BagIt Usage Instructions

Matt Schultz, MetaArchive Cooperative
Stephen Eisenhaur, UNT Libraries
Nick Krabbenhoeft, MetaArchive Cooperative

2014/04/29
Version 1.0

Contents

1. Overview of BagIt

1.1. What is BagIt?

BagIt is a file packaging format intended for storing and moving digital content in an organized fashion. A BagIt-formatted package is called a “bag,” and a bag is really nothing more than a directory with a particular structure and some key metadata stored in text files that provide information about the Bag’s content (file inventory, file checksums, payload size, etc.).

You can read more about the BagIt specification here:
https://confluence.ucop.edu/display/Curation/BagIt

There are a few different tools for creating bags, some of which we’ll guide you through using in this document.

1.2. Organizing Your Bags

The BagIt format requires no special organization for the files you are packaging. All of your original file and folder names and hierarchies are preserved, and you are always free to Bag up your collections as is. Nevertheless, even though BagIt doesn’t prescribe a hierarchy or organization onto a group of files you should give some thought to how you can Bag your collection(s) at appropriate levels so that they can be received by another party as meaningful units. As you do so, you can label your Bags accordingly with locally meaningful information such as control numbers, unique identifiers, file/folder name conventions, etc.

1.3 Using Bag Metadata

Bags contain a number of metadata fields that help to describe the Bag's contents and origin. This metadata is contained in a file named **bag-info.txt**. Most BagIt tools have ways of predefining the set of metadata, ensuring consistent use across bags.

Below is a listing of potential metadata fields with a description.

Field Name	Description
Source-Organization	The organization where the bag was made. (No abbreviations)
Organization-Address	The address of the Source-Organization.
Contact-Name	The name of the person responsible for the bag.
Contact-Phone	Phone number of the person from Contact-Name.
Contact-Email	Email address of the person in Contact-Name.
External-Description	A description of the bag's contents for those outside of your organization.
Bagging-Date	The date the bag was created.
External-Identifier	A sender-supplied identifier for the bag.
Bag-Size	The size of the bag. This is usually created by the bagging tool.
Payload-Oxum	A byte count and file count signature for the bag. This is usually created by the bagging tool.
Bag-Group-Identifier	A unique name given to a group of more than 1 bag.
Bag-Count	This bag's sequence number, if part of a group of bags.
Internal-Sender-Identifier	The ID assigned to this content internally at your institution, if any.
Internal-Sender-Description	A written description of the contents of the bag based on internal standards.

1.3.1. Creating a Custom Metadata Profile

The following instructions are for use with the graphical user interface (GUI) BagIt utility known as Bagger (see 2.2.1 Creating Bags Graphically Using Bagger below). Bagger allows users to define custom metadata profiles with a text file formatted as JSON data. Users may define any metadata fields they wish. Additionally, any metadata field may be pre-populated with a default value, have a standard value across all bags, or have a controlled list of values.

Below are the instructions for creating a custom profile.

1. Open a new file in a text editor such as Notepad on Windows or SublimeText on Windows and Mac (the free version asks user to purchase a license every ~15 saves). **Programs like Microsoft Word and TextEdit will not save JSON correctly.**

2. The entire profile must be contained in curly brackets, {}.

3. Each metadata field must be defined in quotation marks followed by a colon, an opening curly bracket, a property in quotation marks, followed by a value in quotation marks, a closing curly bracket, followed by a comma. Example below:

   ```
   "Organization-Address" : {
      "defaultValue" : "Recommended, insert mailing address"
   },
   ```

4. Within the curly brackets for each metadata field, four properties may be defined. Each property is contained in quotation marks followed by a colon and the value of the property in quotation marks, except for true and false values. Properties must be separated by a comma. Multiple values can be defined but must be separated by a comma.

   a. `fieldRequired` - Define if a field must be completed or not. If true, the user must provide information. If false, the field is optional. If this property is not defined, the default is false.

      i. `"fieldRequired": true,`

        ii. true and false (Boolean) values do not need to be in quotes

    b. `requiredValue` - Define a value that will always be included and cannot be edited. This is useful for fields that will always contain the same information such as addresses.

        i. `"requiredValue": "fixed value",`

    c. `defaultValue` - Define a value to populate the field. This value can be edited. It may contain either a recommended value for the field such as “owner@example.com” or instructions to complete the field such as “Please format tel. as xxx-xxx-xxxx”.

        i. `“defaultValue": "some value",`

    d. `valueList` - Define a controlled list of values to populate a field. Users may not enter values not on this list.

        i. `“valueList" : ["first value",”second value”,....”last value”],`

        ii. The entire list is contained by square brackets.

5. Save the finished profile as “profilename”-profile.json in the correct directory.

    a. On Mac OS X and Linux, this is `~/bagger`.

    b. On Windows, this is `C:\"Documents and Settings"\<user>\bagger.`

6. Restart Bagger to load your profile.

2. Getting Started with BagIt

2.1 Getting Started - Deciding How to Create Your Bag

There are always two different approaches you can take when creating a bag, regardless of which tools you decide to use, and it's very important to understand the distinction:

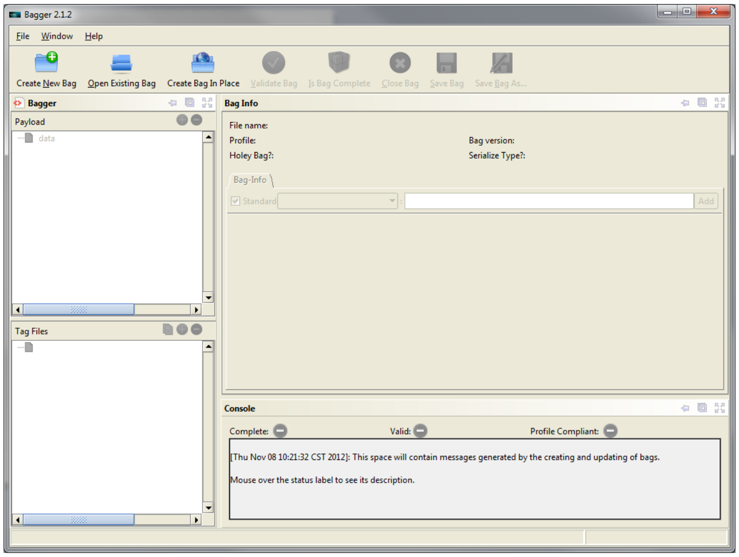
In-Place Bag Creation	Creating a Copy
Creating a bag "in place" means transforming the original folder where your data resides into a bag. **This causes all of the file locations in that folder to change**, which is problematic if those files are being hosted by a web server or any other software that expects the data to be in the same place it was before. For instance, consider if you're bagging a folder named "Theses" which contains several PDF files. If you turn this folder into a bag in-place, those PDF files will be moved to a new folder named "data" within "Theses". Be sure to understand the consequences this may have before trying it on live data.	The alternative to creating a bag in-place is creating a brand new bag somewhere else, with a copy of the original data. This leaves the original data unmoved and unchanged, but keep in mind that **this effectively doubles the hard drive space the data will need to occupy** on the system. If you're trying to bag up gigabytes of data, you may need to consider the practical implications of duplicating that much data.

2.2 Getting Started – Using BagIt Tools

2.2.1 Creating Bags Graphically Using Bagger

If you administer your collections directly on a system with a graphical interface, the easiest way to create a bag is to use Bagger. Bagger is an easy-to-use application for creating and verifying bags. It's easy to run on Windows computers, though it can also work on Mac OS X and Linux systems with some finessing (see Bagger README files for details on this).

This guide uses version 2.1.3 of Bagger, but the latest stable release is always recommended.



File name:

Profile:

Holey Bag?:

Bag version:

Serialize Type?:

Complete:

Valid:

Profile Compliant:

```
[Thu Nov 08 10:21:32 CST 2012]: This space will contain messages generated by the creating and updating of bags.

Mouse over the status label to see its description.
```

Requirements

Bagger requires Java.

The Bagger Application needs to access the Java Runtime Environment (i.e. Java Runtime Environment 6) on the user's machine. For Linux/Ubuntu systems use OpenJDK Runtime Environment 6 (preferably the latest release).

If Java Runtime 6 is not installed or it is not set in the System Path, then alternatively the JAVA_HOME environment variable needs to be set in the bagger.bat (i.e. Windows) or bagger.sh (Linux/Ubuntu) files provided in the bagger-2.1.3 folder as follows:

<u>i) WINDOWS (File Path has space)</u>

```
SET JAVA_HOME="C:\Program Files\Java\jre6\bin"

%JAVA_HOME%\java.exe -jar bagger-2.1.3.jar -Xms512m -classpath
spring-beans-2.5.1.jar;bagger-2.1.3.jar
```

<u>ii) WINDOWS (File Path with no spaces)</u>

```
SET JAVA_HOME=C:\jre6\bin

%JAVA_HOME%\java.exe -jar bagger-2.1.3.jar -Xms512m -classpath
spring-beans-2.5.1.jar;bagger-2.1.3.jar
```

<u>iii) Linux/Ubuntu</u>

```
JAVA_HOME = /usr/java/jre/bin

$JAVA_HOME/java.exe -jar bagger-2.1.3.jar -Xms512m -classpath
spring-beans-2.5.1.jar;bagger-2.1.3.jar
```

Note: The above steps are just examples and could be avoided if the Java Runtime Environment 6 is set in the System Path, where the path or name of the Java Runtime Environment folder could be different.

Setup

To install and open Bagger:

1. Visit http://sourceforge.net/projects/loc-xferutils/files/loc-bagger.

2. In the table on that page, click on the zip file for the most recent release of Bagger.

3. After the download finishes, extract the files to a location of your choice (if you're not sure, choose your Desktop).

4. Open the folder you extracted bagger into in the last step.

5. If you're using Windows, double-click the icon titled **bagger.bat** (MS-DOS Batch File) to launch the app.
   Note: Be sure to launch Bagger using the bagger.bat file rather than the bagger-x.x.x.jar file. If you launch the .jar file directly, Bagger will not have enough resources to handle large sets of data.

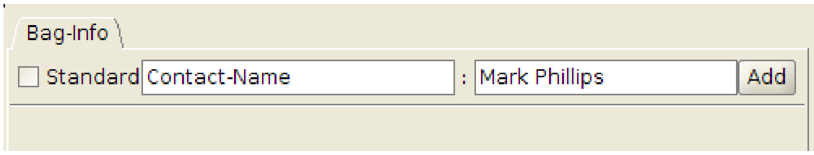
6. Exit Bagger by closing its window.

7. Bagger is now ready to use.

Creating a Bag

1. Launch Bagger, if it is not already open, according to step 5 from the "Setup" instructions above.

2. First, we add metadata to the bag. Please refer to 1.3 Using Bag Metadata for a description of potential metadata. If you are using a custom profile as described in 1.3.1. Creating a Custom Metadata Profile, skip to step 3 and select your profile from the drop down menu.



☐ Standard Contact-Name : Mark Phillips

   i. In the middle part of the window, in the "Bag Info" section, click on the drop-down menu to the right of the word "Standard" and select the tag you want to set. (If the tag you want to set is not in the list of available tags, uncheck the checkbox to the left of the word "Standard" and you will be able to type the tag name yourself.

ii. Type the tag's value into the text box to the right of the tag's name.

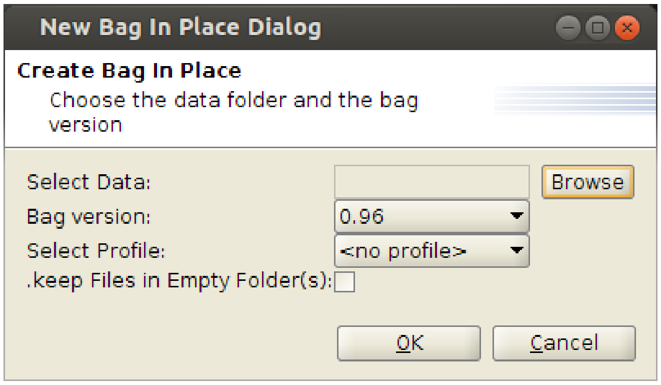
iii. Click the **Add** button to the right of the text box to finish adding the tag.

iv. Repeat the last 3 steps for any additional tags you wish to add.

3. Depending on which bagging approach you've chosen to use, follow the appropriate instructions below in order to save the bag:

a. For in-place bag creation:

i. Click the **Create Bag In Place** button on the toolbar. A dialog window will appear.



New Bag In Place Dialog

Create Bag In Place

Choose the data folder and the bag version

Select Data:

Bag version: 0.96

Select Profile: <no profile>

.keep Files in Empty Folder(s): □

ii. Click the **Browse** button to the right of "Select Data". A file browser dialog will appear.

iii. Navigate into the folder you wish to transform into a bag.

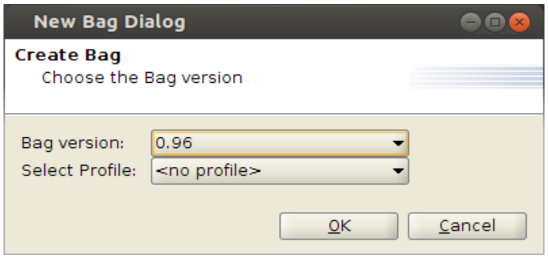
iv. Click the **OK** button.

v. A message will appear, confirming that the bag has been saved. Click the **OK** button to dismiss it.

b. To create your bag as a copy:

i. Click the **Create New Bag** button on the toolbar. A dialog window will appear.

ii. Click the **OK** button.

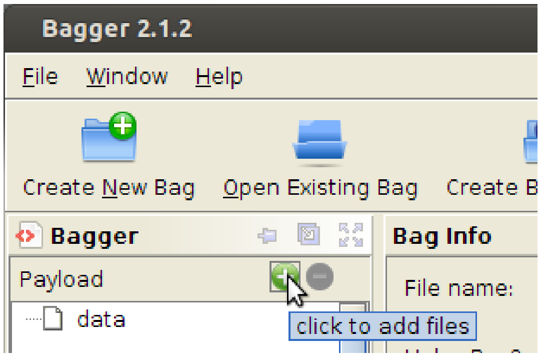


Bag version: 0.96

Select Profile: <no profile>

iii. If you selected a profile, complete the metadata fields presented.

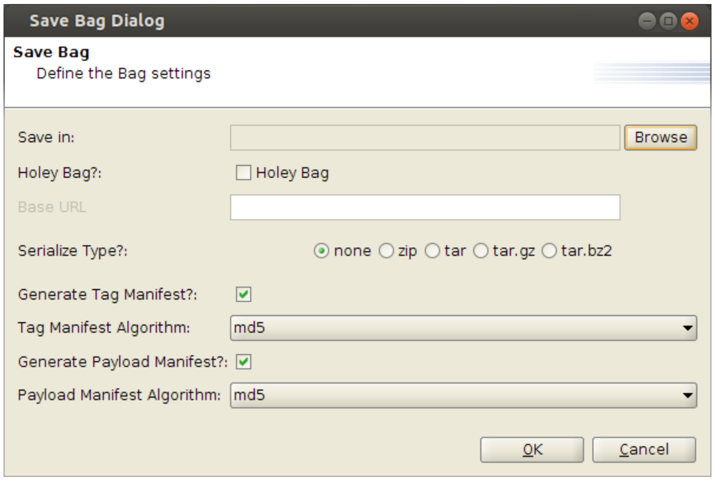
iv. In the left column of the Bagger window, click the green "+" button to the right of the "Payload" heading.



v. Navigate to and select the file or folder you wish to copy into the bag, then click the **Open** button.

vi. Repeat the previous two steps until all desired content has been added into the bag.

vii. Click the **Save Bag As** button. A dialog window will appear.



Save Bag
Define the Bag settings

Save in:

Holey Bag?: ☐ Holey Bag

Base URL

Serialize Type?: ● none ○ zip ○ tar ○ tar.gz ○ tar.bz2

Generate Tag Manifest?: ☑

Tag Manifest Algorithm: md5

Generate Payload Manifest?: ☑

Payload Manifest Algorithm: md5

viii. Click the **Browse** button. A file browser dialog will appear.

ix. Navigate to the location where you wish to store the new bag, give it a name, and then click the **Save** button.

x. Click the **OK** button. (Do not change any additional settings.)

xi. A message will appear, confirming that the bag has been saved. Click the **OK** button to dismiss it.

4. Exit the application.

2.2.2 Creating Bags from the Command-Line

If you administer your collections on something like a Linux server, you may prefer a command-line approach. In our experience, the easiest command-line BagIt tool to use is bagit.py, a module for Python that you can read about here: https://github.com/edsu/bagit.

Setup

Installation of bagit.py is handled through the Python package manager, pip. If pip is not installed on your system, you should be able to install it using your distribution's package manager. (For Ubuntu and other Debian-based systems, try `sudo apt-get get install python-pip`). If you need additional help obtaining pip, visit the project's homepage: http://www.pip-installer.org/

Once you have pip installed, install bagit.py by running:

```
sudo pip install bagit
```

Creating a Bag

Note: bagit.py only supports creating bags in-place. If this is unacceptable, you'll need to duplicate the target directory first and then use bagit.py on the duplicate.

Use the following command to transform a directory into a bag:

```
bagit.py path/to/your/directory
```

You can specify metadata values for the bag while creating it as follows:

```
bagit.py –source-organization=“example organization” \
     --organization-address=“example address” \
     …
     path/to/your/directory
```

Use `bagit.py --help` to see a full list of flags. If you need to set metadata tags that don't have a command-line flag, or if you'd prefer not to use the provided flags, you can set them by hand by opening **bag-info.txt** in a text editor after bagging.

2.3 Getting Started - Transferring Your Bags to Another Party

The three most common methods for transferring BagIt packages to another party include:

1. Making the Bag(s) accessible on a local server that another party can visit to ingest the bag(s) via an http or https request. Your institution may need to adjust firewall rules to

allow the requesting IP addresses to perform their retrieval. Use of firewalls and/or encryption is recommended.

2. Uploading the Bag(s) to a destination server provided by another party, perhaps through the use of some combination of ssh and scp or through sftp. In these events the receiving party should provide you with all of the necessary logins and advise on the use of various client software.

3. Copying the Bag(s) to portable media such as flash drives, external hard drives, etc.

BagIt is designed to facilitate checking for completeness and integrity of all contained files in such transfers. Additional BagIt helper scripts for various transfer scenarios are also available at: https://github.com/metaarchive. They include the following:

- **Bag Split/Unsplit Utility:** a python script to split a large bag upon ingest into smaller bags to facilitate network transfers or archival management with the reverse option to reassemble the original bag after dissemination.
- **Bad Filename Detection Tool:** a python script to detect problematic system files and bad file names that can sometimes interfere with a transfer validation.
- **Bag Description Service:** a python script to augment bag content with technical metadata produced by the FCLA Description Service.

3. Further Reading

If you'd like to learn more about BagIt, here are some resources we find helpful:

- BagIt Wikipedia article: http://en.wikipedia.org/wiki/BagIt
- BagIt official specification: http://tools.ietf.org/html/draft-kunze-bagit-08
- BagIt introduction presentation: https://docs.google.com/presentation/d/1FcFqLa3OUUA7uhYEbzQRfD3K5WLh2OTGYCE8e9fY36U/edit