

Adobe InDesign CS2 & SDK  
Keith Alcock, TCS Member  
tcs@keithalcock.com

InDesign is Adobe's mid-level document layout and design application for newsletters, magazines, newspapers, etc. Its price and presumably capabilities huddle between PageMaker on the low end and FrameMaker on the high end. Competitors include the familiar Corel WordPerfect and Microsoft Word for smaller documents and lesser known programs like QuarkXPress with which entire newspapers are published. Strangely enough, I am unlikely to produce a document from scratch using InDesign. My intention is to use InDesign as a platform for experimental manipulation of text attributes (e.g., leading, kerning, tracking, scaling). The program includes a very capable palette for this purpose, but manually adjusting attributes for large numbers of characters is tedious and error prone. This is the job for another piece of software which Adobe calls a plug-in.

Most readers will be aware of Word "macros," which have evolved into complex programs written in Visual Basic for Applications. WordPerfect supports a similar WordPerfect Script and QuarkXPress uses so-called eXTensions with a C interface. InDesign's plug-ins are written in C++. It is an unfortunate necessity to rewrite at least some code for each platform. My sample application is what I call a ransom note generator. It applies a different attribute value to each letter in the document. Each can have different scaling, for example, so that it looks as if the letters were collected from different sources and glued together. This sounds simple enough, but hurdles abound. Word bogs down quickly under these conditions, WordPerfect does not support horizontal scaling at all, and Quark hyphenation goes haywire. Can InDesign handle it? I am pleased to report that it can!

To create a plug-in for InDesign, one needs to download the SDK (software development kit) from the Adobe website (see below). While the SDK is free, support is not, of course. Still, this beats Quark's developer program, which requires a fairly expensive membership for any access whatsoever. The kit downloads and installs just like any other software, resulting in a large number of PDF documents, C++ support files, HTML pages, example code, and help files comprising thousands of pages. I managed to scratch the surface only enough to be dangerous. For the Windows operating system, Microsoft Visual Studio 2003 is the supported development platform. One application, DollyXs, is written in Java and requires the runtime environment to be installed. It produces a blank template VC++ project for a plug-in. The template includes a pre-configured custom compilation step that would be difficult to configure manually. For this first project, however, I had better luck searching out one of the myriad example projects and modifying it to fit my purposes.

Portable, pluggable C++ architectures are very complex and not recommended for casual programmers or those revisiting C++ like me. A familiarity with COM is very helpful and a partial COM type library is provided for JavaScript support, but the

interface appears incomplete compared to the C++ interface. The SDK makes use of templates, functors, iterators, and other constructs that make the code compact but difficult to debug and nearly impossible to read for human and compiler alike. Luckily, setting a breakpoint is very simple, so that execution can be observed. Be sure to use the release project settings, however, since the debug version requires a special debug edition of InDesign that isn't free. Since the program takes a substantial amount of time to start, slowing edit/compile/debug cycles, I would typically create a skeleton plug-in first with extra menu items to load and unload a separate DLL containing any code that does not interact directly with the application. That way InDesign can continue to run across debugging sessions. It's not Java hot code replacement or Smalltalk live objects, but it helps.

I found the learning curve very steep and was forced periodically into trial and error programming. For example, documentation for the parameters to `ITextSelectionSuite::ChangeTextSelection()`, especially direction (kNext, kPrev, kStart, kEnd, and kAll), was so terse that I ran through many combinations before stumbling across the right one. Nevertheless, I completed and tested the ransom note plug-in. It runs slowly, but I have undoubtedly overlooked a more efficient interface to text selections. More importantly, the resulting document does not overwhelm InDesign and hyphenation does not suffer for all the manipulations so that the platform is suitable for my experiments. In browsing the documentation, I discovered interesting capabilities previously unknown to me such as support for paragraph composers, hyphenation services, custom text attributes, and integrated undo. The SDK's comprehensive interfaces and InDesign's large collection of standard text attributes combine to produce a very satisfying platform for typography. With a small dose of customization, InDesign can place more capabilities at your fingertips than you may have thought possible.

### **Company/Product information**

Vendor: Adobe Systems Incorporated

URL: <http://www.adobe.com>

Product: Adobe InDesign CS2 (Creative Suite 2)

Price: \$699 online from Adobe

SDK: <http://partners.adobe.com/public/developer/indesign/sdk/index.html>

### **Abbreviated system requirements**

Processor: Intel Pentium III or 4 [or equivalent]

Operating System: Windows 2000 SP3 or XP

Graphics: 1024 x 768 x 16-bit

For a screen shot of the plug-in menu and ransom note document, please see my website at <http://www.keithalcock.com>.