accusoft®

Enhancing Efficiency with Automated Document Assembly

eGuide

Introduction

In the first installment of our contract automation series, we looked at some of the unique challenges organizations face when it comes to creating, managing, and editing contracts and other confidential documents. We also explored how document assembly can help them solve a variety of these issues.

Document assembly is a form of programmatic automation that uses logic-based systems to create documents from pre-existing text and data without any direct guidance from the end-user. The automation process can incorporate conditional and variable text based on the specific use case, which makes it ideal for assembling legal documents, contracts, and letters. In this eGuide, we'll take a look at how document automation works and how it can benefit organizations in the legal, government, and insurance industries.

Streamlining Contract Management

Most organizations utilize contracts to formalize business relationships, but creating good contracts can be a complicated process. There is often a tradeoff between speed and security, and few companies have the proper collaboration tools in place to achieve both.

A hastily-drafted contract typically contains numerous errors, any one of which could expose the parties involved to significant liability. More importantly, the rush to create a contract quickly leads people to use familiar, but insecure, communication channels like email, which could potentially compromise confidential information contained within the contract. As the contract goes through this rapid editing and negotiation process, it's easy to lose track of which version is the most up-to-date.

On the other hand, taking careful steps to ensure accuracy and security during contract creation can draw out the negotiation process for far too long. Clients, partners, and vendors want to close deals quickly and will often take their business elsewhere if it takes weeks, or even months, to create a suitable contract. Without a system in place for creating similar contracts quickly, organizations will struggle to scale effectively and remain competitive in an increasingly fast-paced business environment.

Overcoming Human Error

Most of the challenges involved with contract creation are directly related to the manual labor involved. Whether they're creating a new contract from scratch or adapting an existing contract template, organizations often turn to traditional word processing programs like Microsoft Word or Google Docs out of sheer habit. Someone then has to draft new language or copy existing text and paste it into the document. Not only is this a very time-consuming process, but it also tends to be mistake-prone, which creates the potential for scrivener's errors.

A scrivener's error is an unintended typographical or formatting mistake in a contract. It could be a simple as a misspelled word or missing punctuation, but the term also applies to omissions that could significantly impact the contract's terms. Although some errors are obvious mistakes, others can result in protracted disputes that require a court ruling to resolve. Given the stakes involved, any measure that reduces the risks of human error during the contract creation process is immensely beneficial to an organization.

Automated Document Assembly

Automating the document assembly process ensures that contracts will be created according to specific, templated guidelines. This is especially helpful in situations where there is a great deal of similarity between contracts and minimal tweaks are required (such as service agreements or sales renewals). Information can be programmatically extracted from various sources and inserted into the new document for review and revision, effectively eliminating the tedious and error-prone manual assembly process.

Take, for example, a situation where a firm is receiving multiple documents containing potential contract information, such as the names of parties, specific obligations, and grounds for remuneration. All of these documents may also feature extensive editing markups and redlines that need to be taken into consideration when drafting the final contract. Extracting that information for assembly requires a solution capable of programmatically scanning and retrieving raw text, sifting through it to determine whether it includes the necessary clauses for a chosen template, and then assembling it while also inserting comments from various versions of the document into the correct location in the final, assembled contract.

Automation tools can reference templates and previous contracts to provide suggested clause content that might need to be included. Since every element of the document can be tracked programmatically, it's easy to monitor who is making changes and inform all collaborators about them. End users can then resolve, unresolve, or delete comments as a batch before outputting the final contract agreement.

How Programmatic Document Assembly Works

Implementing an effective contract automation process typically requires two distinct components: an AI solution that can sift through raw text using a combination of machine learning and natural language processing algorithms to identify key contract elements and a document editor capable of extracting and searching text within that application. In this section, we'll focus on the role a capable document editor plays in the contract assembly process.

The first step in the assembly process involves the use of a document upload API that converts existing text into a new document with a unique document ID that can be referenced programmatically. Once that document is created, a search API request can identify specific field placeholders in a contract template and replace them with data from the uploaded text. Typical placeholder fields use information like company name, phone numbers, or agreement dates. After the proper data is inserted, the current contents of the document can be downloaded to the appropriate application using another API.

Critically, this process is performed completely through code, making it far more efficient and accurate than manual contract assembly. Since template fields are being filled automatically, there's no risk of an omission or improper keystroke creating an error in the contract.

Document editor APIs can also be used to replace specific content markers with paragraphs or other text (such as clauses in a list format) programmatically. More importantly, the document-in-progress can be shared in a view-only format that allows other contributors to review changes and updates without granting them the access to make their own changes. This helps to maintain strict version control over a contract and avoid any confusion over whether or not it's the most up-to-date version of the document.

Integrating a Contract Automation Solution

For software developers building cutting edge LegalTech, InsurTech, and FinTech applications, contract automation and document assembly features are often just one aspect of their platforms. Their key differentiators tend to take the form of powerful AI solutions, database management capabilities, and intuitive user interfaces that help promote more efficient workflows and a better customer experience. Designing versatile editing tools capable of handling contracts and other essential documents from assembly to their final, completed form takes time and resources that developers often cannot spare if they want to get their products to market quickly.

Given their constraints, many developers turn to separate, third party applications or open source toolkits to meet their document editing needs. Unfortunately, both choices present problems. While third-party applications, such as Microsoft Office, provide a great deal of functionality, they also expose customers to substantial risk. Opening a document with these programs typically means moving it outside the secure environment of the user's application, which could create myriad compliance issues for an organization. The risks are even greater for a cloud-based service because it necessarily means granting a third party access to confidential data. Last, but not least, most third-party document solutions lack effective version control features to identify which document is the most up-to-date.

Developers often turn to open source code to integrate the editing features they need, but these toolkits tend to be quite limited. They're often outdated, making them unsuitable for the document assembly and automation needs of today's fast-moving, high-volume organizations. More importantly, the old adage of "you get what you pay for" is typically quite true of freely available open source code. Even setting aside concerns about bugs that could create all sorts of headaches during integration, the fact that no one is actively maintaining the code or taking responsibility for it means that critical security gaps often go unresolved. In fact, one study found that the number of known open source vulnerabilities increased by 130% from 2018 to 2019.

Contract Automation and API Integrations

Integrating document editing functionality through a dedicated API solution offers developers a far more stable and secure alternative to using external applications or open source tools. By incorporating advanced contract assembly and editing features directly into their applications, developers can maintain both security and control when it comes to document management. Once files are uploaded into their secure software, users can access a variety of automation and editing tools entirely within that protected environment. They can set controls to manage who can view and change files, streamline the document creation process with programmatic assembly, and make batch edits to multiple contracts entirely through code to minimize human error.

Developers also save valuable time by integrating a dedicated API solution that's customized to their application's needs. Rather than spending months building their own document editing tools from scratch or wasting weeks troubleshooting an open source solution that only partially meets their needs, they can quickly integrate powerful editing functionality in a matter of days to keep their project on (or ahead of) schedule. Integrating a ready-made solution allows them to get their product to market faster with robust features that will help it stand out from the competition.

They also gain the benefit of ongoing support. As developers continue to refine and upgrade the capabilities of their applications, they need to preserve the core functionality their customers rely upon to meet their business needs. Working with an actively supported document editing API integration ensures that they'll be able to offer new features to their users without having to worry about compromising performance in other areas. Even better, updates to their editing solution can help them to better meet the changing needs of their customers.

Conclusion

By integrating document assembly features into their applications, developers can provide their customers with powerful contract automation tools that help them to minimize errors and streamline workflows. Keeping the document editing process completely within their secure applications allows users to collaborate effectively without concerns over security or version confusion. Thanks to API integrations, developers can quickly incorporate these essential document handling features into their software so they can spend their valuable time and resources on their core functionality.

In our next installment of our contract automation eGuide series, we'll take a closer look at PrizmDoc Editor, Accusoft's powerful document assembly and editing API integration. From its beginnings as an internal solution to its current status as an industry-leading automation toolkit, we'll break down how PrizmDoc Editor helps developers bring comprehensive document management and editing features to innovative applications deployed across a variety of industries.

accusoft®

@Accusoft