

Unique Id: 009DDF20-D0004880-1C02A1

Copyright 1999 Compaq Computer Corporation. All rights reserved

SOURCE: Compaq Computer Corporation INFORMATION BLITZ

INFORMATION BLITZ TITLE:

New Developers' Toolkit Supplement for Tru64 UNIX V4.0D,
V4.0E, and V4.0F

DATE: 10-SEP-1999

INFORMATION BLITZ #: TD 2697-CR

AUTHOR: John Parks

TEL#: (603)884-1124

EMAIL: John.Parks@compaq.com

DEPARTMENT: Tru64 UNIX Compiler Technology Group, Enterprise Computing

=====

PRODUCT NAME(S) IMPACTED: Tru64 UNIX V4.0D, V4.0E & V4.0F

PRODUCT FAMILY(IES):

PRODUCT NUMBERS:

Storage _____
Systems X
Networks _____
PC _____
Software _____
Other (specify) _____

ANNOUNCEMENT:

The Developers' Toolkit Supplement V4.0F kit delivers state-of-the-art software development tools on Tru64 UNIX V4.0D, V4.0E, and V4.0F. The tools include

Compaq C compiler
Atom-based Program Analysis tools
Graphical Program Analysis tools
Visual Threads
Ladebug debugger

and deliver better performance and brand-new features to these platforms.

The kit is a fully-supported, free upgrade for all users who have an active Developers Toolkit license (which all users doing software development should have).

The kit is available to download from:

<http://www.unix.digital.com/dtk>

The kit contains the indicated versions of the following components:

1) Compaq C (V6.1-110)

The Compaq C compiler, formerly DEC C, is a standards-compliant, multidialect, and feature rich implementation of the C language. It contains a highly optimizing code generator specifically designed to exploit the 64-bit Alpha architecture. It is particularly well-suited and contains extended support for systems programming, parallel programming, and mathematical computing.

This new version of Compaq C provides:

- * a proven and reliable compiler that incorporates many important bug fixes from previous versions.
- * an improved code generator/optimizer that exploits the latest Alpha hardware technology.
- * a significantly improved diagnostic facility: better messages, more messages (including uninitialized variable detection, run-time array bounds checking, and unused header file detection), and improved command-line and #pragma diagnostic controls (including lint-style message groups).
- * a full implementation of the OpenMP Parallel Programming API for shared memory multi-processing.
- * new features from C9X (the proposed revision to the ANSI/ISO C Standard), including restricted pointers, variable length arrays, interspersed declarations and statements, and initializers with designators.

2) Atom-based Program Analysis Tools (V1.0)

The Program Analysis Tools help profile and debug applications through shell commands. The tools in this version provide new commands that speed learning and use. They offer new defaults and options, improved support for pthread applications, and new reference pages and tutorials. The tools include:

- * Hiprof: A call-graph profiler that collects information on the cpu-time that a program uses. It instruments the program, executes it, and displays the profile using an improved gprof. A new profiling style provides an enhanced alternative to traditional `cc -pg` profiling, allowing shared libraries to be profiled and per-source-line cpu-time to be displayed.
- * Pixie: An instrumenting instruction-count profiler, primarily for use in profile-directed optimization (with the `cc` command's `-feedback`, `-cord`, and `-om` options) and test coverage

analysis. This version of pixie supports recent compiler releases and provides per-thread profiles. A new prof command allows the display of per-source-line profiles for the cc -p and uprofile profilers.

- * Third Degree: A memory-access checker. It instruments a program, executes it, and displays a log of memory-related errors that it detected. By default, the third command unintrusively checks for memory leaks and malloc/free errors. Optionally, it can also report accessing invalid addresses and reading uninitialized heap and stack. The new -g option allows the instrumented program to be debugged while running checks.
- * Atom: A framework for writing new program analysis tools. Atom provides a high level API to its instrumentation services, so that a new tool's instrumentation and run-time analysis code can be implemented in a few lines of C. The API is compatible with the prerequisite Tru64 UNIX header files, and the new -g and -w options help debug and productize tools.

3) Graphical Program Analysis Tools (V2.0-7)

The Graphical Program Analysis Tools make it easy to analyze the run-time behavior of a program for efficiency, memory bugs, and use of shared resources. The tools include:

- * Process Viewer: Graphically displays performance information about processes and child processes running on specified (local or remote) Tru64 UNIX systems.
- * 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.
- * Performance Profiler: Gathers and analyzes, in graphical form, run-time statistics on your application, such as CPU usage by function or line and test coverage.
- * Heap Analyzer: Finds and displays in graphical form memory errors and memory leaks in your application.
- * Man Page Browser: Used for searching, navigating, and printing manpages in a graphical, scrollable hypertext window.

4) Visual Threads (V1.1-015)

Visual Threads lets you analyze and debug multi-threaded applications. It can be used to:

- * automatically diagnose common problems such as deadlock, protection of shared data, and thread usage errors.
- * monitor the thread-related performance of the application,

helping you identify bottlenecks or locking granularity problems.

It is a unique debugging tool because it can be used to identify problem areas even if an application does not show any specific problem symptoms. Visual Threads provides the following:

- * **Event Collection:** Collects detailed information about the significant thread state changes that occur during the running of a multithreaded application.
- * **Event Display:** At any time, you can view the dynamic display of events as they occur, with options to filter for specific types of events.
- * **Automatic Rule-based Analysis:** Detects violation conditions by evaluating the events against a set of enabled rules. Several predefined rules look for data protection errors, deadlock conditions, and programming errors.
- * **Rule Customization:** Lets you use templates to define your own rules to specify criteria for violation conditions. In particular, you can specify rules to monitor or identify performance problems.
- * **Customized Actions:** Suspends execution of the application when it detects violation conditions. You can choose from several options at this point, including invoking the debugger in the appropriate context for the application.
- * **Playback:** Records events to a trace file so you can play back and analyze them later. This is especially useful if you want to analyze a trace file iteratively, perhaps using different rules each time as you discover more information.
- * **Statistics:** Provides easy access to object-level statistics and current state information, including use of resources (for example, the mutexes held by a particular thread). You can use this information to look for performance problems and to fine tune the application.
- * **Thread and Event Visualization:** Graphically and continuously displays the state of the program threads and the frequency of thread related events.

Visual Threads can be used with any Tru64 UNIX application that uses POSIX threads (DECthreads) or is written in Java. It is designed for multi-threaded applications of all sizes - applications with anywhere from two threads to hundreds of threads.

New features for this version of Visual Threads include many user interface enhancements, and:

- * The ability to specify application environment variables from within the Visual Threads GUI.
- * The command vttrace can be used to create a trace file outside

the Visual Threads GUI.

- * A new statistic for monitoring cumulative locked and wait time on locks.
- * Control panel additions: scrolling capability, lines or tickmarks to show the number of CPUs (and compare to active threads), and multiple colors to distinguish blocked threads.

5) Ladebug (V4.0-54)

The Ladebug debugger is a powerful fully symbolic debugger that helps you locate run-time errors (bugs) in your code. Key features allow you to:

- * Perform source-level debugging
- * Attach to a running process
- * Debug programs with shared libraries
- * Debug multithreaded applications
- * Debug multiprocess applications, including programs that fork/exec
- * Perform remote client/server debugging
- * Perform kernel debugging
- * Provide internationalization support for local language characters

Ladebug supports the debugging of programs written in C, C++, Fortran, Fortran 90, Ada and COBOL. Ladebug features expression evaluation using the syntax of the source programming language.

```
*****< NOTE >*****
*
* INFORMATION IN THIS DOCUMENT REPRESENTS OPERATIONAL EXPERIENCES AND
* SUGGESTIONS BY COMPAQ OR PARTNER EMPLOYEES. COMPAQ SHALL NOT BE
* RESPONSIBLE FOR ANY ERRORS OR OMISSIONS CONTAINED IN THIS DOCUMENT,
* AND RESERVES THE RIGHT TO MAKE CHANGES TO IT WITHOUT NOTICE.
*
```

```
<>UPDATE /TEXT_UPDATE/UNIQUE_IDENTIFIER="009DDF20-D0004880-1C02A1"-
/TITLE="[TD 2697-CR] New Dev Toolkit Suppl for Tru64 UNIX V4.0D-V4.0F - BLITZ"-
/BADGE=(AUTHOR="999997",ENTER="120172",MODIFY="120172",-
EDITORIAL_REVIEW="120172",TECHNICAL_REVIEW="999997")-
/NAME=(AUTHOR="JOHN PARKS",ENTER="CATHY MARTIN",-
MODIFY="CATHY MARTIN",EDITORIAL_REVIEW="CATHY MARTIN",TECHNICAL_REVIEW="JOHN PARKS")
/DATE=(AUTHOR="10-SEP-1999",ENTER="10-SEP-1999",-
```

```
EXPIRE="10-SEP-2001",FLASH="16-SEP-1999 09:47:56.98",MODIFY="16-SEP-1999",-
EDITORIAL_REVIEW="10-SEP-1999",TECHNICAL_REVIEW="10-SEP-1999")-
/GEOGRAPHY="USA"/SITE="EIRS"/OWNER="TIM-BLITZ"-
/FLAGS=(USA_CUSTOMER_READABLE,NOPOST_MESSAGE_DISPLAY,NOLOCAL,-
EUR_CUSTOMER_READABLE,GIA_CUSTOMER_READABLE,NOINIT_MESSAGE_DISPLAY,-
EDITORIAL_REVIEWED,FIELD_READABLE,FLASH,TECHNICAL_REVIEWED,READY)
```