

CHOOSING A PRINT SOFTWARE DEVELOPMENT KIT (SDK)

A white paper by Global Graphics Software, part of
Hybrid Software

Contents

Executive summary	3
The challenge	4
The role of PDLs in printing	4
What is a print SDK?	6
Performance under pressure: the need for speed in print SDKs	6
Building smarter print solutions without being a print expert	7
Making the selection	9
Conclusion	12
Glossary	13

Executive summary

Modern print workflows are increasingly complex, requiring software that can reliably process and manage high volumes of richly formatted content. Central to this capability is the use of Page Description Languages (PDLs) such as PDF, PostScript, and PCL—languages that define how content is laid out and rendered for print. However, selecting a suitable print Software Development Kit (SDK) goes far beyond simply parsing these formats.

Many SDKs on the market claim PDF support, but only a select few are truly optimized for the unique demands of print environments. These demands include handling real-world files (often imperfect or non-compliant), supporting rich color management, simulating overprints, managing variable data printing (VDP), and operating at scale with high-speed output.

The challenge is compounded by a growing skills gap in the print industry, as experienced professionals retire and new developers enter without deep print expertise. This makes it critical for SDKs to be “smart”—offering print-aware APIs, automation, and built-in intelligence that empower developers to create reliable, high-performance print applications without requiring domain expertise.

Beyond technical capabilities, factors like cross-platform compatibility, GPU utilization, support for modern programming languages, and access to responsive technical support all contribute to the overall value and long-term viability of a print SDK.

This document provides a roadmap for selecting the right SDK—covering essential features, performance considerations, and evaluation criteria. By choosing a mature, feature-rich, and print-specialized SDK, organizations can build scalable, future-ready solutions that deliver both innovation and production-grade reliability.

The challenge

If your organization is involved in print workflows, whether as a service provider or a technology supplier, you will need to work with Page Description Languages (PDLs) such as PDF, PostScript, or PCL.

To develop software that handles these file types, you will need a Software Development Kit (SDK) that understands these formats. Your requirements might include extracting job information, generating thumbnails, converting between PDLs, transforming color, creating impositions, or adding print marks.

There are over 30 SDKs available that work with PDF (a common PDL), and many of them are designed to support a wide range of markets, such as digital archiving and electronic documents. Some of these SDKs may excel in areas like interactive forms or document security, but lack key features required for high-quality print workflows.

So how can you identify the right PDL SDK for use in the print industry? A dedicated print SDK that is purpose-built for print-related tasks will provide the tools and performance you need to ensure reliable and efficient processing of print jobs.

The role of PDLs in printing



Page Description Language (PDL) documents are widely used in the printing industry to produce high-quality output from digital content. A PDL defines how a page should appear when printed, including details such as fonts, text, graphics, and images, as well

as instructions for sizing and positioning each element on the page. This allows for precise control over the print process and ensures the final output closely matches the original design. The most commonly used PDLs are PostScript, PDF, and PCL.

Within the PDF family, several print-specific standards have been developed to meet the needs of professional workflows. Standards like PDF/X-1 and PDF/X-4 are designed to ensure predictable and consistent printing by enforcing rules around color management, font embedding, and transparency. These variants are widely adopted in commercial printing environments where reliability and output fidelity are essential.

In addition to the mainstream PDLs, there are several specialized formats used in specific sectors of the printing industry. XPS (XML Paper Specification), developed by Microsoft, is sometimes used in enterprise or Windows-centric environments but is less common in high-volume print workflows. PPML (Personalized Print Markup Language) is designed for variable data printing, allowing personalized content to be efficiently produced at scale. IJPDS (Inkjet Print Description Stream) is a PDL optimized for high-speed production inkjet printers, enabling rapid transmission of print data to devices operating at very high throughput rates. While these formats are not as widely adopted, they serve important roles in niche or high-performance applications.

If we just take one PDL (PDF) the challenge of processing it correctly for print is substantial. The PDF specification alone spans over 1,000 pages, with detailed definitions for color spaces, transparency blending, image compression, overprint behavior, and other features critical to high-quality print output. It includes hundreds of graphical operators that a print workflow must interpret precisely to ensure accurate rendering. Beyond the core spec, PDF relies on several supporting standards that are essential in print environments, such as ICC profiles for color management, OpenType for font rendering, and XMP for handling print-related metadata. Even aspects like trapping, spot colors, and output intents require close attention. To handle print-ready PDFs reliably, a solution must go far beyond basic file parsing and support the full print-focused feature set of the PDF ecosystem.

What is a print SDK?

A print SDK is specialized software tools designed to meet the unique needs of the print industry developers. Developers use a print SDK to create print workflow components that interrogate, manage or change documents as they move from one stage to the next.

A print SDK is one that understands process color (CMYK) and spot colors (additional inks). It will have built-in color management that uses ICC profiles to ensure the correct result on the press. A print SDK should have the ability to produce separations and to screen its output or be capable of integrating with a component to apply screening to contone (continuous-tone) raster data.

Performance under pressure: the need for speed in print SDKs



There is always a desire to complete any kind of processing as quickly as possible, but in printing, speed is often critical. This is not only because printing systems may need to keep up with the pace of physical or mechanical processes, such as those on a press or finishing device, but also because print SDKs frequently handle very large files, both in terms of size and resolution. They also manage high-volume workflows that can involve tens of thousands, hundreds of thousands, or even millions of pages.

This is especially true in variable data printing (VDP), where each page may be personalized with unique text, images, or layout. It's not uncommon for VDP jobs to consist of 100,000 pages or more, and in enterprise or transactional environments, jobs with several million pages are routine. In these scenarios, even saving a few seconds per page can lead to dramatic time and cost savings, reducing overall processing time by hours or even days, and improving throughput across the entire production line.

One powerful feature in PDF for improving performance in print workflows is the use of XForms (also known as Form XObjects). These allow reusable content—such as logos, backgrounds, headers, or templates—to be defined once and referenced multiple times throughout the document. This not only reduces the overall file size, making PDFs more efficient to store and transfer, but more importantly, it enables the RIP (Raster Image Processor) to optimize rasterization. Since the content is identical across multiple pages, the RIP can cache and reuse the rendered output, dramatically reducing the processing time for each page. In high-volume jobs where the same elements appear repeatedly—such as in transactional print or direct mail—leveraging XForms can significantly boost throughput and reduce system load.

Building smarter print solutions without being a print expert

An SDK provides APIs (Application Programming Interfaces) that hide the underlying complexity of a PDL, allowing developers to work with a consistent, PDL-agnostic model of the document, and to interact with its elements through well-defined interfaces.

For print, simply having access to those elements is not enough. There is a need for semantic understanding of often complex representations of color, graphics, and text. A Print SDK must provide APIs that are specifically tuned to the needs of developers building print workflow components—developers who may not be experts in print technology. This is especially important today, as many of the seasoned print professionals who built and maintained traditional workflows are now retiring from the industry. Unfortunately, printing is no longer seen as a cutting-edge field by many new developers, and the influx of specialized expertise is not keeping pace.

As a result, being “smart” is no longer just a nice-to-have for a Print SDK—it is essential. Operators expect their digital press systems to anticipate their needs and make intelligent decisions on their behalf. To enable those capabilities, developers must rely on the SDK’s built-in “print intelligence” to deliver advanced, reliable solutions without needing deep domain knowledge. A well-designed SDK helps bridge this growing skills gap, empowering teams to build powerful print solutions in an increasingly resource-constrained environment.

Making the selection

The following section outlines key topics and questions to discuss with each PDL SDK vendor to help you make an informed decision.

Multiple PDLs

If your application is guaranteed to work with only a single page description language (PDL), you may have a wider range of SDK options to choose from. However, if there’s any chance you’ll need to support multiple PDLs—such as PDF, PostScript, and PCL—it’s far more efficient to select a single SDK that handles all of them natively. Relying on multiple SDKs from different vendors can introduce complexity, compatibility issues, and increased maintenance overhead. A unified SDK with built-in support for multiple PDLs ensures consistency across your workflow and simplifies development, integration, and long-term support.

Maturity matters

There are hundreds of PDF creation tools on the market, and while many aim to follow the specification, true compliance is difficult to guarantee. PDF files can contain subtle errors—not only in their structure and markup, but also in underlying technologies such as font programs (for example, TrueType fonts are themselves executable programs). Handling these issues gracefully goes beyond strict spec compliance; it requires deep, real-world experience.

A mature SDK is shaped by years of handling imperfect, non-compliant, or broken files from diverse sources. Each issue reported and resolved contributes to its robustness and reliability. In contrast, a newly developed SDK—even one with excellent theoretical compliance—may struggle when faced with the messy reality of real-world PDFs. For production environments where reliability is critical, it’s wise to choose a print SDK with a

proven track record and the battle-tested resilience that only time and real-world exposure can provide.

Variable Data Printing

Efficient variable data printing (VDP) is another must-have for a print SDK. This capability allows a PDF template to be replicated with variable data (such as text, images and barcodes) sourced from a database to produce PDFs containing thousands or millions of personalized pages.

Industry standards

A reliable print SDK must adhere to industry standards, capable of producing compliant PDF/X files.

Overprint simulation

For a PDF SDK to function correctly in print workflows, it must support overprint simulation (or emulation). Without this capability, PDF files will not appear the same on screen as they do when printed, leading to inaccurate previews and potential production errors. Overprint handling is essential for preserving the visual fidelity of documents, particularly in professional print environments where spot colors and transparency effects are used.

Pass standard test suites

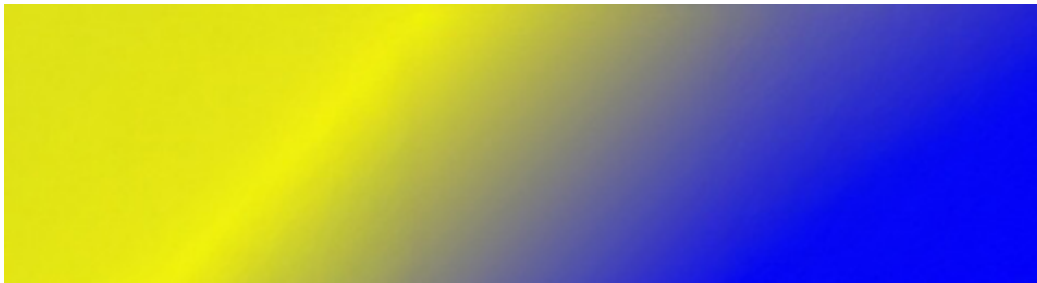
One way to assess the print capabilities of a PDF SDK is by running it against an industry-recognized test suite, such as the Ghent Workgroup PDF Output Suite. These test files are designed to uncover issues in rendering and color handling by using visual indicators—such as crosses within color patches—that highlight discrepancies. A correctly functioning SDK will render these pages without any visible crosses, signalling accurate processing and readiness for production use. Successfully passing such tests is a strong indicator that the SDK meets the quality standards expected in professional print environments. The latest version of the Ghent PDF Output Suite can be found on the Ghent Workgroup website (www.gwg.org/gos5/).

Color Management Module (CMM)

A Color Management Module (CMM) is responsible for accurately converting colors between different color spaces, ensuring visual consistency from design to output. When selecting a print SDK, it's important to ensure it includes a reliable, industry-proven CMM. Look for

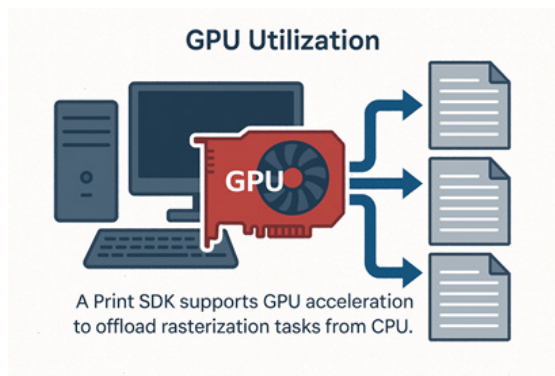
capabilities that go beyond basic CMYK support, such as extended gamut processing, accurate spot color handling, and support for ICC color profiles. These advanced features are essential for maintaining color fidelity in demanding print workflows.

Smooth gradients



The ability to render smooth shades and gradients is essential for producing high-quality vignettes and color blends. A robust print SDK should support both 8-bit and 16-bit rendering to ensure smooth tonal transitions without visible banding, especially in high-resolution or color-critical print applications.

GPU utilization



Originally driven by gaming and more recently accelerated by AI workloads, Graphics Processing Units (GPUs) are now a standard component in most systems—including desktops, laptops, and servers. However, they remain largely underutilized in the print industry. A print SDK that supports GPU acceleration can unlock significant performance gains by offloading rasterization tasks from the CPU. This not only speeds up rendering but

also frees the CPU to handle other parallel operations, improving overall system efficiency. When multiple GPUs are available, workloads can be distributed even further, enabling high-throughput processing for demanding, high-volume print environments.

Cross-platform compatibility

Cross-platform support is a key requirement for any print SDK, as modern print workflows often span multiple operating systems. For example, it's common to see a Windows-based job submission interface working alongside a Linux-based RIP, chosen for its performance and stability. An SDK that offers consistent APIs across platforms allows for flexible workflow design, easier integration, and reduced development and maintenance effort.

Equally important is the ability to extend your print solutions into the cloud. As workflows increasingly move toward hybrid or fully cloud-based models, your SDK should be ready to support that transition. You shouldn't have to switch technologies or rewrite components just to move from desktop to cloud. Look for a print SDK with proven capabilities not only on desktop and server environments, but also on mobile and in the Cloud—ensuring scalability and future readiness for your print solutions.

Programming languages

```

from jawsakoIF_python import *
mako = IJawsMako.create()
IJawsMako.enableAllFeatures(mako)

# Input
assembly = IJawsMako.create();
pageContent = IJawsMako.enableAllFeatures(mako);

# Render
renderer = IJawsMako.create();
renderedImage = renderer.render(pageContent);
IDOMPNGImage = IDOMPNGImage.encode(mako, renderedImage,
    IOOutputS

import com.globalgraphics.JawsMako.jawsakoIF.*;
var mako = IJawsMako.create();
IJawsMako.enableAllFeatures(mako);

// Input
var mako = IJawsMako.create();
IJawsMako.enableAllFeatures(mako);

// Render
var assembly = IJawsMako.create();
var pageContent = IJawsMako.enableAllFeatures(mako);
var renderedImage = IJawsRenderer.create(mako).render(pageContent);
IDOMPNGImage = IDOMPNGImage.encode(mako, renderedImage,
    IOOutputS

const auto mako = IJawsMako::create();
IJawsMako::enableAllFeatures(mako);

// Input
const auto assembly = IPDFInput::create(mako)->open("myfile.pdf");
const auto pageContent = assembly->getDocument()->getPage()->getContent();

// Render
const auto renderer = IJawsRenderer::create(mako);
const auto renderedImage = renderer->render(pageContent);
IDOMPNGImage::encode(mako, renderedImage,
    IOOutputS::createToFile(mako, "rendered.png"));
  
```

To maximize developer productivity and ease of integration, a print SDK should support the most commonly used programming languages. At a minimum, this includes C++ and C#,

which are widely used in performance-critical and Windows-based applications. Support for additional languages such as Java or Python can be a significant advantage, offering greater flexibility for teams working across different platforms or prototyping environments.

Developer Support

While open-source solutions may appear attractive due to their low upfront cost, they often lack timely and reliable support. In contrast, commercial print SDKs typically offer dedicated technical support, providing faster resolutions, regular updates, and critical fixes when needed. For high-stakes print projects where reliability and turnaround time are crucial, access to responsive, expert support can make a significant difference in maintaining smooth operations and meeting production deadlines.

Conclusion

Choosing the right Print SDK is a strategic decision that has a direct impact on the performance, reliability, and future scalability of your print solutions. While there are many SDKs that support PDLs like PDF, not all are built with the needs of the print industry in mind. Print workflows demand far more than basic compliance—they require intelligent handling of color, layout, performance optimization, and real-world file resilience.

A print-specific SDK should offer more than just tools; it should provide a foundation of built-in expertise. From support for multiple PDLs and robust color management to overprint simulation, VDP optimization, and cross-platform/cloud readiness, the right SDK will empower your development team to build advanced, production-ready applications without needing deep print knowledge.

As the industry evolves—and as the pool of experienced print professionals shrinks—developers need smarter, more capable tools to bridge the gap. By selecting an SDK with proven pedigree, comprehensive features, and strong vendor support, you're not just enabling development; you're ensuring your print solutions are built to meet today's demands and tomorrow's opportunities.

Glossary

ICC profile	An ICC (International Color Consortium) profile is a file that describes how a device, like a monitor or printer, reproduces color, ensuring accurate color rendering across different devices and mediums.
Ghent Workgroup	The Ghent Workgroup, or the GWG, is an international organization made up of graphic arts users, associations, educational institutions & developers building best practices for print and packaging workflows. The resulting recommendations, specifications, and workflow testing suites make it easy for increasing numbers of professionals around the world to successfully create, process, and exchange graphic arts files for numerous applications. www.gwg.org
PDL	A PDL or Page Description Language is a computer language that describes the appearance of a printed page.
PDF/X	A "PDF subset standard" describing constraints on PDF for delivering files for production printing.
RIP	A RIP or Raster Image Processor converts text and image data from many file formats including PDF, TIFF™ or JPEG, into rasters that a printing device understands.

About the author

David Stevenson, Product Manager



David Stevenson is product manager for Global Graphics Software's Mako Core™ SDK and responsible for the performance component in SmartDFE™, the Digital Front End for high-speed, single-pass inkjet presses. Throughout his career, David has specialized in electronic documents of one sort or another, starting with Xerox Corporation as a product manager for Venture Publisher, an early star of desktop publishing on PCs. That was later followed by a 13-year career at Adobe, specializing in various aspects of PDF: creative print workflows, electronic forms and accessibility. At Global Graphics he has continued to focus on PDF technology and solutions for PDLs.

About Global Graphics Software

Global Graphics Software develops smart software components that enhance digital print devices. For print OEMs and Independent Software Vendors, our award-winning Digital Front Ends, core SDKs and technologies meet the most demanding printing applications—from commercial printing to labels and packaging, and industrial inkjet. Customers include HP, Canon, Durst, Mimaki, Mutoh, Roland, Kodak and Agfa. The roots of the company go back to 1986 and to the iconic university town of Cambridge, and, today the majority of the R&D team is still based near here. Global Graphics Software is the OEM business unit of Hybrid Software (Euronext: HYSG).



April 2025

Contact us:
info@globalgraphics.com

www.globalgraphics.com

Global Graphics Software Inc.
6601 S Tamiami Trail,
Suite 176
Sarasota
FL34231, USA
Tel: +1-941 925 1303

Global Graphics Software Ltd
Building 2030
Cambourne Business Park
Cambourne, Cambridge
CB23 6DW UK
Tel: +44 (0)1954 283100

Global Graphics KK
613 AIOS Nagatacho Bldg.
2-17-17 Nagatacho, Chiyoda-ku,
Tokyo 100-0014
Japan
Tel: +81-3-6273-3198