



SYMBOLICS GENERA SOFTWARE ENVIRONMENT

Symbolics' Genera™ 7.0 software environment, which includes Symbolics Common Lisp™, is a technological breakthrough in program development and applications delivery. Release 7.0 of Genera introduces new concepts in ease of use, application packaging and delivery, and enhanced programmer productivity. Genera maintains its position as the most productive environment for developing and delivering complex software solutions through uniformity in syntax, commands, and functions; through applications delivery features for constructing user interfaces and packaging applications; and through new tools for software development.

The Genera software environment was developed in parallel with Symbolics 3600™ family of advanced symbolic processors, making these systems the ideal hosts for Genera's suite of software features and utilities.

PRODUCTIVITY TOOLS FOR SOFTWARE DEVELOPMENT AND APPLICATIONS

SOFTWARE DEVELOPMENT

The Genera software environment has a proven track record for reliability, performance, and maintainability. These features combine to make Genera the highest-productivity software development system available today, and help companies easily manage large software projects.

PRODUCTIVE DEVELOPMENT TOOLS

Symbolics offers tools that contribute to creativity, orderly software development, and high programmer productivity: the attributes necessary in a profitable software engineering environment. Symbolics productivity tools encourage rapid prototyping, incremental software modification, easy maintenance, and overall project synchronization. With Genera, programmers can spend most of their time thinking about their programs and very little time being distracted by the details of the computer system, which increases programmer productivity and speeds time to market.

SYMBOLICS COMMON LISP

Symbolics Common Lisp is a complete implementation of the Common Lisp language, with extensions such as Flavors object-oriented programming and the window I/O system.

Conversion utilities and functions aid the transition from earlier versions of Zetalisp to Symbolics Common Lisp where incompatibilities occur.

With the unique Symbolics package hierarchy, software developers can stay with the standard features of Common Lisp for maximum portability to other Common Lisp environments, or use the Symbolics extensions for maximum programming flexibility and sophistication.

SMARTSTORE™ STORAGE MANAGEMENT FACILITIES

Always the performance leader, Symbolics has increased the speed and power of their systems even more. With Symbolics SmartStore storage management facilities, programmers can increase the efficiency and performance of their applications. SmartStore facilities that do this include:

Table Management System

Symbolics Table Management System provides a single programmer interface for all methods of storing and manipulating tabular data in virtual memory, and improves application performance.

Since optimal data representation depends on the number of tabular entries, the Table Management System dynamically changes the representation as the number of data entries varies. This eliminates the need to select a worst-case representation as a hard-coded part of application design. It also eliminates the need to reprogram when moving from prototype into full implementation.

In addition, Symbolics has increased the performance of hash tables substantially.

World Optimization Tool

The World Optimization Tool reorders data and functions in virtual memory to reduce paging operations so that fewer pages need to be accessed when running a program. This results in a dramatic speed-up of operations. Developers can use the World Optimization Tool to enhance the performance of many programs.

Debugger

The new Symbolics Debugger in Genera 7.0 is more robust and does more work than ever before. For example:

- Through mouse-driven debugging, programmers can set and clear breakpoints by pointing at the displayed code with the mouse.
- With Source Level Debugging, when an error occurs while executing a program written in any language, the debugger lets developers examine the source code where the error occurred. Programmers don't have to spend valuable time looking at assembly code to find the source of an error.

FILE PROTECTION

Genera 7.0 File Protection is implemented through the "access control list" paradigm. With File Protection, the system requires a user password before allowing access to directories of files. Genera File Protection precludes inexperienced users from accidentally accessing or deleting a file.

SYMBOLICS WHEELS: PRODUCTIVITY TOOLS FOR APPLICATIONS

Object-Oriented Programming: New Flavors

Symbolics Flavors has always been the industry-leader in object-oriented programming systems. The New Flavors system is more intuitive, more powerful, more reliable, and significantly faster than previous versions.

New Flavors features include improved ordering of flavor components, providing more understandable and consistent rules governing flavor combination. This makes it easier to design and implement complex applications correctly, and results in easier maintenance.

New Flavors also supports generic functions, providing the same programmer interface and argument checking as with standard functions. Genera continues to support message passing as an alternative to generic functions.

Other Flavors features include automatic updating of instances after flavor redefinition, and substantial performance increases in creating new objects (instances) and in compiling flavors.

Symbolics assists in the critical area of applications packaging and delivery by providing fast tools for completing the application and creating a professional product before distribution to end users. With other computer systems, completing the last 20% of the project can take 80% of the time.

Wheels applications delivery features in Genera include advanced software tools that make it easy to:

- Configure the application.
- Maintain all underlying programs.
- Design and develop the user interface of the application.
- Package the application for ultimate delivery to end users.

APPLICATION CONSTRUCTION TOOLS

Symbolics Application Construction Tools make it is easy to maintain several versions of an application—a requirement for delivery of the same application to varying end-user markets.

Using Symbolics Application Construction Tools, new applications features can be developed and tested individually, then incrementally added to the main application. Developers can stay in sync with each other and the inconvenience of incompatible patches is eliminated.

Uniform Syntax

For each language and subsystem, the Application Construction Tools use a uniform, streamlined syntax. Thus, applications written in Ada, FORTRAN, Pascal, Prolog, and/or Lisp are defined and configured consistently.

Incremental Source Modification

Each Symbolics world is a hierarchy of systems that are built on and interface with each other, but can still be modified incrementally. Each system and portion thereof can be patched, maintained, and compiled as a separate version, without recompiling the entire system.

Through incremental source modification, system improvements can be incrementally released for general use by the system developer. Developers can take advantage of the new features without stopping their work and recompiling code. Changes can be made to all versions or to specific versions that are targeted to particular end users.

USER INTERFACE FEATURES

The Symbolics user interface is highly interactive and responsive. Applications delivered on Symbolics systems retain this critical characteristic.

With Wheels, standard Symbolics *user interface features can be embedded directly into an application* without low-level programming. They make delivery of an application faster because designers are freed from detail work. Applications are more intuitive and understandable to end users because features are uniform and consistent throughout the system.

Applications designers needn't program the underlying support for user interface features because it is incorporated into the Wheels delivery substrate. The new user interface provides significantly easier ways for programmers to design and implement:

- User-input prompts.
- Error checking.
- Mouse-sensitive items.
- Windows.
- Menus.
- Graphics.
- Screen Layout.
- Scrolling.

SemantiCue™ Enhanced Interaction System

With Genera, the Symbolics system “understands” the context of the application and anticipates a user’s response.

For example, if an application prompts for a file name, the user can either type the name or use the mouse to select a name from current or previous screens. The system automatically and transparently validates the user’s entry. If the input is not appropriate to the prompt’s context (e.g., a Lisp object instead of a file name), the system will cue the user and even suggest a correct response.

Genera provides a uniform interface for all operations to accept input. Previously, “prompt-and-read” completion, pathname merging, “choose-variable-values,” and the Command Processor each had separate ways for accepting input. With the new input interface, even inexperienced programmers will find it easy to build context-specific interfaces into the application. Additionally, earlier forms can coexist with the new form for accepting input.

Showcase™ Display Facilities

It is very easy to extend an application to work with the variety printers and consoles that are supported by the Symbolics 3600 family. The underlying substrate of Genera already knows how to format output appropriately for a given device. An application that is developed on a large screen can be easily transported to a smaller screen without rewriting code, for example.

Showcase allows developers to concentrate on the application itself and significantly reduces the time-consuming work of porting it from the development environment to the final delivery environment.

With Showcase Display Facilities, applications developers can, at a very high level, arrange for output to be produced in columnar, tabular, and graphical forms. The system automatically adjusts the sizes, widths, and spacings of the columns, table entries, and graph nodes to accommodate both the data being produced by the application and the specific characteristics of the output device (screen or printer).

Scrolling

Any window on the screen can scroll in all directions—left, right, up, and down—for maximum flexibility. This built-in feature saves the programmer time and effort because screen management and scrolling are two of the most significant and difficult parts of applications development.

CAD/CAE applications, for example, make extensive use of scrolling and other screen management operations for graphic objects. On other computer systems, the programmer needs to understand the underlying principles of screen management, memory, and graphics. With Symbolics, the programmer can concentrate on the application—the underlying implementation is already provided in the Symbolics system.

Windows that Remember

Traditionally, if a user needs to remember the output of an operation but doesn’t store it in memory or jot it down, the output has to be regenerated by repeating the operation. With Genera’s new windowing features that

remember, users can simply scroll to the original output containing the needed information.

In many computer systems, windows support view-mode only. On Symbolics systems, when the user sees the file (as a file name or as a pictured object), the file can be instantly accessed by pointing to it with the mouse.

Screen output such as a pathname is known to be a pathname—not just a string or a bit pattern—and can be grabbed with the mouse and used as input to a program or command that requires a pathname. This saves time for the applications developer because the linking process is pre-packaged and easy to use.

FRAME-UP™ LAYOUT DESIGNER

Developers can interactively format the screen layout for the application, including many windows with advanced, unique features through Frame-Up software. The developer interactively places windows on the screen, then stretches and shapes them to accommodate the application.

In other computer systems, the developer is forced to define screen real estate and content at a much lower level in the code. With Frame-Up, the contents of each window are defined at a high level, during prototyping, and extended when finalizing the end-user application interface. On the Symbolics system, moving from prototype to implementation now takes a fraction of the time required on other systems.

All of the delivery features of the Symbolics offer important productivity advantages for the development of applications, too.

GENERA ENVIRONMENT FEATURES

The Genera environment supports multilanguage programming, an extensive network environment, and tools and utilities that enhance programmer and applications user productivity.

MULTILANGUAGE PROGRAMMING ENVIRONMENT

Symbolics offers more languages than any other symbolic processing computer company.

Symbolic processing languages include Symbolics Common Lisp, Zetalisp®, and Prolog. Additionally, the Genera environment supports traditional, industry-standard, high-level languages including FORTRAN-77, Pascal, and Ada®. Developers in these languages need not know Lisp to take advantage of the powerful features of the Genera environment.

In the Genera multilanguage environment, all software tools and development utilities are available to all languages. Routines and programs written in various languages can be combined into large application systems and can call each other freely.

Numerical computations are aided by standard hardware features using IEEE floating-point representation and by the optional Symbolics Floating Point Accelerator.

COMPLETE NETWORK ENVIRONMENT

The Genera environment supports a range of network services and protocols for communications among Symbolics systems as well as other vendors' hosts.

- Host-to-host networking services between Symbolics systems and the IBM®/SNA environments are possible through Symbolics SNA Facility.
- Ethernet industry-standard hardware, available for each workstation, supports local-area networking.
- Available protocols include SNA, DECnet™, TCP/IP (the DoD DARPA standard), and Chaosnet (standard software for communications among Symbolics computers).
- Generic network implementation is based on Symbolics Flavors object-oriented software.
- Multiple protocols are supported simultaneously over a single Ethernet cable.
- Uniform system commands can be used, regardless of the protocol selected.
- The Dialnet phone network has mail and remote login services.
- Global network name and address management are provided through a distributed network database and control system.
- User-defined network protocols are available through system-supplied functions.
- Within an application, the choice of a particular protocol to provide a particular network service is made by the network system automatically and transparently.

Other features of the Genera environment, that benefit both developer and end user, include:

- Sophisticated operating system environment written entirely in Lisp.
- Optimized environment designed to provide fast, interactive response.
- Full selection of features that facilitate rapid prototyping of complex problems.
- Fully integrated set of advanced development tools including editors, debuggers, compilers, and inspectors.
- Highly flexible user interface with a variety of window configurations and menu styles under programmer control.
- Extensive system-level functions both documented and modifiable by developers.
- Complete set of the Symbolics documentation available online through the award-winning Document Examiner™ as well as in printed form. Users can generate their own copies on Symbolics laser graphics or dot matrix printers.
- Concurrent execution of application code in multiple languages.
- All standard program development utilities are window-oriented, support fill-in-the-blank command completion, and can be keyboard and mouse activated.
- Compatibility with earlier operating system releases for most new features. Easy conversion facilities (software tools, documentation, and training) for incompatible features.

PRODUCTIVE APPLICATIONS DEVELOPMENT ENVIRONMENT

Applications are easier to develop, maintain, and support on a Symbolics system. Because the Genera environment is conducive to writing modular, concise, and powerful code, applications are also easier to extend. Software written in the Symbolics environment can include more end-user benefits in fewer lines of clearer code: the result is better-performing programs and programmers.

Symbolics provides the tools and high-productivity software environment necessary to complete a development project and deliver the final product. Extensive software development tools and delivery features of the Genera environment are not available from any other computer vendor. Additionally, Symbolics offers the same software features across our full product line. The Genera environment provides for:

- Rapid prototyping.
- Advanced editing and debugging.
- Version synchronization across several development groups.
- Application configuration.
- Performance enhancements.
- Fast product release to end users.

An Integrated Environment for Developing and Delivering Applications

The Symbolics 3600 family blends the versatile and powerful Genera software environment with processing power that can run the most complex applications. Symbolics high-end development systems, coupled with the low-end delivery systems for symbolic processing applications, provide for the most productive and powerful product line that exists in the industry today.

For more information about the Genera environment or our other symbolic processing products, contact:

Symbolics, Inc.
National Sales Administration
4 New England Tech Center
555 Virginia Road
Concord, MA 01742
617-259-3600

International Sales:

Symbolics, Ltd.
London, England
44-494-443711

Symbolics, GmbH
Frankfurt, West Germany
49-6196-47220

Copyright © 1986. All Rights Reserved. Symbolics, Inc. Symbolics, Symbolics 3600, Genera, Symbolics Common Lisp, Document Examiner, SmartStore, Frame-Up, Showcase, SemantiCue, and Wheels are trademarks of Symbolics, Inc. Ada is a registered trademark of the U.S. Department of Defense. DECnet is a trademark of Digital Equipment Corporation. IBM is a registered trademark of International Business Machines Corporation. Zetalisp is a trademark of Symbolics, Inc.

Symbolics believes that the information in this publication is accurate. Specifications are subject to change without notice. Symbolics is not responsible for any inadvertent errors.