itext
by apryse

Apsveicam iText ar 25. jubileju!
Szczęśliwej 25. rocznicy dla iText!
iText ಗೆ 25ನೇ ವಾರ್ಷಿಕೋತ್ಸವದ ಶುಭಾಶಯಗಳು!
iText को 25वीं वर्षगांठ की शुभकामनाएँ!
iTextның 25 еллык юбилее белән котлыйбыз!
Šťastné 25. výročí iTextu!
iTextને 25મી વર્ષગાંઠની શુભકામનાઓ!
Joyeux 25e anniversaire à iText !
iText-কে ২৫তম বার্ষিকীর শুভেচ্ছা!
សូមអបអរសាទរឆ្នាំទី 25 សម្រាប់ iText!
Gefeliciteerd met de 25e verjaardag van iText!
iText အတွက် 25 နှစ်ပြည့်နေ့ကို ဂုဏ်ပြုပါသည်!
З 25-годдзем, iText!
iText'in 25. yıl dönümünü kutlarız!
iTextning 25 yillik yubileyi bilan tabriklaymiz!
สุขสันต์วันครบรอบ 25 ปี iText!
Happy 25th Anniversary to iText!
Su 25-uoju iText jubiliejumi!
Selamat ulang tahun ke-25 untuk iText!
25
YEARS
iText ପାଇଁ 25ତମ ବର୍ଷଗାଠିର ଶୁଭେଚ୍ଛା!
iText ਨੂੰ 25ਵੇਂ ਸਾਲਗਿਰਹ ਦੀਆਂ ਵਧਾਈਆਂ!
Buon 25° anniversario a iText!
عيد ميلاد سعيد الخامس والعشرون لـ iText!
Feliz 25º aniversário para iText!
З 25-річчям, iText!
iText-in 25 illik yubileyi mübarək!
iText के 25वें वर्षगांठ की शुभकामनाएँ!
С 25-летием, iText!
Feliz 25º aniversário iText!
iText کے 25ویں سالگرہ کی مبارکباد!
iTextக்கு 25வது ஆண்டு விழாவிற்கு வாழ்த்துகள்!
יום הולדת 25 שמח ל-iText!
Herzlichen Glückwunsch zum 25. Jubiläum, iText!
iText-тің 25 жылдығымен құттықтаймыз!
iText కు 25వ వార్షికోత్సవ శుభాకాంక్షలు!
Sugeng tanggap warsa kaping 25 kanggo iText!
iText-ന് 25-ാം വാർഷികം ആശംസകൾ!

This is a special *iText* PDF

For this release, we have decided to create a PDF file that showcases a lot of what is possible with iText by Apryse.

This is no ordinary PDF, as it is PDF/A-4f, PDF/UA-2 and WTPDF compliant, created with the help of pdfHTML.

You can read more about it below.

Table of Contents

Release iText Core 9.1

✓

We said the previous 9.0 release was extra-special, but 25th anniversary releases don’t come along every day! So, this time we've prepared an extra-*extra*-special PDF showcase to demonstrate the improved SVG capabilities, faster tables, along with a bunch of advanced iText features. The pdfHTML add-on was used to turn these release notes into a document conforming to PDF/A-4f. You can see the SVG Text improvements and how it benefits pdfCalligraph, along with the major table and CSS enhancements.

As always, it features a dynamically generated table of contents and bookmarks, layers, SVG rendering, and automatic pagination using our Events engine.

We've also embedded the source code and required resources to recreate the document, along with a MAC-protected version of the document (use the password "itext" to open it), and digitally signed it with a National Identity Card too. You can even validate it using the included Java and C# code samples!

Release date: Feb 14th, 2025

To celebrate both iText's 25th anniversary and Valentine's Day, we bring you iText Core version 9.1. There's a lot to love about this release, with a huge performance increase in table creation, massively extended SVG support, and further Digital Signing goodies.

That's by no means all though, as we're also releasing new add-on versions across the iText Suite PDF SDK.

Extended SVG Support

Many additions and enhancements have been made to our in-house implementation since it was introduced, with coverage of the specification steadily increasing to meet customer needs. This release sees our biggest leap yet with over 40 tickets being closed – full marks to our incredible dev team!

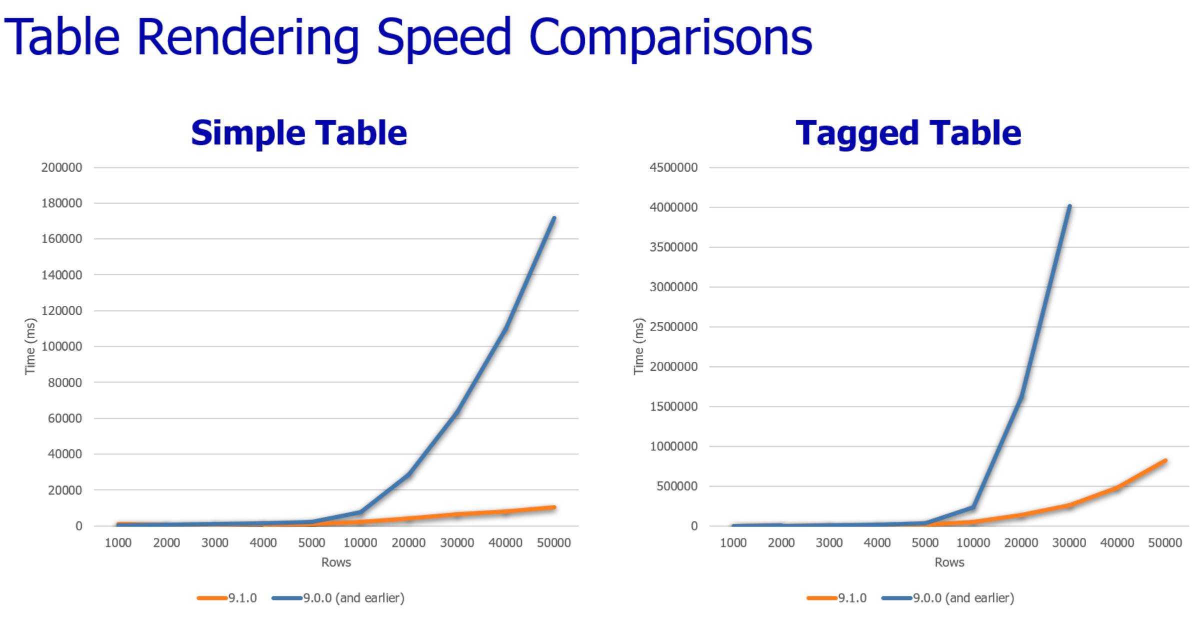
Newly added are support for text clipping paths, 'marker-mid' properties, text decoration, and passing markers from `<g>` elements to children. We've improved general font handling, while some other improvements to draw attention to are in the support for relative size attributes, text positioning, 'direction' properties, stroke opacity, and dash patterns.

There's also improved support for CSS-specific SVG, which you can find more details on in the pdfHTML release notes. However, we'll call out the improved support for different CSS origins in SVG and referencing external resources with the `@importurl()` rule.

We've significantly improved the SVG module's usage of the advanced typography features enabled by the pdfCalligraph add-on. This, along with many other things is shown off in the release notes PDF.

Increased Table Performance

With iText Core 9.1 the table rendering code has been highly-optimized, particularly when it comes to tagged tables. If you look at the comparison graphs below, you can see there is a huge reduction in the time iText takes to create a table.



Time taken to render tables is greatly reduced, especially as table size increases

Digital Signatures

We're continually working on iText's digital signing and validation capabilities to provide a unique breadth of support in the market. MAC integrity protection support is extended to support two-step signing. There's also new code examples for signing with the Cloud Signing Consortium (CSC) API, which we recently wrote about in Part V of our Digital Signing with iText series. You can find these in the GitHub samples repositories linked below.

Alongside that, we've made some general improvements (if you know, you know) to signing and validation. In particular, the workaround for certificates where the `pathLength` parameter is set to 0 for the `basicConstraints` extension is no longer required.

PDF/UA-2

Our PDF/UA-2 implementation is improved, specifically, when using the Annot tag for content elements in PDF 2.0 documents. Don't tell anyone, but our team is preparing further PDF/UA-2 goodies for our next release.

Pull Requests

For this release, we'd like to thank Stefan Bechtold for submitting a PR adding support for styling tables with nth-last pseudo class selectors. Thanks also to Artyom Skrobov for squashing a bug when decoding an empty PdfString, and finally Zuzu-Typ for fixes to the kernel PDF encryption constants documentation.

Bug Fixes and Miscellaneous

There's an update for merge handling when remote and embedded go-to actions are present, and we resolved a customer issue related to decoding Xref streams with missing bytes.

Many bugs have been resolved for SVG rendering and CSS layout. In addition, general bugfixes have been made to font and text handling, form fields, signing and validation, and more.

Other Stuff

As always, you can see the Changelog below to see the full rundown on what's new in Core, and details of other improvements and bugfixes for this release.

If you use iText for digital signing, you may be interested in the Digital Signatures Hub which contains a ton of useful resources and examples.

Don't forget that in addition to the resources on our Knowledge Base, on our GitHub you can find a ton of useful up-to-date samples in the following repos:

Java

- https://github.com/itext/itext-publications-examples-java
- https://github.com/itext/itext-publications-book-java
- https://github.com/itext/itext-publications-signing-examples-java
- https://github.com/itext/itext-publications-signatures-java
- https://github.com/itext/itext-publications-highlevel-java
- https://github.com/itext/itext-publications-jumpstart-java

.NET

- https://github.com/itext/itext-publications-samples-dotnet

If you want to create ZUGFeRD/Factur-X-e-invoices with iText Core, we have both Java and .NET code samples available targeting the current ZUGFeRD/Factur-X specification. They demonstrate how to embed the XML invoice data and add the metadata required for conformance. Read this article to learn more about ZUGFeRD/Factur-X, and using these code samples to create EN 16931-compatible e-invoices.

Bear in mind that our `master` branch contains samples for the current stable release, while the default `develop` branch is for the bleeding edge commits towards the next release.

Also, don't forget to check out the release-related examples below, as well as the updated Core add-ons in the iText Suite we've released this time:

iText Suite 9.1 Releases

- Release pdfCalligraph 5.0.1
- Release pdfHTML 6.1
- Release pdfOCR 4.0.1
- Release pdfSweep 5.0.1
- Release pdfXFA 5.0.1

Release Related Examples

- Adding a digital signature placeholder in HTML templates
- Adding Bookmarks / Table of Contents to a pdfHTML conversion document
- Overview: Using CSS in SVG with iText SDK

Downloads

	GitHub	**Maven**	**NuGet**	**Artifactory**
iText Core – 9.1.0 (Java)	link	link	N/A	link
iText Core – 9.1.0 (.NET)	link	N/A	link	link

Changelog

New features

- Implement Text - Advanced Fonts
- SVG Text text-decoration

- Clipping paths on text
- Support ‘marker-mid’ property
- Support underlined text, dashed lines
- Pass markers from <g> element to its children
- Support display none and visibility hidden (except text)

Improvements

- Table performance improvements
- Typography usage in SVG
- Support relative sized SVG in img and object HTML elements
- Support ‘direction’ property
- Text position improvements
- Add default fonts to BasicFontProvider for SVG
- Check childLength calculations with several fonts
- Text Whitespace handling
- Support stroke dash pattern in layout for text elements
- Support MAC integrity protection in two step signing
- OCSP verifier support alternate CertID representations
- Provide injecting of http, ocsp and crl clients in validation chain
- Re-evaluate Basic constraints handling of pathLenConstraint
- Improve documentation for FontProvider.addFont methods
- PdfFontFactory.registerDirectory recursive search
- Add support for multiple parents in PdfLayer
- Inconsitent id when creating formFieldElements
- Make EventManager thread-safe
- Update verapdf to the latest version

Bug fixes

- BIDI is not correct when converting SVG
- <polygon>: tops of figures are cut
- Fix percent width&height resolving of top level SVG when it is used from html
- Improve relative values support
- Support color related style attribute and values for SVG image
- Division by zero in AbstractBranchSvgNodeRenderer#calculateAndApplyViewBox results in NPE at
- PdfFormXObject#calculateBBoxMultipliedByMatrix level
- SVG transformation matrix applied incorrectly
- Improve viewBox applying for AbstractBranchSvgNodeRenderer
- Get rid of unexpected deformation when rotating a marker with preserveAspectRatio="none"
- Wrong displaying of svg if invalid url attribute for fill property set
- Rounded rectangle with rx or ry negative displayed incorrectly
- Incorrect display of ellipse with absent Rx, Ry
- SVG, SXP: transformations with invalid arguments cause NPE
- addAnnotation method doesn't annotate content elements with Annot tag when PDF version is 2.0
- Renderer splitup issue
- Flex container with table splitup issue
- Error decoding stream with FlateDecodeFilter and DecodeParams
- Timestamp signature validation fails over encapMessageContent
- Set print flag to the newly created signature field in PdfSigner
- Default font size is calculated incorrectly for rotated form fields
- Fix explicit destination handling in case of Remote go-to and Embedded go-to actions

Deprecation of Older iText Versions

We're taking this opportunity to announce End of Life dates for deprecated iText versions. EOL for iText 7.1 will be April 2025, and iText 7.2 will be October 2025. As for iText 8.0, this will reach EOL in October 2026.

EOL for these versions means they will transition to maintenance mode, and only receive security patches. However, just as with iText 5 we will continue to provide support for our customers. Even though iText 5 has been in maintenance mode since 2016, we regularly release new versions to address CVEs and other security-related bugfixes. See CVEs or the iText 5-specific CVEs page for more information.

Contributors

We'd like to shout out the following contributors for this anniversary release:

- Core team
    - Eugene Bochilo
    - Dmitry Chubrick
    - Angelina Pavlovets
    - Alexandr Pliushchou
    - Vitali Prudnikovich
    - Dmitry Radchuk
    - Andrei Stryhelski
    - Nanou Persoons
    - Glenn Volckaert
    - Alexandr Fedorov
    - Guust Ysebie
    - Yulian Gaponenko
    - Raf Hens

- Product / Marketing
    - André Lemos
    - Ian Morris
- Infrastructure / Devops
    - Marco Andries
    - Yauheni Borbut
- Research
    - Michael Demey
    - Vlad Lipskiy
- Input from other teams
    - Rainer Plöckl
    - Alison Anderson
- Input from outside
    - Michael Klink
    - Matthias Valvekens

eBooks

- Blockchain for PDF Documents
- iText: Jump-Start Tutorial for .NET
- iText: Jump-Start Tutorial for Java
- iText: Converting HTML to PDF with pdfHTML
- iText: Building Blocks
- Best iText 7 Questions on StackOverflow

Installation Instructions

- Installing iText for Java
- Installing iText for .NET
- Installing iText Community for Java developers
- Installing iText Community for .NET developers

Examples (latest ones)

- PAdES Signing High Level API
- iText Core: Signature Appearance Improvements
- iText 8.0.2 Delivers PDF/A-4 Support
- High-level Annotation Flattening
- FIPS & SHA3 Examples for iText Core 8.0.0

FAQ (latest ones)

- I upgraded to iText version 7 or 8, how do I use my license?
- How to change a background Image into a watermark by altering the opacity?
- How to change the text color of an AcroForm field?
- Is it safe to remove XFA?
- How to fill XFA form using iText without breaking usage rights?

Features of this PDF

- Used pdfHTML to generate the PDF content (the HTML content is attached)
- It is both PDF/A-4f and PDF/UA-2 compliant (making it also a WTPDF)
- There is a MAC protected (AES 256 Encryted, SHA 256 Protected. Password is 'itext') version of this PDF attached (had to be separate, as PDF/A-4 does not allow it)
- Digitally signed using a Portuguese Identity Card (scroll below to check the code on how to validate it!)
- The main logo is generated using iText's custom SVG rendering engine
- Dynamically generated table of contents and bookmarks
- *Creatively* used Layers to toggle between Java and .NET code below
- Automatic Pagination by using our Events engine
- Fonts were *subsetted* for a smaller file size

To run the project (using .NET 8), just check the instructions on the attached file `README.md`.

Signed off by:

André Ventura Lemos

Digitally signed by ANDRÉ VENTURA LEMOS
Date: 2025.02.17 15:27:59 +01:00
Reason: Release notes for iText 9.1.0
Location: Ghent (Belgium)

Generated by: Content: Ian Morris Source code: Guust Ysebie

SVG to PDF conversion by iText

The SVG module is a core component of iText which performs the operations of converting an SVG image into a PDF document, or low-level PDF syntax for actual document pages. Note, that current page and banner on the first page are created by converting two SVG files using external CSS styles into PDF syntax. See attached svgOverview.svg, svgExample.svg and svgStyle.css to check how it looks in browser.

- SVG is converted into vector image, so it is possible to e.g. copy text, unlike a rasterized image
- Text and *tspan* elements support:
  - interaction with **pdfCalligraph** for complex scripts
  - relative and absolute positioning for text
  - text clipping path
  - ~~text-decoration property~~
  - text-anchor and direction properties
- Basic shapes such as ellipse, circle, rectangle, line, path, polygon, polyline are supported, same for **group**, **image**, **marker**, **symbol**, **use** tags
- Most common SVG and CSS attributes
  - CSS is supported in **style** attribute or tag and as external stylesheet
- Relative values support (percents, **em**, **rem**)
- Clipping paths support
- Linear gradient and pattern support
- **svg** tag support:
  - nested SVGs
  - **preserveAspectRatio** attribute
  - relative values for **width** and **height**
- Non-scaling-stroke **vector-effect** is supported for basic shapes

Addendum: Signature validation example java

```java
package  com.itextpdf.signatures;
import com.itextpdf.kernel.pdf.PdfDocument;
import com.itextpdf.kernel.pdf.PdfReader;
import com.itextpdf.signatures.validation.SignatureValidationProperties;
import com.itextpdf.signatures.validation.SignatureValidator;
import com.itextpdf.signatures.validation.ValidationOcspClient;
import com.itextpdf.signatures.validation.ValidatorChainBuilder;
import com.itextpdf.signatures.validation.context.CertificateSource;
import com.itextpdf.signatures.validation.context.CertificateSources;
import com.itextpdf.signatures.validation.context.TimeBasedContext;
import com.itextpdf.signatures.validation.context.TimeBasedContexts;
import com.itextpdf.signatures.validation.context.ValidatorContext;
import com.itextpdf.signatures.validation.context.ValidatorContexts;
import com.itextpdf.signatures.validation.report.ValidationReport;
import java.io.IOException;
import java.time.Duration;

public class  ValidationExample {
    public static void  main(String[] args) {
        String path = "<path/to/signed/pdf>";
        SignatureValidationProperties properties = GetSignatureValidationProperties();
        properties.addOcspClient(new  ValidationOcspClient());
        properties.addCrlClient(new  CrlClientOnline());
        IssuingCertificateRetriever certificateRetriever = new  IssuingCertificateRetriever();
        ValidatorChainBuilder validatorChainBuilder = new  ValidatorChainBuilder();
        validatorChainBuilder.withIssuingCertificateRetrieverFactory(
                () -> certificateRetriever).withSignatureValidationProperties(properties);
        ValidationReport report;
        try (PdfDocument document = new  PdfDocument(new  PdfReader(path))) {
            SignatureValidator validator = validatorChainBuilder.buildSignatureValidator(document);
            // Validate all signatures in the document.
            report = validator.validateSignatures();
        } catch (IOException e) {
            throw new  RuntimeException(e);
        }
        System.out.println(report);
    }

    private static  SignatureValidationProperties  GetSignatureValidationProperties() {
        SignatureValidationProperties properties = new  SignatureValidationProperties();
        properties.setRevocationOnlineFetching(ValidatorContexts.all(), CertificateSources.all(), TimeBasedContexts
                .all(), SignatureValidationProperties.OnlineFetching.ALWAYS_FETCH);
        properties.setFreshness(ValidatorContexts.All(), CertificateSources.all(), TimeBasedContexts.of(
                TimeBasedContext.HISTORICAL), Duration.ofDays(- 5 ));
        properties.setContinueAfterFailure(
                ValidatorContexts.of(ValidatorContext.OCSP_VALIDATOR, ValidatorContext.CRL_VALIDATOR),
                CertificateSources.of(CertificateSource.CRL_ISSUER, CertificateSource.OCSP_ISSUER, CertificateSource
                        .CERT_ISSUER), false);
        return properties;
    }
}
```