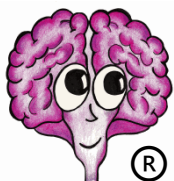


DocuBrain® Forms Guide



A DocuBrain® Product

<https://docubrain.com/>

By Prevo Technologies, Inc.

<https://prevo.com/>



DocuBrain® Forms Guide

By Prevo Technologies, Inc.

Copyright © 2025, Prevo Technologies, Inc. All rights reserved.

Published by Prevo Technologies, Inc., 1111 Keener Rd, Seymour, TN, 37865.

This guide is distributed with software that includes an end user license agreement (EULA). This guide, as well as the software described in the EULA, is furnished under license and may be used or copied only in accordance with the terms of the EULA. Except as permitted by the EULA, no part of this guide may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of Prevo Technologies, Inc. Please note that the content in this guide is protected under copyright law even if it is not distributed with software that includes a EULA.

The authoritative end user license agreement (EULA) can be found at:

<https://docubrain.com/licenses>

The content of this guide is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Prevo Technologies, Inc (PTI). PTI accepts no responsibility or liability for loss or damage occasioned to any person or property through use of the material, instructions, methods, or ideas contained herein, or acting or refraining from acting as a result of such use. PTI disclaims all implied warranties, including merchantability or fitness for any particular purpose.

DocuBrain and the DocuBrain logo are registered trademarks of Prevo Technologies, Inc.

Table of Contents

1. Introduction	1
2. Getting Started.....	2
3. Overview of Features.....	3
3.1. DocuBrain Forms Editor	3
3.2. Forms Engine.....	4
3.3. Electronic Notifications.....	4
3.4. Form Versioning.....	4
3.5. Printing and Exporting	5
4. DocuBrain Forms Editor Tutorial	6
4.1. Layout of the User Interface	6
4.1.1. Toolbar	6
4.1.2. Left Panel	7
4.1.3. Right Panel	8
4.1.4. Status Bar	9
4.2. Creating a Form.....	9
4.3. Connecting to the Forms Engine Repository	23
4.4. Uploading a Form to the Forms Engine Repository.....	24
4.5. Downloading from a Form from the Forms Engine Repository	27
5. DocuBrain Electronic Forms Specification	31
5.1. Styles	31
5.1.1. Style.....	32
5.2. Settings	34
5.2.1. SecurityAlias.....	34
5.3. Page.....	35
5.4. Form Components	41
5.4.1. CheckBox.....	45
5.4.2. ComboBox/Option	46
5.4.3. DateTextBox.....	46
5.4.4. DigitalSignatureBox.....	47

5.4.5. NumberTextBox	48
5.4.6. RadioButtonGroup/RadioButton	48
5.4.7. StaticText.....	49
5.4.8. StaticImage.....	50
5.4.9. TextArea	51
5.4.10. TextBox.....	51
5.4.11. TimeTextBox	52
6. The TechDoc Server-Side Forms Reference.....	53
6.1. Forms Management Menu	53
6.1.1. Forms Management.....	53
6.1.2. Forms Editor	53
6.1.3. Publishing a Form.....	54
6.1.4. Modifying a Form	55
6.1.5. Deleting a Form	56
6.1.6. Showing a Form.....	57
6.1.7. Viewing a Form	60
6.1.8. Submitting a Form.....	60
6.1.9. Exporting a Form	60
6.1.10. Creating a Form Submission	64
6.1.11. Modifying a Form Submission	65
6.1.12. Deleting a Form Submission	66
6.1.13. Resuming a Form Submission	67
6.1.14. Showing a Form Submission	67
6.1.15. Showing a Form Submission Signature	69
6.1.16. Viewing a Form Submission	70
6.1.17. Searching for Form Submissions.....	71
6.1.18. Purging Form Submissions.....	74
7. Suggestions and Feedback.....	78

Revision History

[illegible]

1. Introduction

Electronic forms are used very widely throughout the world of computing. Nearly every user of a computer at one point or another has completed an electronic form of some fashion. While electronic forms and data acquisition are one of the most common uses of a computer, there is not a single standard used across the board. While there are open standards available for use, they are very cumbersome and do not fully encompass all of the protentional uses for electronically collected data. Additionally, many of the other often used standard are developed by various vendors and are proprietary, closed source, and not human readable.

With all of this in mind, the DocuBrain Electronic Forms XML specification was developed. This specification is human readable, hand codable, and very easy to understand. Forms using this specification are inserted into TechDoc as regular documents so they pick up all of the abilities of a document in TechDoc such as history, versioning, collaboration, etc. These form documents are then simply published as forms and thereby becomes forms in TechDoc. The forms can then be viewed, edited, completed and submitted, etc.

When a user views a form or creates a form submission, they can do so using any standard modern web browser on any type of computing device without the need of any additional web browser plugin. Additionally, these form submissions can be exported to various forms including: XML, CSV, and PDF, and can be printed on any modern printer with the need for any special capabilities. When printed, form submissions should render nearly pixel-perfect across any device and printer however, some browsers and printers may need additional configuration of their print media and print margins to accomplish this.

The end result is a simple XML format that can be used to reproduce both your current paper forms and digital forms, in their entirety, capturing data in the same fashion, but now being backed by a very powerful document management system. While form submissions may be collected and used independently, they are integrated throughout TechDoc.

2. Getting Started

This guide contains the information you need to understand to design, build, and manage forms within DocuBrain products. Starting from the ground up, we will cover from the fundamentals of form design to publishing and data collection via form submission using the TechDoc form engine.

- [Overview of Features](#) – Provides a high-level overview of the features and capabilities.
- [DocuBrain Forms Editor Tutorial](#) – A tutorial detailing the use of the DocuBrain Forms Editor.
- [DocuBrain Electronic Forms Specification](#) – Covers the DocuBrain Electronic Forms Specification and includes various XML examples of what each element and their structure and attributes look like.
- [The TechDoc Server-Side Forms Reference](#) – Overview of the TechDoc Forms Engine and how to interact with it.

3. Overview of Features

DocuBrain TechDoc features a complete electronic forms solution without the need for any additional software or web browser plugin installations for designers or submitters. DocuBrain electronic forms are constructed using our web-based GUI tool the DocuBrain Forms Editor right inside of the web browser that's included as a part of the TechDoc Document Manager. This tool is used to construct forms using our open [DocuBrain Electronic Forms Specification](#) and can be used to deploy the forms directly to the Document Manager. Once deployed, the form can be made available and begin receiving form submissions from both users with a TechDoc account and users external to TechDoc that do not have an account. Once completed, the form submissions can be printed and/or saved individually or as a bundle to any printer or various file formats including PDF, XML, and CSV.

The electronic forms specification includes a variety of different types of controls including controls to capture digital signatures. In the next section, we'll go over the use of the DocuBrain Forms Editor, how to use the tool, construct a form, and deploy the form to a TechDoc Document Manager.

3.1. *DocuBrain Forms Editor*

The DocuBrain Forms Editor is an easy to use and intuitive web-based electronic form design tool. It allows users the ability to graphically design electronic forms following the DocuBrain Electronic Forms standard and deploy them directly to the Forms Engine inside of a TechDoc Document Manager. The DocuBrain Forms Editor features:

- Drag and drop modeling
- Forms built on the open XML-based DocuBrain Electronic Forms Specification
- Code free forms design
- Customizable look & feel
- Connects to TechDoc Document Managers
- Collaborative form design
- Form repository with versioning
- Form deployment
- Example Forms

3.2. Forms Engine

The TechDoc Forms Engine is extremely fast and lightweight. The Forms Engine follows the [DocuBrain Electronic Forms Specification](#) and is fully integrated with TechDoc. Here are just a few features of the TechDoc Forms Engine:

- Completely web-based, accessible from anywhere, and is even usable with mobile devices
- Forms administration and management dashboard
- Web-based form design tool
- Email-based notification and alerts
- Usable by both TechDoc users and external users without a TechDoc account
- Form versioning
- And much more

3.3. Electronic Notifications

The Forms Engine is fully integrated with TechDoc's email and notification system. If needed, a forms administrator can directly email submitters with a customized email and provide links to various places within the Document Manager and directly include a link to create a form submission. The submitter can follow this link and create a form submission with or without a TechDoc account.

3.4. Form Versioning

Once designed, forms are deployed to a TechDoc Document Manager. The form initially starts life as a normal TechDoc document so it gains all of the abilities of a document, things like: change tracking and history, attachments, keywords, etc. Since a form is a document, it also gains versioning abilities. When a form is first deployed, it will be designated as the first version (1.0). Over time, as needs change, the form can be modified adding or removing fields as needed. Depending on the size of the change, the existing 1.0 version of the form can be modified, or the form can be published as a new version (2.0 in this case). The already existing submissions when viewed, printed, etc. will be rendered using the original 1.0 version, while new submitters can begin filling out the new 2.0 version. Form versioning provides great flexibility as needs change over time.

3.5. Printing and Exporting

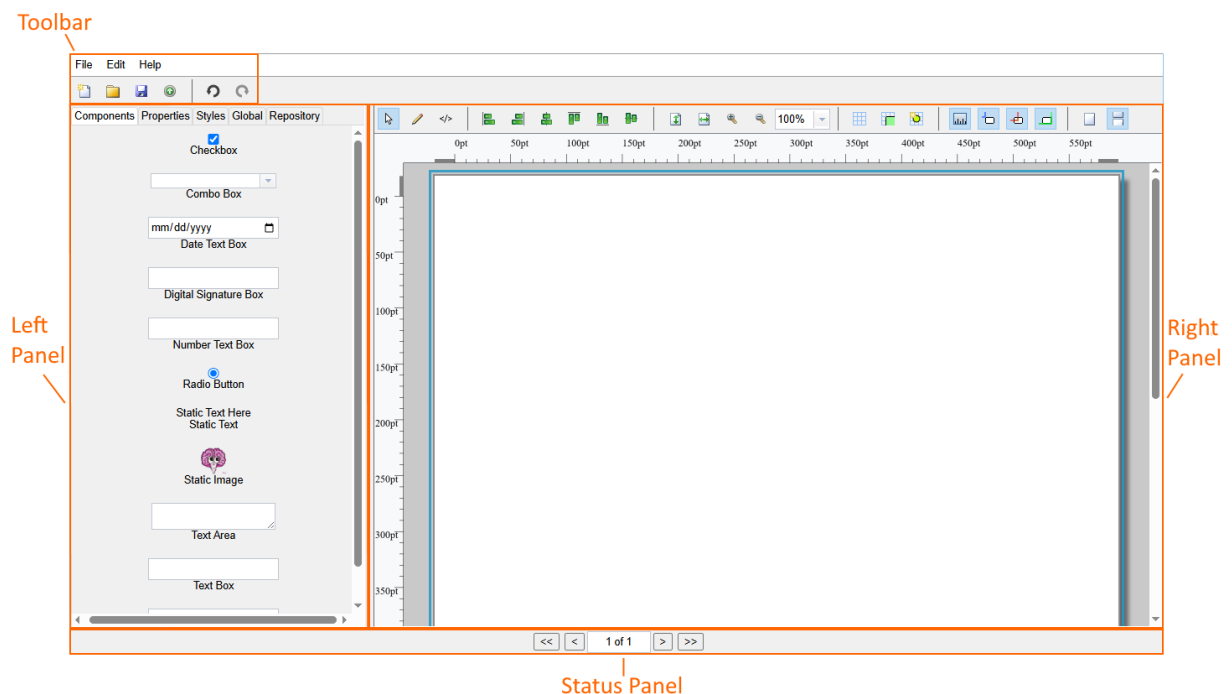
One of the main goals of the DocuBrain Electronic Forms implementation was to provide both great flexibility in the design of a form while also maintaining the ability to render and print the form on any platform and any printer and achieve the exact same look and feel. This means that forms can have a rich look and feel, more similar to designing a web page, while maintaining the ability to look the same when printed or exported to PDF. This also gives the ability to take forms that you are currently using now, either digital or paper, and replicate their look and feel in our specification. Then when printed or exported, they will look just as the forms you are already using and are used to seeing.

4. DocuBrain Forms Editor Tutorial

In this section, we will go over all the basics of using the DocuBrain Forms Editor. We will not touch much on the [DocuBrain Electronic Forms Specification](#) itself as knowledge of the specification isn't necessary to design a form. However, the specification may be viewed in the following section, if needed.

4.1. Layout of the User Interface

The editor is split into four main areas: the [Toolbar](#), [Left Panel](#), [Right Panel](#), and [Status Bar](#).









4.1.1. Toolbar

On the toolbar, you will find all of the menus for the editor and the short cut buttons. The toolbar contains the following menus:

- **File** – Used to open, close, and save forms as well as view examples and upload forms to a TechDoc Document Manager forms repository.

- **Edit** – This menu is only used when a form is open. It contains the undo/redo options to undo or redo an action to a form such as un-delete a component. This menu also contains the actions to add, insert, and delete a page.
- **Help** – The Help menu contains a link to this help file and the About dialog that shows the editor's version and your license info.


The icons of the toolbar from left to right are shortcuts to toolbar menu items. There are as follows:


-  File-> New Form
-  File-> Open Form
-  File -> Save Form
-  File -> Upload Form
-  Edit -> Undo
-  Edit -> Redo

4.1.2. Left Panel




The left panel contains five tabs: Components, Properties, Styles, Global, and Repository tab.

The Components tab displays all of the components available to add to a form. To add a component to a form, simply left-click and hold as you drag the component onto the form. When you release the left mouse button, the component will drop and become part of that page.




On the Properties tab, you will find all of the properties for the active form page or component. To view the properties of a component or page, click the pencil icon  to switch to edit mode, click the page or component, and then select the Properties tab. The Properties tab contains all of the properties and attributes of a page or component except styling (which you'll find on the Styles tab).

On the Styles tab, you will find all of the styles for the selected form page or component. To view the properties of a component or page, click the pencil icon  to switch to edit mode, click the page or component, and then select the Styles tab. The Styles tab

contains all of the styling for the page or component; things such as color, background color, size, position, etc.

The Repository tab is used to connect to the TechDoc Forms Engine inside of the Document Manager where you are running the Forms Editor. If you need to connect to a different Document Manager, you must log into that Document Manager and visit the Forms Editor there. Click the connect button  to connect to the Forms Engine and retrieve the list of all the forms you have access to. Once the list of forms has been retrieved, you can select a form and click the download button  to download the form from the Forms Engine (repository) and load it in the editor to view or edit. A form can be deleted from the repository by selecting it from the list and clicking the delete button . The internal ID of the form, the name of the form, and the description will be displayed for each form in the list.

4.1.3. Right Panel

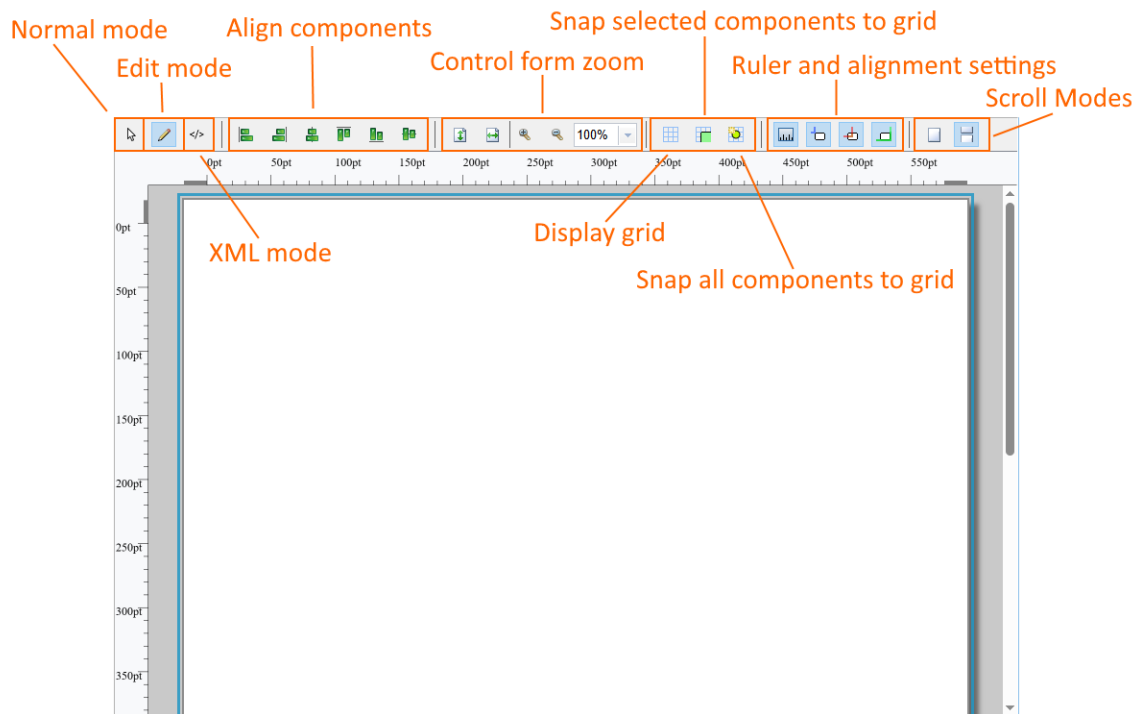
The right panel is the workspace of the editor. When you create or load a form, it is loaded into the workspace. The workspace has three modes: normal mode , edit mode , and XML mode .

Normal mode is used to interact with the form in its normal mode; this is how the form is presented to the user when they view the form or create a submission of the form. This mode is useful as you are able to work with and interact with the form just as the submitter would. This allows to you test out your components, review default settings, combo box options etc. as well as test out components like the digital signature component to make sure you've given the right amount of space for the user to draw their signature. Additionally, when using validation, you can test out your validation on components like a text field. If you've built a regular expression for validation, you can try out various types of input and see that your missing and error messages are appropriate for the conditions you test.

Edit mode is used when designing/editing a form. A form must be in edit mode to add/remove components to a page, add/remove pages to the form, etc. When in edit mode, you can click on a page or component to changes its properties and styling. This mode is used most often.

XML mode is used to review the generated XML for the current state of a form. This tab is mostly only useful if you are familiar with the [DocuBrain Electronic Forms Specification](#) and want to see how the editor has built the form. This tab can be useful to track down potential issues in your form.

The toolbar displayed over the right panel contains various buttons and settings, see the image below for a description of each:

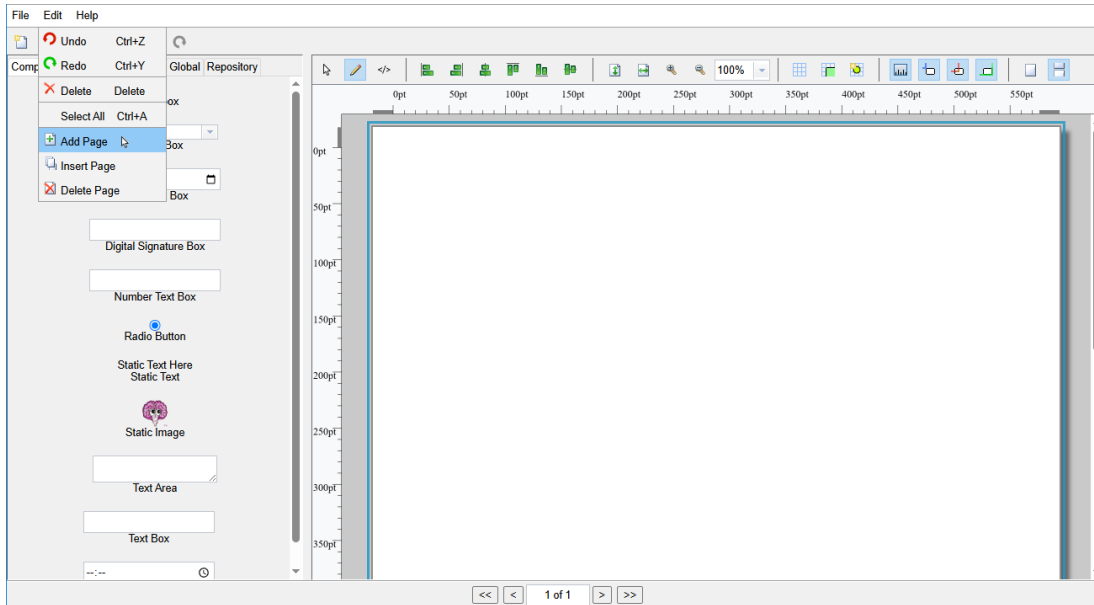


4.1.4. Status Bar

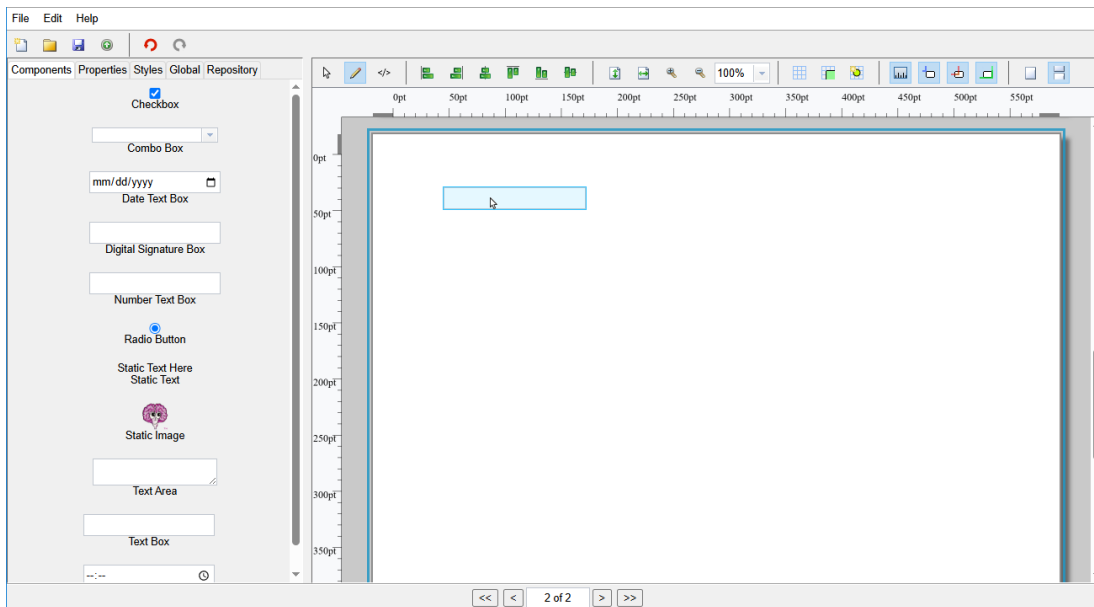
The status bar is the bottom bar of the editor. It is used to display status messages when the editor is busy doing things such as saving or loading a form, retrieving a list of forms from a repository, etc. The status bar can be very useful to watch the progress of actions that can take just a bit to complete. For example, if you are connecting to a forms repository over a slower internet connection that contains many forms, it could take several seconds longer to load.

4.2. *Creating a Form*

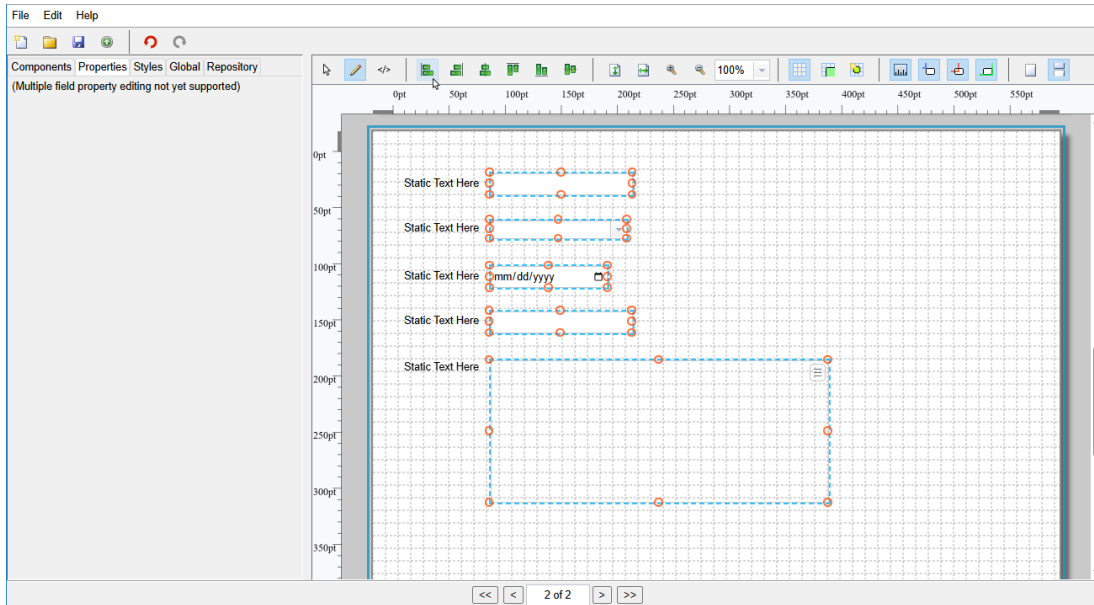
In this section, we will cover what you need to know to construct a basic form. Start by launching the editor. You will then be presented with a new empty form containing just one page. We'll start by adding a second page to the form.




Now we have a form containing two pages. We'll start by adding a few components to the first page. Left-click and hold on the Text Box component on the Components tab, drag it over the page, and release the mouse button to drop the Text Box onto the page.

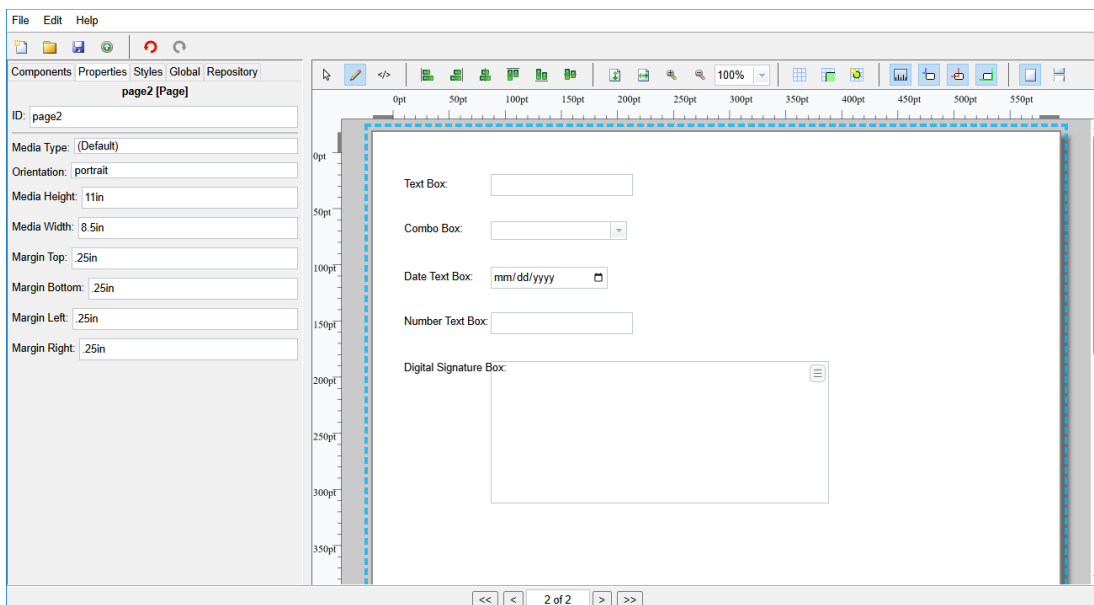


Continue to add a few other components. For this example, we'll add a Combo Box, Date Text Box, Number Text Box, Digital Signature Box, and Static Text components in front of each we'll use for labels.

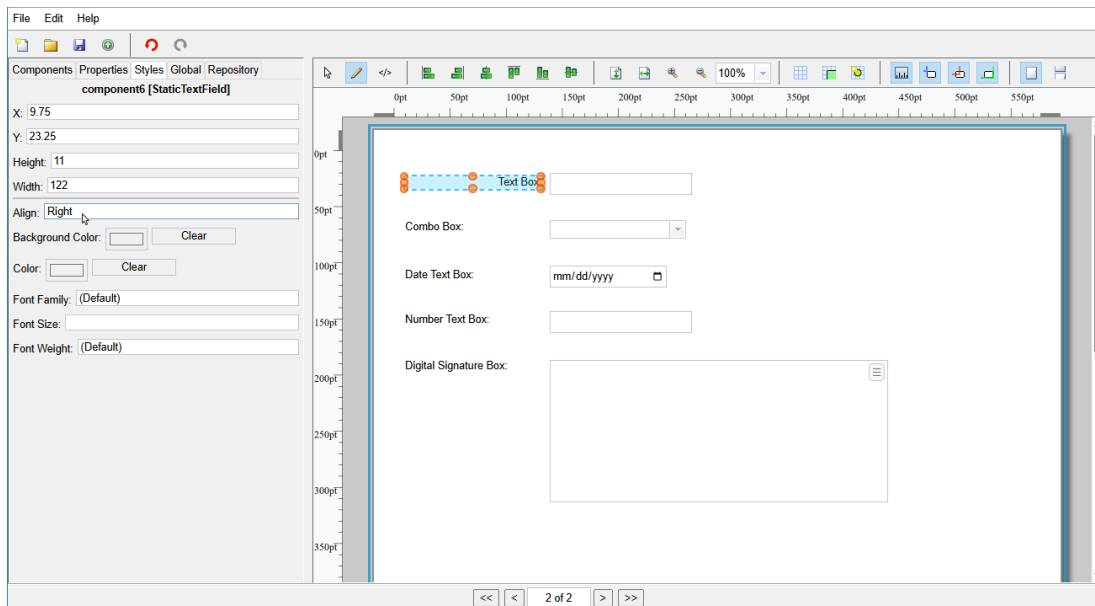


We've arranged the components mentioned above top to bottom with the Static Text labels in front of them. To get all of the components perfectly aligned to the left we could have manually adjusted the X location of each under the Properties tab, but instead we selected all of the components and then clicked the left align button . This will adjust all of the components to perfectly align to the left. We then performed the same action to all of the Static Text labels.

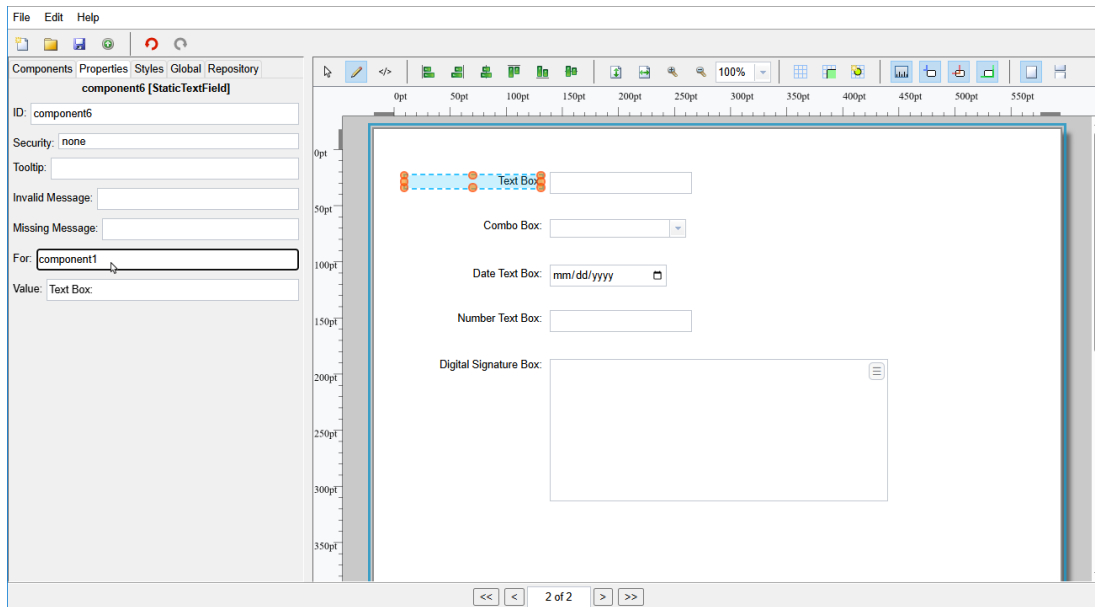
Now let's set the text on each of the Static Text labels. To do this, we'll select a label, then under the Properties tab, we'll set its value. For each label, we'll just set the type of the component as the value.



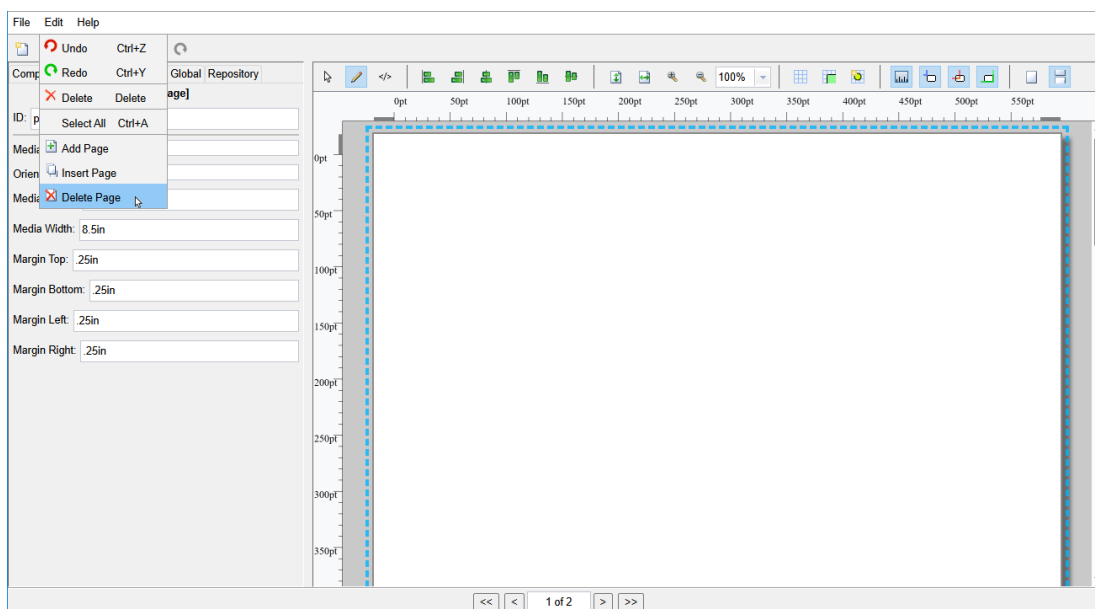
Now that we have all of the labels setup, we can see that the alignment is off because the text is different lengths, and left aligned. To correct this, we'll scoot all of our components to the right just a bit to make room and then right align the text of all of the labels so the text is justified to the right (the trailing) side of the component. To do this click a label, then under the Styles tab select Align Right. Repeat the steps for the remaining labels.



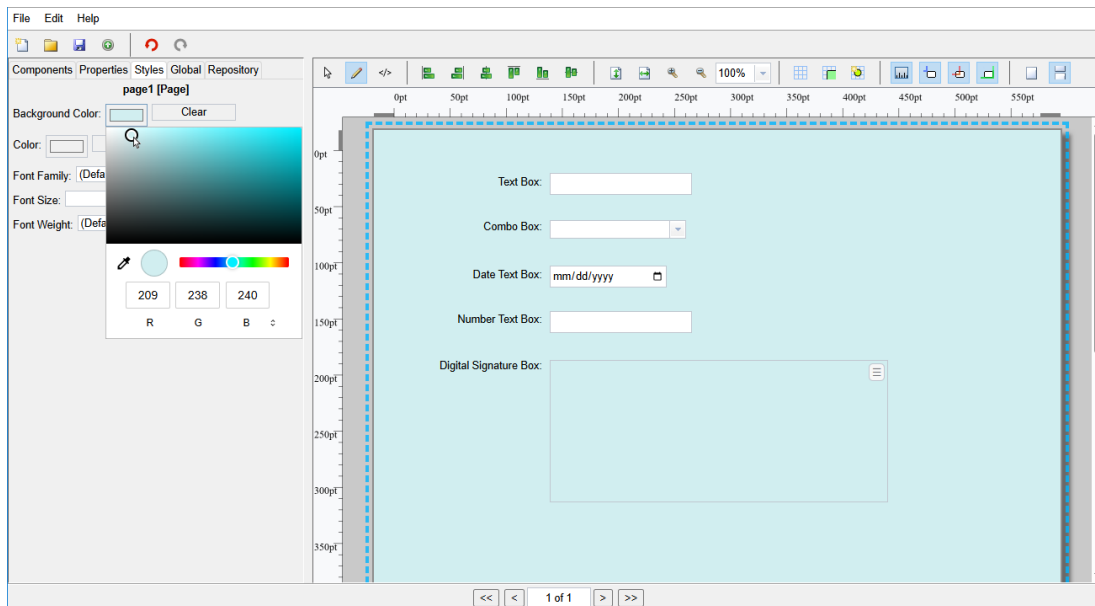
We should now have all of the labels nicely right aligned to the left of their corresponding component. The last thing we should do since we are using Static Text components as labels, is set each of the Static Component's For field with the ID of the component they are labeling. This will set the for attribute of the corresponding HTML label when the form is rendered to aid in accessibility and the use of screen readers. To do this, first click an input component to view its ID under the Properties tab. We can then copy the ID of the component and set it as the For value under the Properties tab for the corresponding label.



Repeat these steps for the remaining labels. Now all of the labels should have their For values set to match the ID of the input component they are labeling. Static Text components can be used for other reasons such as instructions, page titles, etc. and in these cases don't need their For values set. If you're following along constructing your example form using this tutorial, if you look at the bottom, you'll notice we've been adding components to the second page. At the beginning, when we added a second page to the form, the editor automatically switched to the second page. To correct this, we can simply scroll to the first page, and then delete the page using the action under the Edit menu.

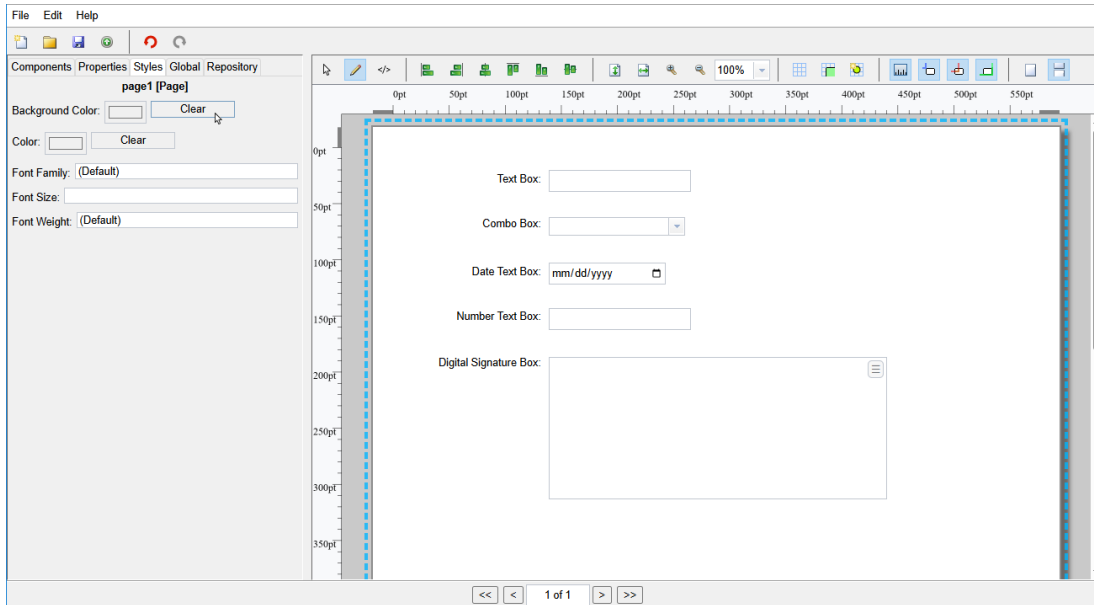


After deleting the first page, we should be back to a single page form. This is fine for the remainder of the example, but this shows how additional pages can be added, removed, and inserted if needed. We'll now play around with styling just a bit. We'll start by setting the background of the page to be a light blue color. To do this, click the page and then click the Styles tab. On the Styles tab we'll click the color selection box next to Background Color and choose a light blue color.

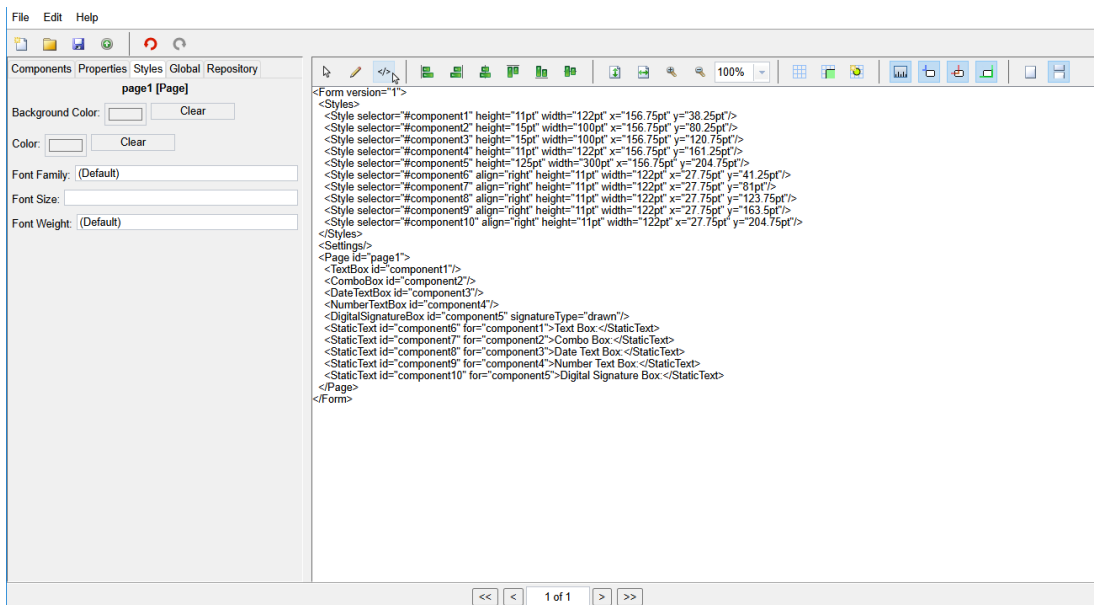


Now the background color of the page will be a light blue. Notice on the Styles tab, we can perform other page level styling like setting the default font used on the page, the size of the font, the default foreground color, etc.; these styles will only apply to this page. Additionally, a component on the page can override the default page color, font size, etc., by selecting the component, going to its Styles tab, and setting its color, font size, etc. The order of precedence in styling is similar to the order of precedence in CSS used for HTML. Global styles under the Global tab apply to everything, but can be overridden by page level styles and component level styles. Page level styles apply only to the page and its components, but components on the page are free to override the page and global styles if needed.


After seeing the light blue background color, we decide that plain white was better so we select the page again, click the Styles tab, and then click the Clear button next to Background Color.



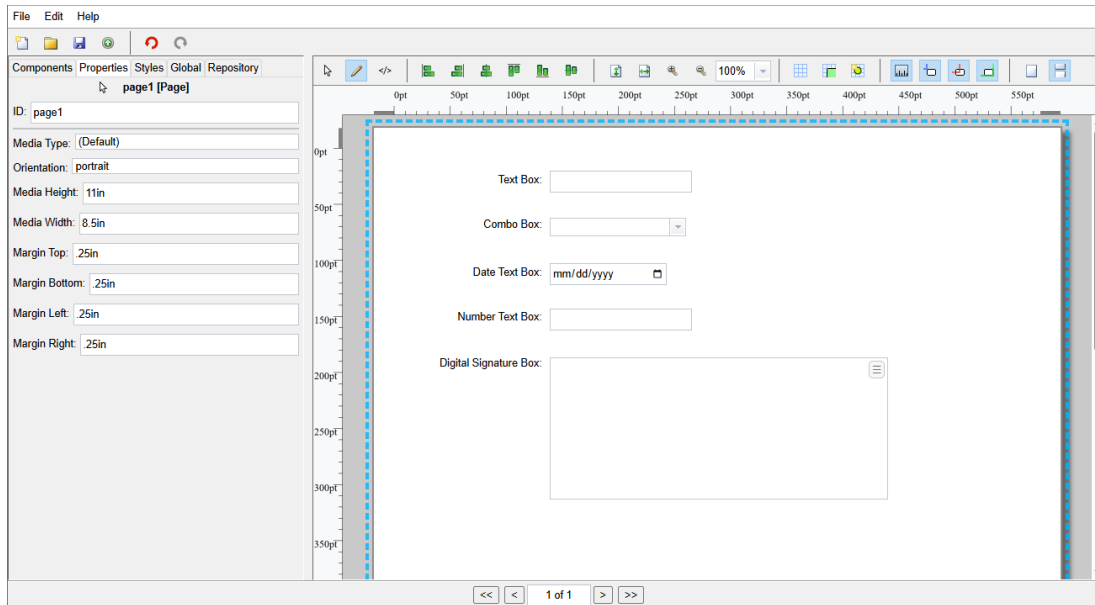
We are now back to a plain white background color. This demonstrates how to use some of the styling and the styling that is available. To view the XML code we have generated so far, we can click the XML mode button `</>` on the toolbar.



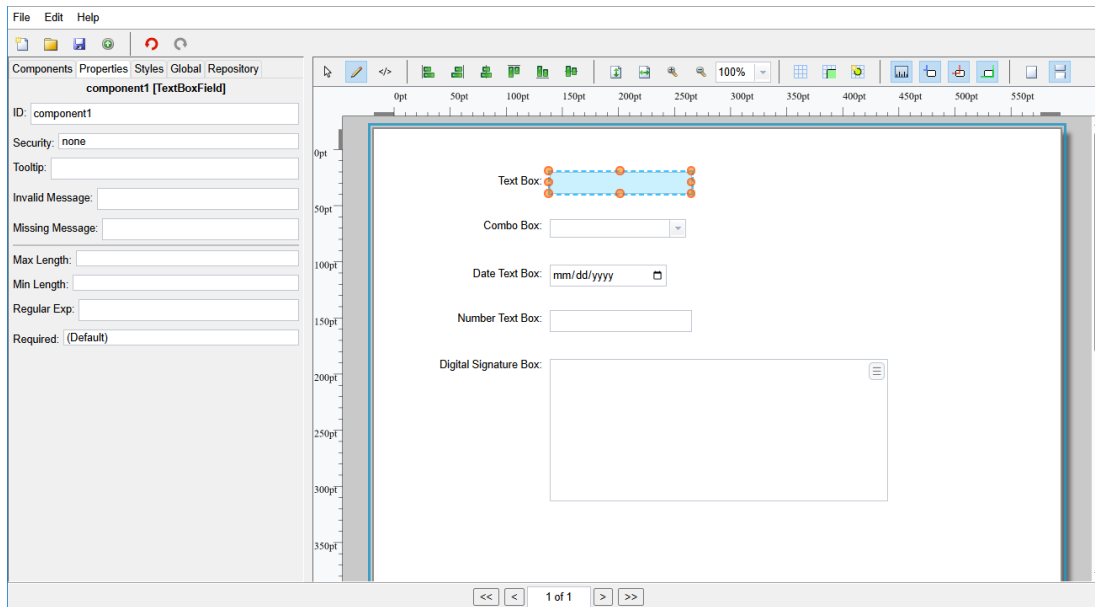
The XML code mode is mostly used for debugging purposes. We can view the various lines displayed. We can see each of the `<Style>` elements for each of the components; these style lines notate the X and Y position of the top left corner of each of the components related to the top left of the page. We can also see the `<Page>` element for


our one page, and each of the component elements for each of the components we placed on the page. If you're ever curious as to the XML being generated or need to view the inner specifics of a form, you can always click to XML mode to see what's going on. When you're ready, switch back to edit mode by click the Edit mode button .

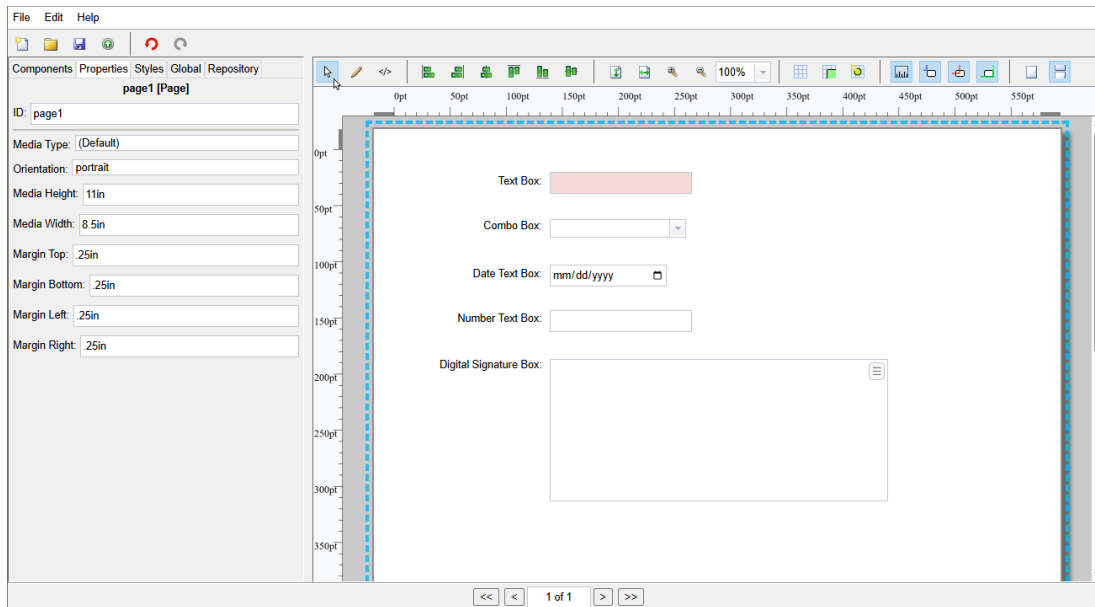
Let's now dig into the properties of the page and the various components we placed on the page. Start by clicking the page, and then click the Properties tab.



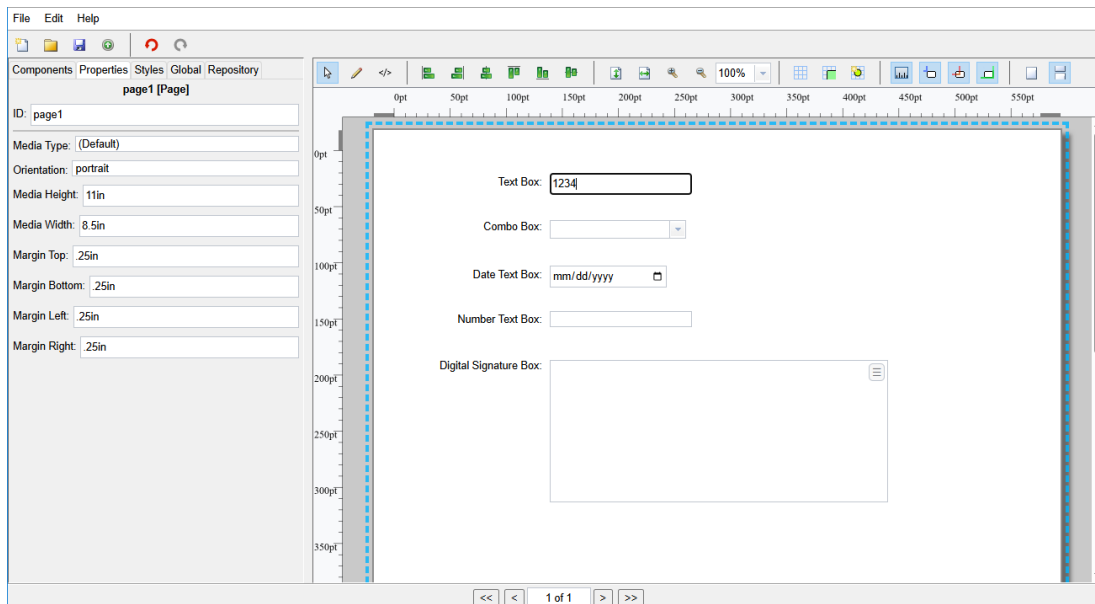
On the Properties tab for the page, we can adjust things like the ID of the page, the media type, orientation, and the physical size and margins of the page. The [DocuBrain Electronic Forms Specification](#) supports a very large number of built-in media types. Nearly all of the common US and European media sizes are available of use as well as many others. On the off chance, you need an unlisted media size, you are always free to plug in the physical size of the media in inches or centimeters and the corresponding margins typical for that media type. Additionally, we can set the orientation of the page if needed. Moving on, we'll click the Text Box component on the page to view its properties.



The Properties tab for the Text Box allows you to change the ID of the Text Box just like the page; the remainder of this tab is different. We can adjust things like the security on the field (who can input/change the contents of this field when submitting), the tooltip (tooltips are shown when we select/hover over a component), and the invalid and missing messages (these are part of validation when validation isn't met). For this Text Box component, we're going to set a few properties and then flip to normal mode to test them in action. Let's set an invalid message of "Text must be between 4 and 6 characters long" and a missing message of "Text must be specified". Moving on, set the Max Length to 6, the Min Length to 4, and mark this component as required (meaning it can't be left blank). Now we can switch to normal mode by clicking the Normal mode button  on the toolbar.

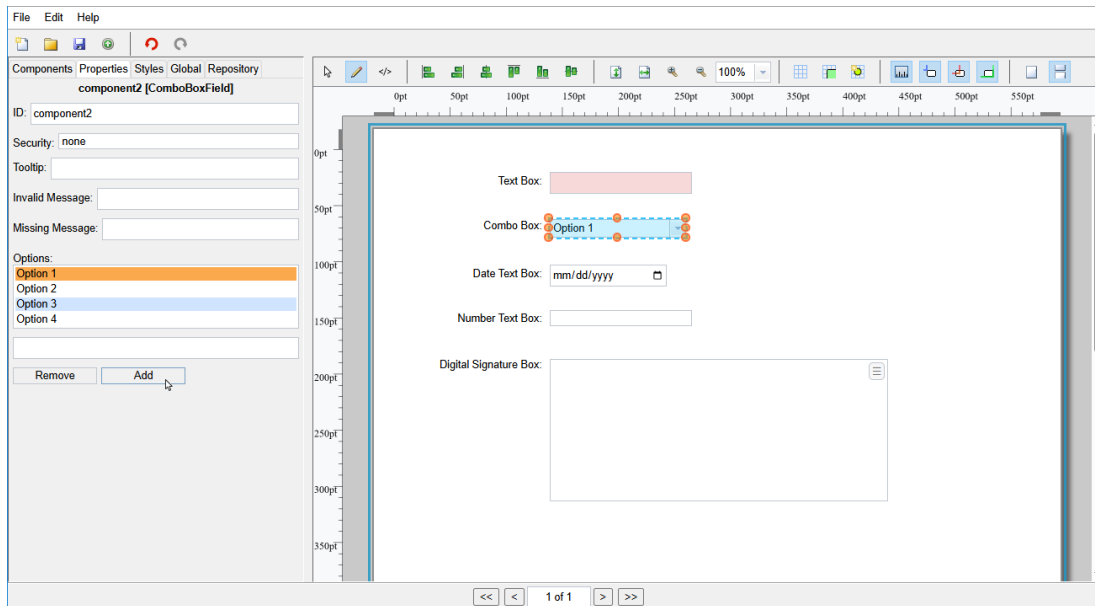


Notice that the Text Box component appears red; this is because the Text Box does not currently pass validation as we stated there must be between 4-6 characters in this component. When the user is completing this form, they will see our validation error message instructing them as to what's required. If 4 characters are input, we'll see the box turn white again because it has passed validation.

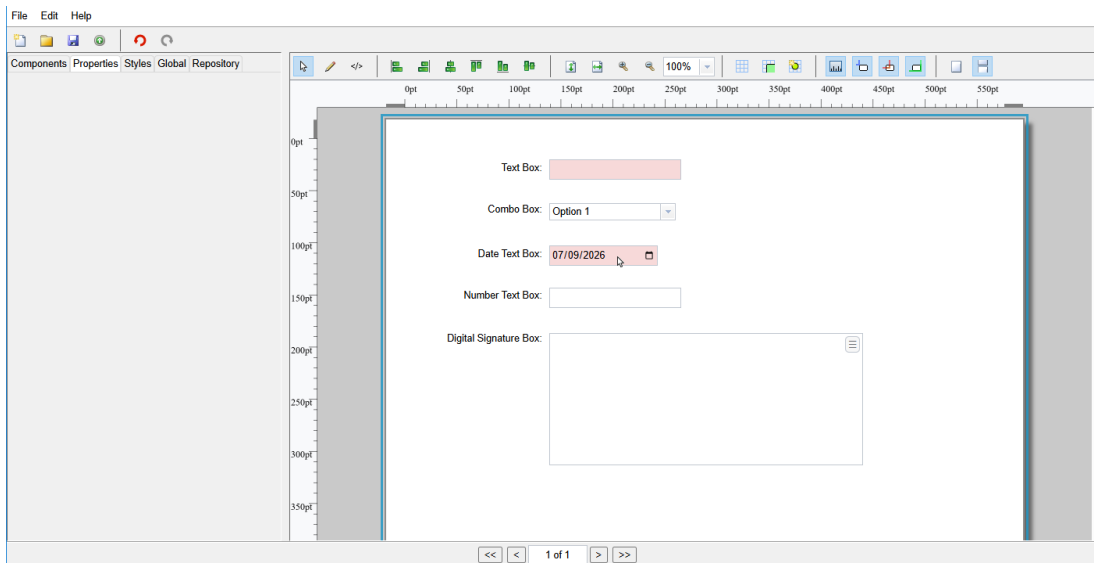


It's important to note, while normal mode is handy for testing our components, text and settings made in normal mode will be saved if not reverted. For example, if we leave the

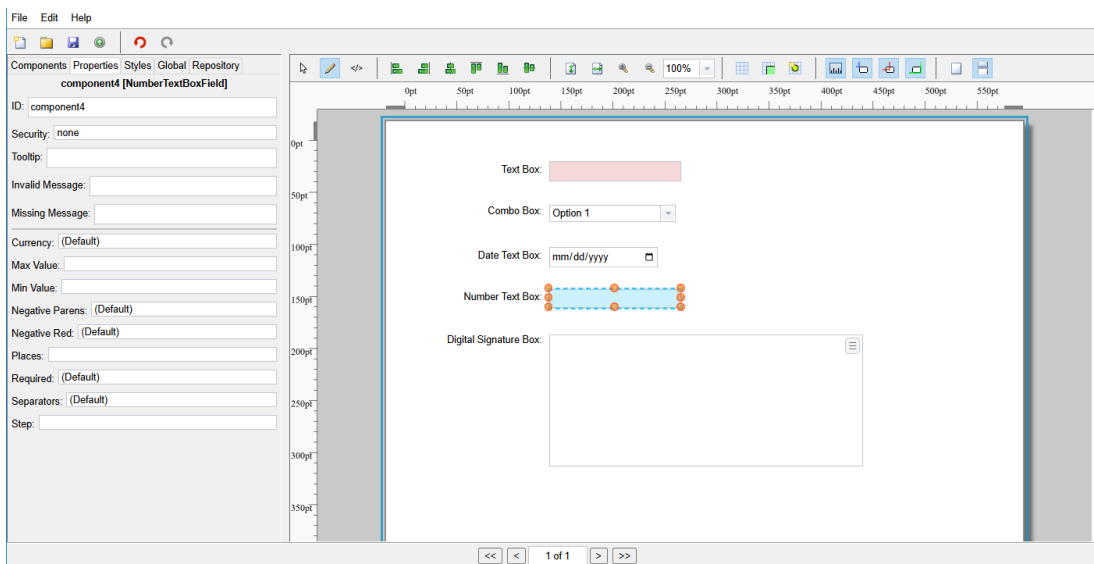
text “1234” in the Text Box component and save the form, “1234” will be the default value for this component. This mechanism works this way by design so that default values can be set if needed. We’ll remove the “1234” characters from the Text Component and then flip back to edit mode to view the properties on the other components. Next click the Combo Box component to view its properties.



The Properties tab for the Combo Box component lists many of the same common attributes as it did for the Text Box component however, for this component instead of Max Length, Min Length, etc., we have the ability to build the options for the combo box. Using the field and Add button, add four options named “Option 1”, “Option 2”, “Option 3”, and “Option 4”. We now have four options for our combo box. If needed, we could flip to normal mode and pick one of the four options as the default value for this field. Moving on, click the Date Text Box so we can view its properties. We’re going to set a Min and Max value for the date range we’ll allow. Using the Max value field, set a value of 12/31/2025, and place a value of 01/01/2025 into the Min value field. This will force the user to pick a date in the year 2025 if they enter a date. We’ll leave required turned off. This will result in the field always appearing white (because we didn’t require the user to enter a date) however, if they do enter a date, it must be within the specified range. If they enter a date outside of the range, the date will turn red.

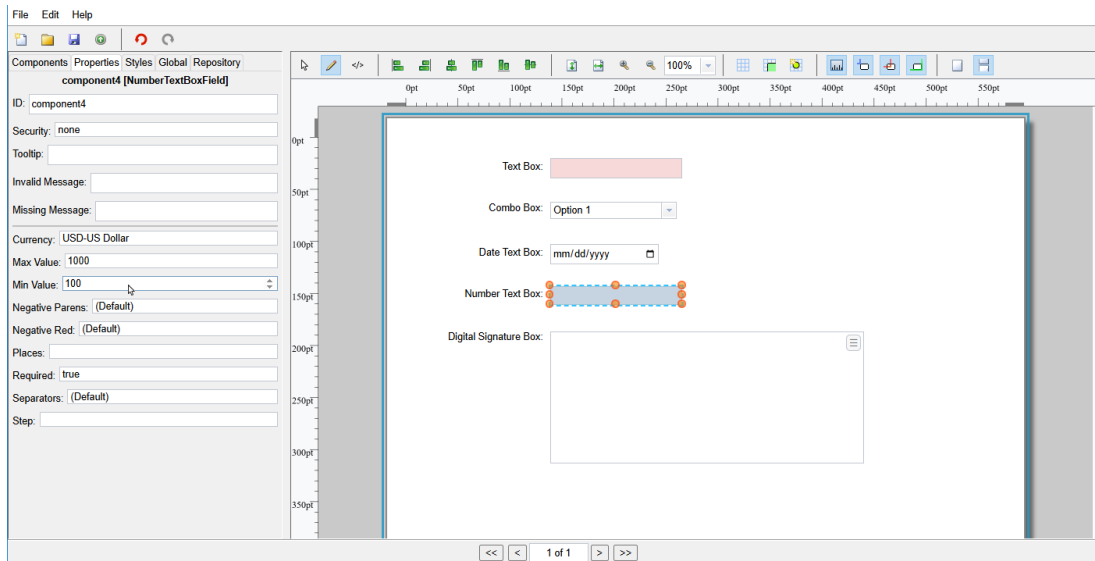


If you wish, flip to normal mode and enter a date outside of the specified range to see that it turns red. After doing this, you can remove the date by clicking the numbers entered and pressing the backspace key. Now flip back to edit mode and click the Number Text Box component to view its properties.



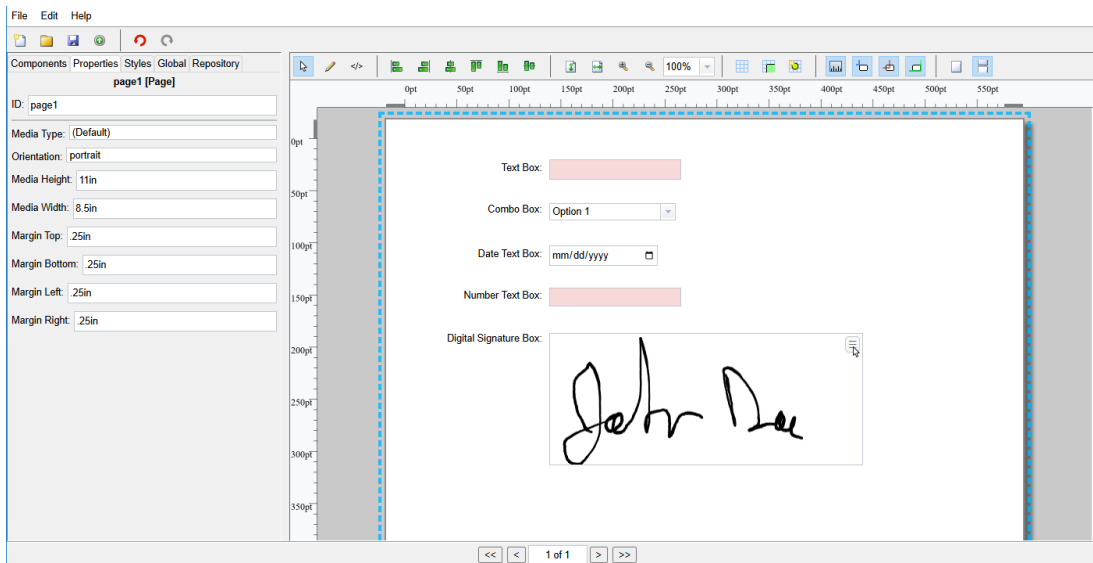
Number Text Box supports a wide variety of properties since it can function as a plain number input field, or a currency input field. Some accounting uses of this field may require the field to turn red or use parenthesis when this field is negative. For this example, we are going to make this a USD based currency by selecting USD-US Dollar from the Currency field, mark it required, and set a min and max value of 100 and 1000

respectively. We only want to specify the number without the currency sign when setting min and max values.

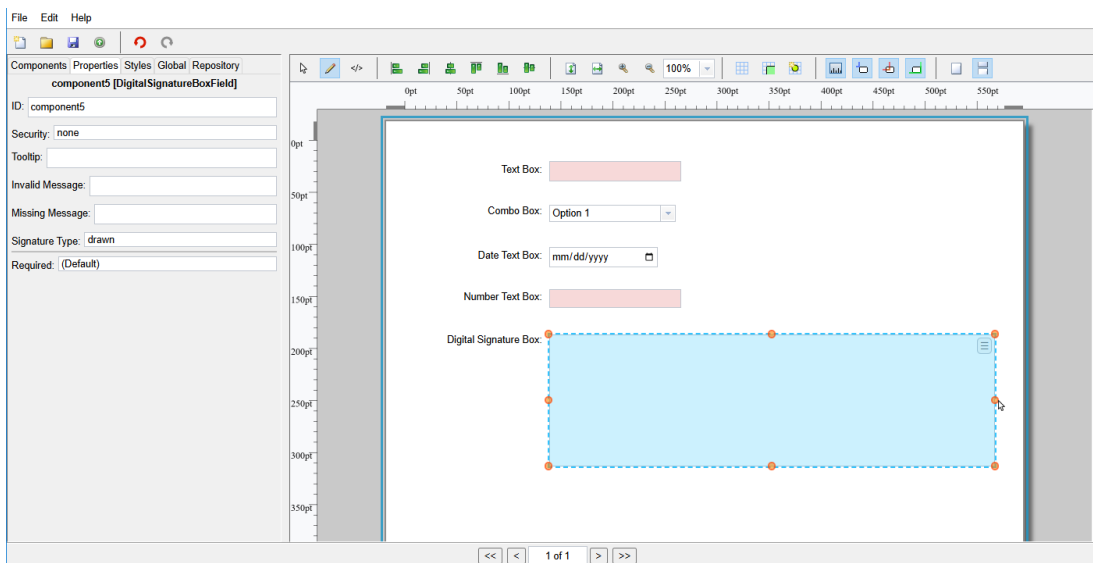


Lastly, let's experiment with the Digital Signature Box. When we click the Digital Signature Box in edit mode, we see its properties. Aside from the common properties, the Digital Signature Box has just four choices: drawn, image, drawnOrImage, and typed. This allows us to accept digital signatures in a few different ways. Both drawn and drawOrImage allow the user to draw their signature with the mouse or their finger on a touch device. Image and drawOrImage support the user uploading a picture of their signature, a driver's license photo, etc. Finally, typed is used to simply allow the user to type their full name. Typically, when a Digital Signature Box is used, a block of instructions should be placed above it telling the user what to do. Additionally, a checkbox can be added to say "By clicking this box, I agree to..." and this box can be marked required so they are required to accept the terms before submitting the form.

We'll leave the signature type set as drawn and flip to normal mode. Here we can experiment with the signature component and see if it is of appropriate size for the intended use. For example, initials require much less room than a full legal signature.



Notice if we draw out the generic signature “John Doe”, it just barely fits. We should probably make the signature field just a bit longer. Click the button in the upper right-hand corner of the Digital Signature Box component and choose the Clear option to clear the signature we drew. Then we can flip to normal mode, select the Digital Signature Box, and then drag the right side of it to make it longer.




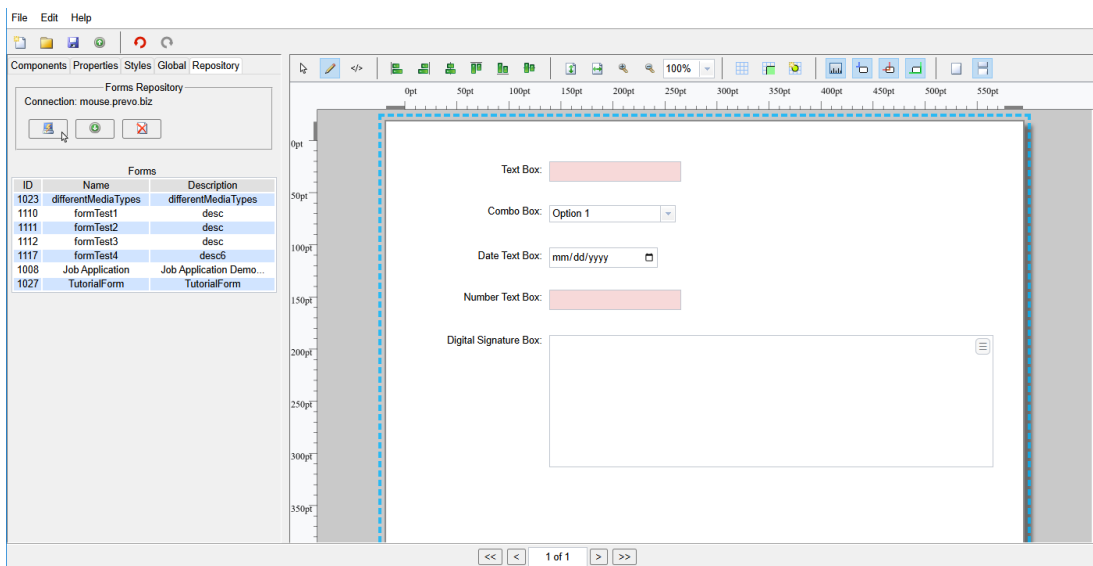
We now have a basic form that we can deploy to the TechDoc Document Manager Forms Engine and test out. In the follow sections, we'll cover the steps to connect to the Forms Engine repository, upload/publish this form, as well as download and delete existing forms. The editor features many other component types, properties, and styles

that can be manipulated as well as other functionalities like undo/redo, grid snapping, guide lines for alignment, etc. Feel free to experiment further with all of the options and tools that are available.

4.3. Connecting to the Forms Engine Repository

The DocuBrain Forms Editor has the ability to connect directly to a TechDoc Document Manager's Forms Engine to deploy and publish forms and edit existing forms.


Let's start by clicking the Repository tab on the Left Panel and then clicking the connect button .

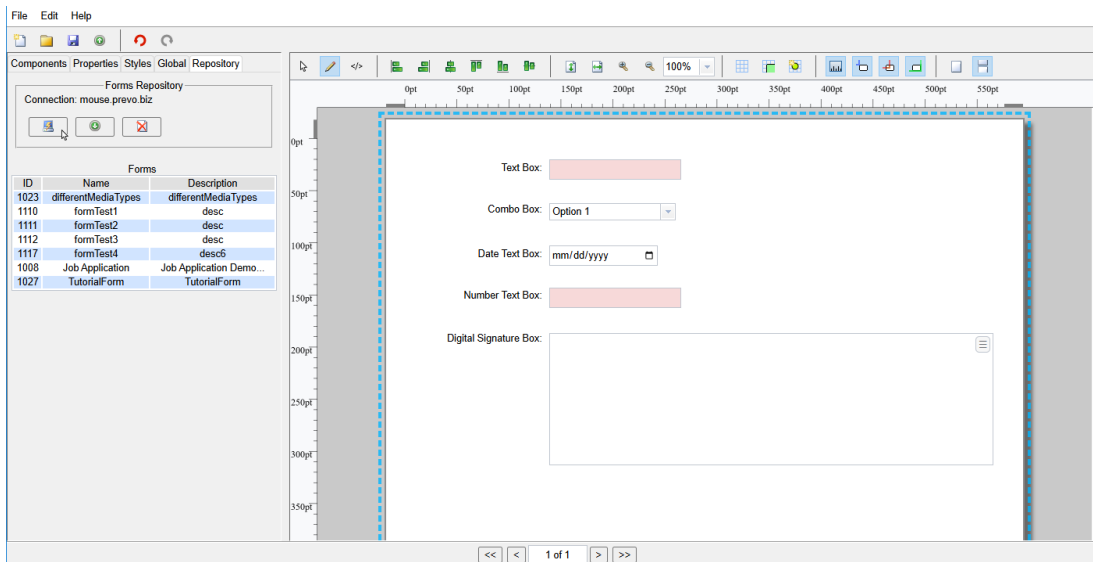



When the connect button is clicked, the Forms Editor connects to the Forms Engine in the Document Manager and downloads a list of all of the forms you own or have owner access to. These forms will be listed on the left panel with their IDs, Names, and Descriptions.

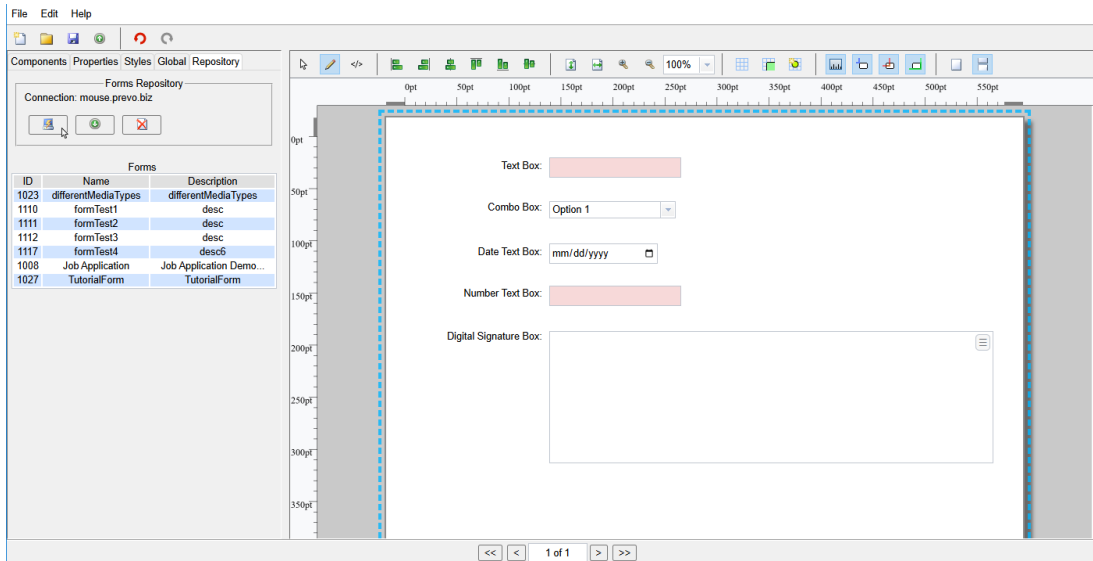
4.4. Uploading a Form to the Forms Engine Repository

In this section, we will cover uploading a form to the Forms Engine repository. Locate and open a form to upload or use the example form created in the [previous section](#). To open a form in the editor, click Open on the File menu. On the Open Form dialog, navigate to the form, select it, and then click the Open button. Once a form is open in the editor it may be uploaded to the Forms Engine repository.

On the main editor window, click the Repository tab on the Left Panel and then click the connect button .

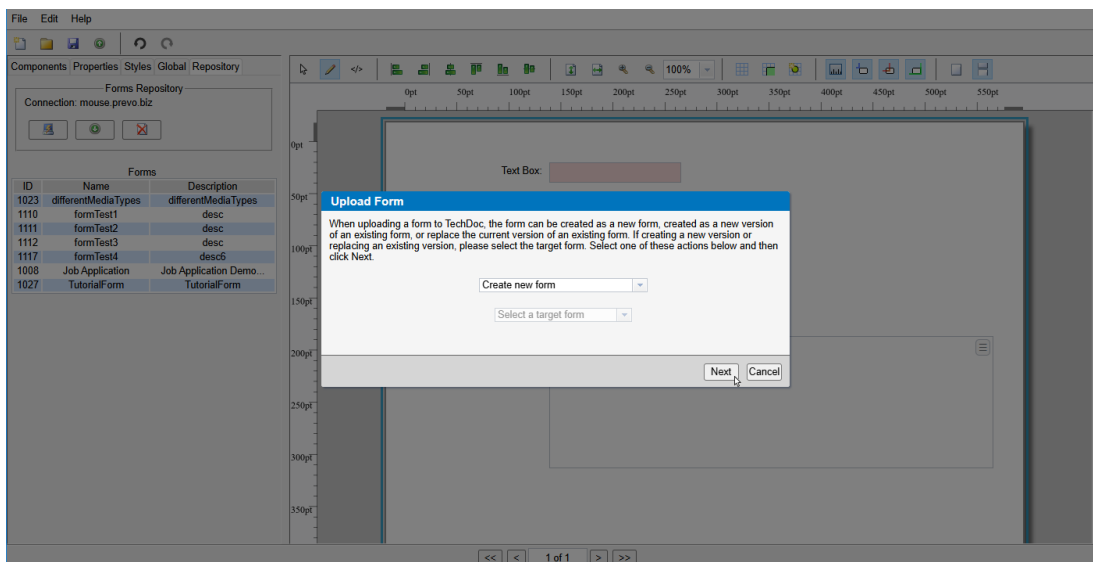


When you connect, the editor will download the information for all of the forms in the Forms Engine repository that you own or have owner access to. If the Forms Engine does not yet contain a form, the list will be empty. Now that we have connected to the TechDoc Forms Engine repository, let's upload our form. To do this, simply make sure your form is loaded in the editor and then click the upload button  on the main toolbar or select Upload Form from the File menu.



The editor will then ask you a few questions to see if you want to create a new form in the Forms Engine, or replace an existing form. For this example, we'll be creating a new form so we'll select "Create New Form". You can replace an existing form if needed; this allows you to either replace the existing form's latest version or replace an existing form with a new version (if you wish to keep the current version). As the Forms Engine supports versioning, it may be helpful over time to keep multiple versions of a form as needs change and forms are amended.

Make sure "Create New Form" is selected, and then click the Next button.



On the next screen, we'll need to fill in all of the typical information for a document in TechDoc. Since forms are first created as a document (to support history, versioning, change tracking, attachments, etc.), we'll need to fill in the information for the form's document object. For this test, we have created a root level cabinet called /Forms, but you can create the form's document in your Document Manager anywhere you like and have access to. Additionally, we need to give the document a number, description, and choose the category and type. We'll let the document default to using our organization and stick with a web search of no for now.

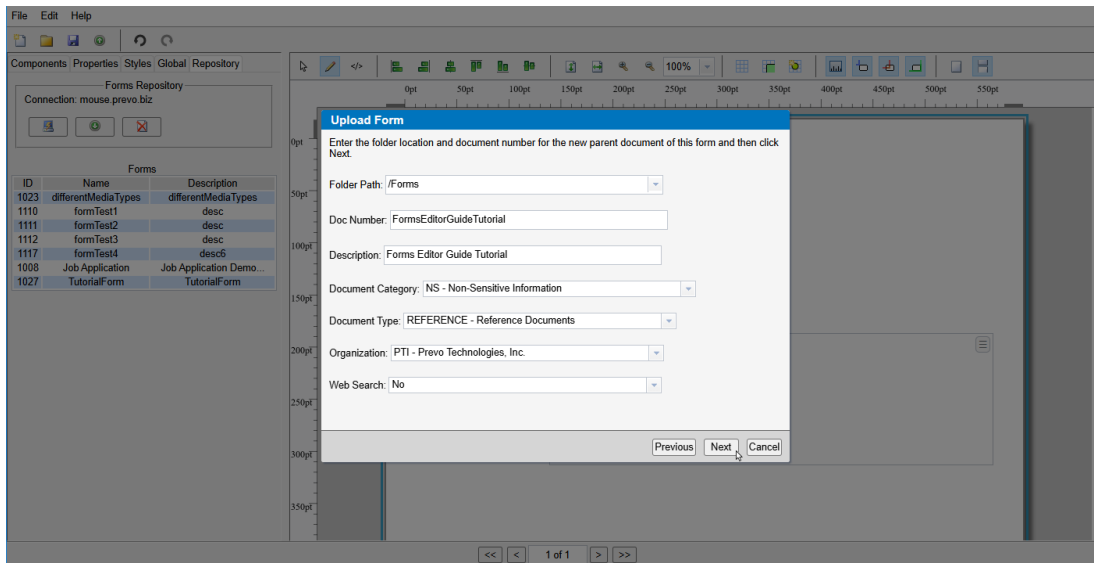
The screenshot shows the 'Upload Form' dialog box in the TechDoc Forms Editor. The dialog contains the following fields and options:

- Folder Path:** /Forms
- Doc Number:** FormsEditorGuideTutorial
- Description:** Forms Editor Guide Tutorial
- Document Category:** NS - Non-Sensitive Information
- Document Type:** REFERENCE - Reference Documents
- Organization:** PTI - Prevo Technologies, Inc.
- Web Search:** No

At the bottom of the dialog are buttons for 'Previous', 'Next', and 'Cancel'. The 'Next' button is highlighted. In the background, a table titled 'Forms' is visible with the following data:

ID	Name	Description
1023	differentMediaTypes	differentMediaTypes
1110	formTest1	desc
1111	formTest2	desc
1112	formTest3	desc
1117	formTest4	desc6
1008	Job Application	Job Application Demo...
1027	TutorialForm	TutorialForm

Finally, enter a reason for uploading this form and creating its document, and then click the OK button.





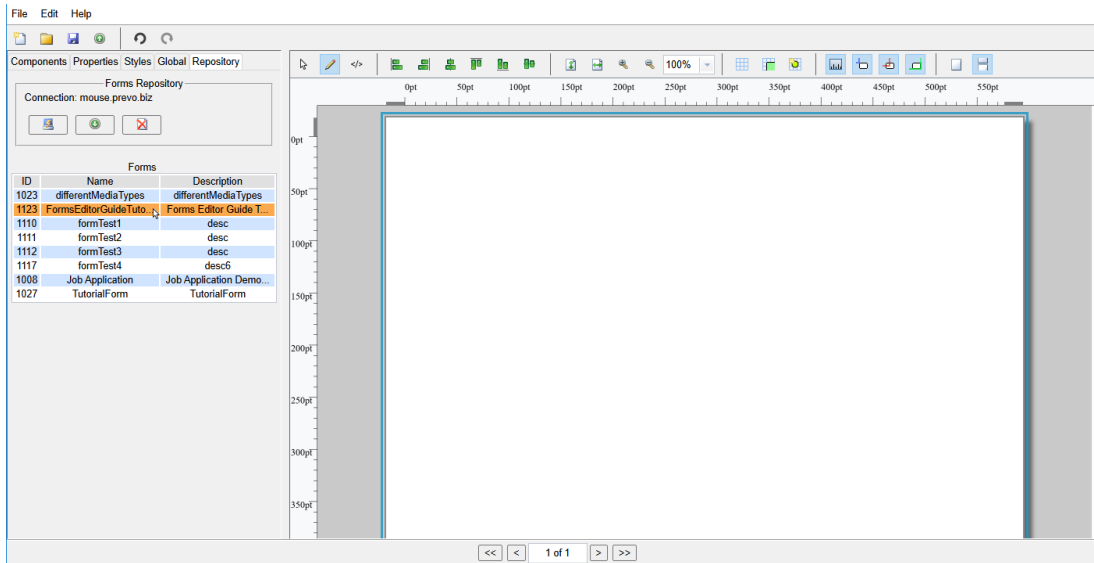
The form has now been uploaded to the Document Manager's Form Engine. The form can be viewed by logging into the DM, visiting the Forms area, and navigating to the form using whatever document number you entered. The form may be viewed by clicking the View link on the side menu. If the form looks as it should and is ready for submissions, the form must be modified and made available. Click Modify on the side menu and set Available to Yes. Make any other changes you need and then click OK.

Now that the form is available, you should be able to click Create Submission on the side menu and create your first test submission of the form. The remainder of the server-side reference can be found in the [TechDoc Server-Side Forms Reference](#) section below.

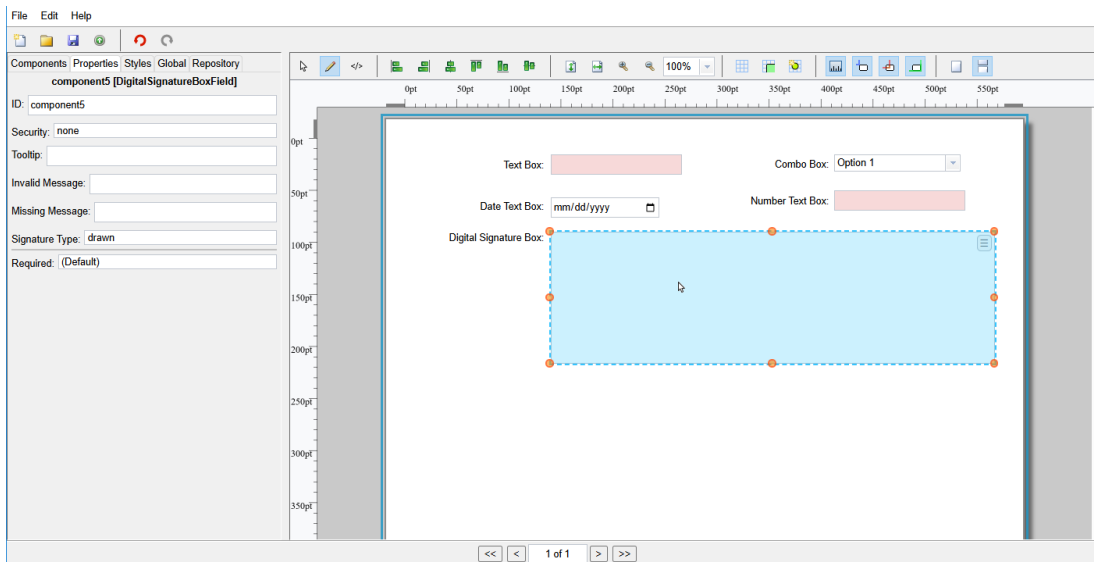
4.5. Downloading from a Form from the Forms Engine Repository

The DocuBrain Forms Editor allows you not just to upload to the Forms Engine, but also download a form, edit it and then upload it back. In order to download from a repository, you must own the form you are trying to download or have owner access to it.

First, we need to connect to the repository. Click the Repository tab on the Left Panel, and then click the connect button . After the editor connects to the repository it will return a list of the forms in the repository that you have access to. Choose the form from the list displayed that you wish to download, and then double click it or click the download button .



Once the form download has completed, it will be loaded into the editor where you can then view and/or edit the form. For this example, we will use our example form that we created in a [previous section](#). We've downloaded the form back from TechDoc and we've moved the components around so that they are now oriented side by side.



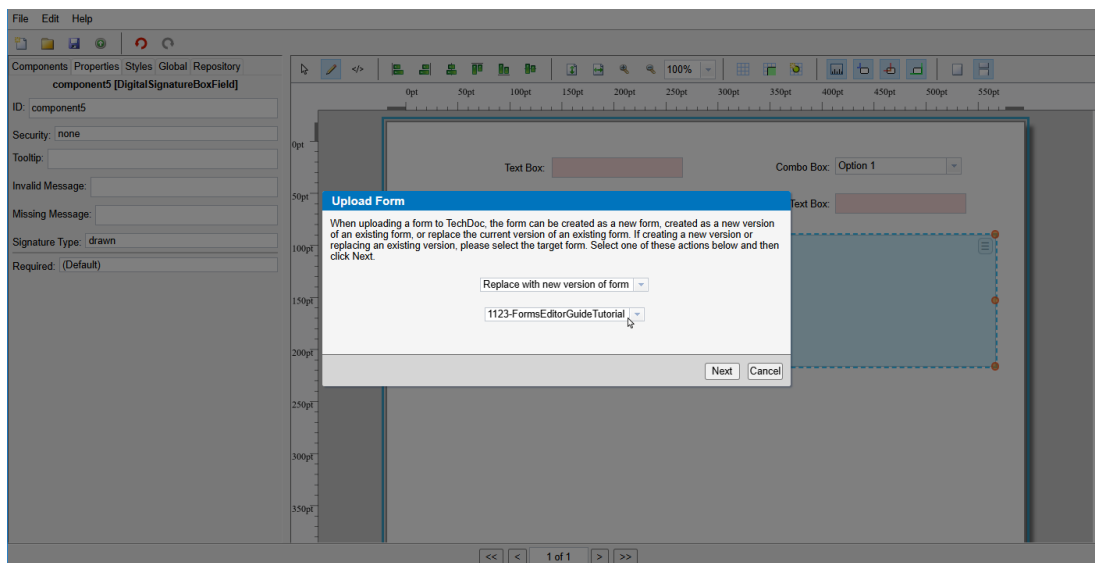
We have completed the changes we wanted to make so we have gone ahead and saved it locally. You do not have to save a form locally that you have downloaded and modified from a repository. You are free to pull down forms, make changes, and then send them right back.

Now we will upload this form to TechDoc, and we have a couple options:

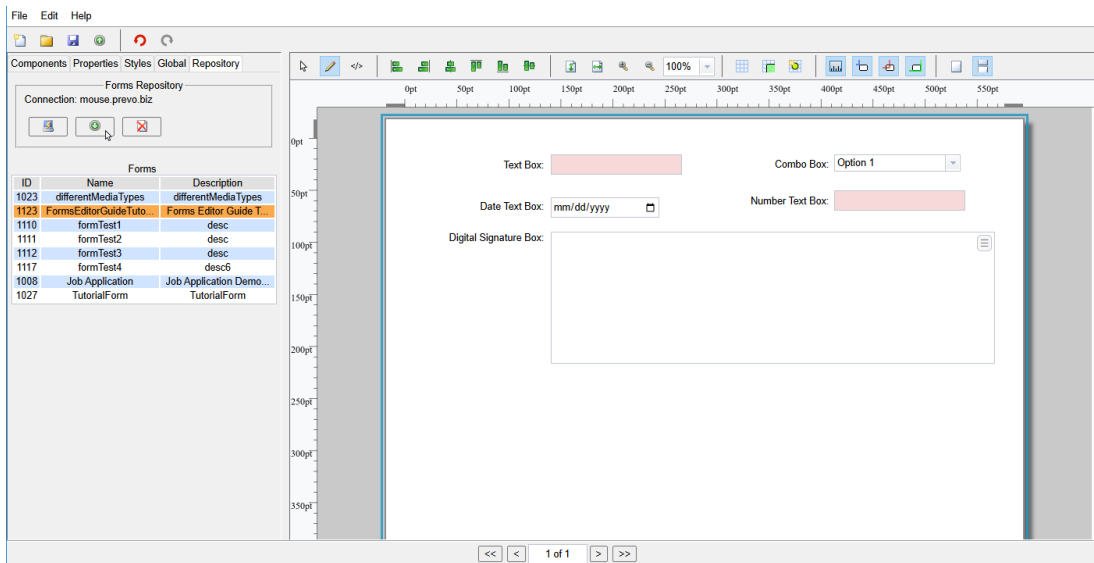
- We can delete the original form from the Forms list on the Left Panel and upload this form using the “Create new form” option. This will get rid of the old form and this form will become the new version 1 of the form.
- For the second option, we can just leave the original version of the form and select “Replace existing version of form”. This will accomplish the same as the previous bullet; the old version of the form will be gone, and this will become the new version 1 of the form.
- For the third option, we can leave the original form untouched in the repository and upload this form leaving the original version 1 and creating this modified version as version 2. To do this, we select “Replace with new version of form”.

You will have to decide what is right in your case. It best to only replace forms that are not currently available and in the middle of receiving submissions; you are however free to do so. If a user is in the middle of a submission and they attempt to submit their completed form and the form has changed, they may receive errors if fields have changed, validation rules have changed, etc. Because of this, it is best to mark a form as not available, and then replace the form. A better option may be to replace the form as a version 2. This will allow the current users to complete their submission on the version 1 form and new submitters can begin completing version 2. Of course, you are also always welcome to just upload this as a new form all together as well.

For this example, we will go ahead and upload the modified example form as a version 2 leaving the original version 1 of the form in place. Select “Replace with new version of form”, select the original form from the list, and then click Next. Enter a reason and then click OK.



We've now uploaded this form and created a version 2 of the original form. Now if we download the form from the Repository tab, we will begin getting the new version 2 of the form.



5. DocuBrain Electronic Forms Specification

This chapter covers the DocuBrain Electronic Forms Specification supported by TechDoc. The specification has been designed to be human readable and hand codable (when needed), and as such has been implemented using XML (Extensible Markup Language). The specification consists of a root tag called Form with three main sections: Styles, Settings, and Pages. A form must contain a single root element called Form.

```

1 <Form version="1">
2   <Styles>
3     ...
4   </Styles>
5   <Settings>
6     ...
7   </Settings>
8   <Page>
9     ...
10  </Page>
11 </Form>

```

The Form element must contain the attribute “version” specifying the version of the DocuBrain Electronic Forms Specification the form uses. At the time of the writing of this guide the current forms version is 1. The Form element has no other attributes at this time.

5.1. Styles

The Styles element is used to style the electronic form and styles can be applied globally, to all form components of a specific type, or to individual pages or components by their ID. Each style must be enclosed in a Style element under Styles.

```

1 <Form version="1">
2   <Styles>
3     <Style selector="*" color="#0000FF" />
4   </Styles>
5   <Settings>
6     ...
7   </Settings>
8   <Page id="page1">
9     ...
10  </Page>
11 </Form>

```

The Styles element has no attributes of its own.

5.1.1. Style

The Style element is used to specify one or more styles at: the global level, to a specified type of form component, or to a specific page or component using their ID.

```

1 <Form version="1">
2   <Styles>
3     <Style selector="TextBox" backgroundColor="#00FF00" align="center"/>
4   </Styles>
5   <Settings>
6     ...
7   </Settings>
8   <Page id="page1">
9     ...
10  </Page>
11 </Form>

```

The Style element may use as many attributes as needed and supports the following attributes:

- selector (required) – The selector attribute is used to specify the target or targets for a style. The selector attribute may have only one of the three types:
 - * – The star selector means global and the style should apply to all pages and components in the form.
 - TYPE_HERE – The type selector is used to specify the component type this style should apply to. Replace TYPE_HERE with Page or the type of component the style should apply to such as TextBox to target all text boxes.
 - #ID_HERE – The ID selector is used to specify a specific component or page by its ID. Replace ID_HERE with the ID of the page or component to target preceded by a #.
- align (optional) – The align attribute is optional and may be used to specify the alignment of the contents of a component. For example, align="right" can be specified on a TextBox to align the text in the text box to the right instead of the default left. Align may have one of the three types:
 - left – Meaning to align to the left.

- center – Meaning to align to the center.
- right – Meaning to align to the right.
- backgroundColor (optional) – Used to specify the background color of the component. Background color must be specified using HTML style color codes such as #FF0000 to indicate full true red.
- color (optional) – Used to specify the foreground color of the component. Color must be specified using HTML style color codes such as #00FF00 to indicate full true green.
- fontFamily (optional) – Used to specify the font family to use for a component. Once of the following font family strings may be used:
 - Arial, Helvetica, sans-serif
 - Verdana, Geneva, sans-serif
 - Courier New, Courier, Monaco, monospace
 - Georgia, serif
 - “Palatino Linotype”, “Book Antiqua”, Palatino, serif
 - “Times New Roman”, Times, serif
 - Arial, Helvetica, sans-serif
 - “Arial Black”, Gadget, sans-serif
 - “Comic Sans MS”, cursive, sans-serif
 - Impact, Charcoal, sans-serif
 - “Lucida Sans Unicode”, “Lucida Grande”, sans-serif
 - Tahoma, Geneva, sans-serif
 - “Trebuchet MS”, Helvetica, sans-serif
 - Verdana, Geneva, sans-serif
- fontSize (optional) – Used to specify the size of the font in relation to the normal size 100. If you wanted the font of a component to appear twice as big as the standard size you would specify 200.

- **fontWeight** (optional) – Used to specify how thick or thin the characters of a font are displayed. Currently the values normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, and 900 are accepted. 100 specifies the thinnest, 400 is the same as normal, and 700 is the same as bold.
- **height** (optional) – Used to specify the height of a component in points.
- **width** (optional) – Used to specify the width of a component in points.
- **x** (optional) – Used to specify the X location of a component relative to the top-left corner of the page in points.
- **y** (optional) – Used to specify the Y location of a component relative to the top-left corner of the page in points.

5.2. *Settings*

The Settings element is used to store global settings for a form. Each setting should be contained in its own element under the Settings element.

```

1 <Form version="1">
2   <Styles>
3     ...
4   </Styles>
5   <Settings>
6     <SecurityAlias original="officeUse1" alias="Human Resources" />
7   </Settings>
8   <Page id="page1">
9     ...
10  </Page>
11 </Form>
```

The Settings element has no attributes of its own.

5.2.1. **SecurityAlias**

Currently the only setting available for use is SecurityAlias which is used to create an alias for one of the 10 office-use-only security levels that can be set on a field. These office-use-only security levels are officeUse1, officeUse2, etc. to officeUse10. A security level can be placed on a field to allow only certain users to enter/change the value of a field. To allow certain users or groups access to the protected field, the associate access command on forms can be used inside of TechDoc to grant those certain users or groups access to fields with the specified security level.

While aliases are not necessary, it is a good idea to use them to convey what each of the security levels mean to a particular form. If aliases aren't used, the original names officeUse1, office2, etc. will be used and shown for association in TechDoc. This can make it more difficult to remember what each level is for, so the use of aliases is highly recommended.

For example, if officeUse1 is being used to say only users/groups that are with Human Resources should be able to edit a field, you would want to enter `<SecurityAlias original="officeUse1" alias="Human Resources" />`. This way throughout the remainder of the form, any field needing this level of security can specify `security="Human Resources"` instead of having to specify `security="officeUse1"`. Additionally, when associating access on the form in TechDoc, "Human Resources" will be shown as the alias on "officeUse1" to better convey its usage.

```

12 <Form version="1">
13   <Styles>
14     ...
15   </Styles>
16   <Settings>
17     <SecurityAlias original="officeUse1" alias="Human Resources" />
18   </Settings>
19   <Page id="page1">
20     <TextBox id="textbox1" security="Human Resources"/>
21   </Page>
22 </Form>

```

SecurityAlias supports the following attributes:

- **original (required)** – The original attribute is required and should specify one of the 10 office-use-only security levels you wish to alias such as officeUse1. The levels available for use are officeUse1, officeUse2, etc. to officeUse10.
- **alias (required)** – The alias attribute is required and should specify the use of the security level.

5.3. Page

A Page element is use to define each page of the form. For example, if a form has three pages, there will be three Page elements under the Form element.

```

1 <Form version="1">
2   <Styles>

```



```

3    ...
4    </Styles>
5    <Settings>
6    ...
7    </Settings>
8    <Page id="page1">
9    ...
10   </Page>
11   <Page id="page2">
12   ...
13   </Page>
14   <Page id="page3">
15   ...
16   </Page>
17 </Form>

```

Each Page element will contain all of the fields/components used on that page. A Page element supports the following attributes:

- **id (required)** – Every page is required to have a unique ID. An ID must start with a letter (a-z or A-Z) and can be followed by letters, numbers (0-9), hyphens (-), underscores (_), colons (:), or periods (.).
- **marginBottom (optional)** – marginBottom may be used to specify the margin between the bottom edge of the page and the start of printable area. By default, this margin is set to .25in. A page margin may be specified using points (pt), inches (in), millimeters (mm), or centimeters(cm).
- **marginLeft (optional)** – marginLeft may be used to specify the margin between the left edge of the page and the start of printable area. By default, this margin is set to .25in. A page margin may be specified using points (pt), inches (in), millimeters (mm), or centimeters(cm).
- **marginRight (optional)** – marginRight may be used to specify the margin between the right edge of the page and the start of printable area. By default, this margin is set .25in. A page margin may be specified using points (pt), inches (in), millimeters (mm), or centimeters(cm).
- **marginTop (optional)** – marginTop may be used to specify the margin between the top edge of the page and the start of printable area. By default, this margin is set to .25in. A page margin may be specified using points (pt), inches (in), millimeters (mm), or centimeters(cm).

- **mediaHeight (optional)** – mediaHeight may be used to specify the physical height of the paper this page represents. By default, mediaHeight is set to 11in. mediaHeight may be specified using points (pt), inches (in), millimeters (mm), or centimeters(cm).
- **mediaWidth (optional)** – mediaWidth may be used to specify the physical width of the paper this page represents. By default, mediaWidth is set to 8.5in. mediaWidth may be specified using points (pt), inches (in), millimeters (mm), or centimeters(cm).
- **orientation (optional)** – orientation is used to specify the orientation of the page, either "portrait" or "landscape". If the orientation is not specified, it defaults to portrait. Portrait orientation says to display the long side vertically and the short side horizontally while landscape says to display the long side horizontally and the short side vertically.
- **mediaType (optional)** – mediaType may be used to specify the type of media this page is designed on. By default, mediaType is set to letter which represents an 8.5in x 11in sheet of paper. mediaType supports the following types:

North American Standards:

- comm-10 4.125inx9.5in
- executive 7.25inx10.5in
- ledger 11inx17in
- legal 8.5inx14in
- letter 8.5inx11in
- monarch-envelope 3.875inx7.5in
- architecture-a 9.0inx12.0in
- architecture-b 12.0inx18.0in
- architecture-c 18.0inx24.0in
- architecture-d 24.0inx36.0in
- architecture-e 36.0inx48.0in
- roc-8k 10.75inx15.5in

- roc-16k 7.75inx10.75in

ISO Standards:

- a0 841mmx1189mm
- a1 594mmx841mm
- a2 420mmx594mm
- a3 297mmx420mm
- a4 210mmx297mm
- a5 148mmx210mm
- a6 105mmx148mm
- a7 74mmx105mm
- a8 52mmx74mm
- a9 37mmx52mm
- a10 26mmx37mm
- b0 1000mmx1414mm
- b1 707mmx1000mm
- b2 500mmx707mm
- b3 353mmx500mm
- b4 250mmx353mm
- b5 176mmx250mm
- b6 125mmx176mm
- b7 88mmx125mm
- b8 62mmx88mm
- b9 44mmx62mm

- b10 31mmx44mm
- c0 917mmx1297mm
- c1 648mmx917mm
- c2 458mmx648mm
- c3 324mmx458mm
- c4 229mmx324mm
- c5 162mmx229mm
- c6 114mmx162mm
- c7 81mmx114mm
- c8 57mmx81mm
- c9 40mmx57mm
- c10 28mmx40mm
- designated-long 110mmx220mm

Japanese Standards:

- jis-b0 1030mmx1456mm
- jis-b1 728mmx1030mm
- jis-b2 515mmx728mm
- jis-b3 364mmx515mm
- jis-b4 257mmx364mm
- jis-b5 182mmx257mm
- jis-b6 128mmx182mm
- jis-b7 91mmx128mm
- jis-b8 64mmx91mm
- jis-b9 45mmx64mm

- jis-b10 32mmx45mm
- chou2 111.1mmx146mm
- chou3 120mmx235mm
- chou4 90mmx205mm
- exec 216mmx330mm
- hagaki 100mmx148mm
- kahu 240mmx322.1mm
- kahu2 240mmx332mm
- oufuku 148mmx200mm
- you4 105mmx235mm

Chinese Standards:

- dai-pa-kai 275mmx395mm
- juuro-ku-kai 198mmx275mm
- pa-kai 267mmx389mm
- prc-16k 146mmx215mm
- prc-32k 97mmx151mm
- prc-1 102mmx165mm
- prc-2 102mmx176mm
- prc-3 125mmx176mm
- prc-4 110mmx208mm
- prc-5 110mmx220mm
- prc-6 120mmx320mm
- prc-7 160mmx230mm
- prc-8 120mmx309mm

- prc-10 324mmx458mm

A Page element may contain one or more of the form components from the form components section.

5.4. *Form Components*

Form components may only be placed into a Page element. Like pages, all form components must have a unique ID. Form components may also have tooltips, invalid and missing messages (used when validation fails or a value is missing).

Below are the attributes common to all form components:

- constraints (optional) – Different components may specify the constraints attribute which is comprised of various sub-attributes. Review the sub-attributes below to see which sub-attributes apply to which controls and how they may be used:
 - currency (optional) – Used only by the NumberTextBox to specify the currency to use such as 'USD' US Dollars or 'EUR' Euros. The following currencies are supported:

ARS - Argentine Peso / \$ 1.234,56

AUD - Australian Dollar / \$ 1,234.56

BRL - Brazilian Real / R\$ 1.234,56

CAD - Canadian Dollar / \$ 1,234.56

CHF - Swiss Franc / fr. 1.234,56

CLP - Chilean Peso / \$ 1,234.56

CNY - Chinese Yuan Renminbi / ¥ 1,234.56

COP - Colombian Peso / \$ 1,234.56

CZK - Czech Koruna / 1.234,56 Kč

DKK - Danish Krone / kr. 1.234,56

EUR - Euro / €1.234,56

GBP - UK/GB Pound Sterling / £1,234.56
 HKD - Hong Kong Dollar / HK\$ 1,234.56
 HUF - Hungarian Forint / 1.234,56 Ft
 ILS - New Israeli Shekel / ₪ 1.234,56
 INR - Indian Rupee / ₹ 1,234.56
 JPY - Japanese Yen / ¥ 1,234.56
 KRW - Republic of Korea Won / ₩ 1,234.56
 MAD - Moroccan Dirham / 1,234.56 د.م.
 MXN - Mexican Peso / \$ 1,234.56
 MYR - Malaysian Ringgit / RM 1,234.56
 NOK - Norwegian Krone / kr 1,234.56
 NZD - New Zealand Dollar / \$ 1,234.56
 PHP - Philippine Peso / ₱ 1,234.56
 PLN - Polish Zloty / 1.234,56 zł
 RUB - Russian Ruble / 1.234,56 p.
 SAR - Saudi Riyal / 1,234.56 ﷲ
 SEK - Swedish Krona / 1.234,56 kr
 SGD - Singapore Dollar / \$1,234.56
 THB - Thailand Baht / 1,234.56 ฿
 TRY - Turkish Lira / 1,234.56 ₺
 TWD - New Taiwan Dollar / 元 1,234.56
 USD - US Dollar / \$1,234.56
 VND - Vietnamese Dong / 1.234,56 đ
 ZAR - South African Rand / R 1,234.56

- `maxLen` (optional) – Used to specify the max number of characters that the component may contain. Used only by `TextBox`, and `TextArea`.
- `maxValue` (optional) – Used by `NumberTextBox` to specify a max range value, `DateTextBox` to specify a max date, and `TimeTextBox` to specify a max time.
- `minLen` (optional) – Used to specify the min number of characters that the component must contain. Used only by `TextBox`, and `TextArea`.
- `minValue` (optional) – Used only by `NumberTextBox` to specify a min range value, `DateTextBox` to specify a min date, and `TimeTextBox` to specify a min time.
- `negativeUseParens` (optional) – Used only by `NumberTextBox` to specify that when a number is negative it should be enclosed in parentheses. When enabled, parentheses are used in lieu of the negative sign. Use "true" or "false".
- `negativeUseRed` (optional) – Used only by `NumberTextBox` to specify that when a number is negative, it should be displayed with red text. When enabled, red text used in lieu of the negative sign. Use "true" or "false".
- `places` (optional) – Used only by `NumberTextBox` to specify the number of decimal places the component allows. This number must be between 0-10. For example, you can specify 3 to say you'll allow a number followed by three decimal places.
- `regExp` (optional) – Used to specify a regular expression that should be used to validate the contents of a `TextBox` component. Regular expressions must only be specified using the JavaScript standard for regular expressions and are currently only available on the `TextBox` component.
- `required` (optional) – Used to specify whether or not the component is required. "true" or "false". This may be specified on `CheckBox`, `DateTextBox`, `DigitalSignatureBox`, `NumberTextBox`, `TextArea`, `TextBox`, and `TimeTextBox`.
- `separators` (optional) – Used only by `NumberTextBox` to specify the separator style to use. This is only for visual purposes, the attributes: `value`, `minValue` and `maxValue` must follow style a. To specify a separator style, use one of the following:

a - 1234.56

b - 1,234.56

c - 1234,56

d - 1.234,56

e - 1 234,56

- step (optional) – Used only by NumberTextBox to specify the decimal step to allow, 1 for a whole number and .X for a fractional number. For example, a step of .01 would allow and increment/decrement by a one hundredth.
- wrap (optional) – Used only by TextArea to specify how the text should be wrapped when submitted. Either hard or soft.
- id (required) – Every component is required to have a unique ID. An ID must start with a letter (a-z or A-Z) and can be followed by letters, numbers (0-9), hyphens (-), underscores (_), colons (:), or periods (.).
- invalidMessage (optional) – A component will display an invalid message when the component doesn't pass validation if validation is being used. invalidMessage can be specified to override the default message "The value entered is not valid". Validation can be enforced by using the minValue and maxValue attributes in the constraints attribute (where applicable), or the regExp attribute in the constraints attribute (where applicable). See the constraints section for more information on the available constraints.
- missingMessage (optional) – A component will display a missing message when it is marked required and doesn't have a value. missingMessage can be specified to override the default message "This value is required". The "missing" condition varies based on the type of control.
- security (optional) – A component may optionally specify the security required to enter/edit its value. These security levels can be assigned to users and groups via the associate access servlet on a form inside of TechDoc. The security levels are as follows:
 - none - The default setting that says that anyone who has access to complete a form submission can alter the field.
 - owner - Only the owner of the form or a user with owner access in TechDoc may change or alter the field.

- readonly - readonly says that the value may never be changed, not even by the owner. In this case the value is hard coded in the form at design time and can never be changed when creating a submission of the form.
- officeUse1-officeUse10 - The office-use-only security settings allow up to ten separate access levels on a field. In order to use these, you must perform an associate access on the form in TechDoc and give access to the users and/or groups that should be allowed to alter the fields using these security settings. In lieu of officeUse1-officeUse10, an alias can be used if defined in the settings using security aliases.
- tooltip (optional) – A component may specify a tooltip to display when the component is hovered over. Tooltips work the same as traditional tooltips with the only exception being, if the component is currently in the missing or invalid state, the missing message or invalid message will be shown instead.
- value (optional) – A component may set its default value using the value attribute. For radio buttons, value should always be present as it identifies the display value for each radio button.

5.4.1. CheckBox

Displays an HTML checkbox component. The checked attribute can be used to set the initial state of the checkbox. If omitted, the checkbox will default to being unchecked.

```

1 <Form version="1">
2   <Styles>
3     ...
4   </Styles>
5   <Settings>
6     ...
7   </Settings>
8   <Page id="page1">
9     <CheckBox id="checkbox1" checked="true" />
10  </Page>
11 </Form>
```

A CheckBox element supports the following attributes:

- [Common Form Component Attributes](#)

- checked (optional) – Whether or not the checkbox is checked by default. A value of true means the checkbox should appear checked by default, false means the checkbox should appear unchecked by default.

5.4.2. ComboBox/Option

Displays an HTML select component. The options a ComboBox lists should be enclosed in child Option elements. To preselect an option, the selected attribute can be specified with the ID of the option.

```

1  <Form version="1">
2    <Styles>
3      ...
4    </Styles>
5    <Settings>
6      ...
7    </Settings>
8    <Page id="page1">
9      <ComboBox id="combobox1" selected="option2">
10        <Option id="option1">Value 1</Option>
11        <Option id="option2">Value 2</Option>
12        <Option id="option3">Value 3</Option>
13      </ComboBox>
14    </Page>
15 </Form>

```

A ComboBox/Option element supports the following attributes:

- [Common Form Component Attributes](#)

5.4.3. DateTextBox

Displays an HTML input text component that also has date validation and is accompanied by a date picker widget. To preset a date, the value attribute can be specified. The value attribute (and minValue/maxValue if used) must be in the date format dd/MM/yyyy.

```

1  <Form version="1">
2    <Styles>
3      ...
4    </Styles>

```

```

5  <Settings>
6    ...
7  </Settings>
8  <Page id="page1">
9    <DateTextBox id="datetextbox1" value="01/01/2001" />
10 </Page>
11 </Form>

```

A DateTextBox element supports the following attributes:

- [Common Form Component Attributes](#)

5.4.4. DigitalSignatureBox

Displays a combination of an HTML input text component (for typed signatures), an HTML 5 canvas (for drawn signatures), and/or a file upload component for selecting and uploading an image of a signature.

```

1  <Form version="1">
2    <Styles>
3      ...
4    </Styles>
5    <Settings>
6      ...
7    </Settings>
8    <Page id="page1">
9      <DigitalSignatureBox id="digSig1" signatureType="typed" />
10     <DigitalSignatureBox id="digSig2" signatureType="drawn" />
11     <DigitalSignatureBox id="digSig3" signatureType="image" />
12   </Page>
13 </Form>

```

A DigitalSignatureBox element supports the following attributes:

- [Common Form Component Attributes](#)
- signatureType (required) – This attribute must specify the type of signature to capture:
 - drawn – A signature drawn using the mouse or finger on touch devices.

- image – A signature captured by the user uploading an image of their real signature.
- drawnOrImage –A signature drawn using the mouse or finger on touch devices or captured by the user uploading an image of their real signature.
- typed - A signature captured by the user typing their full name into a text box.

5.4.5. NumberTextBox

Displays an HTML input text component that also has number validation. This field can also specify a currency symbol to be used to capture currencies as well. To preset a value, the value attribute can be specified with the number/amount; when specifying a preset currency value, the currency symbol shouldn't be included.

```

1  <Form version="1">
2    <Styles>
3      ...
4    </Styles>
5    <Settings>
6      ...
7    </Settings>
8    <Page id="page1">
9      <NumberTextBox id="numberTextBox1" />
10     <NumberTextBox id="currencyTextBox1" constraints="currency: 'USD'"
      value="12345.00" />
11   </Page>
12 </Form>

```

A NumberTextBox element supports the following attributes:

- [Common Form Component Attributes](#)

5.4.6. RadioButtonGroup/RadioButton

Displays an HTML input radio button component. RadioButtons must be contained inside of a RadioButtonGroup element. To preselect a radio button, the checked attribute can be specified with the ID of the radiobutton to preselect.

```

1 <Form version="1">
2   <Styles>
3     ...
4   </Styles>
5   <Settings>
6     ...
7   </Settings>
8   <Page id="page1">
9     <RadioButtonGroup id="radioButtonGroup1" checked="radioButton2">
10      <RadioButton id="radioButton1" value="Coffee" />
11      <RadioButton id="radioButton2" value="Tea" />
12    </RadioButtonGroup>
13  </Page>
14 </Form>

```

A RadioButtonGroup/RadioButton element supports the following attributes:

- [Common Form Component Attributes](#)
- checked (optional) – This attribute can be used to set the default selected radio button of the group. The value of checked should be the ID of the corresponding radio button to check by default.

5.4.7. StaticText

Displays a standard HTML paragraph element. StaticText components are used to display blocks of text such as titles or instructions.

```

1 <Form version="1">
2   <Styles>
3     ...
4   </Styles>
5   <Settings>
6     ...
7   </Settings>
8   <Page id="page1">
9     <StaticText id="staticText1" >Hello World!</StaticText>
10  </Page>
11 </Form>

```

A StaticText element supports the following attributes:

- [Common Form Component Attributes](#)
- for (optional) – When StaticText components are used as a label for another component they should this attribute to state the ID of the component they are a label for.

5.4.8. StaticImage

Displays an image using an HTML image element. The imageHeight and imageWidth attributes must specify the height and width of the image in pixels with the px suffix and the mimeType attribute should specify the image's mime type. The image data itself should be a base 64 encoded and included at the text content of the StaticImage element. Below is an example of a 64 by 64-pixel PNG image where are pixels are full red.

```

1 <Form version="1">
2   <Styles>
3     ...
4   </Styles>
5   <Settings>
6     ...
7   </Settings>
8   <Page id="page1">
9     <StaticImage id="staticImage1" imageHeight="64px" imageWidth="64px"
      mimeType="image/png">iVBORw0KGgoAAAANSUhEUgAAAEAAAABABAMAAABYR2z
      tAAAAAXNSR0IArs4c6QAAAAARnQU1BAACxjwv8YQUAAAwUEXURf8AAAAAAAAA
      AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAN9XvKMAA
      AAJcEhZcwAADsMAAA7DAcdvqGQAAAAZSURBVEJH7cExAQAAAMKg9U9tBn8gAIBLD
      QhAAAH1leSjAAAAAEIFTkSuQmCC</StaticImage>
10  </Page>
11 </Form>

```

A StaticImage element supports the following attributes:

- [Common Form Component Attributes](#)
- imageHeight (required) – Specifies the image's height in pixels.
- mimeType (required) – Specifies the image's mime type.
- imageWidth (required) – Specifies the image's width in pixels.

5.4.9. TextArea

Displays a standard HTML textarea. The rows and cols attributes can be specified to specify the number of rows and columns in the text area. Preset text can be specified by placing text in the text content of the TextArea element.

```

1  <Form version="1">
2    <Styles>
3      ...
4    </Styles>
5    <Settings>
6      ...
7    </Settings>
8    <Page id="page1">
9      <TextArea id="textArea1" rows="4" cols="20">Default text here!</TextArea>
10   </Page>
11 </Form>

```

A TextArea element supports the following attributes:

- [Common Form Component Attributes](#)
- cols (optional) – The number of columns in the TextArea based on the average width of its font.
- rows (optional) – The number of rows in the TextArea based on the average height of its font.

5.4.10. TextBox

Displays a standard HTML input text component. A preset value can be specified using the value attribute.

```

1  <Form version="1">
2    <Styles>
3      ...
4    </Styles>
5    <Settings>
6      ...
7    </Settings>
8    <Page id="page1">
9      <TextBox id="textBox1" value="example" />

```



```

10 </Page>
11 </Form>

```

A TextBox element supports the following attributes:

- [Common Form Component Attributes](#)

5.4.11. TimeTextBox

Displays an HTML input text component that also has time validation and is accompanied by a time picker widget. A preset time can be specified using the value attribute. The value attribute (and minValue/maxValue if used) must be in 24-hour time.

```

1 <Form version="1">
2   <Styles>
3     ...
4   </Styles>
5   <Settings>
6     ...
7   </Settings>
8   <Page id="page1">
9     <TimeTextBox id="timeTextBox1" value="17:59:59" />
10  </Page>
11 </Form>

```

A TimeTextBox element supports the following attributes:

- [Common Form Component Attributes](#)

6. The TechDoc Server-Side Forms Reference

TechDoc features a complete electronic forms engine. The engine is tightly integrated with TechDoc and provides the abilities to: create, and publish electronic forms, acquire submissions from both users of TechDoc and external users without TechDoc accounts, and the ability to save out (to various formats including PDF) and print form submissions individually or as a complete set. TechDoc's electronic forms engine, forms designer, and implementation are based entirely around HTML 5 and do not require a particular web browser or web browser extension. Forms should be submittable by any user using any of the standard web browsers on a desktop computer, laptop, tablet, or mobile device. The following sections detail the various parts of the forms area of TechDoc, how to navigate each, and the function of each.

6.1. Forms Management Menu

The Forms Management menu contains all of the functions for creating and managing electronic forms, and form submissions as well as a link to the Forms Editor. Currently, this menu is only available to Users with the Forms or Forms Manager privilege or Administrators.

6.1.1. Forms Management

Navigation: [\[DocMgr > Forms\]](#)

A User with the Forms Manager or Admin privilege will have access to all of the Forms Management functions. A User with the Forms privilege will only have access to their Forms.

6.1.2. Forms Editor

The DocuBrain Forms Editor is a WYSIWYG (What You See Is What You Get) web browser-based application that allows DocuBrain forms to be created or modified. Internally, forms use the DocuBrain Electronic Form XML format.

The editor allows the user to manipulate these types of forms without the need to know this internal XML format. For more information about DocuBrain Electronic Forms or using the Forms Editor, please refer to the DocuBrain Forms Editor Guide available in the [documentation section](#) of the DocuBrain web site.

- The User must have the Forms or Forms Manager privilege.

Navigation: [\[DocMgr > Forms > Side Menu > Forms Editor\]](#)

Once you exit the DocuBrain Forms Editor, you will be returned to the previous location in TechDoc that sent you to the editor in the first place or to the Forms Dashboard where you can continue working within the Doc Manager.

6.1.3. Publishing a Form

Publish Form is used to publish a generation as an electronic form in the document manager. The file of the generation must be an XML file built using the DocuBrain TechDoc Electronic Form XML Format. After publishing the generation as a form, the form can be viewed under the Forms area in the document manager where it can be further configured if necessary.

- The User must have the Forms or Forms Manager privilege.

Navigation: *[DocMgr > Explorer > Select Desired Generation > Side Menu > Publish As Form]*

Step 1:

1. Enter a name for the Form in the Name box. The Form name must be unique within the same Document Manager. Name is a required field. The maximum length of this field is 255 characters.
2. Enter the description of the Form in the Description box. The maximum length of this field is 255 characters.
3. In the Available box, click the down arrow, and select Yes (form will be available and accept form submissions) or select No (form will not be yet available to accept form submissions). This setting allows you to toggle whether or not the form will accept submissions. This setting must be Yes to begin accepting submissions even if an Available Start Date is specified.
4. If needed, enter the date the Form should become available in the Available Start Date box. Use the first box for the date and the second box for the time. If Available Start Date and/or Available End Date are left empty, the Form will be available to submit as long as Available is 'Yes' and the Max Submissions have not been reached.
5. If needed, enter the date the Form should no longer be available in the Available End Date box. Use the first box for the date and the second box for the time. If Available Start Date and/or Available End Date are left empty, the Form will be available to submit as long as Available is 'Yes' and the Max Submissions have not been reached.
6. In the Days To Complete box, enter the number of days a user has to complete a submission once they've started one. By default, this field is 0 meaning the user must complete the form once they've started and saving/resuming a form submission is not allowed. Otherwise, this number indicates that the user may take X number of days to complete the submission. Additionally, this also

enables the user to save progress and resume a submission at a later time if they are unable to complete it in a single sitting.

7. In the Max Submissions box, enter the maximum number of submissions this form should allow or 0 if the form should allow an infinite number of submissions.
8. Enter the reason for publishing the Form in the Reason box. Reason is a required field. The maximum length of this field is 255 characters.
9. Click the Cancel button to cancel the command, or click the OK button to publish the Form.

Notes:

- A new Form record will be created.
- A history record will be generated for publishing of the Form.

6.1.4. Modifying a Form

Modify Form modifies an existing Form in the Document Manager.

- The User must have Owner access to the Form.
- The Form name cannot be the same as any other Form in the system.

Navigation: *[DocMgr > Forms > Select Desired Forms > Side Menu > Modify]*

Step 1:

1. If applicable, modify the Form name in the Name box. Form name must be unique within the same Document Manager. Name is a required field. The maximum length of this field is 255 characters.
2. If applicable, modify the Form description in the Description box. The maximum length of this field is 255 characters.
3. If applicable, modify the Availability setting by clicking the down arrow, and selecting Yes (form will be available and accept form submissions) or No (form will not be yet available to accept form submissions). This setting allows you to toggle whether or not the form will accept submissions. This setting must be Yes to begin accepting submissions even if an Available Start Date is specified.
4. If applicable, modify the date the Form should become available in the Available Start Date box. Use the first box for the date and the second box for the time. If Available Start Date and/or Available End Date are left empty, the Form will be available to submit as long as Available is 'Yes' and the Max Submissions have not been reached.
5. If applicable, modify the date the Form should no longer be available in the Available End Date box. Use the first box for the date and the second box for the time. If Available Start Date and/or Available End Date are left empty, the Form

will be available to submit as long as Available is 'Yes' and the Max Submissions have not been reached.

6. If applicable, In the Days To Complete box, modify the number of days a user has to complete a submission once they've started one. By default, this field is 0 meaning the user must complete the form once they've started and saving/resuming a form submission is not allowed. Otherwise, this number indicates the that the user may take X number of days to complete the submission. Additionally, this also enables the user to save progress and resume a submission at a later time if they are unable to complete it in a single sitting.
7. If applicable, In the Max Submissions box, modify the maximum number of submissions this form should allow or 0 if the form should allow an infinite number of submissions.
8. Enter the reason for modifying the Form in the Reason box. Reason is a required field. The maximum length of this field is 255 characters.
9. Click the Cancel button to cancel the command, or click the OK button to modify the Form.

Notes:

- The Form record will be modified.
- A history record will be generated for modifying the Form.

6.1.5. Deleting a Form

Delete Form deletes an existing Form in the Document Manager.

- The User must have the Forms privilege.
- The User must have Delete access to the Form. Delete access is determined by the User being a Document Administrator or the actual owner of the Form.

Navigation: *[DocMgr > Forms > Select Desired Form > Side Menu > Delete]*

Step 1:

The Form to be deleted and the Form attributes are displayed.

1. Click the Cancel button to cancel the command, or click the Next button to delete the Form.

Step 2:

The Form to be deleted and the Form attributes are displayed.

1. Enter a reason for deleting the Form in the Reason box. This is a required field. The maximum length of this field is 255 characters.
2. Click the Cancel button to cancel the command, click the Previous button to return to the previous screen, or click the OK button to delete the Form.

Notes:

- The Form record will be deleted.
- All Form Submissions of the Form will be deleted.
- A history record will be generated for deletion of the Form.

6.1.6. Showing a Form



Show Form displays the details for multiple Forms or a specific Form.

All Forms

Navigation: *[DocMgr > Forms > Side Menu > All Forms]*

All Forms displays all the Forms that has been created in the Document Manager.

Only the Document Administrator can show All Forms.

- The Name, Availability, and Description are displayed for each Form.
- The number of Forms is shown.
- The Forms are displayed in alphabetical order by name.
- Click on  to View a specific Form.
- Click on  to Show Info for the specific Form.

My Forms



Navigation: *[DocMgr > Forms > Side Menu > My Forms]*

My Forms displays all the Forms that you are currently the owner of in the Document Manager. If you have Owner access to a Form but are not the actual owner of the Form, it will not be displayed here.

- The User must have the Forms privilege.

If you have not created a Form in the Document Manager, the following message will be displayed: "You don't have any Forms."

- The Name, Availability, and Description are displayed for each Form.
- The number of Forms is shown.

- The Forms are displayed in alphabetical order by name.
- Click on  to View a specific Form.
- Click on  to Show Info for the specific Form.



Owner Forms

Navigation: *[DocMgr > Forms > Side Menu > Owner Forms]*

Owner Forms displays all the Forms in the Document Manager that you have Owner access to. If you are the actual owner of a Form, it will not be displayed here.

- The User must have the Forms privilege.

If you have not been associated for Owner access to any Form in the Document Manager, the following message will be displayed: "You don't have Owner access to any Forms."

- The Name, Availability, and Description are displayed for each Form.
- The number of Forms is shown.
- The Forms are displayed in alphabetical order by name.
- Click on  to View a specific Form.
- Click on  to Show Info for the specific Form.



Submitter Forms

Navigation: *[DocMgr > Forms > Side Menu > Submitter Forms]*

Submitter Forms displays all the Forms in the Document Manager that you have Form Submitter access to.

- The User must have the Forms privilege.

If you have not been associated for Form Submitter access to any Form in the Document Manager, the following message will be displayed: "You don't have form submitter access to any Forms."

- The Name, Availability, and Description are displayed for each Form.
- The number of Forms is shown.
- The Forms are displayed in alphabetical order by name.
- Click on  to View a specific Form.
- Click on  to Show Info for the specific Form.

A Specific Form

Navigation: [\[DocMgr > Forms > Select Desired Form\]](#)

Form Info displays the full details for a specific Form.

The User must have Read access to the Form. Read access is determined by the User being a Document Administrator, or having Read access to the parent Document of the Form.

Field Name	Description
Name	The name of this Form.
Description	The description of this Form.
External UUID	The external UUID of this Form. This field is primarily used to interface with external systems.
Doc Number	The document number of the parent document for this form.
Gen Number	The generation number of the parent generation for this form.
Published	The date this form was published.
Modified	The date and time the Form was last modified.
Available	Yes - The Form is available and can receive submissions as long other conditions don't apply. For example, if the Available Start Date has not yet been reached but the Form is available, a user will still not be able to create a submission until the Start Date is reached. No - The Form is not available and cannot receive any/anymore submissions regardless of other availability conditions.
Available Start Date	When specified, a Form will not accept submissions until the start date has been reached. When left blank, this form may start accepting submissions at any time as long as no other availability conditions prevent it from doing so.
Available End Date	When specified, a Form will not accept submissions after the end date has been reached. When left blank, this form may continue accepting submissions indefinitely as long as no other availability conditions prevent it from doing so such as a maximum number of submissions.

Days To Complete	The number of days the user has to complete a submission once they've started one. When left blank, there is no limit and the user may take as long as needed.
Max Submissions	The maximum number of submissions this Form will accept. When left blank, this Form will accept an unlimited number of submissions as long as it is available.

6.1.7. Viewing a Form

Navigation: *[DocMgr > Forms > Select Desired Form > Side Menu > Preview]*

View Form displays the Form in its rendered version as a user will see it when they are filling out a submission. This function can be useful when designing a form or to check up on a form from time to time to make sure it's functioning properly. Note: When viewing a Form in preview mode, you will not be able to make a submission and other functionalities may be limited. If you wish to test a form in its entirety you should simply complete a submission to see the form works as designed and then delete that submission.

Once Preview has been clicked on the Side Menu, a new tab should appear in the browser displaying the Form in its rendered version. You should be able to page through the Form (if it has multiple pages), verify the form looks as it should, and test various field constraints such as max length, formatting, etc. When you are finished looking at a Preview of a Form, you may simply close the tab.

6.1.8. Submitting a Form

Once a form submission has been created and the user has completed filling it out, the user will then click the Submit button. During the submit process, all of the data is packaged up and sent to the Submit Form servlet. Once the Submit Form servlet receives the data, the form submission is finalized, marked as completed and is stored in the database.

6.1.9. Exporting a Form

Export Form gives form owners the ability to export one or more existing form submissions to a variety of file formats for use elsewhere.

Note:

- The user must have the Forms Manager, Admin privilege, or be the owner of a form to export it.

Navigation: *[DocMgr > Forms > Select Desired Form > Side Menu > Export]*

Step 1:

To search for form submissions to export, enter data in one or more search fields, adjust the search options if necessary, and press the Search button to submit the request. Pressing the Clear Input button will reset all of the search criteria.

Here are a few rules to remember when specifying search criteria:

- The search criteria are not case sensitive
- The asterisk (*) represents zero or more characters
- The question mark (?) represents exactly one character
- Multiple asterisks and/or question marks are allowed in the same word
- Consecutive asterisks are not allowed but consecutive question marks are allowed
- Search fields cannot contain only asterisks
- Asterisks and question marks may be positioned anywhere in a word as long as they don't break any of the previous rules

The table below describes the fields and search criteria:

Field	Search Criteria
External UUID	Enter the external UUID or part of the external UUID with wildcards.
Create Date	<p>Search for form submissions created on a specific date. For example: submissions created on 02/03/2001. Enter 02/03/2001 to 02/03/2001 in the date boxes. Use: mm/dd/yyyy.</p> <p>Search for submissions created for a range of dates. For example: submissions created from 01/18/2001 to 01/23/2001. Enter 01/18/2001 to 01/23/2001 in the date boxes. Use: mm/dd/yyyy.</p> <p>Search for submissions created since a specific date. For example: submissions created from 01/19/2001 to present date. Enter 01/19/2001 in first date box. Leave second date box blank. Use: mm/dd/yyyy.</p> <p>Search for submissions created prior to a specific date. For example: submissions created prior to 01/23/2001. Enter 01/23/2001 in second date box. Leave first date box blank. Use: mm/dd/yyyy.</p>
Last Modify Date	Search for form submissions last modified on a specific date. For example: submissions last modified on 02/03/2001. Enter 02/03/2001 to 02/03/2001 in the date boxes. Use: mm/dd/yyyy.



	<p>Search for submissions last modified for a range of dates. For example: submissions last modified from 01/18/2001 to 01/23/2001. Enter 01/18/2001 to 01/23/2001 in the date boxes. Use: mm/dd/yyyy.</p> <p>Search for submissions last modified since a specific date. For example: submissions last modified from 01/19/2001 to present date. Enter 01/19/2001 in first date box. Leave second date box blank. Use: mm/dd/yyyy.</p> <p>Search for submissions last modified prior to a specific date. For example: submissions last modified prior to 01/23/2001. Enter 01/23/2001 in second date box. Leave first date box blank. Use: mm/dd/yyyy.</p>
Completion Date	<p>Search for form submissions completed on a specific date. For example: submissions completed on 02/03/2001. Enter 02/03/2001 to 02/03/2001 in the date boxes. Use: mm/dd/yyyy.</p> <p>Search for submissions completed for a range of dates. For example: submissions completed from 01/18/2001 to 01/23/2001. Enter 01/18/2001 to 01/23/2001 in the date boxes. Use: mm/dd/yyyy.</p> <p>Search for submissions completed since a specific date. For example: submissions completed from 01/19/2001 to present date. Enter 01/19/2001 in first date box. Leave second date box blank. Use: mm/dd/yyyy.</p> <p>Search for submissions completed prior to a specific date. For example: submissions completed prior to 01/23/2001. Enter 01/23/2001 in second date box. Leave first date box blank. Use: mm/dd/yyyy.</p>
Submitter	Click on the down arrow and select the user from the list.
IP Address	Enter the IP address or part of the IP address with wildcards.
Status	Click on the down arrow and select the status from the list.
New Field	To search on a particular field in the form, you may select and add that field to the search. Once added, you may enter the exact data you are looking for or just part of the data and use wildcards. To add a form field, click on the down arrow and select the field from the list and then click the Add button.

1. To search for submissions to export, enter data in one or more search fields. Refer to the table above for help on searching.

2. Click the Cancel button to cancel the command, click the Search button to submit the request, or click the Clear Input button to reset all of the search criteria.



Step 2:

A listing of all of the form submissions that matched the search criteria is displayed.

- If no form submissions were found that matched the search criteria, the following message will be displayed: "No form submissions found matching the specified search criteria".
 - The Form Name and External UUID are displayed for each form submissions.
 - Click on either  or  to Show Form Submission Info.
1. Click the Cancel button to cancel the command, click the Clear Input button to reset all of the search criteria, if you need to refine your search, enter data in another field and click the Search button, or click the Next button to continue.

Step 3:

A listing of all of the form submissions that matched the search criteria is displayed. You must choose the form submissions to be exported by placing a check in the checkbox in front of the form submission. You can check each form submission individually or you can click the Select All button to select all the form submissions.

- The Form Name and External UUID are displayed for each form submission.
- Click on  or the  to Show Form Submission Info.
- By default, an empty checkbox is displayed in front of each form submission.

Note:

Only form submissions with a check in the checkbox in front of the form submission will be exported.

1. Choose a form submission to be exported by individually placing a check in the checkbox in front of the form submission or click the Select All button to select all of the form submissions.
2. Click the Cancel button to cancel the command, click the Previous button to return to the previous screen, click the Unselect All button to uncheck all previously checked form submissions, click the Next button to continue.

Step 4:

1. Click the Cancel button to cancel the command, click the Previous button to return to the previous screen, or click the Next button to continue.

Step 5:

1. Select the format the form submissions should be exported to. Currently, the following formats are supported:
 - CSV - Produces a comma separated values file that may be imported into other applications.
 - HTML - Produces a single, all inclusive, portable, static HTML file of the form submissions. This static HTML file when viewed in most modern browsers will look very similar to the original live form without any of the live feedback, scripted actions, etc. Note: as HTML rendering varies from browser to browser, the HTML file may appear slightly different.
 - PDF - Produces a single PDF file that may be opened in any application that supports PDF. The PDF file should be a near mirror match of the original form to the extent supported by the PDF format. As the original form is HTML based and PDF is not, there may be small differences due to the lack of capabilities in PDF.
 - XML - Produces a single XML file that can be used by other applications or developers to import the form submission data for their own use.
2. Enter a reason for exporting the form submissions in the Reason box. This is a required field. The maximum length of this field is 255 characters.
3. Click the Cancel button to cancel the command, click the Previous button to return to the previous screen, or click the OK button to export the form submissions.
4. Once the OK button has been clicked, TechDoc will begin building your export form request and the download should begin shortly. Normally, the download will start within a second or two but with larger exports (hundreds or thousands) the download may take a bit longer to begin. There is no limit on the number of submissions that can be exported to a single file but keep in mind, not all applications handle larger files. If you experience issues with a PDF, CSV, etc. being very slow/impossible to open/load, it may be more appropriate to export the submissions into chunks so that you have multiple smaller files instead of one large unusable file.

6.1.10. Creating a Form Submission

Create Form Submission is used to create and begin a new form submission. After a form submission has been created, it's up to the user to complete this submission before the form submission time limit expires, or the form's max number of submissions is reached; both of these are dictated by the settings on the Form itself. Once a submission has been created, the user should try, if possible, to complete the submission in a single sitting and get it submitted. However, if the form allows it, the user may elect to complete the form

at a later time by utilizing the save progress feature. If a form supports saving/resuming submissions, a save button will appear next to the submit button. A user can click this button at any time to save the progress made on the submission. Then at a later time, the user may revisit the submission (using the same URL as originally used to create the submission) to resume the submission.

Navigation: *[DocMgr > Forms > Select Desired Form > Side Menu > Create Submission]*

6.1.11. Modifying a Form Submission

Modify Form Submission can be used to modify the status an existing Form Submission in the Document Manager. To modify the contents of a Form Submission, the Form Submission Status must first be set back to incomplete. Then because the Form Submission is "incomplete" it may be resumed and corrected where need be by the original User, Forms Manager, or Admin.

- The User must have Owner access to the Form Submission.

Navigation: *[DocMgr > Forms > Select Desired Form > Side Menu > All Submissions > Select Desired Form Submission > Side Menu > Modify]*

Step 1:

1. If applicable, modify the Form Submission status.

Status	Definition
Incomplete	A Form Submission that has been started but is not yet complete.
Submitted	A Form Submission that has been submitted by the user but has not yet been processed.
Processing	A submitted Form Submission that is currently being processed.
Processed	A Form Submission that has completed the processing phase and should require no more attention.
Rejected	Used to flag a Form Submission that has been submitted but was rejected because it was not complete, contained invalid data, missed required data, etc.

2. Enter the reason for modifying the Form Submission in the Reason box. Reason is a required field. The maximum length of this field is 255 characters.
3. Click the Cancel button to cancel the command, or click the OK button to modify the Form Submission.

Notes:

- The Form Submission record will be modified.
- A history record will be generated for modifying the Form Submission.

6.1.12. Deleting a Form Submission

Delete Form Submission deletes an existing Form Submission in the Document Manager.

- The User must have the Forms privilege.
- The User must have Delete access to the Form Submission. Delete access is determined by the User being a Document Administrator or the actual owner of the Form Submission.

Navigation: *[DocMgr > Forms > Select Desired Form > Side Menu > All Submissions > Select Desired Form Submission > Side Menu > Delete]*

Step 1:

The Form Submission to be deleted and the Form Submission attributes are displayed.

1. Click the Cancel button to cancel the command, or click the Next button to delete the Form Submission.

Step 2:

The Form Submission to be deleted and the Form Submission attributes are displayed.

1. Enter a reason for deleting the Form Submission in the Reason box. This is a required field. The maximum length of this field is 255 characters.
2. Click the Cancel button to cancel the command, click the Previous button to return to the previous screen, or click the OK button to delete the Form Submission.

Notes:

- The Form Submission record will be deleted.
- A history record will be generated for deletion of the Form Submission.

6.1.13. Resuming a Form Submission

A Form Submission once created, may be saved and resumed if allowed by that Form. Saving and resuming a submission allows the user the ability to complete a form submission little by little over the course of several days, weeks, etc. if necessary. Upon the initial creation of the submission, the user may at any time click the Save button in the lower right-hand corner of the form submission window. At a later time, the user can navigate back to the Form Submission in the DM and click Resume or use the URL they originally accessed to create the submission to begin with; typically, this is the case for a Remote User. The user can save and resume as many times as needed as long as the Form is still available and the days to complete limit on the form hasn't expired.

Navigation: *[DocMgr > Forms > Select Desired Form > Side Menu > All Submissions > Select Desired Submission > Side Menu > Resume Submission]*

6.1.14. Showing a Form Submission

Show Form Submission displays the details for multiple Form Submissions or a specific Form Submission.

All Submissions

Navigation: *[DocMgr > Forms > Select Desired Form > Side Menu > All Submissions]*

All Submissions shows all of the Form Submissions of the selected Form.

- The User must have Read access to the Form.

On this screen, all of the Form Submissions of the selected Form are displayed. Some Forms may have a large number of submissions making it difficult to find a particular submission. If needed, the Date Range boxes can be used to say only show Form Submissions created on and after the starting date (left box) and/or before the ending date (right box).

- To display Form Submissions for a specific date. For example, display Form Submissions created on 02/03/2001. Under Show Form Submissions within the Date Range box enter 02/03/2001 to 02/03/2001. Use: mm/dd/yyyy.
- To display Form Submissions for a range of dates. For example, display Form Submissions created from 01/18/2001 to 01/23/2001. Under Show Form Submissions within the Date Range box enter 01/18/2001 to 01/23/2001. Use: mm/dd/yyyy.
- To display Form Submissions since a specific date. For example, display Form Submissions created from 01/19/2001 to present date. Under Show Form

Submissions within the Date Range box enter 01/19/2001 in first date field. Leave second date field blank. Use: mm/dd/yyyy.

- To display Form Submissions prior to a specific date. For example, display Form Submissions created prior to 01/23/2001. Under Show Form Submissions within the Date Range box enter 01/23/2001 in second date field. Leave first date field blank. Use: mm/dd/yyyy.
 - To display all Form Submissions under Show Form Submissions within the Date Range box leave both date fields blank.
1. Enter the desired date(s) in the Date Range box.
 2. Click the Clear Input button to clear the Date Range box or click the OK button to display the Form Submissions.

A Specific Form Submission

Navigation: *[DocMgr > Forms > Select Desired Form > Side Menu > All Submissions > Select Desired Form Submission]*

Form Submission Info displays the full details for a specific Form Submission.

- The User must have Read access to the Form Submission. Read access is determined by the User being a Document Administrator, or the actual owner of the Form Submission.

Field Name	Description
Form	The parent Form the submission belongs to.
External UUID	The external UUID of this Form Submission. This field is primarily used to interface with external systems.
Doc Number	The document number of the parent document of the form.
Gen Number	The generation number of the parent generation of the form.
Created	The date and time this Form Submission was created.
Modified	The date and time the Form Submission was last modified.
Completed	The date and time the Form Submission was completed.
Submitter	The User that created this Form Submission.
IP Address	The IP Address of the Submitter.

<p>Status</p>	<p>Form Submissions can be marked with various statuses to indicate their current state in the system. Currently, TechDoc only works with two Statuses directly. Once a Form Submission is started by a user, it is marked as incomplete. Once a user completes the Form Submission and clicks the Submit button, the system marks the submission as Submitted. From here, it is up to a Forms Manager, Admin, or Workflow Process to pick up the Form Submission and perform additional processing. The various Statuses are:</p> <ul style="list-style-type: none"> • Created - A Form Submission that has been created but not yet started. • Incomplete - A Form Submission that has been started but is not yet complete. • Submitted - A Form Submission that has been submitted by the user but has not yet been processed. • Processing - A submitted Form Submission that is currently being processed. • Processed - A Form Submission that has completed the processing phase and should require no more attention. • Rejected - Used to flag a Form Submission that has been submitted but was rejected because it was not complete, contained invalid data, missed required data, etc.
----------------------	---

Form Submission Entries

Form Submission Entries displays each of the form fields and their values as submitted by the user. The first column displays the internal ID of the form field (as defined by the Form) and the value that was input by the user.

6.1.15. Showing a Form Submission Signature

Show Form Submission Signature displays the details for a digital signature on a Form Submission.

A Specific Form Submission Signature

Navigation: *[DocMgr > Forms > Select Desired Form > Side Menu > All Submissions > Select Desired Form Submission > Select Desired Form Submission Entry Signature]*

Form Submission Signature Info displays the full details for a specific Form Submission Signature. Additionally, validation is performed when a Form Submission Signature is viewed to assure that both the signature itself and every single Form Submission Entry is

exactly as it was at the time the signature was created and submitted by the user. If a single change is made to a single entry or the signature is altered or tampered with in any way, the signature will not pass validation. Signatures that pass validation will display yes with a green checkmark on the Valid line; signatures that fail validation will display no with a red X. When a signature fails validation, the reason(s) will be displayed.

- The User must have Read access to the Form Submission. Read access is determined by the User being a Document Administrator, or the actual owner of the Form Submission.

Field Name	Description
Signature Image	The image of the actual signature of the user. This image is either drawn by mouse or finger, or may be an uploaded image of the user's real signature. Signature Image is present only when the signature is image based.
Typed Signature	The user's signature as they typed it in. Typed Signature is present only when the signature is text based.
Valid	Displays whether or not this signature passed validation. Signatures that pass will be displayed as yes with a green checkmark; those that fail validation will be displayed as no with a red X.
Submitter	The user that performed the form submission.
Authenticator Name	The name of the authenticator used to authenticate the user prior to their submitting of the form.
Authenticator Data	The unique identifier used to map the user from their authenticator to their TechDoc user account.
IP Address	The IP Address of the submitter.
Signed Date	The date and time the Form Submission Signature was created.
Submission External UUID	The external uuid of this Form Submission. This field is primarily used to interface with external systems.

6.1.16. Viewing a Form Submission

View Form Submission can be used to display a rendered Form Submission (the final product). From this screen, one can print the Form Submission if needed; the printed

result should look nearly identical to the rendered version on the screen. When viewing a Form Submission, the submission will be read only; the data will not be editable. If this Form Submission requires editing, it must first be modified back to the "Incomplete" status. From there one can resume the Form Submission and make any corrections needed. Once satisfied, the user can then click Submit to submit the updated version of the Form. If more than one Form needs to be printed or saved, the Export Form function should be used instead.

Navigation: *[DocMgr > Forms > Select Desired Form > Side Menu > All Submissions > Select Desired Submission > View]*

6.1.17. Searching for Form Submissions

Search Form Submissions gives form owners the ability to search all of the submissions made for a particular form.

Note:

- The user must have the Forms Manager, Admin privilege, or be the owner of a form to search submissions.

Navigation: *[DocMgr > Forms > Select Desired Form > Side Menu > Search Submissions]*

Step 1:

To search for form submissions, enter data in one or more search fields, adjust the search options if necessary, and press the Search button to submit the request. Pressing the Clear Input button will reset all of the search criteria.

Here are a few rules to remember when specifying search criteria:

- The search criteria are not case sensitive
- The asterisk (*) represents zero or more characters
- The question mark (?) represents exactly one character
- Multiple asterisks and/or question marks are allowed in the same word
- Consecutive asterisks are not allowed but consecutive question marks are allowed
- Search fields cannot contain only asterisks
- Asterisks and question marks may be positioned anywhere in a word as long as they don't break any of the previous rules

The table below describes the fields and search criteria:

Field	Search Criteria
-------	-----------------



External UUID	Enter the external UUID or part of the external UUID with wildcards.
Create Date	<p>Search for form submissions created on a specific date. For example: submissions created on 02/03/2001. Enter 02/03/2001 to 02/03/2001 in the date boxes. Use: mm/dd/yyyy.</p> <p>Search for submissions created for a range of dates. For example: submissions created from 01/18/2001 to 01/23/2001. Enter 01/18/2001 to 01/23/2001 in the date boxes. Use: mm/dd/yyyy.</p> <p>Search for submissions created since a specific date. For example: submissions created from 01/19/2001 to present date. Enter 01/19/2001 in first date box. Leave second date box blank. Use: mm/dd/yyyy.</p> <p>Search for submissions created prior to a specific date. For example: submissions created prior to 01/23/2001. Enter 01/23/2001 in second date box. Leave first date box blank. Use: mm/dd/yyyy.</p>
Last Modify Date	<p>Search for form submissions last modified on a specific date. For example: submissions last modified on 02/03/2001. Enter 02/03/2001 to 02/03/2001 in the date boxes. Use: mm/dd/yyyy.</p> <p>Search for submissions last modified for a range of dates. For example: submissions last modified from 01/18/2001 to 01/23/2001. Enter 01/18/2001 to 01/23/2001 in the date boxes. Use: mm/dd/yyyy.</p> <p>Search for submissions last modified since a specific date. For example: submissions last modified from 01/19/2001 to present date. Enter 01/19/2001 in first date box. Leave second date box blank. Use: mm/dd/yyyy.</p> <p>Search for submissions last modified prior to a specific date. For example: submissions last modified prior to 01/23/2001. Enter 01/23/2001 in second date box. Leave first date box blank. Use: mm/dd/yyyy.</p>
Completion Date	<p>Search for form submissions completed on a specific date. For example: submissions completed on 02/03/2001. Enter 02/03/2001 to 02/03/2001 in the date boxes. Use: mm/dd/yyyy.</p> <p>Search for submissions completed for a range of dates. For example: submissions completed from 01/18/2001 to</p>

	<p>01/23/2001. Enter 01/18/2001 to 01/23/2001 in the date boxes. Use: mm/dd/yyyy.</p> <p>Search for submissions completed since a specific date. For example: submissions completed from 01/19/2001 to present date. Enter 01/19/2001 in first date box. Leave second date box blank. Use: mm/dd/yyyy.</p> <p>Search for submissions completed prior to a specific date. For example: submissions completed prior to 01/23/2001. Enter 01/23/2001 in second date box. Leave first date box blank. Use: mm/dd/yyyy.</p>
Submitter	Click on the down arrow and select the user from the list.
IP Address	Enter the IP address or part of the IP address with wildcards.
Status	Click on the down arrow and select the status from the list.
New Field	To search on a particular field in the form, you may select and add that field to the search. Once added, you may enter the exact data you are looking for or just part of the data and use wildcards. To add a form field, click on the down arrow and select the field from the list and then click the Add button.

1. To search for submissions, enter data in one or more search fields. Refer to the table above for help on searching.
2. Click the Cancel button to cancel the command, click the Search button to submit the request, or click the Clear Input button to reset all of the search criteria.

Step 2:

A listing of all of the form submissions that matched the search criteria is displayed.

- If no form submissions were found that matched the search criteria, the following message will be displayed: "No form submissions found matching the specified search criteria".
- The Form Name and External UUID are displayed for each form submissions.
- Click on either  or  to Show Form Submission Info.
- Click the Search button to perform an updated search, click the Cancel button to cancel the command, click the Clear Input button to reset all of the search criteria.

6.1.18. Purging Form Submissions

Purge Form Submissions gives form owners the ability to purge one or more form submissions in a single action. This functionality is especially useful when cleaning up old or incomplete submissions.

Note:

- The user must have the Forms Manager, Admin privilege, be the owner of a form, or have delete privileges to the form submissions.

Navigation: *[DocMgr > Forms > Select Desired Form > Side Menu > Purge Submissions]*

Step 1:

To search for form submissions to purge, enter data in one or more search fields, adjust the search options if necessary, and press the Search button to submit the request. Pressing the Clear Input button will reset all of the search criteria.

Here are a few rules to remember when specifying search criteria:

- The search criteria are not case sensitive
- The asterisk (*) represents zero or more characters
- The question mark (?) represents exactly one character
- Multiple asterisks and/or question marks are allowed in the same word
- Consecutive asterisks are not allowed but consecutive question marks are allowed
- Search fields cannot contain only asterisks
- Asterisks and question marks may be positioned anywhere in a word as long as they don't break any of the previous rules

The table below describes the fields and search criteria:

Field	Search Criteria
External UUID	Enter the external UUID or part of the external UUID with wildcards.
Create Date	Search for form submissions created on a specific date. For example: submissions created on 02/03/2001. Enter 02/03/2001 to 02/03/2001 in the date boxes. Use: mm/dd/yyyy.



	<p>Search for submissions created for a range of dates. For example: submissions created from 01/18/2001 to 01/23/2001. Enter 01/18/2001 to 01/23/2001 in the date boxes. Use: mm/dd/yyyy.</p> <p>Search for submissions created since a specific date. For example: submissions created from 01/19/2001 to present date. Enter 01/19/2001 in first date box. Leave second date box blank. Use: mm/dd/yyyy.</p> <p>Search for submissions created prior to a specific date. For example: submissions created prior to 01/23/2001. Enter 01/23/2001 in second date box. Leave first date box blank. Use: mm/dd/yyyy.</p>
Last Modify Date	<p>Search for form submissions last modified on a specific date. For example: submissions last modified on 02/03/2001. Enter 02/03/2001 to 02/03/2001 in the date boxes. Use: mm/dd/yyyy.</p> <p>Search for submissions last modified for a range of dates. For example: submissions last modified from 01/18/2001 to 01/23/2001. Enter 01/18/2001 to 01/23/2001 in the date boxes. Use: mm/dd/yyyy.</p> <p>Search for submissions last modified since a specific date. For example: submissions last modified from 01/19/2001 to present date. Enter 01/19/2001 in first date box. Leave second date box blank. Use: mm/dd/yyyy.</p> <p>Search for submissions last modified prior to a specific date. For example: submissions last modified prior to 01/23/2001. Enter 01/23/2001 in second date box. Leave first date box blank. Use: mm/dd/yyyy.</p>
Completion Date	<p>Search for form submissions completed on a specific date. For example: submissions completed on 02/03/2001. Enter 02/03/2001 to 02/03/2001 in the date boxes. Use: mm/dd/yyyy.</p> <p>Search for submissions completed for a range of dates. For example: submissions completed from 01/18/2001 to 01/23/2001. Enter 01/18/2001 to 01/23/2001 in the date boxes. Use: mm/dd/yyyy.</p> <p>Search for submissions completed since a specific date. For example: submissions completed from 01/19/2001 to present date. Enter 01/19/2001 in first date box. Leave second date box blank. Use: mm/dd/yyyy.</p>

	Search for submissions completed prior to a specific date. For example: submissions completed prior to 01/23/2001. Enter 01/23/2001 in second date box. Leave first date box blank. Use: mm/dd/yyyy.
Submitter	Click on the down arrow and select the user from the list.
IP Address	Enter the IP address or part of the IP address with wildcards.
Status	Click on the down arrow and select the status from the list.
New Field	To search on a particular field in the form, you may select and add that field to the search. Once added, you may enter the exact data you are looking for or just part of the data and use wildcards. To add a form field, click on the down arrow and select the field from the list and then click the Add button.

1. To search for submissions to purge, enter data in one or more search fields. Refer to the table above for help on searching.
2. Click the Cancel button to cancel the command, click the Search button to submit the request, or click the Clear Input button to reset all of the search criteria.



Step 2:

A listing of all of the form submissions that matched the search criteria is displayed.

- If no form submissions were found that matched the search criteria, the following message will be displayed: "No form submissions found matching the specified search criteria".
 - The Form Name and External UUID are displayed for each form submissions.
 - Click on either  or  to Show Form Submission Info.
1. Click the Cancel button to cancel the command, click the Clear Input button to reset all of the search criteria, if you need to refine your search, enter data in another field and click the Search button, or click the Next button to continue.

Step 3:

A listing of all of the form submissions that matched the search criteria is displayed. You must choose the form submissions to be purged by placing a check in the checkbox in front of the form submission. You can check each form submission individually or you can click the Select All button to select all the form submissions.

- The Form Name and External UUID are displayed for each form submission.
- Click on  or the  to Show Form Submission Info.
- By default, an empty checkbox is displayed in front of each form submission.

Note:

Only form submissions with a check in the checkbox in front of the form submission will be purged.

1. Choose a form submission to be purged by individually placing a check in the checkbox in front of the form submission or click the Select All button to select all of the form submissions.
2. Click the Cancel button to cancel the command, click the Previous button to return to the previous screen, click the Unselect All button to uncheck all previously checked form submissions, click the Next button to continue.

Step 4:

1. Click the Cancel button to cancel the command, click the Previous button to return to the previous screen, or click the Next button to continue.

Step 5:

1. Enter a reason for purging the form submissions in the Reason box. This is a required field. The maximum length of this field is 255 characters.
2. Click the Cancel button to cancel the command, click the Previous button to return to the previous screen, or click the OK button to purge the form submissions.

Notes:

- The selected form submissions will be purged.
- A history record will be generated for the purging of form submissions.

7. Suggestions and Feedback

We hope you find the forms specification, designer, and engine to be very useful for your electronic form needs. As the forms engine is quite vast and very new, we have chosen to implement just the core features to begin with. As we continue to build upon the Forms Specification, Forms Editor, and engine, we can begin to add more and more features. If you have any suggestions about form components to add, or features for the design tool or forms engine, please let us know by sending us email at suggestions@docubrain.com

If you do happen to find a problem, please feel free to use the same email above. While we do prefer suggestions and compliments, we also accept criticism (preferably constructive :-)

At Prevo Technologies, Inc., we take the quality of our products very seriously. If you do find an issue or something that you feel can be improved upon, please let us know.