

HIGHLIGHTS

- **World's best-selling TriCore compiler tool set**
- **Complete integrated development environment**
- **Highly optimizing C/C++/EC++ compiler**
- **MISRA C code checking**
- **TC1 v1.3 support:**
 - Derivatives
 - Auto Cstart code
 - Optimized libraries
 - FPU/MMU/PCP 2
 - Compiler bypasses
 - Assembler checks
 - Protected libraries
- **CrossView Pro debugger**
 - TriCore I/S simulator
 - On-Chip Debug Support
 - Advanced Breakpoints
 - OSEK kernel aware
- **Available for PC/Windows, PC/Linux and SUN/Solaris**

THE TASKING TRICORE TOOLSET

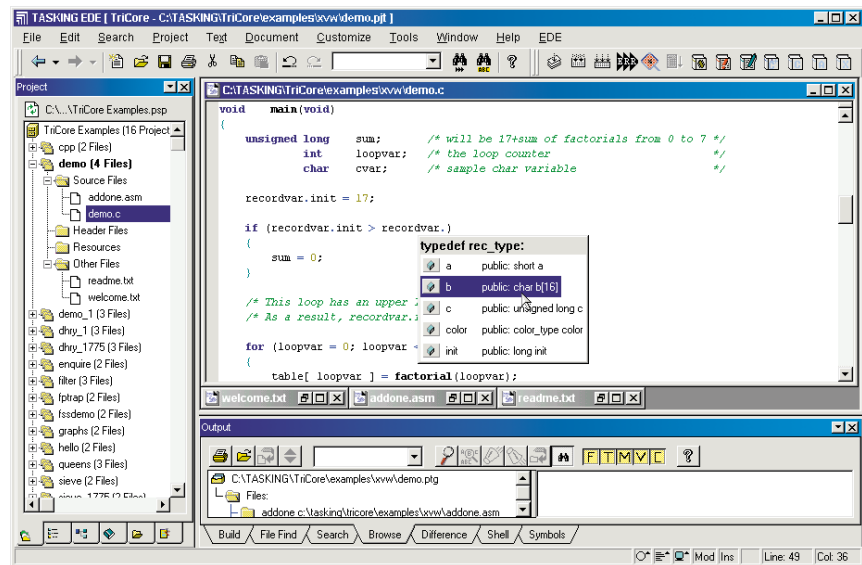
The TriCore™ architecture from Infineon Technologies is a revolutionary single-core 32-bit microcontroller/DSP design optimized to offer the best of three worlds: real-time capabilities of a high-speed microcontroller, computational prowess of a DSP and the best performance/price ratio of RISC load/store architecture.

Built upon the strong co-operation with Infineon, the TASKING TriCore Software Development Toolset is commonly regarded as the de facto industry standard for TriCore architecture software developments.

EMBEDDED DEVELOPMENT ENVIRONMENT

With the TASKING TriCore EDE, you can create and maintain projects the easy way. All project related aspects, such as the application source files, the tool options (compiler, assembler, linker/locator, CrossView Pro debugger), file management, and the options of the build process, are managed from one central point. File dependencies as well as the sequence of operations required to build the application are handled automatically. The TriCore EDE offers many productive features for application and code development:

- **Project Spaces allow grouping of multiple projects in one view, thus offering improved project management for more complex developments.**
- **Right-Mouse-Button clicks expedite a variety of tasks within the EDE (e.g. creating new files, adding files to a project, etc.)**
- **CodeSense advanced coding assistance offers rich type-ahead features, which help in selecting the next expected function parameter, or available structure members. When positioning your mouse pointer over a function name, the function prototype will be displayed.**



- **Tags Browsing offers a graphical overview of the applications cross-references and allows easy navigation through the available variables and functions.**
- **CodeFolio offer easy insertion of 'snippets' template code, thus adding to coding efficiency. It allows macro expansion and prompted input as you insert the code.**
- **HTML View Window allows browsing through the product manuals, project or code documentation or even surfing the net.**
- **XML Collapsible Grid Viewer displays the hierarchy of elements and element attributes in XML documents.**
- **Split Windows provide full control over source code by allowing you to splits your file horizontally or vertically to up to four edit windows.**

FULL TC 1 V1.3 ARCHITECTURE SUPPORT

The TASKING TriCore tools are fully prepared for the complete range of Infineon TriCore architectures, including TC1v1.2 and TC1v1.3. Specific support for the TC1 v1.3 extensions are available in the TriCore- and PCP Assembler:

- FPU (Floating Point Unit)
- MMU (Memory Management Unit)
- PCP2 (Peripheral Control Processor 2)

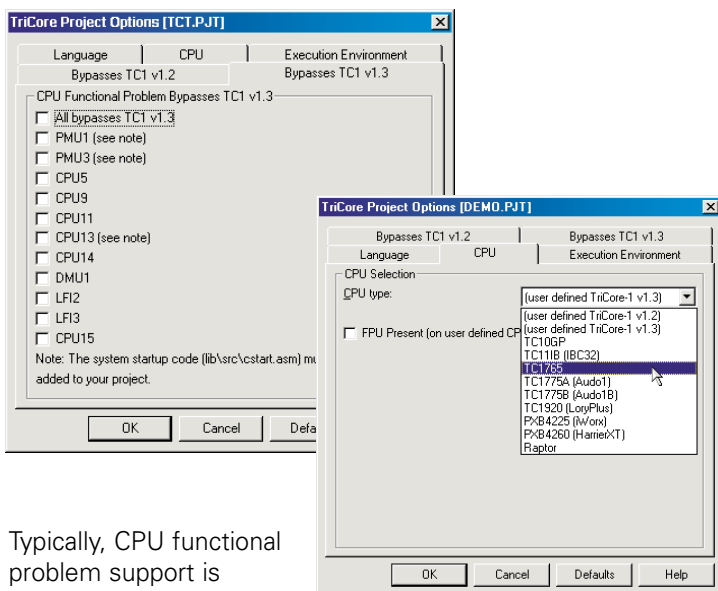
TriCore Derivatives support

TASKING tools provide support for all publicly available TriCore derivatives. Among the supported TriCore devices are:

- TC10GP
- TC111B
- TC1765
- TC1775B
- TC1920
- PXB4225
- PXB4260
- Raptor
- PMB8860
- PMB8880
- and more...

CPU functional problem support

Infineon Technologies regularly publishes microcontroller errata sheets reporting deviations from the electrical and timing specifications. As an integral part of best architecture support, the TASKING TriCore tools provide bypasses and checks for identified silicon defects.



Typically, CPU functional problem support is provided throughout the complete tool chain:

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

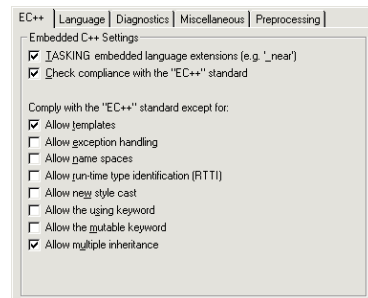
Protected C-library sets built with bypasses for all identified TriCore CPU functional problems

C++/EC++ COMPILER

The ISO C++ compliant compiler delivers the power of object-oriented design and coding techniques for the TriCore family. The benefits of C++ can be incorporated into an existing C application, one module at a time, providing a graceful migration from C to C++.

Scalable C++

Compatibility with the evolving Embedded C++ (EC++) standard allows selective disabling of C++ features that may not be essential for your embedded application. By selecting (partial) compliance with the EC++ standard, code-size overhead and run-time inefficiency can be minimized.



C COMPILER

Based upon TASKING's renowned expertise, the TriCore C compiler is highly optimized and fully compliant with the ISO C standard taking full advantage of the TriCore architecture.

The TASKING compilers are tested for ISO/ANSI conformance against authoritative validation suites, such as Plum Hall and Perennial. Additionally, the optimization techniques of the compilers are tested with various large real-world applications as well as industry benchmark standards such as Nullstone and EEMBC.

The TriCore C compiler features include:

- Extensive optimizations for very efficient code
- Re-entrant code with re-entrant libraries
- `_near` and `_far` memory qualifiers
- C level I/O register access
- C level (Fast) Interrupt definition
- Complete C and runtime libraries

Optimizations

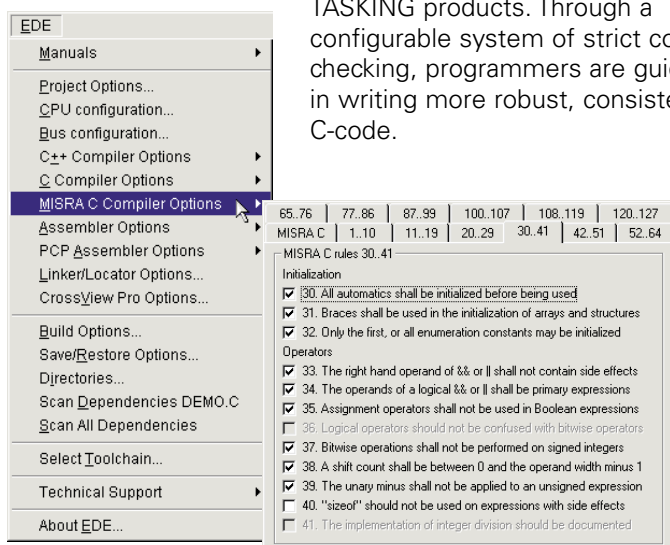
The TriCore compiler tools offer a wide variety of optimizations to reduce code size and execution time, exploiting new techniques for achieving the optimal fit between the compiler and the target architecture.

Optimization technologies include:

- Common Sub Expression Elimination detects and eliminates repeating (sub-) expressions.
- Various Loop and Jump optimizations to speed up execution and reduce code size.
- The Instruction Pipeline scheduler rearranges instructions to take advantage of parallel capabilities of the TriCore architecture.
- Dead assignment, Dead storage and Dead code elimination remove unreachable code or invariant data.
- Peephole optimizations replace instruction sequences with equivalent but faster and/or shorter sequences, or delete obsolete instructions.

MISRA C

Based on the “Guidelines for the use of the C language in vehicle based software” published by the Motor Industry Software Reliability Association (MISRA®), Altium is the only vendor offering advanced code checking in standard software development tools with their TASKING products. Through a configurable system of strict code checking, programmers are guided in writing more robust, consistent C-code.



A predefined configuration for compliance with the required rules in MISRA C guidelines is ready prepared, but it is also possible to enable a custom set of MISRA C rules to suit company requirements. To ensure compliance with the MISRA C rules throughout the entire project, the TriCore Linker/Locator can generate a MISRA C Quality Assurance report. This report lists the different modules in the project with the respective MISRA C configurations, which were used to compile them and could be filed under the company's Quality Assurance system.

Data Types

In addition to the standard ISO C types, the TASKING TriCore compiler has been extended with embedded and DSP specific data types, such as `_fract`, `_sfract`, `_accum`, `_circ`, `_sat`, `_bit`, `_packb` and `_packhw`. The keyword `_circ` has been defined to cater for circular buffer pointers. The `_sat` data modifier allows enabling and disabling of the TriCore saturation logic. The keywords `_bit`, `_packb` and `_packhw` are available for bit data types and packed automatics respectfully.

User Inline C Functions and Inline Assembly

The `_inline` keyword enables the definition of user inline C-functions. To deliver the fastest possible implementation, `_inline` function calls are stripped from their redundant 'calling and parameter passing' overhead. Inline assembly in the C compiler is supported via `#pragma asm` / `#pragma endasm`. The TASKING TriCore Inline assembly supports C-level variable passing and return values, as well as the specification of scratch registers, as to allow the compiler to perform optimizations.

Intrinsic Functions

The TASKING TriCore toolset has a wealth of built-in intrinsic functions. Intrinsic functions appear as normal C functions, but the code generator interprets them, as to generate more

efficient code. Several pre-declared functions are available to generate inline assembly code at the location of the intrinsic function call, avoiding the standard parameter-passing overhead.

Libraries

The TriCore compiler toolset contains all the necessary ISO C libraries, run time libraries, and floating-point libraries. Particularly the Floating-point libraries are supplied in many variants.

Special FPU-optimized floating-point libraries offer specific support for the TriCore devices that are equipped with the FPU. TriCore devices that do not include the FPU are supported with the software floating-point libraries.

All floating-point libraries come in double- and single precision variants, offering the choice between (IEEE 754-1985 compliant) precision and increased speed.

Trapping/non-trapping variants of the floating-point libraries offer the choice between run-time error checking and increased speed.

TRICORE / PCP ASSEMBLERS

The TASKING TriCore tool chain incorporates separate assemblers for the TriCore and the PCP (Peripheral Control Processor). The PCP assembler complies with PCP1 and PCP2 instruction sets.

TriCore/PCP Assembler features includes:

- Full macro and conditional assemblers
- Branch/call instruction optimizations
- Extensive section directives
- Error file with textual error reporting
- IEEE695 object output format with HLL debugging extensions
- Absolute list file generation

Linker/Locator

The linker/locator is an essential part of the software building process that enables linking and location of data and code in target memory. The Locator locates a linker-file to absolute addresses. The ability to accurately describe the available memory and control the behaviour of the locator is crucial for successful development of embedded applications.

Linker/Locator features include:

- Automatic or user defined allocation of code and data in memory
- Data/Code section initialization
- Incremental linking
- Industry standard IEEE695 and ELF/Dwarf 1.1 object output format including HLL debug-information
- Support for industry standard ELF/Dwarf2 objects and library archives ensuring co-operation with available TriCore Real Time Operating Systems and Libraries
- Complete map file
- MISRA C Quality Assurance report generation
- SREC and Intel HEX ROM image output formats

CROSSVIEW PRO DEBUGGER

An easy-to-use interface with powerful and extensive debugging features helps you debug your applications faster. CrossView Pro provides multiple, resizable, and independently controlled windows.

You choose the windows you need to view the different aspects of your code during debugging. It combines the flexibility of the C language with the control of code execution found in assembly language, bringing functionality that reduces time spent on testing and debugging.

Functionality includes:

- Simple as well as advanced debugging features
- Intuitive source window
- Tracking scope and monitoring local variables
- Bubbly-Spy™ for easy inspection of variables and functions
- Double click and right mouse button functions
- Clipboard copy and paste

Source Window

The source window is the main debugging window. It allows you to view source; step through your application; set and clear breakpoints and assertions; watch and show variables; search for strings, functions, lines and addresses; call functions; evaluate expressions; and view performance analysis data. The source window can display code in C/C++ source, assembly or a mixed mode that allows a simultaneous view on your C/C++ source, intermixed with the corresponding assembly code. Immediate access to your source files is possible by jumping directly from the CrossView Pro Source window into the EDE editor at the exact source line.

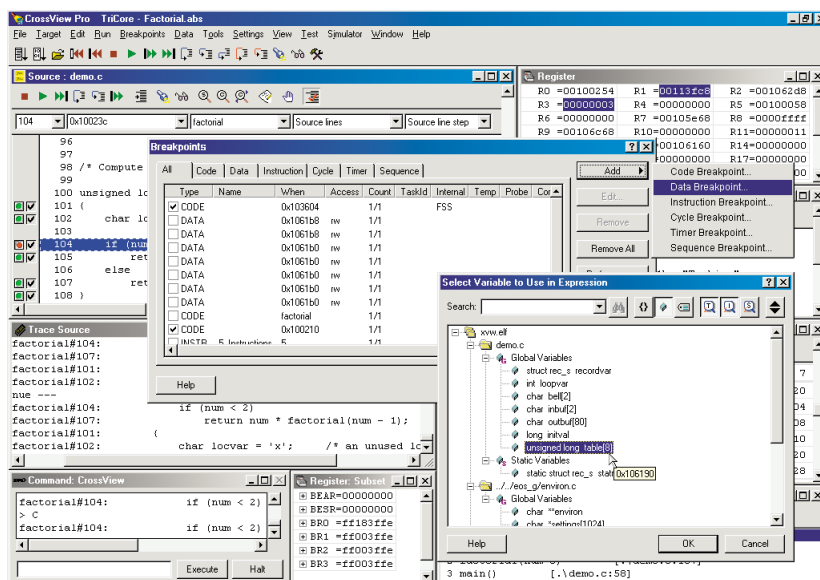
Multiple Information Windows

CrossView Pro offers a wealth of information windows allowing you to navigate through your application, monitor and modify Data objects, CPU registers, the call-stack and memory locations.

The Data window enables you to watch or show data, browse for locals or globals, double-click to modify values or to expand and contract complex data structures. Within this window, you can reformat (change display of radix and type) on an element-by-element basis. You can show or watch locals from any stack level, automatically track and display locals, and easily copy any variable as show or watch.

Register windows allow display and modification of CPU register values. Register windows are fully configurable to display any set of CPU registers. By defining multiple Register windows you can easily organize your focus.

The stack window displays the contents of the function-call stack frame. You can easily configure stack-level breakpoints, navigate to the function call's source and monitor local variables for selected functions.



The Memory window enables you to monitor and modify any memory location, with complete control over size and format of the data, as well as view coverage of the memory range.

Advanced Breakpoints

Breakpoints halt program execution and return control to the user. In addition to industry standard code- and data breakpoints, you can define your application to halt based upon Instruction counts, Cycle counts, or Timer counts. All types of breakpoints can be defined as 'stop-and-go' Probe-points. Probe points briefly halt and immediately resume execution of the application. During the brief period that the application is halted, only user-specified actions will be performed. Through this mechanism, Probe points allow least-intrusive debugging of time critical applications.

Finally, any number and type of breakpoints can be combined into Breakpoint-sequences. This allows specification of the most complex conditions you want to examine.

I/O Simulation

CrossView Pro I/O Simulation (IOS) allows the use of standard ISO C system calls such as open(), read(), printf() and scanf() within your embedded application as to interface with the host PC file I/O services.

Using IOS, you can read from and write to files on the host PC or a CrossView Pro Virtual I/O window directly. IO Simulation will work in any CrossView Pro target execution environment: I/S Simulator, OCDS and supported In-Circuit Emulators.

Multiple Execution Environments

CrossView Pro supports multiple execution environments with the same standard interface.

TriCore Instruction Set Simulator. With CrossView Pro and the bundled TriCore Instruction Set Simulator, you can debug your application on the host platform even before your target hardware is available. The Simulator supports all instructions of the TriCore Instruction Set.

Target debugging through OCDS. Taking advantage of the TriCore On-Chip-Debugging-System (OCDS) facilities, CrossView Pro offers high quality in-circuit-emulation functionality at a low cost. Via the host PC's parallel interface, CrossView Pro can communicate with TriCore device directly.

Selected TriCore In-Circuit Emulators / JTAG Interfaces.

CrossView Pro can be used in combination with Hitex and Signum emulators and the Abatron BDI200 JTAG interface. This gives you the benefit maintaining the same debugging environment while moving from I/S Simulator to target debugging.

Program Performance Analysis

CrossView Pro provides several performance analysis capabilities to help you further optimize your application as well as shorten your debugging session.

Code Coverage and Profiling

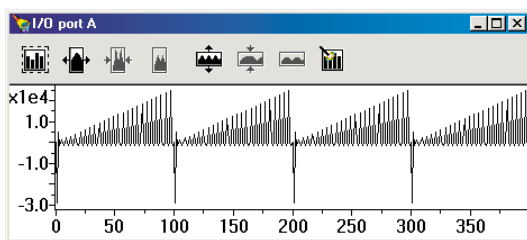
Coverage enables you to check whether your specific parts of your application code actually have been executed. Based on the code coverage reports you can build a complete test suite for your product and improve the quality of your application.

Profiling allows you to perform timing analysis on the complete application or particular parts of it. Profiling information can be shown in the left margin of the source window, but is also presented in table format, providing you the full overview.

Based upon the profiling information you can easily decide which functions should be optimized for speed.

Graphical Data Analysis

CrossView Pro's outstanding Programmable Graphical Data Analysis simplifies quick detection of gross errors in signal processing routines, such as typically used in DSP applications. By displaying large sets of data in meaningful visual diagrams, CrossView Pro allows you to analyze the data without the need of reviewing or post-processing large files of raw data. You can also view the same set of data in several ways at the same time (E.g. in time- and frequency domain).



Four different analysis types are ready prepared: x-t plotting, x-y plotting, FFT (Fast Fourier Transformation) power spectrum and Eye diagram. The C-language scripts for these pre-defined graphs can be easily used as the basis for custom Data Analysis windows.

EASY Debugging RTOS-based applications

The TASKING Kernel aware Debugging Interface (KDI) defines the communication between CrossView Pro and an RTOS-Aware Debug Module (RADM). The RADM adds the capability to CrossView Pro to read, format and report kernel data structures. The KDI specification describes the open interface that can be used to add kernel-awareness to CrossView Pro for any commercial or proprietary RTOS. CrossView Pro RADMs are available for most popular commercial real-time operating systems including: CMX, any OSEK ORTI compliant RTOS, OSEKWorks and Nucleus PLUS.

The RADM extends CrossView Pro with impressive Kernel-Aware Debugging capabilities, such as:

- 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

Our integrated RTOS support enables you to debug your RTOS-based application more quickly and easily.

COOPERATION WITH THIRD PARTIES


Our extensive third party cooperation ensures that you have access to the tools you need to be your most productive. TASKING works closely together with manufacturers of In-Circuit-Emulators, Real Time Kernels and Evaluation boards for the TriCore.

Infineon DAVe

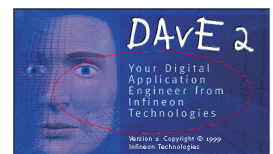
Infineon Technologies' Digital virtual Application Engineer "DAvE" can assist you in selecting the TriCore device most suitable for your particular application. He also assists you with configuring and initializing the relevant peripherals within the selected device and generating a complete set of source files that can be used as the basis for further development of your project. The sources cover configuration and initialization of the peripherals and include function templates as to make use of the peripherals.

The TriCore tools are optimally tuned for effortless import of projects created with Infineon's DAVe.

Compatibility with DAVe2 can be as simple as:

- Create a new project in EDE
- Add the DAVe .DPT file
- And click the 'refresh' button 

Also, after having adjusted the device configuration within DAVe2, the refresh button in EDE ensures all forthcoming modifications to be properly inherited.



CUSTOMER SUPPORT

When you purchase a TASKING product, it is the beginning of a long-term relationship. 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 maintenance period is included with the purchase of TASKING products and entitles you to enhancements and improvements as well as individual response to problems. Annual maintenance agreements are available to prolong this initial support period.

PRODUCT PACKAGING & ORDERING CODES

Each TASKING product comes with full printed documentation. This documentation is also available on-line in the form of a Windows Help system, HTML and PDF and provides full-text search capabilities for quick and easy access to topics.

Product Code	Package contents
TK060-012	EDE, C/C++/Safer C compiler, Assembler, Linker/Locator
TK060-044	CrossView Pro OCDS/Simulator debugger

Demonstration versions of the TriCore Software Development tools are available on CD-ROM or downloadable from our web site at:

<http://www.tasking.com/products/tricore/>

INTERNET

Web site: www.tasking.com

Developers forum: www.yahogroups.com/group/TASKINGforum

DISTRIBUTOR

ALTium SALES OFFICES



NORTH AMERICA

333 Elm Street
Dedham MA 02026-4530 USA
Telephone: +1 781 320 9400
Facsimile: +1 781 320 9212
Email: tasking.sales.na@altium.com

THE NETHERLANDS

Plotterweg 31
3821 BB Amersfoort
The Netherlands
Telephone: +31 33 455 85 84
Facsimile: +31 33 455 00 33
Email: tasking.sales.nl@altium.com

GERMANY

Eltinger Straße 61
D-71229 Leonberg
Germany
Telephone: +49 71 52 979 910
Facsimile: +49 71 52 979 9120
Email: tasking.sales.de@altium.com

FRANCE

21 Avenue du Québec
91951 Les Ulis Cedex
France
Telephone: +33 1 69 59 26 10
Facsimile: +33 1 69 59 26 11
Email: tasking.sales.fr@altium.com

SWITZERLAND

Unterdorfstrasse 1
CH-4334 Sisseln
Switzerland
Telephone: +41 62 866 41 11
Facsimile: +41 62 866 41 10
Email: tasking.sales.ch@altium.com

JAPAN

ASAHI-GIN Gotanda Building 7F
23-9, Nishi-Gotanda 1-chome
Shinagawa-ku Tokyo 141-0031
Japan
Telephone: +81 3 5436 2501
Facsimile: +81 3 5436 2505
Email: tasking.sales.jp@altium.com