Bluebeam Revu

bFX Guide

Notices

This document is for informational purposes only and is provided by Bluebeam, Inc. The accuracy of the information is not guaranteed as Bluebeam products and corresponding reference documents continually evolve to adapt to market conditions. **Bluebeam makes no warranties, express or implied, as to the information in this document.** No portion of this document can be reproduced, distributed, archived or transmitted in any form, by any means (electronic, mechanical, photocopying, recording, or otherwise), for any purpose, without the express written permission of Bluebeam, Inc. Further, Bluebeam may have patents, patent applications, copyrights, trademarks, or other intellectual property covering the subject matter included in this document. Furnishing this document does not provide any license to these patents, trademarks, copyrights or other intellectual property. Any rights must be expressly provided in a written and authorized license agreement.
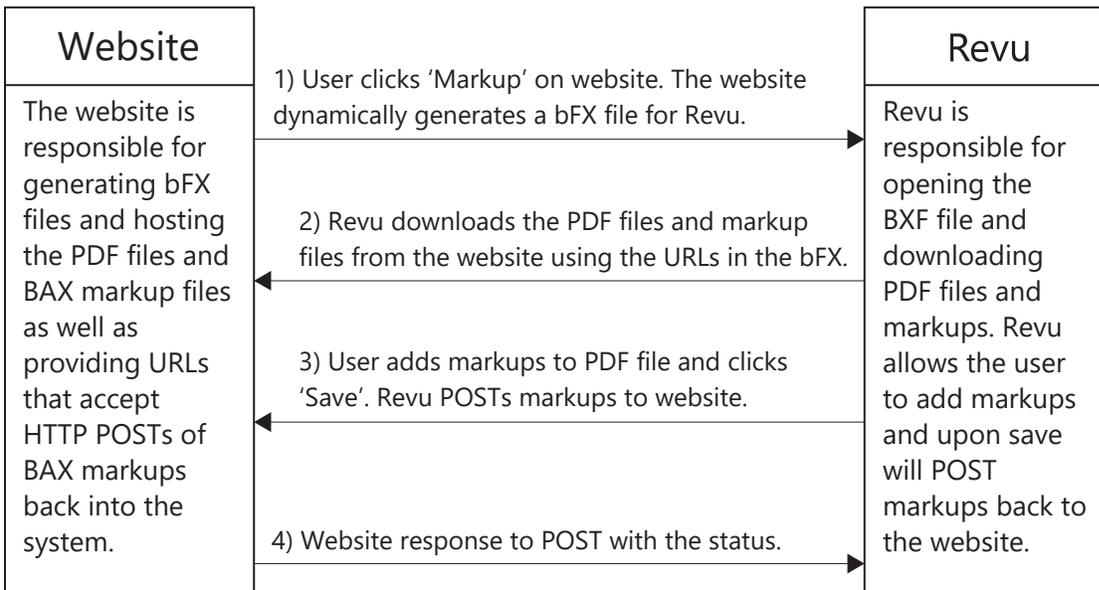
Table of Contents

Introduction

Bluebeam File Exchange (bFX) is included with every version of Revu and gives you the ability to open PDF files from remote locations and share markups on those PDFs within an internal network. A minor level of engineering is required to set up a local bFX service on your local network sever. This guide provides information on creating and accessing bFX files in Revu.

The exchange format is an XML file with the .bfx file extension. The system relies on HTTP GET and POST protocols in order to interface with the remote system. The markups are stored in an XML file with the .bax extension. The typical usage of bFX technology is as follows:

1. A third party entity creates a website that dynamically constructs bFX files based on user input. This dynamic website could be implemented using a server side scripting technology such as ASP or PHP. The bFX file contains URLs to PDF files to download as well as separate URLs to BAX markup files for interfacing with the remote system using HTTP GETs and POSTs.

2. The user would then open the bFX file in Revu, which downloads the appropriate PDF files and BAX markups.

3. The user adds markups to the PDF files and saves the file in Revu. Bluebeam then attempts to POST a BAX file back into the remote system based on the POST URL specified in the originating bFX file.

4. In response to the POST, the website then dynamically sends back a POST Response XML file.

The following diagram illustrates how such a system would work.

Website		Revu
The website is responsible for generating bFX files and hosting the PDF files and BAX markup files as well as providing URLs that accept HTTP POSTs of BAX markups back into the system.	1) User clicks 'Markup' on website. The website dynamically generates a bFX file for Revu. 2) Revu downloads the PDF files and markup files from the website using the URLs in the bFX. 3) User adds markups to PDF file and clicks 'Save'. Revu POSTs markups to website. 4) Website response to POST with the status.	Revu is responsible for opening the BXF file and downloading PDF files and markups. Revu allows the user to add markups and upon save will POST markups back to the website.

The bFX file may contain URLs to more than one BAX file so that a user may see the markups of other users. The bFX file contains information about the current user so that the markups of any other users would be read-only. The bFX file may also specify security settings to restrict the user from being able to perform certain functions such as printing.

Prior to Revu 20.2.90, a bFX file must also specify a valid provider identifier that is registered with Bluebeam, Inc. When a bFX file is downloaded and when markups are posted back into the system, Revu performs a validation check on the provider. An internet connection is therefore required for bFX to function.
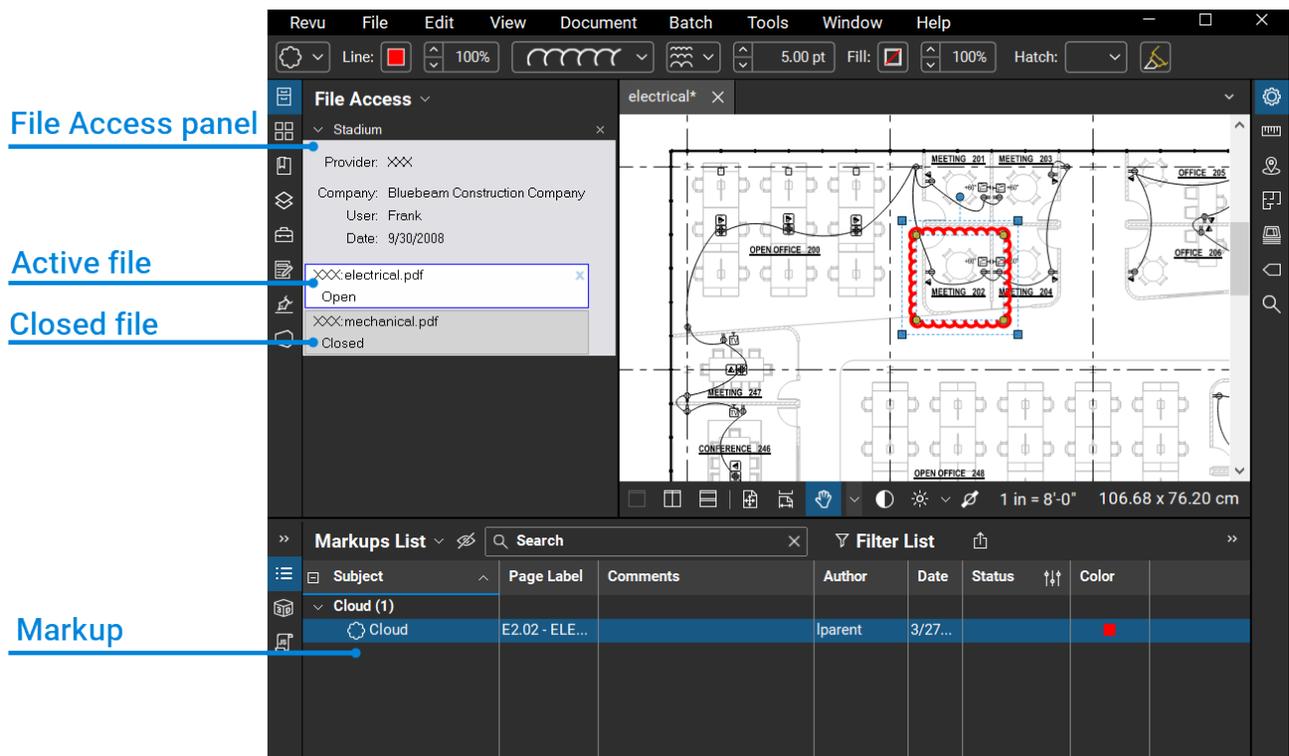
In Revu 20.2.90 and later, bFX does not include security controls or infrastructure and is best suited for use on a local intranet, not over the public internet.

To add a layer of security on top of the bFX protocol, Bluebeam recommends constructing URLs in the bFX file that contain time-restricted tokens that are unique to each specific user. For example:

https://test_domain/markup_post?ID=u123&auth=ABC123

where the *auth* token was generated for the user logged in to the bFX website and is valid for a reasonable time that allows the user to mark up the file.

Revu User Interface

Within Revu, the File Access panel provides a list of files associated with a bFX file. Select a file in order to activate it, or, if the file is not currently open, to download that file and then activate it. The File Access panel alsoprovides information about the project, date, user name and more. To post the markups back into the system, select **Save**.

bFX Format

bFX files are XML files with the .bfx extension. Bluebeam Revu is registered with the operating system to open .bfx files. A bFX file must specify a version. The current version is 1.1 and requires Revu 6.5.1 or later. Typically, a bFX file is dynamically generated for a particular user.

Sample bFX file

```xml
<?xml version="1.0" encoding="utf-8"?>
  <BluebeamRemotePackage Version="1.1" ID="pk123">
  <Provider>XXX</Provider>
  <LogoURL>www.bluebeam.com/logo.png</LogoURL>
  <ProviderURL>www.bluebeam.com</ProviderURL>
  <UserID>u123</UserID>
  <UserName>Frank</UserName>
  <CompanyName>Bluebeam Construction Company</CompanyName>
  <UserType>Subcontractor</UserType>
  <Date>DATEFORMAT</Date>
  <Project Name="Stadium" ID="p123">
    <Company>The Isotopes</Company>
    <ProjectURL>www.bluebeam.com/isotopes.html</ProjectURL>
    <File Name="electrical.pdf" ID="f123">
      <FileURL>www.bluebeam.com/isotopes/files/electrical.pdf</FileURL>
      <Size>23452</Size>
      <Date>DATEFORMAT</Date>
      <Load>True</Load>
      <Security>
        …
      </Security>
      <Comment UserID="u123" ID="c123">
        <FileURL>www.bluebeam.com/isotopes/files/markups/u123.bax</FileURL>
        <ResponseURL>www.bluebeam.com/MarkupPost.asp?ID=u123</ResponseURL>
      </Comment>
      <Comment UserID="u456" ID="c456">
        <FileURL>www.bluebeam.com/files/markups/u456.bax</FileURL>
      </Comment>
    </File>
    <File>
      …
    </File>
  </Project>
</BluebeamRemotePackage>
```

Fundamentals

Date format

The date format follows the XML standard of "yyyy-MM-ddTHH:mm:ss.fffffffZ" which includes year, month, date, hours, minutes, seconds, and time zone offset.

Markups

Any number of different BAX markup files can be referenced for a particular PDF file. If a user has not made any comments yet, the best practice would be to specify a ResponseURL only. This way, a new BAX file would be introduced into the system the first time the user selects **Save**.

If a markup file is referenced for a different user, the best practice would be to omit the ResponseURL. This way, the comments would be treated as read-only. Only markups with a ResponseURL are editable by the user.

Note that the terms *markups* and *comments* are interchangeable. In the Revu user interface and for the majority of this document, the term *markup* is used; however, the actual tags within the bFX XML use the term *comment*.

Posting

Notice in the sample bFX file that the ResponseURL takes a parameter. This illustrates a typical implementation for handling the posting of markups. The BAX markup file would be posted to the URL using the HTTP POST protocol, and the ID parameter would instruct the system of which user the POST corresponds to. This implementation style allows the system to place the markups in a database with the correct association to the user, or alternatively in a file system based approach to the particular user's folder.

The FileURL could also be implemented using similar techniques where an ID parameter would be passed as a parameter. Note that these techniques are all implementation-specific to the third party entity interfacing with Bluebeam File Exchange.

Detailed XML element descriptions

BluebeamRemotePackage

The BluebeamRemotePackage element is the base element within a bFX file. There may only be one entry and it is required.

Attributes	
Version	Required. Current version is 1.1.
ID	Required. Must be unique to each individual bFX file.
Tags	
Provider	Required. Must be issued from Bluebeam, Inc.
ProviderURL	Optional. A URL to the provider's website.
UserID	Required. Must be unique from all other users in system. This ID represents the ID of the current user for whom the bFX file was generated for.
UserName	Required. The name of the user that will be shown in the user interface of Revu.
CompanyName	Required. The name of the provider that will be shown in the user interface of Revu.
LogoURL	Optional. A URL to a graphic image that will be shown in the user interface of Revu.
UserType	Required. Implementation-specific identifiers that are tracked by Bluebeam. For example the types could be "GC," "Subcontractor," or "Owner."
Date	Required. Date value specifying the date and time that the bFX file was constructed.
Project	At least one required. See Below.

Project

The Project element is meant to group PDFs in a BRX file by Project. Revu shows the PDF files in the File Access panel grouped by the Projects. Projects are found either in the Package element, or they can be nested within other Project elements.

Attributes	
Name	Required. The name of the project that will be shown in the user interface of Revu.
ID	Required. Must be unique to each project specified in the bFX file.
Tags	
ProjectURL	Optional. A URL to the project's website.
File	At least one required. See below.
Project	Optional. Projects may be nested.

File

The File element wraps a PDF file that is accessible via a URL. At least one File element is required. It is supported under the Project element.

Attributes	
Name	Required. The actual file name that will show in the user interface of Revu.
ID	Required. Must be unique to each PDF file in the bFX file.
Tags	
FileURL	Required. The URL to download the PDF file. File system based URLs are permissible using the file:// syntax.
Size	Optional. A number value specifying the size of the file in bytes. Highly recommended as it allows Revu to show accurate progress when downloading the file.
Date	Optional. A date value specifying the date stamp of the file.
Load	Optional. A boolean value specifying whether or not to immediately download the file. For instance, if a bFX file references 200 PDF files, it might not make sense to have all of them automatically download. The

	default value is True.
Security	Optional. Supports only one. See Below.
Comment	Optional. Supports more than one. See Below.

Security

The Security element is supported under the File element and is optional. A PDF file may already have security restrictions embedded within the file. The bFX security is a secondary layer of security; the PDF security will take precedence.

Tags	
PrintDoc	Optional. Boolean value specifying whether or not printing the PDF should be allowed.
CopyForm	Optional. Boolean value specifying whether or not copying text from the PDF should be allowed.
Commenting	Optional. Boolean value specifying whether or not commenting from the PDF should be allowed.*
FillFormFields	Optional. Boolean value specifying whether or not form filling in the PDF should be allowed.*
CopyFromWAccess	Optional. Boolean value specifying whether or not copying text for accessibility reasons should be allowed.
PrintDocHigh	Optional. Boolean value specifying whether or not printing using high quality resolution is allowed. This value is ignored if PrintDoc is set to False.
SaveAs	Optional. Boolean value specifying whether or not the user may save the PDF file locally.

*These fields are reserved for future use and are not recommended to be used with version 1.1 of the bFX format.

Comment

The Comment element wraps a BAX markup file that is accessible via a URL. It optionally includes a response URL so that the user can post markups back into a remote system. Comment elements exist under the File element. Multiple Comment elements are allowed so long as none of the comments share a common UserID.

Attributes	
UserID	Required. Specifies the UserID associated with the markups. If the markups belong to the user for whom the bFX file was created for, then it is imperative that this value matches the UserID at the Package level, otherwise the comments would be readonly.
ID	Required. Must be unique to each Comment specified in the bFX file.
Tags	
FileURL	Optional. URL to download the BAX markups from. If this entry is omitted it is assumed that the user has not yet created any markups for the PDF file.
ResponseURL	Optional. URL instructing Revu where to HTTP POST the BAX markups to when the user hits save. The remote website is expected to respond with a 'POST Response File' as specified below. If this entry is omitted then the markups will be treated as read-only.

POST Response format

The POST Response file is another XML file that should be dynamically generated on the third party website in response to an HTTP POST of BAX markups. One of two different flavors of the Response XML should be generated based on either success or failure. Note that in both cases, a Version is required and is currently 1.0.

Success

In the success case, the Result Code should always be "0". A file URL is also required in the success case. This is meant for the scenario where a FileURL wasn't specified in the bFX file as the user had not yet made any comments for the file because it is likely that the ResponseURL would be different from the FileURL. Even if the bFX file contained the FileURL, it is still mandatory to be in the POST Response.

```
<?xml version="1.0" encoding="utf-8"?>
<BluebeamPostResponse Version="1.0">
  <Result Code="0">
    <FileURL>url to comments</FileURL>
  </Result>
</BluebeamPostResponse>
```

Failure

In the failure case, the Code would be non-zero. The error codes are implementation-specific and are meant to be used for business logic errors emitted from the remote system. For example, the post might fail because the user account had been canceled. The HelpURL is optional and would be shown to the user in an error dialog so that the user may click to learn more about why the error happened.

```
<?xml version="1.0" encoding="utf-8"?>
<BluebeamPostResponse Version="1.0">
  <Result Code="1">
    <Message>User does not have persmission to post.</Message>
    <HelpURL>http://www.bluebeam.com/ErrorCode1.html</HelpURL>
  </Result>
</BluebeamPostResponse>
```