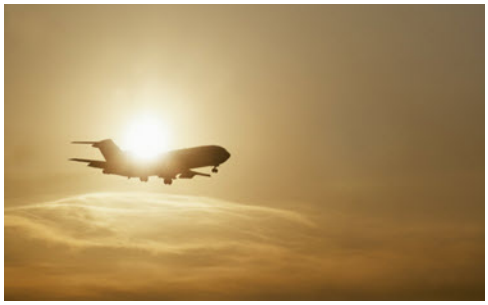
ESCHER TECHNOLOGIES

eCv
ESCHER C VERIFIER



```
size_t bsearch
(   const float* array table,
    size_ nItems,
    float key
)
pre(table.isndec)
pre(nItems == table.lim)
pre(nitems >= 2)
post(result in 0..(nItems - 2))
post(table[result] <= key)
post(key < table[result + 1])
{   size_t low = 0;
```

Escher C Verifier

Now you can write provably-correct software in C!

Proving that software meets its functional specifications has traditionally required specialist computer languages and skills. The Escher C Verifier brings provable correctness within reach of many more software developers - by combining precise specification, advanced automated reasoning, and the most popular programming language used for developing embedded software.

What is the Escher C Verifier?

eCv is a tool that empowers you to develop critical embedded software in C together with proofs of its correctness, robustness and security.

Unlike other formal tools, *eCv* delivers high productivity by generating a very high proportion of software verification proofs without user intervention, using state-of-the-art automated reasoning technology.

Easy to introduce into your process

eCv extends the C language with additional keywords and constructs needed to express specifications and strengthen the type system of C. These annotations are invisible to ordinary C compilers, so you can compile your annotated program as normal.

If you want to apply *eCv* to existing software, you don't need to verify it all at once. With *eCv* you can annotate and verify individual C source files, if you provide minimal specifications for any external functions they call.

Use the C language safely

eCv uses a carefully-chosen verifiable subset of the C language that avoids classic vulnerabilities and strengthens the type system. If you're already coding to the MISRA-C 2004 standard, you'll find that your programs are well on the way to being compatible with *eCv*.

What does eCv prove?

eCv always tries to prove that your program is free from out-of-bounds array indexing, null pointer de-referencing, arithmetic overflow and other "undefined behaviour", and that each loop in your program will terminate. If you write annotations to express functional specifications and required safety properties, *eCv* will attempt to prove they are satisfied too.

Easy to learn and use

The automated reasoning technology of *eCv* avoids the need for user involvement in constructing proofs. Additionally, if *eCv* fails to find a proof, it will often suggest the missing precondition or invariant that makes proof possible. These features make *eCv* easier to learn and use than traditional formal methods.

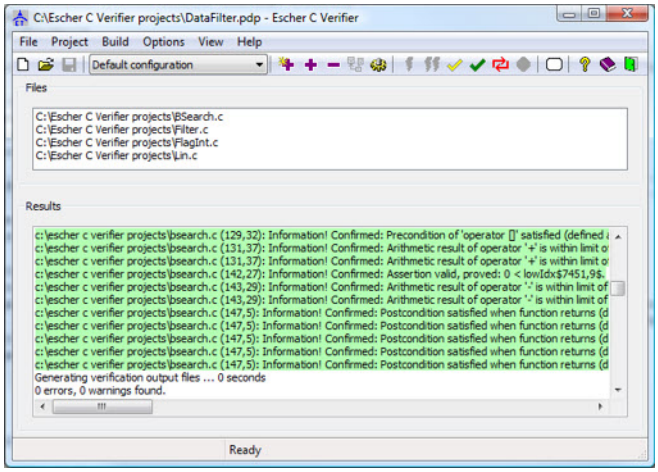
Built with mature technology

eCv uses the same Verified Design-by-Contract paradigm and powerful automated reasoning engine as *Perfect Developer* - the formal modelling tool used industrially to model and verify diverse applications, including SIL 4 defence software and business application logic.

Find out more today!

To discuss how *eCv* can help you develop more reliable C software in less time, email **critical@eschertech.com** or telephone us on +44(0)20 8144 3265.

Escher C Verifier in action



What others say about us

"Our need is to meet the requirements of defence standard 00-55 to Safety Integrity Level 4. Escher Technologies software met our requirements best."

"We were especially impressed by the automation of verification proofs, which will substantially reduce our costs, and by the level of support provided by Escher Technologies."

Guy Mason, Senior Software Engineer at General Dynamics UK Ltd.

"We have used Perfect Developer for about four years and we have received excellent support from Escher Technologies throughout."

"The ability of the theorem prover to identify problems in specifications is extremely valuable and leads directly to high-quality code. We were impressed by the code generator."

John Warren, Precision Design Technology Ltd.

Technical Specifications

Development platform requirements

PC with fast x86 or x64 processor and 2Gb or more main memory.

Windows XP, Vista or 7 operating system, 32- or 64-bit (contact us if you require a Linux edition).

Text editor (syntax configuration files including *eCv* keywords are supplied for several popular editors).

C'90, C'99 or C++ compiler with provision for running the preprocessor separately

Supported source code languages

Verifiable subset of C'90, including most constructs permitted by MISRA-C 2004. Optional support for some C'99 and some C++'98 constructs.

Output

Verification summary displayed in user interface or saved to file. Analysis of unproven verification condition and associated suggestions saved to file in a choice of formats. Successful proofs of verification conditions can be saved to file in a choice of formats (HTML, LaTeX, or plain text).

About Escher Technologies

Escher Technologies was founded in 1995 to research and develop leading-edge software development technology.

Our mission is to reduce the cost of developing dependable software, so that reliability can be the norm rather than the exception, even for non-critical software.

Although our team has a strong commercial background, we maintain close links with the automated reasoning and formal methods research communities in leading universities worldwide.

For more information visit **http://www.eschertech.com** or email **critical@eschertech.com**

eCv
ESCHER C VERIFIER

ESCHER
TECHNOLOGIES

September 2010. Specifications subject to change without notice. MISRA is a trademark of MIRA Limited.
Escher Technologies Limited, Mallard House, Hillside Road, Ash Vale, Aldershot GU12 5BJ, United Kingdom.
Telephone: +44 (0)20 8144 3265 Registered in England no. 3128474