



TASKING™

VX-Toolset for TriCore



Altium's TASKING VX-toolset for TriCore consists of:

- ISO C++ compiler, scalable to EC++
- C compiler, ISO C'99 compliant, with integrated 'MISRA C' enhanced code checking
- C compiler for the TriCore's Peripheral Control Processor (PCP)
- Assembler/Linker/Locator
- Assembler for the PCP
- C/C++ libraries, run-time libraries, floating point libraries
- Integrated Development Environment (IDE) based on Eclipse™
- Debugger integrated into Eclipse, with two execution environments:
 - Simulator
 - 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, XE166, XC2000 or TriCore-based development projects.

The TASKING TriCore compiler has become the de facto development solution for projects from automotive Tier-1 and OEM vendors where reliable, safe and secure applications require a proven and stable compiler technology.

The TASKING TriCore toolset

Altium's latest embedded software development tools for the TriCore microcontroller family offers substantial performance gains over previous generations of TriCore development tools. Based on Altium's sophisticated Viper compiler technology, this suite of tools is released as the TASKING VX-toolset for TriCore. 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 TriCore 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 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.

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. Along with the standard Eclipse, C/C++ compiler toolset and simulator modules, additional functionality include options such as a C compiler for the TriCore's Peripheral Control Processor (PCP), on-hardware debugging through an OCDS solution, USB-to-JTAG wiggler, TCP/IP stack reference design, plus flash memory programming support.

The C compiler for the PCP is a unique product on the market and only available from Altium. It maximizes the usability of the PCP by dealing with the inherent limitations of a restricted co-processor like the PCP and through generation of efficient and compact code. While the PCP compiler is an optional module, the PCP assembler is included in all bundles.

The OCDS debugger supports debugging using the JTAG bus on a wide range of standard evaluation boards from various manufacturers, as well as on custom developed boards, while third-party debug support is guaranteed by leading tool vendors, such as Hitex, Lauterbach and PLS.

Altium's TASKING VX-toolset for TriCore supports all Infineon Technologies TriCore derivatives, but it is also ready to support selected new family members. Add-on support for special custom variants is available from Infineon or Altium and enhances the toolset with awareness for such a specific derivative.

The VX-toolset comes with a selection of example projects that can be imported into the IDE. These projects can be an excellent starting point to get familiar with the TriCore and the toolset. In the Professional and Premium Editions we provide a TCP/IP stack reference design project. Through this example one can test a comprehensive "real life" project on a TriCore TC1130 evaluation board and with the rights to even include the stack in an own application.

Eclipse IDE

The Integrated Development Environment (IDE) that is built on the Eclipse framework provides a seamless workbench for the complete tool chain including the debugger of the VX-toolset. The IDE provides facilities for project configuration and management, C/C++ and assembly code aware editing, build management, debugging, profiling and more. It provides wizards to help you to set up your embedded TriCore project and to configure your target board settings to debug your project on hardware.

The Eclipse editor supports C, C++, assembly language and header files with syntax highlighting, auto completion, context assistance and tool tips. As you would expect from a de facto standard IDE, it provides full support for all relevant source code version control systems. The Eclipse environment provides you with a single platform for many different embedded product toolsets from different vendors. The standardization on an industry-wide IDE significantly reduces your learning curve, removes the barriers of changing development tools for different architectures, increases your productivity and ultimately reduces the time to market with your end product. The availability of plug-in modules to enhance or extend the feature set of the Eclipse IDE ensures that you can build the workbench according to your development needs. With the concept of the open Eclipse framework, third-party tool vendors can now develop plug-ins that tightly integrate into various IDEs from different vendors, contrary to proprietary IDEs for which custom connectivity needs to be created.

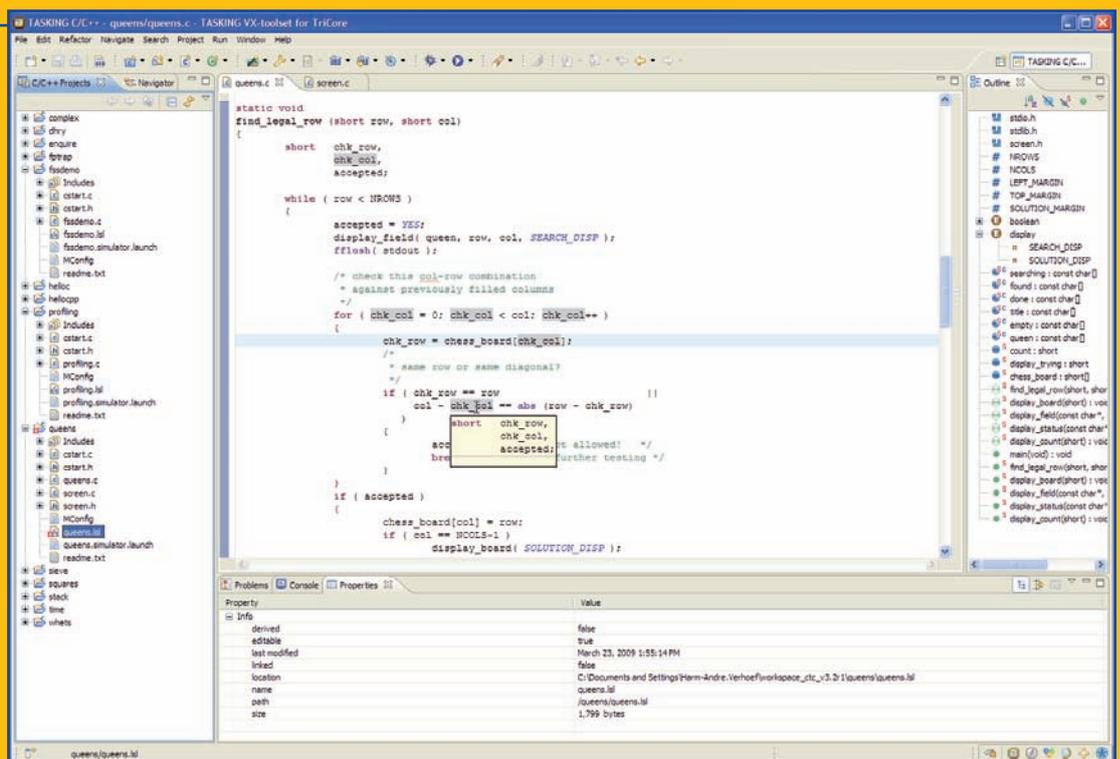
The VX-toolset's IDE is based on the most current v3.4 release of Eclipse and the C/C++ Development Tools (CDT) plug-in version 5.0. Altium has built the integration blocks for the toolset and extensions to Eclipse to make the whole environment a coherent workbench.

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



Eclipse Integrated Development Environment for editing and project management.

Target architecture support

Altium's TASKING VX-toolset supports all TriCore microcontroller derivatives. From within the Eclipse IDE you can easily select the Infineon TriCore device of your choice for your project:

- Industrial and Multi-market devices: TC1100, TC1115, TC1130, TC1161-TC1167, TC1197
- Automotive devices: TC1736, TC1762-TC1767, TC1775, TC1792, TC1796, TC1797

as well as all of their individual variants.

Altium's active relationships with Infineon Technologies enable us to support new derivatives already in the toolset prior to their availability in volume. The VX-toolset also allows for adding support of new derivative variants easily and quickly through new processor definitions in an XML configuration file to the Eclipse IDE. Please consult our support engineers for the latest status of support for new 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, ELF/DWARF 2, parts of DWARF 3, Infineon EABI, IEEE-754, S-Record and ORTI.

Compiler optimizations include:

- 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

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

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.

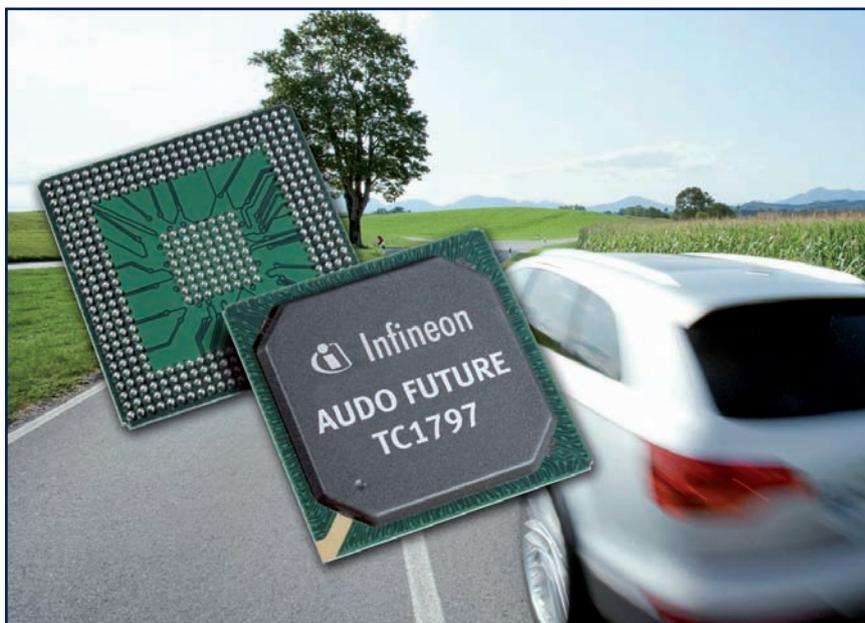
These optional checks are implemented by generating additional code and/or enabling additional code in the standard C library. Run-time error checking has a nominal effect on code size and execution speed and can be enabled on a module-by-module basis, making it practical for use in debugging large applications.

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

MISRA C

Altium was the first company to fully integrate MISRA C support into C compilers for embedded development purposes. MISRA C guides programmers in writing more robust C-code by defining selectable C-usage restriction



Unique features TASKING VX-toolset

- Market's-only C compiler for the PCP
- The code generation chain compiler-assembler-linker is not based on open-source code
- Very stable and reliable code-generation, with proven use at many automotive vendors in powertrain and body control applications
- Efficient optimizations for fast and small code
- Support for MISRA C guidelines
- Parallel build support for efficient compilation on multi-core systems
- Eclipse based IDE with integrated debugger
- Ability to add support for new processors through dynamic definition in an XML configuration file
- TCP/IP reference design as sample project
- Legacy version support services available

rules. Through a system of strict error checking, the use of error-prone C-constructs can be prevented. The latest step in this innovation is configurability of the compliancy checking. The MISRA rules, which the application's source code should be compliant with, can be set as 'required' or 'advisory' and the diagnostic level of the generated messages by the compiler can be defined as either 'warning' or 'error'. This allows you to configure the individual rules of the MISRA C compliancy validation according to the quality standards set by your company.

The VX-toolset supports the new MISRA-C:2004 standard as well as the original MISRA-C:1998 guidelines.

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 TriCore 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.

C compiler for PCP

Altium is offering a unique C compiler for the TriCore's Peripheral Control Processor (PCP). Despite the limited functionality and restricted instruction set of the PCP, we have been able to develop a fully functional C compiler. The C compiler delivers code at an unexpectedly high performance level and provides several special extensions for PCP programming, such as:

- Through a memory qualifier the user can put variables in the TriCore linear space, enabling sharing of memory between the TriCore and the PCP.
- As the PCP uses a static stack mechanism for performance purposes, it is not trivial to have code that can be interrupted. The compiler is however equipped with an option that allows code to be generated that is interruptible.

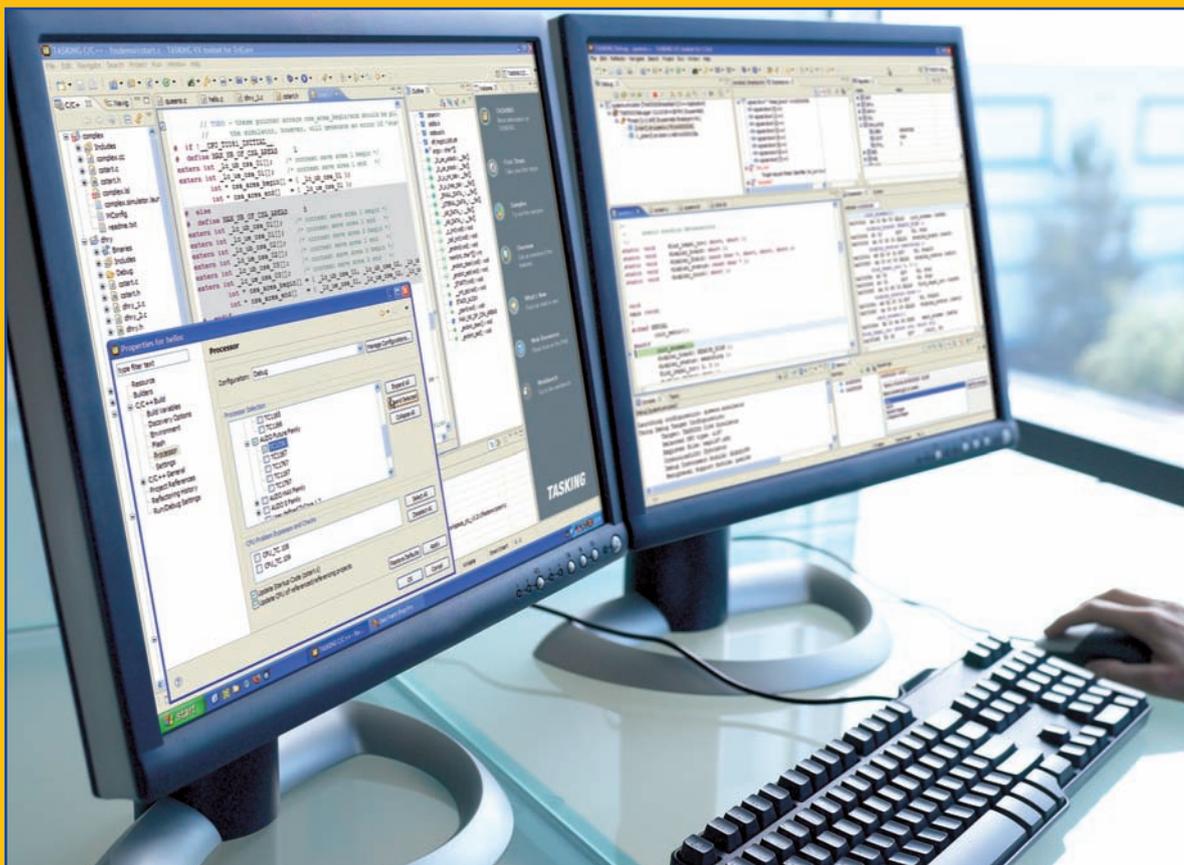
- Compact code is achieved by enabling the compiler to optimize on an application wide scope, as well as by using the memory partitioning technique that reduces the number of times that the data pointer needs to be reloaded.

The PCP C compiler is fully integrated into the Eclipse IDE and is part of the Premium Edition product bundle. Programming the PCP in assembly code is possible with all Editions, as the PCP assembler is included in all three bundles.

C++/EC++ compiler

Fully aware of the undeniable trend towards higher level language programming, the TASKING VX-toolset for TriCore 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 TriCore family. The object-oriented benefits of C++ can be incorporated into your TriCore 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.



Full control over editing and debugging your code.

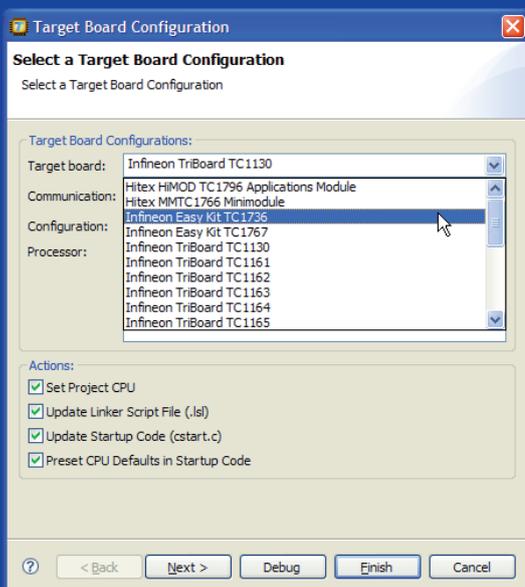


Evaluation board and custom board support

The OCDS debugger in the TASKING VX-toolset supports 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:

- Hitex Development Tools
- Infineon Technologies
- Isle
- Phytex
- 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 USB cable, a parallel cable or USB-JTAG miniWiggler.



USB-JTAG miniWiggler

TriCore instruction set simulator debugging

The TriCore simulator debugger features instruction set simulation, allowing you to extensively debug your application on the host platform, even before your target hardware is available. A plug-in for instruction set simulation of the PCP is included.

OCDS debugging through Infineon DAS support

Making the most of the On-Chip-Debug-Support (OCDS) facilities built into the Infineon TriCore microcontrollers, 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 XC8xx, 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 TriCore debugger is compatible with selected Infineon starter kits with an on-board wiggler through a parallel cable or a USB cable, as well as compatible with Infineon's USB-JTAG 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

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.

The RADM extends the debugger with impressive kernel-aware debugging. Features include:

- Display levels of kernel information
- Examine and modify kernel data structures
- Obtain a summary of all tasks
- View contexts of tasks
- Inspect message contents (pipes, queues, mailboxes)
- Status of synchronization mechanisms
- Interrupt service routine status

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, communication protocols (CAN, Flexray), and CASE and UML tools for the TriCore architecture. 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.

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/C++ compiler integrated into one environment. In addition to a software simulator, it offers you an on-hardware debugging solution – using an On-Chip Debug System (OCDS) solution with a USB-to-JTAG wiggler. The OCDS debugger is the most cost-effective on-hardware debug solution you can get. The optional USB-to-JTAG miniWiggler may be ordered from Altium, or you can use selected starter kits from Infineon that come with an on-board wiggler. The truly unique part of the Premium Edition is the C compiler for the Peripheral Control Processor (PCP). If you plan to write your code for the PCP in C language, this Premium Edition is the best and only option on the market.

The Professional Edition provides everything from the Premium Edition, including the OCDS debugger, but without the PCP C compiler. So if you develop your

application based on a TriCore derivative without the PCP, this Professional Edition is a full suite solution for you. Also, if you are fine with programming your TriCore with PCP in assembly code instead of C, this package is a good choice. Add the optional USB-to-JTAG miniWiggler in case your hardware board comes without an on-board wiggler.

The Standard Edition is your choice for C/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 upgrade to the Professional or Premium Editions at a later time, offering you all included functionality under one single user interface.

Migrating from VX-toolset v2.x

For upgrading a project that has been developed with the TASKING VX-toolset for TriCore release v2.x, we provide a project conversion utility. This tool helps to convert an EDE/Codewright based project into the Eclipse IDE.

TASKING VX-toolset Editions for TriCore

	Standard Edition	Professional Edition	Premium Edition
Eclipse IDE, with TriCore project configuration wizards	✓	✓	✓
C and C++ compiler for TriCore	✓	✓	✓
TriCore assembler/linker/locator	✓	✓	✓
C compiler for Peripheral Control Processor (PCP)			✓
PCP assembler	✓	✓	✓
Simulator debugger	✓	✓	✓
On-Chip Debug (OCDS) debugger		✓	✓
USB-JTAG miniWiggler for OCDS debugging		Option	Option
Flash programming		✓	✓
TCP/IP stack reference design		✓	✓