

elj.com → What's New → Jul 2001

elj.com



Extraordinarily Large Jumble

(Eiffel Liberty)



A Web Site dedicated to exposing an eclectic mix of elegant technologies

[Eiffel Quick Links: FAQ | CLE | Forum | GNU Eiffel | Hal | ISE | OT | EE, VSA | elj-win32 | SWIG-E]
 [Eiffel Docs: dmoz:Eiffel | Invite | ETR | Kernel | Syntax | Style | OOSWE in Eiffel, OOPiE | EAI | C++??]
 [Quick Links: Cetus | LW | LWN | L2D * | scripting.com | SlashDot * | Haskell | Perl | Python | Sather | SWIG]
 [Linux: Debian WN | GNOME | KDE:news | Linux-Mag | LinuxStart]

What's New (03 - 18 Jul 2001) at elj.com

What's New Archive for: Jul 01 Jun 01 May 01 Apr 01 Mar 01 Feb 01 Jan 01
 Dec 00 Nov 00 Oct 00 Sep 00 Aug 00 Jul 00 Jun 00 May 00 Apr 00 Mar 00 Feb 00 Jan 00
 Dec 99 Nov 99 Oct 99 Sep 99 Aug 99 Jul 99 Jun 99 May 99 Apr 99 Mar 99 Feb 99 Jan 99

For 18 months I've been concerned that the weight of a massive user community such as Windows would push Tel from the round hole* it was designed to fit into a square hole.
 London, 03 Dec 98 <http://elj.com/elj-daily.cgi>

The Random Eiffel Quotes Banner Generator

[C++ Report] Eiffel for Native Speakers of C++

<http://www.elj.com/new/0107/>

12/17/2001

profiling support, etc.. To use sedb (SmallEiffel DeBugger), just add the -trace flag in your traditional command line of compilation or activate the trace mode in your ACE file.

- * Added a traditional -help flag for all commands to print a summary of available options. In ACE files as well as in "loadpath.se" files the \${SmallEiffelDirectory} environment variable is automatically defined using the value of the \${SmallEiffel} environment variable.
- * Full compatibility with the new ELKS'2001 STRING class. Most modifications are simple renamings which are automatically pointed out by obsolete warnings. For obvious uniformity reasons, some other classes of the library have also been modified (ARRAY, LINKED_LIST, DICTIONARY, etc.). As an example, nb_occurrences of ARRAY is now obsolete and the obsolete warning tells you that occurrences is the new name to be used. This work is still a work in progress. Please, contact Arno Wagner (in charge of the STRING class modification) on our mailing list (SmallEiffel@l...).
- * The new TUPLE type is now implemented.
- * Added class TIME_IN_GERMAN to handle date and time in German. See lib_show/basic_time for examples.
- * Feature is_equal of class COMPARABLE is now ELKS compatible.
- * Added feature internal_key in class DICTIONARY.
- * Various bug fixes and improvements.

VAULT: a programming language for reliable systems
 (elj-ID 764) (Created Jul 17, 2001, 22:00 UTC) (By Rob DeLine and Manuel A. Fahndrich)

<http://www.elj.com/new/0107/>

12/17/2001

18 Jul 2001

[elj-submit]: EiffelStudio

(elj-ID 763) (Created Jul 17, 2001, 10:13 UTC) (By Richard Mouli) (Seen at elj-submit)
 (Submit by Richard Mouli)

I hope you are already aware :-) The official release of EiffelStudio is announced at www.eiffel.com. The useful page is here

Final release -0.75 is now available

(elj-ID 763) (Created Jul 17, 2001, 21:54 UTC) (By Dominique Colnet) (Seen at SmallEiffel Mailing List Archive) (Submit by Geoff)

I am happy to announce the final release -0.75 of SmallEiffel the GNU Eiffel Compiler.

Please, visit and download SmallEiffel from our site:

SmallEiffel.loria.fr

What's new since the previous release:

Release - 0.75 - July 16th, 2001

- * Added ACE file support (ACE stands for Assembly of Classes in Eiffel). The most important advantage is that assertion level checking can now be selected differently for each class. This is also true for the runtime trace mechanism (-trace). In ACE file mode you have to provide a special *.ace file to drive the compiler. See lib_show/ace directory for examples. Note that the traditional command line mode will continue to be supported.
- * There is now a powerful Eiffel level debugger with on-line help documentation. This debugger features step by step execution, navigation into the run-time stack, multi-conditions breakpoints, garbage collector invocation,

<http://www.elj.com/new/0107/>

12/17/2001

(Seen at comp.lang.eiffel) (Submit by Geoff)

Programming language researchers have come up with many great ideas since the birth of the C programming language, over thirty years ago. Despite these advances, C still remains one of the most popular languages for software development. To put many of these advances into the hands of C programmers, we're creating the Vault programming language.

Besides C's syntax, Vault inherits C's focus on imperative programming, its bit-level control over data layout, and its simple run-time system (no garbage collector). In addition, Vault provides the following improvements over C:

- * Vault forces the programmer to think about error conditions, which means that no implicit type errors can occur at runtime. Every operation that may potentially fail takes on a form in Vault that requires the programmer to deal with the possible failure. For example, Vault distinguishes between possibly-NULL and non-NULL pointers. Dereferencing a possibly-NULL pointer is simply not possible. Instead the pointer must be tested, yielding in one branch a non-NULL pointer that can be dereferenced as long as the pointer is not changed, and in the other branch a NULL pointer. We apply the same philosophy to all data and operations that have possible failure modes: Operations are only allowed when the compiler can guarantee that they won't fail. By programming in Vault, a developer can discover at compile time errors that C programmers typically find only through testing (if at all), such as NULL pointer dereferences or dangling pointers.
- * Vault enforces programmer specified rules through a mechanism based on tracking resource keys. A resource

<http://www.elj.com/new/0107/>

12/17/2001