

استفاده از سیستم عامل سیمبین

CARBIDE.C++ V1.3

V1.3



symbian
Press

Accredited Symbian Developer

(برنامه‌نویس مجاز Symbian)، یک مهارت

استاندارد صنعتی برای برنامه‌نویسان

نرم‌افزارهای گوشی‌های هوشمند است.



آزمون برنامه‌نویس مجاز Symbian، توانایی‌های داوطلبان را در مقابل برنامه آموزشی اطلاعات پایه‌ای می‌سنجد. این دوره آموزشی حاوی موضوعات تئوری و عملی می‌باشد که هم فهم و هم کاربرد C++ را در حوزه سیستم عامل Symbian می‌آزماید.

هر آزمون شامل مجموعه‌ای از سؤالاتی است که از مخزن سؤالات همان دوره انتخاب شده‌اند. سؤالات وزن‌دار هستند و از امتحان‌دهنده فقط سؤالاتی در سطح توانایی‌هایش که در طول مدت امتحان نشان خواهد داد، پرسیده خواهد شد. افرادی که در امتحان قبول شوند، عنوان معتبر صنعتی «برنامه‌نویس مجاز Symbian» را دریافت خواهند کرد.

برای اطلاعات بیشتر، به سایت‌های ذیل مراجعه کنید:
و www.symbian.com/developer/academy
<http://www.majinate.com>

استفاده از سیستم عامل سیمبین

CARBIDE.C++ V1.3

Carbide.C++ ویرایش ۱.۳

از سری «استفاده از سیستم عامل سیمبین»

ویرایش اول، ۰۴/۰۸

منتشر شده توسط:

Symbian Software Limited
2-6 Boundary Row
Southwark
London SE1 8HP
UK
www.symbian.com

علائم تجاری، حق تألیف، رفع مسئولیت

کلیه علائم تجاری Symbain OS, Symbian و دیگر علامت‌های وابسته به Symbian متعلق به شرکت نرم افزاری Symbian Software Ltd می باشند. Symbian کلیه حقوق مربوط به علائم تجاری اشخاص ثالث را که در این مستند به آنها اشاره شده است، تصدیق می‌نماید. © 2008 کلیه حقوق مادی و معنوی این مستند متعلق به شرکت Symbian Software Ltd می باشد. هیچ قسمتی از این مستند نباید بدون مجوز مکتوب و صریح از طرف شرکت Symbian Software Ltd تکثیر شود. شرکت Symbian Software Ltd هیچ ضمانتی مبنی بر دقت و یا مناسب بودن اطلاعات این سند نمی‌دهد. اطلاعات داخل این سند فقط جهت ارائه اطلاعات کلی و عمومی است و به هیچ وجه نباید برای هدف دیگری استفاده شوند.

مؤلف:

اندرو مارک اسمیت

ویراستار:

اشلی گادوین

بازبینی:

جو استیکبوری

نیل تیلور

دیوید دورانت

مایک تروژیلو

مشاور طراحی:

آنابل کوک

سابینا آسلام

ترجمه به فارسی:

موسی مرادی

بازنگری ترجمه:

مصطفی هادیان دهکردی

سمیرا عباس‌نژاد

آزیتا اسماعیلی

فهرست محتویات

۶ مقدمه
۶ Eclipse اسکوی برنامه‌نویسی
۷ Carbide.c++ محیط یکپارچه توسعه
۱۰ Symbian C++ وارد کردن پروژه‌های موجود
۱۲ Symbian جدید ایجاد یک پروژه
۱۴ ویژگی‌های پروژه
۱۶ تنظیمات سراسری
۱۸ پیمایش و ویرایش کد
۲۰ ساخت و اجرای یک برنامه
۲۲ نشست‌های اشکال‌زدایی
۲۴ Carbide.c++ ویرایش ۱.۳ ویژگی‌های جدید در
۲۶ اطلاعات و یاری
۲۷ منابع برنامه‌نویسی

مقدمه

این کتابچه حاوی اطلاعات کلی جهت آشنایی با نرم‌افزار Carbide C++ ویرایش ۱.۳ به عنوان یک محیط یکپارچه توسعه^۱ جدید برای برنامه‌های Symbian C++ است که آن را می‌توان از فروم نوکیا (www.forum.nokia.com) و یا سایت برنامه‌نویسی UIQ (developer.uiq.com) دریافت کرد.

سکوی برنامه نویسی Eclipse

Carbide C++ بر مبنای سکوی برنامه نویسی Eclipse بنا شده است. این سکوی در نوامبر سال ۲۰۰۱ توسط شرکت IBM و هفت شرکت دیگر، به عنوان یک پروژه متن‌باز با هدف ساختن یک IDE رایگان و قابل اجرا روی سیستم عامل‌های مختلف، ایجاد گردید و توسط بنیاد Eclipse (www.eclipse.org) و شرکت‌های عضو (شامل Symbian و نوکیا) پشتیبانی می‌شود. Eclipse در این مدت به یک انجمن متن‌باز تبدیل شده است که پروژه‌های بنا شده در آن، بر تهیه یک سکوی برنامه‌نویسی توسعه پذیر و چارچوب کاری^۲ جهت ساخت نرم‌افزار، متمرکز شده‌اند. در نتیجه تعداد رو به افزایشی از شرکت‌ها، در حال تولید محصولات تجاری و یا رایگانی هستند که بر تکنولوژی Eclipse بنا شده‌اند؛ به عنوان نمونه می‌توان به Carbide C++ اشاره کرد.

هدف اصلی Eclipse، ایجاد محیطی جهت سهولت یکپارچگی ابزارهای مختلف است. برای رسیدن به این هدف، Eclipse، از یک معماری که شامل چارچوب‌های کاری بزرگ با قابلیت استفاده چندباره است، استفاده می‌کند. ابزارهای مختلف می‌توانند از مزایای این چارچوب‌های کاری بهره‌مند شوند. از جمله Eclipse حاوی چارچوب‌های کاری ذیل می‌باشد:

چارچوب کاری برای تهیه واسط کاربری پایدار، چارچوب کاری برای مدیریت و دسترسی بر اجزای برنامه‌نویسی شده از طریق سیستم فایل، چارچوب کاری برای

¹ IDE

² Framework

ویرایشگرهای متن، برنامه‌سازها، نشست‌های اشکال‌زدایی^۳، برنامه‌نویسی گروهی، کنترل ویرایش^۴ و غیره. سهولت یکپارچه‌سازی کد های جدید با سکوی موجود به عنوان واحدهای افزونه^۵ کاربردی، از مهمترین ویژگی‌های معماری Eclipse است. البته پشتیبانی از کد افزونه تنها مختص Eclipse نیست، چون همه IDEهای مطرح امروزی، افزونه‌ها را پشتیبانی می‌کنند ولی ویژگی یکتای Eclipse این است که در آن بین چهارچوب کاری‌های پایه با چهارچوب کاری‌های تهیه شده توسط برنامه‌نویس های مختلف به منظور توسعه امکانات، هیچ تفاوتی وجود ندارد.

محیط مجتمع توسعه Carbide.c++

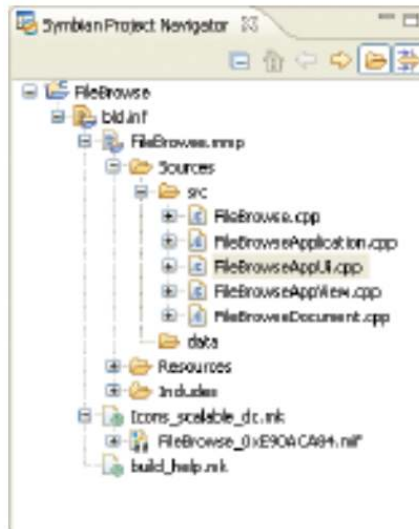
Carbide.c++ بر مبنای Eclipse ویرایش ۳.۳ بنا شده است که با داشتن افزونه‌های مضاعف، توانایی کار کردن با فایل‌های Symbian C++ و همچنین ساخت پروژه‌های Symbian را دارا می‌باشد. از آنجا که Carbide.c++ بر پایه Eclipse است، کاربران می‌توانند با نصب افزونه‌های رایگان و یا تجاری دیگر، آن را بیشتر سفارشی کنند. از جمله این افزونه‌ها می‌توان به سیستم کنترل ویرایش، سیستم مدل‌سازی UML و غیره اشاره کرد.

نصب Carbide.c++ راحت است زیرا خودش دارای برنامه نصب‌کننده بر روی سیستم عامل ویندوز می‌باشد. بعد از نصب و اجرای برنامه، برنامه‌نویس با محیطی روبرو خواهد شد که عیناً همانند محیط استاندارد Eclipse است؛ در شکل بعدی آن را می‌بینید. برنامه‌نویس‌هایی که با محیط Eclipse و یا دیگر محصولات برپایه آن آشنا هستند، به سهولت استفاده از Carbide.c++ پی خواهند برد زیرا افزونه‌های Carbide.c++ از اصول طراحی Eclipse و چارچوب کاری آن پیروی می‌کنند.

³ Debugging Sessions

⁴ Version Control

⁵ plugin



از آنجا که Carbide.c++ برای کار با برنامه‌های Symbian C++ طراحی شده است، به طور پیش‌فرض، سمت راست محیط IDE، نمای ^۶ c/c++ project را نشان می‌دهد که برای نمایش فایل‌های کد، سرآیند^۷ و منبع که تشکیل دهنده یک پروژه معمولی Symbian هستند، طراحی شده است.

همانطور که تصویر بعد نشان می‌دهد، زیر این نما، نمای Symbian project navigator قرار دارد که ساختار پروژه Symbian C++ را همان گونه‌ای که در فایل MMP آدرس داده شده و در فایل bld.inf تعریف شده است، نشان می‌دهد.

برنامه‌نویس‌هایی که با ساختن دستی این گونه فایل‌ها به خوبی CodeWarrior برای سیستم عامل Symbian، آشنایی دارند، به سهولت فهم این نما پی خواهند برد.

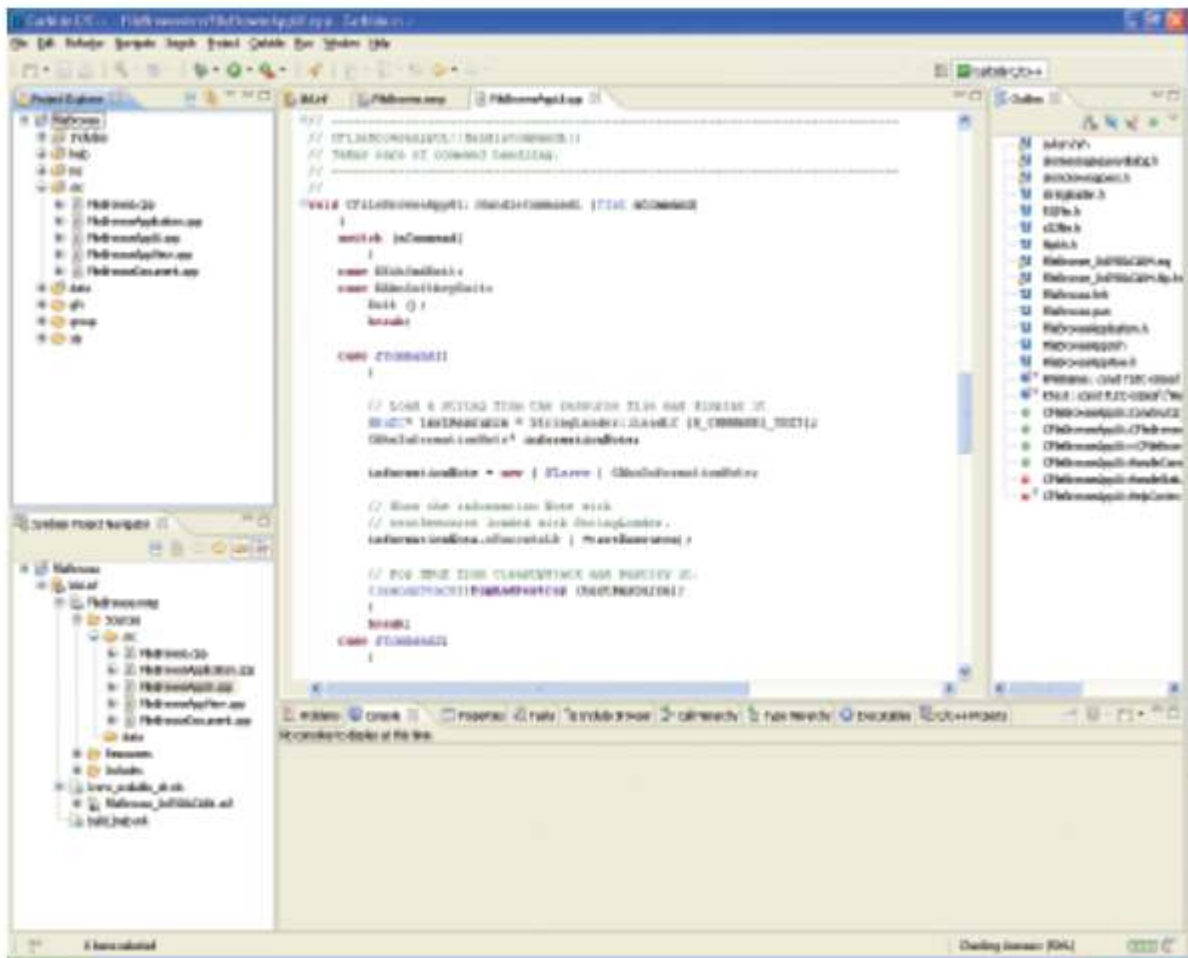
در مرکز محیط IDE، یک ویرایشگر کد C و C++ وجود دارد. برنامه‌نویس می‌تواند چندین پنجره ویرایشگر را به صورت همزمان باز کند و با کلیک کردن روی دکمه زبانه^۸ بالای پنجره مورد نظر و یا با زدن کلید میانبر تعریف شده، به آن پنجره برود. بالاخره در سمت راست IDE به طور پیش‌فرض، دو نما باز شده است: نمای Outline و نمای Help.

⁶ View

⁷ Header

⁸ Tab

- نمای Outline فایل‌های سرآیند استفاده شده و توابع تعریف شده در فایل کد C++ انتخاب شده در ویرایشگر را نشان می‌دهد.
- نمای Help اطلاعات کمکی در مورد عنصر انتخاب شده توسط نشانگر ماوس در محیط IDE را نمایش می‌دهد. همچنین امکان خواندن هر موضوع کمکی مربوط به Carbide.c++ و ویژگی‌های آن را فراهم می‌کند.



مجموعه این پنجره‌ها که به طور خودکار موقع فراخوانی⁹ یک پروژه Symbian C++ ظاهر می‌شوند، در محیط با یکدیگر ترکیب شده و دورنمای¹⁰ C/C++ Carbide.c++ را شکل می‌دهند که این دورنما به شکل یک آیکن قابل کلیک در گوشه سمت راست بالای محیط دیده می‌شود. در شکل بعد این آیکن را می‌بینید Carbide.c++ حاوی

⁹ Load

¹⁰ Perspective

دورنماهای مختلفی است که از قبل برای مراحل مختلف برنامه‌نویسی طراحی شده‌اند. به عنوان مثال، یک دورنمای debug موجود است که مجموعه متفاوت دیگری از پنجره‌ها و نماها را در محیط باز می‌کند که مناسب برای نشست‌های اشکالزدایی است.



البته توجه داشته باشید که برنامه‌نویس می‌تواند به دلخواه خود پنجره‌ها و نماهای مربوط به یک دورنما را تغییر دهد و Carbide.c++ هم موقع اجرای دوباره، این تنظیمات را به یاد می‌آورد.

دورنمای فعلی را می‌توان توسط `Window > Open Perspective` تغییر داد. آرایش دورنمای فعلی، ذخیره شده و دورنمای جدید باز می‌شود. نماهایی که در حال حاضر در معرض دید نیستند را می‌توان توسط `Window > Show View` باز کرد. نمای مورد نظر در موقعیت قبلی و یا در صورتی که تاکنون در دورنمای فعلی باز نشده است، در موقعیت پیش‌فرض باز می‌شود.

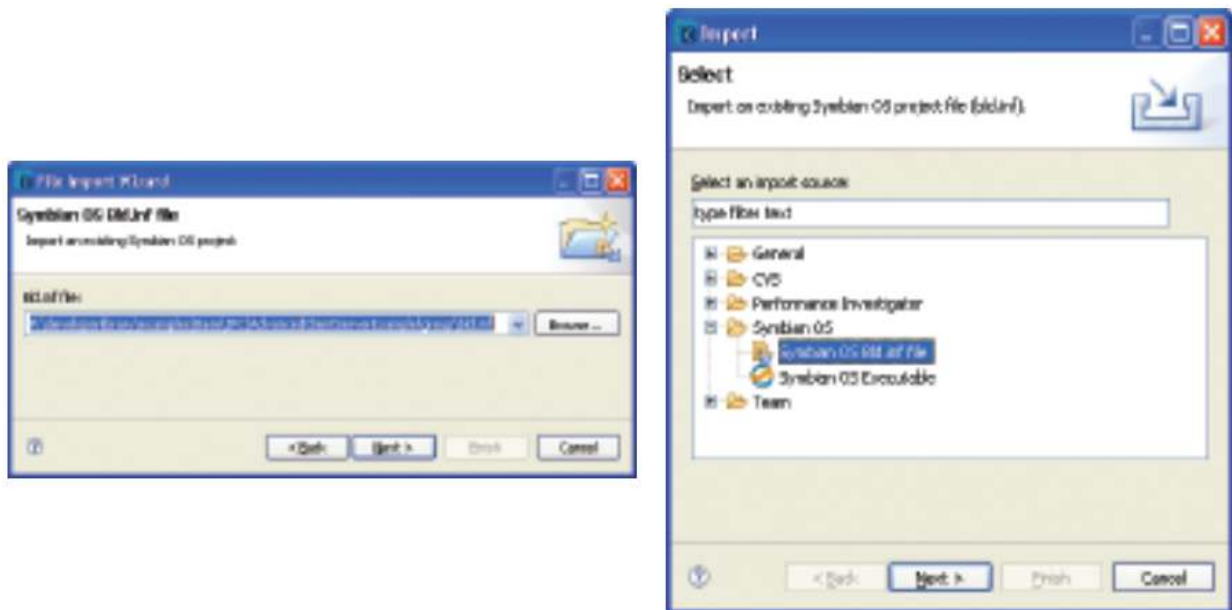
وارد کردن^{۱۱} پروژه‌های موجود Symbian C++

بعضی برنامه‌نویس‌های Symbian C++، از قبل پروژه‌های Symbian مختلفی را در کامپیوترشان دارند که با استفاده از CodeWarrior Symbian و یا مجموعه ابزارهای خط فرمانی Symbian آنها را می‌ساختند. Carbide.c++ حاوی ابزار کمکی است که برنامه‌نویس می‌تواند به سادگی این پروژه‌ها را بدون اینکه نیاز به ساخت یک پروژه جدید و اضافه کردن دستی تمام فایل‌ها به آن باشد (البته این کار هم امکان‌پذیر است) وارد Carbide کند. Eclipse امکان ساخت جادوگرها^{۱۲} به منظور وارد و صادر کردن اطلاعات از محیط را پشتیبانی می‌کند. در Carbide.c++ با استفاده از این امکان، جادوگر وارد کردن فایل `bld.inf` تهیه شده است. همانطور که تصویر هم نشان

¹¹ Import

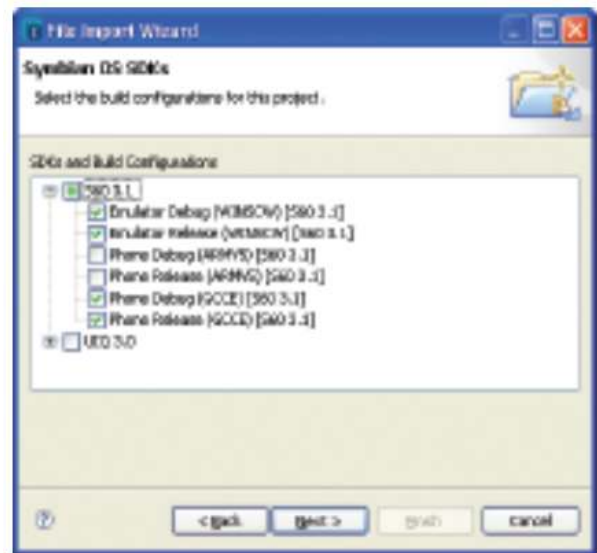
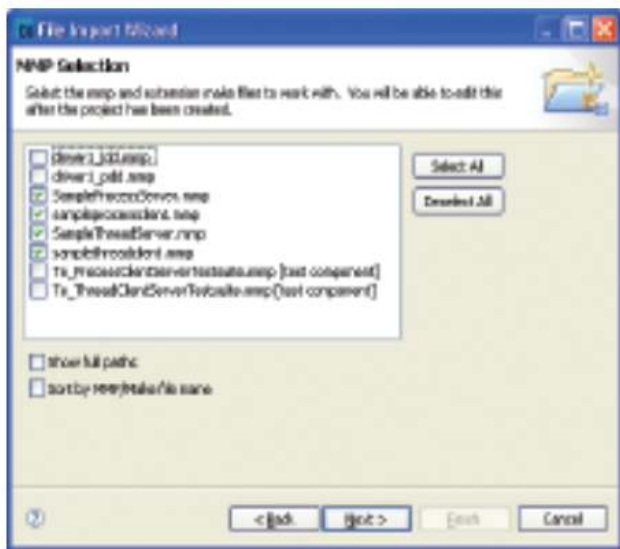
¹² Wizard

می‌دهد، برنامه‌نویس توسط منوی `File > Import` یک پنجره `import` اجرا می‌کند که دارای انتخاب‌های مختلفی است؛ در این میان گزینه‌ای برای وارد کردن فایل `bld.inf` وجود دارد که در تصویر اول زیر نشان داده شده است. بعد از انتخاب این گزینه، از برنامه‌نویس خواسته می‌شود که فایل `bld.inf` را از حافظه کامپیوتر انتخاب کند که در شکل دوم ذیل می‌بینید.



بعد از اتمام این گام، `Carbide.c++` تمام `Symbian SDK`‌های نصب شده روی کامپیوتر میزبان را نشان می‌دهد و برنامه‌نویس موظف است یک `SDK` متناسب با پروژه را انتخاب کند که در تصویر بعد هم می‌بینید. در گام بعد، برنامه‌نویس می‌تواند انتخاب کند که کدام یک از فایل‌های `MMP` ارجاع داده شده در فایل `bld.inf`، به منظور تجزیه^{۱۳} کردن فایل(های) `MMP` مناسب به کار رود.

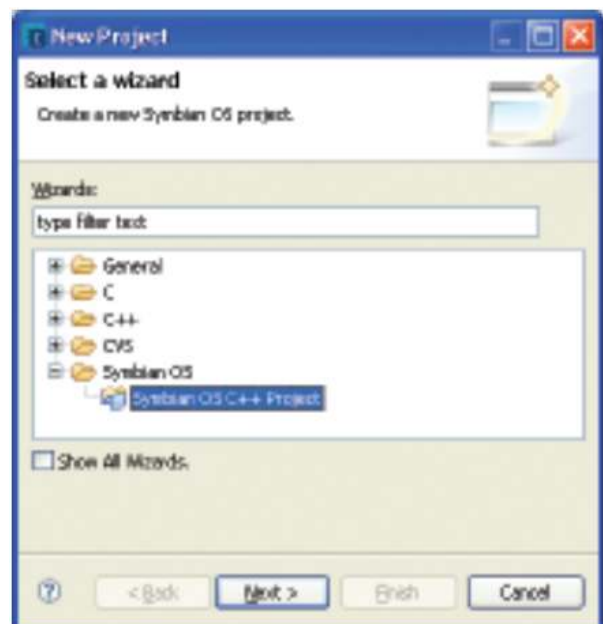
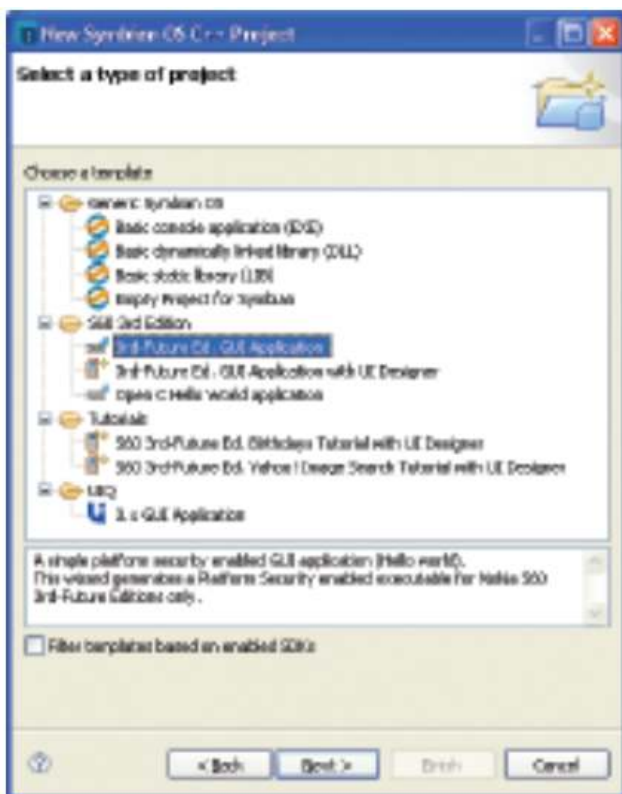
¹³ Parse



در نهایت، برنامه‌نویس می‌تواند نام پروژه و دایرکتوری ریشه را انتخاب کند. در پایان اجرای این جادوگر، یک پروژه Symbian C++ در Carbide.c++ باز شده و آماده برنامه‌نویسی می‌باشد.

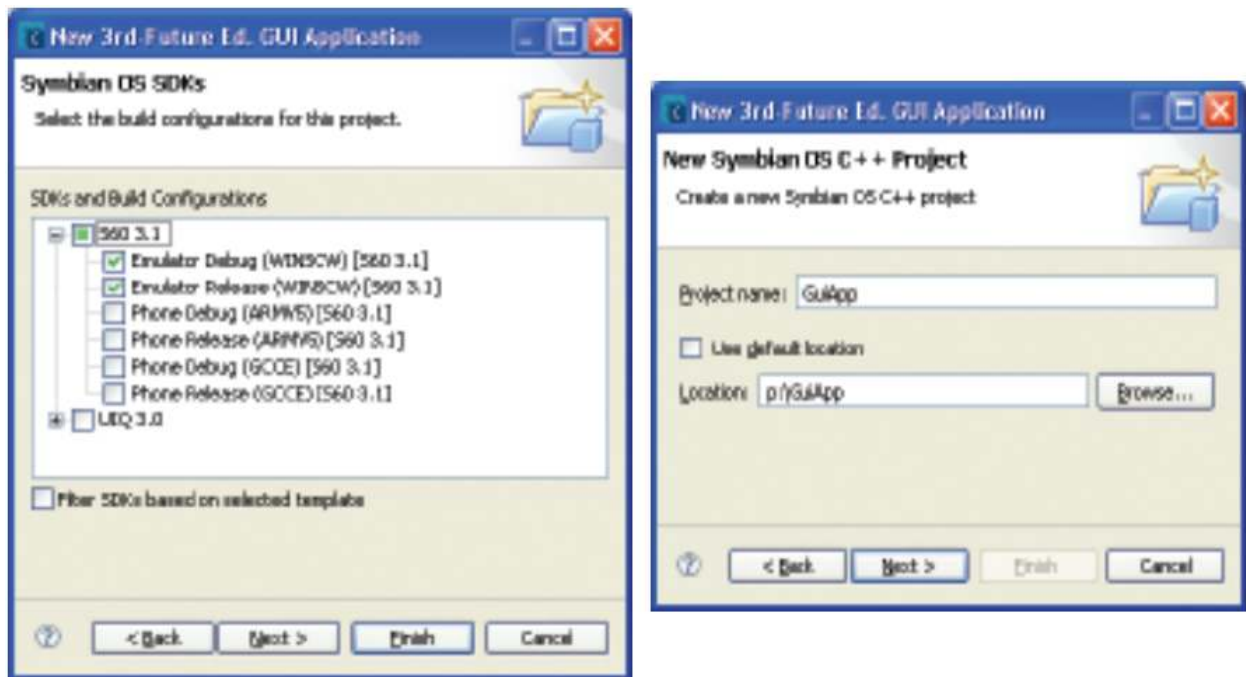
ایجاد یک پروژه Symbian C++ جدید

Carbide.c++ حاوی جادوگری است که امکان ساخت یک پروژه جدید را فراهم می‌کند. این جادوگر را می‌توان از طریق منوی File > New Project اجرا کرد. در تصاویر زیر هم می‌بینید:



Carbide.c++ حاوی الگو^{۱۴}های اختصاصی برای انواع پروژه‌های مختلف است (در صورت بالا نشان داده شده است). استفاده کردن از یکی از این الگوها، یک پروژه جدید ساخته و در همان لحظه، بسته به نوع پروژه انتخاب شده، تعدادی فایل و مقداری کد هم تولید.

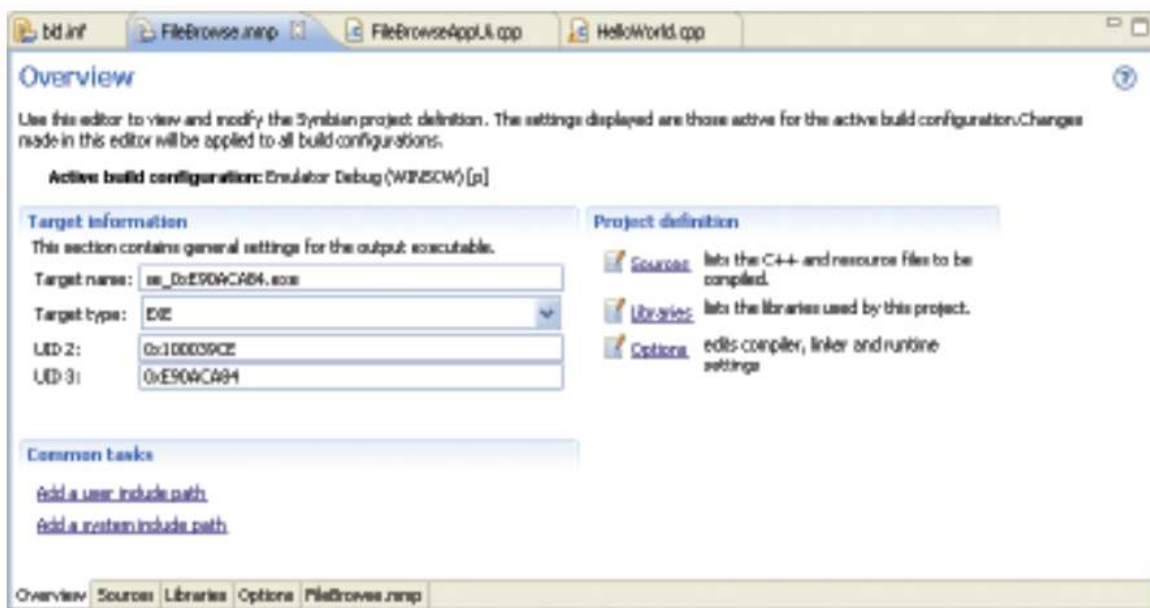
از آنجایی که Carbide.c++ به طور خورکار، SDKهای نصب شده روی کامپیوتر میزبان را شناسایی می‌کند، الگوهای پروژه را می‌توان براساس SDKهای شناسایی شده، فیلتر کرد.



مرحله بعدی این جادوگر، مسیر ذخیره پروژه و سپس تنظیمات ساختی که باید فعال شوند را می‌پرسد، که در تصاویر بالا نشان داده شده است. مرحله آخر جادوگر به برنامه‌نویس اجازه می‌دهد که اطلاعاتی مانند UID برنامه (گرچه این مورد را بعداً هم می‌توان تغییر داد) و پوشه‌هایی که برای ذخیره فایل‌های مختلف استفاده خواهند شد را مشخص کند. پس از این مرحله، اجرای جادوگر کامل می‌شود و نتیجه آن، یک پروژه جدید با تعدادی فایل اولیه خواهد بود که بعداً می‌توانند توسعه یابند.

امکان تنظیم مسیره‌های مربوط به پروژه، سفارشی‌سازی^{۱۶} متغیرهای محیطی مربوط به پروسه ساخت، تولید خودکار فایل نصب شدنی SIS با استفاده از فایل PKG و ساخت دو فایل Certificate/Key مربوط به Symbian Signed، برای برنامه‌هایی که گوشی-های دارای Symbian ویرایش 9.x را هدف قرار داده اند، فراهم می‌آورد. قسمت Carbide Project Settings، به برنامه‌نویس، امکان ساخت اجزای پروژه (فایل‌های MMP)، مبنی بر فایل bld.inf وارد شده را می‌دهد.

Carbide.c++ امکان تغییر فایل bld.inf پروژه و فایل‌های MMP مربوطه، توسط یک ویرایشگر گرافیکی را نیز فراهم می‌آورد، که این ویرایشگر به برنامه‌نویس امکان تغییر ویژگی‌های پروژه، بدون اینکه با ساختار نحو^{۱۷} و پارامترهای این فایل‌ها آشنایی باشد، می‌دهد. این ویرایشگرها را می‌توان با باز کردن فایل‌های MMP از نمای Symbian Project Navigator یا C/C++ Project اجرا کرد. ویرایشگر MMP در تصویر زیر نشان داده شده است:



بنابراین برنامه‌نویس می‌تواند تمام اطلاعات لازم را توسط ویرایشگرهای گرافیکی تنظیم کند. روش دیگر، کلیک کردن روی زبانه bld.inf یا فایل MMP در پایین پنجره

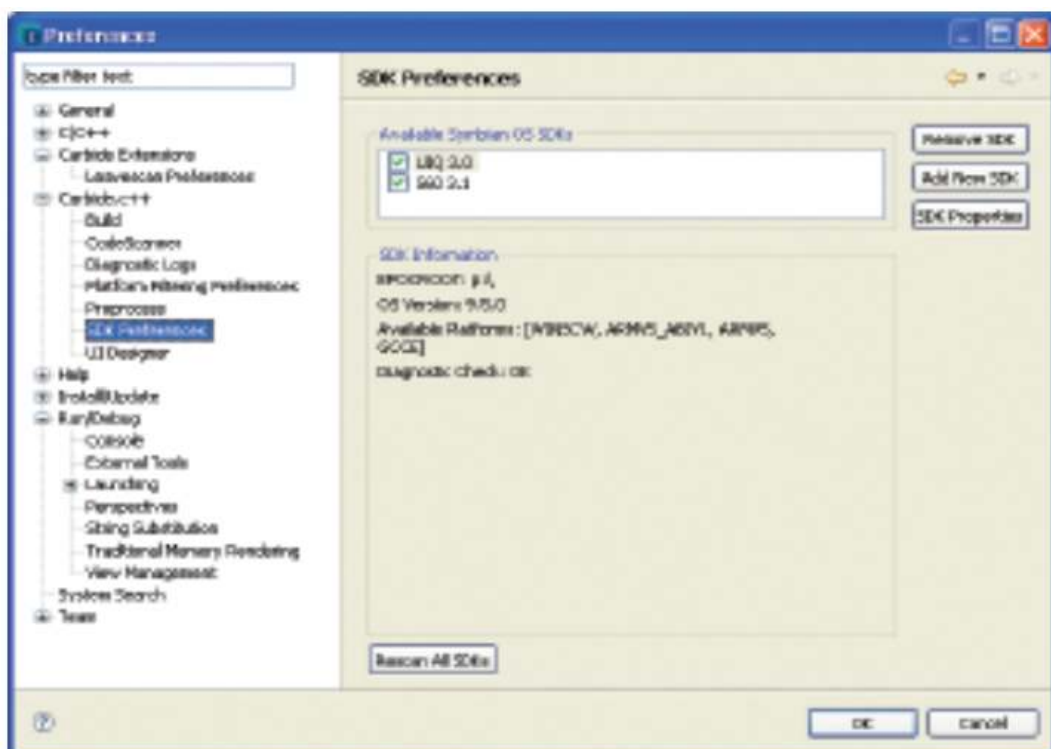
¹⁶ Customization

¹⁷ Syntax

و ویرایش دستی این فایل‌ها به کمک ویرایشگر متنی است. توجه داشته باشید که Carbide.c++ همیشه این فایل‌ها را در هماهنگی با تغییراتی که کاربر در بیرون ویرایشگرها انجام می‌دهد، نگه می‌دارد. از جمله تغییرات بیرونی می‌توان به افزودن فایل‌ها کد به پروژه توسط رابط کاربری اصلی Carbide.c++ اشاره کرد. همچنین ویرایش دستی این فایل‌ها، باعث به روز شدن خودکار اطلاعات نمایش داده شده در ویرایشگرهای گرافیکی می‌شود.

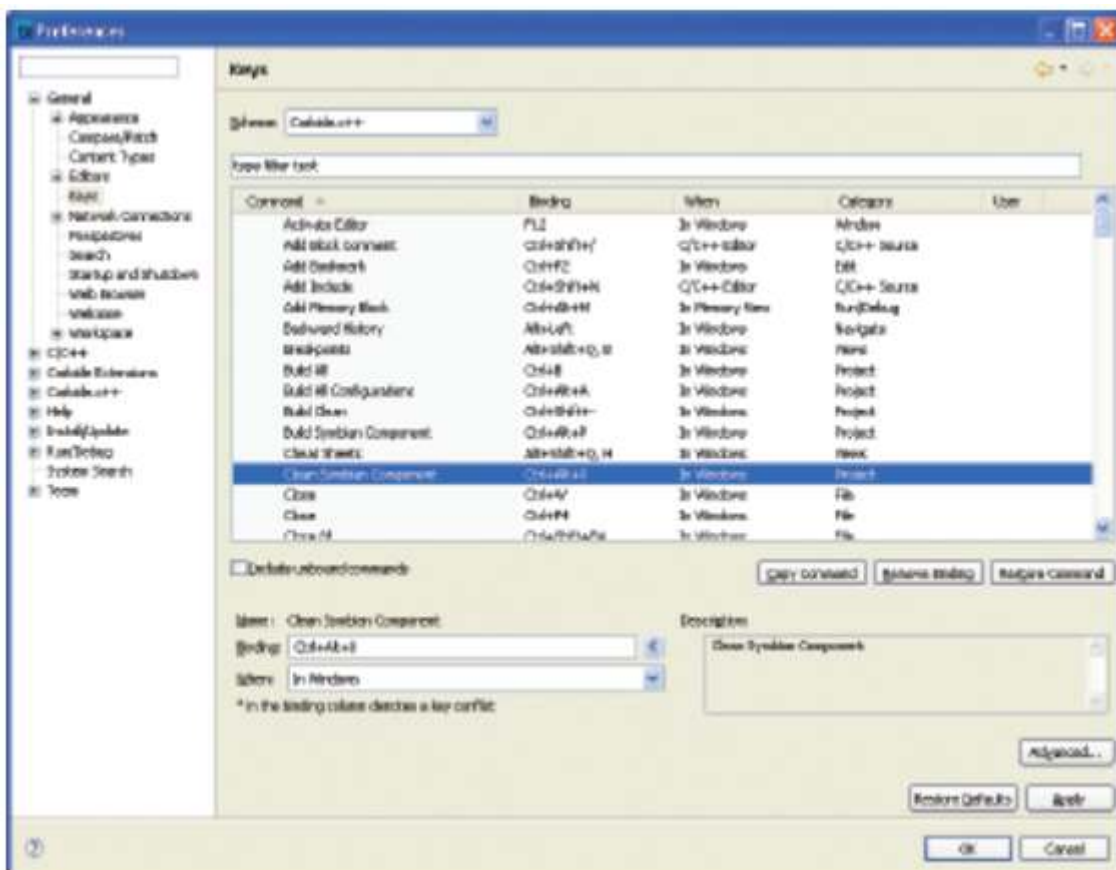
تنظیمات سراسری

علاوه بر تنظیمات مخصوص پروژه، Carbide.c++ به کاربر امکان مشخص کردن تنظیمات سراسری که رفتار محیط و همه پروژه‌های Symbian C++ را تحت تأثیر قرار خواهند داد را نیز می‌دهد. این تنظیمات از پنجره Preferences قابل دسترسی هستند که از طریق منوی Window > Preferences می‌توان آن را اجرا کرد. همان طور که در تصویر زیر هم نشان داده شده است، در سمت چپ این پنجره، برنامه‌نویس می‌تواند گروه تنظیماتی که باید تغییر داده شوند را انتخاب کند، مثلاً تنظیمات عمومی، تنظیمات ویرایشگر و غیره. در حالی که در سمت راست، تنظیمات موجود برای گروه انتخاب شده، ظاهر می‌شوند.



تنظیمات مخصوص برنامه‌نویسی Symbian C++، در درخت Carbide.c++ قرار دارند. درخت SDK Preferences، به برنامه‌نویس امکان مدیریت Symbian SDK‌های نصب شده و تغییر ویژگی‌های آنها را می‌دهد. مورد Build، حاوی انتخاب‌هایی برای سفارشی‌سازی رفتار ساخت پیش‌فرض پروژه‌ها، به عنوان مثال این که کد آزمایشی ساخته شود یا نه، است.

Carbide.c++ برای کارهای معمول که برعهده محیط هستند، میانبرهای صفحه‌کلید تعریف کرده است. مورد Keys در داخل General Preferences، امکان تغییر این میانبرها را فراهم می‌آورد. در اینجا می‌توان یک مورد خاص را تغییر داد و یا اینکه یک طرح^{۱۸} از پیش تعریف شده را انتخاب کرد. مثلاً طرح‌های CodeWarrior و Visual Studio.



¹⁸ Scheme

پیمایش و ویرایش کد

مقدار قابل توجهی از زمان برنامه‌نویس موقع استفاده از یک IDE، صرف دیدن و تغییر کد می‌شود. Carbide.c++ دارای تعداد زیادی مشخصه مفید برای ویرایشگر است، به عنوان مثال:

جستجو

منوی Search... > Search امکان جستجوی عبارات را در فایل‌ها فراهم می‌کند. جستجو را می‌توان به فایل‌های با نوع خاص و یا به زیرمجموعه‌ای از فایل‌های تعریف شده توسط محیط کار، پروژه و یا پوشه محدود کرد. نتایج را می‌توان در نمای Search و یا System Search مشاهده کرد.

میانبرهای صفحه کلید

بسیاری از مشخصه‌های ویرایشی و یا پیمایشی Carbide.c++، دارای میانبرهای صفحه کلید وابسته به خود هستند. به عنوان مثال، Alt-Left و یا Alt-Right میان فایل‌های باز شده در ویرایشگر، جلو و عقب می‌روند، Ctrl+/ متن انتخاب شده را کامنت می‌کند و یا از حالت کامنت خارج می‌کند، Ctrl+Alt+C پروژه را کامپایل می‌کند و Ctrl+Alt+H یک سلسله‌مراتب صدا زدن را برای مشخصه زیر نشانگر ماوس باز می‌کند. میانبرهای موجود برای یک دورنما را می‌توان در Window > Preferences مشاهده کرد.

سلسله‌مراتب نوع^{۱۹} و صدا زدن^{۲۰}

زدن F4، سلسله‌مراتب نوع را برای کلاسی که زیر نشانگر ماوس است، باز می‌کند. این کار، سلسله‌مراتب وراثت و توابع تعریف شده در هر سطح را نشان می‌دهد. همان طور که در بالا هم توضیح داده شد، زدن Ctrl+Alt+H، سلسله‌مراتب صدا زدن را برای مشخصه زیر نشانگر باز می‌کند و این کار تمام مکان‌های داخل پروژه که در آنها تابع انتخاب شده صدا زده می‌شود را فهرست می‌کند که برای عیب‌یابی کمک مفیدی است.

¹⁹ Type Hierarchy

²⁰ Call Hierarchy

الگوهای کد

تایپ کردن lit و فشردن Ctrl+Space، آن را به ("LIT")_ تغییر خواهد داد. تعداد زیاد دیگری از این الگوهای کد موجود است که در Window > Preferences > C/C++ > Editor > Templates فهرست شده‌اند، به عنوان مثال، الگوهای استفاده شده برای Cleanup Stack و ساختارهای کنترلی مثل حلقه‌های for و while. امکان تعریف الگوهای جدید نیز در اینجا فراهم است.

تاریخچه محلی

کلیک راست روی یک فایل در Project Explorer و انتخاب Team > Show Local History، فهرستی از ویرایش‌های قدیمی فایل را که پس از هر تغییری در فایل ذخیره شده‌اند، نمایش می‌دهد. از این نما می‌توان ویرایش فعلی فایل را با موارد قبلی مقایسه کرد و در صورت نیاز آن را برگرداند.

وظایف^{۲۱}

نمای Tasks روشی برای مدیریت وظایف برنامه‌نویسی فراهم می‌کند. وظایف را می‌توان مستقیماً به نمای Tasks اضافه کرد و یا اینکه کامنت‌های حاوی کلمه TODO را به کد اضافه کرد و متن کامنت به طور خودکار به عنوان یک وظیفه جدید به نمای Tasks اضافه خواهد شد. این وظایف ایجاد شده توسط کامنت‌ها، یک راه سریع برای دسترسی به کد مربوطه از نمای Tasks را فراهم می‌کنند.

ویرایش در قسمت‌های مختلف صفحه

با درگ کردن برگه ویرایشگر یک فایل باز به قسمت دیگری از محیط، یک پنجره ویرایشگر دیگر را می‌توان ایجاد کرد. با تکرار این عمل، برنامه‌نویس می‌تواند هر تعداد پنجره ویرایشگر لازم را باز کند تا بتواند در یک زمان، بیشتر از یک فایل کد و همچنین قسمت‌های مختلف یک فایل را ببیند.

²¹ Tasks

ساخت و اجرای یک برنامه

محیط Carbide.c++ امکان باز کردن تعداد زیادی پروژه مختلف به صورت همزمان را فراهم می آورد. در نتیجه به کاربر، امکان ساخت همه پروژه‌های باز و یا تعدادی پروژه انتخاب شده را می‌دهد. مراحل ساخت یک پروژه در Carbide.c++ به صورت زیر است:

- پروژه‌ای که باید ساخته شود را انتخاب کنید (در نمای C/C++ Project روی پروژه کلیک کنید)

- نحوه ساخت فعال مورد نظر را با استفاده از منوی `Project > Active Build Configuration` انتخاب کنید.

- پروژه را با انتخاب منوی `Project > Build Project` بسازید.

Carbide.c++ پروژه را با کامپایل کردن و لینک کردن^{۲۲}، می‌سازد و به انتخاب شما فایل SIS را با توجه به نحوه ساخت فعال مورد نظر تولید می‌کند. اگر لازم باشد، می‌توان فایل SIS را در پایان ساخت، با انتخاب فایل PKG مناسب، تولید کرد. این کار فایل SIS را تولید کرده و به دلخواه شما آن را امضا^{۲۳} می‌کند. خطاها و اخطارها در پایین محیط در نمای Problems نشان داده می‌شوند و برنامه‌نویس می‌تواند روی پیام‌ها کلیک کرده و به طور خودکار متنی که خطا یا اخطار را تولید کرده است را ببیند. کد تولید کننده خطا یا اخطار، در ویرایشگر زیرش خط کشیده می‌شود و آیکن مناسب خطا و یا اخطار، علامت‌گذاری می‌شود. نمای Console، دستوراتی که Carbide.c++ برای ساخت پروژه اجرا می‌کند را به همراه پیام‌های خطا و اخطاری که توسط کامپایلر، لینکر و یا دیگر ابزارهای ساخت تولید شده‌اند، نمایش می‌دهد.

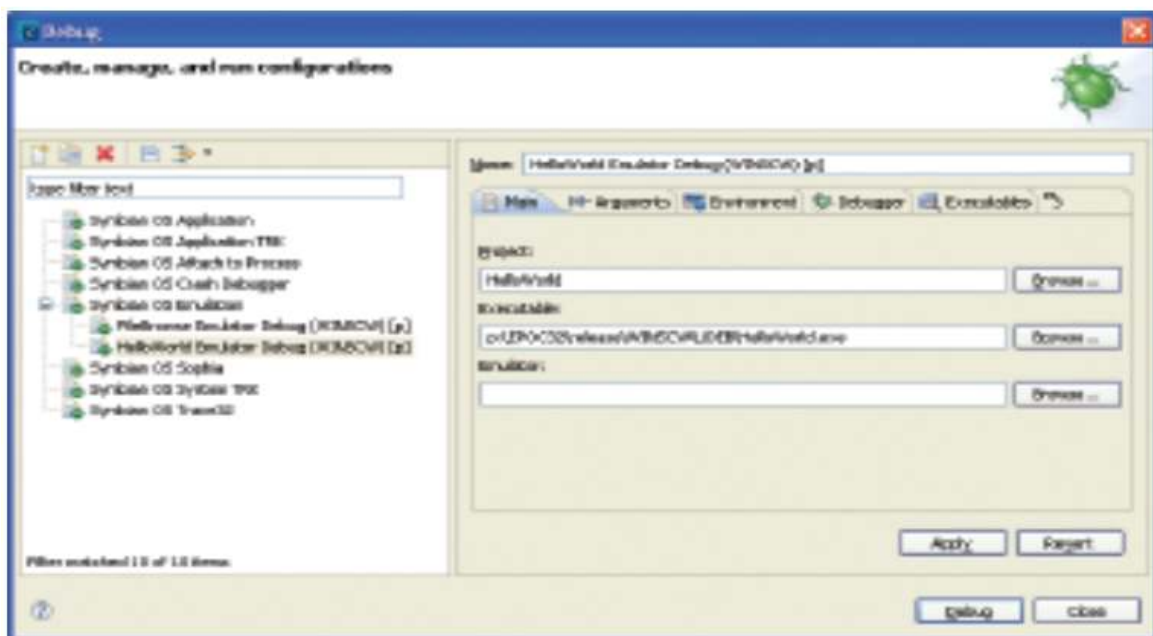
پس از اینکه عملیات ساخت تکمیل شد و تمام خطاها برطرف شدند، برنامه را می‌توان در شبیه‌ساز Symbian و یا گوشی موبایل، با توجه به نحوه ساخت، اجرا کرد. Carbide.c++ به طور خودکار یک محیط اجرا برای پروژه کامپایل شده ایجاد می‌کند و

²² Linking

²³ Signing

بنابراین برنامه‌نویس می‌تواند با انتخاب منوی Run > Run (و یا با زدن Ctrl+F11) برنامه را اجرا کند، محیط اشکال‌زدایی را هم می‌توان با انتخاب منوی Run > Debug (و یا با زدن F11) اجرا کرد. تنظیمات این محیط‌ها را می‌توان تغییر داد و همچنین محیط‌های اجرای جدید را به کمک منوی Run > Open Run Dialog... و یا محیط‌های اشکال‌زدایی جدید را به کمک منوی Debug Configuration و یا انتخاب Run > Open Debug Dialog... ایجاد کرد.

همانطور که تصویر زیر نشان می‌دهد، در پنجره Run Configuration، برنامه‌نویس می‌تواند یک محیط اجرایی جدید که هدفش شبیه‌ساز Symbian و یا گوشی موبایل است را ایجاد کند و تنظیمات مختلف مثل دستورات خط فرمان برای برنامه، درگاه ارتباطی بلوتوث یا USB برای Carbide.c++ به منظور ارسال فایل SIS به گوشی برای اجرا و یا اشکال‌زدایی برنامه روی موبایل، مکان فایل‌های کد و غیره را مشخص کند. با این روش، محیط‌های مختلفی را می‌توان برای پروژها و هدف‌های خاص ساخت و در نشست‌های^{۲۴} متوالی Carbide.c++ استفاده کرد.



نشست‌های اشکال‌زدایی^{۲۵}

نشست‌های اشکال‌زدایی را می‌توان از طریق یکی از محیط‌های اشکال‌زدایی که در قسمت قبل توضیح داده شد، اجرا کرد. یک محیط اشکال‌زدایی به برنامه‌نویس اجازه می‌دهد نحوه اشکال‌زدایی برنامه را سفارشی‌سازی کند، به عنوان مثال، استثناهایی^{۲۶} که Carbide.c++ باید بگیرد و یا اینکه آیا یک نقطه توقف برنامه باید در تابع E32Main برنامه قرار داده شود یا نه. هر زمان که برنامه توسط دیباگر اجرا شود (چه در شبیه‌ساز Symbian و چه در گوشی موبایل)، Carbide.c++ به دورنمای Debug وارد می‌شود که در تصویر بعدی هم نشان داده شده است. Carbide.c++ از تمام ویژگی‌های اشکال‌زدایی که در محیط‌های معمول دیگر یافت می‌شوند، پشتیبانی می‌کند. ریسه^{۲۷}‌های تشکیل دهنده برنامه در حال اجرا، در نمای Debug قابل دید هستند. برنامه‌نویس می‌تواند یک ریسه و یا پروسه را به منظور دریافت اطلاعات از پشته^{۲۸} ریسه و یا متغیرهای محلی‌اش، در این نما انتخاب کرده و آزمایش کند.

برگه Breakpoints به برنامه‌نویس امکان ساخت نقاط توقف برنامه^{۲۹} و نقاط نظارت^{۳۰} را می‌دهد. نقاط توقف برنامه را می‌توان گروه‌بندی کرد و از آن به صورت انفرادی و یا گروهی، به عنوان مثال طبق موجودیت در فایل یا پروژه‌ای که برای آن ساخته شده‌اند، استفاده کرد. موقع اشکال‌زدایی توسط شبیه‌ساز، می‌توان با قرار دادن نقاط نظارت، وضعیت یک متغیر خاص را نشان مشخص نمود، این کار برای اشکال‌زدایی اشکالات خرابی حافظه مفید است، زیرا Carbide.c++، ریسه‌هایی که متغیرهای تحت نظارت را تغییر می‌دهند را موقتاً غیرفعال می‌کند.

²⁵ Debugging Sessions

²⁶ Exception

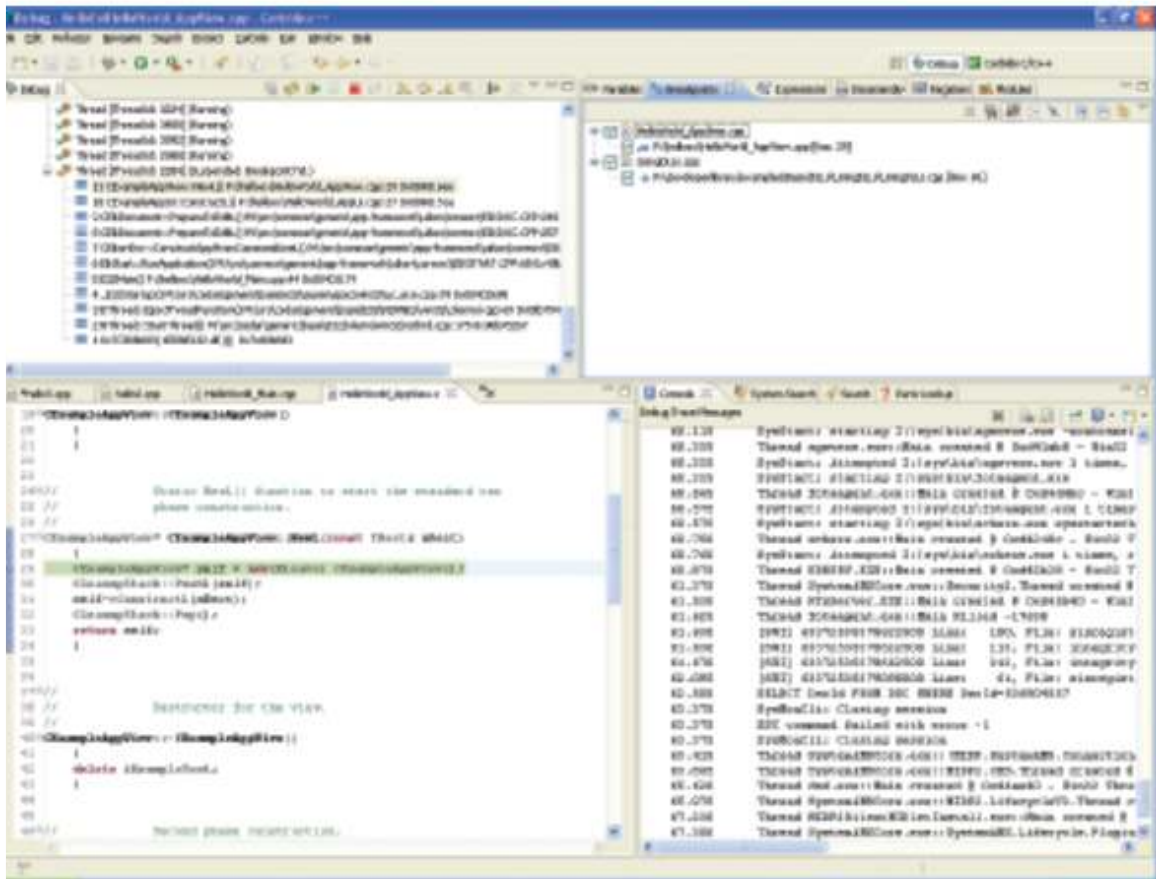
²⁷ Thread

²⁸ Stack

²⁹ Breakpoints

³⁰ Watchpoints

مقادیر متغیرهای محلی و سراسری را می‌توان در نمای Variables آزمایش کرد که این نما قادر به تفسیر اطلاعات مخصوص Symbian، مثل توصیف‌گرها^{۳۱} است و می‌تواند مقادیر آنها را نشان دهد. مقادیر ثابت‌ها^{۳۲} و آدرس‌های حافظه را می‌توان به ترتیب در نماهای Registers و Memory مشاهده کرد و تغییر داد. از این نماها می‌توان برای تحلیل و تغییر وضعیت یک برنامه استفاده کرد.



در نهایت، کاربر می‌تواند داخل توابع شود و یا از روی آنها عبور کند، برنامه را تا نقطه توقف بعدی اجرا کند و بالاخره در حالی که مشغول دیدن کد و خط جاری اجرا شده در ویرایشگر است، نشست اشکال‌زدایی را خاتمه داده و یا دوباره شروع کند.

³¹ Descriptor

³² Register

ویژگی‌های جدید در Carbide.c++ ویرایش ۱.۳

تعدادی ویژگی جدید در Carbide.c++ ویرایش ۱.۳ وجود دارد که تاکنون توضیح داده نشده‌اند. آنها عبارتند از:

- ویژگی‌های ساخت سیستمی انتخابی که باعث افزایش سرعت ساخت پروژه‌ها می‌شوند. به این گزینه‌ها می‌توان از طریق `Preferences > Carbide.c++ > Build` دسترسی داشت. اینها به برنامه‌نویس، امکان هدایت Carbide.c++ برای مدیریت ساخت مجزای وابستگی‌ها^{۳۳} و اجرای پروسه‌های ساخت به طور همزمان در رایانه‌های چندهسته‌ای را می‌دهند.
- ویژگی جستجوی سیستمی از طریق منوی `Search > System...` در دسترس است که امکان جستجو در پوشه‌های دلخواه در رایانه را می‌دهد و فقط محدود به فایل‌های محیط کار فعلی نیست.
- فایل‌های کد پروژه را می‌توان با راست‌کلیک کردن روی فایل در نمای Symbian Project Navigator و انتخاب `Preprocess`، پیش‌پردازش کرد. پیش‌پردازشگر C روی تمام فایل اجرا شده و خروجی را در نمای `Console` نمایش خواهد داد.
- موقع شروع یک محیط کار جدید، تنظیمات محیط کار فعلی را می‌توان به محیط کار جدید، کپی کرد. این گزینه را می‌توان از طریق انتخاب `File > Switch Workspace > Other...` و انتخاب مواردی که باید کپی شوند، اجرا کرد.
- نمای `Platform Security` تحلیلی از قابلیت‌هایی^{۳۴} که یک فایل MMP نیاز خواهد داشت، ارائه می‌دهد. با کلیک راست روی یک فایل MMP و انتخاب `Run`

³³ Dependencies

³⁴ Capabilities

Capability Scanner on Project MMP، قابلیت‌هایی که برنامه نیاز خواهد داشت، در نمای Platform Security به صورت خلاصه نشان داده خواهند شد.

- همانطور که اکوسیستم Carbide.c++ در حال رشد است، تعداد در حال افزایشی افزونه برای گسترش قابلیت‌های Carbide.c++ نیز در دسترس است. بعضی از این افزونه‌ها مخصوص Symbian هستند و توسط Symbian و یا Nokia ساخته شده‌اند در حالی که بقیه کلی‌تر بوده و مخصوص Eclipse هستند و توسط مراجع دیگر ساخته شده‌اند. در صفحه ویکی این کتابچه در آدرس زیر، اطلاعاتی در مورد افزونه‌های مفید موجود است:

developer.symbian.com/carbide_booklet_wikipage

اطلاعات و یاری

در این کتابچه ما مقداری اطلاعات اساسی در مورد Carbide.c++ شامل چگونگی وارد کردن، ساختن، اشکال‌زدایی و اجرای پروژه‌های Symbian C++ ارائه دادیم. اطلاعات بیشتر در رابطه با تمام ویژگی‌های Carbide.c++ را می‌توان از خود IDE، از طریق منوی Help > Help Contents به دست آورد.

Help همچنین برای Eclipse و ویژگی‌های مخصوص Carbide.c++ بنا شده روی Eclipse و همچنین برای Symbian SDK‌های نصب شده، موجود است. منبع مفیدی از اطلاعات در سایت فروم نوکیا برای Carbide.c++ موجود است (www.forum.nokia.com/carbide) که حاوی مقالات آموزشی، برگه‌های اطلاعات^{۳۵}، سؤالات متداول^{۳۶} و پیوندهایی به فروشگاه الکترونیک فروم نوکیا که می‌توان از آنجا Carbide.c++ را خریداری نمود، می‌باشد.

دیگر منابع مفید اطلاعات، شامل فروم‌های مختلف و مقاله‌های ارسال شده در شبکه توسعه Symbian (developer.symbian.com) و سایت جامعه توسعه UIQ (developer.uiq.com) است. یک صفحه ویکی هم این کتابچه را همراهی می‌کند:

developer.symbian.com/carbide_booklet_wikipage

³⁵ Data sheets

³⁶ Frequently Asked Questions (FAQ)

منابع برنامه نویسی

شبکه توسعه Symbian
developer.symbian.com

خبرنامه شبکه توسعه Symbian
developer.symbian.com/register.action

ابزارها و Symbian SDKها
developer.symbian.com/main/tools

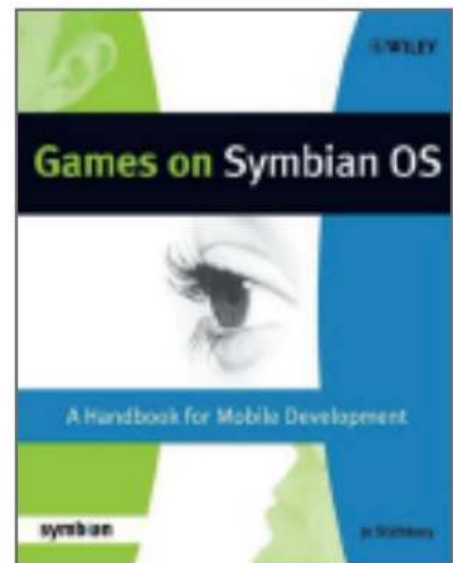
فروم نوکیا
forum.nokia.com

جامعه برنامه نویسی UIQ
developer.uiq.com



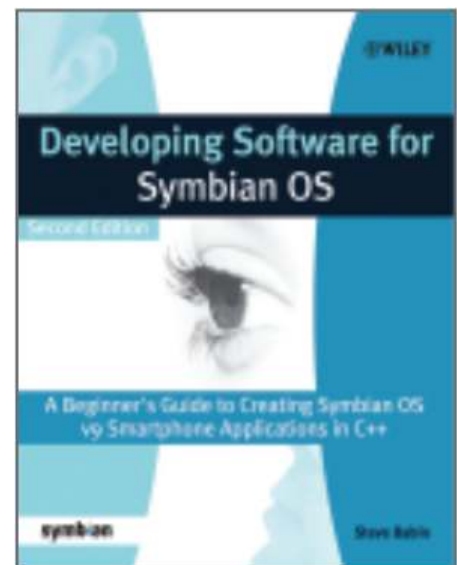
Games on Symbian OS: A Handbook for Mobile Development

این کتاب، از سری Technology از انتشارات Symbian است. این کتاب، جنبه‌های کلیدی بازار بازی‌های موبایل را با تمرکز دقیق بر ساخت بازی‌ها برای گوشی‌های هوشمند بر پایه Symbian ویرایش 9.x، شرح می‌دهد.



Developing Software for Symbian OS, Second Edition

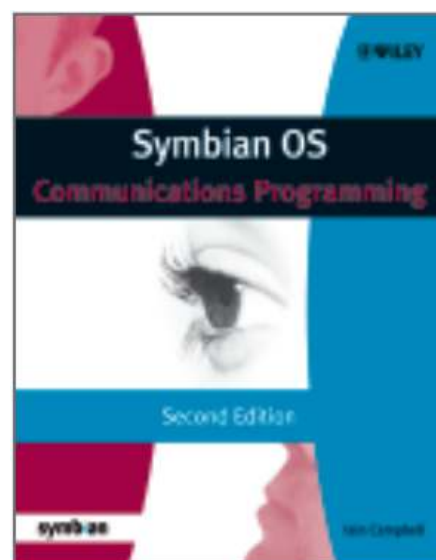
ویرایش دوم کتاب Developing Software for Symbian OS، به برنامه‌نویسان تازه‌کار در Symbian کمک می‌کند تا برنامه‌های مخصوص گوشی‌های هوشمند بسازند. کتاب اصلی برای Symbian ویرایش ۹ به روز شده است و الان دارای یک فصل جدید در مورد ثبت کردن و امنیت و تغییرات کلی برای Symbian ویرایش ۹ و تغییرات در محیط برنامه‌نویسی است.





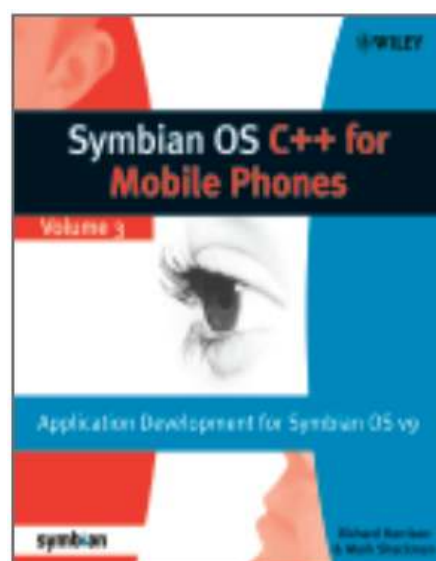
Symbian OS Communications Programming, Second Edition

کتاب Symbian OS Communications Programming – Second Edition که Symbian ۹.۱ و ۹.۲ را هدف گرفته است، شما را با عملکرد عظیم ارتباطات در Symbian آشنا کرده و نحوه انجام کارهای معمول در هر ناحیه را شرح می‌دهد.



Symbian OS C++ for Mobile Phones, Volume 3

این کتاب به شما کمک خواهد کرد تا یک برنامه‌نویس مؤثر Symbian شوید و به شما فهم عمیقی از اصول اساسی که Symbian بر آنها استوار است را می‌دهد.



کتاب‌هایی دیگر از



The Symbian OS Architecture Sourcebook

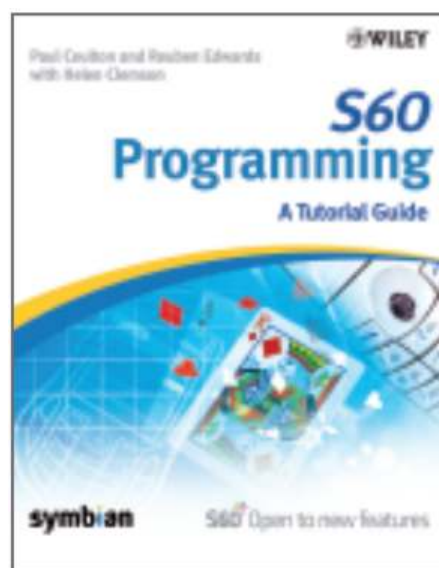
این کتاب یک شامل یک گذر سریع در مورد معماری Symbian است که یک آشنایی با ایده‌های شی‌گرایی (OO) در نرم‌افزار به همراه کاوش دقیق معماری سیستم عامل Symbian را فراهم می‌کند.



S60 Programming

کتاب S60 Programming که کاملاً به روز برای Symbian ویرایش ۹ و سری ۶۰ ویرایش ۳ است، یک پایه اساسی برای توسعه نرم‌افزار برای Symbian است.

این کتاب سودمند بر تجربیات نویسندگان در برنامه‌نویسی و تدریس درس دانشگاهی در زمینه توسعه نرم‌افزارهای Symbian استوار است.



کتاب‌هایی دیگر از



برای تمام برنامه‌نویسان C++ سیمبین

Symbian OS C++ for Mobile Phones - جلد اول

نوشته ریچارد هریسون

Symbian OS C++ for Mobile Phones - جلد دوم

نوشته ریچارد هریسون

Symbian OS Explained

نوشته جو استیکبوری

Symbian OS Internals

نوشته جین سیلس

Symbian OS Platform Security

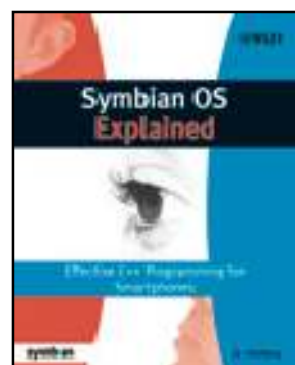
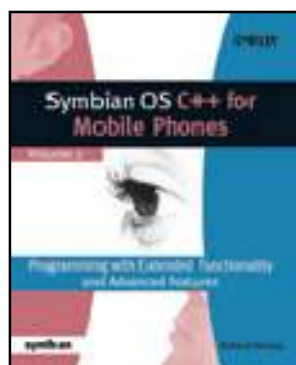
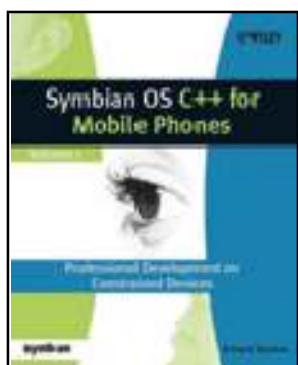
نوشته کریگ هیت

Smartphone Operating Concepts With Symbian OS

نوشته مایک جیپینگ

Acrcdited Symbian Developer Primer

نوشته جو استیکبوری و مارک جیکوبس



کتابچه‌هایی از

symbian
Developer
Network



کتابچه‌های منتشر شده

Coding Standards

Coding Tips

Performance Tips

Getting Started

Java ME on Symbian OS

P.I.P.S

Carbide.c++

Data Sharing Tips

Essential S60 – Developers' Guide

Essential UIQ – Getting Started

Ready for ROM

کتابچه‌های ترجمه شده در این زبان‌ها موجود هستند:

اسپانیایی

روسی

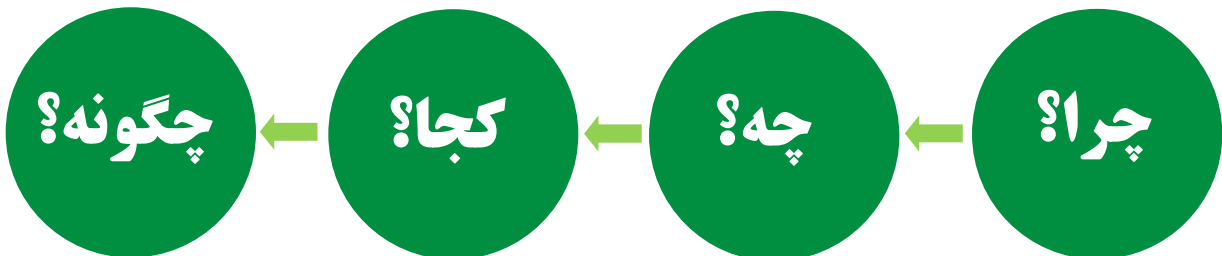
چینی

ژاپنی

کره‌ای



CARBIDE.C++ V1.3



این کتابچه یک معرفی بر محیط مجتمع توسعه برای برنامه‌های Symbian C++، یعنی *Carbide.c++* ویرایش ۱.۳ است. این کتابچه، ویژگی‌های جدید، نحوه وارد کردن پروژه‌های C++ Symbian موجود، نحوه ایجاد یک پروژه جدید و نحوه ساخت و اجرای برنامه‌ها توسط *Carbide.c++* را به شما می‌گوید.

Carbide.c++ بخشی از سری «استفاده از سیستم عامل Symbian» است که به منظور تهیه اطلاعات برای برنامه‌نویسان Symbian، طراحی شده است.

انتشارات Symbian

انتشارات Symbian، کتاب‌هایی را منتشر می‌کند که به منظور ایجاد ارتباط موثق، به موقع، مرتبط و اطلاعات عملی درباره سیستم عامل Symbian و دیگر تکنولوژی‌های مربوطه، طراحی شده‌اند. می‌توانید در سایت زیر، اطلاعاتی درباره سری انتشارات Symbian، پیدا کنید:

developer.symbian.com/books