

SoftBench 5.0: The Evolution of an Integrated Software Development Environment

The vision and objectives of the original SoftBench product have enabled it to continue to be a leader in the integrated software development market. For example, since SoftBench 1.0, over 80 third-party software tools have been integrated with SoftBench.

by **Deborah A. Lienhart**

HP SoftBench is an integrated software development environment designed to facilitate rapid, interactive program construction, test, and maintenance in a distributed computing environment. The SoftBench product contains an integration framework and a set of software development tools, as well as the ability to integrate tools from other sources.

SoftBench was released in 1989 and presented in the June 1990 HP Journal.¹ At that time, no one would have guessed the market changes that would occur during SoftBench's life. Fortunately, the vision and objectives of the original product designers have allowed SoftBench to continue to be a leader in the integrated software development market.

This article presents the actions that have made SoftBench a standard in the integrated software development market, the original SoftBench objectives that have stood the test of time, and the new technologies that have been incorporated into SoftBench. Other articles in this issue will present more information about the new technologies in SoftBench.

The different versions of SoftBench released since its introduction in 1989 are shown in Fig. 1.

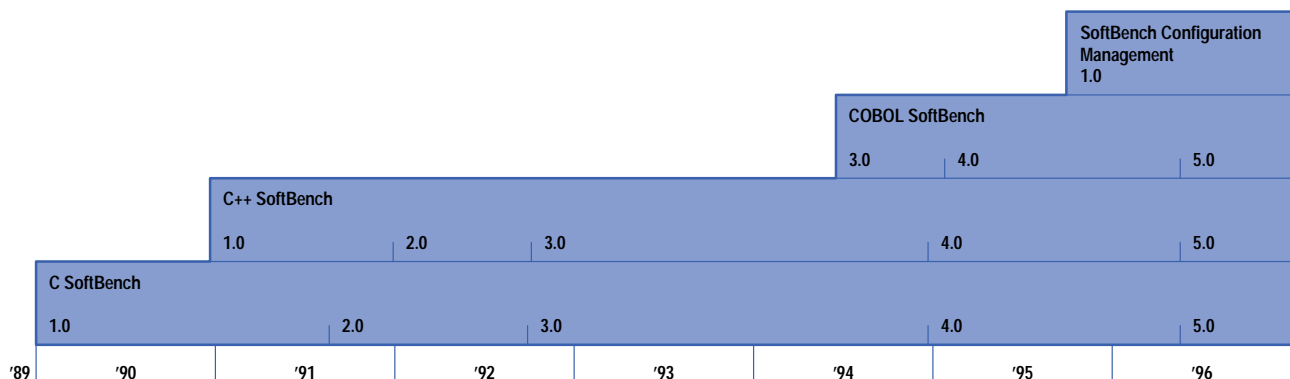


Fig. 1. The different versions of SoftBench released since 1989.

Making SoftBench the Standard

SoftBench defined the open, integrated CASE (computer-aided software engineering) market. The first big challenge was to make SoftBench pervasive in the market. We used several approaches, including leading standards development, working with software tool providers, and licensing the framework source code. HP started and supported CASE Communiqué, a standards body that focused on defining the messages used for intertool communication. This work was adopted as the basis of intertool communication standards for software development tools by the ANSI X3H6 Committee.

HP worked with software tool providers, both through CASE Communiqué and with independent software vendor (ISV) programs, to provide SoftBench integration for their tools. There have been over 80 third-party software tools integrated with SoftBench, and we continue to see interest from software tool vendors who want to integrate their tools with SoftBench.

The source licensing program was interesting to many companies for a number of reasons. Some companies ported SoftBench to their hardware, added some tools, and sold it to their customers. Several other companies have ported SoftBench to their own hardware for use by their internal development organizations. One company, SAIC (Science Applications International Corporation), contracts with customers to provide cross-development support for other, usually

non-UNIX,® platforms. This is used mainly for legacy system support or to develop software for platforms that can't support a native application development environment. This is the only part of the source licensing program that is still active.

Article 5 describes the activities of the Science Applications International Corporation.

The SoftBench broadcast message server (BMS) framework was adopted by other HP products and some customers' products, in addition to its use in SoftBench. The biggest user of the BMS framework is HP VUE (Visual User Environment). BMS provides the same open integration of desktop tools in HP VUE that it provides for software tools in SoftBench. For software developers, BMS also provides an integration of the desktop tools with software development tools.

Original Objectives of SoftBench

The SoftBench framework continues to provide the foundation for SoftBench and has stood the test of time. The following are the original objectives of the SoftBench architecture and the changes that have taken place.

Support Integrated Toolsets. This goal dictated that the SoftBench tools should cooperate to provide a task-oriented environment that lets users concentrate on what they want to do, not how to do it. SoftBench continues to provide a task-oriented environment by allowing tools to be started from each other. For example, in most SoftBench tools you can show the references for any symbol via the static analyzer.

Some users prefer a tool-focused environment, so SoftBench 5.0 has a new ToolBar to make it easier to see what tools are available in SoftBench (see Fig. 2).

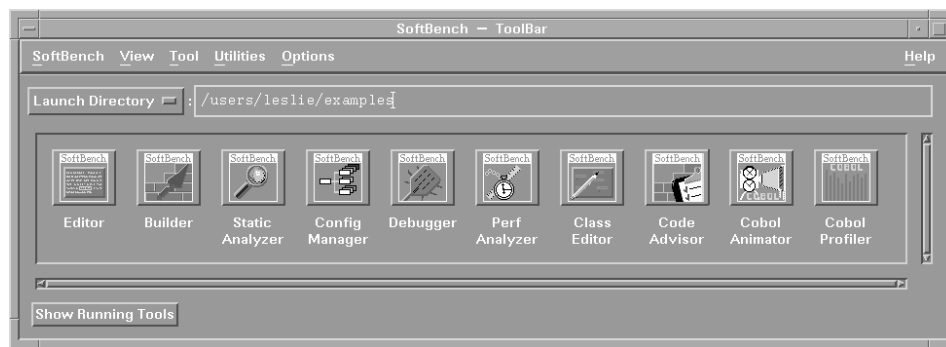


Fig. 2. A ToolBar screen.

Support Interchangeable Tools. The concept of plug and play, which allows users to exchange a SoftBench-supplied tool with one of their preference, has guided SoftBench's architecture and the development of standards in the CASE industry. Text editors and configuration management systems are the most common tools that users customize.

Support a Distributed Computing Environment. This goal required that all tool execution, data, and display should be designed for a network environment. This objective was based on the scenario of a software development team using a group of workstations with varying capabilities and shared project files on a central server. Providing distributed computing support in SoftBench has not only allowed it to work well in this scenario, but also has provided additional benefit in the ability to target computers and embedded systems that cannot support an application development environment. The biggest SoftBench customers make use of this capability.

Leverage Existing Tools. The reason for this objective was to protect customers' investments in software development tools by allowing these tools to fit into the SoftBench environment without modifying source code. This has worked well for lightweight integrations, but most customers have decided that the increased value of a deeper integration is worth adding a simple module to their source code.

Support Software Development Teams. Originally SoftBench included integration with RCS (Revision Control System) and SCCS (Source Code Control System) configuration management tools and support for accessing shared project files. In SoftBench 5.0, the SoftBench configuration management product SoftBench CM was added. SoftBench CM is based on the history management server, which has been used internally in HP for many years. SoftBench CM provides global source code management for software development teams whose members can be located anywhere around the world.

Support Multiple Work Styles. Software engineers do a number of different tasks during the course of a project, including design, prototyping, construction, defect fixing, and maintenance. Each of these tasks requires a different emphasis of the software development tools. For example, construction makes extensive use of the editor and builder, defect fixing is centered in the debugger, and maintenance starts with the static analyzer. Each of the tools is accessible from the others, which allows a task to have quick access to multiple tools or to transition between tasks.

Support Other Life Cycle Tools. SoftBench supports the integration of other tools that support the software life cycle, including documentation, test, defect tracking, and design tools. Most of the third-party tools integrated with SoftBench are in these categories.

Build on Standards. SoftBench has always been built on standards, such as the UNIX operating system, NFS and ARPA networking, the X Window System, and the OSF/Motif appearance and behavior. In SoftBench 5.0 we added integration with the Common Desktop Environment (CDE),² including CDE drag and drop.

New Technology in SoftBench

In the years since the first release of SoftBench, the breadth of tool support and functionality of the tools has increased significantly. This section briefly describes some of these additions.

Static Database. In SoftBench 3.0 a new object-oriented static database was placed under SoftBench's static analyzer. Earlier versions of the static analyzer could only analyze 30,000 to 40,000 lines of source code before reaching capacity limitations. The new static database does not have capacity limitations and performance is acceptable for up to about one million lines of source code.

In addition to the capacity and performance improvements, the object model of the new static database makes it more flexible for adding language types and queries. The SoftBench static analysis database is described in **Article 3**.

Rule Engine. In SoftBench 5.0 a rule engine was implemented as part of the SoftBench CodeAdvisor product. A rule is implemented as a C++ class, which can access information in the static database and any other information available to it. The rules are run by the rule engine, which is integrated into the SoftBench program builder.³

A set of C++ coding rules is included in SoftBench 5.0. These rules check for dangerous coding practices, which are the ones that would create memory leaks or have unanticipated side effects. Information and examples needed to create rules are included in the SoftBench software developer's kit.

The SoftBench CodeAdvisor is described in **Article 4**.

New Languages. The first release of SoftBench supported the C language. C++ SoftBench was added in 1991. C++ enhancements were made to the SoftBench tools and a C++-specific tool, the C++ Developer, was added. The C++ Developer was designed to be a training tool. It had a graphic display of the class inheritance hierarchy, and the user could add or delete classes and inheritance relationships from the graph. It could also automatically fix common coding problems before they were caught by the compiler. In SoftBench 5.0, the C++ Developer was replaced by the graphic editing functionality in the SoftBench static analyzer's class graph.

COBOL SoftBench³ was added to the product family in 1994. It provides encapsulations of most of the MicroFocus COBOL tools.* The SoftBench development environment makes it easier for users to transition to the UNIX operating system from mainframe development environments. COBOL SoftBench provides a common development environment for C, C++ , and COBOL. This is especially helpful when debugging an application that is a combination of COBOL and C or C++.

MicroFocus' Animator and SoftBench program debugger pass control of the application between themselves as the application moves between modules implemented in different languages.

SoftBench CM. The SoftBench configuration management product was introduced in 1995. It is based on the history management server, an internal tool that has been used for most of HP's software development.

SoftBench CM is a scalable configuration management tool that offers efficient code management capabilities for team members and work groups, including those who are geographically dispersed in distant locations. Based on a client/server architecture that is designed to allow access to multiple local or remote servers, SoftBench CM is easily accessed from any of the SoftBench tools (see Fig. 3).

SoftBench CM can manage different versions of any type of file. Many of our customers use SoftBench CM to version nonsoftware files, including project documents and bitmaps. A PC user interface has been developed that allows users in mixed UNIX and PC environments to create versions of their PC-based files along with their UNIX-based files.

Graph Views. In the original version of SoftBench there were only textual interfaces. In SoftBench 3.0, graphical interfaces were added to many of the SoftBench tools, including the dependency graph browser in the program builder, the static graph browser in the static analyzer, and the data graph browser in the program debugger.

In SoftBench 4.0 the underlying graph library was replaced by an implementation based on a third-party graphics library called ILOG Views. This implementation is much faster and will handle a lot more nodes than the old implementation. The static graph browser was replaced with three specialized graphs for files, functions, and classes.

In SoftBench 5.0, graphical editing capability was added to the SoftBench static analyzer's class graph and its name was changed to the class editor. **Article 2** describes the C++ SoftBench class editor.

* The COBOL SoftBench family is based on HP MF COBOL, HP's implementation of MicroFocus COBOL, which is based on technology from MicroFocus, Ltd.

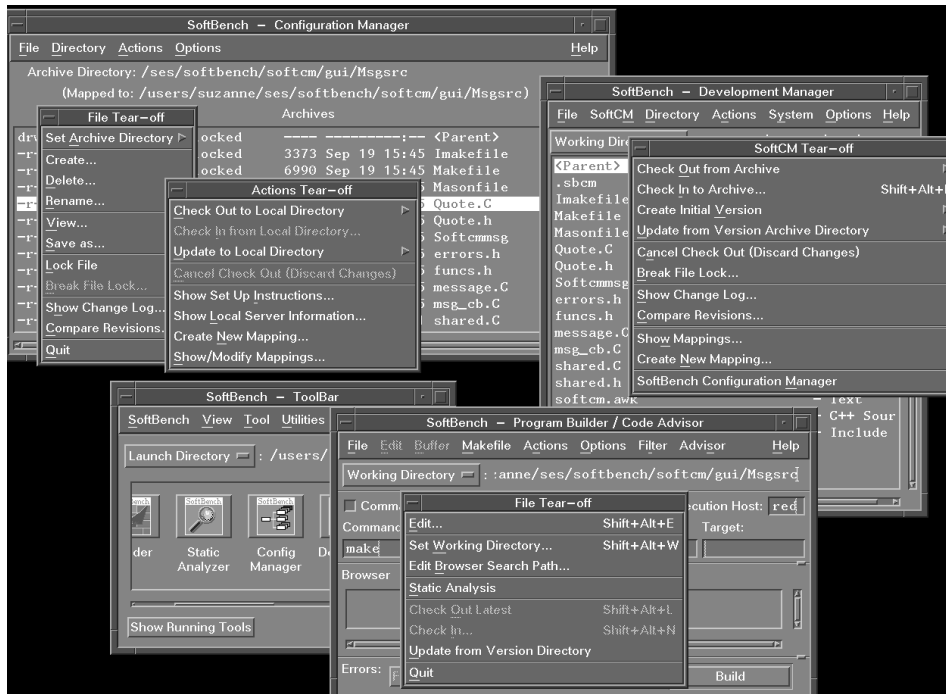


Fig. 3. A SoftBench CM screen.

More Platforms. SoftBench originally supported HP 9000 Series 300 workstations and HP 9000 Series 800 file servers. Support was added for the HP 9000 Series 400 and Series 700 workstations and HP 9000 Series 800 servers with X terminals. In 1991 SoftBench was released for SunOS and in 1993 support was added for Sun's Solaris operating system.

DDE Debugger. In SoftBench 4.0, the underlying debugger for SoftBench's program debugger was changed from xdb to the HP Distributed Debugging Environment (DDE) from HP's Massachusetts Language Laboratory. HP DDE's architecture isolates most of the debugger from specific information about the target operating system, computer language, user interface, and debug format. The SoftBench team implemented the SoftBench program debugger user interface on top of HP DDE and ported the whole thing to the Solaris operating system. This is the only development environment that supports both the HP-UX* and Solaris operating systems with a common debugger, using the compilers supplied by the system vendors.

ToolBar. SoftBench is a powerful development environment and as the user base has expanded we've placed more emphasis on making it easier to learn and use. Many times users requested tools that were already in SoftBench so we added an iconic ToolBar to make the available tools visible. The ToolBar supports drag and drop integration with HP VUE and CDE.

Conclusion

When SoftBench was first envisioned, UNIX software development tools consisted of compilers and debuggers, and real software engineers didn't use windows. SoftBench was the first integrated application development environment running on the UNIX operating system.

There wasn't much to work with then, just RFA (remote file access) and TCP/IP networking and the beginnings of the X Window System. Motif came along during the development of the first release of SoftBench and NFS came along later. When HP's Software Engineering Systems Division (SESD) developed the BMS (broadcast message server) for interprocess communication and it was included in HP VUE, it changed the capability of desktops for everyone.

Over the years SESD has developed new technology for the challenges brought on by the C++ language and larger applications. We also added a lot of graphics as the technology became available and workstation performance increased.

In the future, SoftBench will face new challenges associated with developing distributed applications that run in heterogeneous environments. We can look to the original objectives and architecture for a path that has stood the test of time.

References

1. *Hewlett-Packard Journal*, Vol. 41, no. 3, June 1990, pp. 6-68.
2. *Hewlett-Packard Journal*, Vol. 47, no. 2, April 1996, pp. 6-65.

3. C. Carmichael, "COBOL SoftBench: An Open Integrated CASE Environment," *Hewlett-Packard Journal*, Vol. 46, no. 3, June 1995, pp. 82-85.
4. A. Iyengar, T. Grzesik, V. Ho-Gibson, T. Hoover, and J. Vasta, "An Event-Based, Retargetable Debugger," *Hewlett-Packard Journal*, Vol. 45, no. 3, June 1995, pp. 33-43.

OSF and Motif are trademarks of the Open Software Foundation in the U.S.A. and other countries.

HP-UX 9.* and 10.0 for HP 9000 Series 700 and 800 computers are X/Open Company UNIX 93 branded products.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Limited.

X/Open is a registered trademark and the X device is a trademark of X/Open Company Limited in the UK and other countries.

-
- ▶ [Go to Subarticle 1A](#)
 - ▶ [Go to Next Article](#)
 - ▶ [Go to Journal Home Page](#)