This is a special *iText* PDF

As per usual, 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/UA-2 compliant created with the help of pdfHTML (one of the areas we have improved in this version).

Furthermore, we actually used Python to generate this PDF! Don't believe us? Just check the attached project, and read about it on our Technical Tales.

You can read more about it below.

Table of Contents

- .NET MAUI AOT Compilation
- Pull Requests
- Bug Fixes and Miscellaneous
- Other Stuff
- iText Suite 9.2 Releases
- Release Related Examples
- Downloads
- Changelog
- Deprecation of Older iText Versions
- Contributors

Release iText Core 9.2.0

As always, we've made the Core release notes into a showcase PDF document. This time we're utilizing the new HTML to PDF/UA API, which you can read more about in the release notes for pdfHTML 6.2.0. As you'll find, it conforms to the latest PDF/UA-2 standard for accessible PDF documents.

The other main difference is it's been generated and digitally signed with Python. We'll explain how that's possible below, but we've embedded the source code and resources required to recreate the document yourself. Have fun!

Release date: May 15th, 2025

Following the 25th anniversary release last time, you might think there wouldn't be much to expect from iText Core 9.2. But hold onto your hats, because we have some very nice stuff to talk about! This includes support for .NET MAUI AOT compilation, automated validation for PDF/UA-2 documents, and some other goodies including...iText for Python???

.NET MAUI AOT Compilation

The standout feature of this iText Core release is its support for Native Ahead-of-Time (AOT) compilation in .NET MAUI development. After introducing GraalVM Native Image compilation on the Java side last year, we're thrilled to bring similar perks to our .NET developer community. Since the addition of the NativeAOT runtime for iOS and macOS development in .NET 9's Multi-platform App UI (MAUI), we began implementing support as it could bring significant benefits for iText development.

In addition to the usual benefits of optimized native executables—like smaller file sizes, faster startup times, and reduced memory usage—Ahead-of-Time (AOT) compilation really shines when it comes to iOS development. This is mainly because there are some pretty strict rules around traditional Just-In-Time (JIT) compilation for apps that are headed for the App Store. So, leveraging MAUI's AOT runtime could really be a game changer.

Automated PDF/UA-2 Validation

Automated checks for PDF/UA-1 creation were introduced with iText Core 8.0.4. This release extends this feature to include the new PDF/UA-2 standard published last year.

The earlier checks were based on the PDF Association's Matterhorn Protocol PDF/UA conformance testing model. For the PDF/UA-2 checks, we draw from the profile used by the industry-standard validation tool, veraPDF. That said, the way we've implemented the system

of checkpoints and handle failure conditions is broadly the same. So, rest assured, iText Core will help you create PDFs that meet the new standard for universal accessibility.

While at present we're not aware of any legislation mandating the switch to PDF/UA-2 from the current PDF/UA-1 standard, there are some pretty compelling reasons to consider adopting the latest version, particularly when it comes to handling PDF 2.0 documents. PDF/UA-2 brings improved support for modern Unicode to the table, and rolls out new document structure elements like `Title`, `DocumentFragment`, `Aside`, `FENote`, and `Artifact` which take advantage of enhancements in the PDF 2.0 specification.

In addition, we have refactored the conformance-checking mechanism by introducing the new `PdfConformance` abstraction to support multiple standards (e.g. PDF/A, PDF/UA), with `PdfAConformance` and `PdfUAConformance` now existing as enums, enabling modular, extensible validation.

Pull Requests

We'd like to thank Aviad Pineles for their PR to improve CSS style sheet logic for pdfHTML. While we eventually went with a different implementation of this fix, we're very grateful for the inspiration!

Bug Fixes and Miscellaneous

In line with our efforts on the PDF/UA-2 checks and the upgraded API for converting HTML to PDF/UA, we also have some enhancements and fixes for general PDF 2.0 conformance. One of the standout updates is a dedicated checker that evaluates tag structure by examining the parent-child tag relationships as outlined in the PDF 2.0 specification. We've also addressed a number of issues related to PDF 2.0 tag conversion and structure repair operations, making sure everything operates seamlessly.

We would also like to highlight an additional change: in response to a bug report on StackOverflow, we identified and resolved a line wrap issue affecting non-wrapping italic and bold simulated text within table columns. This problem could lead to incorrect page breaks when the content width exceeded the column limits. To fix it, we've adjusted the table renderer logic to handle cases where text formatted in such ways exceeds the layout box width.

Other Stuff

Those of you who were intrigued by the reference to Python at the beginning will be interested in a new article in our Technical Tales section of the Knowledge Base. Vlad Lipskiy from our Research Team has been very busy over the past few weeks exploring how Python.NET can enable direct usage of iText from Python. In addition to porting a wealth of code samples, he's also written a superb article which takes a deep-dive into how it all works. Make sure to check it out!

Another great article is by Guust Ysebie from the iText SDK development team. If you caught the iText Core 9.1.0 release, you might remember how we managed to massively speed up table rendering, particularly in the case of tagged tables. If you wondered how this was possible, Guust has written up the whole story from beginning to end. It's a very entertaining and enlightening read, so if optimization is your bag you'll find a lot to enjoy.

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 9.2 Releases

- Release pdfCalligraph 5.0.2
- Release pdfHTML 6.2.0
- Release pdfOCR 4.0.2
- Release pdfOptimizer 4.0.2
- Release pdfSweep 5.0.2
- Release pdfXFA 5.0.2

Release Related Examples

- New High-Level HTML to PDF/UA API
- Refactored PDF Conformance Architecture

Downloads

	GitHub	Maven	NuGet	Artifactory
iText Core – 9.2.0 (Java)	link	link	N/A	link
iText Core – 9.2.0 (.NET)	link	N/A	link	link

Changelog

New features

- Support for .NET MAUI AOT compilation
- Automated checks for PDF/UA-2 Validation
- Introduce PDF 2.0 tag structure checker

Improvements

- toLowerCase/toUpperCase locale independent usage
- Use System.Text.Json instead of Newtonsoft.Json for netstandard 2.0
- Added copyright attribution clarification for jsoup-related files

Bug fixes

- Cell with non breakable content and not enough horizontal space leads to page break and data loss
- Font selection breaks tag structure
- BitsPerComponent: 1 and Indexed DeviceRgb failes with "The color depth 1 is not supported error."
- Wrapping ListItem content in a Div with list symbol inside causes wrongly aligned content
- KeepWithNext property causing formatting issue
- PDF 2.0 does not allow DIV, P tags to be children of the P tag
- PDF 2.0 does not allow P, Hn tags to be children of the Form tag
- PDF 2.0 documents should not contain Document tag if it does not contain any content
- Improve AESCipherCBCnoPad documentation
- sxHeight is not set in OS/2 table metrics when processing TTF fonts
- Standard security handler issue with long value for user access permissions
- Wrong relative width calculations for hr tag
- Distinguished Name comparison should be done using the equivalent method instead of the equals method
- Incorrect producer line for LTA signatures in AGPL mode
- PdfDocument Constructor causes endless loop
- setAlternativeDescription doesn't set Contents entry for PdfSigner
- PdfUAConformanceException when SignatureFieldAppearance contains Image

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

◦ Yulian Gaponenko ◦ Raf Hens	• 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)

- Refactored PDF Conformance Architecture
- PAdES Signing High Level API
- iText Core: Signature Appearance Improvements
- iText 8.0.2 Delivers PDF/A-4 Support
- High-level Annotation Flattening

Features of this PDF

- Used pdfHTML to generate the PDF content (the HTML content is attached)
- It is PDF/UA-2 compliant

- 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
- The Table of Contents has been dynamically generated
- Pagination was also done automatically using our Event engine
- Fonts were *subsetted* for a smaller file size

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

Signed off by:

Digitally signed by ANDRÉ VENTURA LEMOS
Date: 2025.05.15 17:05:24 +02:00
Reason: Release notes for iText 9.2.0
Location: Ghent (Belgium)

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