

### Introduction

The Developers' Toolkit for HP Tru64 UNIX® provides a set of key application development tools for the Tru64 UNIX Operating System. It can be found in the Tru64 UNIX CDROM set.

The Developers' Toolkit for Tru64 UNIX is a prerequisite for most Tru64 UNIX development tools, languages, and environments. The Developers' Toolkit contains the following components:

- Compaq C for Tru64 UNIX, an ANSI conformant C Compiler
  - Debuggers (Ladebug, dbx)
  - The ATOM API
  - Program analysis tools (ATOM-based) for profiling and performance analysis
  - Visual Threads (thread-related profiling and debugging)
  - Porting Assistant
  - Post-link optimizers (spike, cord, om)
-

### The C Compiler

The Developers' Toolkit includes the Compaq C for Tru64 UNIX compiler, a standards compliant, multi-dialect, full-featured, and highly-optimizing implementation of the C language specifically developed to exploit the 64-bit Alpha architecture. It is invoked by the `cc` command.

The compiler provides options to compile the source code in conformance with standards including "ANSI C" (ANSI X3.159-1989[1992], or equivalently ISO/IEC 9899:1990[1994]), and conforms to the XPG4-UNIX standard with the `c89` command. It also provides IEEE floating point support conformant with IEEE Standard 754 (equivalent to IEC 60559:1989).

The compiler supports these popular C dialects:

- Relaxed ANSI mode: which compiles according to the ANSI C standard but allows some common programming practices disallowed by the standard. This is the default.
- Strict ANSI mode: which strictly enforces the ANSI C standard (with a separate control for the namespace used in standard headers).
- Common mode: which supports the K&R programming style, with certain ANSI extensions where the K&R behavior is undefined or ambiguous.
- Microsoft® compatibility mode: which interprets source programs according to certain language rules followed by the C compiler provided by Microsoft with Visual C++®.
- VAX C compatibility mode: which compiles according to the ANSI C standard, but also supports a number of VAX C extensions that are incompatible with that standard and that change the language semantics.

The compiler's diagnostic facility is highly customizable with `#pragmas` and command-line options. Compiler options are available to generate listing files with source, symbol table, and machine code, and to generate complete debug and traceback information in object modules for use with the Ladebug and dbx debuggers.

The compiler provides the following data types:

- 8, 16, 32, and 64-bit signed and unsigned integer types
- IEEE 32-bit float, 64-bit double, and 128-bit long double floating-point types
- 64-bit pointer type (support for 32-bit pointers is also provided to facilitate porting and reduce memory consumption)

It contains extended support for systems programming, parallel programming, and mathematical computing with features including:

- Support for the OpenMP parallel programming API
  - Fine-grained control over optimization behavior with `#pragmas` and command-line options
  - User-defined assembly language sequences using `#pragmas` and intrinsic functions that expand and optimize these instruction sequences inline
  - `#pragmas` and command-line options for controlling data layout
  - IEEE floating-point (with optional handling of exceptional values like NaN, INF, and denormals)
  - Built-in functions to generate efficient code sequences for processor interlocked memory access
  - Shared-library support
  - Program profiling in conjunction with `pixie`, `prof`, `gprof` and `hiprof`
  - Run-time feedback optimization
  - Inter-language calls with other Tru64 UNIX languages
  - Compaq C is also available on the OpenVMS and Linux Alpha operating systems.
-

### Ladebug

Ladebug is a symbolic source level debugger that helps programmers locate run-time programming errors (bugs) in their code. Ladebug provides extensive debugging support for the following languages: C, C++, FORTRAN-77, and FORTRAN-90. Ladebug provides limited support for Compaq COBOL, and DEC Ada.

Both a standard and an international implementation of Ladebug ship with this version of the Developers' Toolkit. They are two distinct kits, with the standard implementation being the default. See the product documentation for details.

#### Ladebug Features

- Debug simple programs
  - Load/unload one or more programs into the debugger
  - Attach/detach to a running process and debug a program
  - Display a source listing of a program
  - Track the program counter location relative to the program's sources
  - Set breakpoints to stop program execution when specified sections of code are executed
  - Set watchpoints to stop program execution when a specified area of memory or specified variable is read or written
  - Add conditions to breakpoints and watchpoints so that program execution will only stop at the specified break or watch event when the condition is true
  - Step into a routine
  - Step through the execution of a program one line at a time
  - Examine and navigate through the call stack
  - Examine and change variable values
  - Examine the contents of memory in various formats
  - Catch/ignore signals and unaligned access
  - Create a backup "snapshot" of your debugging session in case you accidentally step over a section of program code that was of interest to you
- Use command line editing (simple emacs binding)
- Use Ladebug within emacs
- Customize debugging environment by using scripts and aliases for commands and command sequences
- Disassemble and examine machine code and examine machine register values
- Debug core files
- Debug programs with shared libraries
- Debug multiple programs or processes
  - Control one or more running processes
  - Debug one or more programs in one debugging session
  - Debug across fork/exec
- Debug multithreaded applications
- Debug the kernel

Ladebug provides a choice of command-line or graphical user interface. The command-line interface has dbx-like command syntax.

The Ladebug GUI supports all Ladebug command-line functionality. The GUI main window includes the basic debugging and convenience features; optional windows, various pop-up menus and dialog boxes, and a command-entry prompt.

#### Key features of the Ladebug GUI

- Logical and easy access to the most common debugging functions
- Convenient call stack maneuvering
- Convenient display of program execution through the source viewer display
- Easy to use interface for setting complex breakpoints
- Source navigation through the source browser
- An I/O window for debugger output and command input

Over time Ladebug will replace the dbx debugger as the default debugger.

---

### The dbx Debugger

The dbx debugger is a portable, symbolic source code debugger for customers who desire a debugger with a consistent, industry-standard, command-line user interface.

#### dbx Features

- Provide support for the C, FORTRAN, and Pascal languages
- Enable debugging of multithreaded applications via kernel threads, does NOT support POSIX threads/threads.
- Attach/detach to a running process and debug a program (same or multiple executables)
- Enable local and remote kernel debugging
- Display a source listing of a program
- Set breakpoints and tracepoints to stop program execution when certain conditions are met (for example, a program variable value changes)
- Step through the execution of program one line at a time
- Examine the stack of currently active functions
- Examine and change program variable and data structure values
- Disassemble and examine machine code and examine machine register values
- Debug programs with shared libraries
- Customize the debugging environment by using scripts and aliases for commands and command sequences
- Directly examine memory in various formats
- Display values in different scopes
- Display variables
- Debug across fork/exec

---

### Program Analysis Tools

The ATOM API can be used to create simple or sophisticated tools. ATOM uses the target application program, an instrumentation file and an analysis file to create a new executable that collects analysis data for a wide variety of purposes.

Tools are created using two sets of C routines, an instrumentation file and an analysis file. In the instrumentation file, the ATOM API is used to specify where calls to analysis routines should be inserted in an application. A single call to an ATOM routine creates a new executable. The analysis routines are called, at run time, by instructions ATOM inserted in the new executable. These routines run in the background of the application and analyze it for attributes like quality and performance. While ATOM greatly expedites tool creation, there is vast flexibility in designing the scope of the tool. HP has created several tools with ATOM, including hiprof, pixie and Third Degree.

- Hiprof provides procedure call information on the program that is similar to but more powerful than the gprof (cc -pg) profiler. It shows the amount of CPU time utilized by each primary procedure and the subordinate procedures they call.
- Pixie profiles instruction execution counts. The compiler, spike, cord, and om optimizers can use this data to achieve better code performance. Pixie profiles also support test-coverage analysis.
- The Third Degree tool profiles heap-memory for programs written in C and C++. It helps you identify three memory bugs: memory leaks, reading uninitialized memory and accessing an invalid memory address.

All these tools support shared libraries and threaded programs and are licensed with the Developers' Toolkit.

---

### Graphical Program Analysis Tools

The Graphical Program Analysis Tools (GPA) help application developers examine the run-time behavior of their applications on Tru64 UNIX.

The Graphical Program Analysis Tools can locate problems in your code, explain the problems using diagnostic messages, and even suggest necessary changes through reports. For example, using these tools you can:

- Instrument an application, run the application, and look at the memory usage data in one step or separate steps
- Find poorly tested areas in your code
- Locate and correct performance bottlenecks
- Find and fix memory leaks and memory errors
- Find and fix problems with writing past the ends of memory blocks
- Get information about all processes, including child processes, running on local or remote systems
- Invoke utilities and commands to monitor and manage the remote system
- Record all the process information to a file for later review

The Graphical Program Analysis Tools consist of the following:

- **Process Viewer:** Graphically displays performance information about processes and child processes running on specified (local or remote) Tru64 UNIX systems. (Also known as pview)
- **Memory Profiler:** Graphically displays how your application uses memory over the course of its execution, for example, showing how much is allocated for various purposes and how frequently your application is allocating and deallocating memory. This tool helps you understand if an application uses memory inefficiently, for example, fragmented memory allocations. (Also known as mview)
- **Performance Profiler:** Gathers, analyzes and displays, in graphical form, run-time statistics about your application, such as CPU usage by function or line and test coverage. (Also known as dxprof)
- **Heap Analyzer:** Finds and displays in graphical form memory errors and memory leaks in your application. (Also known as dxheap)
- **Man Page Browser:** used for searching, navigating, and printing manpages in a graphical, scrollable hypertext window.

In addition, online help is provided for each tool in a hypertext window.

You can use the Process Viewer, Memory Profiler, Performance Profiler, and Heap Analyzer standalone or, if you want to use them from your Windows NT® PC, as part of the HP Enterprise Toolkit - UNIX Edition for Tru64 UNIX (SPD 70.03.xx).

The Graphical Program Analysis Tools require a typical Alpha workstation or server for doing software development, using Common Desktop Environment (CDE) or Motif® interface environment software.

You can run tools directly from the CDE desktop interface or invoke a tool directly from the command line (for example, pview, mview, dxprof, or dxheap).

The Graphical Program Analysis Tools provide full language support for Compaq C++, Compaq C, and Compaq Fortran (both f77 and Fortran 90).

---

### Visual Threads

Visual Threads is the diagnostic tool you need to analyze and refine your multithreaded applications. You can use it to debug potential thread-related logic problems, even those hard-to-find problems that only occur due to slight timing differences. You can also use Visual Threads to pinpoint bottlenecks and performance problems, by using its rule-based analysis and statistics capabilities and visualization techniques.

Visual Threads can be used with any Tru64 UNIX application that uses POSIX threads (the POSIX, CMA or DCE API) or is written in Java. It is designed for multithreaded applications of all sizes – applications with anywhere from two threads to hundreds of threads. It can be used to identify potential problem areas, even if an application does not show any specific problem symptoms. Use it for prevention as well as correction!

Visual Threads includes the following powerful features:

- Detailed information about significant thread-related state changes ("events")
- Automatic analysis of common threading problems based on predefined rules applied to the event stream
- Rule customization for application-specific parameters and actions
- Automatic statistics gathering, by sampling the event stream
- Categories of analysis: data protection errors (race conditions), deadlocks, programming errors, lock activity and performance
- Graphical visualization of the frequency of thread related events and thread state, snapshots of historical program state, and object-specific graphs for each collected statistic
- Performance analysis including lock contention and lock granularity, lock profiling, statistical hotspots, processor utilization, detection of shared cache lines, and analysis of data access locality (on AlphaServer GS Series systems).

### The Porting Assistant

The Porting Assistant is an integrated graphical toolset designed to reduce the time and cost of porting applications to Tru64 UNIX. It helps when porting applications written in C, C++, and Fortran from other UNIX platforms such as Sun™ OS [TM], HP-UX, IBM® AIX and ULTRIX, and from non-UNIX platforms such as OpenVMS. The Compaq Porting Assistant is a Tru64 UNIX toolset based on Motif.

The Porting Assistant guides the developer through the porting process by suggesting a systematic, iterative porting approach. First, the Porting Assistant analyzes source files and locates changes that are potentially needed for the application to run on Tru64 UNIX. Then, through extensive, graphical hyperlinked information, it helps programmers understand what changes are needed and why they need to make them. Finally, it aids in making each change either by using the integrated editor or through its global replace capability.

In analyzing software code, the Porting Assistant locates:

- 32-bit dependencies
- Conditional code that might also need Tru64 UNIX branching
- References to files that do not exist on Tru64 UNIX
- Calls to library functions that do not match Tru64 UNIX definitions or have different semantics on Tru64 UNIX
- Platform-dependent signal handling code
- Function/subroutine calls in Fortran that are inconsistent with their definitions
- Makefile actions that do not exist on Tru64 UNIX
- Important command options and arguments that differ between platforms

To help developers understand each porting problem, the Porting Assistant graphical hyperlinked information:

- Provides porting tips that complement HP's extensive porting guides
- Explains the relevance of each of the analysis steps
- Explains the semantic differences in functions on different vendor's platforms
- Provides detailed help on many individual diagnostics
- Provides hyperlinked access to relevant reference pages

When making code changes, the Porting Assistant:

- Provides developers with a choice of vi, emacs or a Motif® editor
- Steps developers through the problem areas one at a time, bringing each into the editor to be examined and changed
- Provides a global search/replace facility to make changes across files
- Filters out unwanted diagnostics
- Suggests what sharable libraries and archives to use
- Reruns analysis at users request after changes are made
- Provides support for mapping Fortran compile command lines in a makefile to the equivalent Tru64 UNIX command line
- Provides the addition of a keyword lookup function for the DXML Library, which includes commonly used Fortran math functions
- Provides support for mapping LINPACK and EISPACK routines to LAPACK routines

The Porting Assistant requires a minimum of 64 MB of memory and 10 MB of disk space.

The Porting Assistant supports CDE and has its own icon in the Developers' Toolkit group.

---

### Post-Link Optimizers

The post-link optimizers optimize a whole linked executable or, in some cases, a shared library. Optimizations include rearranging procedures to facilitate better cache mapping, reducing the instruction cache miss rates. The tools are as follows:

- **Spike** is a new tool for performing code optimization after linking. Such tools are called binary or post-link optimizers. Because it can operate on an entire program, Spike is able to do optimizations that cannot be done by the compiler.

Some of the optimizations that Spike performs are code layout, deleting unreachable code, and optimization of address computations. Spike is most effective when it uses profile information to guide optimization.

Spike can process binaries linked on Tru64 UNIX Version 4.0F or later systems. Binaries that are linked on Version 5.1A or later systems contain information that allows Spike to do additional optimization.

Spike can be invoked as part of the compilation process, by specifying the `-spike` option with the `cc` command. In addition, Spike can be invoked on the command line by applying the `spike` command to a binary file after compilation.

Spike delivers equal or better performance than `om`. Spike will replace `om` and `cord` in a future release.

- **cord** - Rearranges procedures in an executable to facilitate better cache mapping
  - **om** - A post-link optimizer that can optimize executables and reorder them
- 

### Hardware Requirements

Please refer to the applicable Tru64 UNIX Operating System Version 5.1B QuickSpecs for specific hardware supported.

---

### Software Requirements

The Developers' Toolkit requires Tru64 UNIX Operating System Version 5.1B.

---

### Growth Considerations

The minimum hardware and software requirements for any future version of this product might be different from the requirements for the current version.

---

### Distribution Media

This product is distributed on CDROM and is also available via download.

---

### Ordering Information

The software license and the binaries for the Developers' Toolkit on Tru64 UNIX can be obtained separately.

Traditional Software Licenses: QL-MT5A\*-AA

Concurrent Use Software License: QL-MT5AM-3B

Software Media/Documentation: Binaries for the Developers' Toolkit may be obtained from one of the following:

- For Tru64 UNIX V5.1A, kits for Visual Threads, Porting Assistant, and Graphical Program Analyzers can be found on the Associated Products CD-ROMs. The other components are included in the base operating system kit (order number QA-MT4AA-H8).
- Future updates can will be posted at: <http://www.tru64unix.compaq.com/dtk>

Software Documentation (hardcopy): QA-MT5AE-GZ

Software Product Services: QT-MT5A\*-\*\*

\* Denotes variant fields. For additional information on available licenses, services, and media, refer to the appropriate price book.

---

### Software Licensing

This software is furnished only under a license and is available on a traditional or Concurrent Use basis. The Developer's Toolkit license provides the right to use the software as described in this QuickSpec, and is furnished under the licensing of Compaq Computer Corporation's Standard Terms and Conditions. Tru64 UNIX Version 5.1B has been designated a minor release therefore Licenses (traditional, or Concurrent User) for prior versions of the Developers' Toolkit must be updated to Version 5.1B either through the purchase of a Service Agreement that includes the rights-to-use new versions, or through the purchase of Update Licenses. For more information about Compaq's licensing terms and conditions, contact your local HP agent.

---

### Software Product Services

A variety of service options are available from HP. For more information, contact your local HP office.

---

### Software Warranty

This software is provided by HP with a 90-day conformance warranty in accordance with the Compaq warranty terms applicable to the license purchase.

---

© Copyright 2003 Hewlett-Packard Development Company, L.P.

The information contained herein is subject to change without notice.

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained.