

Document Engine Admin Guide

Introduction

Document Engine is a managed package that provides a simple yet powerful and flexible way to implement PDF document generation functionality in your Salesforce instance. Whether you need basic document templates or complex, dynamic PDFs with conditional logic and multi-language support, Document Engine offers an intuitive solution that scales with your needs. With Document Engine, you can:

- Create PDF documents for any standard or custom Salesforce object
- Design templates using HTML and CSS
- Use dynamic content with merge fields and Handlebars expressions
- Support multiple languages with localization
- Preview documents during template content design
- Deploy templates between Salesforce orgs

Getting Started

Prerequisites

Package Installation Note: We recommend using the **Install for Admins Only** option when installing the package. If it was installed for non-admin profiles, we strongly recommend reviewing the *DocumentEngineAdmin* [custom permission](#) assignments (please do not confuse with the *DocumentEngineAdmin* permission set) and removing the permission from non-admin user profiles. This will ensure proper configuration consistency.

After the Document Engine has been installed, complete the following steps:

- Assign the **Document Engine package license** to all users who will use PDF generation functionality.
- Assign **DocumentEngineAdmin** or **DocumentEngineUser** permission set to each licensed user.
 - ◇ [DocumentEngineAdmin](#) permission set enables template configuration as well as document generation. Such users should also have the ViewSetup standard permission. Usually the permission set is assigned to System Administrators.
 - ◇ [DocumentEngineUser](#) permission set enables document generation only. Such users do not need extra admin level permissions.

Important! Please make sure the admin and user permission sets are properly assigned; otherwise, you may get unexpected UI errors that don't clearly indicate the missing permissions issue.

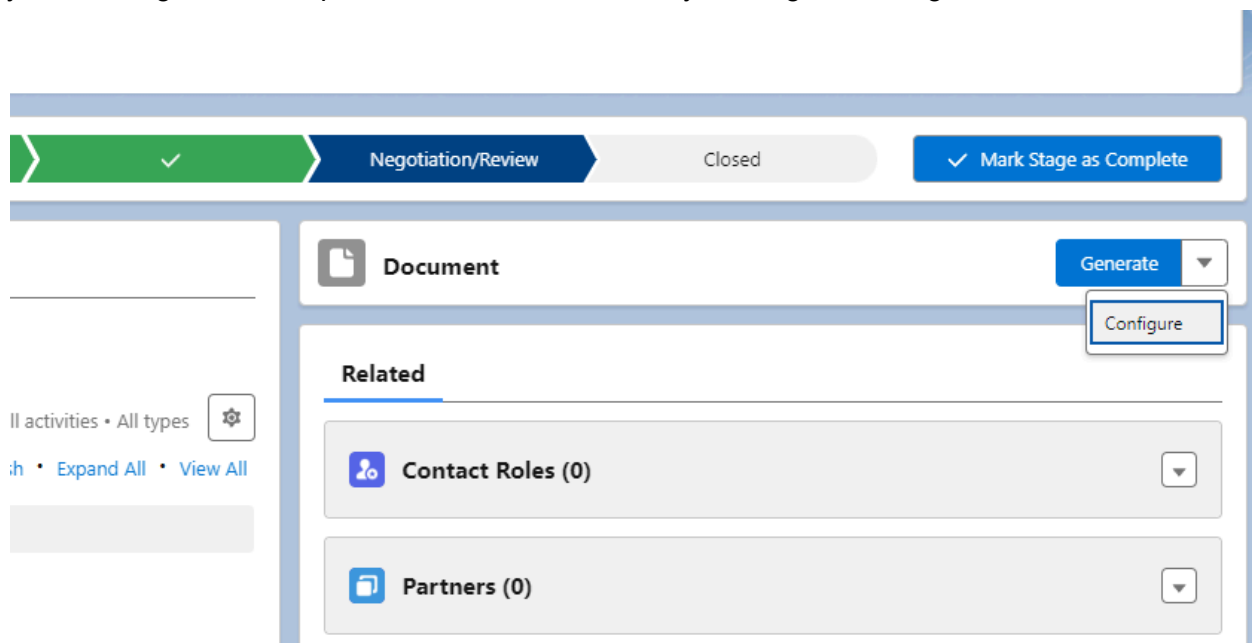
Initial Configuration

Document generation configuration and usage is available via the documentGenerator component that should be added to a Lightning record page of an object for which you need to provide PDF generation ability.

In order to use the component:

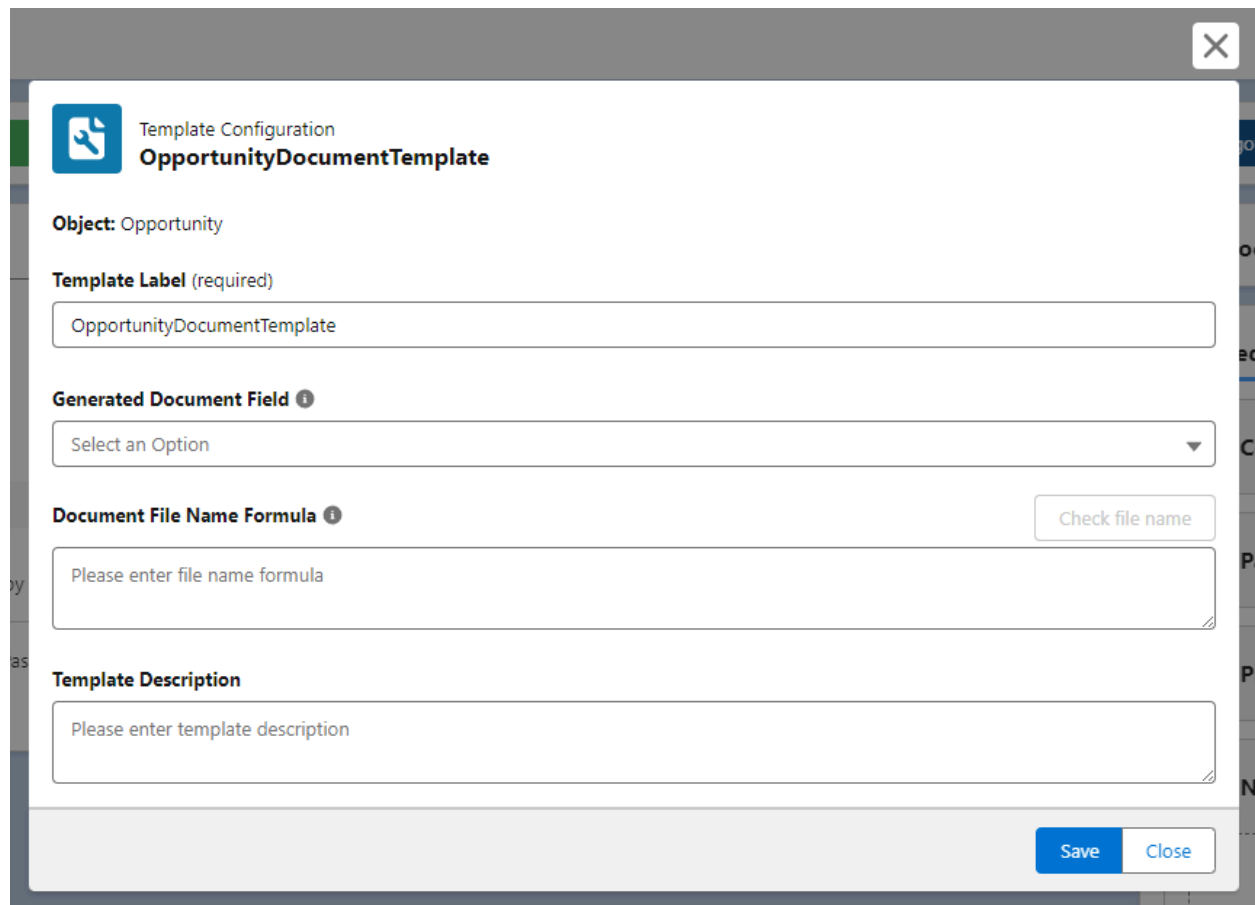
- Navigate to a Lightning Record Page of an object
- Edit the page and drag the documentGenerator component where you'd like to place it, for instance, a sidebar.
- Configure the component properties:
 - ◇ **Document Template Name:** Enter a unique template name (e.g., "OpportunityDocumentTemplate"). The name should follow Salesforce API name conventions and can only contain letters (A-Z, a-z), numbers (0-9), and underscores.
 - ◇ **Config Mode:** Set to true if you want to configure templates (admin only). False can be used later once the related template is configured and ready to use.
 - ◇ **Document Title:** Title to display on the component, by default it is "Document"
 - ◇ **Document Icon:** Icon to display on the component. It is an icon name from the standard [Lightning icons set](#). By default, the value is set to "document".

Once the component is added to the page, you should see the Generate button. If Config Mode is set to true, you will also see the Configure button in the actions dropdown menu. If you try to click the Generate button and there is no template with provided template name the app will ask you to configure the template first. You can do that by clicking the Configure button.



Template Creation

The Configure button displays a modal window containing a template detail page. If the template with the specified name doesn't exist yet, you should see a template creation form.



The screenshot shows a modal window titled "Template Configuration" for the "OpportunityDocumentTemplate". It contains the following fields:

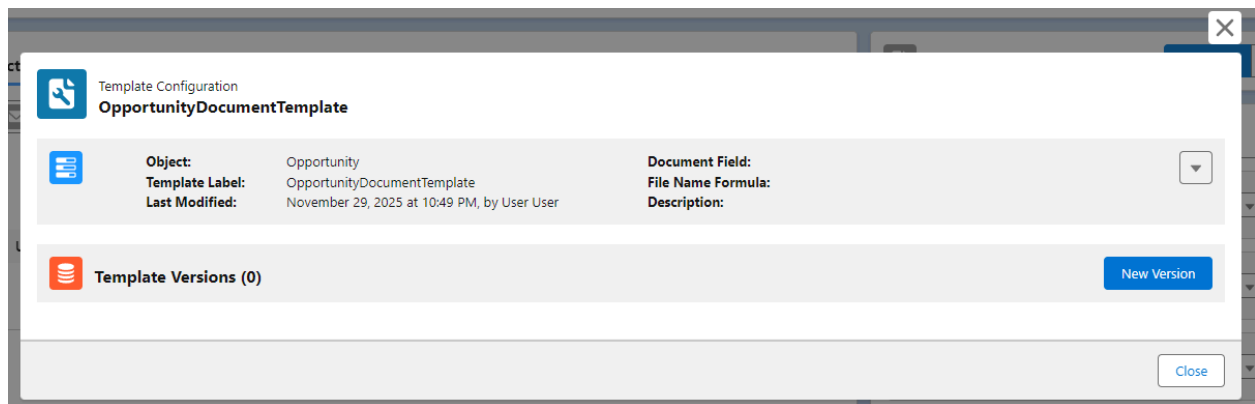
- Object:** Opportunity
- Template Label (required):** A text input field containing "OpportunityDocumentTemplate".
- Generated Document Field:** A dropdown menu with "Select an Option" as the current selection.
- Document File Name Formula:** A text input field with the placeholder "Please enter file name formula". To its right is a "Check file name" button.
- Template Description:** A text input field with the placeholder "Please enter template description".

At the bottom right of the modal are "Save" and "Close" buttons.

- **Template Label:** You can change the template's label, which is the same as the template API name by default. Changing the label won't affect the related template's API name.
- **Generated Document Field:** A dropdown field to select a lookup field of the nbse__GeneratedDocument__c type. The field should be created manually. During PDF generation, the logic creates a new nbse__GeneratedDocument__c record and sets it to the specified lookup field. The field is not required for document preview during template design, but must be populated for actual PDF generation usage.
- **Document File Name Formula:** A dynamic formula that defines the filename for a generated PDF. The formula can use standard Salesforce functions and reference fields of the related object, similar to a regular formula field. If the field is blank, the app uses a default name consisting of the file creation date followed by the API name of the Salesforce object the template applies to.
- **Template Description:** Any text description that you want to add to the template.

Template Versions

Once a new template is saved, you are redirected to its detail view that contains a list of related Template Versions. A Template Version is the object that stores the actual document layout and data logic. To avoid confusion between templates and template versions, templates are called Template Configurations. For simplicity, throughout this documentation we will refer to Template Configurations as Configurations and Template Versions as Versions.

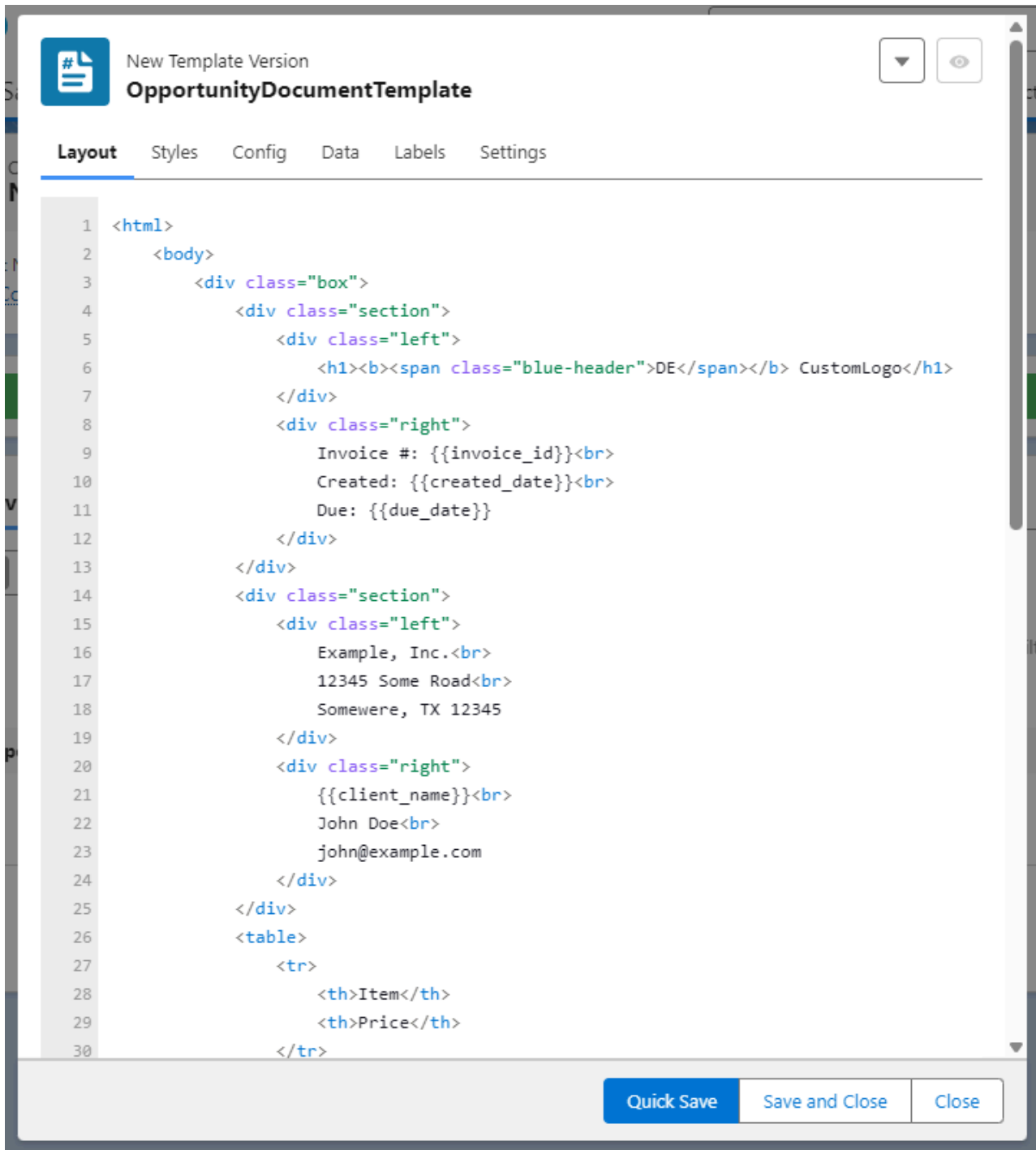


To create a new Version, click the New Version button. This will redirect you to the Version design page, which is split into two parts: a multi-tab Version editor on the left and the PDF preview window on the right. The right side is also used to display additional configuration UI described later.

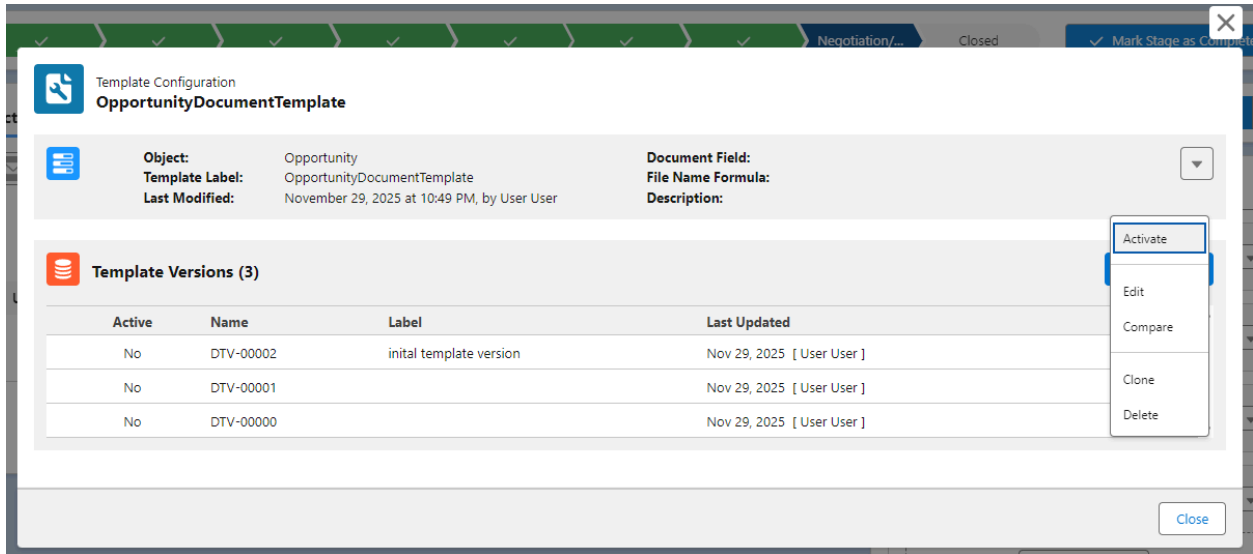
The Version editor has 6 tabs:

- **Layout:** HTML markup for the document
- **Styles:** CSS styles for formatting
- **Config:** PDF page configuration (margins, size, page numbers etc.), please refer to <https://document-engine.io/docs/api/#print-config> for details
- **Data:** Optional data builder class selection
- **Labels:** Multi-language label definitions
- **Settings:** Additional versions settings

Initially a new version contains a default sample code of a quote template in Layout, Styles, and Config tabs, and a sample JSON data in the Data tab.



Configurations can have many related Versions, but only a single Version related to a given Configuration can be activated at a time. Activation is done by clicking the Activate button in the Version's dropdown menu. During the activation process, the Version's content is copied to a corresponding nbse__DocumentTemplate__c custom metadata record.

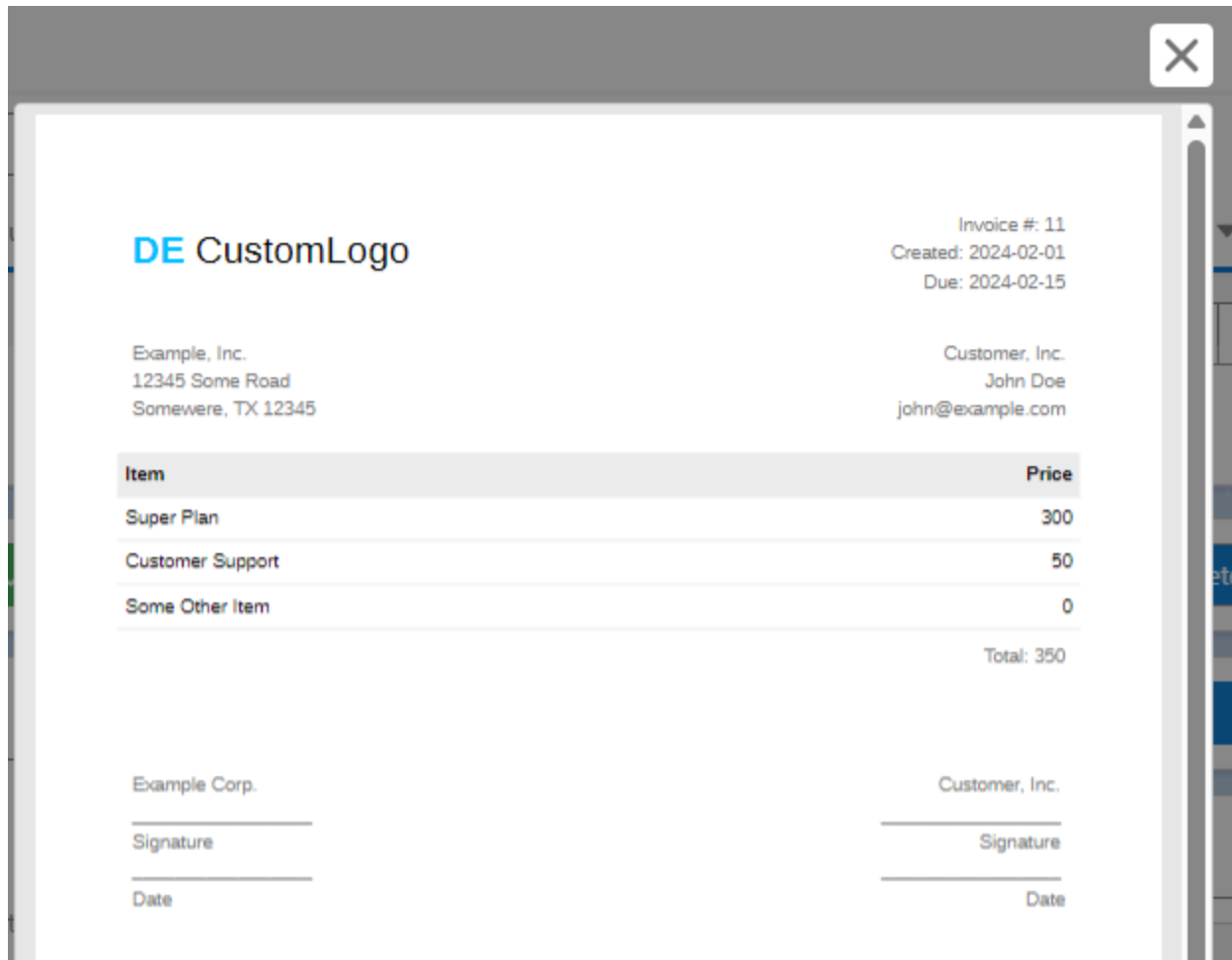


Once a Version is activated the configuration process is completed and users can start using PDF generation on the configured Lightning page. The custom metadata record can be deployed to other sandbox or production org and converted to a ready-to-use Version with a few clicks.

Note: It is expected that template activation takes some time, especially on a production instance, as the custom metadata update initiates a Salesforce deployment process.

Template Preview

The preview is launched automatically when a template version editor is opened. Also it can be re-launched via the 'Refresh Preview' button in the top-right corner of the editor section. In case the Version contains layout errors that block document rendering the preview action displays an error message with details. Usual errors are related to invalid handlebars expressions. The preview section provides a great QA tool while you are working on document content design. Each time you change Version's Layout or Styles you can check how the resulting document looks. Another important thing to note is that since the Version editor is opened via the lightning component on a record detail page, the preview is rendered using that record as a source. So you preview your template using an existing record right in the editor page, which effectively speed-ups the development process. If you configure multi-language support via the Labels tab you can also preview the document in all specified locales via the dropdown action menu near the preview button.



Template Layout Design

The Layout section of a Version should contain a valid HTML layout of the document. Feel free to remove the default layout provided for new Versions and redesign it as you like. We recommend avoiding inline CSS and adding required CSS styles in the dedicated Styles tab instead. Document Engine provides a powerful and flexible way for defining dynamic content and referencing record data via Handlebars blocks and expressions. As the topic is wide, we decided to move it to a separate guide. Please refer to the Document Template Design User Guide on our website.

Generating Documents via Apex

Document generation can be performed via the `DocumentEngineCallable` global Apex class. Available actions are `generateDocument` and `deleteDocument`. The callable result has three methods to extract results:

```
Boolean isSuccess();  
String getErrorMessage();  
Map<String, Object> getData();
```

If document creation is successful the `getData()` method returns this map:

```
{  
    'id' => <GeneratedDocument__c record Id>,  
    'title' => <document file name>,  
    'fileId' => <ContentDocument record Id>  
}
```

Document creation

```
final Callable docEngineCallable = new nbse.DocumentEngineCallable();  
final nbse.DocumentEngineCallable.DocumentEngineCallableResult result =  
(nbse.DocumentEngineCallable.DocumentEngineCallableResult)  
docEngineCallable.call(  
    'generateDocument',  
    new Map<String, Object> {  
        'recordId' => '<object record Id>',  
        'templateName' => '<template name>'  
    }  
);
```

Document deletion

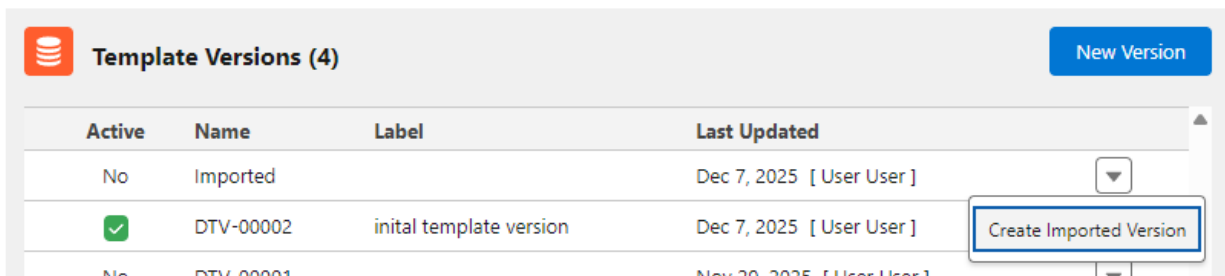
```
final Callable docEngineCallable = new nbse.DocumentEngineCallable();  
docEngineCallable.call(  
    'deleteDocument',  
    new Map<String, Object> {  
        'recordId' => '<object record Id>',  
        'generatedDocumentId' => '<GeneratedDocument__c record Id>'  
    }  
);
```


Deploying Document Templates

Activated document templates are saved to custom metadata instances so they can be exported to a Salesforce DX repository and used in a DevOps Center release pipeline or directly deployed between orgs.

The metadata record doesn't include the entire Template Configuration. If no Template Configuration exists, create one manually using the same API name that was used in the source org.

Deployed custom metadata appears in the Configuration Version list and can be converted to a Template Version record, then tested and activated as usual. Until that exported version is activated, it won't affect any existing Document Generation process events for a Configuration with the same API name and active Version. This is intentional to support a safe, step-by-step release process.



Active	Name	Label	Last Updated
No	Imported		Dec 7, 2025 [User User]
☒	DTV-00002	initial template version	Dec 7, 2025 [User User]
No	DTV-00001		Nov 20, 2025 [User User]