



## **Librarian**

Integrating Secure Workflow and  
Revision Control into Your  
Production Environment

# Contents

---

<b>Overview</b> .....	<b>3</b>
<b>File Storage and Management</b> .....	<b>4</b>
The “Library” .....	4
Folders, Files and File History .....	4
Managing Your Files .....	5
<b>Revision Control</b> .....	<b>8</b>
<b>File States, Publishing and Workflow</b> .....	<b>9</b>
File States .....	9
File Publishing and Workflow .....	9
Don’t Forget about Security! .....	12
<b>Librarian API</b> .....	<b>13</b>
Beginning a Visual Studio Project with the Librarian API .....	13
Using the Librarian API .....	13
<b>Security</b> .....	<b>16</b>
Administration Console Permissions for Librarian .....	16
Relevant BarTender Permissions in Administration Console .....	17
Organizing Users and Groups According to their Functional Roles .....	18
Advanced Security .....	18
Other Security Considerations .....	18
<b>Appendix A: Integrating Librarian with BarTender and Companion Applications</b> .....	<b>20</b>
BarTender .....	20
Administration Console .....	20
Batch Maker .....	20
Reprint Console and History Explorer .....	20
Integration Builder .....	20
Print Station .....	21
<b>Appendix B: Syntax for Referencing Files in the Library</b> .....	<b>22</b>
Review of Standard Windows Syntax .....	22
Referencing Files in the Library .....	22
<b>Related Documentation</b> .....	<b>24</b>

## Overview

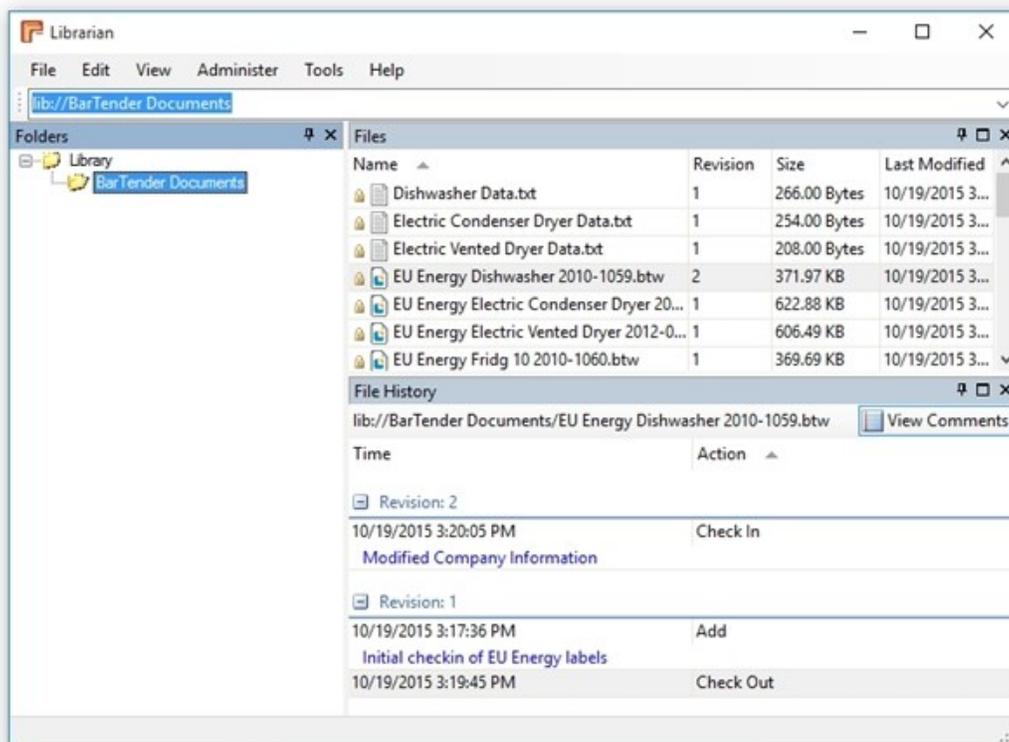
Librarian is a companion application to BarTender that lets groups of designers, administrators and other users cooperatively and securely manage the storage and revision of BarTender documents and other related files. Librarian can help you better manage any printing system, but it is particularly valuable in highly regulated and/or secure environments, such as the medical, pharmaceutical, military and chemical industries

**NOTE:** Librarian is supported only by the Enterprise Automation edition of BarTender.

Librarian allows you to store files in a centralized database rather than “loose” on various hard drives, so files are easier to track and manage. File check-in and check-out prevents users from overwriting each others' changes. You can use revision control to retain copies of all previous versions of a file so that you not only have a history of who has made what changes and when, but you can retrieve actual copies of the previous versions and even “rollback” to an older version if desired. Librarian can be used with Security Center, letting you control which users can access files within Librarian, and what they can do with them.

Librarian allows you to specify custom named “states” for files that are in different phases of the publishing process, and build upon user-defined file states by adding rules for the allowable transitions between the file states. This makes it easy to manage multiple levels of review prior to publishing.

There is an API which, using any .NET language, allows a developer to check files in and out of Librarian, roll back to specific revisions of a file, and get file information that is saved to the BarTender System Database.



# File Storage and Management

## The “Library”

Librarian securely stores BarTender documents and related files in its “Library,” which is actually a dedicated set of tables within the BarTender System Database. By using this centralized file storage location, you can easily organize and manage multiple files that would otherwise be “loose” on your hard drive. Bringing files into the Library and managing them once they are there is straightforward and relies on traditional Windows file management techniques whenever possible.

### Configuring the BarTender System Database

If you have not yet configured your BarTender System Database by running the System Database Setup utility, then you will be prompted to do so the first time you run Librarian. The System Database Setup utility will configure either SQL or SQL Express for use by the BarTender Suite.

### Implementing a Secure Database

It is important to remember that any individual with knowledge of databases can use SQL Enterprise Manager (and other database tools) to directly browse and edit a SQL database – without even installing Librarian. Therefore, if achieving the highest possible security is a priority for you, your systems administrator must take additional steps to prevent unauthorized users from accessing your SQL (or SQL Express) database using software other than Librarian.

For more information, refer to the **BarTender Security** white paper:

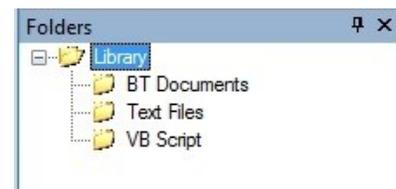
<http://www.seagullscientific.com/support/white-papers>

## Folders, Files and File History

Librarian makes it easy to view the files and folder structure you have placed into the Library, as well as a file's revision history.

### Folders Pane

The Folders pane, located on the left side of the Librarian window, lets you organize your files in any folder structure that works for you. You can add new folders, move them and rename them just as if they were folders in Windows Explorer. The root directory, “Library,” provides a basic storage location for all of your Librarian files. This directory can be accessed by any number of applications in the BarTender Suite.



### Files Pane

When you click on a folder in Librarian’s **Folders** pane, the corresponding files in that folder are displayed in the **Files** pane at the top-right of the main Librarian window.

In addition to displaying a timestamp and file size, just as the standard Windows file system does, the **Files** pane also displays how many times a file has been revised and (if states have been defined) what publishing “state” each file is currently in.

Name	Revision	Size	Last Modified
Dishwasher Data.txt	1	266.00 Bytes	10/19/2015 3...
Electric Condenser Dryer Data.txt	1	254.00 Bytes	10/19/2015 3...
Electric Vented Dryer Data.txt	1	208.00 Bytes	10/19/2015 3...
EU Energy Dishwasher 2010-1059.btw	2	371.97 KB	10/19/2015 3...
EU Energy Electric Condenser Dryer 20...	1	622.88 KB	10/19/2015 3...
EU Energy Electric Vented Dryer 2012-0...	1	606.49 KB	10/19/2015 3...
EU Energy Fridg 10 2010-1060.btw	1	369.69 KB	10/19/2015 3...

## File History Pane

The **File History** pane, directly below the **Files** pane, displays a variety of revision history information not available within the Windows file system at all. This includes a summary of the most recent transaction type on the file, who performed that change, and what state the file is in (if states have been defined).

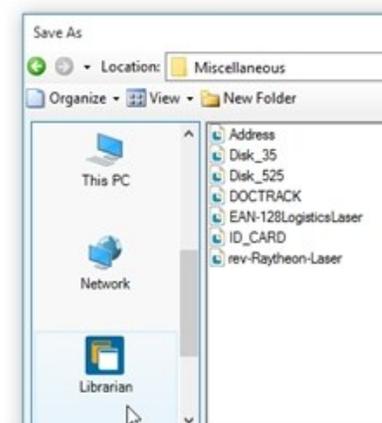
Time	Action
Revision: 2	
10/19/2015 3:20:05 PM	Check In
Modified Company Information	
Revision: 1	
10/19/2015 3:17:36 PM	Add
Initial checkin of EU Energy labels	
10/19/2015 3:19:45 PM	Check Out

## Managing Your Files

### Adding Files to the Library

There are a number of ways that you can add files to Librarian. Choose the one that works best for you:

- From Windows Explorer, you can drag a file into Librarian's **Folders** or **Files** pane.
- From the Librarian **File** menu, select **Add**, or select **Add** from the context menu in the **Folders** or **Files** panes.
- From BarTender, click the Librarian icon in the **Save As** dialog. You can browse to the Library folder where you want to save the file.



## Getting the Latest Revision

When you get the latest revision of a file, Librarian copies the last stored copy of the file in the database to a location on your hard drive. *It does not check out the file or make it writable on your drive.* It simply lets you view the latest changes made to the file.

To copy the latest version of a file, select the desired file in the **Files** pane, and then select **File > Get Latest Revision** or select **Get Latest Revision** from the context menu. This opens the **Get File(s)** dialog, where you can browse to and select a folder on your network into which you want to copy the selected file.

## Checking In and Out

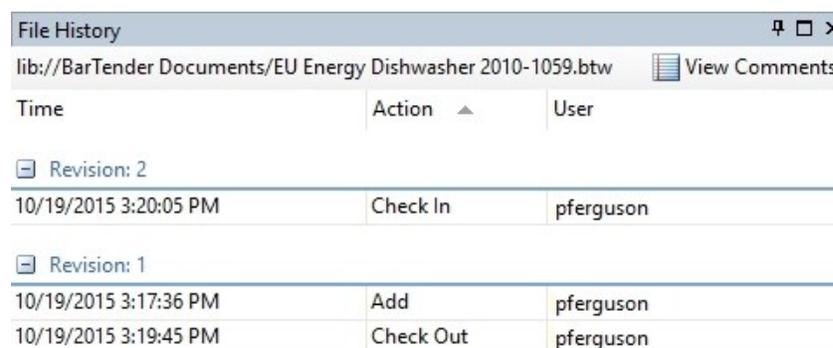
When files are simply stored “loose” on a hard disk, users can accidentally (or even deliberately) open and edit files that other users are currently working on. This creates confusion and increases the possibility that one user might overwrite another user’s changes. Librarian solves this problem with its "Check In" and "Check Out" functionality. When a user checks out the file, it makes the file writable *only for that user*. All other users are prevented from editing the file until the original user checks the file back in to the library.

When a file is checked out, it displays a red check mark (✓). When it is checked in, it displays a yellow lock (🔒).



Name	Revision	Size	Last Modified
🔒 EU Energy Dishwasher 2010-1059.btw	2	371.97 KB	10/19/2015 3...
✓ EU Energy Electric Condenser Dryer 20...	1	622.88 KB	10/19/2015 3...
🔒 EU Energy Electric Vented Dryer 2012-0...	1	606.49 KB	10/19/2015 3...

You can see who has a file checked out from the file's **Properties** dialog or in the **File History** pane. Scan the **User** column; if a user has a file checked out, his or her name will appear next to the **Check Out** action.



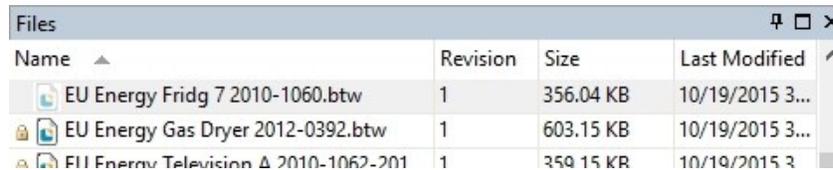
Time	Action	User
Revision: 2		
10/19/2015 3:20:05 PM	Check In	pferguson
Revision: 1		
10/19/2015 3:17:36 PM	Add	pferguson
10/19/2015 3:19:45 PM	Check Out	pferguson

If you accidentally check out a file and you want to check it back in without making any changes to the file, select the file and then choose **Undo Check Out** from the **File** menu. Any changes you made will be discarded, and the file will go back to the state it was in before you checked it out.

## Deleting and Undeleting Files or Folders

Any items you delete are removed from the Library, but can be restored to the Library at any time. To delete an item in the library, select **Delete** from the item's context menu or from the **File** menu. You can also select the item and press DELETE on your keyboard.

You can see deleted files by selecting **Show Deleted Items** from the **View** menu. When **Show Deleted Items** is enabled, all previously deleted folders and files will be shown, but their file type icons will be faded.



Name	Revision	Size	Last Modified
EU Energy Fridg 7 2010-1060.btw	1	356.04 KB	10/19/2015 3...
EU Energy Gas Dryer 2012-0392.btw	1	603.15 KB	10/19/2015 3...
EU Energy Television A 2010-1062-201	1	359.15 KB	10/19/2015 3...

To restore a previously deleted file, select the item(s) you want, then select **Undelete** from the **File** menu or from the file's context menu. After undeleting, the file is restored to the Library and you can continue to manage the file as before.

## Revision Control

Any time you save a new version of a file using a conventional file system such as the one built into Windows, the previous version is lost. Sometimes, however, you may want to review a list of previous versions and possibly even retrieve (or “rollback” to) them. These functions are helpful for troubleshooting revision problems, and essential for audit preparation. Using Librarian's revision control features, you can:

- **View your file history**, including revision numbers, state changes, the names of users who have made changes, the time and date of the revisions and any optional comments added to the file.
- **Get older file versions**, provided you have the proper security permissions to do so.
- **Rollback to earlier file revisions**, in the case you or another user may have made undesirable changes to a file.

For more information, refer to the **Revision Control** white paper:

<http://www.seagullscientific.com/support/white-papers>

## File States, Publishing and Workflow

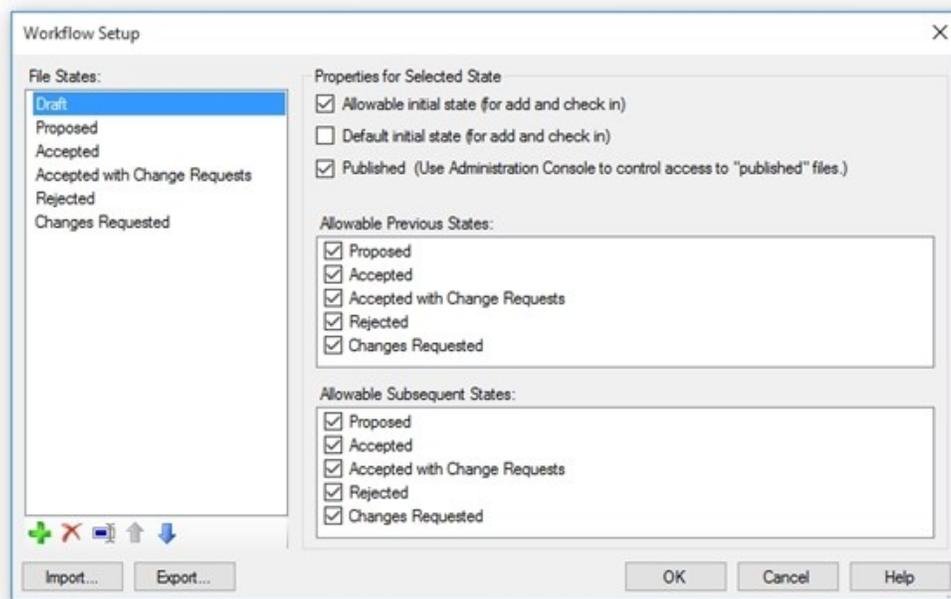
Librarian has tools to make your document production process work smoothly, and to let you keep track of a document's progress from first draft to publishing.

"Workflow" is the process of creating files and managing reviews and revisions until the file is deployed or published. Because workflow management philosophy and guidelines are often unique to a company or organization, Librarian's **Workflow Setup** dialog gives you full control over the publishing process, including the approval steps required, and who can authorize each step. This makes it easy for you to define a system of rules for creating, reviewing and publishing files, thereby avoiding the chaos of "rogue" and improvised file deployment.

### File States

Librarian lets you set a custom "state" for a file, so you can keep track of it throughout the production process. Examples of file states might include "First Draft," "Editorial Review," "Project Manager Review," "Legal Review," and so on. You can create states that are specifically tailored to your publishing needs.

You can set file states from the **Workflow Setup** dialog, available from the **Administer** menu.



For more information about file states, refer to the following topics in the BarTender help system:

- [Workflow Setup Dialog](#)
- [Assigning States to Files](#)

### File Publishing and Workflow

As long as a workgroup of colleagues strictly adheres to agreed-upon rules for how a defined system of states is used, the use of states alone represents an easy way to improve document

management. Unfortunately, such a voluntary system is highly vulnerable to the habits and attention to detail of the people using it. Librarian lets you gain better control over the management of your files by defining rules for workflow transitions. This is a major step toward the implementation of a powerful workflow management and file publishing system. You will not only be able to better organize your files, but you will also maintain more orderly control over the revision and publishing process.

In addition to defining states, Workflow Setup allows you to define a set of workflow rules to help you control your file publishing process. Once you have defined your available states, you can use the options in **Properties for Selected State** to create your desired workflow using those states.

### **Specifying Initial States for a File**

If you are using Librarian's File States functionality, any time you add a new file to the Library or check in a new revision of an existing file, you will have to assign an "Initial" state to that file in the library. Typically, when you define a workflow, you will want only some of your defined file states to be available for use as this initial state.

### **Defining the Default Initial State**

Even if you have more than one state that has its **Allowable initial state** option enabled, you can enable the **Default initial state** option for only one of them. This defines the default state displayed in the **Add File** and **Check In** dialogs.

### **What it Means to "Publish" a File**

In any file development environment, whether you are developing a BarTender document, user help, a computer program or any other kind of document, you can have both published and unpublished files in your development cycle. The Workflow model supported by Librarian assumes the following about these files:

- **Unpublished Files:** These include files that are not yet ready for use and/or viewing by the "general public" in an organization. Unpublished files can be in development, waiting for approval from a manager, and possibly even "rejected" as not yet suitable for publishing.
- **Published Files:** Once the development of a file is complete, it may be released to the general public by the author, a manager or administrator. Once "published," the file is viewable and usable by the general public.

When using Librarian, a file is considered to be "published" if the current state assigned to one or more revisions of a file has the **Published** property in the **Workflow Setup** dialog enabled.

### **Effects of the "Published" Property on File Access**

The Published property is useful when Librarian is used in conjunction with BarTender's security settings in Administration Console. Users whose permissions are limited to **Open Published Files** will only be able to open files that have been assigned a "published" state. If two or more revisions of a file stored in the Library have states with the **Published** property enabled, users whose permissions are limited to **Open Published Files** will be able to open only the most recently published revision.

Here is an example of how this might work:

1. A file is added to the Library and is assigned an “unpublished” file state.
2. As part of the publishing process, somebody manually assigns the file a “published” state. For example, a file state called “Approved” may have its **Published** property enabled.
3. Somebody then “checks out” the file for the purpose of revising it.
4. When that user checks the file back in to the Library, a second revision of that file with the same name will automatically be created by Librarian. An “initial state” gets assigned to this new revision, and its “Published” property is disabled.

Consider the different behavior encountered by the following two Librarian users, each with different permissions defined in Administration Console:

- **Users who have only Open Published Files permission:** In the example above, this class of users will be able to open only the initial revision of the file, because it is the only one that has been assigned a state with the **Published** property enabled.
- **Users with Modify Files permission:** When opening the file from the BarTender **File** menu, this class of users will have the option of opening both the published and the more recently revised unpublished version of the file.

For more information, refer to the **BarTender System Security** white paper:

<http://www.seagullscientific.com/support/white-papers>

## Defining Valid File State Transitions

When you first create file states, no rules are automatically created governing how and when you can assign them to a file or change a file’s state. Therefore, defining file states alone merely gives you a system by which a group of cooperative workers have the opportunity to keep better track of their file publishing process.

To implement a true file publishing “workflow,” you also need to define allowable file state transitions. For example, if you choose to have states called **Draft**, **Proposed**, and **Accepted**, you can impose an approval process by requiring that files that are set to **Draft** must first be **Proposed** before they can be **Approved**. (This is just an example. You can name your file states anything that you like, and create your own custom rules governing their workflow.)

To define the allowable transitions between states, use the **Allowable Previous States** and **Allowable Subsequent States** controls in the **Workflow Setup** dialog.

Once you have done this, any time you open the **Change State** dialog in order to change the state of a file, only states that have been specified as **Allowable Subsequent States** for the currently active file state will be listed as options in the **New State** dropdown list.

## Importing and Exporting Workflows

All of the settings in the **Workflow Setup** dialog can be imported from and exported to files. If you have already defined some states, and you import a Workflow, your existing list of available states will be replaced. In this case, if a named state in the current workflow is already in use by one or more files when a new workflow is imported, and it is not also named in the new workflow, then you will be prompted to remap the old state to one of the new states.

### ***Don't Forget about Security!***

For secure file management and revision control, you also need to properly configure your system's security settings. This includes properly assigning access rights using Windows' security system and configuring the Librarian sections of Administration Console. Refer to the [Security](#) chapter in this white paper.

## Librarian API

The Librarian API lets you develop custom applications, in any .NET language, that use Librarian's primary functions to embed revision control and workflow settings into your environment. Using the Librarian API, you can check files in and out of Librarian, roll back to specific revisions of a file, and get file information that is saved to the BarTender System Database.

### *Beginning a Visual Studio Project with the Librarian API*

To begin a project in Visual Studio 2005 or higher:

1. From the **File** menu, select **New**, and click **Project**.
2. In the **Project Types** panel, under either the **Visual C#** or the **Visual Basic** node, select **Windows**.
3. In the **Templates** pane, select **Windows Application**.
4. Name the project and click **OK**.
5. After the project is created, add a reference to `Seagull.BarTender.Print`.
6. In the **Solution Explorer** pane, right-click on the **References** node and select **Add Reference** from the context menu.
7. In the **Add Reference** window, select the **.NET** tab, select `Seagull.BarTender.Print`, and click **OK**.
8. Make sure the project is able to build and compile by pressing **Ctrl+Shift+B** on your keyboard. If successful, you are ready to begin your project.

### *Using the Librarian API*

#### **Adding Files to the Repository**

When you add a file to Librarian, you enable users on the network to access the file from the BarTender System Database. Subsequent edits or changes to the file are then also recorded to the System Database, which allows users to revert to or track previous changes made to a file.

A BarTender file is added to the repository by calling the Library class **AddFile** method. The following code demonstrates how to add a file to Librarian.

In C#:

```
Library library = new Library();
LibrarianFile file = library.AddFile("lib://Formats/Address.btw",
@"C:\Users\Administrator\Documents\Address.btw", "Added Address.btw to the
library.");
```

The first parameter passed in the `AddFile` method is the path of the folder where you want to save your document, along with the file name. The second parameter specifies the path and filename of the file to add, and the third parameter is the comment that describes the addition of the file into the repository. The Library class provides an overload for the `AddFile` method to specify additional advanced options for adding a file.

## Checking In Files

A file can be checked into the library using the LibrarianFile's CheckIn method. The CheckIn method retrieves a file from a specified location on your computer, and saves it back to a folder in the Library.

Several CheckIn overloads exist; the simplest requires the path and filename of the file on your computer, as well as a comment describing changes made to the file. The following code shows how to check the file "Address.btw" into the **Formats** directory of the Library.

In C#:

```
Library library = new Library();
LibrarianFile file = library.GetFile("lib://Formats/Address.btw");
int revisionNumber = file.CheckIn
(@"C:\Users\Administrator\Documents\Address.btw", "Modified the label design.");
```

The first parameter in the CheckIn method specifies where to retrieve the file on your computer. The second parameter specifies a comment describing changes made to the file while it was checked out. Additional parameters can specify advanced check in options.

## Checking Out Files from the Repository

A file can be checked out from the library using the LibrarianFile's CheckOut method. The CheckOut method saves the file to a specified location on your computer and makes the file available for editing.

Several CheckOut overloads exist; the simplest requires only the path and filename of the location where you want to save the file. The following code shows how to check out the file "Address.btw" from the Repository.

In C#:

```
Library library = new Library();
LibrarianFile file = library.GetFile("lib://Formats/Address.btw");
file.CheckOut(@"C:\Users\Administrator\AppData\Local\Temp\Address.btw");
```

The first parameter in the CheckOut method specifies where to save the file. Additional parameters can specify advanced check out options.

## Displaying Folder Contents

Files and folders in Librarian are stored in folders, consisting of a single root directory and its subdirectories. In Librarian, the root directory is the Library folder. This directory can contain subdirectories that you create, such as "Templates" or "Images." Each subdirectory can contain any number of subdirectories.

You can list the contents of a directory in Librarian using the LibrarianFile's Name and/or Path properties combined with an iteration statement, such as a For loop. The following code shows how to display a list of files in a directory stored in Librarian.

In C#:

```

Library library = new Library();
LibrarianFolder folder = library.GetFolder("lib://Templates");

// Iterate through the subfolders
StringBuilder listingBuilder = new StringBuilder();
foreach (LibrarianFolder subfolder in folder.Folders)
    listingBuilder.AppendFormat("Folder\t{0}\t{1}\tChildren: {2}\n",
        subfolder.Name, subfolder.Path, subfolder.Children.Count);

// Iterate through the files in the directory
foreach (LibrarianFile file in folder.Files)
    listingBuilder.AppendFormat("File\t{0}\t{1}\tRevision: {2}\n", file.Name,
        file.Path, file.RevisionNumber);

// Display the contents of the folder
System.Windows.Forms.MessageBox.Show(listingBuilder.ToString(), string.Format
    ("Folder Contents ({0})", folder.Path));

```

In the above example, a list of subdirectories and files is saved to a list of strings, listingBuilder. Once the list has been populated with the contents of a directory, the name, path, and/or revision number of each subdirectory or file is displayed in a message box.

For more information, refer to the **BarTender .NET SDK > Librarian API** book in the BarTender help system.

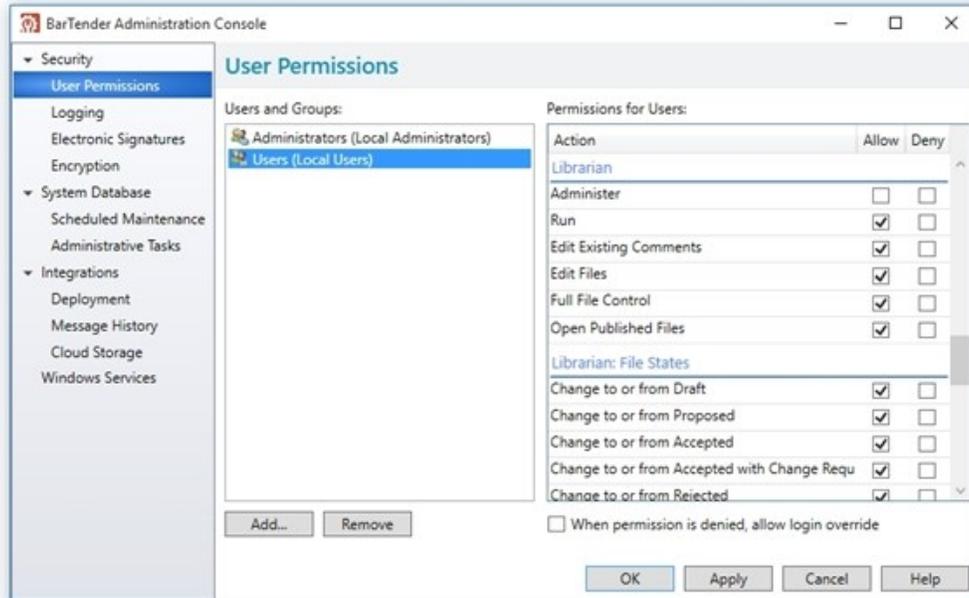
**NOTE:** The **BarTender .NET SDK** book will appear in BarTender help only if you have installed the BarTender .NET SDK.

## Security

To implement a secure printing system, you must ensure that the user permissions specified in Administration Console allow only authorized personnel to access Librarian and BarTender.

### Administration Console Permissions for Librarian

There are two sections of permissions in Administration Console that apply to Librarian: **Librarian** and **Librarian: File States**.



### Librarian Permissions

The **Librarian** section of Administration Console defines the following permissions:

- **Run:** Allows users to run the Librarian application. What users are able to do once they run Librarian is defined by the rest of the permissions listed here. Users that have *only* the **Run** permission granted will still be able to view files and their history.
- **Administer:** Lets users access System Setup and Workflow Setup from Librarian's **Administer** menu, and to modify their settings. *This permission should only be granted to users who require the highest level of administrative control within Librarian.*
- **Edit Existing Comments:** Controls whether or not users can edit existing comments that have been logged when someone has checked a file in, renamed a file or performed some other action on it. The ability to add new comments is controlled by the **Edit Files** permission below.

- **Edit Files:** Lets users perform the following actions:
  - Add comments to a file
  - Add files to the library
  - Check in files
  - Check out files
  - Roll back to previous revisions
  - Open files stored in the Library
- **Full File Control:** Allows users to move, rename, delete and undelete files within the Library, in addition to **Edit Files** permissions, above.
- **Open Published Files:** Allows users to open files with a state marked **Published**.

### Librarian: File States Permissions

The **Librarian: File States** section of Administration Console lists all of the user-defined file states in Librarian. The rules defined in Librarian’s **Workflow Setup** dialog for “previous” and “subsequent” allowable states determine file state changes that are possible in Librarian. In addition, using Administration Console, you can control the states that individual users and groups can assign to a file. This is especially useful for limiting the number of users who are authorized to change files to a “published” state.

For more information about file states, refer to the [File States, Publishing and Workflow](#) chapter.

### Relevant BarTender Permissions in Administration Console

Two of the permissions listed in the BarTender permissions section of Administration Console apply to Librarian:

- The **Print Published Documents** permission allows users to print the latest published revision of a document stored in the Library. Authorizing a user to print documents means that they will also have permission to print published documents. However, because a file can only be assigned a state if it is kept in the Library, files stored loose on a hard drive rather than in the Library can never be assigned a “published” state. Therefore, denying BarTender users permission to **Print Published Documents** *prevents them from even opening documents from their individual hard disks*. This can prevent the loading, creation and printing of unapproved documents.
- **Print Unpublished Documents** lets a BarTender user print both unpublished documents stored in the Library, and all documents that are stored loose on a computer or the network. Users who are not given this permission will only be able to open documents that are stored in the Library *and* whose state is has its **Published** property enabled. The **Print Unpublished Documents** permission includes permission to **Print Published Documents**.

For more information about the **Published** property, refer to the [File States, Publishing and Workflow](#) chapter.

## **Organizing Users and Groups According to their Functional Roles**

When planning the Librarian file states and Administration Console permissions that your file management system will use, it is useful to classify users into functional roles and set their permissions accordingly. When you have multiple users performing similar functional roles, it is easier to organize them into domain user groups so that you can manage all of the group's permissions at once, instead of requiring permission settings to be separately defined for each individual user.

Each organization has its own favored approach to user security and file publishing. Although there is no "one size fits all" security model that will work for everybody, certain general categories of users will typically be required in order to implement "secure" printing in many organizations. For example, you might organize users into File Designers, File Editors, Print Operators, and Publishers.

## **Advanced Security**

Here are some more advanced techniques for creating and implementing a secure file publishing workflow within Librarian.

### **Electronic Signatures**

One way to improve the security in your publishing system is to specify in Administration Console that users must enter an electronic signature (or user credentials) when changing a file's state. This requirement can be specified in the **Librarian: File States** section of the **Electronic Signatures** tab in Administration Console.

### **Logging**

From the Administration Console **Logging** node, you can log any or all user requests to perform actions listed in the **Librarian** section and the **Librarian: File States** section to the BarTender System Database. Then, you can review which users are requesting to perform which actions.

### **Encryption**

From the Administration Console **Encryption** node, you can encrypt individual BarTender documents. Note that encryption is an advanced security technique, and comes with certain risks that you must guard against.

### **Controlling the Printing of Unpublished Documents**

Any time a user prints out an unpublished (and therefore an unapproved) BarTender document, it represents a security risk. Accordingly, the **Print Unpublished Documents** permission should only be granted on an as-needed basis. In addition, you may wish to require the entry of electronic signatures when printing out unpublished documents. This guards against the possibility of unauthorized individuals using File Editors' or Publishers' computers to print unpublished BarTender documents. It also ensures that you have a log of exactly which authorized users printed out unpublished documents and when.

## **Other Security Considerations**

It is important to remember that individuals with only moderate knowledge of databases can use SQL Enterprise Manager (and other database tools) to directly browse and edit a SQL database –

without even installing Librarian. Therefore, if achieving the highest possible security is a priority for you, your systems administrator must take additional steps to prevent unauthorized users from accessing your SQL (or SQL Express) database using software other than Librarian.

## Appendix A: Integrating Librarian with BarTender and Companion Applications

You can use files stored in Librarian with many of the applications included in the BarTender Suite. This list details how each application can work with Librarian in your production environment.

### **BarTender**

You can edit BarTender documents and access related files (such as databases or images) stored in Librarian directly from BarTender.

### **Administration Console**

Use Administration Console's **Librarian** and **Librarian: File States** permissions to control whether users can:

- Edit existing comments in a document stored in Librarian
- Edit files stored in Librarian
- Have full file control over files stored in Librarian
- Open published files stored in Librarian
- Change file states.

For more information, refer to the [Security](#) chapter.

### **Batch Maker**

BarTender batch files (with an extension \*.btbat) can be saved in the Library and edited with Batch Maker. Additionally, batch files can refer to BarTender documents that are stored in the Library.

### **Reprint Console and History Explorer**

Reprint Console and History Explorer only interact with Librarian indirectly via BarTender. BarTender (.btw) documents being viewed and/or reprinted may reside in the Librarian database.

### **Integration Builder**

Integrations can be saved to and opened from the Library, and Integration Builder can reference files in the Library.

#### **Librarian Syntax and Integration Builder**

Librarian referencing syntax can be used in Integration Builder to reference a file from a command line. Integration Builder can use Librarian syntax with BTXML script, which is used when executing commands from within Integration Builder that launch and control BarTender print jobs.

For more information on Librarian syntax, refer to [Appendix B: Syntax for Referencing Files in the Library](#).

## ***Print Station***

You can choose a Librarian folder as the root folder for Print Station. This is the folder that Print Station uses to display available documents to be printed.

For more information about the companion applications, refer to their corresponding white papers:

<http://www.seagullscientific.com/support/white-papers>

## Appendix B: Syntax for Referencing Files in the Library

Just as there is a standard syntax for referencing files on a hard drive, Librarian also provides a syntax for referencing files in the Library.

### Review of Standard Windows Syntax

Because Windows users often use their mouse to click through folders and subfolders to find files, they may not remember that the correct file path syntax that Windows uses is:

```
C:\<path>\<filename>
```

For example, to open a BarTender document, you might reference a file in this manner:

```
C:\MyTemplates\ShippingLabel.btw
```

### Referencing Files in the Library

The syntax for referencing files in Librarian's Library more closely resembles an internet location than a hard disk location. The general syntax is:

```
lib://<path>/<filename>
```

Note that this internet-style URL uses forward slashes instead of backslashes. The "lib://" at the beginning is always required to reference files in the Library. Using this syntax, the file reference used in the previous example would change to the following when accessing a file in the Library instead of on hard disk:

```
lib://MyTemplates/ShippingLabel.btw
```

You can reference a file in the Library using any of these methods:

- From the BarTender or Librarian **File** menu, select **Open**. In the **File Name** field, type or paste the text string for the file you want to reference.
- From Batch Maker, select **Open Batch** from the **File** menu, and type or paste the text string for the batch you want to reference.
- From Print Station, type or paste the text string for the file you want to print into the **Search** field.
- Execute BarTender using command line parameters. Here is a sample command line that instructs BarTender to open a document from the Library and launch a print job:

```
bartender.exe /F="lib://templates/shipping/overnight.btw" /P
```

- When BarTender is reading data-sourced graphics while printing.
- By running BarTender XML Script.
- Executing commands from within Integration Builder that launch and control BarTender print jobs.

For more information on how Librarian interacts with BarTender and the companion applications, see [Appendix A: Integrating Librarian with BarTender and Companion Applications](#).

## Related Documentation

### White Papers and Manuals

- Getting Started with BarTender
- BarTender System Security
- Revision Control
- Administration Console

For downloadable versions, visit:

<http://www.seagullscientific.com/support/white-papers>

### BarTender Help System

- [Librarian](#)

