame system for providing inheritance for database items. Tigris allows users to create their own object-oriented databases, and is used internally by Tigre to store fonts, icons, and screen descriptions. Beta begins June 15, 1990. Version 1.0 release is Q3/Q4 1990. Full source code is included.

Platform availability:

Apple Macintosh
Sun-3 and -4 workstations

Objectworks for Smalltalk-80 version: 2.5

For more information contact:

Tigre Object Systems
11 Glen Canyon Rd
Santa Cruz, CA 95060
Phone: (408) 426-5308
Fax: (408) 479-4196
E-mail: ucscclveda!tigre@ucbvax.berkeley.edu

HUMBLE™

Xerox Special Information Systems

HUMBLE™ is an expert system shell for the Smalltalk-80 environment. The implementation and features of HUMBLE demonstrate the tremendous leverage provided by the purely object-oriented Smalltalk-80 environment. HUMBLE supports forward and backward chaining, redirectable I/O, and multiple knowledge bases. Full source code is included, and the programmer's interface is documented.

Platform availability:

Apple Macintosh
Sun-3 and -4 Series
Xerox 6085, 1186, and 1108
Apollo DOMAIN Series
Sun386i
80386 MS DOS

Pricing: Macintosh and PC: \$395.
Other platforms: \$995.

Educational, quantity, dealer, and VAR discount schedule available upon request.

Objectworks for Smalltalk-80 version: 2.5

For more information contact:

Vista Software Products
Xerox Special Information Systems
P.O. Box 5608, Pasadena, CA 91107-U.S.A.
Phone: (818) 351-2351 (in California)
(800) 233-9055 (outside of California)

ANALYST™ Workstation System **Xerox Special Information Systems**

The ANALYST™ Workstation System is an integrated information analysis environment suitable for end-users and application developers. Features include Information Centers, maps and images, business graphics, spreadsheets, personal databases, forms, word processing, and interpress and postscript printing. Existing applications can be customized and new ones can be added with ease. Full source code is included in the Developer Environment. Version 3.2 provides new capabilities, most notably a on-screen linking between or across windows. Comprehensive reference manuals are included. The runtime version includes the Assistant, as well as the runtime version of Objectworks for Smalltalk-80.

Platform availability:

Apple Macintosh
Sun-3 and -4 Series, Sun386i
Xerox 6085, 1186, and 1108.
Apollo DOMAIN Series
80386 MS DOS

Pricing: PC and Macintosh: \$1095. for development environment,
\$1245. for runtime version.

Other platforms: \$1995. for development environment,
\$2590. for runtime version.

Educational, quantity, dealer, and VAR discount schedule available upon request.

Objectworks for Smalltalk-80 version: 2.5

For more information contact:

Vista Software Products
Xerox Special Information Systems
P.O. Box 5608, Pasadena, CA 91107-0608
Phone: (818) 351-2351 (in California)
(800) 233-9055 (outside of California)

ASSISTANT™
Xerox Special Information Systems

ASSISTANT™ is an add-on to ANALYST. It allows ANALYST Information Centers to search for patterns that the user defines, create new links based on those patterns, and inform the user about these new links. Possible applications include tasks for which complex relationships among large collections of information must be tracked and analyzed. ASSISTANT V1.1 requires ANALYST V3.2.

Platform availability:

Apple Macintosh
Sun-3 and -4 Series, Sun386i
Xerox 6085, 1186, and 1108.
Apollo DOMAIN Series
80386 MS DOS

Pricing: \$495

Educational, quantity, dealer, and VAR discount schedule available upon request.

Objectworks for Smalltalk-80 version: 2.5

For more information contact:

Vista Software Products
Xerox Special Information Systems
P.O. Box 5608, Pasadena, CA 91107-0608
Phone: (818) 351-2351 (in California)
(800) 233-9055 (outside of California)

R&D Analyst ***Strategic Decisions Group***

Strategic Decisions Group specializes in decision consulting, which is a process that helps decision-makers arrive at quality decisions on a timely basis. By developing their options, clarifying the potential consequences of their actions and understanding their preferences, decision-makers can choose and act with confidence--even in the face of complexity, uncertainty and difficult trade-offs.

For more than 30 years, the Strategic Decisions Group staff has pioneered decision consulting and its underlying discipline: decision analysis. Decision analysis is applicable to any human endeavor and has been used to assist decision-makers throughout business and industry. It also applies to everyday decision-making such as patients and physicians making medical decisions. Strategic Decisions Group has lead the field of decision consulting by pioneering the automation of decision analysis. The resulting technology--Intelligent Decision Systems--makes decision analysis available to a much wider range of decision-makers than ever before possible.

R&D Analyst is SDG's first commercial intelligent decision system (IDS). R&D Analyst assists managers, scientist and engineers in making decisions about R&D projects and portfolios. The system interacts with the decision-maker to actively formulate, evaluate and appraise the client's decision model and its alternatives.

The IDS will serve as a platform for building custom intelligent decision systems for SDG's clients. Designed to be used on a Macintosh computer on the executive or manager's desk, the IDS concentrates on a clear and thorough formulation of the problem and evaluates the optimal strategy based on this model. The process is iterative; if the decision-maker is not comfortable with the resulting decision, the model is updated and refined to include better information and to examine genuine performances.

Platform availability:

Apple Macintosh

Objectworks for Smalltalk-80 version: 2.3

For more information contact:

Strategic Decisions Group
2440 Sand Hill Road
Menlo Park, California 94025-6900
Phone: (415) 854-9000
Fax: (415) 854-6718

Trax (MIDI - Laboratory #1)

Andreas Tonne

Being the first part of a series of MIDI tools, Trax implements a real-time sequencer for MIDI events.

Trax consists of several layers:

- The primitive layer implements an interface for MIDI user-primitives and maps the byte data to MIDI events.
- The module 'music notation #1' supports time oriented classes like TimestampInterval, TimeSignature, Tempo, etc.
- The sequencer layer implements a flexible and fast sequencer for an associated track hierarchy. These tracks are a kind of collection of MIDI events on arbitrary but fixed instruments. Any number of tracks with unlimited MIDI events can be managed by the sequencer.

The MIDI hardware is represented by a collection of instrument descriptions which reflect the capabilities of the real instruments. MIDI setups describes the initial state of instruments for a particular track.

- The interface layer contains various browsers and editors on the MIDI data.
- The module 'External representation #1' implements a mapping of arrangements, songs and tracks to the MIDI-standard file standard.
- The system is delivered with the full source code

Platform availability:

Apple Macintosh /
80386-based MS-DOS
Sun-3, Sun-4, or Sun386i
Hewlett-Packard HP 9000 Series 300
Apollo DOMAIN Series
Atari ST CPU replacements like Hyper Cache are highly recommended.

Pricing: 881,84 DM. Trax is currently in beta test.

For more information contact:

Mr. Georg Heeg
Smalltalk-80 Systems
Baroper Strasse 337
D-4600 Dortmund 50
Federal Republic of Germany
Phone: (x49-2 31)97500 10
Fax: (x49-2 31)97500 120
E-mail: georg@unido.(uucp,bitnet)

Javelina™
CERL Music Project

Javelina™ is a package used for designing digital signal processing software based on descriptions of algorithms based on their time-, frequency- or complex-domain behavior. The tools translate discrete-time systems specified using symbolic mathematical notation into algorithms that can then be used to generate machine language programs in a virtual digital signal processing machine code that can be translated to the native assembly language for any of several digital signal processors.

Platform availability:

Apple Macintosh
Sun-3 and -4 workstations
IBM PC/AT and compatibles
Tektronix 4400 Series
Apollo DOMAIN workstations

Pricing: Write for information.

For more information contact:

Kurt Hebel
252 Engineering Research Lab
103 S. Mathews
University of Illinois
Urbana, IL 61801-2977-U.S.A.
Phone: (217) 333-0766

DoubleTalk and HyperScore ***Hyperscore User's Group***

The DoubleTalk package is a structural music notation system based on the use of marked Petri nets called Predicate-Transition Networks (PrT Nets) that can be edited and animated to produce scores (note lists).

The HyperScore system is a toolkit for designing new notations and building computer software for editing, simulating and animating them. It includes a set of classes for representing basic music-structural elements (such as chords, ostinati or motives), a language for building scores by sending commands to these elements in real-time, and a set of graphical editor tools for note object and event-list editors.

Platform availability:

Apple Macintosh
IBM PC and compatible
Sun-3 and -4 workstations
Hewlett-Packard HP 9000 Series 300 workstations
Apollo DOMAIN workstations

Pricing: Contact Hyperscore User's Group

For more information contact:

DoubleTalk and Hyperscore
Hyperscore User's Group
P.O. Box 60632
Palo Alto, CA 94301
Email: hyperscore-users@ParcPlace.com

Kyma™
Symbolic Sound Corporation

Kyma is an environment for the specification and manipulation of digital audio signals. In Kyma, each object on the display represents a Sound that can be auditioned, altered, or used in the construction of another Sound. A digital signal processor (DSP) is used to provide audio feedback.

Objectworks for Smalltalk-80 was chosen for its flexibility, portability across platforms, excellent debugging facilities, and its elegance as a language. Kyma has been under development for three years by two developers with no prior Smalltalk experience.

The system is currently in Beta test at several sites and will be available by October 1990. The Kyma system includes both the stand-alone software and the digital signal processing hardware (including the appropriate interface card for the host computer).

Platform availability:

Macintosh II, SE/30, IIfx, IIfx and IIfx.

For more information contact:

Carla Scaletti
Symbolic Sound Corporation
P.O. Box 2530 Station A
Champaign, IL 61825-2530
Phone: (217) 328-6645

Net/Command Avant-Garde

Net/Command is a powerful turnkey operations and fault management system for complex multi-vendor information systems and networks. Net/Command simplifies system and network operations by providing timely information and effective control at a single workstation. It provides an end-to-end view of network, host, and application performance, condition, and operations. Features include unified surveillance, alert filtering and management, console emulations, system databasing, color graphics and text reports, and system security. Version 2.1 includes LAN and Host alarm management, real-time network maps, and threshold triggered and manually created alerts. Source code is not included.

Platform availability:

Sun 3/60, 3/80
Sun Sparc

Pricing: Starts at less than \$50,000

Objectworks for Smalltalk-80 version: 2.5

For more information contact:

Avant-Garde Computing
8000 Commerce Parkway
Mount Laurel, NJ 08054
(609) 778-7000
FAX (609) 778-9580

Avant-Garde Computing, Europe
Clivemont Road
Maidenhead, Berkshire SL6 7BU
England
Phone: 44 (0) 628 771 902
Fax: 44 (0) 628 770 458

Address and Letter System

Georg Heeg

The address and letter system is an office system to provide functionality for address management, simple word processing and automatic mail creation.

Addresses are entered in plain text and parsed by the software according to a standard address format. This format supports country dependent salutations. Each address is associated with a set of plain text comments which may be used to store phone numbers, e-mail addresses and arbitrary text.

Mail texts are entered in standard Smalltalk-80 format. They may contain Smalltalk-80 expressions to incorporate parts of the selected address or even to do some arbitrary computation when the mail text is formatted.

- Free address formats.
- 'Dynamic' mail texts that allow personalized letters.
- Multi-function browser.
- Easy to use address filters.
- Almost wysiwyg mail texts.
- Postscript and Atari laserprinter support.
- Data storage based on our ISAM package.
- The system is delivered with the full source code.

Platform availability:

Apple Macintosh
80386-based MS-DOS
Sun-3, Sun-4, or Sun386i
Hewlett-Packard HP 9000 Series 300
Apollo DOMAIN Series
DECstation 3100 and 5000
PCS Cadmus and Cadmus RCU

Pricing: 490,00 DM

For more information contact:

Mr. Georg Heeg
Smalltalk-80 Systems
Baroper Strasse 337
D-4600 Dortmund 50
Federal Republic of Germany
Phone: (x49-2 31)97500 10
Fax: (x49-2 31)97500 120
E-mail: georg@unido.(uucp,bitnet)

Seminar Image and TextPage System

Georg Heeg

The Seminar Image and TextPage System is a simple graphics and text formatter written within the Smalltalk-80 system that has been used at the University of Dortmund (Germany) for the preparation of various course notes and interactive on-line Smalltalk-80 system tutorials.

Platform availability:

Apple Macintosh
PCS-Cadmus
Atari Mega-ST
Sun-3 and -4 workstations
Hewlett-Packard HP 9000 workstations
Apollo DOMAIN workstations

Pricing: Write for information.

For more information contact:

Mr. Georg Heeg
Smalltalk-80 Systems,
Baroper Strasse 337
D-4600 Dortmund 50
Federal Republic of Germany
Phone: (x49-2 31)97500 10
Fax: (x49-2 31)97500 120
E-mail: georg@unido.(uucp,bitnet)

Taskon/Exie - Executive Information Environments

Taskon

Taskon A/S offers a set of consultancy services together with software and hardware for creating individual customized information environments for managers on different levels. The product is called TASKON/Exie. TASKON/Exie provides the manager with customized access to information and customized task-oriented tools for working with the information. The TASKON/Exie tools utilize existing customer systems through integration bridges. Further, the TASKON/Exie tools support teamwork through easy distribution of information together with suitable tools. The distribution of information and tools can be done ad hoc or according to a defined work process. Information can be aggregated in many levels to different managers or manager groups.

Platform availability:

all platforms supported by Objectworks for Smalltalk-80 version 2.5

Pricing: TASKON/Exie is delivered through customer projects, and is customized to each customer and customer users. The price depends on the degree of customization.

Objectworks for Smalltalk-80 version: 2.5

For more information contact:

Gro Oftedal
Gaustadalleen 21
0370 Oslo3
Norway
Phone: +47 2 95 85 00
Fax: +47 2 60 44 27

MOBY
F B Informatik

The MOBY (Modellierung von Buerosystemen - Modeling of Office Systems) package is a large tool for analysis of office automation processes that uses marked Petri nets (PrT nets) for the modeling of complex procedures. MOBY includes editors for designing and marking complex PrT nets and then performing analysis, simulation or other processing with these models. Further developments of the MOBY and PeWo Petri Net Simulation tools from Dortmund and Oldenburg are found in the new Graph and Net tools called the Olli image. The basic graphical structure editors support directed graphs, state diagrams, BE Nets, KI Nets, Kausal Nets and PrT Nets. The tools for graph and net editing and simulation are extensible to other representations such as music event lists or real-time control commands.

Platform availability:

Apple Macintosh
PCS-Cadmus
Atari Mega-ST
Sun-3 and -4 workstations
Hewlett-Packard HP 9000 workstations
Apollo DOMAIN workstations

Pricing: Write for information.

For more information contact:

Prof. Dr V. Klaus
F B Informatik
University of Oldenburg
PF 2505
D-2900 Oldenburg
Federal Republic of Germany

BANKET™
BehavHeuristics, Inc.

BANKET, the Behavioral Adaptive Network Knowledge Engineering Tool, is a proprietary software system that implements a behavior analytic approach to building adaptive networks. BANKET can be applied successfully to problems in which knowledge is not easily reducible to rules or formulas, as well as to situations where change is a fact of life and other static forms of knowledge will not suffice. BehavHeuristics's first commercial product based on BANKET, the Airline Marketing Tactician, addresses a major problem faced by the airline industry, namely the flight-by-flight "inventory management" of seats allocated to different discount and full-fare ticket classes, and the setting of overbooking limits to compensate for passengers who book seats but are "no-shows" at departure time.

Platform availability:

Apple Macintosh
Sun-3 and -4 workstations
IBM PS/2, 386, 486 and compatibles
Tektronix 4400 Series
Apollo DOMAIN workstations

Objectworks for Smalltalk-80 version: 2.5

For more information contact:

BehavHeuristics, Inc.
335 Paint Branch Drive
College Park, MD 20742-U.S.A.
Phone: (301) 454-0311

ProfiSEE
Ciba Geigy and Georg Heeg

ProfiSEE, pronounced prophecy, is a tool for Professionals and Profit in industry. It has been specifically developed for chemical pharmaceutical industry to gain insight through SEEing a plant or productive unit in operation - hence ProfiSEE. The system grew out of a specific application of Visual Interactive Simulation Modeling (VISM) in dyestuff production. ProfiSEE fosters close cooperation between an Operation Research group and its clients through improved communication with VISM as vehicle. It embodies the core principles of VISM and makes several concrete steps in the direction of graphical programming. ProfiSEE's implementation is at the forefront of software engineering technology.

ProfiSEE is tailor made for improving the turnaround time and quality of simulation projects in the chemical-pharmaceutical industry. It applies Pareto's rule to modeling: 80% of the model development effort has been replaced by graphical programming. This requires graphical editors for two tasks: 1) creating and manipulating product recipes, production schedules, and factory layouts; and 2) creating icons representing different classes of equipment, and forms (patterns) representing the state of equipment.

Besides making the modeling process more efficient and reducing the possibility of erroneous model input, this approach gives the users, usually chemists or production managers, insight into the underlying logic of the model as well as the different input data driving the model.

An experiment in ProfiSEE is driven by several input components which have been segmented in a very natural manner:

- intermediate products - which intermediates are available, where and in which form;
- product recipes - what quantities, time, equipment and personnel are required at which stage; and
- factory layouts - which resources, devices and connections are available where and when.

Pricing: ProfiSEE is available upon special request.

For more information contact:

Mr. Georg Heeg
Smalltalk-80 Systems
Baroper Strasse 337
D-4600 Dortmund 50
Federal Republic of Germany
Phone: (x49-2 31)97500 10
Fax: (x49-2 31)97500 120
E-mail: georg@unido.(uucp,bitnet)

Instantiations, Inc.

Instantiations is a leading supplier of object-oriented technology services including consulting, training, programming, custom engineering, and advanced development. The company has extensive experience with Smalltalk-80 and other object technologies and has worked with clients world-wide on successful projects including object-oriented designs, prototypes, and application development. Standard course offerings include Designing Object-Oriented Software, Smalltalk and C++ programming. Customized and on-site training is available. Senior members of instantiations' staff have been leaders in the development and use of Smalltalk-80 since 1980, and all staff members have 5 - 10 years of Smalltalk application development experience.

Mike Taylor
Instantiations, Inc.
921 S.W. Washington
Suite 312
Portland, OR 97205
Phone: (503) 242-0725
Fax: (503) 242-0729

Knowledge Systems Corporation

Rapid development services for complex software applications. For applications such as: manufacturing process and design control, product development, information modeling, project screening, simulation environments, and graphical programming environments. The Rapid Prototyping Center - Provides KSC's clients with object-oriented analysis, design, and programming services aimed at reducing software life cycle cost through the rapid evolution of functional, low cost prototypes. Smalltalk Apprentice Program - Offering analysis, design, and development training which quickly builds internal Smalltalk experts.

Knowledge Systems Corporation
114 MacKenan Drive, Suite 100
Cary, NC 27511-U.S.A.
Phone: (919) 481-4000

The McKenna Consulting Group

The McKenna Consulting Group provides services to engineering and management in Object-Oriented Technology. Management services include introductory seminars on the technology and feasibility assessments. Engineering services include Object-Oriented analysis and design as well as prototype implementations in Smalltalk. Also available are design audits which are independent reviews of ongoing work to improve the quality and suitability of existing designs. Jeff McKenna is the principle of The McKenna Consulting Group. Jeff has been involved with the technology for over 8 years and is an active member of the object-oriented community.

The McKenna Consulting Group
8825 SE Umatilla Street
Tualatin, Oregon 97062
Phone: (503) 692-1889
Fax: (503) 692-3194

The Object People Inc.

The Object People Inc. offer specialized training and consulting services in object-oriented technology tailored to Smalltalk-80 and Smalltalk/V. Courses include both entry level and advanced courses in Smalltalk. A three-day entry level course is designed for experienced software practitioners who are new to Smalltalk and to the Object-Oriented Programming Paradigm. A three day advanced course is offered in the design of interactive, graphical applications using the model-view-controller paradigm. The Object People also offer courses in object-oriented fundamentals, object-oriented design and object-oriented software engineering. Course enrollment is limited to ensure quality and to encourage active participation by the student. Hands-on exercises reinforce lecture sequences and provide practical experience in Smalltalk programming. Courses are normally offered on-site at the convenience of the client.

The Object People Inc.
91 Second Avenue
Ottawa, Canada K1S2H4
Phone: (613) 230-6897

Company Index

<u>Page</u>	<u>Company Name</u>	<u>Product Name</u>
11	Andreas Tonne	New Interface Framework
29	Andreas Tonne	Trax
8	Ascent Logic	RDD System Designer
33	Avant-Garde	Net/Command
38	BehavHeuristics, Inc.	Banket
30	CERL Music Project	Javelina
39	Ciba Geigy and Georg Heeg	ProfiSEE
13	Clique Software	FrameWork
14	Clique Software	HyperDoc
15	Clique Software	Trace-Tool-Kit
16	Clique Software	VICK (Visual Construction Kit)
37	F B Informatik	MOBY
34	Georg Heeg	Address and Letter System
7	Georg Heeg	Financial Accounting System
17	Georg Heeg	International Character Set
9	Georg Heeg	ISAM Toolbox
35	Georg Heeg	Seminar Image and Textpage
18	Georg Heeg	Terminal Emulation Package
31	Hyperscore User's Group	Doubletalk

Company Index (continued)

<u>Page</u>	<u>Company Name</u>	<u>Product Name</u>
31	Hyperscore User's Group	Hyperscore
19	Instantiations	Application Organizer 80
40	Instantiations, Inc.	Consulting Services
20	Knowledge Systems Corporation	Draw-80
40	Knowledge Systems Corporation	Consulting Services
21	MasPar Computer Corporation	MasPar Programming Environment
22	Servio Logic Corporation	FACETS
10	Servio Logic Corporation	Gemstone Smalltalk Interface
28	Strategic Decisions Group	R&D Analyst
32	Symbolic Sound Corporation	Kyma
36	Taskon	Taskon/Exie
41	The McKenna Consulting Group	Consulting Services
41	The Object People Inc.	Consulting Services
24	Tigre Object Systems	Tigre Application Development Environment Tigris Object Oriented Database Management System
25	Xerox Special Information Systems	Humble
26	Xerox Special Information Systems	The Analyst
27	Xerox Special Information Systems	The Assistant

ParcPlace Systems

Navigator

Navigator Framework

Navigator™ is a graphical framework in which to find, access, and edit information. It is especially suitable as a user interface to information where that information is represented as hierarchical relationships. The functionality of the navigation framework includes an iconic point-and-select approach to traversing pathways of related information. It includes text and image editors and methods to quickly return to prior places visited. The navigation framework can be applied, for example, to accessing hierarchical file systems, system documentation, and applications.

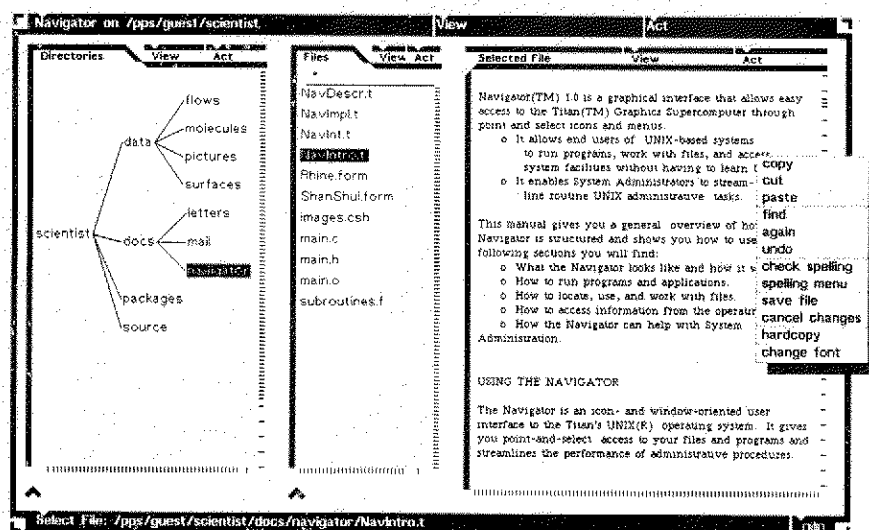
The first example customization of the Navigator Framework has been developed by ParcPlace for use with the UNIX® operating system on the Ardent Titan™ graphics supercomputer. By substituting an interactive, graphically oriented interface for text-based commands, Navigator makes it easier, faster, and more intuitive for users and system administrators to use the UNIX file system and do system administration tasks.

Navigator Example: Visual interface to the UNIX operating system

As an example use of the navigation framework, the Navigator for UNIX makes the full power of the operating system accessible to users without forcing them to learn the complex UNIX text-based command scheme.

For *users* of UNIX-based applications, the Navigator makes UNIX itself transparent. Users benefit from easy and flexible access to UNIX file creation, naming, and location. The user can quickly learn to navigate around the system.

For *system administrators* of UNIX-based systems, the Navigator provides assistance in many of the time-consuming and routine tasks of basic system administration.



Using the Navigator for UNIX, directories, file structures, and text editor can be visually displayed.

Functionality

Navigator for UNIX fulfills these basic functions. It hides UNIX while supplying full UNIX capabilities. It simplifies UNIX system administration tasks. Future plans include a toolkit for associating icons with UNIX scripts, applications, and system objects.

Features for Users

With its visual access to the UNIX file system, Navigator makes it easy to:

- Execute programs
- Create, copy, link, move, delete, rename, change ownership, and change the protection status of files and directories
- Edit files from their iconic representation

Features for System Administrators

The Navigator for UNIX also provides powerful, graphically oriented tools for system administration, making time-consuming tasks quick and easy with its point-and-select style of use.

- Manage user accounts
- Configure system utilities
- Manage groups and protections
- Update system and network configuration files
- Monitor system load and activity
- Perform file system backup and backup retrieval

Information

To receive further information on Navigator Framework and Navigator for UNIX, please contact ParcPlace Systems.



ParcPlace Systems

2400 Geng Road
Palo Alto, CA 94303
Toll Free: 1 800 822 5T80
(800 822 7880)
in California: 415 859 1000

Navigator is a trademark of ParcPlace Systems, Inc. UNIX is a registered trademark of AT&T. Titan is a trademark of Ardent Computer Corporation.
Copyright © 1988 by ParcPlace Systems, Inc.
All Rights Reserved. 9/88

S M A L L T A L K - 8 0™



The Smalltalk-80™ environment is the most highly integrated object-oriented programming environment available, a portable and extensible system that gives software developers three benefits that derivative products cannot match:

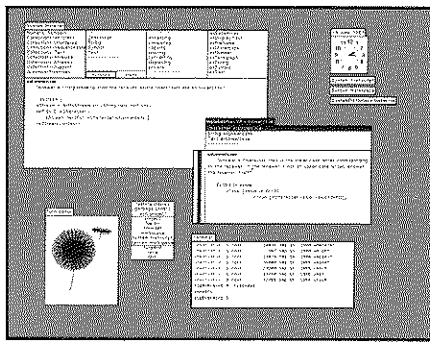
- ▶ It allows rapid incremental development of truly portable graphical interactive applications.
- ▶ It is the fastest development environment available, with increases in productivity of 10 to 15 *times* not uncommon.
- ▶ It lowers the cost of development, maintenance, training, and documentation.

PARCPLACE SYSTEMS

STRUCTURE

The Smalltalk-80 system has two major parts, the *virtual image* and the *virtual machine*. The virtual machine links the hardware and operating system "platform" to the virtual image. The virtual image contains the language and runtime system, and system libraries for managing the graphical user interface and program development tools. The runtime and development systems remain essentially unchanged across the following platforms, for which virtual machines have been written:

- Sun 2, 3, and 4;
- Hewlett-Packard 9000, Series 350 and Series 320;
- Apple Macintosh II, Plus, and SE;
- Apollo Domain 3000 and 5XX.



PORTABILITY

Because the Smalltalk-80 runtime environment changes so little from machine to machine, programs written on one of the workstations above can be ported to all of the others with a minimum of reprogramming effort. The environment allows developers to build applications with highly integrated, high-level graphical user interfaces using windowing systems, menus, and pointing/selection devices. These applications can be ported with minutes or—in the case of large-scale systems—a few hours of reprogramming.

The net effect is to allow programmers to develop essentially entire families of software products each time they complete a single development project, because the application they develop using Smalltalk-80 tools will be easily portable across a large and ever increasing number of workstations.

PRODUCTIVITY

Because the Smalltalk-80 environment is object-oriented, it yields two major increases in productivity. The first is development *by refinement*. You can rapidly build systems because you are refining existing objects instead of starting every development project from scratch. The second is the ability to place new code

immediately in context and examine its potential impacts on the existing system before proceeding.

LOWER COSTS

The third major benefit of the Smalltalk-80 system is its extreme simplicity and history of bug-free operation. These lower the costs of learning to use it, teaching others to use it, and maintaining applications built with it.

FEATURES

Most software designers are intensely sceptical when vendors allege that a development package is "fully integrated." They've heard it all before. In the descriptions of features below, however, "fully integrated" means exactly that. Once you enter the Smalltalk-80 system, you stay in it to access *all* of the system's major elements. You can de-bug it, fine-tune it, edit it, and browse it, all without leaving Smalltalk-80's flexible environment. And while some systems may have scaled-down versions of some of the following features, no system has full-function versions of them all. The Smalltalk-80 environment:

1. Has the original (and most powerful) high-level user interface using overlapping windows, menus, and a pointing/selection device. The development environment and the applications you build use the same interface.
2. Is remarkably integrated. Facilities for debugging, text editing, performance monitoring, change management, querying, and more are all immediately available within the system where and when you need them.
3. Is reusable and extensible. The Smalltalk-80 environment is purely object-oriented, which allows you to build programs incrementally and fast, using

efficient existing code for an astonishing range of programming tasks.

4. Is enriched with hundreds of efficient, expertly designed, built-in data models.
5. Has code for automatic allocation and reclamation of storage that can be used regardless of data type or procedure.
6. Allows applications built on one platform to run on others essentially without modification.
7. Has a wide range of built-in applications, including a paint system, multifont text editor, browsers, inspectors, and two major simulation programs.
8. Has had a great number of applications written for it by third parties, including expert systems, database management systems, spreadsheets, music tools, and more.
9. Is well integrated with the features and facilities of the host hardware and operating system. For example, it runs on the Sun under the SunView windowing system, gives access to Desk Accessories on the Macintosh, gives access to Unix shells, and supports host function keys.

Sun 2, Sun 3, Sun 4, and SunView windows are trademarks of Sun Microsystems, Inc. Apple is a trademark and Macintosh is a trademark licensed to Apple Computer, Inc. Domain is a registered trademark of Apollo, Inc. Unix is a trademark of AT&T Technologies, Inc.

© 1987 by ParcPlace Systems. All rights reserved.

REQUIREMENTS

The following table describes the hardware and operating system requirements for the Smalltalk-80 system on the various platforms.

WORKSTATION	OPERATING SYSTEM	MEMORY	
		RAM	DISK
Sun 2	Sun Unix 3.0 or greater	4 MB	20 MB
Sun 3			
Sun 4			
Apollo Domain 3000	Apollo Domain 9.5	4 MB	20 MB
Apollo Domain 5XX			
HP 9000, Series 350*	HP-UX 5.22	4 MB	20 MB
HP 9000, Series 320*			
Apple Macintosh II	Macintosh OS	2 MB	10 MB
Apple Macintosh Plus			
Apple Macintosh SE			

*The Hewlett-Packard machines require an HP-HIL ID module and a two- or three-button mouse.



ParcPlace Systems

Smalltalk-80 Programming Environment **Version 2.3**

Sun-3 Workstation

Smalltalk-80 - The Complete Object-Oriented Programming Environment

Unrivalled functionality. Total customizability. Powerful visuals. Completely portable. Smalltalk-80™, the complete integrated programming system, extends the notion of productivity through an accomplished set of development and information access tools, a highly acclaimed object-oriented language, tested and refined applications, and full portability. Smalltalk-80 allows professional application programmers to bring their products to market faster, to update and extend their products faster, and to maintain code on current and future platforms.

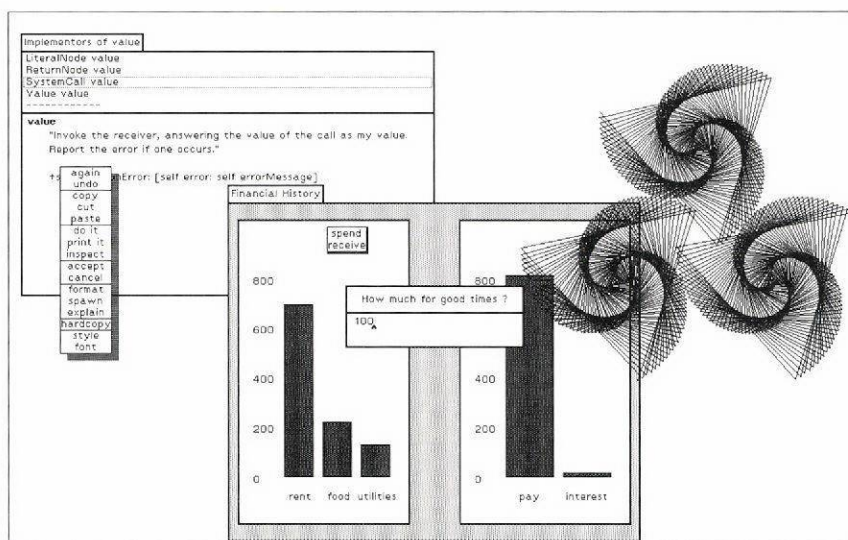
Smalltalk-80

Smalltalk-80 is a powerful, easy to use, customizable software development environment which contains all components needed for rapid development of sophisticated applications. The environment provides a seamless interface to the operating system and hardware of the host platform as well as a new level of portability across platforms.

Improved Productivity with Smalltalk-80

The Smalltalk-80 system provides well integrated tools for modifying existing applications, as well as adding complementary new functionality. It includes access to a library of reusable presentation and interaction components, and frameworks for specific kinds of applications.

- Program by refining from a library of reusable code modules which have been pre-tested and fully debugged.
- Use built-in applications and simulations for rapid development of prototype programs.
- Reusable frameworks allow the programmer to specialize a set of functions.
- Ability to move fluidly between writing code, debugging, and inspecting all levels of system operation.
- Powerful graphical user interface based on overlapping windows,



pop-up menus, multiple fonts, and mouse-oriented control.

- Ability to link and access subroutines written in other common languages, such as C.
- Quick compilation of new code allows a programmer to easily test new modules and to modify them incrementally until reaching the desired result.

Extensive Development Tools

- Expandable tool set includes source code browsers, text and graphics editors, performance monitors, dynamic cross referencing, symbolic debugger, inspectors, explanation and querying.

- Access to external operating system specific facilities.

Widely Portable

For the first time, professional application programmers have a wide and interchangeable choice of platforms on which to develop software products. ParcPlace products act as environment enhancers to UNIX™, Mac OS, and MS-DOS® operating systems and are currently available on the following processors: MC 680X0, Intel 80386, Sun SPARC™ and MIPS™.

- Most applications written on any of the supported platforms will run, with little or no modification, on any other supported platform, regardless of operating system or hardware.

Easy to Learn

- A wealth of on-line documentation and examples makes code easy to read and reuse.
- Accompanying manuals, as well as published books and articles, provide a full range of exemplary and reference material.
- Training available, both in-house and on-site, covering introductory and advanced level programming.

Continuous Support

- Software support is offered to single users and customer sites. Support includes a customer hotline, electronic bulletin board, quarterly newsletter, upgrades, Smalltalk-80 books, technical notes, and hands-on training.

Many Available Applications

- Currently available commercial applications include software development (CASE), map handling, CAD/CAM, spreadsheets, expert systems, dynamic simulation, integrated document preparation, calendars, music and sound processing, and natural language parsing and translation.

Smalltalk-80

The Smalltalk-80 fully integrated object-oriented programming system is an environment that is unmatched in power and ease of use for general system-level customization by professional application programmers.

Language & Tools

- Rich set of context-specific browsers, inspectors, and dynamic cross-referencing tools provide informed access to all system code and reusable code modules.
- Seamless access to text editors.
- Interactive symbolic debugger available at all times for correcting problems and for inspecting implementation of exemplary code.

- Performance monitoring tools assist in development of optimized, quality code.
- Automatic storage allocation and reclamation.
- User defined primitives can be used to call and link routines written in languages such as C.

Host Platform Functions

- RS-232 serial port access.
- Support for color.
- Sockets for the UNIX operating system.
- Access to UNIX operating system shells.
- Support for host platform function keys.
- Integration with the Sun window environment.

Additional Features

- Easy to use visual user interface.
- Extensive example set, including a full simulation event-driven kit, to get the programmer "up and running."
- PostScript™ printing facilities.
- Full image compatibility with other industry workstations.

Requirements

- Sun-3™ Workstation with system console and mouse.
- Sun OS version 3.0 or later.
- 4MB or more of RAM.
- 10 MB or more of free disk storage space.
- Tape device on network for loading.

Customer Support Services

ParcPlace Systems offers close contact with the Smalltalk-80 system technical staff on a continuing basis and assures the latest improvements and upgrades to all supported products. These services are included with the Site Customer Support:

- Three hours of Hotline Support.
- Free upgrades to new software releases.
- Smalltalk-80 Newsletters.
- Privileged access on ParcBench™, the ParcPlace Bulletin Board Service.
- Smalltalk-80 Books on advanced programming and user interface.

Single users may purchase individual upgrades for each software release.

In addition, ParcPlace offers an extensive Smalltalk-80 training program. Introductory and advanced programming, and a masters course on product packaging, provide hands-on training for programmers at different levels of experience. Smalltalk-80 customer workshops provide an opportunity for customers to explore issues of general interest.

Orders and Information

To receive further information on the Smalltalk-80 system, customer support services, and training, please contact ParcPlace Systems.

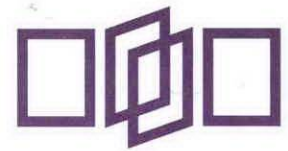


ParcPlace Systems

2400 Geng Road
Palo Alto, CA 94303
Toll Free: 1 800 822 ST80
(800 822 7880)
in California: 415 859 1000

Smalltalk-80 and ParcBench are trademarks of ParcPlace Systems, Inc. Apple and AppleShare are registered trademarks of Apple Computer, Inc. Macintosh is a trademark of Apple Computer, Inc. MIPS is a trademark of MIPS Computer Services, Inc. MS-DOS is a registered trademark of Microsoft Corporation. SPARC is a trademark of Sun Microsystems, Inc. TOPS is a registered trademark of Sun Microsystems, Inc. PostScript is a trademark of Adobe Systems, Inc.
Copyright © 1988 ParcPlace Systems, Inc.
All rights reserved. 6/88.

ParcNotices



VOLUME 2 NUMBER 1 WINTER 1991

Objectworks\ C++ Release 2 ships, Objectkit\ C++ is next

Volume customer shipment of Objectworks\ C++, Release 2, occurred on schedule, on November 21, 1990. Now, C++ programmers everywhere will learn what our beta users learned months ago...that the C++ language isn't enough! They can now exploit the benefits of object-oriented programming.

Release 2 provides a set of tools for creating, finding, understanding and reusing C++ code, enabling programmers to build robust applications more efficiently. Objectworks\ C++ supports traditional Unix tools for C pre-processing, C compiling and linking, in addition to debugging and source code control systems. Performance enhancements support large system development and incorporate the AT&T C++ Language System, Release 2.1.

Users of the beta release of Objectworks\ C++ are excited by these advances. From Data I/O, Paul Brownlow says, "I selected Objectworks\ C++ because the tools present code in a way that is consistent with object-oriented technology. That is particularly useful in reducing the time it takes to develop and debug applications."

continued on page 4

Objectworks\ Smalltalk enters Phase 2 of rollout

On November 19th, ParcPlace celebrated the on-time shipment of Objectworks\ Smalltalk, Release 4, for Sun, DEC and Macintosh platforms. Most Objectworks\ Smalltalk 2.5 users have updated to Release 4 via their Objectsupport subscription, and have already provided positive feedback to us.

Texas Instruments was one of our Release 4 beta sites. According to John Hebley, software project manager, the straightforward migration from Version 2.5 enabled them to port their applications quickly. TI is also incorporating Release 4 color and graphics features into their application.

The beta program for Phase Two, which will bring Objectworks\ Smalltalk, Release 4 to the HP 9000 Series 300 and 400, HP/ Apollo and IBM RS/6000 workstations, as well as the MS Windows 3.0 environment, has started. We're looking forward to on-time shipment in March 1991.

Objectkit\ Smalltalk Advanced Programming, also shipping in March, includes Benchmark facilities; Checker and Profiler tools; Class Reports and Parser-Generator tools; a Project Browser and a full Protocol Browser; a terminal emulator communications facility; and a binary object storage facility (BOSS).

Objectkit lists at US\$500 in the US and Canada, and will be available direct from ParcPlace and from our international distributors. For more information, call 800-759-PARC. ♦

Inside ParcNotices

Objectsupport™	
Coming Events	2
Introducing ...	2
Training Schedule	2

Objectworks®\ Smalltalk	
Facets For OW\ST	5
BOSS! NOW!	7
ParcPlace Partners	5
Product Release	1
Q & A	5
TechTalk	6

Objectworks®\ C++	
Libraries	11
OWC++ & X11	11
Product Release	1
Q&A	8
SQL & C++	9
TechTips	8
Typesafe Linking	10

ParcBench®	3
-------------------	---

New Hotline Number!!



800-253-3415

Now Toll Free from Canada Too!!

Introducing...Roy Benjamin

He may be the newest face on the Objectsupport team, but he's not new to OOP. Roy Benjamin joined ParcPlace Systems in August of 1990, after one and a half years as a project leader at Consilium. He was introduced to C++ during development of their new X Window product, and this whetted his interest in object-oriented programming. After some initial investigation into companies developing object-

oriented products, he decided that ParcPlace was where he would like to work.

Roy brings to the Objectsupport team a strong working knowledge of C++. Being accustomed to using the standard C++ tools, he's been like a kid in a candy shop for the past six months, learning all of the different tools and capabilities provided in the Objectworks\C++ product. Working with customers who need support has been a good learning experience for him, as well. "The reason I enjoy being on the Objectsupport team is because I am always learning," Roy says. "The problem solving aspect is what challenges me most, and the resources I have down the hall to help me are phenomenal! So far, no problem has been able to stump me!"

In the short time Roy has been with ParcPlace, he's seen the rollover of the Objectsupport program and overseen the beta program of Objectworks\C++. The benefits of the new Objectsupport program have been immediately useful to Roy. "Since every Objectsupport subscription now has a name linked, it is much easier to track the 'audit trail' of support for that particular person. I can see what they've done, what they are trying to do, and give advice as to how to accomplish their goal." And although the beta period for Objectworks\C++ went very well, he's happy that the product is now shipping. "Since I enjoy problem solving, the in-bound calls are the real challenges for me. The beta program went so smoothly, my outbound calls were just routine maintenance," he says.

Problem solving in the Objectworks environment isn't enough for Roy however. During his diverse career, he's been a fireman, a contract administrator, an accountant, and a freelance programmer, prior to his start at Consilium. If you hadn't guessed, yes, Roy likes the challenge of learning new things and the challenges of keeping up-to-date with computer technology. Some of the "problems" he solves in his spare time are: those everyday items related to raising his two children, finding the right parts for his vintage Corvairs, and negotiating white water rapids. Compared to these, technical questions and problems are a snap! ♦

COMING EVENTS

. ParcPlaces and ParcPeople

Check this list and make notes on your calendar. We would like to see you at any or all of these events featuring ParcPlace Systems, members of the ParcPlace staff, or ParcPlace Distributors.

Trade Shows

- UniForum '91 1/22-24/91
Dallas TX, USA
- Software Development '91 2/12-15/91
Santa Clara CA, USA
Speakers: Doug Pollack, Elizabeth Gibson
- Windows & OS/2 Conference 3/5-7/91
San Jose CA, USA
Speaker: Doug Pollack
- CeBit '91 3/13-20/91
Hanover, Germany
- Objectworld 6/3-6/91
San Francisco CA, USA

Speaking Engagements

- TOOLS '91 - Adele Goldberg 3/4-8/91
Paris, France
- SCOOP West 3/11-14/91
San Jose CA, USA
Doug Pollack, Elizabeth Gibson

TRAINING SCHEDULE.

- Object-Oriented Methodology Workshop 2/12-15/91
- Introduction to Object-Oriented Concepts and C++ 2/25/91 - 3/1/91
- Introduction to Object-Oriented Concepts and Smalltalk 3/4-8/91
4/8-12/91
6/3-7/91

Sitting on the ParcBench.....

ParcBench is the ParcPlace 24-hour, 7- days-a-week electronic bulletin board. If you want to "sit" on the ParcBench, you can reach it at 415-691-6716, observing the following parameters:

- 300/1200/2400 bps
- no parity
- 8 bits
- 1 stop bit

Following is a listing of new contents added to ParcBench since the last edition of ParcNotices. Filenames are case sensitive and are listed alphabetically, without regard to capitalization.

C++files download area:

start_oracle executes Oracle's embedded SQL preprocessor.
start_oracle.sed auxiliary file for start_oracle.
 From Reza Forushani, AT&T.

ST80files download area:

Gamma_random_sampling.st Version 2.5 fixes to Gamma probability distribution class
 by Rick Cameron, Mobile Data International
 ...!uunet!van-bc!mdivax1!cameron
R4-compatible2.5.st Backward compatibility file for converting Version 2.5 code to Release 4.
 Unsupported code.
samuelson-browserMods.st adds buttons for resizing BrowserViews in various ways.
 From Bruce Samuelson, University of Texas, Arlington
 bruce@utafl.lonestar.org
SparseArray.st for Version 2.5, from ParcPlace Newsletter Summer 1990.

ST80support download area:

Boss-fileout.ws workspace code for filing out BOSS from Version 2.5 so that it can be filed in to
 Release 4.
simple-cursorKeysR4.st a very basic implementation of cursor keys for ParagraphEditor. Extensions are
 left as an exercise for the user.

TechNotes download area:

DLLwriteup.txt Interfacing to Windows 3.0 DLLs from Smalltalk (Version 2.5 and Release 4)
Technote101.txt Objectworks for Smalltalk-80, Version 2.5 freezing on SparcStations, SunOS 4.1
Technote102.txt Objectworks for C++, version 1.01 and 1.02 freezing on SparcStations, SunOS 4.1
Technote103.txt DecStations crashing when running Objectworks for Smalltalk-80, Version 2.5
Technote104.txt OpenLook (olwm) close/resize hang
Technote105.txt Smalltalk userprim with SunOS 4.x
Technote106.txt BOSS - from Version 2.5 to Release 4

USENETfiles download area:

altformat.st ST80 version 2.2. An alternative source formatter.
 From Darren New, University of Delaware.
ArrowedSpline.st ST80 version 2.3. Subclass of Spline that adds arrowheads to curves, indicating
 the order in which you supply the defining points.
 From Ross Morley, Philips R&D Center, Sunnyvale, CA.
BDF2StrikeFont-BrouHaHa.st BrouHaHa ST80 version 2.3.1. May be a duplicate of the BDF to StrikeFont
 converter from the University of Manchester Smalltalk archives.
 From Eliot Miranda, University of London.

continued on page 4

.... and more on the ParcBench

USENETfiles download area:

CollectionMenu.st	Allow selection of an object from a Collection via a HierarchicalMenu. From Pieter van der Meulen, Philips R&D Center, Sunnyvale, CA.
DisplayIfWhile.st	Display if some condition is true while some block that takes a long time to finish is still executing. Examples included. From Pieter van der Meulen, Philips R&D Center, Sunnyvale, CA.
FLV-UnixEnvVar.st	ST80 Version 2.5. Allow Unix environment variables to be used as the root component of a pathname in a FileListView. From Dr. Kevin Waite, CS, Glasgow University, Scotland.
FuzzyBrowse.st	ST80 Version 2.5. Allows you to find a method by approximate name. From Edwin Luebben.
grope.st	ST80 version 2.2. Examples of filing out code by relationships other than method protocol, class, class category, changeset, etc. as provided by standard Smalltalk. From Darren New, University of Delaware.
hardcopy.st	ST80 version 2.2. Print screen snapshot with 10 second delay to bring up menus, etc., or snap immediately if you hold down the left shift key. From Darren New, University of Delaware.
iconSlurper.st	ST80 version 2.3. Read Macintosh ICON resources decompiled by MPW or a similar tool into Smalltalk-80. From Jordan Bortz.
newBDF2StrikeFont.st	A bug fix version of Chris Ryland's BDFFont.st. Posted by, and bug fix supplied by, Kaveh R. Ghazi, Rutgers University.
PBM.st	ST80 Version 2.5. Convert between Smalltalk Forms and Jef Poskanzer's Portable Bitmap Format ... which then allows you to convert to/from PICT, GIF, MacPaint, Sun icon, PostScript, X11, etc. Jef Poskanzer's package is freely available for Unix platforms from a variety of Internet and UUCP sites. From Dr. Kevin Waite, CS, Glasgow University, Scotland.
Pt-nearest.st	ST80 Version 2.5. Implementation of point on a line segment nearest to another point. The standard Point>>nearestIntegerPointOnLineFrom:to: finds the point on an infinitely long line (between 2 given points) nearest to a third point. From Dr. Kevin Waite, CS, Glasgow University, Scotland.
tristateButton.st	ST80 version 2.3. SwitchView with three visible states (black, gray, white) that responds to either a mouse click or a held-down mouse button. From Ross Morley, Philips R&D Center, Sunnyvale, CA. ♦

continued from page 1

Two new features won rave reviews from Chris Bloom, cc:Mail software engineer. C++ made development of a portable Mail Engine possible, but "Objectworks\C++ has made it easier by increasing productivity. Objectworks has the best graphical source browser and visual debugger I've ever used."

Class libraries are offered through an add-on development tool, Objectkit\C++, which provides two C++ class libraries from AT&T and, as a service to our customers, two unsupported C++ class libraries: Stanford InterViews and the National Institutes of Health Library. No importing of C++ source code is necessary. Objectkit\C++ libraries are shipped with Objectworks\C++ environment (.owc) files. When used in conjunction with Objectworks\C++, these class libraries can be browsed quickly and accurately, immediately after they are copied off the distribution tape. The Objectworks\C++ development environment is simply invoked with the environment file as an argument.

Objectkit\C++ lists for US\$500 in the US and Canada. It is scheduled to ship in the first quarter of 1991, and will be available from ParcPlace directly, and through international distributors. If you haven't seen the new release of Objectworks\C++, contact your sales representative at 1-800-759-PARC, or 415-691-6700. ♦

ParcPlace starts Partners Program for Smalltalk developers

To further support third-party Smalltalk application developers and systems integrators, and to increase awareness of the variety of applications implemented in the Smalltalk language, ParcPlace Systems has instituted a new program—**ParcPlace Partners**.

In this issue of *ParcNotices*, we introduce the program and, in subsequent issues, we will focus on some of our Partners and their services.

ParcPlace Partners provides a support infrastructure for value-added resellers, systems integrators and independent software vendors who increase market acceptance of Smalltalk through development and implementation of real-world business applications.

A variety of marketing and technical services will support the development, porting and sales of Partners' Smalltalk applications. These services include:

- Listing in Objects Illustrated Applications & Services Directory
- Co-marketing programs
- Development of case studies for distribution to trade publications as well as prospective customers
- Participation in Partners Forums.

Through the Objectsupport program, **ParcPlace Partners** will receive software updates, early releases of new versions of Objectworks\Smalltalk, unlimited support hotline access, a technical newsletter, and electronic bulletin board facilities.

Are you eligible for **ParcPlace Partners**? You are if you offer a commercially available product or service for *Objectworks\Smalltalk, Release 4*.

You may also be eligible if you are in the process of developing or porting an application to *Objectworks\Smalltalk*. Applications still under development are reviewed on a case by case basis to determine eligibility.

If you are interested in becoming a **ParcPlace Partner**, or would like to be considered for inclusion in this column, please contact Melissa McCutchin at 415-691-6711, for details. Contact Catherine Tucker at 415-691-6759, if you are currently developing an application to be considered for eligibility in the program. ❖

Smalltalk Questions and Answers

Q: Can I run Objectworks for Smalltalk, Version 2.5 on my HP 9000 under HP/UX 6.5?

A: No, you must upgrade to HP/UX 7.0, because HP/UX 6.5 has a non-standard implementation of **gethostbyname()** which causes Smalltalk's **st80ID validation** code to always fail.

Q: Is there any way to make the pop-up menus come up faster in Objectworks\Smalltalk Release 4 under OpenWindows 2.0?

A: If you have focus-follows-mouse mode set in your window manager, you're generating lots of window enter/exit events. If you switch to click-for-focus, menus will pop up about twice as fast.

Q: In Objectworks\Smalltalk, Release 4, the new windows seem to pop up wherever they want and set to some default size. I like the old behavior where I got a framing rectangle prompt and got to place and size the new window. Can I still do this in Release 4?

A: Yes. Just execute the expression:

ScheduledWindow promptForOpen: true.

continued on page 6

Servio's Facets now available from ParcPlace

Facets, a 4GL application development tool for the Objectworks\Smalltalk environment, was introduced in October 1990 by Servio. Facets is a comprehensive suite of object-oriented tools designed to assist programmers in the development of Objectworks\Smalltalk applications.

These tools include Schema Designer, Forms Designer, Report Writer, Menu Builder and Help Author. They enable developers to quickly generate screens, reports, menus, and other interface components in order to create a fully functional program. In addition, Facets includes complete source code and an on-line hypertext-based, context-sensitive help system to assist the developer every step of the way.

Facets is immediately available for Release 2.5 of Objectworks\Smalltalk. It lists for US\$1,000 in the US and Canada, and is available direct through ParcPlace and from international distributors. For information on this exciting tool, and availability for Objectworks\Smalltalk, Release 4, contact ParcPlace at 800-759-PARC. ❖

TechTalk for Smalltalk

This TechTalk column is devoted to presenting problems, describing probable causes, posing questions, and providing solutions for users of Objectworks\Smalltalk. Problems and solutions are platform specific.

Version 2.5 DECstations **Problem:** On certain versions of UWS/Ultrix, Objectworks for Smalltalk-80 Version 2.5 can cause the DECstation to crash.

Cause: The cause of the problem is a bug in the Ultrix paging system. The bug was originally reported to DEC for UWS 2.1, based on Ultrix 3.0. DEC accepted the bug report, assigning it the identifier: SPR # SMV-02547.

Solution: The bug was fixed in UWS 2.2D, based on Ultrix 3.1. It requires that UWS 2.2D be installed from scratch. An upgrade install will NOT install the bug fix. Ultrix 4.1 also has the bug fix installed and works correctly with both Version 2.5 and Release 4 of Smalltalk. *NOTE:* Ultrix 4.0 does NOT have the bug fix installed.

Release 4.0 Sun SPARC-stations **Problem:** Running Objectworks\Smalltalk, Release 4 under `olwm` and clicking on the close or resize widget with the middle mouse button can hang Smalltalk.

Cause: The behavior is caused by a known bug in Sun's `olwm` that causes X clients using passive input mode to hang if you click on a window manager widget with the middle mouse button when you have click-for-input-focus set.

Objectworks\Smalltalk, Release 4 is an X client that uses passive input mode. You should be able to get the same behavior from an xterm window. Shelltools and cmdtools are not affected because they use active input mode. Sun has assigned this Bug ID 1039020 in their X11/NeWS Version 2 Bug List. Watch future releases from Sun for their bug fix.

Solution: Either avoid using the middle mouse button with `olwm` window widgets by using the left mouse button instead, or switch from click-for-input-focus to focus-follows-mouse. In OpenWindows 2.0, use the following procedure to switch from click-to-type to follow-the-mouse:

- select "Properties" from the `olwm` root menu

- in the Workspace Properties dialog pop-up, select the category "Miscellaneous"
- click on Move Pointer next to the item labeled "Set Input Area:" and then click on the Apply button.

Version 2.5 SunOS 4.1 **Problem:** Some user-defined primitives linked with Version 2.5 virtual machines may crash if compiled and linked under SunOS 4.x. We have one report that the problem has occurred with SunOS 4.0.3c.

Cause: Version 2.5, developed under SunOS 3.5, uses `sbrk()` to allocate space for Smalltalk object memory. By SunOS 4.1, Sun acknowledged having changed `malloc()` so that it is now incompatible with `sbrk()`. Since many C library calls rely on `malloc()`, it is possible that user-defined primitives may invoke `malloc()` and thus trigger the incompatibility with `sbrk()` at runtime.

Solution: Use SunOS 3.5 or upgrade to Objectworks\Smalltalk, Release 4. Release 4 was developed on SunOS 4.1 and uses `malloc()` instead of `sbrk()`, so it avoids this problem. ❖

More Questions and Answers

continued from page 5

Thereafter, when you create new Smalltalk Windows, Smalltalk will expect you to place and size them, the same way as in Version 2.5 and before.

Q: *Macintosh applications let you select "Open and Save As..." from the File menu. You get a "Standard File Dialog" pop-up in which you can select a name to open a file, or type in a name to save a file. How can I get these file names into an instance of a Smalltalk Filename object?*

A: The following fileIn works in conjunction with the MacExtra.st fileIn from the Utils folder in Version 2.5 and Release 4. **nameOfType:** returns a full pathname (that starts with the disk-volume name) from an Open dialog. **saveAsNameWith-Prompt:** lets you open a "Save As..." dialog and returns the full pathname formed from the file name you type into the dialog box.

MacOSFilename class methodsFor:
'pathnames'

nameOfType: anArray

"return an absolute pathname from a SFDialg 'open'."

continued on page 12

But I want my BOSS!! Now!!

Your Release 4 application requires binary object streaming service (BOSS), and you can't wait until Objectkit\Smalltalk Advanced Programming is available in March 1991? The new BOSS included with the Objectkit is a significantly enhanced version featuring:

- Improved performance for fast object loading
- Better detection of, and recovery from, attempts to load information that is stored in an obsolete or incompatible format
- Improved performance in runtime systems, including the ability to load classes and methods
- Ability to edit BOSS files in a content-independent way
- Random access within BOSS data.

What to do in the meantime? You can use the old version by moving BOSS from Version 2.5 into Release 4. But, there are one and one-half tricks to be aware of when doing so:

1. BOSS isn't simply implemented in the classes in **category System-Binary Storage**. It also requires methods going by four different names scattered about the system. So to make a proper file-out, you've got to include these methods.

1/2. In light of the foregoing, you must not forget that there are classes defined in Version 2.5 which do not exist in Release 4. So it is not advisable to file-out any methods defined in such classes.

Of course, there is the additional problem that objects BOSSed-out of Version 2.5 won't necessarily BOSS-in correctly to Release 4, because their class may no longer exist or their representation may have changed.

This will definitely be a problem for classes as interesting as **Text**, **Form** and **CompiledMethod**. However there is nothing that can be done for the time being.

A **workspace** to execute in a running Version 2.5 system is provided below. The resulting file, when filed-in to Release 4, works just as well as BOSS does in Version 2.5.

File-out BOSS from Version 2.5

```
I outStream banished I
"Build list of relevant classes/meta-classes not
present in R4."
banished := IdentitySet new.
"We're dumping Text even though it exists in Release
4 because of rep changes."
#(#Form #TextCompositor #Text #StrikeFont
#TextStyle)
do: [:className I
  I class I
  class := Smalltalk at: className.
  banished add: class.
  banished add: class class].
"Open output stream."
outStream := 'BOSS.st' asFilename writeStream.
[outStream timeStamp.
Transcript
  cr;
  show: 'First file-out scattered boss methods...'.
"For all classes/meta-classes..."
Smalltalk allBehaviorsDo: [:beh I
  "(Except banished ones...)"
  (banished includes: beh) ifFalse: [
    "File-out any method named the following."
    #(#storeBinaryOn: #readBinaryFrom:tag:
    #readBinaryContentsFrom:
    #storeBinaryContentsOn:)
    do: [:sel I (beh includesSelector: sel)
      ifTrue: [beh fileOutMessage: sel
        on: outStream
        moveSource: false
        toFile: nil]]]].
outStream cr; cr.
Transcript
  cr;
  show: 'OK, now do boss classes proper...'.
SystemOrganization
  fileOutCategory: 'System-BinaryStorage'
  on: outStream]
"Be sure to close the file stream, no matter what."
valueNowOrOnUnwindDo: [outStream close].
```

NOTE: Look at Tech Note 106 on ParcBench for an electronic version of this text. ♦

Want to tell your secrets? See your name in lights!

As a user of our Objectworks products, you undoubtedly have learned something over time that will help others who have not yet delved into the areas you have. Why not help other users by dusting off a useful bit of code, or writing up a short piece on something you've learned to do, and submit your contribution to Technical Support (email to support@parcplace.com)? Be sure to include a statement giving ParcPlace permission to publish your contribution, and whether or not you want your name or organization mentioned. We really appreciate your contribution...and so will your fellow Objectworks users! ♦

Objectworks\C++ TechTips

This TechTips column is devoted to topics primarily of interest to Objectsupport customers who use Objectworks\C++ and the C++ programming language.

C++ .h file include strategies

C++ programs often have extensive header file include levels, because a class definition always includes the parent class definitions, all the way to the root class, in addition to processing parent class file include statements. As classes are reused, so are the classes they in turn use. It is not at all unusual for the preprocessor to visit 100 or more header files for each .c source file compiled. There are some strategies to minimize this problem.

It is already a general practice to `#ifndef` data definitions within header files to avoid multiply defining classes and data structures. A common idiom is:

```
#ifndef _FOO_
#define _FOO_
...declare foo etc...
#endif
```

The first strategy (defensive include) is to be defensive about including `foo.h` to begin with:

```
#ifndef _FOO_
#include "foo.h"
#endif
```

This will prevent `foo.h` from being revisited if the preprocessor has already "seen" the file.

Another C++ strategy, which is a departure from common C language practice, says, "Never include a file unless compilation cannot succeed without it." For C++, this means never include files in the class definition (.h or .hxx) file that are not required for a complete definition of the class. Files that are needed only for translation of the source file (.c or .cxx) should be included there only, and included defensively as well.

Finally, it is possible to minimize the need for including class definition files where the class is not a parent of the class being defined, by defining the class to have a pointer to an object (as an instance variable), rather than an actual instance of the class. For example:

```
#ifndef _FOO_
#include "foo.h"
#endif
```

```
class bar {
private:
foo afoo;
... };
```

can be replaced with:

```
class foo;
class bar {
private:
foo *afooPtr;
... };
```

thus avoiding inclusion of `foo.h` for class `bar` and all its child classes. This is especially important if the class `foo` is some distant child of its root class. The file `foo.h` can instead be defensively included in `bar.c`, and the pointer `afooPtr` 'new'd in the constructors for `bar`. ♦

Objectworks\C++ Questions and Answers

Q: We will be running Objectworks\C++ on the new diskless Sun SLC workstations (max 16 mb primary memory), networked to a SPARCserver. Do we need a local disk for swapping?

A: In our experience, diskless workstations have not been a problem. Whether it is faster using the local disk depends on your Ethernet traffic level, the interface type, the speed of the SPARC server, and the size of the working set. Increasing memory, up to the point when the program and data can all reside in memory, reduces swapping so programs run faster whether the swap area is local or across the network.

Q: I am running C++ cfront 2.1, (delivered with Objectworks\C++, Release 2) on a SPARCstation 1. The preprocessor complains about "unreasonable include nesting" when my file hierarchy is deeper than 11 levels. I do not want to flatten my file hierarchy, and I do not want my .c files to know about the inheritance hierarchy. What can I do?

A: Unfortunately, the nesting level capacity is a function of Sun's C preprocessor. Try replacing the Sun C preprocessor with the GNU C preprocessor. You can substitute `/bin/gcc -E -v` for `/lib/cpp` in the Tool Chest Standard Tool Locations. We have verified that GNU 1.36 handles at least 12 levels of nesting.

continued on page 11

Integrating embedded SQL preprocessors

Reza Forushani's team at AT&T contributed the following example of integrating an SQL preprocessor for Oracle Pro*C version 1.3 linkage.

Because Objectworks \ C++ does not accept files with a .cpc extension, it is necessary to rename all embedded SQL files using the .CXX extension, and rename non-SQL files using the .C extension. You can do this by modifying the shell script listed in the Objectworks \ C++ Installation Guide.

The following shell script, invoked whenever the C preprocessor is executed, looks for .CXX extension files and creates .CPC temporary files after running the Pro*C SQL preprocessor. The intermediate files are normally deleted unless an error occurs during the Pro*C step. Files without the .CXX extension proceed directly to /lib/cpp after the standard /lib/cpp is called. You will need to specify this shell script in the Objectworks \ C++ Tool Chest Standard Tool Locations.

```
#!/bin/sh
index=$#
arg=""
while [ $index -gt 1 ]
do
    arg="$arg $1"    shift 1    index='expr $index - 1'
done
fn='basename $1'
if [ 'echo $fn|cut -f2 -d"."' = "CXX" ]
then
    PCCFLAGS="include=${ORACLE_HOME}/c/lib ireclen=256 \
oreclen=256 rebind=no select_error=no sqlcheck=no"
    name='echo $fn|cut -f1 -d"."'
    fname=/tmp/$name.CPC
    pcc $PCCFLAGS HOST=c iname=$fn name=$fname >&2
    ret=$?
    if [ $ret -ne 0 ]
    then
        exit $ret
    fi
    rm $name.lis
    # clean up Pro*C output for use with C++
    sedcmd $fname /lib/cpp $arg $fname    rm $fname
else
    /lib/cpp $arg $1
fi
```

The following sedcmd script removes unnecessary declarations in the Pro*C output and can be used when additional preprocessing is needed prior to the C preprocessor.

```
sed '/^extern void sql[a-z,0-9]*(); $/d' $1 > tmp.c mv tmp.c $1
```

Finally, it is important to modify header files for C++ usage. For example, to modify sqlcpr.h:

```
#ifndef __cplusplus
    extern void sqlab2(unsigned long*, unsigned char**, unsigned long*,
unsigned short*, unsigned short**);
    [ other C prototypes ]
#else
    extern "C" { void sqlab2(unsigned long*, unsigned char**, unsigned long*,
unsigned short*, unsigned short**); }
    [ other C++ prototypes ]
#endif
```

NOTE: These shell scripts are available on the ParcBench C++ files download area. ❖

/bin/ld (or the mysteries of C++ typesafe linking)

The basics of linking C++ programs are similar to linking C programs—one or more .o files and libraries are presented to /bin/ld for assembly of an executable binary image. However, linking C++ programs occasionally introduces complexity not often found when linking C programs. The C++ object paradigm permits design and use of code systems that are much more complex than most traditional code systems.

The reuse of object libraries such as *AT&T*, *National Institute of Health* and *Stanford InterViews*, in addition to application specific code and libraries, can present a daunting task to the linker. When combined with the inter-relationships of object constructors and destructors, and C++ typesafe linkage, and because libraries of C++ objects often refer to objects from other libraries, the task may seem almost unmanageable.

Typesafe Linkage in C++ Typesafe linkage, new to C++, assures that the arguments to a function correctly match the function prototype. This in turn assures that the correct number and type of arguments are provided to functions at run time. Since function symbol table entry points are not mapped to function addresses until link time, information about the correct number and type of the function arguments is usually lost during compilation.

C++ solves this problem by contriving new names for functions based on the functions' argument types. For example, a function declared as `void f ( )`; is given the contrived name `f_Fv` ;.

A function implemented as `void f ( int X ) { ... }` is given the contrived name `f_Fi`. The contrived name is always the same for both the function prototype seen by function callers, and the implementation. This is also how C++ supports function argument overloading.

Contrived function names are often said to be “mangled” by the C++ translator. This mangling creates problems for the linker when one or more of the header files included during preprocessing is not designed with C++ function name mangling in mind. Here, the linker does not resolve link references to mangled function names.

In C++ , the translator can be told not to contrive a new typesafe name for a function. For example, `extern "C" f ( )`; provides a function prototype for function `f` with no arguments. With “C” linkage, such a declaration will typically be used for a function from a C library. The name is not changed to `f_Fv` by the translator.

The order of the libraries as they are presented to /bin/ld is critical to resolving symbol table references. /bin/ld (and most other linkers) make only one pass at the .o and library .a archive files presented on the command line. It is possible to create a link paradox of symbol references ld cannot resolve.

Normally, code and data segments of each .o file are ordered and included into the executable a.out in their entirety; this is not true for libraries. The linker keeps a table of references to data and functions by name, and seeks to find each name in the next .o or library file it processes. The table entry is deleted when the symbol is resolved. User-written code (.o and .a files) should be put before system and object libraries on the link line to avoid one of the most common missing-link-reference errors — placing the iostream archive too early in the library list. Libraries of C++ objects often contain references to code or data from another C++ library, usually not the case with most system libraries, which either do not contain external references, or have a prescribed order for ld.

C++ Link Paradox A link paradox, created with a circular reference across two libraries, can easily occur with C++ libraries, because object systems and the libraries built from them frequently reference each other.

For instance:

```
part1.a  has code for symbols XX and ZZ
          references to symbols SS & UU;
part2.a  has code for symbols SS and UU
          references to symbols XX & ZZ.
```

Limiting the number of separate libraries created and maintained within the C++ programming project will prevent the circular reference. It is also possible to include an archive more than once on the link command line.

Note that the .o files in these archives would have linked correctly because the entire contents of each .o file is included in the executable, whether used or referred to or not. See the /usr/man/man5/a.out.5 man pages for additional information about linking and the structure of an a.out. ♦

continued from page 8

Q: Will the Objectworks\C++ Process Inspector allow me to debug both C and C++ code?

A: Yes, this debugger works on both C and C++ source code, provided the `-g` option is used in the Tool Chest Standard Tool Locations User's debugging C compiler command. Some C source code may not be C++ compatible, but you can link in C dbx-compatible objects and debug in C source, to prevent incompatibilities.

Q: Are all the imported source files compiled into one `a.out` which runs on Sun? Is the resulting `a.out` used by the Objectworks\C++ Process Inspector?

A: Yes. Separate `.o` files are kept, corresponding to each `.c` file. These `.o` files are linked to one executable that runs on the Sun and is used with the Process Inspector. The separate `.o` files can be archived as one `.a` file for library construction, if desired.

Q: What can I do when my `cpp` goes into an infinite loop, such as when I use `generic.h`?

A: Using the Sun `/lib/cpp` can cause the `cpp` to go into an infinite loop, whether you are using the standard `/lib/cpp` or the C++ version distributed with the Sun C++ language system.

The error does not occur when the `-B` switch is used in the C preprocessor command. In the Tool Chest Standard Tool Locations, add `-B` at the end of the C preprocessor command.

NOTE: Using this switch may cause the wrong line to be highlighted in the Source Browser or Process Inspector.

Q: I imported a C++ program that generated a great many warning messages, and displayed no error messages when it first compiled. Yet I could not start the Process Inspector. What's happening?

A: The most likely explanation is that not all of the warnings and error messages will be accessible for correction in the Errors and Warnings Browser, or on the Error List in the Tool Chest. Error messages may be "clipped" from the bottom of the list, below the warnings, leaving you without a clue as to why the Process Inspector will not start.

Fix the warnings that are visible, then recompile and check for errors again. Or, you can try to discover the errors by compiling with `CC` outside of Objectworks\C++. ♦

Shell script builds and archives Objectworks\C++ libraries

To easily build and maintain Objectworks\C++ archives (libraries) of C++ objects, first set the Target executable: in the C++ Program Setup of the Tool Chest C++ Programming Environment to the name of the archive you want to maintain.

Next, depending on the Mode selected in the Tool Chest Processing Options dialog box, set either the Debugging link editor command, or the Optimizing link editor command in the Tool Chest Standard Tool Locations to the name of a script that will archive the `.o`'s you expect to build.

The setting for the link editor command should look something like this:

```
/bin/sh path-to-script/name-of-script
```

Be sure to provide a shell to execute the script. You can include anything inside the script that would ordinarily be needed to archive the `.o` files with `ar` and `ranlib`, including building a dynamic library with `ld`. You can browse classes as well as modify and rebuild the target library with Objectworks\C++.

NOTE: Objectworks\C++ may confuse any writing to `stdout` from the script with warnings or errors from `ld`, which it expects to run. ♦

Using Objectworks\C++ X11 without OpenWindows libraries

The X11 version of Objectworks\C++ is linked to the Sun `libX11` dynamic library. If this library is not present on your Sun workstation, you may get a message like `ld.so:libX11.so.1: not found` after trying to run `$OWCDIR/bin/owcX11`.

You can create a symbolic link from the X11 library you have to the one you don't have. For example: `ln -s /usr/lib/libX11.so.4.2 /usr/lib/libX11.so.1`.

If you don't have write permission to `/usr/lib`, create the link in any writable directory. Add that directory to the `LD_LIBRARY_PATH` environment variable, then re-try the command `# $OWCDIR/bin/owcX11 $OWCDIR/environment/initial.owc`.

It may be necessary to run `ldconfig`, to update the `ld.so` performance cache. If you made the symbolic link to a non-standard directory, remember to specify it to `ldconfig`. ♦

continued from page 5

```
"MacOSFilename nameOfType: nil."
"MacOSFilename nameOfType: #('TEXT' 'HPS3')."
|m l
m := MacOSFilename getFileTypes: anArray.
^m notNil
"nil = clicked on CANCEL"
  ifTrue: [m stringFromVRef , m tail]
  ifFalse: [m]

saveAsNameWithPrompt: aString
"return an absolute pathname from a SFDialo
'save'."
"MacOSFilename saveAsNameWithPrompt:
'Save As...'"
| dummy m l
dummy := MacOSFilename new named:
  " volumeReference: 0.
m := dummy putFileWithPrompt: aString.
"#putFileWithPrompt: returns a 6-element array
representing a Mac Pascal record of type SFReply
(see Inside Mac, Standard File Package Routines,
I-519). 1st is a boolean field which is true if you
clicked SAVE, false if you clicked CANCEL. The
4th field is the vRefNum, and the 6th the filename
typed into the SFDialo."
^(m at: 1)
  ifTrue:
    [m := MacOSFilename new named: (m at: 6)
      volumeReference: (m at: 4).
      "... make a new MacOSFilename just so we
      can turn its vRefNum into a pathname ..."
      m stringFromVRef , m tail]
  ifFalse: [nil]!
```

Q: When writing user primitives in Release 4, I'd like to be able to refer to the instance variables of an object by name rather than by relative position when using **UPinstVarAt()** and **UPinstVarAtPut()** functions. How can I do that?

A: One way is to generate a "C" enum type that lists all of the instance variables of a class. The following method takes a string argument as the name of a disk file into which to write the enum list. It's implemented in class **Behavior** so that you can send the message to any class.

Behavior methodsFor: 'accessing instances and variables'

allIvarsAsCEnumsToFile: aFilenameString

"Write out all the inst vars of the receiver as a set of C enums to aFilenameString."

"PositionalIOBuffer allIvarsAsCEnumsToFile: /tmp/posiobuf.h."

```
| file prefix l
prefix := self printString , '_'.
file := aFilenameString asFilename writeStream.
file nextPutAll: 'typedef enum {';
  cr; tab; nextPutAll: prefix , '0_FILLER'.
  self allInstVarNames
  do: [:ivar l
    file nextPut: $,; cr; tab; nextPutAll: prefix , ivar].
  file cr; nextPutAll: '}' , prefix , 'ivar;'; cr.
  file close! !
```

For example, if you select the executable comment in the method and **do it**, you get a file named **/tmp/posiobuf.h**, whose contents are:

```
typedef enum {
  PositionalIOBuffer_0_FILLER,
  PositionalIOBuffer_input,
  PositionalIOBuffer_output,
  PositionalIOBuffer_buffer,
  PositionalIOBuffer_bufferAmount,
  PositionalIOBuffer_currentBufferIndex,
  PositionalIOBuffer_bufferDirty
}PositionalIOBuffer_ivar;
```

ParcNotices

Copyright ParcPlace Systems, Inc. 1989, 1990, 1991. All rights reserved.
ParcPlace Systems, 1550 Plymouth Street, Mountain View, CA 94043. (415) 691-6700.

Publisher: Adele Goldberg
Managing Editor: Paul Greenblatt
Associate Editor: Kathryn Oppenheimer

Contributing Staff: Brian Alexander, David Arctur, Roy Benjamin, Greg Ching, Helen Cygnarowicz, Peter Deutsch, Elizabeth Gibson, Frank Jackson, Mike Khaw, Glenn Krasner, Abe Saad, Allan Schiffman, Alan Sloane and Catherine Jaeger Tucker.

Objectsupport, Objectkit, ParcPlace Partners, and ParcNotices are trademarks of ParcPlace Systems, Inc. Objectworks, ParcPlace, ParcBench and Smalltalk-80 are registered trademarks of ParcPlace Systems, Inc. Other product names may be the trademarks of their respective companies.

We make every effort to verify the accuracy of the information in this publication. In the event that we err, neither ParcPlace nor the editors/authors shall accept responsibility for any loss, liability, or damage caused directly or indirectly by the information contained in this publication.

— ParcNotices is printed on recycled paper. —

ParcPlace

N E W S L E T T E R



New Products From ParcPlace

Smalltalk-80 on 80386 MS-DOS PCs

You can now get the complete ParcPlace Systems Smalltalk-80 development environment on 80386-based MS-DOS personal computers, such as the Compaq Deskpro 386, the IBM PS/2 Model 70 and 80, and compatibles.

Version 2.4 of the Smalltalk-80 system, which has been shipping since November, provides all the power and flexibility of the Smalltalk-80 system *and* fully exploits the 80386. The MS-DOS implementation brings the productivity gains of the Smalltalk-80 system to a new class of machines, allowing developers to get products, upgrades, and extensions to market more quickly.

Version 2.4 is now available on non-680x0-based platforms, including the Sun-4 (SPARC-based), and Sun386i, Deskpro 386, and PS/2 (80386-based). Version 2.3 is available on 680x0-based platforms: Apple Macintosh, Sun-3, H-P 9000 and Apollo workstations.

President's Message

In 1988, ParcPlace became an independent corporation with the charter to develop and market premier object-oriented software technology. The Smalltalk-80 system is now available on many operating systems (MS-DOS, UNIX, Macintosh) and host architectures (680x0, 80386, SPARC, MIPS). In 1988, we also shipped a powerful and novel interface to UNIX: the Navigator (available on the Ardent Titan personal supercomputer).

Next year, you can count on ParcPlace to complete development on new products for programmers. We're especially excited about Cynergy! We will continue to support and upgrade the Smalltalk-80 language and programming environment.

1988 has been a very good year for the object-oriented community and for ParcPlace—and 1989 promises to be even better! We wish you happy holidays and all the best in 1989.

Adele Goldberg, President and CEO

Cynergy: More than the Sum of its Parts

ParcPlace recently announced development of Cynergy, an object-oriented, interactive programming environment for rapid development of C++ software. Cynergy's total power will be even greater than the sum of its parts!

The Cynergy programming environment will provide some of the most popular features that you have come to count on in the Smalltalk-80 environment: browsers, interactive object code debuggers, inspectors, and more. This is integrated with Glockenspiel's Designer C++, the most widely used C++ in the world.

Ted Goldstein of ParcPlace says, "Cynergy will be an outstanding product for professional programmers: *the first of its kind for C++*. It will integrate editing, searching, browsing, compilation and debugging—all in one tool."

Cynergy will be marketed and supported by ParcPlace Systems; initial delivery on Sun workstations is scheduled for Spring of 1989.

For a taste of Cynergy, see this issue's Picture Book (pages 5-6).

Winter 1988
volume 1 number 2

ON THE INSIDE

- OOPSLA '88
- Benchmarks
- Picture Book
- Recent Publications
- Technical Tips
- Users' Groups
- Training Program
- and more...

ParcPlace Signs Agreement with Fuji Xerox

In August, ParcPlace CEO Adele Goldberg traveled to Tokyo to sign a technology assistance and distribution agreement with Fuji Xerox. This establishes a formal exchange program between Fuji Xerox and ParcPlace, making Fuji Xerox ParcPlace's distributor in Japan and other Pacific Rim countries. It also confirms ParcPlace's exclusive worldwide ownership of the Smalltalk-80 technology.

Since 1986, Fuji Xerox has been active in the licensing, marketing, sales and support of the Smalltalk-80 system and applications in Japan and the Far East.

Request for Participation

Knowledge Systems Corporation of Cary, North Carolina is engaged in a study of Object-Oriented Design (OOD) and the various methodologies supporting it. KSC is interested in the opinions and experiences of anyone in the OOP industry who is involved in OOD, especially if you have attempted either to apply traditional methodologies, such as SADT, DSSD, and IDEF, or to develop a specific OOD methodology.

To participate in the study, contact either Sam Adams or Dr. William Mettrey at Knowledge Systems Corporation, 2000 Regency Parkway, Suite 270, Cary, NC 27511. (919) 481 4000.

ParcRanger Program Gains Momentum

A number of universities have already enrolled in the ParcRanger Program—a special new site licensing agreement with ParcPlace Systems that gives educational institutions easy access to the Smalltalk-80 system at very low costs. Among those signed up as ParcRangers are Georgia Institute of Technology, Princeton University, Purdue University, Rutgers University, University of California at San Diego, University of Illinois, and Vanderbilt University. The ParcRanger Program will be expanded internationally in 1989.

For more information, contact ParcRanger Coordinator, ParcPlace Systems, 2400 Geng Road, Palo Alto, CA 94303. (415) 859 1000 or (800) 822 7880.

Another Smalltalk Platform

Georg Heeg reports from Germany that he and his colleagues have the Smalltalk-80 system running on the PCS Cadmus RCU. The Cadmus RCU is a workstation based on the MIPS R2000 RISC CPU. Georg Heeg, Inc. distributes the ParcPlace Systems product line in West Germany. If you'd like more information, contact Georg Heeg, Inc. at Baroper Str. 337, 4600 Dortmund 50, Federal Republic of Germany, or send E-mail (via uucp or bitnet) to georg@unido.

The ParcPlace Newsletter is published quarterly by ParcPlace Systems, Inc. Copyright © ParcPlace Systems, Inc. 1988. All rights reserved.

Publisher: Adele Goldberg
Managing Editor: Eileen Keremitsis
Associate Editor: Susan King
Contributing Staff: Nanette Harter, Christina Herdell, Glenn Krasner, Steve Levine, Stephen Pope, Janet Ramos, Jodi Ranuio, Dean Ritz, Kenny Rubin, Allan Schiffman, Bill Shauck, and Helene Tannor.

Send submissions to: Eileen Keremitsis, Newsletter Editor, ParcPlace Systems, 2400 Geng Road, Palo Alto, CA, 94303-USA. The E-mail addresses are: newsletter@ParcPlace.COM or {ucbvax, decvax, seismo, ihnp4, pyramid}!sun!parcplace!newsletter. For subscription information, contact: Newsletter Subscriptions, at ParcPlace.

Cynergy, ParcBench, ParcRanger, and Smalltalk-80 are trademarks of ParcPlace Systems, Inc. Xerox and Dorado are trademarks of the Xerox Corporation. Fuji Xerox 1161 and Fuji Xerox 1121 are trademarks of Fuji Xerox. UNIX is a registered trademark of AT&T. Sun-3, Sun-4, Sun386i, SPARC, and Sun Workstation are trademarks of Sun Microsystems, Inc. Apple is a registered trademark of Apple Computer, Inc. Macintosh is a trademark of Apple Computer, Inc. Apollo is a registered trademark of Apollo Computers. Apollo Series 3000 and Series 4000 are trademarks of Apollo Computers. Tektronix is a trademark of Tektronix, Inc. Cadmus is a trademark of Periphere Computer Systeme GmbH. Atari and Atari MegaST4 are trademarks of Atari Research. Butterfly is a trademark of Bolt, Beranek & Newman, Inc. IBM PS/2 is a registered trademark of International Business Machines Corporation. Compaq and Compaq Deskpro 386 are trademarks of Compaq Computer Corporation. MS-DOS is a registered trademark of Microsoft Corporation.

OOPSLA '88—Another Big Success

Only in its third year, OOPSLA—Object-Oriented Programming Systems, Languages and Applications—is already the third largest of ACM's technical conferences. Attendance at OOPSLA '88 was up by 50% from 1987, to nearly 1,800. The conference was held at the Sheraton Harbor Island in San Diego, California, September 25-30.

There were over 180 technical presentations this year and the tutorial program was extended to three full days. With both the level of interest on the rise and a growing maturity of the object-oriented field, developers, researchers and users met to share information on a wide spectrum of topics, including applications, design, databases, user interfaces, tools and environments, software engineering, languages, implementation, and theory.

As before, the ParcPlace booth was hopping. At the show, we announced the Smalltalk-80 system for 80386 MS-DOS platforms, and our Navigator and Cynergy demonstrations drew enthusiastic crowds.

Next year's OOPSLA conference will be in New Orleans, Louisiana, October 2-6. Kent Beck, of Apple Computer, will be Program Chair for OOPSLA '89; George Bosworth, of Digitalk, will be Conference Chair. The following year (1990), OOPSLA is scheduled to meet with ECOOP, the European Conference on Object-Oriented Programming, for a joint conference in Ottawa, Canada.

For the ParcPlace staff, it is always enjoyable to meet with Smalltalk-80 users and discuss how you are using our products now and what you'd like to see in the future—and OOPSLA provides a good forum for this kind of exchange. We look forward to seeing you at OOPSLA '89!



At the ParcPlace exhibition booth, we showed the Smalltalk-80 system on a variety of workstations and computers, including one 80386 laptop computer—the Dynabook dream is getting closer to reality!

New Benchmark Results

	Sun386i 250	Sun 4 110	Sun4 280	IBM PS/2 Model 70 (20MHz)	Compaq Deskpro 386 (20 MHz)
BitBLT	18	30	31	9	10
TextScanning	104	351	240	71	312
ClassOrganizer	318	378	528	202	275
PrintDefinition	328	353	500	193	258
PrintHierachy	625	769	1111	357	455
AllCallsOn	78	103	143	51	67
AllImplementors	148	204	295	74	118
Inspect	239	364	433	118	105
Compiler	162	256	297	116	135
Decompiler	345	475	590	141	141
KeyboardLookAhead	282	503	531	173	258
KeyboardSingle	377	752	703	248	396
TextDisplay	152	319	311	93	145
TextFormatting	639	958	1162	411	523
TextEditing	232	441	440	145	227
Performance Rating	204	327	376	120	170

Benchmark Comparisons

Processor	Machine	Implementor	Performance (geoMean)
SPARC	Sun-4/280	ParcPlace Systems	376
SPARC	Sun-4/110	ParcPlace Systems	327
68020	Cadmus MSW/5	ParcPlace Systems/Georg Heeg, Inc.	250
68020	Sun-3/260	ParcPlace Systems	206
80386	Sun386i/250	ParcPlace Systems	204
68020	Cadmus MSW/4	ParcPlace Systems/Georg Heeg, Inc.	185
68020	Sun-3/60	ParcPlace Systems	163
80386	Compaq Deskpro 386 (20 MHz)	ParcPlace Systems	170
68020	HP 9000/350-c	ParcPlace Systems	151
68020	Macintosh II	ParcPlace Systems	127
80386	IBM PS/2 Model 70 (20 MHz)	ParcPlace Systems	120
68020	Sun-3/160	ParcPlace Systems	116
68020	Apollo Series 4000	ParcPlace Systems	112
proprietary	DoradoWorkstation	Xerox PARC/SCL	100
68020	Integrated Solutions, Inc.	ParcPlace Systems	100
68020	Sun-3/50	ParcPlace Systems	89
68020	Apollo DN580 Turbo	ParcPlace Systems	85
68020	Apollo Series 3000	ParcPlace Systems	75
68010	*Fuji Xerox 1161	Fuji Xerox	47
proprietary	*Fuji Xerox 1121	Fuji Xerox	41
68010	Fuji Xerox 1161	Fuji Xerox	41
68010	Atari Mega ST/4	ParcPlace Systems/Georg Heeg, Inc.	41
68010	Cadmus 9230	ParcPlace Systems/Georg Heeg, Inc.	40
68000	Macintosh SE	ParcPlace Systems	35
68000	Macintosh Plus	ParcPlace Systems	28
* (Japanese extended language)			

Picture Book

In this issue, the Picture Book highlights Cynergy, a new product that we are developing at ParcPlace Systems.

Cynergy

Cynergy is an interactive programming environment for C++ developed by ParcPlace. Cynergy will be based on the standard C++ compiler being sold by Glockenspiel, which incorporates the standard AT&T technology developed by Bjarne Stroustrup at AT&T Bell Labs at Murray Hill, NJ.

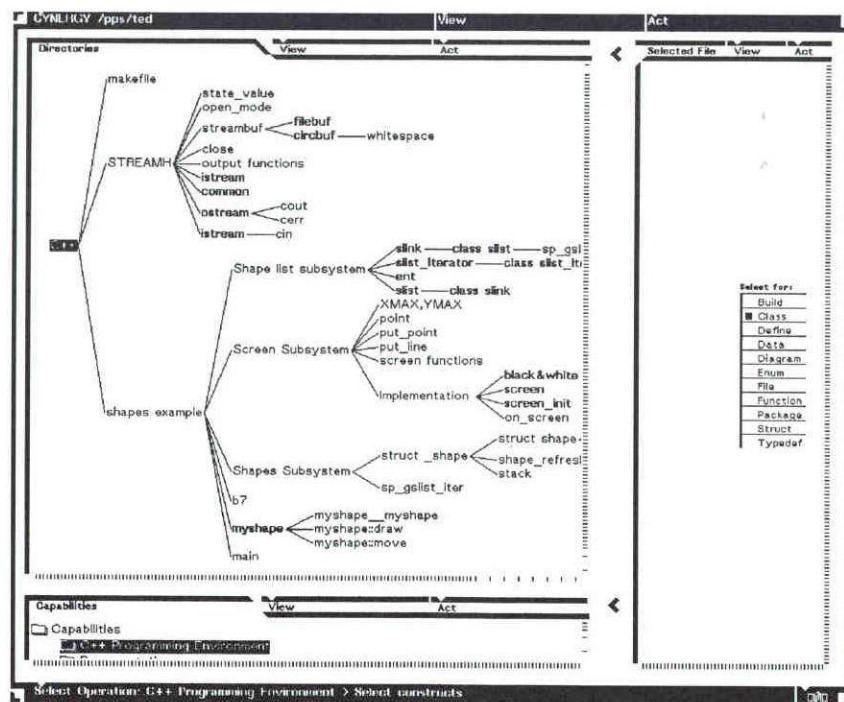
The Cynergy integrated programming environment provides:

- cut and paste editing,
- incremental compilation,
- automatic dependency control,
- access to host editors,
- incremental linking,
- an integrated source and object code debugger, and
- a code version change management utility.

It also provides a Smalltalk-80-like browser, in which a programmer can interactively create, modify, and search for code in a seamless manner. Future versions will support group development and large programming projects.

Cynergy's graphical user interface provides support for interactive dialog creation, and for tree and list manipulation—to give easy access to data structures, classes and functions—within a new "look and feel."

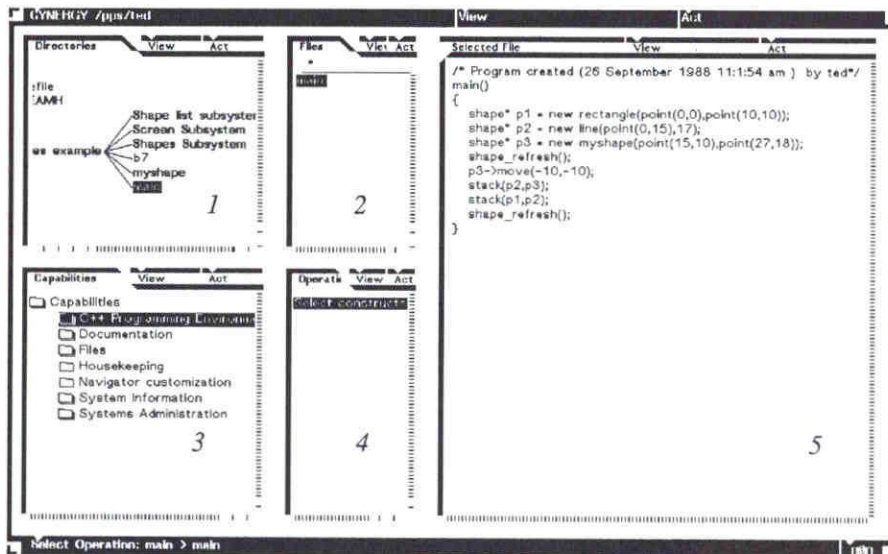
Cynergy is scheduled for release during the spring of 1989 on the Sun-3 workstation. Support for additional ParcPlace platforms is also planned. For more information, contact the Sales Department, ParcPlace Systems, 2400 Geng Road, Palo Alto, CA 94303.



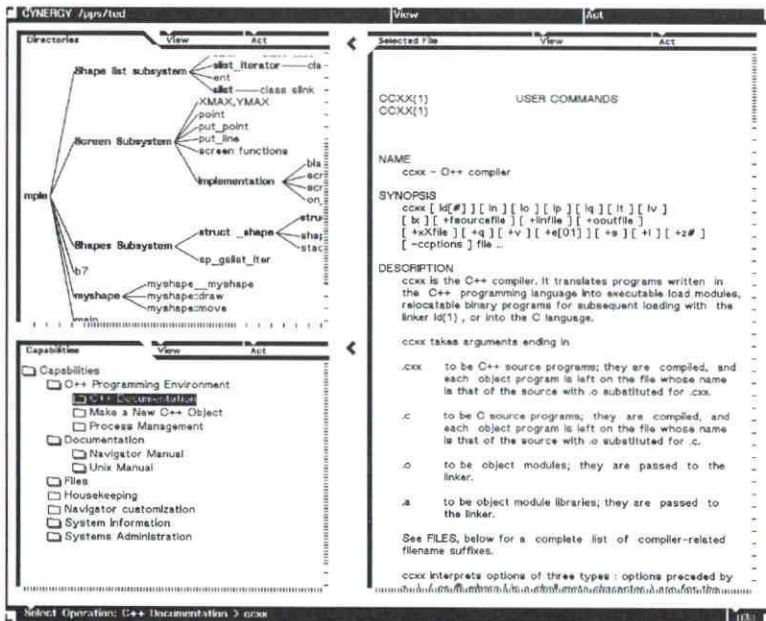
In a complex hierarchy of code, the Cynergy browser allows the programmer to filter the components in order to see the overall structure.

continued on next page

continued from previous page



The Cynergy Browser with its five basic subviews: (1) the Structure subview, showing the overall structure of the program; (2) the Node subview, indicating functional control; (3) the Capabilities subview, showing the various facilities you can access; (4) the Operations subview, giving fine grain control of the program; and (5) the Contents subview, or workspace, showing the actual contents of the program or file.



From within Cynergy you can access standard on-line documentation.

If you would like us to include your graphical applications in future installments of the Picture Book, please contact the Newsletter Editor at ParcPlace, (415) 859 1000.

Recent Smalltalk and Object-Oriented Programming Publications

The following is a selection of articles on Smalltalk-80 and its applications that appear in the *Proceedings of OOPSLA '88*, San Diego, CA, September 25-30, 1988.

Danny Epstein and Wilf R. LaLonde. "A Smalltalk Window System Based on Constraints."

Kenneth S. Rubin, Patricia M. Jones, Christine M. Mitchell, and Theodore C. Goldstein. "A Smalltalk Implementation of an Intelligent Operator's Associate."

C.A. Scaletti and R.E. Johnson. "An Interactive Environment for Object-Oriented Music Composition and Sound Synthesis."

S. Sridhar. "Configuring Stand-Alone Smalltalk-80 Applications."

Dave Thomas and Kent Johnson. "Orwell—A Configuration Management System for Team Programming."

David Ungar and Frank Jackson. "Tenuring Policies for Generation-Based Storage Reclamation."

Allen Wirfs-Brock and Brian Wilkerson. "An Overview of Modular Smalltalk."

Rebecca Wirfs-Brock. "An Integrated Color Smalltalk-80 System."

The following articles on Smalltalk-80 and its applications can be found in the *Proceedings of ECOOP '88*, Oslo, Norway, August 15-17, 1988.

Colin Low. "A Shared, Persistent Object Store."

Marcel Schelvis and Eddy Bledoe. "The Implementation of a Distributed Smalltalk."

Calendar

- Software Development '89, February 14-17, Hyatt Regency Hotel, Burlingame, CA. For information, contact Miller Freeman Publications, (415) 995 2471.

ParcPlace will exhibit at Software Development '89. Stop by our booth to visit, meet our staff and see the latest technology!

- CHI '89. Conference on human factors in computing systems, sponsored by ACM/SIGCHI and a number of other groups, will be held in Austin, TX, April 30-May 4. For information, contact Claudia Raun, MCC. (512) 338 3798.

- ECOOP '89. The third annual European Conference on Object-Oriented Programming will be held at the University of Nottingham, England, July 10-14. For information, contact Joanne Walsh, British Informatics Society, Ltd., 13 Mansfield Street, London W1M 0BP, UK. Telephone +44 1 637 0471.

- A Conference on Object-Oriented Database Management System Applications will be held at Santa Clara University, Santa Clara, CA, August 9-11. For information, contact Ms. Joy Neely. (408) 554 4929.

- IJCAI '89. The eleventh International Joint Conference on Artificial Intelligence, co-sponsored by the American Association for Artificial Intelligence, will be held in Detroit, Michigan, August 20-25. For information, contact Claudia Mazzetti at AAAI. (415) 328 3123.

- OOPSLA '89. The fourth annual ACM conference on Object-Oriented Programming Systems, Languages and Applications will be in New Orleans, October 2-6. For information, contact George Bosworth, at Digitalk. (213) 645 1082.

Smalltalk Users' Groups

OOPSTAD

Object Oriented Programming for Smalltalk Application Developers Association (OOPSTAD) is an independent non-profit organization, for the benefit of programmers, developers and users of Smalltalk. To join OOPSTAD and receive a subscription to the quarterly newsletter, *HOOPLA!* write: OOPSTAD, P.O. Box 1565 Everett, WA 98206-1565, or dial into their bulletin board at (206) 252 9048; OOPSTAD is also on UUCP and Applelink.

Portland PC Club's Smalltalk SIG

The Portland PC Club's newly formed Smalltalk Special Interest Group focuses on training new users and helping with more advanced application problems. Membership privileges include access to the Smalltalk area on the BBS and the *Silicon Forest Gazette*. For information, call the Smalltalk SIG leader, Kirk W. Fraser, at (503) 232 5147 or write c/o Expertek, 2534 S.E. Taylor, Portland, OR 97214.

S.F. Bay Area Special Interest Group

OOPSIG, the Object-Oriented Programming Special Interest Group, sponsored by the Software Entrepreneurs' Forum, holds meetings in the San Francisco Bay Area on the third Tuesday of every month. Avron Barr will speak at the December meeting. For more information, contact OOPSIG's chair, Dean Ritz, ParcPlace Systems, at (415) 859 1000.

To have your object-oriented or Smalltalk special interest group listed here, write to: Newsletter Editor, ParcPlace Systems, 2400 Geng Road, Palo Alto, CA 94303.

HyperScore User's Group

The HyperScore User's Group (HUG) was recently formed to provide a forum for musicians and computer scientists who are interested in HyperScore, a Smalltalk-80-based computer music tool. HUG currently distributes the unsupported version 5.0 of the HyperScore ToolKit (for the Macintosh, Atari, PC, and Sun). To join HUG, or order the HyperScore ToolKit, contact John Tangney, HUG, 1317 Masonic St., San Francisco, CA 94117.

New and Improved Applications in Smalltalk-80

- The Center for Computer Aids for Industrial Productivity (CAIP) at Rutgers University is currently using the Smalltalk-80 system to develop a user interface for a high performance scientific-engineering computational engine (Bolt, Beranek and Newman, Inc.'s Butterfly). The aim is to create an interactive computation system for scientific prototyping. The system is still under design but if you would like more information, contact Richard L. Peskin, Director, CAIP Parallel Computing Laboratory, Rutgers University, Piscataway, NJ 08855.

- Xerox Special Information Systems (XSI) has announced the release of an enhanced version (3.0) of the Analyst, an integrated information management system that runs in the Smalltalk-80 environment. For information, contact Tracy Loa, XSI, 250 N. Halstead, Pasadena, CA 91107-7018. (818) 351 2351, ext. 2614.

- Knowledge Systems Corporation (KSC), recently introduced the Smalltalk Collection, a library of reusable Smalltalk utilities and applications. For more information, contact Reed Phillips, KSC, 2000 Regency Parkway, Suite 270, Cary, NC 27511. (919) 481 4000.

Time-Sharing Citizenry for Smalltalk-80 under UNIX

by Allan M. Schiffman

Smalltalk-80's "total environment" personal-computing heritage and its emphasis on interactive performance have had the result that most implementations are imperfect citizens of time-sharing operating systems such as UNIX.

The current versions of the Smalltalk-80 system (2.3 and 2.4) address the most significant "platform citizenship" issue: coexistence with the platform's standard screen/window manager. Still, one major issue remains: some applications may be adversely affected by the control architecture of the MVC component of the Smalltalk-80 system (e.g., polling for interactive input) which uses CPU cycles even when "no useful work" is being done.

This architecture is quite appropriate in many circumstances, particularly where the Smalltalk-80 application is the primary application being run and other system processes provide mere housekeeping services. It is also appropriate when a user is interacting with both a Smalltalk-80 and a non-Smalltalk-80 application, or when a Smalltalk-80 program is cooperating with a non-Smalltalk-80 application.

However, MVC polling becomes particularly non-optimal when the Smalltalk-80 process must time-share with one or more compute-bound applications. In this case, Smalltalk may very well be consuming unnecessary CPU cycles, slowing down the compute-bound jobs. Moreover, some time-sharing schedulers may penalize Smalltalk-80 for being "compute-bound" as far as the scheduler is concerned.

But the beauty of the Smalltalk-80 system is that anything is possible and most things are easy! So, for applications that are willing to sacrifice a tiny amount of interactive response in order to cooperate

better, we present one implementation of the "time-sharing citizenry modification" here, in less than 100 lines of Smalltalk and 50 lines of C (including comments). This implementation makes use of the user-defined primitive interface supplied with the current Smalltalk-80 system.

Normally, when the system is idle (i.e., no compute-bound code is running at `UserSchedulingPriority` or higher, which

continued on next page

Note: The code listed in this article is also available on ParcPlace System's electronic bulletin board, ParcBench, (415) 859 1044.

The C Code

```
/*
 * await.c —
 * "Time-sharing citizenry" support for Version 2.3 (Sun-3 only)
 * 14-Oct-88 Allan M. Schiffman, ParcPlace Systems
 */
/*
 * All this primitive does is invoke the UNIX "pause" system call.
 * What pause does is wait until some UNIX signal occurs, then return.
 * Since any mouse, keyboard or timer activity will trigger a signal,
 * (in version 2.3 on Sun-3s!)
 * this primitive can be used by the caller to do the UNIX
 * equivalent of a Smalltalk "Processor yield", that is, give other
 * (UNIX) processes a chance to use CPU cycles.
 */

#include "userprim.h"

/*
 * <primitive: -3> (InputState class)awaitEvent.
 * Wait for some input or timer activity. Returns self. Cannot fail.
 */

upHandle
primAwait(receiver)
upHandle receiver;
{
    UPbegin(receiver, 0);

    /* wait for any Unix signal */
    pause();
    /* some activity occurred, return to VI */

    UPreturnHandle(receiver);
}

char *
UPinstall()
{
    UPaddPrimitive(-3, primAwait, 0);
    return "Time-sharing citizenry mod";
}
```

This code creates a user-defined primitive (number -3 used here) that does the UNIX pause system call. For more information on the user-defined primitive interface, see the Smalltalk-80 Programming System User's Manual. Note: this code applies only to 'async IO' UNIX implementations, currently version 2.3 on Sun workstations. With modification, it will run on version 2.4.

continued from previous page

is what happens in reaction to a menu pick), the system's user interface code is operating. Under the MVC control architecture this means either the method `ControlManager>>searchForActiveController` or `Controller>>controlLoop` is running.

Both of these methods have approximately the same effect; they poll the current mouse state (e.g., location and buttons) to see if the mouse has moved into a view (and should therefore activate a different controller) or if a mouse button has been pressed (and should therefore, say, pop-up a menu). This architecture delivers good response to interactive input, at the cost of constantly burning CPU cycles.

These "polling routines" quite correctly send `Processor yield` every trip around their loop to give other processes at the same priority a chance to run. Normally there are no other processes at `UserSchedulingPriority`, so the yield has no effect. The key to our "citizenry" implementation is that we provide a process that constantly runs at `UserSchedulingPriority`.

This process, which we call the `YieldProcess`, only gets control when other processes yield to it; it contributes no overhead when compute-bound Smalltalk processes are running. If the `YieldProcess` determines that there has been no interactive input for a sufficient time period, it invokes a user-primitive that makes a system call giving up the CPU until some interactive input occurs.

Try viewing CPU-time used via some standard utility (such as a Suntools `perfmeter`); look at the CPU usage before the change and then afterwards. Observe that with this modification, CPU usage drops to zero when you have not touched the keyboard or mouse for a while.

Allan M. Schiffman is Manager of System Architecture at ParcPlace Systems.

The Smalltalk-80 Code

```
**** Add two new class variables to InputState:
    SleepyCount
    YieldProcess

**** Change InputState>>run (this is mostly the code that is in the system
already, we're just adding the statement "SleepyCount ← 0." in three places):

run
    "This is the loop that actually processes input events."
    | word type param highTime lowTime |
    [true]
        whileTrue:
            [InputSemaphore wait.
             "Test for mouse X/Y events here to avoid an activation."
             word ← self primInputWord.
             type ← word bitShift: -12.
             param ← word bitAnd: 4095.
             "Mouse X" type = 1 ifTrue: [SleepyCount ← 0. x ← param]
             "Mouse Y" ifFalse: [type = 2 ifTrue: [y ← param]
             "Delta time" ifFalse: [type = 0
                                     ifTrue: [timeProtect critical: [deltaTime ← deltaTime + param]]]
             "Key down" ifFalse: [type = 3 ifTrue: [SleepyCount ← 0. self keyAt: param put: 1]
             "Key up" ifFalse: [type = 4 ifTrue: [SleepyCount ← 0. self keyAt: param put: 0]
             "Reset time" ifFalse: [type = 5
                                     ifTrue: [InputSemaphore wait.
                                                 highTime ← self primInputWord.
                                                 InputSemaphore wait.
                                                 lowTime ← self primInputWord.
                                                 timeProtect critical:
                                                     [baseTime ← (highTime bitShift: 16) + lowTime.
                                                      deltaTime ← 0]]
                                     ifFalse: [type = 7 ifTrue: [self primWindowEvent: word]
                                                ifFalse: [type = 8 ifTrue: [self primUserEvent: word]
                                                         ifFalse: [self primWindowEvent: 15 "Bad event type" ]]]]]]]]]

**** Change InputState>>install:

install
    "Initialize and connect the receiver to the hardware.
    Terminate the old input process if any."

    InputProcess == nil ifFalse: [InputProcess terminate].
    self initState.
    self class startYieldProcess.
    InputSemaphore ← Semaphore new.
    InputProcess ← [self run] newProcess.
    InputProcess priority: Processor lowIOPriority.
    InputProcess resume.
    self primInputSemaphore: InputSemaphore

**** Add two new methods to (InputState class) classified 'private':

awaitEvent
    "Wait for some indication of input activity to occur and
    give other system-level processes a chance to run in the meantime."

    <primitive: -3>
    ↑self

startYieldProcess
    "Start the yield process (terminate the old one if it exists),
    which does a await (yield to other OS processes) and then
    yields to other Smalltalk processes. Called by InputState>>install"

    YieldProcess == nil ifFalse: [YieldProcess terminate].
    SleepyCount ← 0.
    YieldProcess ← [true]
        whileTrue:
            [SleepyCount >= 1000
             ifTrue: [self await]
             ifFalse: [SleepyCount ← SleepyCount + 1].
             Processor yield]] newProcess.
    YieldProcess priority: Processor userSchedulingPriority.
    YieldProcess resume

**** Now (re)initialize the InputProcess and YieldProcess by executing in a workspace:

InputSensor install
```

1989 Smalltalk-80 Training Program

ParcPlace has just announced the 1989 schedule for the Smalltalk-80 Training Program. Monthly class offerings range from introductory through intermediate, to the masters' level.

Prices for the classes are:

Introductory Class	4 days	\$1,200
Intermediate Class	4 days	\$1,400
Masters Class	2 days	\$800

All classes include example software. Regularly scheduled classes are held in Palo Alto; on-site classes are available on request.

If you'd like more information or wish to enroll, contact: Training Manager, ParcPlace Systems, 2400 Geng Road, Palo Alto, CA 94303. Telephone (415) 859 1000 or (800) 822 7880. E-mail to training@ParcPlace.COM.

Schedule of Upcoming Smalltalk-80 Programming Classes 1989

February 14-17	Introductory
March 7-10	Intermediate
March 30-31	Masters
May 2-5	Introductory
June 6-9	Intermediate
June 22-23	Masters
August 15-18	Introductory
September 12-15	Intermediate
September 18-19	Masters
November 14-17	Introductory
December 5-8	Intermediate
December 14-15	Masters

To register for a ParcPlace Systems Smalltalk-80 Programming Class, complete this form and return it to: Training Manager, ParcPlace Systems, 2400 Geng Road, Palo Alto, CA, 94303. U.S.A.

Name: _____

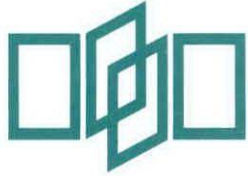
Organization: _____

Address: _____

Phone: _____

Class	Dates	Number of Persons Attending	Cost per Person	Total Cost

Check number: _____ P.O. number: _____ \$ _____



ParcPlace Systems
2400 Geng Road
Palo Alto, CA 94303

FIRST-CLASS MAIL
U. S. POSTAGE PAID
PALO ALTO, CA
Permit No. 378

ADDRESS CORRECTION REQUESTED

ParcPlace

N E W S L E T T E R



Winter 1990
Vol. 3 No. 3

Smalltalk marches into the mainstream

Market momentum builds behind *Objectworks\Smalltalk, Release 4*

Excitement in the trade press.

Unprecedented visibility at a mainstream goliath of a trade show.

Major interest from major corporations.

These are some of the trends that evidence a higher level of acceptance for Smalltalk in the corporate programming community, and specifically for Release 4 of *Objectworks\Smalltalk*. Release 4 began shipping for Sun, DEC and Macintosh users on November 21.

With features such as true color, platform-based graphics and windowing, international characters and, as before, binary portability across all supported platforms, a demonstration of Release 4 prompted one trade journalist to provide us with a memorable quote: "Well, I'll be damned!"

A long list of prestigious trade magazines have noted the arrival of Release 4 with great interest. A sampling follows:

Jon Udell, *Byte*: "Objectworks offers what is arguably the world's cleanest approach to portability...There's no recompilation."

Journal of Object-Oriented Programming: "[Release 4 is] more portable and flexible than any other object-oriented programming system."

Paul Krill, *UNIX Today*: "The new release also adds the Smalltalk Portable Imaging Model, which enables porting of graphics that will look the same on any supported platform."

Joshua Greenbaum, *Computer Systems News*: "The enhanced Smalltalk will be especially useful to integrators and corporate developers working on portable applications that include objects specific to a firm's business needs."

(continued on page 3)



PC Magazine, in its seventh annual Awards for Technical Excellence at COMDEX, presented a Lifetime Achievement Award to Adele Goldberg, president and CEO of ParcPlace Systems, for her contributions to the personal computer industry.

President's message: 1990 was a year of excitement, hard work

As we come to the end of the year and reflect on what we've accomplished, I am pleased to say that 1990 was a year of excitement and hard work. Our efforts to deliver object-oriented development tools to commercial organizations were well received, as object-oriented technology moved into the mainstream of the Fortune 500. Our new releases of both *Objectworks\C++* (Release 2) and *Objectworks\Smalltalk* (Release 4) were met with great interest industry-wide, and most importantly, within our user community. We thank you for your business, and we look forward to sharing many more years of working together.

As we look ahead to 1991, we see the momentum for Objectworks in both the C++ and Smalltalk communities continuing to increase. As the momentum builds, we will continue to deliver leading-edge development tools through enhancements to Objectworks and with add-on Objectkits. We will also increase our delivery of high-quality training for C++ and Smalltalk languages, and object-oriented software engineering. Thank you for your continued interest and support of ParcPlace. Best wishes for a prosperous new year. —Adele Goldberg, president and CEO

Inside...

- 2 Highlights of *Objectworks\C++*, Release 2
- 5 New service for third-party developers
- 6 Pictorial presentation of prominent parts of the new releases
- 8 Tech tip: bit caching
- 10 New directions in educational services
- 12 What's in a patent?

...and more

Objectworks\C++, Release 2, goes out the door with beta raves

Objectworks\C++, Release 2, started on its way to customers on November 21, laden with new features. Now, C++ programmers everywhere will learn what our beta users learned recently: C++ needs the added plus of *Objectworks*!

And Release 2 provides plenty of new pluses:

- Use any C preprocessor, compiler and linker.
- Step into and debug C as well as C++ code, aiding your migration to object-oriented technology.
- Import code four times faster than the previous release.
- Build much larger applications with the same memory as before.
- Use it with third-party source code control systems to facilitate team programming.
- Link object files into the executable directly, without the intervening step of creating a library.
- Use your customary text editor or the built-in editor.
- Includes Release 2.1 of AT&T's C++ Language System.

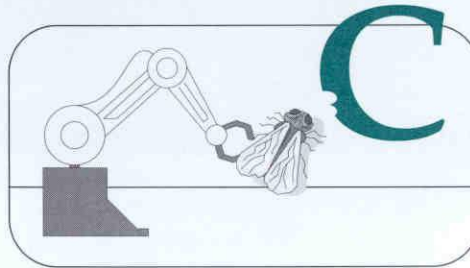
Beta users of *Objectworks\C++* are excited about the advances incorporated into Release 2.

"Too often C programmers buy C++ and think they're programming with object-oriented technology, but they're really just programming in C," says Paul

Brownlow of Data I/O. "I selected *Objectworks\C++* because the tools present code in a way that is consistent with object-oriented technology. That is particularly useful in

reducing the time it takes to develop and debug applications."

Two of the new features, in particular, won rave reviews from Chris Bloom,



For a pictorial look at the new features in *Objectworks\C++*, see page 7

a software engineer at cc:Mail. The C++ language has made the development of a portable Mail Engine possible, he notes, and "*Objectworks\C++* has made it easier by increasing productivity. *Objectworks* has the best graphical source browser and visual debugger I've ever used."

To complement the integrated tools that *Objectworks\C++* contains, class libraries are being offered in the optional *Objectkit\C++*:

- AT&T Standard Library and Standard Library Extension, which include code for managing lists, strings, time, associative arrays, complex arithmetic, concurrency and i/o streams. These libraries are also useful for simple finite-state machine development, error handling, fast special-purpose memory allocation, and measurement of execution time.
- Stanford InterViews Library, containing classes for building graphical user interfaces.
- National Institutes of Health C++ Library, which provides C++ transliterations of many useful Smalltalk classes, including arrays, dates, times, stacks, strings and semaphores.

Unlike the AT&T class libraries, the Stanford InterViews and NIH libraries are provided as a service to our customers—they are not supported by ParcPlace.

Objectkit\C++, scheduled to ship in the first quarter of 1991, will be available for US\$ 500 in North America—for pricing elsewhere, contact your distributor (listed on page 11). □

The new release was sent automatically to *Objectworks\C++* customers who have Object-support and annual updates.

...Smalltalk momentum

(continued from page 1)

Customers who got an early peek at Release 4 also contributed to the general excitement:

Diana Merry-Shapiro, Teradata Corp.: "The accessibility to platform graphics will allow us to develop more sophisticated graphics applications."

Brian Moore, Andersen Consulting: "As an end user, I believe the native host windows, fonts and color support are the big deal. But as a developer, the new imaging model is the big bang."

In spite of the breadth of improvements over Version 2.5, beta testers reported little difficulty in converting their applications with the aid of the conversion documentation and backward-compatibility code. Estimates of the time required to incorporate the new features in existing applications hovered around 10 percent of original development time.

Ken Auer of Knowledge Systems Corporation summarized the comments of many beta users, saying the conversion process for his applications was "nontrivial but nonpainful."

John Hebley, software project manager at Texas Instruments, Inc., says, "Not only are we excited about the new capabilities in Release 4, but ParcPlace has made the upgrade process from Version 2.5 fast, easy and straightforward."

While beta testers were pounding bare-fisted on the new release, Microsoft was proudly displaying *Objectworks\Smalltalk*, Release 4 in its pavilion at COMDEX. Why? Of all the applications that run under Microsoft's Windows 3.0, why show off our product?

"*Objectworks\Smalltalk* is one of the few software packages that utilizes the 32-bit mode," answers Catherine Tucker, the product's manager at ParcPlace. That gives twice the throughput of 16-bit programs, besides eliminating the 640 KB memory barrier. "Not to mention the fact that we provide their customers with transparent portability to the UNIX and Macintosh platforms," she adds.

Meanwhile, IBM gave a major boost to the corporate credibility of Smalltalk by licensing Smalltalk from ParcPlace and by announcing a development and license agreement with Digitalk, makers of Smalltalk/V. While Digitalk is nominally a competitor of ParcPlace, the news was welcomed by Douglas Pollack, vice president of marketing at ParcPlace.

"Of those companies that decide to try Smalltalk as a result of IBM's confidence in the technology," he says, "we will serve those who require a more robust, more extensive and, of course, more portable system."

ParcPlace is working with IBM to port *Objectworks\Smalltalk*, Release 4 to the RISC System/6000, a further demonstration of IBM's commitment to Smalltalk as an object-oriented development technology. That version of Release 4 will begin shipping in the first quarter of 1991, along with versions for HP 9000, Apollo and MS-Windows 3.0 users. □

Objectkit provides extra facilities for advanced programmers

The *Objectkit\Smalltalk Advanced Programming* was announced in September, offering reusable code modules and advanced programming tools beyond those provided in *Objectworks\Smalltalk*, Release 4. To be shipped in the first quarter of 1991, the kit will include:

- Parser compiler
- Terminal emulator
- Binary Object Storage System (BOSS)
- Checkers for memory statistics, errors in comments, and more
- Performance profiling utilities
- Hierarchy-wide protocol browser
- ProjectBrowser
- Benchmarks
- Complex and meta numbers
- Manual writer

Objectkit\Smalltalk Advanced Programming can be ordered immediately for U.S. \$500 in North America—for pricing elsewhere, contact your sales representative. □

ParcPlace to distribute Servio's Facets, a Smalltalk 4GL tool

Servio Corporation and ParcPlace Systems announced in December that ParcPlace will distribute Facets, a fourth-generation-language tool for developing applications in Smalltalk.

Facets is a comprehensive suite of tools, including: Schema Designer, Forms Designer, Report Writer, Menu Builder, and Help Author. Facets comes with complete source code as well as a hyper-text-based online help system. Servio's GemStone object database manager is not required.

Facets is available immediately for Version 2.5 of *Objectworks\Smalltalk*, for US\$ 1000 in North America. It will be available for Release 4 in the first quarter of 1991. □

Objectworks for Smalltalk-80 now on Japan's most popular PC

Fuji-Xerox Information Systems has made Objectworks for Smalltalk-80 Version 2.5 available on the NEC-PC9801, which has an installed base of more than 3 million systems in Japan.

The package includes support for the Japanese front-end processor ATOK7, widely used with the Japanese word processor Ichitaro. It also provides an interface for sending bitmapped images to the NEC-PR201 serial printer.

For further information, contact Fuji-Xerox (address on page 11). □

Tigre rolls out interface design kit with database manager

Unveiled at OOPSLA '90, the Tigre™ Programming Environment from Tigre Object Systems is a sophisticated application building tool consisting of an interface design kit and a multi-user object-oriented database manager.

Written in *Objectworks\Smalltalk*, Release 4, the Tigre Programming Environment cuts application development

time dramatically while offering unprecedented capabilities. For example, you can create multi-media applications with full-motion video and sound.

The Tigre Programming Environment is scheduled to ship during the first quarter of 1991. For more information, write to Tigre Object Systems, 11 Glen Canyon Rd., Santa Cruz, CA 95060 or telephone (408) 476-1854. □

Application Organizer 80 offered by Instantiations

Instantiations offers Application Organizer 80, a set of tools for developers of Smalltalk-80 applications. The product is designed to help programmers optimize designs, modularize code and minimize the size of the delivered application for Version 2.5 of *Objectworks for Smalltalk-80*. The package includes the following tools:

- Image Generator
- Inheritance Browser
- Package Browser
- Package Organizer
- Workspace Enhancements

For more information, contact: Instantiations, 921 S.W. Washington, #312, Portland, OR 97205, (503) 242-0725. □

Recent publications

Elizabeth Gibson, "Objects—Born and Bred," *Byte* magazine, October, 1991.

Wilf R. LaLonde and John R. Pugh, *Inside Smalltalk*, Prentice Hall, Englewood Cliffs, NJ, 1990 (two volumes).

Philippe Dugerdil, *Smalltalk-80, Programmation par objets*, Presses Polytechniques et Universitaires Romandes, Lausanne, Switzerland. 1990.

Adele Goldberg and Kenny Rubin of ParcPlace Systems recently completed a series of columns directed at project managers, in the *Hotline on Object-Oriented Technology (HOOT)*, SIGS Publication, Inc., 310 Madison Ave., Suite 503, New York, NY 10017. □

Calendar

ParcPlace will provide speakers and/or exhibits at the following conferences and trade shows during the first half of 1991:

UniForum '91, January 22-24, Dallas, Texas.

LOOK '91, February 4-6, Copenhagen, Denmark.

Software Development '91, February 13-17, Santa Clara, California.

Windows & OS/2 Conference, March 5-7, San Jose, California.

SCOOP West, March 11-14, San Jose, California.

Objectworld, June 3-6, San Francisco, California.

Look for ParcPlace distributors at these events:

LOOK '91, February 4-6, Copenhagen, Denmark.

TOOLS '91, March 4-8, Paris, France.

CeBit '91, March 13-20, Hanover, Germany. □

ParcBench®

ParcBench, the ParcPlace Systems bulletin board service, is open to any ParcPlace Systems customer. Dial into ParcBench on the data line, (415) 691-6716. ParcBench accepts calls at 300/1200/2400 bps, 8 bits, no parity and 1 stop bit. □

ParcRangers on campus

ParcPlace Systems encourages study, research and application development in *Objectworks\Smalltalk* and/or in *Objectworks\C++* on university campuses through the ParcRanger site licensing program. For more details, contact the ParcRanger Program at ParcPlace. □

Nationwide user group

If you are interested in participating in a nationwide Smalltalk User Group, contact Kathryn Oppenheimer at ParcPlace. □



PartnersPlace

This issue of the ParcPlace Newsletter introduces a new column, which will appear on a regular basis. Called PartnersPlace, this column will address a variety of issues relevant to ParcPlace Partners using *Objectworks\Smalltalk*. We'll start by introducing

Introducing... new support from ParcPlace for those who develop Smalltalk applications

the ParcPlace Partners program itself. In subsequent columns, you'll learn about individual Partners and the applications they offer for *Objectworks\Smalltalk* Release 4.

The new ParcPlace Partners program was created to further our support for third-party Smalltalk application developers and systems integrators, and to increase the awareness of the variety of applications being implemented in the Smalltalk language. ParcPlace Partners provides a support infrastructure for value-added resellers, systems integrators and independent software vendors who are increasing market acceptance of Smalltalk through the development and implementation of applications that address real-world business issues.

A variety of marketing and technical services that support the development, porting and sales of Smalltalk applications are provided to Partners. These services include:

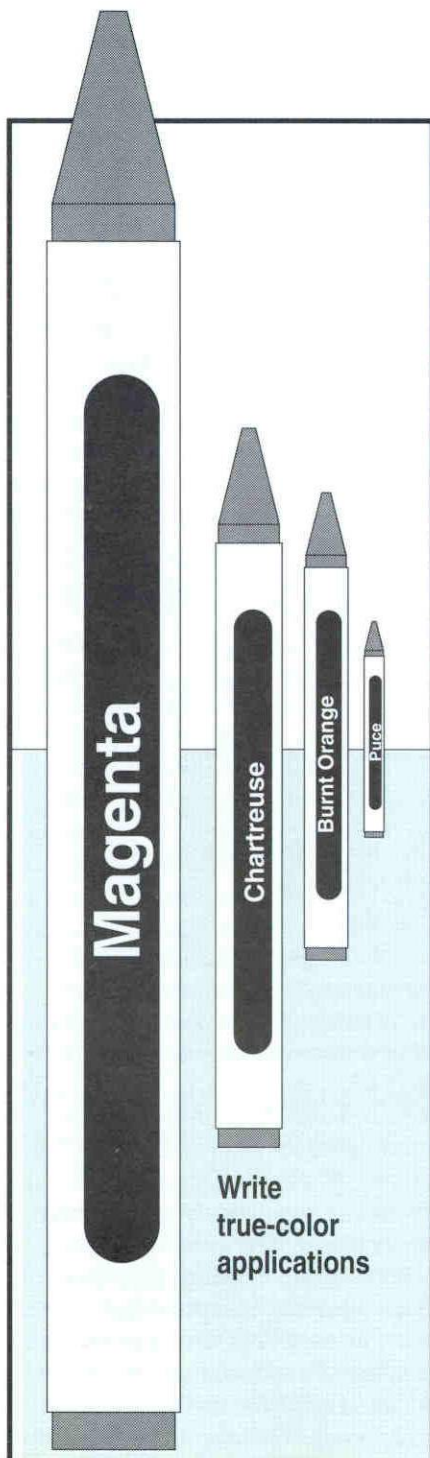
- Inclusion in Objects Illustrated Applications & Services Directory
- Co-marketing programs
- Development of case studies for distribution to trade publications as well as for internal use
- Participation in Partners Forums
- And, of course, appearances in the PartnersPlace column.

Through the Objectsupport program, technical services include: a support hotline, a technical newsletter, an electronic bulletin board, discounted software updates and early releases of new versions of *Objectworks\Smalltalk*.

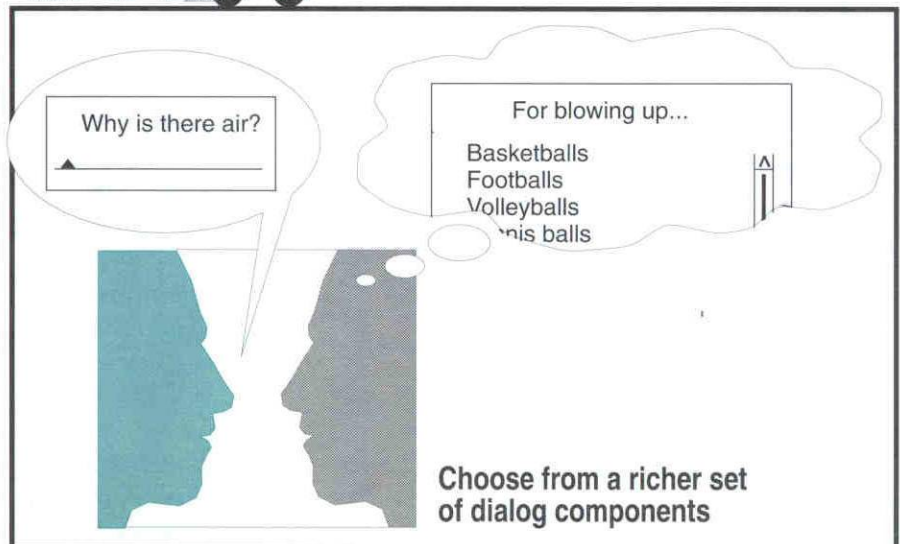
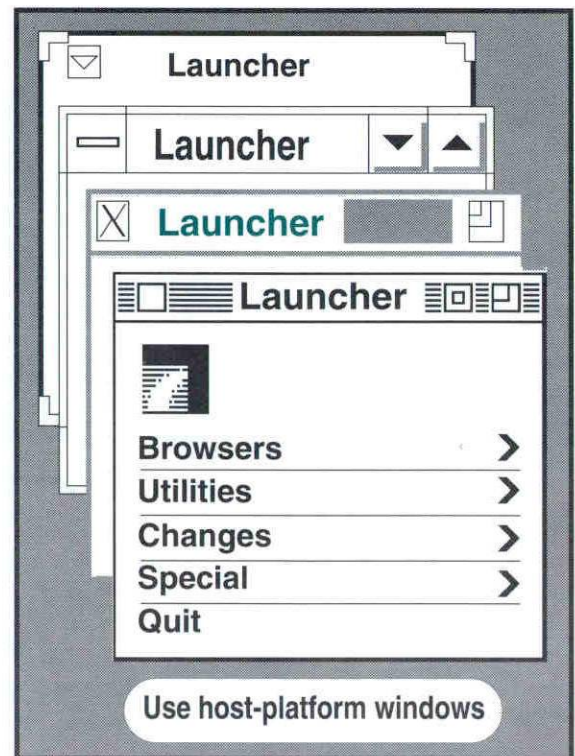
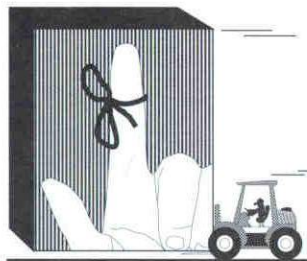
Partners are also eligible for significant discounts on initial development licenses and tuition for courses offered through ParcPlace's Educational Services.

To be eligible for the Partners Program, you must offer a commercially available product or service for *Objectworks\Smalltalk*. Applicants who are still in the process of developing or porting a product to *Objectworks\Smalltalk* are also eligible, provided they plan to ship within one year of the Partners Program application date. If you are interested in becoming a ParcPlace Partner, please contact Melissa McCutchin at (415) 691-6711 for further details and application requirements. □

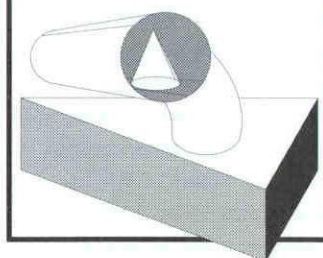
New features
in
Objectworks\Smalltalk, Release 4



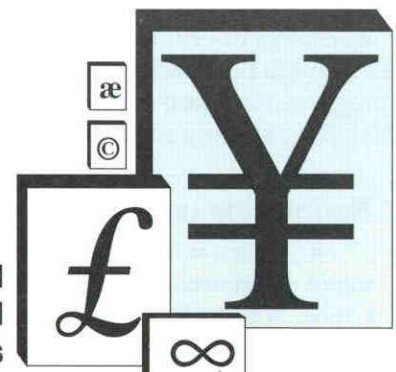
Customize the
memory policy



Draw pie slices, filled
polygons, elliptical
arcs, etc.

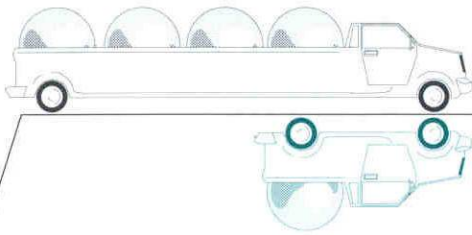


Transcend
ASCII
characters

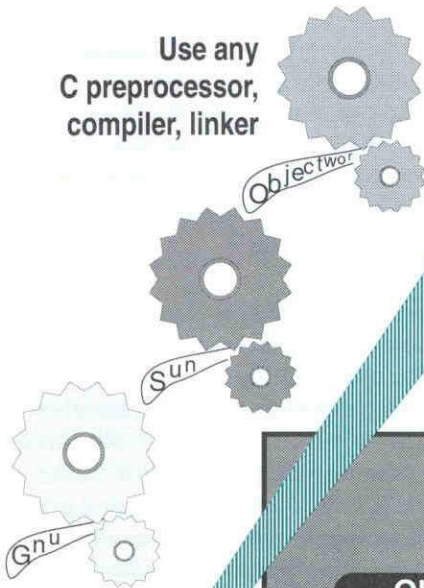


New features
in
Objectworks/C++, Release 2

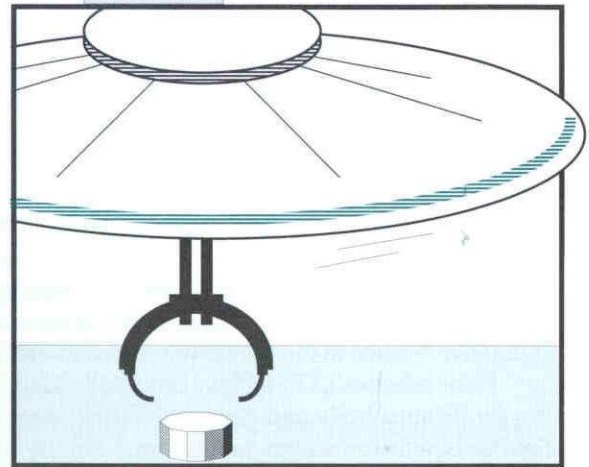
Put four times the code in the same memory



Use any
C preprocessor,
compiler, linker



Link object files
directly—no need
to put them in
libraries

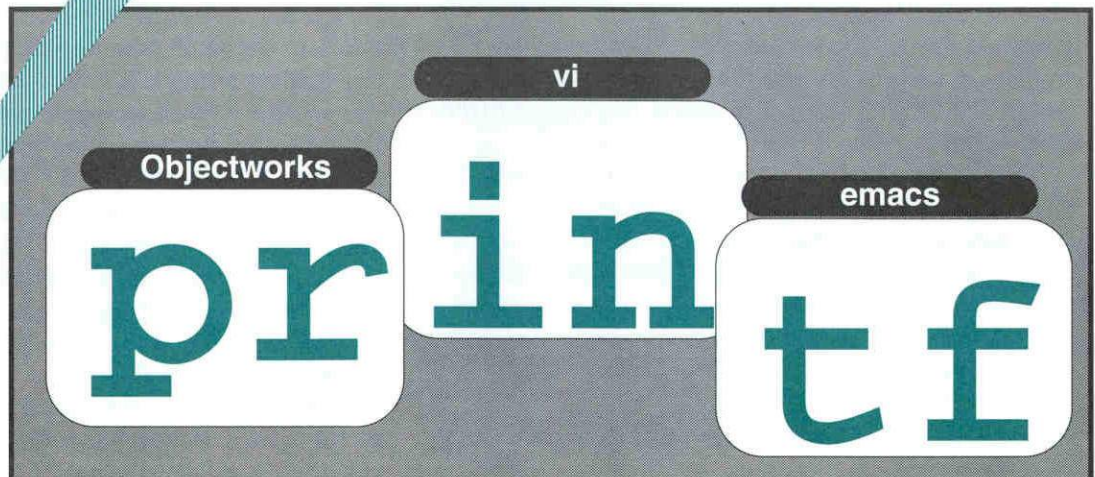


Import code
four times faster

Debug both
C and C++



Use any editor



Any window system that permits windows to overlap must provide a way to redisplay window contents in two different situations:

- When the contents of the window are changed, making the portion affected by the change *invalid*.
- When a previously obscured portion of the window becomes visible—that portion is said to be *damaged*, and requires *repair*.

In the former case, invalidation, the application has to redraw the affected area of the window with valid contents. However, damage repair, because it merely involves reinstating a previously drawn portion of the window, can be dealt with in two ways:

- Redraw the damaged area as if it were invalid.
- Routinely save window contents (typically in a bitmap called a *shadow buffer*) and arrange for the damaged area to be refreshed from this shadow buffer. This approach is known as *bit-caching*.

The bit-caching approach is uniformly fast, and relieves the application of the burden of repairing damage. On the other hand, each window requires a bitmap large enough to hold all of its pixels. The total requirement might be much larger than the size of the screen since windows can overlap. Also, the shadow buffer must be kept up-to-date, which can be awkward and time-consuming. For example, some window systems implement shadowing by doing every graphics operation twice—once to the window and once to its shadow.

Prior releases of ParcPlace Smalltalk used bit-caching for its simplicity and predictable responsiveness. Smalltalk windows were (at the time) strictly monochrome, so the space overhead seemed acceptable. Also, since the Smalltalk window system normally only permitted drawing to fully-exposed windows, the problem of keeping the shadow buffer consistent was not too difficult (in effect, the window was copied to the shadow buffer when it was about to be buried).

For Release 4, the situation changed in several ways, making bit-caching less appropriate:

- Release 4 presumes color, so shadow buffers might require 4, 8 or even 24 times more memory, depending on the screen's color resolution.
- Release 4 encourages display on windows that are partially or completely obscured.

Allan M. Schiffman is vice president for product technical strategy at ParcPlace.

Why bit caching is gone from Smalltalk (and how to bring it back)

by Allan M. Schiffman

- Release 4's redisplay control regime was designed around incremental redisplay (more on this below).

Host window systems typically provide for incremental damage repair, that is, when a portion of a window becomes visible, the window system supplies a *damage event* that notes the affected area. This permits great efficiency of redisplay since only the affected region needs to be repainted. This strategy is reflected in

Release 4's architecture—composite views are traversed after damage such that only affected subviews are required to redisplay (even affected views are provided clipping hints to minimize their work).

Basically, each window's sensor maintains a list of damaged regions. Damage notifications from the host window system are appended to this list, as are invalidation messages from the window's subviews. At appropriate times (such as when the controller begins its control activity) the damage list is iterated over, with each damaged area invoking a *#displayOn:* for the affected views (with a preset clipping region affecting only the damaged area).

This scheme works out quite nicely in practice, and is sufficient for all of the applications that are part of the standard programming environment. However, if you are porting applications from earlier versions of the system or if your applications have displays that are very time-consuming to redraw, even in part, you might prefer to have the old bit-caching back.

Easily done.

The code on page 9 implements class *CachingWindow*, a subclass of *ScheduledWindow*. It copies a window's contents into a shadow buffer every time the window's controller does *#controlTerminate* (although not very efficiently if the window is obscured) and damage repair is done using the shadow buffer. The shadow buffer is invalidated whenever any graphics operation is performed. To use this class, simply have your application instantiate a *CachingWindow* in place of a *ScheduledWindow*.

Caveat emptor, of course, and if you make serious use of this code you should make the obvious performance improvements, such as retaining the shadow buffer *Pixmap* when it is invalidated rather than discarding and recreating it each time. Even more appropriate for industrial-strength use might be a class *CachingWrapper* so that only individual (hard to redraw) views are bit-cached, rather than entire windows. But we'll save that for a future tech note... □

CachingWindow definition

```
ScheduledWindow subclass: #CachingWindow
instanceVariableNames: 'shadow '
classVariableNames: ''
poolDictionaries: ''
category: 'Interface-Framework'
```

CachingWindow comment

Instances of CachingWindow attempt to make damage repair go faster by saving a backing pixmap (the "shadow buffer") for the window. It does this by copying back the window's pixels into the shadow every time a "dirty" window makes the transition from "active" to "inactive". When the shadow is valid, damage repair is done by copying the shadow to the window. A window goes inactive on command from its controller and a window is considered "dirty" with respect to the shadow if any GraphicsContext is constructed on it (e.g. if it receives the message #graphicsContext).

Instance Variables:
shadow<Pixmap>shadow buffer for refreshing on damage

CachingWindow methods

accessing
deactivate

"Sent by the window's controller to say that the window is no longer active (its controller has relinquished control). For CachingWindows, this is the time to copy bits from the window into the shadow."

self saveShadow

graphicsContext

shadow := nil.
^super graphicsContext

events

displayDamageEvent: aDamageList

"Override ScheduledWindow>>displayDamageEvent: to refresh from shadow."

```
self shadowValid
ifFalse: [super displayDamageEvent: aDamageList]
ifTrue: [ | wgc pgc zero |
  pgc := shadow graphicsContext.
  wgc := super graphicsContext.
  zero := Point zero.
  1 to: aDamageList size do: [:i |
    | rect |
    rect := aDamageList at: i.
    wgc copyArea: rect
      from: pgc
      sourceOffset: zero
      destinationOffset: zero].
  self flush]
```

private

makeShadow

shadow := Pixmap extent: self extent

saveShadow

"Validate the shadow by copying the window's bits into it as needed."

```
self shadowValid
ifFalse:
  [ | wgc pgc damageList zero |
    self makeShadow.
    wgc := super graphicsContext.
    pgc := shadow graphicsContext.
    zero := Point zero.
    damageList := pgc
      copyArea: shadow bounds
      from: wgc
      sourceOffset: zero
      destinationOffset: zero.
    damageList isEmpty
      ifFalse:
        [pgc paint: self background.
          1 to: damageList size do: [:i |
            pgc displayRectangle: (damageList at: i)].
          self component == nil ifTrue: [^self].
          pgc paint: self foregroundColor.
          1 to: damageList size do: [:i |
            pgc clippingRectangle: (damageList at: i).
            self component displayOn: pgc copy]]]
```

shadowValid

"Answer whether the shadow buffer is correctly set-up for the window."

```
^shadow notNil and:
  [shadow isOpen and: [self extent = shadow extent]]
```

StandardSystemController method

basic control sequence

controlTerminate

"Provide a place in the standard control sequence for terminating the receiver (taking into account the current state of its model and view)."

^view deactivate

ScheduledWindow method

control

deactivate

"Sent by the window's controller to say that the window is no longer active (its controller has relinquished control). Default is to do nothing."

Education offerings updated for Release 4 and client needs

The *Introduction to Smalltalk* course has been revamped up to Release 4 of *Objectworks\Smalltalk*, with the first offering of the new course being held Dec. 3-7, soon after the product began shipping.

"Structurally, the course remains a comprehensive introduction to the language and the development tools," says David Arctur, course developer and instructor. "But we've done a lot of fine-tuning. No slide was untouched."

The course is intended for programmers who are new to object-oriented programming, the Smalltalk language, and *Objectworks\Smalltalk* as a development environment. Although the material touches on the new graphics and view-building facilities, those subjects are covered more thoroughly in the advanced class.

The advanced course, *Designing Interactive Applications in Objectworks\Smalltalk*, will be offered on dates to be announced later. The course is being redesigned because much of the class library relating to user interfaces has been modified. The new course will combine material from two former courses: *Intermediate Smalltalk-80* and *Prototype to Product*.

The *Object-Oriented Methodology Workshop*, which is presently offered only at client sites, has also been updated. It includes the following modules:

- Object-Oriented Software Engineering
- Object-Oriented Analysis
- Object-Oriented Design
- Object-Oriented Project Management

The Object Behavior Analysis segment has been extended to include more examples. In addition, an outline is provided for a system requirements specification, which can be used as a template in creating your own requirements for an object-oriented application. For further

Schedule of classes at ParcPlace Systems

1991	January 14-18	<i>Introduction to Smalltalk</i>
	March 4-8	<i>Introduction to Smalltalk</i>
	April 8-12	<i>Introduction to Smalltalk</i>
	June 3-7	<i>Introduction to Smalltalk</i>

For more information, or to register for a class, call (415) 691-6700 weekdays 8 - 5 p.m. Pacific time, or send e-mail to training@parcplace.com.

Additional U.S. offerings

Knowledge Systems Corporation offers ParcPlace's *Introduction to Smalltalk* course by request. For more information, contact Sam Adams or Bill Felton, KSC, 114 MacKenan Drive, Suite 100, Cary, NC 27511-6446, (919) 481-4000.

information about the methodology, refer to Dr. Gibson's article in the October 1990 edition of *Byte* magazine, titled "Objects: Born and Bred."

The project management material, compiled from interviews with application developers, has been the subject of a series of articles in the *Hotline on Object-Oriented Technology*. Dr. Adele Goldberg, president and CEO of ParcPlace, and computer scientist Kenny Rubin co-developed the course (and co-authored the HOOT series).

The course can be customized to meet client site needs, Dr. Gibson points out, by shifting the emphasis among the various modules of the methodology course.

"For example, analysis and design would naturally dominate when the audience is made up of front-line programmers," she says, "while management personnel might find the project management module more pertinent."

The content is independent of any particular object-oriented programming language—it has been well received by C++ teams as well as Smalltalk programmers. The course is suitable for a wide range of audiences. This flexibility has helped to make the *Object-Oriented Methodology Workshop* a very popular service. □

United States and Canada

Headquarters

ParcPlace Systems
1550 Plymouth St.
Mountain View, CA 94043
Telephone: (415) 691-6700
(800) 759-PARC
Facsimile: (415) 691-6715

Western Region

ParcPlace Systems
Attn: Mr. Will Doerlich
1550 Plymouth St.
Mountain View, CA 94043
Telephone: (415) 691-6734
Facsimile: (415) 691-6715
E-mail: wdoerlich@parcplace.com

Central Region

ParcPlace Systems
Attn: Mr. Steve Lance
7 W. Square Lake Rd.
Bloomfield Hills, MI 48302-0462
Telephone: (313) 452-5657
Facsimile: (313) 333-7999
E-mail: lance@parcplace.com

Eastern Region

ParcPlace Systems
Attn: Mr. Keith Sturges
1025 Old Country Rd., Suite 302-N
Westbury, NY 11590
Telephone: (516) 997-3805
E-mail: keith@parcplace.com

France

Mr. Georges Reich
TNI Industries
Z.I. du Vernis, Technopôle BREST-Iroise
Case Postale 1
F-29608 Brest Cedex, France
Telephone: 33.98052485
Facsimile: 33.98494533

Sweden, Denmark, Norway, Finland

Mr. Bo Segnestam
Enator Functional Products AB
Färögatan 7 - Box 1213
S-16428 Kista, Sweden
Telephone: 46.8.7507200
Facsimile: 46.8.7506394

Germany, Switzerland, Austria, Belgium, The Netherlands, Luxembourg

Mr. Georg Heeg
Smalltalk-80-Systeme
Baroper Strasse 337
D-4600 Dortmund 50
Federal Republic of Germany
Telephone: 49.231.975990
Facsimile: 49.231.9759920
E-mail: georg@unido.{uucp,bitnet}

Japan, Taiwan, Republic of Korea, Philippines, Thailand, Indonesia

Mr. Toshiyuki Nohira
Fuji-Xerox Co., Ltd.
13-16 Akasaka 3-Chome, Minato-ku
Tokyo 107, Japan
Telephone: 81.3.2249426
Facsimile: 81.3.5898721
E-mail: nohira.KSPAFOX@Xerox.COM

United Kingdom

Mr. Andy Smith
Artificial Intelligence International, Ltd.
Gresham House, 53 Clarendon Road
Watford, Herts WD1 1LA, England
Telephone: 44.923.247707
Facsimile: 44.923.247836
E-mail: ASmith@ail.co.uk

The ParcPlace Newsletter is published quarterly by ParcPlace Systems, Inc. Copyright ©1990 by ParcPlace Systems, Inc. All rights reserved.

Publisher: Adele Goldberg
Managing Editor: Brian Alexander
Associate Editor: Kathryn Oppenheimer
Contributing Staff: David Arctur, Duane Bay, Roy Benjamin, Greg Ching, Helen Cygnarowicz, L. Peter Deutsch, Roger Fisher, Elizabeth Gibson, Frank Jackson, Catherine Jaeger Tucker, Michael Khaw, Glenn Krasner, Michel Montandon, Bill Ostler, Douglas Pollack, Jodi Ranuio, Abe Saad, Allan M. Schiffman and Linda Siener.

Send submissions to: Newsletter Editor, ParcPlace Systems, 1550 Plymouth Street, Mountain View, CA 94043 — U.S.A. Or, send E-mail to newsletter@parcplace.com or ...!uunet!parcplace!newsletter. For subscription information, contact: Newsletter Subscriptions at ParcPlace.

Objectsupport, ParcNotices, ParcRanger and Smalltalk-80 are trademarks of ParcPlace Systems, Inc. ParcPlace, ParcBench and Objectworks are registered trademarks of ParcPlace Systems, Inc. Other product names may be the trademarks of their respective companies.

Printed on recycled paper.



ParcPlace Systems
1550 Plymouth Street
Mountain View, CA
94043 — U.S.A.

Commentary

Patent nonsense, and what to do about it

The laws regarding copyrighting and patenting of software (beyond the traditional use of copyright to prevent copying of actual code or specific screen images) are in a state of turmoil. There is controversy as to whether the new, broadened concepts of "look-and-feel copyright" and software patenting are helpful or harmful to the industry. Recent lawsuits and counter-suits have made it clear that this is an important issue, with high stakes.

At least one organization, the League for Programming Freedom, has taken a strong position that the broader kinds of protection are harmful to the industry. The following statement indicates our position with respect to the activities of this organization.

The management of ParcPlace Systems, Inc., supports the League for Programming Freedom in its efforts to eliminate software patenting and broad, nontraditional expansion of copyright laws to software. These laws and interpretations seriously compromise the benefits to customer and industry that have resulted from the open exchange of ideas, and from the freedom to compete by adding value to de facto standards.

Given the current existence of these laws and interpretations, ParcPlace Systems may nevertheless pursue patenting or broad copyrighting for some elements of its technology, as a matter of prudent business practice.

Nothing in this endorsement should be taken as implying or requiring that any particular employee of, or other person affiliated with, ParcPlace Systems, Inc., individually supports the League for Programming Freedom or its activities.

To find out more about the League for Programming Freedom, write to them at:
PO Box 9171, Cambridge, MA 02139. □

Angela Coppola
Central Intelligence
7404 Arden Rd
Cabin John, MD 20818

FIRST CLASS MAIL
U.S. POSTAGE PAID
MOUNTAIN VIEW, CA
PERMIT NO. 246

ParcPlace

N E W S L E T T E R



Spring 1992
Vol. 5 No. 1



ParcPlace names William Lyons as new president and CEO

On April 29, ParcPlace Systems named former Ashton-Tate CEO William P. Lyons as its president, CEO and a member of the board of directors. Adele Goldberg, the founder of ParcPlace and co-inventor of the Smalltalk language, will continue as chairman of the board.

(continued on page 6)

New Objectbuilder\C++ provides drag-and-drop interface design

ParcPlace Systems now offers the first user interface builder for C and C++ programmers that supports both the Motif and OPEN LOOK standards for X Window System applications.

Objectbuilder\C++ is a comprehensive set of tools for developing user interfaces that make use of the Objectkit\OI library of graphical user interface classes. In addition, Objectbuilder\C++ provides the following capabilities:

- **WYSIWYG environment**—The What-You-See-Is-What-You-Get development environment of Objectbuilder\C++ features a drag-and-drop style of interaction. You just pick an element such as a dialog or menu from a palette and drop it into your application window.
- **Automatic code generation**—The C++ source code and makefiles required to build a runnable user interface are automatically generated. Customizations that you make to

(continued on page 2)

Chairman's message: New interfaces to users and applications

A few years ago, we made a commitment to support object-oriented software developers in the C++ realm as well as those in Smalltalk land. With the introduction of Objectbuilder\C++ (see story above), we can now offer our customers a high-level tool for assembling user interfaces swiftly, in addition to a well-integrated development environment for coding the underlying applications.

Our friends at Solbourne developed Objectbuilder and its foundation library (see the Objectkit\OI story on page 2). Their customers have been using the library and interface-building tools for some time, motivating several cycles of refinement in the product. As a result, ParcPlace can introduce a mature, time-tested interface builder to extend our support for the C++ programmer. In addition, at the recent Uniform, we gave a preview of our new interface for Objectworks\C++, which complies with both Motif and OPEN LOOK.

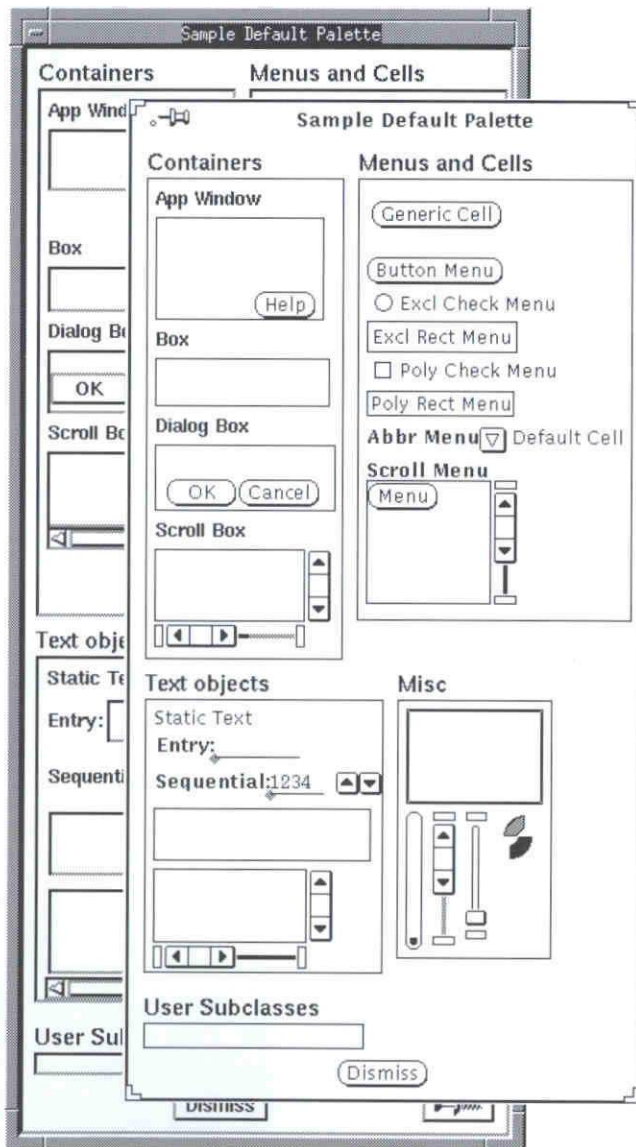
And speaking of interfaces, the new Objectkit\Smalltalk C Programming provides a convenient Smalltalk interface to applications written in C (see page 5).

—Adele Goldberg, chairman of the Board of Directors

Inside...

- 2 The kit behind the new Objectbuilder\C++
- 3 Recent publications
- 4 What's in the new release of Objectworks\Smalltalk?
- 5 A better doorway from Smalltalk to C
- 6 ParcPlace is moving, figuratively and literally
- 8 Do you know your ABC's objectively?

...and more



Objectbuilder\C++ lets end users choose Motif or OPEN LOOK from same executable

(continued from page 1)

that code are recognized and automatically integrated into newly generated versions of the user interface.

- **C, too**—Existing C-based Xevent callbacks are easily called from C++ and Objectkit-based application code. Your investment in C is protected.
- **Custom palettes**—Several palettes of graphical objects are included to access the standard user interface objects provided by Motif and OPEN LOOK. You can create new palettes that contain subsets of these objects, customized to suit your applications and internal standards.
- **Object editors**—Every object in the user interface has a variety of attributes that control the display and function of the object. Objectbuilder\C++ provides resource editors for controlling colors, fonts and other key aspects of your user interface.
- **Custom objects**—You can create new OI classes and add them to Objectbuilder\C++ to implement application-specific functions.

Objectbuilder\C++ is available on the SunTM SPARC[®] workstation under Sun OS 4.1 running X Window SystemTM X11R3, R4, R5, or OpenWindows 2.0 and 3.0. A C++ language system conforming to USL C++ 2.1 is also required (and available from ParcPlace).

Objectbuilder\C++ is now available. The standard price of this new product is US\$2,995, and Objectsupport for one year is US\$450. ParcPlace is offering Objectbuilder\C++ at the special introductory price of US\$1,895 through June 30, 1992. Contact your sales representative to place an order. □

New Objectkit\OI complies with both Motif and OPEN LOOK standards

ParcPlace's new Objectkit\OI is the first C++ class library that complies with both the Motif and OPEN LOOK standards.

Objectkit\OI is a library of C++ user interface classes that simplifies the development of Motif and OPEN LOOK Graphical User Interfaces (GUIs) for X Window System applications. By applying member function callbacks, dynamic reparenting, subclassing, translations and X resource control, Objectkit\OI increases the power of your X applications.

Motif, OPEN LOOK 2D, and OPEN LOOK 3D are supported in both the development and runtime environments. A single set of source files is compiled and linked to a single library to produce a single executable image. Your executable can be either statically or dynamically linked with Objectkit\OI and run as either a Motif or an OPEN LOOK application, selectable at runtime. Thanks to the library's object-oriented technology, code size is not

(continued on page 3)

New Objectkit\OI provides interface library

(continued from page 2)

increased even though your applications support both the Motif and OPEN LOOK standards for user interface appearance and behavior.

Model-dependent aspects such as pushpins, keyboard accelerators, and objects that exist in only one model are handled automatically. Wherever possible, the desirable features specified by only one model are provided in all models; you don't need to code specifically for a particular model. Any object can be created in any model, allowing models to be mixed on different screens or even on the same screen.

The end user selects the desired look and feel using a command-line argument or an X resource file.

Other features

- **16-bit fonts**—16-bit fonts are supported. Application text is maintained in standard X-resource files, allowing localization of applications without re-compiling or relinking.
- **End-user customization**—Runtime command-line arguments and X resources determine an Objectkit\OI application's interaction model and appearance, including object attributes such as color, font, labels, language, help context, translations, accelerators and mnemonics.
- **Ease of use**—All of the facilities required to build sophisticated user interfaces are provided. The open system philosophy of Objectkit\OI also provides experts with the lower level interfaces required for complete control over the appearance and behavior of the user interface.
- **Full C++ subclassing**—New classes can be created, derived either from Objectkit\OI classes or independent of them.
- **Built-in Hypertext help**—The help system is context-sensitive and fully supports the use of embedded Objectkit\OI objects and bitmaps.
- **Compatible with existing C language sources**—Existing C-based X event callbacks are easily invoked from C++ and Objectkit\OI-based application code.
- **Dynamic layout control**—Objectkit\OI's layout facility enables the automatic alignment of objects by rows, columns, or at object labels. Object alignment is dynamically updated as you modify at-

tributes such as font and language. Spacing between objects can also be controlled precisely.

Objectkit\OI is now available on the Sun SPARC workstation under Sun OS 4.1 running the X Window System X11R3, R4 and R5; OpenWindows 2.0 and 3.0. A C++ language system conforming to USL C++ 2.1 is also required. The price of this new product is US\$995, and Objectsupport for one year is US\$225. Contact your sales representative to place an order. □



Recent publications

PC/Computing—"Windows development tool kit," Nov., 1991.

UNIX Review—"ParcPlace Systems," November, 1991.

Datamation—"Universal database access!," November 1, 1991.

MacWEEK—"Smalltalk developers roll out application-building tools," November 12, 1991.

Electronic Engineering Times—"Object-oriented programming tool debut," Nov. 18, 1991.

PC Week—"OOP vendors join forces on workbench," December 2, 1991.

Computerworld—"Garbage collectors do dirty work of OOP," December 16, 1991.

PC Week—"C++ systems help developers locate, fix coding errors," December 16, 1991.

PC Week—"ParcPlace Systems' Objectworks\C++ 2.4," December 16, 1991.

PC Week—"C++ tools ignite programming SPARC," December 16, 1991.

UNIX Today!—"The staying power of Smalltalk," January 24, 1992.

UNIX Today!—"Taking the 'Lego' approach to software design," January 24, 1992.

PC Week—"Objectworks\C++ release 3 to support Motif GUI, RS/6000," February 10, 1992.

UNIX Today!—"Smalltalk speaks to users' needs," February 17, 1992.

UNIX Today!—"Designing wafers the Smalltalk way," January 24, 1992.

UNIX Review—"Why not Smalltalk," May, 1992.

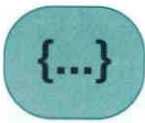
UNIX Review—"Integrated C++ environments, getting better and better," May, 1992.

For reprints of any of these articles, contact Laurie Cox at (415) 691-6700. □

Release 4.1 of Objectworks/Smalltalk features a wealth of improvements

The next release of Objectworks\Smalltalk is scheduled to begin shipping at the end of May, having completed extensive beta testing. Release 4.1 contains several interesting enhancements in diverse areas of the system, at the same time maintaining a high degree of backward compatibility with Release 4.0.

External language support



Perhaps the most important development is a new package that greatly simplifies and enhances the interface to C libraries. Deployment support for external language access is incorporated in Objectworks\Smalltalk, and a C declaration parser for developers is sold separately. With the parser, you can automatically generate Smalltalk methods that invoke functions in one or more C libraries. In operating environments that support dynamic linking—Sun OS, HP-UX and MS-Windows—you can take advantage of that capability. For more details, see the article about Objectkit\Smalltalk C Programming on page 5.

Permanent space and sharing



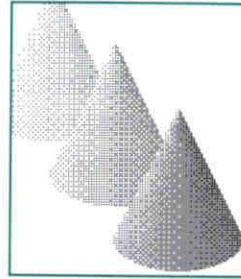
This release adds a new generation to the memory manager, which we call PermSpace (Permanent Space). PermSpace houses semi-permanent objects, including nearly everything in the delivered image. These objects are safely ignored by the high-frequency garbage collectors, resulting in much faster reclamation of memory and more timely finalization. Dead objects in PermSpace can still be reclaimed by invoking the new global garbage collector. On multi-user Sequent machines, PermSpace can be shared by all users of the same image, greatly reducing redundant use of memory resources.

Geometric classes



This release includes classes that represent several commonly used geometric entities. In addition to the existing rectangles, classes have been added for line segments, polylines, splines, Bezier curves, circles and elliptical arcs.

Enhanced documentation on graphics, MVC, deployment



Release 4.1 features enhanced documentation, including greatly expanded chapters on graphics, color, application frameworks (MVC) and deployment. These new chapters feature a generous use of illustration and more than 100 fully executable examples. In addition, the Beginning Tutorial has been expanded with a series of warm-up exercises that illustrate basic Smalltalk syntax, and the application-building chapters have been structured into two-page modules for greater clarity.

ParagraphEditor cleanup and cursor keys



The ParagraphEditor implementation has been streamlined. For example, it no longer requires extra copies of the text for **cancel** operations, which improves performance. Also, cursor keys are enabled.

User event handling

All input events are passed into the VI, rather than only those events that Smalltalk recognizes. This allows application code to handle menu events, for example.

Iconified window support



This release extends the window API to include knowledge and support of whether a window is collapsed to its iconic form.

EdgeWidget/ScrollWrapper redo

Edge widget wrappers, and especially scrolling widgets, in Release 4 were not consistent with the rest of the visual object world. This release re-implements that functionality in a way that is consistent with the remainder of the system.

Enhanced font matching



Font matching has been improved and, as with other systems, we split mechanism away from policy, allowing users to change policies more easily.

(continued on page 5)

Next release of Objectworks\Smalltalk

(continued from page 4)

Graphics performance

Release 4.1 provides support for all standard graphic configurations (1, 2, 4, 8 and many-bit) and its graphics primitives have been re-engineered for performance.

BOSS in base product

Because of its fundamental nature, we brought the Binary Object Storage System (BOSS) back from the Objectkit\Smalltalk Advanced Programming, returning it to the base product.

Better CPU citizenship



We improved the algorithm for yielding the CPU to competing host processes. Rather than continuing to run full-speed for 2.5 seconds after user activity has ceased, the system falls into a quiescent state almost immediately for progressively longer periods of time. The result is better performance for competing applications, without sacrificing responsiveness within Objectworks\Smalltalk.

Release 4.1 includes a large number of other enhancements, and performance has been improved in a number of areas.



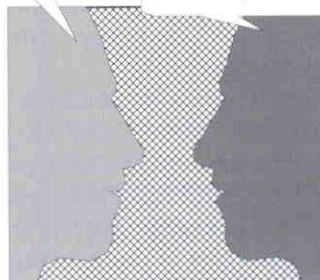
Objectworks\Smalltalk Release 4.1 for Sun SPARC, IBM RS/6000, MS-Windows, Macintosh, HP 700, DECstation and Sequent machines is scheduled to begin shipping in June. The price of this upgrade is US\$400. If you have Objectsupport, the price is US\$150. Objectsupport customers can purchase both Release 4.1 and Objectkit\Smalltalk C Programming for US\$500 until July 31. Contact your sales representative to order your upgrade or to purchase a new license. □

Unixpac signs on as Australian distributor for ParcPlace

Sydney-based Unixpac has joined ParcPlace's growing roster of distributors worldwide (see complete contact information on page 7). Unixpac distributes and supports software applications for all major UNIX workstations, in Australia and New Zealand. □

road isDirty
ifTrue: [roadCrew beautify: road]

if (dirty(road))
beautify(road, roadCrew);



Objectkit\Smalltalk C Programming offers convenient doorway to C world

Smalltalk developers can now link C libraries to their Smalltalk applications—dynamically, on platforms that support dynamic linking. In addition, you can make callbacks into Smalltalk from your C code.

Objectkit\Smalltalk C Programming links the Smalltalk and C worlds. It gives you a simple means of creating Smalltalk applications that take advantage of external C libraries, code and applications available in today's open systems environment. Here are some of the opportunities this Objectkit presents to you:

- Make use of platform resources available through system libraries
- Take advantage of commercially available libraries
- Leverage already-developed C code
- Access specialized hardware through C application programming interfaces

All of this without having to write a single line of C code.

This Objectkit works with Release 4.1 of Objectworks\Smalltalk, which has a new external language interface. This interface includes deployment support for applications that make use of C code, so your customers don't have to buy an Objectkit to run your applications.

Objectkit\Smalltalk C Programming provides specific capabilities that allow you, the developer, to take full advantage of the external language interface delivered with Objectworks\Smalltalk, including:

- A simple interface for creating Smalltalk methods that directly access C functions

(continued on page 6)

ParcPlace names Bill Lyons as CEO

(continued from page 1)

"ParcPlace is to object-oriented technology what Macintosh is to the graphical user interface—and they both came from Xerox PARC," Lyons said. "This is a terrific technology that is on the verge of intersecting with a huge customer requirement."

As chairman of the board, Goldberg continues to provide strategic vision to the ParcPlace product lines of object-oriented development environments. She will also continue to work with ParcPlace customers to help them improve the way application software is developed by lowering the cost of doing business and increasing the opportunities for innovation.

"Bill's experience with corporate developers and end users at IBM and Ashton-Tate will augment our efforts to penetrate the corporate marketplace," Goldberg said. "I am delighted to have attracted someone of Bill Lyons' caliber to ParcPlace as my business partner."

All ParcPlace vice presidents now report to Lyons, who has assumed day-to-day management responsibilities.

Lyons brings to ParcPlace 22 years of experience in sales, marketing, R&D, operations and general management. He spent three years at Ashton-Tate Corporation, most recently as chairman and CEO. While at Ashton-Tate, Lyons completed the merger of the company with Borland International.

Before Ashton-Tate, Lyons spent 19 years at IBM. His last assignment was vice president of software marketing for the National Distribution Division of IBM. Lyons was involved with IBM's personal computer division since 1983, when he was named to head the Entry Systems Division's product management and marketing teams in Boca Raton, Florida. Prior to his personal computer experience, Lyons held a variety of sales and marketing management positions in IBM's Data Processing Division. □

Objectkit\Smalltalk C Programming

(continued from page 5)

- A tool that parses .h files and automatically generates Smalltalk methods, which translate between Smalltalk objects and C data types
- Type checking that helps ensure safe passage across the "bridge"
- Callbacks from C into Smalltalk
- Dynamic loading and unloading of libraries (where supported by the platform)

Accessing C functions is accomplished through a simple Smalltalk interface. A method consisting of a single line of Smalltalk code is all that is necessary to access a C function. In case of a failure, exception handling code can be written in Smalltalk. These methods are part of the Smalltalk environment, and can be referenced by all of the standard system browsers and cross-referencing tools.

The C Programming Objectkit includes an ANSI style C declaration parser that will read .h files and automatically construct Smalltalk methods that parallel the C declarations.

Objectkit\Smalltalk C Programming Release 1.0 for Sun SPARC, IBM RS/6000, MS-Windows, Macintosh, HP 700, DECstation and Sequent machines begins shipping on May 29. The price of this new product is US\$500—support is included with Objectsupport for Objectworks\Smalltalk. Contact your sales representative to place an order. □

Arbor offers two kinds of Object Bridge

Arbor Intelligent Systems has announced availability of its Object Bridge™ in two forms, one for accessing a variety of databases and the other for accessing Neuron Data's Nexpert Object™ expert system from within Objectworks\Smalltalk.

Both products are available on Sun SPARC, MS-Windows and Macintosh workstations. The Nexpert Object version is shipping now, and the database bridge is scheduled for shipment in May, 1992. The database bridge will provide access to Oracle, Ingres, Sybase, DB2, RDB, SQL/DS, dBASE III, Lotus and Excel.

For further information, contact: Arbor Intelligent Systems, 506 N. State St., Ann Arbor, MI 48104; telephone (313) 996-4238; fax (313) 996-4241; email ais@um.cc.umich.edu. □

ParcPlace expanding to new quarters

ParcPlace Systems will move its main offices in the United States to a new location in early June. The larger quarters are needed to accommodate our continuing growth. Our 800 number will remain the same: (800) 759-PARC. The new address is:

ParcPlace Systems
999 E. Arques Ave.
Sunnyvale, CA 94086-4593
(408) 481-9090 (Voice)
(408) 481-9095 (Fax) □

United States and Canada

ParcPlace Systems *Headquarters*

999 E. Arques Ave.
Sunnyvale, CA 94086-4593
Phone: (408) 481-9090 after 6/10/92
(800) 759-PARC
Fax: (408) 481-9095 after 6/10/92

ParcPlace Systems
Attn: Mr. Steve Lance
7 W. Square Lake Rd.
Bloomfield Hills, MI 48302-0462
Phone: (313) 452-5657
Fax: (313) 333-7999

ParcPlace Systems
Attn: Mr. Gerald Swezy
6 New England Executive Park, Suite 400
Burlington, MA 01803
Phone: (617) 863-0511
Fax: (617) 863-0523

ParcPlace Systems
Attn: Mr. Greg Ching
200 Rector Place, Suite 7E
New York, NY 10280
Phone: (212) 945-6436
Fax: (212) 945-6451

ParcPlace Systems
Attn: Mr. Herb Hayde
8300 Boone Blvd., Suite 529
Vienna, VA 22182
Phone: (703) 848-9225

ParcPlace Systems
Attn: Mr. Tim Milovich
14001 Dallas Parkway, Suite 1200
Dallas, TX 75240
Phone: (214) 934-6556
Fax: (214) 934-6792

Australia

Mr. Gary Donin
Unixpac Pty. Ltd.
Level 8, Prudential Centre,
495 Victoria Ave., Chatswood,
NSW 2067, Sydney, Australia
Phone: (61) 2-41-26-304
Fax: (61) 2-41-52-393

France

Mr. Thierry Guéguen
TNI Industries
Z.I. du Vernis, Technopôle BREST-Iroise
Case Postale 1
F-29608 Brest Cedex, France
Phone: (33) 9-80-52-485
Fax: (33) 9-84-94-533

Mr. Jean-Marie Clisson
HumanWare
66, Champs-Élysées
F-75008 Paris, France
Phone: (33) 1-44-35-7000
Fax: (33) 1-42-89-0597

Germany, Switzerland, Austria, Belgium, The Netherlands, Luxembourg

Mr. Georg Heeg
Objektorientierte Systeme
Baroper Straße 337
D-4600 Dortmund 50
Federal Republic of Germany
Phone: (49) 2-31-97-5990
Fax: (49) 2-31-97-59920

Italy

Mr. Giorgio Cioni
Artificial Intelligence Software SpA
Via Rombon 11
I-20134 Milano, Italy
Phone: (39) 2-26-40-107
Fax: (39) 2-26-41-0744

Japan

Mr. Toshiyuki Nohira
Fuji-Xerox Co., Ltd.
13-16 Akasaka 3-Chome, Minato-ku
Tokyo 107, Japan
Phone: (81) 3-35-89-8601
Fax: (81) 3-55-70-1929

Korea

Mr. Kyung Won Chong
SsangYong Computer Systems Corp.
60-1, 3ga Chungmu-Ro, Jung-gu
Seoul 100-705, Korea
Phone: (82) 2-27-04-437
Fax: (82) 2-27-72-385

Mexico

Mr. Octavio Orozco
TEIX Sistemas De Información Estratégica, S.A. De
C.V.
Antonio Anza 43, D.F. 06700,
Col. Roma, Mexico
Phone: (525) 564-7146
Fax: (525) 264-6559

Sweden, Denmark, Norway, Finland

Mr. Tony Markhill
Memory Data AB
Korta Gatan 7, Box 1467
S-171 28 Solna, Sweden
Phone: (46) 8-76-46-700
Fax: (46) 8-29-9903

Taiwan

Mr. James W. Pan
Tek-Hand Corporation
8F-3, No. 30, Sec. 3
Chung Shan N. Road
Taipei, Taiwan, Republic of China
Phone: (886) 2-585-1369
Fax: (886) 2-585-0919

United Kingdom

Mr. Andy Smith
Artificial Intelligence International Ltd.
1 Park View Road
Berkhamsted, Herts HP4 3EY, United Kingdom
Phone: (44) 4-42-87-6722
Fax: (44) 4-42-87-7997

We'll be there...

ParcPlace will be represented at the following trade shows during the coming months. Come visit us.

ATRE, May 31-June 2, Sapporo, Japan. Adele Goldberg will be keynote speaker.

Object Expo, June 2-4, Dallas. Adele will be keynote speaker.

Object Expo - Europe, July 14-17, London. Adele will be keynote speaker.

Object World, July 20-23, San Francisco. □

ParcBench®

ParcBench, the ParcPlace Systems bulletin board service, is open to anyone interested in object-oriented technology.

Dial into ParcBench on the data line, (415) 691-6716. Beginning in June, 1992, a new number will be used—it was not available at press time but you can call the main number to get it. ParcBench accepts calls at 300, 1200, 2400, 4800 and 9600 bps, 8 bits, no parity, 1 stop bit. □

ParcPlace 
NEWSLETTER

The **ParcPlace Newsletter** is published three times a year by ParcPlace Systems, Inc. Copyright ©1992 by ParcPlace Systems, Inc. All rights reserved.

Publisher: Adele Goldberg

Managing Editor: Brian Alexander

Associate Editor: Denise Boucher

Contributing Staff: Roy Benjamin, John Bruzas, Laurie Cox, Richard Dellinger, Paul Greenblatt, Michael Khaw, Glenn Krasner, Linda Lavezzo, Bill Lyons, Michel Montandon, Douglas Pollack, Kenny Rubin, Linda Siener and Marty Yam.

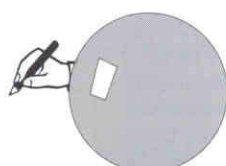
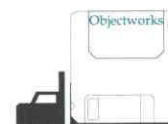
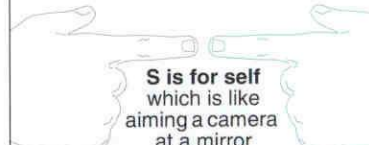
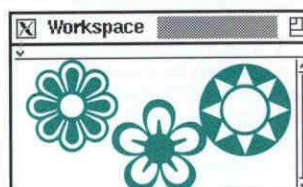
Send submissions to: Newsletter Editor, ParcPlace Systems, 999 E. Arques Ave., Sunnyvale, CA 94086-4593 — U.S.A. Or, send e-mail to newsletter@parcplace.com or ...luunet!parcplace!-newsletter. For subscription information, contact: Newsletter Subscriptions at ParcPlace.

Trademark notices: Objectsupport, ParcNotices, ParcRanger and Smalltalk-80 are trademarks, and ParcPlace, ParcBench and Objectworks are registered trademarks of ParcPlace Systems, Inc. Other product names may be the trademarks of their respective companies.

Printed on recycled paper.

ABC's for object-gifted children

by Brian Alexander (with thanks to Paul Greenblatt)

 input output A is for application which is the software equivalent of a Cuisinart™.	B is for Behaviorist (a behavior-oriented programmer) C is for Class which is like that little tool you use to blow soap bubbles.	 D is for Dependency which is how models think of everyone else.
E is for Encapsulation which is what aging caterpillars do. F is for Finalization which is one of the five stages of grief. G is for Garbage collection which is the sixth stage of grief.	 H is for Halt which is borrowed from B-grade WWII films and means roughly, "Stop and identify your stack trace!"	 I is for Inheritance which is how the upper class passes on its wealth.
J is for Jargon which is a class of messaging behavior that encapsulates its content so that nonmetaclass receivers cannot inherit its meaning. 	K is for Keyword which makes you wonder whether there is a matching Lockword somewhere.  L is for Library which is a warehouse for those little tools you use to blow soap bubbles. 	  M is for Method which is the function of a soap bubble.
N is for nil which is	O is for Object which is the granddaddy of all soap bubbles.	P is for Portability which is the software equivalent of court-ordered desegregation. 
Q is for Queue which is a bread line for hungry objects.	 R is for RGB which is <i>not</i> the color-mixing scheme you learned in art class.	
 S is for self which is like aiming a camera at a mirror	T is for Temporary variable which is like a disposable sandwich bag.	U is for Undefined which is Purgatory for soap bubbles. V is for VM which is like the Marine Corps (it does all the grunt work).
 W is for Workspace which is a virtual reality for behaviorists.	 X is for X which is how UNIXoids do windows.	Y is for YellowButton which is the middle kid in the mouse family.  Z is for Zombie which is a soap bubble stuck between the fifth and sixth stages of grief.

Brian Alexander is a senior technical writer at ParcPlace.

Brian Alexander is a senior technical writer at ParcPlace.