

TASKING™

VX-toolset for C166/ST10



The TASKING VX-toolset for C166/ST10 consists of:

- ISO C++ compiler, scalable to EC++
- C compiler, ISO C'99 compliant, with integrated MISRA C and CERT C static code analyzer
- Assembler/Linker/Locator
- C/C++ libraries, run-time libraries, floating point libraries
- Integrated Development Environment (IDE) based on Eclipse™
- Debugger integrated into Eclipse, with four execution environments:
 - Simulator
 - ROM Monitor debugger, using serial IO
 - On-Chip (OCDS) debugger, using JTAG



Support for Infineon Technologies

Altium is Infineon's premier partner for embedded development tools and is the only vendor offering support for Infineon's complete range of 8-bit to 32-bit microcontrollers, DSPs and 32-bit chip-cards. Whether you are a beginner or a professional, a small business or a large organization, we provide the tools that fit your XC800, C166, XC166, XC2000, XE166 or TriCore-based development projects.

Altium's TASKING C166 compiler is the de facto development solution for C166 projects from automotive Tier-1 and OEM vendors, as well as for wireless telecom applications, with a market share of more than 90% in these segments.

Altium's next-generation TASKING C166/ST10 toolset

Altium's latest embedded software development tools for the C166/ST10 microcontroller family offers substantial performance gains over previous development tools.

Based on Altium's sophisticated Viper compiler technology, this suite of tools is released as the TASKING VX-toolset for C166/ST10. The VX-toolset takes a major step forward by offering unparalleled code optimization performance, a totally new debugger, which is ready for the latest industry trends such as multi-core application development, plus toolset integration into the popular Eclipse™ platform as an Integrated Development Environment (IDE).

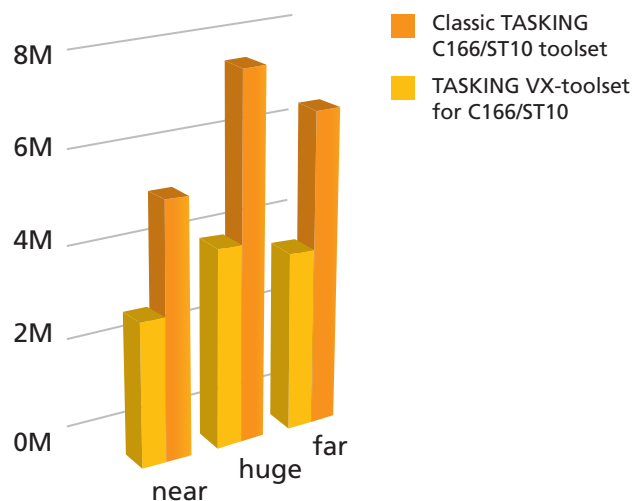
Eclipse integrates the compiler, assembler and linker seamlessly into a single IDE and comes with wizard functionality to set up the developer's application and configuration for target debugging. The new debugger is integrated into Eclipse through a plug-in. In this way the user has all development tools available within one state-of-the-art and industry-standard IDE, with the option to plug-in additional modules from third parties.

Smaller, faster code

Applications short on memory or run-time performance will highly benefit from the new advanced Viper-based compiler. Internal and independent external benchmarking on a variety of applications, as well as dedicated benchmark suites, indicate a code size decrease of around 45% on average when compared to traditional C166 compilers.

Similarly, impressive results are evident when applications are optimized for speed, where the average gain is around 24%. As both improvement figures are the result of compiling large collections of code, developers could expect even higher performance gains in real-world situations. For example, a representative automotive benchmark performed at one of Altium's customers showed a code speed improvement of 46% on actual hardware.

Altium's new TASKING toolset does support all popular Infineon Technologies C166/XC166/XC2000/XE166 and STMicroelectronics ST10/Super10 derivatives as before, but it is also ready for new family members. Devices with



Code size comparison – smaller is better.

Target architecture support

The TASKING VX-toolset supports nearly all C166 and ST10 derivatives. From within the Eclipse IDE you can easily select the device of your choice for your project:

- Infineon Technologies: C166, XC166, XC2000, XE166, E-Gold, as well as all of their individual variants
- IPextreme: C166S V1 core
- Micronas: SDA6000
- STMicroelectronics: ST10x16x, ST10x17x, ST10x25x, ST10x26x, ST10x27x, ST10x28x, Super10M34x, as well as all of their individual variants

Altium's active relationships with these semiconductor vendors enable us to support new derivatives already in the toolset prior to their availability in volume. Please consult our support engineers for the latest status of support for new microcontrollers or cores.

Support for new derivatives can in general be quickly added through a new processor definition XML file.



Support for STMicroelectronics

Altium and STMicroelectronics have been partners since the first ST10 microcontrollers became available. The TASKING development tools for the ST10 are used extensively in many automotive and hard disk drive applications. This relationship has culminated in Altium being the first to support STMicroelectronics' range of Super10 microcontrollers.

Industry standards support and conformance

The various parts of the VX-toolset conform to or provide support for relevant applicable standards, such as ISO/IEC 9899:1999(E), ANSI-C X3.159-1989, ISO/IEC 14882:1998(E), MISRA-C:2004, MISRA-C:1998, IEEE-754, ELF/DWARF 3, S-Record, CERT C and ORTI.

version 6.0. Altium has built the integration blocks for the toolset and extensions to Eclipse to make the whole environment a coherent workbench.

C compiler

Based upon Altium's latest DSP-C compiler technologies, the VX-toolset C compiler is reliable, compliant, competitive, complete, easy to use and generates the most optimal code possible to allow you to take full advantage of the C166 architecture.

The TASKING VX-compiler for C166/ST10 is tested for ISO C'99 and ISO C++ conformity against authoritative validation suites, such as Perennial and Plum Hall. In addition, the optimization techniques of the compilers are tested with various large real-world applications (for example, audio/GSM codec suites), as well as industry benchmark standards such as Nullstone and EEMBC.

Fast and compact

Altium understands that you expect your C166 compiler to produce the most optimal code possible with no fuss. With its Viper compiler technology, the TASKING VX-toolset for C166/ST10, in its default configuration, generates code with the smallest footprint and fastest execution possible. Depending on the specific requirements of your C166/ST10 application, optimizations can then be further tweaked for smaller code size or higher execution speed.

Compiler optimizations include:

- Application wide automatic *near* allocation
- Partial Redundancy Elimination (PRE) detects and eliminates repeating (sub-) expressions
- Various Loop and Jump optimizations speed up execution and reduce code size
- Control-flow and code-reduction optimizations remove dead code and perform transformations to minimize jumps
- Function inlining replaces calls to small functions with inlined copies of the function code
- Peephole optimizations replace instruction sequences with equivalent but faster and/or shorter sequences, or remove obsolete instructions
- Inter-procedural register allocation
- Application wide code compaction, often referred to as reverse inlining

Microcontrollers equipped with a Multiply-Accumulate Coprocessor (MAC) can highly benefit from automatic code generation of MAC specific instructions from native C code. This unique option to C166 compilers offers the user the most comfortable and efficient option to develop high performance code. Additionally, the MAC unit can be programmed through inline assembly code, or by using intrinsic functions that allow the compiler to use registers efficiently.

CPU functional problem support

Semiconductor vendors regularly publish microcontroller errata sheets reporting deviations from the electrical and timing specifications. As an integral part of best practice architecture support, Altium's TASKING VX-toolset for C166

and ST10 provides bypasses and checks for identified silicon defects. CPU functional problem support is provided throughout the complete toolset:

- C-compiler bypasses adapt code generation in order to avoid the identified erratic instruction sequences
- Assembler checks warn the assembly programmer for suspicious or erroneous instruction sequences
- Protected C-library sets are built with bypasses for all identified CPU functional problems

If reliability of your embedded application is essential, be sure to put support for CPU functional problems on your list of compiler selection criteria. Through its close co-operation with semiconductor vendors, Altium offers the most comprehensive support for this with its TASKING compilers.

Code profiling

In addition to the profiling features built into the debugger, the compiler is equipped with a profiler that uses code instrumentation. Code profiling can be used to determine which pieces of your code execute slower than expected and which functions contribute most to the overall execution time of a program. A profile can also tell you which functions are called more or less often than expected. The advantage of this code profiling option in the compiler is that it can give a complete call graph of the application annotated with the time spent in each function and basic block.

Several forms of profiling output can be obtained:

- Flat profile shows how much time is spent in each function, how many times that function has been called, and optionally how often each lexical block within the function is executed. This is very useful if you want to know which functions or lexical blocks consume most cycles
- Call graph profile shows, for each function, which functions called it, which other functions it called, and how many times. There is also an estimate of how much time was spent in the subroutines of each function

Syntax and semantic checks

The compiler offers a vast array of syntax and semantic checks that warn about potential undesirable effects or bugs in your program. Early fixing of source code problems when reported by the compiler generally only takes minutes compared to hours, or days, when the problem is discovered at run time.

Examples of compile-time checks include:

- Validating printf and scanf format strings against the type of the actual arguments
- Using uninitialized memory locations
- Detecting unused variables
- Value tracking, which is used to detect errors such as
 - array subscript out of bounds
 - division by zero
 - constant conditions

Unique TASKING VX-toolset features, not offered by other C166 compiler solutions

- ◆ MAC unit programming in C code
- ◆ Automatic evaluation of the most efficient code according to the developer's settings, using C166 instructions or MAC instructions
- ◆ Ultra efficient code generation, upto 45% more densed and 25% faster
- ◆ Application wide automatic near allocation for faster code, without the need to expliciteley use memory qualifiers
- ◆ Eclipse based IDE with integrated debugger
- ◆ Static Code Analysis for conformance to CERT C secure coding standard
- ◆ Add support for new processors through dynamic definition in XML configuration file
- ◆ Parallel build support for fast compilations on multi-core systems
- ◆ Import wizards for Classic C166 and Keil C166 compiler projects
- ◆ Dedicated support for the iCache of the new XC2000 future devices

Global type checking

The compiler and linker support global type checking. When using this option, the compiler will emit debug information necessary to perform type checking at link time. The linker will report any type mismatches on global/extern pairs.

Run-time error checking

The TASKING VX-toolset's run-time error checking capabilities in the compiler offer a wealth of checks that reveal run-time errors when they first occur. The kind of errors found by run-time error checking are typically hard to find since they manifest themselves through secondary effects or, in the worst case, will not manifest at all prior to your product being shipped. By identifying the source line where the error first occurs, the run-time error checking facilities reduce the time spent in the debugger, and increase the quality of your software. You can specify whether the application will terminate or continue when an error is detected.

These optional checks are implemented by generating additional code and/or enabling additional code in the standard C library. Run-time error checking has a nominal effect on code size and execution speed and can be enabled on a module-by-module basis, making it practical for use in debugging large applications.

The following types of checks are provided:

- Bounds checking verifies all pointer operations to detect buffer overflows and other illegal operations such as
 - accessing uninitialized or null pointers
 - accessing objects outside their declared bounds
 - illegal pointer arithmetic
- Malloc / free checks uncover dynamic memory allocation errors such as
 - buffer overflow
 - write to freed memory
 - multiple calls to free
 - passing an invalid pointer to free
- Report an unhandled case value in a switch without a default part
- Stack overflow detects when the stack grows beyond its allocated size
- Divide by zero issues a message when a division by zero is attempted

Static Code Analysis

Static code analysis is a method to verify all possible paths within a software program without actually executing the program. A static code analysis tool can efficiently locate defects such as out of bound array accesses, memory allocation errors, arithmetic over and under flows, and inconsistent code fragments that go unnoticed during dynamic tests or peer reviews. Static code analysis can be applied early in the software development process, it can be applied on incomplete and incorrect code bases and when no test-cases have to be developed.

Altium has integrated static code analysis functionality for CERT C and MISRA C in its C compilers, with the

advantage that such an analyzer is aware about specific embedded software issues such as: the existence of special function registers, the use of in-line assembly language, C-language extensions such as pointer and memory space qualifiers to address multiple address spaces, and DSP specific data types such as circular buffers, and fixed point data types.

CERT C

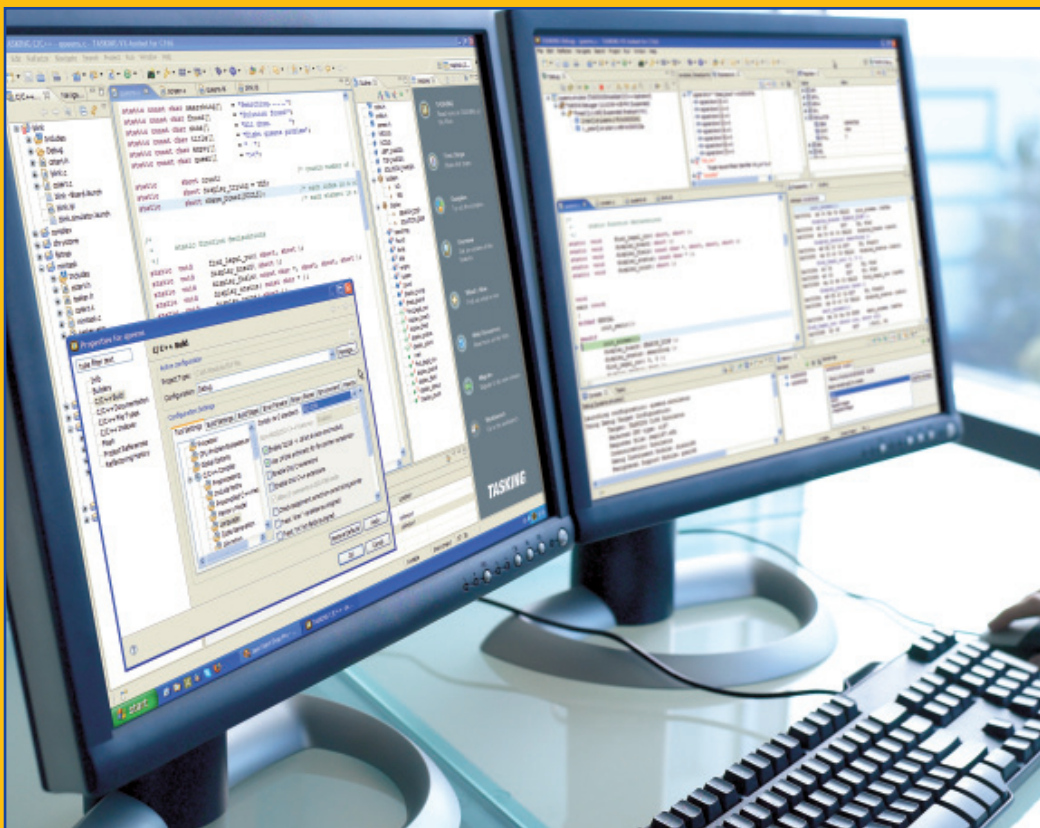
The CERT C/C++ secure coding standard is defined by the Computer Emergency Readiness Team (CERT), founded by the US government. Altium is one of the first vendors to provide a CERT C coding guidelines analyzer built into a C compiler for embedded software development. This feature is available in the Premium Edition.

MISRA C

MISRA C is driven by the Motor Industry Software Reliability Association and guides programmers in writing more robust C-code by defining selectable C-usage restriction rules. Through a system of strict error checking, the use of error-prone C-constructs can be prevented. The TASKING C compiler supports the MISRA-C:1998 and MISRA-C:2004 guidelines.

C++/EC++ compiler

Fully aware of the undeniable trend towards higher level language programming, the TASKING VX-toolset for C166/ST10 offers the full range of C++, C and assembly programming languages. Its ISO C++ compliant compiler allows developers to utilize the power of object-oriented design and coding techniques for the C166 family. The



Full control over editing and debugging your code.

object-oriented benefits of C++ can be incorporated into your C166 application one module at a time, providing appropriate use of assembly, C and C++.

Fully compatible with the Embedded C++ (EC++) standard, the VX-toolset's C++ compiler can be configured to selectively disable C++ features that may not be essential for your embedded application. By selecting full or partial compliance with the EC++ standard, code-size overhead and run-time inefficiency can be minimized.

Essential tools

The VX-toolset includes a macro assembler, a linker/locator, C/C++/run-time/floating-point libraries with source code, a librarian for creating and maintaining user libraries, object file converter and many example projects or reference designs. These projects and reference designs help you to get started with the VX-toolset quickly.

Through a versatile linker-script language the behavior of the linker/locator can be controlled for allocating, positioning and aligning of code and data. For tool interoperability, it supports the latest ELF/DWARF 3 standard and debugger specialists like Hitex, Lauterbach and PLS offer compatibility to the VX-toolset with their respective products for debugging your embedded application.

A new generation *make* with support for faster parallel builds on multi-core platforms is also included.

Flash memory programming

Through the Eclipse IDE and the debugger you can download an application file to flash memory. It allows versatile configuration of external devices as well as microcontroller on-chip flash memory, and the debugger controls the actual flash programming algorithm through a small programming monitor.

Integrated debugger

The VX-toolset's debugger is based on Altium's latest debugger technologies released in 2006. The debugger has been redesigned from the ground up and made ready for market trends like integrated kernel-awareness and multi-core debugging. Utilizing the Eclipse IDE workbench, it comes as a plug-in with a seamless integration to the editing environment. With the VX-toolset for C166/ST10 it provides three execution environments serving various debugging needs:

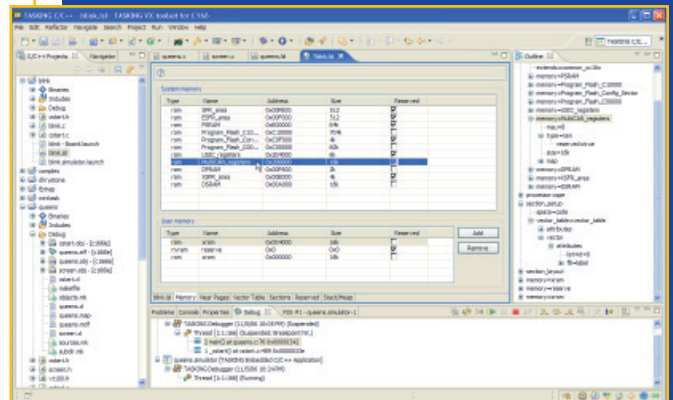
- C166 instruction set simulator debugging
- RAM/ROM monitor debugging
- On-Chip (OCDS) Debugging

Additional to the integrated debugger in the Eclipse IDE, a stand-alone debugger program is provided. This is not a complete GUI-based debugger, but instead it enables automated testing through scripts created by the developer.

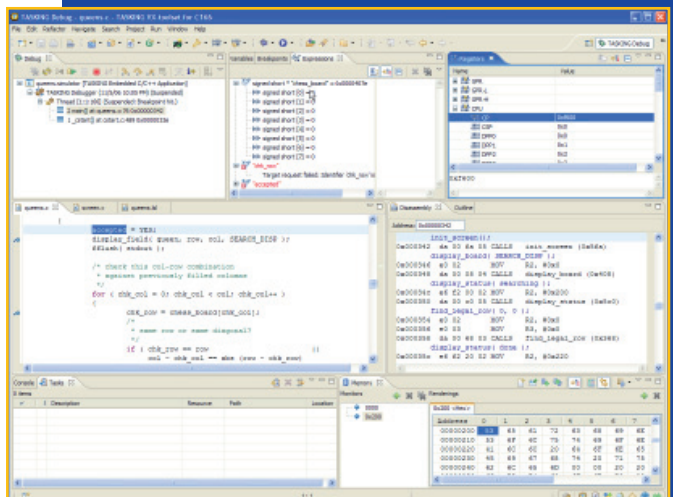
Toolset compatibility

The TASKING VX-toolset for C166/ST10 uses Altium's Viper compiler technology, which has a proven track record for generating stable, reliable, extremely compact and high-performance code. Because of its technical resemblance with the VX-toolsets for TriCore and ARM, the new VX-toolset for the C166/ST10 provides an excellent migration path to for example future 32-bit microcontroller based applications on Infineon's TriCore or STMicroelectronics' Cortex or ARM 7/9 series.

When comparing Altium's TASKING VX-toolset for C166/ST10 with other C166 tools, the major differentiator is that with the TASKING VX-toolset, the user is provided with a future roadmap into complete FPGA embedded systems design and a guarantee of full software compatibility with Altium Designer. This is truly a unique position for the TASKING VX-toolset and provides you with the confidence that your decision future proofs you by providing you with a tool that meets today's needs, plus an entry point into Altium Designer and all the benefits that a unified electronic product development system offers for the future.



Controlling the allocation, positioning and aligning of code and data.



Eclipse debugger perspective.

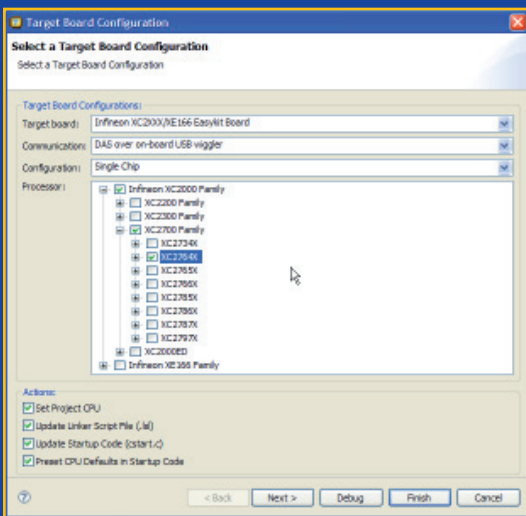


Evaluation board and custom board support

The ROM monitor and OCDS debugger in the TASKING VX-toolset support a wide range of standard evaluation boards and starter kits for rapid and simplified verification of your embedded project on hardware. Standard boards are supported from respected vendors such as:

- Infineon Technologies
- Phytex
- STMicroelectronics
- TKtronic
- TQ-Components

A helpful wizard is provided in the Eclipse IDE to set up the debug configuration for your custom hardware.



Target board configuration wizard

Depending on the debugger environment of your choice, you can connect to hardware through a serial cable, a USB cable or USB-JTAG miniWiggler.



DAP miniWiggler, to connect your PC through USB to the JTAG interface of your board

C166 instruction set simulator debugging

The C166 simulator debugger features instruction set simulation and simulation of a wide range of the microcontroller's on-chip peripherals, allowing you to extensively debug your application on the host platform, even before your target hardware is available. The supported peripherals include general purpose timers, interrupt system, peripheral event controller, parallel ports, SSC/ASC serial ports, phased locked loop, watchdog timer, real-time clock, A/D convertor and CAPCOM.

RAM/ROM monitor debugging

The ROM monitor debugger can be used with any commercial off-the-shelf evaluation board or custom target board. The debugger, running on the host computer system, debugs your application on the target board through the monitor application via a RS232 or CAN interface. All files required to build the monitor are shipped with the VX-toolset ROM monitor debugger, including documentation on how to retarget the monitor to your target board.

For PC/Windows a Remote Evaluation Board Access Server utility is available from our website to allow debugging over a TCP/IP network on an evaluation board that is connected to a remote PC.

OCDS debugging through Infineon DAS support

Making the most of the On-Chip-Debug-Support (OCDS) facilities built into the latest Infineon C166 derivatives, our debugger offers high quality in-circuit-emulation functionality at low cost. The VX-toolset has been tested and qualified with Infineon's Debug Access Server (DAS) solution. The DAS environment is the universal emulation access software architecture for the XC800, XC166/XE166/XC2000 and TriCore microcontroller families, and it excels in stability and reliability. Extensive support for DAS is guaranteed by Infineon and, as a result, Altium has adopted this new debug standard.

Through DAS, the TASKING C166 debugger is compatible with selected Infineon starter kits with an on-board wiggler through a USB cable, as well as compatible with Infineon's DAP miniWiggler, enabling a very cost-effective debug solution for in-hardware testing for custom hardware or other evaluation boards.

Easy debugging of RTOS-based applications

Easy debugging of RTOS-based applications Altium's Kernel-aware Debugging Interface (KDI) defines an open standard interface between our debugger and an RTOS-Aware Debug Module (RADM). The RADM adds debugger capabilities to read, format and report kernel data structures for any commercial or proprietary RTOS. Our generic RADM for OSEK kernels, which is included in the toolset, is based on the ORTI 2.0 and 2.1 language specification.

Which bundle is the best for you?

Altium's TASKING VX-toolset is available in targeted bundles – Standard, Professional and Premium Edition, allowing you to choose the best fit for your application development activities.

The Premium Edition is the best choice if you want to have all essential development tools around the C compiler integrated into one environment. In addition to a software simulator, it offers you two on-hardware debugging solutions – using a ROM monitor solution with a serial I/O interface, or an On-Chip Debug System (OCDS) solution with a USB-to-JTAG wiggler. If your design is based on for example an XC166, XC2000 or XE166 device, the OCDS debugger is the most cost-effective on-hardware debugging solution you can get. The optional USB-to-JTAG wiggler may be ordered from Altium, or you can use a wiggler included in a starter kit from Infineon. If you develop (completely or partially) in C++, the Premium Edition provides you the best solution.

The Professional Edition provides flash memory programming facilities as well as on-hardware debugging through the ROM monitor solution, in addition to the C compiler and the software simulator. This Edition is a cost-attractive total solution bundle for editing, compiling, debugging and flashing.

The Standard Edition is your choice for C programming and debugging with a simulator. For debugging your code on hardware or programming your flash memory, you will require alternative tools from third parties. You can also upgrade to the Professional or Premium Editions at a later time, offering you all included functionality under one single user interface.

TASKING VX-toolset Editions for C166/ST10

	Standard Edition	Professional Edition	Premium Edition
Eclipse IDE, with C166/ST10 project configuration wizards	✓	✓	✓
Assembler/Linker/Locator	✓	✓	✓
C compiler with integrated MISRA C analyzer	✓	✓	✓
Migration utility for classic compiler and Keil™ compiler project conversion	✓	✓	✓
Project import wizard for Infineon's DAVE2	✓	✓	✓
Mini-RTOS	✓	✓	✓
Simulator debugger	✓	✓	✓
ROM monitor debugger		✓	✓
Flash programming		✓	✓
C++ compiler			✓
Integrated CERT C Secure Coding Standard analyzer in C compiler			✓
On-Chip Debug (OCDS) debugger (over USB port)			✓
USB-JTAG wiggler for OCDS debugging			Option

Migration from other C166 toolsets

For upgrading a project that has been developed with the original TASKING C166 toolset or the Keil™ C166 toolset, we provide both project conversion utilities as well as comprehensive guidelines with migration hints and tips. This will help you to make the step to the new technology and optimize your existing application.

The Classic C166 toolset will also remain available from Altium. Please visit the TASKING website for up-to-date information and a product datasheet.

Support for DAVe

The VX-toolset provides support for importing DAVe2 generated projects, through an import wizard. It is possible to copy a project to the workspace, or to create an Eclipse project into the directory of the DAVe project. Any updated DAVe configurations can be incorporated by a simple refresh of the DAVe settings in the project.

Tool partner support for the VX-toolset

Our active and extensive third-party tool vendor program ensures that you have access to the tools you need to be most productive. Altium works closely together with all relevant manufacturers of In-Circuit- Emulators, Real-Time Operating Systems, evaluation boards, CAN libraries, and CASE and UML tools for the C166 and ST10 architecture families. Contact the tool supplier of your choice for

information on Altium's TASKING VX-toolset compatible products, or consult our up-to-date third-party tool vendor catalog on our website.

Customer support

Altium is dedicated to providing quality products and support worldwide. This support includes program quality control, product update service, and support personnel ready to answer your questions by telephone, fax or email.

A 90-day technical support period is included with the purchase of TASKING toolsets and entitles you to enhancements and improvements as well as individual response to problems. Annual support agreements are available to extend this initial support period.

License management

Altium's TASKING VX-toolset for C166/ST10 includes the industry standard FLEXlm license manager, offering stability as well as flexibility. Its license 'borrowing' functionality is a popular feature, allowing laptop users to take a license from the network license pool for the period of time they are off-site, saving on cost for individual licenses.

The FLEXlm license manager also allows you to upgrade at a later stage to one of the more extensive VX-toolset Editions, giving you additional functionality. This is done by issuing you a new FLEXlm license key which simply unlocks the functionality in the existing installation on your system.



Product Information

Standard Edition

Product code: 07-200-119-804

Package contents: Eclipse IDE, C compiler, assembler/linker, simulator

Professional Edition

Product code: 07-200-119-806

Package contents: Eclipse IDE, C compiler, assembler/linker, simulator, ROM monitor debugger, Flash memory programming facilities

Premium Edition

Product code: 07-200-119-808

Package contents: Eclipse IDE, C++ compiler, C Compiler with integrated CERT C secure coding standard analyzer, assembler/linker/locator, Flash memory programming facilities, simulator, OCDS and ROM monitor debugger

*Please check the table on page 9 for a detailed overview of the contents/features in the various VX-toolset Editions.

Option: USB-JTAG miniWiggler*

Product code: 07-290-000-001

*This wiggler is available as an option to the Premium Edition for OCDS debugging. It can also be used with selected TASKING toolsets for the Infineon XC800 and TriCore architectures.

Fully-functional trial version

A fully-functional 15-day trial version of the TASKING VX-toolset for C166/ST10 is available on CD-ROM or downloadable from TASKING's website: www.tasking.com/C166

The trial version counts the days of effective use, which allows for an extensive evaluation period. The OCDS debugger can be evaluated without a separate USB-JTAG wiggler on selected Infineon evaluation boards that do have an on-board wiggler.

Minimum system requirements

PC/Windows platform:

- 2 GHz Pentium-class CPU
- 3 GB free disk space
- 2 GB memory
- 1024 x 768 display

Supported platforms and operating systems:

- Windows XP, Windows Vista, Windows 7 (32-bit, 64-bit¹)
- Sun/Solaris
- PC/Linux²

1. The VX-toolset is provided as 32-bit executable.
2. Selected versions of the VX-toolset only.

For more details contact your local Altium Sales and Support Center or Reseller.

More Information

- Contact your local Altium Sales and Support Center or Reseller
www.tasking.com/contacts
- To order Altium's TASKING VX-toolset for C166/ST10
www.tasking.com/C166

Altium offices worldwide

North America

Altium Inc.
3207 Grey Hawk Court
Suite 100
Carlsbad, CA 92010
Ph: +1 760-231-0760
Fax: +1 760-231-0761
sales.na@altium.com
support.na@altium.com

Germany

Altium Europe GmbH
Philipp-Reis-Straße 3
76137 Karlsruhe
Ph: +49 (0) 721 8244 300
Fax: +49 (0) 721 8244 320
sales.de@altium.com
support.eu@altium.com

France

Protel SA
(Filiale du groupe Altium Limited)
121 rue d'Aguesseau
92100 Boulogne-Billancourt
Ph: 0800 88 05 06
Fax: 0800 82 85 92
email.info.fr@altium.com
support.eu@altium.com

Australia

Altium Limited
3 Minna Close, Belrose
NSW 2085
Ph: +61 2 8622 8100
Fax: +61 2 8622 8140
sales.au@altium.com
support.au@altium.com

China

Altium Information Technology
(Shanghai) Co., Ltd
9C, East Hope Plaza
No.1777 Century Avenue
Shanghai 200122
Ph: +86 21 6182 3900
Fax: +86 21 6876 4015
sales.cn@altium.com
support.cn@altium.com

Japan

Altium Japan K.K.
Nomura Fudosan Yotsuya Bldg 7F
2-12-1 Yotsuya
Shinjuku-ku, Tokyo
160-0004
Phone: +81 3 6672 6155
Facsimile: +81 3 6672 6159
sales.japan@altium.com
support.japan@altium.com

The Netherlands

Altium Technology Centre
Altium BV
Saturnus 2
3824 ME Amersfoort
Ph: +31 33 4558584
Fax: +31 33 4550033
tasking@altium.com

For a full list of TASKING resellers visit www.tasking.com/contacts



Copyright © 2011 Altium Limited. Altium, Altium Designer, Board Insight, DXP, Innovation Station, Live Design, NanoBoard, NanoTalk, OpenBus, P-CAD, SimCode, Situs, TASKING, and Topological Autorouting and their respective logos are trademarks or registered trademarks of Altium Limited or its subsidiaries. All other registered or unregistered trademarks referenced herein are the property of their respective owners, and no trademark rights to the same are claimed.