

Three horizontal bars of equal length, colored orange, green, and blue from left to right.

MetaArchive BagIt Usage Instructions

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 oxums, 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 as meaningful units. As you do so, label your Bags accordingly with locally meaningful information such as control numbers, unique identifiers, file/folder name conventions, etc.

1.3 Using Bag Metadata for MetaArchive

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 setting metadata values themselves (preventing you from needing to edit this file directly).

Below is a listing of bag metadata fields (required/recommended/optional) we would like you to use for your MetaArchive ingest, as well as some instructions on how to appropriately fill out their values.

Field Name	Priority	Description	Example
Source-Organization	Required	The organization where the bag was made. (No abbreviations)	Institution name
Organization-Address	Recommended	The address of the Source-Organization.	Insert proper mailing address here
Contact-Name	Required	The name of the person responsible for the bag.	Insert primary collection contact person
Contact-Phone	Required	Phone number of the person from Contact-Name.	Insert primary collection person's contact phone
Contact-Email	Required	Email address of the person in Contact-Name.	Insert primary collection person's contact email
External-Description	Required	A thorough description of the bag's contents for those outside of your organization.	Can be a truncated summary of Dublin Core collection descriptive metadata or a list of included collections if more than one collection is included in the "bag"
Bagging-Date	Required	The date the bag was created on.	Insert Bag creation date

Field Name	Priority	Description	Help
External-Identifier	Recommended	A sender-supplied identifier for the bag.	control number, unique identifier, or collection folder filename repurposed for naming this bag
Bag-Size	Required	The size of the bag. This is usually set for you by the bagging tool.	Set for you by the BagIt utility
Payload-Oxum	Required	This is usually set for you by the bagging tool.	Set for you by the BagIt utility
Bag-Group-Identifier	Optional	A unique name given to the “bag group” if this bag is part of a bag group (more than 1 bag).	This identifier must be unique across the sender's content, and if recognizable as belonging to a globally unique scheme, the receiver should make an effort to honor reference to it.
Bag-Count	Optional	This bag's sequence number, if part of a “bag group”.	Ex: 1 of 2
Internal-Sender-Identifier	Optional	The ID assigned to this content internally to your institution, if any.	Insert local identifier for the bagged collection if applicable
Internal-Sender-Description	Optional	A sender-local prose description of the contents of the bag.	e.g., Dublin Core collection descriptive metadata

A complete version of the recommended MetaArchive profile above is available from Educopia here (<https://docs.google.com/file/d/0B1gETO3iL-OsejhOZIRtV3c1OGM/edit?usp=sharing>). If you are using Bagger you can save this JSON file in the appropriate directory depending on your platform (see Steps 5 & 6 below).

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 Section **2.3.1. 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. An example of a well-defined file then follows.

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 in the correct JSON format.**
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 proper mailing address"
},
```

4. Within the curly brackets for each metadata field, four properties may be defined. Again, each property is contained in quotation marks followed by a colon and the value of the property in quotation marks. Multiple values can be defined but must be separated by a comma.
 - a. fieldRequired - Define if a field must be completed or not. If not included,



- i. "fieldRequired": true
 - ii. true does not need to be in quotes
 - b. requiredValue - Define a value that will always be included and cannot be edited.
 - i. "requiredValue": {"some value"},
 - c. defaultValue - Define a value to populate the field. This value can be edited. It may contain either a recommend 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": {"some value", "another value", ..., "last value"},
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 - MetaArchive BagIt Ingest Process

2.1 Getting Started – General Overview

Below is a step-by-step description of how to prepare and stage your collections for MetaArchive using BagIt. Please note that as you work through the process you will need to obtain a series of scripts from MetaArchive to complete various steps. These scripts are available for download at <https://github.com/metaarchive>.

1. Make sure the data being bagged is ready to be preserved (this requires scanning for unwanted dotfiles/hidden files (e.g. Thumbs.db, .DS_Store, .htaccess) that can interfere with later bag validation
 - a. You can request a **find-bad-files.py** script and set of instructions from MetaArchive using support@metaarchive.org. The script is available on the MetaArchive GitHub at: <https://github.com/MetaArchive/metaarchive-ga-tools>.
2. Create the initial bag containing the whole collection using either Bagger, the Java-BagIt Library, or bagit.py
 - a. See the Sections **2.2 Getting Started – Deciding How to Create Your Bag** and **2.3 Getting Started – Using BagIt Tools** below
3. If the initial bag is greater than 30GB, split the bag into smaller 30GB bags
 - a. Request the **bag-split.py** script and usage instructions from MetaArchive using support@metaarchive.org. The script is available on the MetaArchive GitHub at: <https://github.com/MetaArchive/bagit-split>.
4. Make the split-bags (which will have been created by default in "<bag>_split/") accessible to LOCKSS via a web server
 - a. See Section **2.4. Getting Started – Staging Your Bags for Transfer** below

5. Place an appropriate LOCKSS manifest.html file in the bags' parent directory
 - a. Request **manifest.html** file from MetaArchive via support@metaarchive.org.
6. Create a Conspectus collection entry and perform a test ingest
 - a. If you have never performed this step with other collections please request an orientation from MetaArchive via support@metaarchive.org.

2.2 Getting Started - Deciding How to Create Your Bag

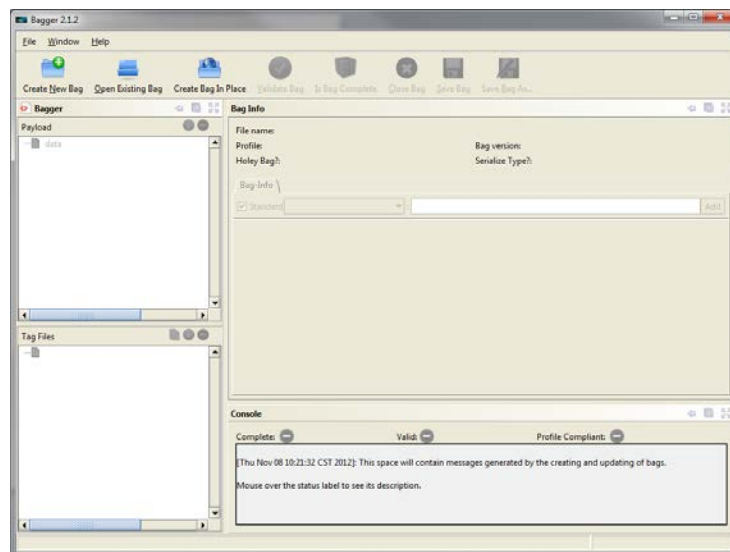
There are always two different approaches you can take when creating a bag, regardless of which tools you decide to use, and it's very important to understand the distinction:

In-Place Bag Creation	Creating a Copy
<p>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 a big problem 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.</p> <p>For instance, consider 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.</p>	<p>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.</p> <p>If you're trying to bag up gigabytes of data, you may need to consider the practical implications of duplicating that much data.</p>

2.3 Getting Started – Using BagIt Tools

2.3.1 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).



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:

Three horizontal bars of equal length, colored orange, green, and blue from left to right.

i) WINDOWS (File Path has space)

```
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
```

ii) WINDOWS (File Path with no spaces)

```
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
```

iii) Linux/Ubuntu

```
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/2.1.3/>
2. In the table on that page, click **bagger-2.1.3.zip**
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-2.1.3.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 the section of this document titled "Using Bag Metadata" for a description of the various metadata tags we'll use and whether each tag is required/recommended/optional.



The screenshot shows a window titled "Bag-Info". Inside, there is a section with a checkbox labeled "Standard" which is currently unchecked. To the right of the checkbox is a text box containing "Contact-Name". Further right is a colon separator, followed by another text box containing "Mark Phillips". To the right of this second text box is an "Add" button.

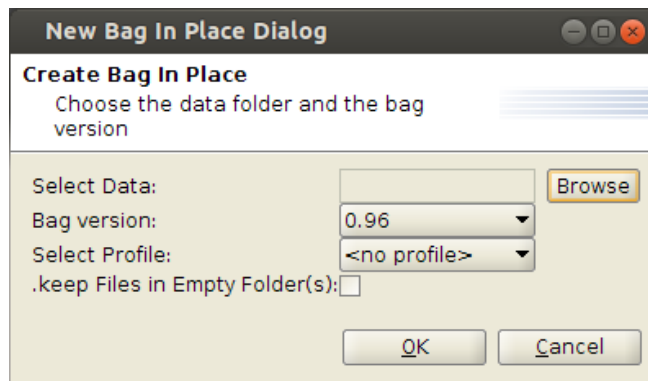
- i. In the middle part of the window, in the "Bag Info" section, click on the drop-down menu to the right of the word "Standard" and select the tag you want to set. (If the tag you want to set is not in the list of available tags, uncheck the checkbox to the left of the word "Standard" and you will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

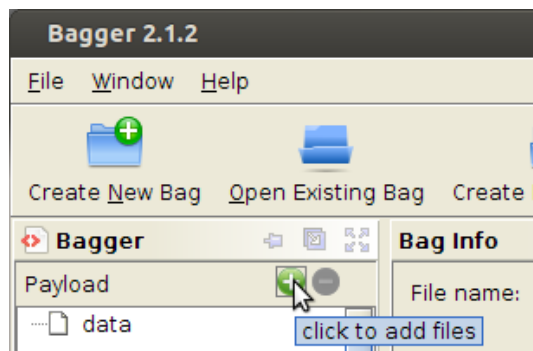


- ii. Click the **Browse** button to the right of "Select Data". A file browser dialog will appear.
- iii. Navigate into the folder you wish to transform into a bag.
- iv. Click the **OK** button.
- v. A message will appear, confirming that the bag has been saved. Click the **OK** button to dismiss it.
- b. To create your bag as a copy:
 - i. Click the **Create New Bag** button on the toolbar. A dialog window will appear.

- ii. Select **<no profile>** from the Select Profile menu.

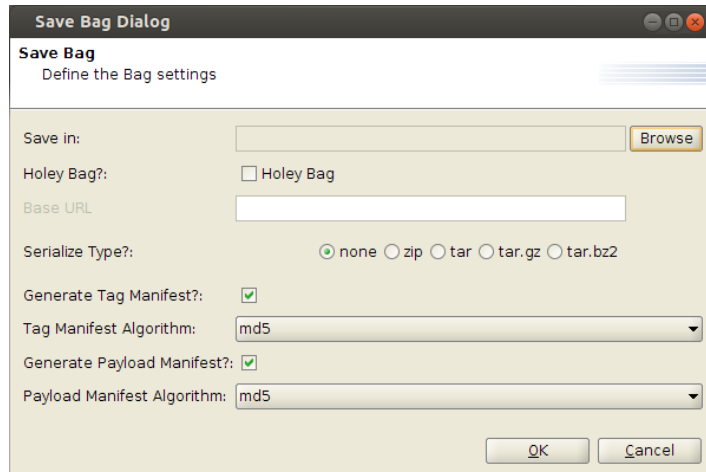


- iii. Click the **OK** button.
- iv. In the left column of the Bagger window, click the green “+” button to the right of the “Payload” heading.



- 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.



The image shows a 'Save Bag Dialog' window with the title 'Save Bag' and subtitle 'Define the Bag settings'. It contains the following fields and controls:

- Save in:** A text input field with a 'Browse' button to its right.
- Holey Bag?:** A checkbox labeled 'Holey Bag' which is currently unchecked.
- Base URL:** A text input field.
- Serialize Type?:** A group of radio buttons with options: 'none' (selected), 'zip', 'tar', 'tar.gz', and 'tar.bz2'.
- Generate Tag Manifest?:** A checkbox which is checked.
- Tag Manifest Algorithm:** A dropdown menu currently showing 'md5'.
- Generate Payload Manifest?:** A checkbox which is checked.
- Payload Manifest Algorithm:** A dropdown menu currently showing 'md5'.
- At the bottom right are 'OK' and 'Cancel' buttons.

- 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 of the settings or checkboxes.)
- xi. A message will appear, confirming that the bag has been saved. Click the **OK** button to dismiss it.

4. Exit the application.

2.3.2 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 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 create a duplicate copy of 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="University of North Texas" \
--organization-address="Some Address" \
--contact-name="John Doe" \
--contact-email="john.doe@unt.edu" \
--external-sender-identifier="ark://..." \
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.

2.4 Getting Started - Staging Your Bags For Transfer

The recommended transfer mechanism for exchanging your BagIt content with MetaArchive will be to make your bag(s) accessible on a local server that MetaArchive servers will visit to ingest the bag(s) via secure http request. Your institution may need to adjust firewall rules to allow the requesting MetaArchive IPs to perform their retrieval. These will be provided prior to ingest.

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>