

Software Acquisition and Preservation Procedures and Workflow

Contents

About this Document	2
Acquiring Software for Preservation	2
Software Format	3
Cataloging the Software	3
Inventory Sheet Locations	3
Cataloging Acquisitions	4
Cataloging Software for Processing	4
Software Submission Information Package	5
Creating the SIP	5
Note On Dependent Software	7
Folder Structure General Layout	7
SIP Structure Example	8
Creating the Software Disk Image	9
Photographing Physical Media	9
Imaging a Physical Media	9
Imaging a Digital Media	10
Disk Image Quality Assurance Testing	11
Getting Started	11
QC Workflow Excel File	11
Install Report Word File	12
Emulators	13
DOSBox	13
EaaSI	13
Outcomes	14
Functional Software	14
Demo or Restricted Software	14
NonExecutable Software	14
Ingesting Software Preservation Collection Content into Medusa	14

File Naming Conventions	14
SIP Ingest structure	15
Collection Location in Medusa	16
Ingesting the SIP into Medusa:	16
Accruing to an Existing Structure	16
Overwriting Duplicate Content	18
Document Version History	19

About this Document

This document is intended to serve as a master workflow for software preservation practice at the University of Illinois. It outlines the major task categories of acquiring, cataloging, structuring, imaging, quality assurance (QA)/quality control (QC) testing and ingesting the software package into the Medusa digital preservation repository.

Acquiring Software for Preservation

- 1) Purchase software
 - a) Software still in production
 - i) Check if it is available through the Software Center:
<https://www.library.illinois.edu/staff/it/howto/softwarecenter/>
 - ii) If not available through the Software Center:
 - (1) There are number of issues with purchasing/licensing software per the State of Illinois procurement guidelines. Contact Library Business and Human Resources Service Center (BHRSC) before ordering anything through Library IT. Note procedures tend to change frequently and are often inconsistent in how they're applied. Prepare to wait.
 - (2) Library IT procedures for requesting Hardware/Software:
<https://www.library.illinois.edu/staff/it/techsupport/hardsoftpurchase/> (accessed 2019-12-13)
 - b) Legacy software through eBay or other secondary sellers
 - i) Contact BHRSC. Prepare to wait.
- 2) Software included in an acquired computing environment
 - i) Retain a copy of the deed of gift or inventory transfer agreement.
 - ii) Image the computing environment. Continue on to imaging and note metadata you will need to run the image in the emulation environment.

Software Format

- 1) Software received via email and/or downloaded
 - a) Print to PDF any associated purchase information such as email invoice or confirmation.
 - b) Review documentation for any sensitive information such as P-Card or credit card numbers.
 - i) If there is sensitive information, redact it using Adobe Acrobat.
 - ii) Store the PDF in PDI\docs directory in SIP, created in later steps.
 - c) Download install files to be placed in the OriginalFiles directory in the SIP.
 - d) If available, download user guides and manuals from the software developer website. Save them to be placed in the PDI/Docs directory.
 - e) Continue on to cataloging the software.
- 2) Software received on physical media
 - a) Software will likely be kept on Tracy's desk in 425 until a physical location in 429 is established.
 - b) Review the file structure on the physical media. If the media contains several versions of installation files demarcated by operating system, create a SIP structure that represents the OS division (see examples below in creating a SIP)

- c) Continue on to cataloging the software

Cataloging the Software

Inventory Sheet Locations

There are currently several software inventory sheets, each serving a slightly different purpose:

File Name: Software_Inventory_Tracking_Jan2020.xlsx

Local Location: \\lib-scproc\BornDigital\BornDigitalReformatting\SoftwarePreservation (Current as of 2020-2-25)

Description: This is the working document for all software that has been processed to be ingested into the digital repository. It is located on the lib-scproc network drive next to the software SIPs it documents.

File Name: SoftwareInventory_Preservation.xlsx

Local Location: G:\AcqCatPres\Preservation&Conservation\Departments\Digital Preservation\Born Digital Curation and Preservation\SoftwarePreservation (Current as of 2020-2-25)

Description: Software inventory list, created by Tracy. This documents software that has been obtained.

File Name: 2020_2_25_SoftwareInventory_PhysicalItems.xlsx

Local Location: G:\AcqCatPres\Preservation&Conservation\Departments\Digital Preservation\Born Digital Curation and Preservation\SoftwarePreservation

Description: A copy of the software inventory list created by Tracy. Modified to pull information for the working document above.

File Name: 2019_softwareCollection_Inventory_WorkingCopy.xlsx

Local Location: G:\AcqCatPres\Preservation&Conservation\Departments\Digital Preservation\Digital Preservation and Processing Tools\HembroughSoftwareDonation (Current as of 2020-2-25)

Description: Inventory list that is specific to the Hembrough Collection. This will be used to supplement the information needed for processing of software from this collection.

Cataloging Software

- 1) Open the latest version of the Software_Inventory_Tracking excel file: (Current as of 2020-2-25)
<\\lib-scproc\BornDigital\BornDigitalReformatting\SoftwarePreservation>
- 2) Create a new entry for each software item you will be adding. Be as specific as you can.
 - a) Full software title, not including version.
 - b) Assign a barcode to each item.
 - i) Physical items will get a barcode sticker (14 numeric code).
 - ii) Digital items will be assigned a 14 character alphanumeric code.
 - iii) This 14 character string will be used for the disk image name, dir/hash file names, and photograph names. Further identification will likely not be needed, as they will be within the folder structure that identifies the software and OS already.
 - c) Software version
 - d) Operating system
 - i) Abbreviate Windows to 'Win' and Macintosh to 'Mac'.

- ii) For multiple operating systems of the same brand, separate oldest and newest with a dash (Ex: Win NT-Vista).
- iii) For multiple operating systems of different brands, separate with “ / ” (Ex: Win / Mac).
- e) Select a software type from the drop-down menu.
- f) License information, if known.
- g) Type of media.
- h) Number of media.
- i) Provenance: where the media came from, if known.
- j) Photographed/Imaged/HashDir Made/Processed/Ingested: For later. List the date each event was completed for that item.
- k) List additional notes, such as if the media came with a physical manual.

Software Title	Barcode	Version	OS	Software Type	License Information	Type of Media	# of Media	Provenance	Photographed	Imaged	Hash/Dir Made	Processed	Ingested	Notes
Band in A Box	30112121782004	2010	Win NT-Vista	Music Production	n/a	External Harddrive	1	physical acquisition	1/28/2020					Contains manual
Band in A Box	30112121781865	2018	Win XP-10 / Mac OSX 10.4 or later	Music Production	n/a	External Harddrive	1	physical acquisition	1/14/2020					Contains manual: PDF found
Cakewalk	30112121786997	Professional 5.0	Win 3.1-95	Music Production	n/a	CD	1	physical acquisition	1/14/2020	1/17/2020	1/24/2020			

Software Submission Information Package

To prepare the software for preservation, create the Submission Information Package (SIP) structure and store the empty directory on the BornDigital storage server working space: (Current as of 2020-2-25)

<\\lib-scproc\BornDigital\BornDigitalReformatting\SoftwarePreservation>

General File Naming Conventions

- 1) Be sure your entries contain no spaces or special characters. File names are limited to text and underscores.
 - a) Instead of spaces, use CamelCase capitalization to indicate a new word (e.g. Microsoft Office Suite would be written as MicrosoftOfficeSuite)
- 1) The top-most directory is the software title or some other unique identifier.
- 2) If a previous version of the software title already exists in Medusa, be sure to use the already established Software Title so that newly accrued content can be merged into the established Software Title directory.
- 3) For software with long titles, abbreviate when possible.
- 4) If there is not already a Software Title use Wikidata as the software title naming authority:

https://www.wikidata.org/wiki/Wikidata:Main_Page
- 5) If the software version contains a period, use an underscore instead. (Ex: 5.0 = 5_0)
- 6) Abbreviate Windows and Macintosh to ‘Win’ and ‘Mac’ respectively.
 - a) Full details of the OS specifications will be included in the README.
 - b) For programs that say they can go on either Windows or Mac, see if there is a clear divide in the installation files. If there is, split them up. If there isn’t, say ‘Win_Mac’.

- 7) If the software comes with multiple disks/media, create a duplicate entry per item, giving each disk a different number/title. This will create a separate folder per item in the InstallFile folder.

Creating the SIP

- 3) Open the EmulationSIPCreator excel file, located here: (Current as of 2020-2-25)
<G:\AcqCatPres\Preservation&Conservation\Departments\Digital Preservation\Born Digital Curation and Preservation\SoftwarePreservation>
- 4) Follow the directions on the "Instruction" tab, using the information from the inventory tracking sheet to generate the folder structure for the software.

	A	B	C	D	E
1	Storage Location	Software Title	Software Version	Operating System / Version Edition	Disk Number (If multiple)
2	\\lib-scproc\BornDigital\BornDigitalReformatting\SoftwarePreservation	BandInABox	2010	Win	
3		BandInABox	2018	Win_Mac	
4		Cakewalk	Professional_5_0	Win	
5		Cakewalk	Pyro_5_0	Win	
6		Finale	25	Win	
7		Finale	25	Mac	
8		FinalePrintMusic	2002	Win	
9		MacOS_9	1_0	Mac	
10		Sibelius	7	Win_Mac	InstallerDisk1
11		Sibelius	7	Win_Mac	Content1
12		Sibelius	7	Win_Mac	Content2
13		Sibelius	7	Win_Mac	Content3

Note On Dependent Software

Some software may be designed to go specifically with another program.

1. If the software is included with the installation files, store it with the installation files.
2. If the software is independent or noted as a requirement but not included in the installation files (like requires Adobe Acrobat v. 3), find and download those files as a separate SIP (if possible), note the dependency in the primary software package README file, and link to the location of the dependent program in Medusa.

Folder Structure General Layout

This organization provides one access point for the specific program, with the different versions of the software in a subfolder and a folder containing operating system specific installation files beneath that. It allows for easy addition of additional operating system installation files related to the specific software title version as they are acquired and imaged.

- **Software Title** (in CamelCase, abbreviating if too long)
 - **Version** (with _ for ".")
 - **README file**
 - Info about software dependencies
 - Info about specific operating system requirements
 - **Operating System** (Win, Mac, Win_Mac)
 - **InstallFiles**
 - Disk Image
 - Barcode of object for file name
 - **PDI**
 - **Docs**
 - Supplementary documents, if available
 - PDFs of manuals, if any
 - **Manifest**
 - Hash
 - Directory listing
 - Barcode of object for file names
 - **ObjectJPG**
 - Photographs of physical carrier, if any
 - Barcode of object for image names

SIP Structure Example

ProTools version 7.3 that runs on a Mac, 7.3 that runs on Windows, and a 6.0 for a Mac:

- ProTools
 - 6_0
 - README file (REQUIRED. Information in this file consolidates essential metadata about the software and package into one file. (See README file template: ReadmeTemplate_example.txt)
 - Mac (Operating System. This may not always be apparent but should be created if you can determine what operating system title and version the software runs in. Note that this information will be included in the README file.)

- InstallFiles (folder where the installation disk image or installation files are stored)
 - ISO/Disk image; install files
 - PDI (folder where contextual and information about the software, its operation, keys, acquisition and installation information is stored)
 - Docs (May contain manuals or other supplemental information related to the software operation or notes about software acquisition, use, installation keys, etc.)
 - Manifest (an inventory of the files in the associated installation image)
 - ObjectJPG (photographs of the physical media and packaging if available).
 - 7_3
 - README file
 - Mac
 - InstallFiles
 - PDI
 - Docs
 - Manifest
 - ObjectJPG
 - Windows
 - InstallFiles
 - PDI
 - Docs
 - Manifest
 - ObjectJPG

Creating the Software Disk Image

Photographing Physical Media

Photograph the housing of the physical media to include in the SIP's ObjectJPG folder. Follow the instructions for photographing from the reformatting project, available on the wiki:

<https://wiki.illinois.edu/wiki/display/LibraryDigitalPreservation/Photographing+Media+and+Enclosures+for+Documentation>

Once the photographs are added to the appropriate folders, update the inventory list to show you have completed that step.

Imaging a Physical Media

Image the media on which the software executable is stored.

- 1) The imaging tool will be dependent upon source media:
 - a) Optical Disk: ISOBuster

- b) Floppy: whichever tool reads the disc. Catweasel, FD5025, Kryoflux
 - c) HDD: FTK Imager or ISO Buster
 - d) SDD or USB: FTK Imager
- 2) Image the software. Refer to the reformatting wiki for more information for using each tool:
(Current as of 2020-2-25)

<https://wiki.illinois.edu/wiki/display/LibraryDigitalPreservation/Media+Reformatting+Services++Internal+Documentation>

- a) Once the objects are imaged, update the inventory list to show you have completed that step.
 - b) Generate hash files and directory listings. Update the inventory list.
- 3) Gather metadata about title and document in README file
- a) Create a software preservation object README file. ReadmeTemplate_example.txt

i) Fields in the README Template:

(1) GENERAL INFORMATION

- (a) Software Title:
- (b) Barcode:
- (c) Version/Edition:
- (d) Software Developer:
- (e) Software Publisher:
- (f) Year Published (or date on label): <suggested format YYYYMMDD>
- (g) License Information/Number:
- (h) Type of Media:
- (i) Number of Media:

(2) SYSTEM REQUIREMENTS

- (a) Operating System(s):
- (b) RAM:
- (c) Other:
- (d) Is authentication required?:
- (e) If yes, what method for authentication?

(3) SOFTWARE DEPENDENCIES

- (a) Other software required to run this program:

(4) FILE OVERVIEW

- (a) Default save file format:
- (b) Other save file formats:
- (c) File formats program can open:

(5) SHARING/ACCESS INFORMATION

- (a) Link(s) to online resources, if available:
- (b) Date resources were accessed:

Imaging a Digital Media

Software will be obtained through a download or be contained on a digital storage device such as an external hard drive, flash drive, etc.

- 1) Review the install file structures and document how the install files are divided (are there different versions of install files based on OS? Is there some other division on the media?).
- 2) Zip the content and copy it into the SIP InstallFiles folder. Update the inventory list.
- 3) Generate hash files and directory listings. Update the inventory list. Gather metadata about title and document in README file

a) Create a software preservation object README file. Use ReadmeTemplate_example.txt

i) Fields in the README Template:

(1) GENERAL INFORMATION

- (a) Software Title:
- (b) Barcode:
- (c) Version/Edition:
- (d) Software Developer:
- (e) Software Publisher:
- (f) Year Published (or date on label): <suggested format YYYYMMDD>
- (g) License Information/Number:
- (h) Type of Media:
- (i) Number of Media:

(2) SYSTEM REQUIREMENTS

- (a) Operating System(s):
- (b) RAM:
- (c) Other:
- (d) Is authentication required?:
- (e) If yes, what method for authentication?

(3) SOFTWARE DEPENDENCIES

- (a) Other software required to run this program:

(4) FILE OVERVIEW

- (a) Default save file format:
- (b) Other save file formats:
- (c) File formats program can open:

(5) SHARING/ACCESS INFORMATION

- (a) Link(s) to online resources, if available:
- (b) Date resources were accessed:

Disk Image Quality Control Testing

Before the software is added to the repository, it needs to be reviewed as part of a quality control process. QC is intended to test the viability of the software image. To do this, you have the option of using a variety of emulation systems including DOSBox and EaaSI (Emulation-as-a-Service Infrastructure).

The main goals of performing the quality control checks are:

1. To see if the disk image(s) can be read and contains all the data needed for that program to run
2. To determine what specific OS requirements for running this program are
3. To see if the program the disk image contains is what the label says it does
4. To see if the program is something that would be useful for the Software Preservation Project

Getting Started

1. Identify which software you will be testing. Open that software's SIP folder.
2. Create a folder within the software's PDI folder labeled "QC".
 - a. The creating of the folder now instead of at the beginning gives a quick visual check to see if QC has or has not been performed on this particular software.
 - b. This folder will house all the documentation that is going to be generated during the QC process.
3. Review the inventory listing to see what kind of environment is recommended to run this program. This will determine what emulator you should use. Through emulation, you will also be able to get some more information as to the specifics on the software to be included in the metadata.

QC Workflow Excel File

1. Locate the file called "2019_SoftwareInstall_Workflow_BLANK.xlsx" This Excel document contains a list of questions to answer while you review the software.
The template is located here: (Current as of 2020-3-24)
<G:\AcqCatPres\Preservation&Conservation\Departments\Digital Preservation\Born Digital Curation and Preservation\SoftwarePreservation>
2. Create a copy of the file in the software's corresponding QC folder, and rename it as "[barcode]_QCworkflow.xlsx"
3. Begin the QC procedure by filling out as much as you can. If the software in question does not fit certain criteria or you have no answer, use "NA"
 - a. Date test was finished
 - i. Filled in at the end with the date you finished the testing.
 - b. Tester name
 - c. About the Software
 - i. Use the inventory list sheet to fill out as much as you can in this category.
 - ii. If there is no information given or available for a particular category, mark "NA".
 - d. Software Source
 - i. Most of this information can also be filled in using the inventory list sheet.
 - ii. If there is no information given or available for a particular category, mark "NA".

- e. Refer to the section below to create the Install Report Word file. From here, you will be documenting everything you do as you try to install the software.
 - f. Local Install Test
 - i. Note the UIUC inventory tag of the computer you are working on, and its OS.
 - ii. Attempt to install the software directly onto your computer.
 - iii. In the Install Report Word file, make note of what you did and record/screen shot any errors or messages you get.
 - g. Emulation Install Test
 - i. Specify what emulator you are using.
 - ii. Document your steps in the Install Report Word file.
 - iii. Include a copy of your testing summary, created at the end of the Word file.
 - h. Software Execution
 - i. Opening Files
 - i. Try to find a file you believe the new software will be able to open.
 - ii. Document what you tried to open. If you can't find anything note it and move on. Try not to spend too long on this part of the process.
 - j. General notes
 - i. Did anything stand out to you during the testing process?
 - ii. Record other relevant things that doesn't fit in any other category.
4. If you perform another test using a different emulator, in the same excel document create a duplicate of the testing tab and fill it out with information from the second test. Label the tabs as the emulator used.

Install Report Word File

While performing the QC, you will document the steps you took to try and install the software, and the options and responses from prompts provided as part of the software installation process.

1. Open a new Word document and save it in the software's QC folder as "[barcode]_InstallReport.docx"
2. At the top, briefly state the name of the software and the emulator you are using.
3. As you go through the ingest procedure, document all the steps you took and screenshot the menus, errors, or messages the program gave you.
 - a. The free program "Greenshot" is very useful for capturing screenshots in this process. Once you designate an area to capture, there is an option in it to recapture the same area again.
 - b. When working with DOS, record the specific commands you used. If you need to repeat a process, you can then copy/paste them.

- c. Don't forget to include notes on other supplementary programs you are using for this process, such as OSFMount for virtual disk mounts, and the settings you have them at.
4. Be thorough, but within reason. As you become more experienced with the QC process you may notice where some information is more important to document over others. In the future we may discuss how to refine this process based on your observations.
 - a. If you're installing using a series of disks, you don't have to screenshot every time it prompts you for another disk. You can screenshot the first prompt message, then summarize the others with just text.
5. Once you are finished, write a brief conclusion.
 - a. Summarize the installation process (1-2 sentences).
 - b. Was the installation successful? Were you able to open some files in the software?
 - c. Any final thoughts or conclusions about the software.
 - d. Copy this conclusion into the appropriate field in the Install Workflow excel file.
6. If you are performing multiple tests using different emulators, create a new word document per emulator. Save them as "[barcode]_InstallReport_[emulator].docx"

Emulators

DOSBox

DOSBox is useful for running programs that require DOS or Windows 3.1.

Documentation for using DOSBox can be found here: (Current as of 2020-3-24)

<https://www.dosbox.com/DOSBoxManual.html>

A copy of the environment that has been set up with graphics and sound can be found here: (Current as of 2020-3-24)

G:\AcqCatPres\Preservation&Conservation\Departments\Digital Preservation\Born Digital Curation and Preservation\SoftwarePreservation\EmulationDocuments\DOSBox Package

Eaasi

Documentation for using Eaasi can be found here: (Current as of 2020-3-24):

<https://www.softwarepreservationnetwork.org/eaasi/>

Local use and decisions documentation is found here:

File Name: 2020_02_UsingEaasiatUI.docx

Local Location: <G:\AcqCatPres\Preservation&Conservation\Departments\Digital Preservation\Born Digital Curation and Preservation\SoftwarePreservation>

Outcomes

After testing, determine if the software would suit the needs of the Software Preservation Project. Some simple questions to help with this:

- Were you able to successfully install the software?

- Is the software capable of reading files, serve as a functional OS environment, or support the reading of files (supplementary programs)?

Functional Software

Ensure that you have all documentation and files in their respective folders in the software's SIP. Prepare the software package for ingest.

Demo or Restricted Software

Look to see if a full version already exists in Medusa. If it does, flag the demo for reevaluation. If it does not, proceed with preparing the software for ingest. In the future, if the full version of something is discovered when a demo already exists in Medusa, flag the demo in Medusa for reevaluation.

NonExecutable Software

After testing, you may find some software is actually a file meant for another program or a presentation/video of a program and not something that can be interacted with.

- Personal files: delete the disk image. This is not relevant to the Software Preservation Project.
- Personal files with executable in the same disk image: flag the content for reevaluation.

Ingesting Software Preservation Collection Content into Medusa

SIP Ingest structure

This is the structure your ingest should be in:

- **Software Title**
 - o **Version**
 - **README file**
 - Info about software dependencies
 - Software metadata
 - Info about specific operating system requirements
 - **Operating System**
 - **InstallFiles**
 - o Disk Image(s) or install file(s)
 - **PDI**
 - o **Docs**
 - Copy of the deed of gift (if available)
 - Inventory transfer agreement (if available)
 - Purchase information (if available). Can include email invoice or confirmation. Redact sensitive information.
 - PDFs of manuals (if available)
 - Supplemental information related to software operation (if available)
 - o **Manifest**
 - Hash

- Directory listing
- **ObjectJPG**
 - Photographs of physical carrier (if available)
- **QC**
 - QCworkflow.xlsx
 - InstallReport.docx

Collection Location in Medusa

- **Repository:** Preservation Services
- **Collection:** <https://medusa.library.illinois.edu/collections/1508>
 - Software Preservation collection (Born Digital)
- **File Group Structure:**
 - “Content” (bit-level store)
 - “[Month] [Year] Accrual” (For tracking accruals, like a regular ingest job)
- **Structure within bit-level store file group:**
 - The top-most directory is the software title or some other unique identifier
 - The second next child directory is the software version number, or the operating system
- **Medusa staging area:**
 - Use a collecting unit’s MedusaStaging for a unit which does not frequently ingest content into Medusa such as SousaArchives.

Ingesting the SIP into Medusa:

Accruing to an Existing Structure

Since the software is all going into one collection, every ingest will be an accrual. If the folders in the ingest are arranged and name the same as what is currently in Medusa, Medusa will merge them together on ingest. UA submission packages are structured like this. This can be useful, if applied correctly.

The submission package for a ProTools version 6.0 for Windows would look like this:

- ProTools
 - 6_0
 - README file
 - Windows
 - InstallFiles
 - PDI
 - Docs
 - Manifest
 - ObjectJPG

Assuming the Example Basic Layout from above was already in Medusa, you would direct the ingest to occur at the top folder level. Medusa would see that there was already a ProTools folder and a 6_0

subfolder so it would then merge them with the old ones. However, since there was no Windows subfolder under 6_0 it would add it in. The final will look like this:

- ProTools
 - 6_0
 - README file
 - Mac
 - InstallFiles
 - PDI
 - Docs
 - Manifest
 - ObjectJPG
 - Windows
 - InstallFiles
 - PDI
 - Docs
 - Manifest
 - ObjectJPG
 - 7_3
 - README file
 - Mac
 - InstallFiles
 - PDI
 - Docs
 - Manifest
 - ObjectJPG
 - Windows
 - InstallFiles
 - PDI
 - Docs
 - Manifest
 - ObjectJPG

While it may seem redundant to include the upper folders in the ingest that will be merged, it makes the ingesting procedure a lot easier. For example, say there was only the Windows folder for the 7.3 version to add. You could navigate down to the 7_3 folder in Medusa and direct it to ingest from there instead of from the topmost folder like the above example. This requires more careful thought and attention with each ingest, and it is easier if things are organized so that you only have to navigate to the first level of the content bit level storage group in Medusa and direct the ingest to go from there.

However, if the upper ProTools folder was omitted in the submission, it would look like this:

- 6_0
 - README file
 - Windows
 - InstallFiles
 - PDI
 - Docs
 - Manifest

- ObjectJPG

When directed to ingest from the topmost folder, this would be the result:

- ProTools
 - o 6_0
 - README file
 - Mac
 - InstallFiles
 - PDI
 - o Docs
 - o Manifest
 - o ObjectJPG
 - o 7_3
 - README file
 - Mac
 - InstallFiles
 - PDI
 - o Docs
 - o Manifest
 - o ObjectJPG
 - Windows
 - InstallFiles
 - PDI
 - o Docs
 - o Manifest
 - o ObjectJPG
- 6_0
 - o README file
 - o Windows
 - InstallFiles
 - PDI
 - o Docs
 - o Manifest
 - o ObjectJPG

THE MOST IMPORTANT POINT TO REMEMBER IS THAT THE ACCRUAL SUBMISSIONS INTO MEDUSA MUST MIRROR THE SAME FOLDER STRUCTURE THAT IS ALREADY IN MEDUSA.

Overwriting Duplicate Content

When Medusa gets an ingest request, it scans the materials submitted and compares it to what is already in the collection. If it finds a duplicate md5 hash, an email report will be sent with more details on the duplicates found and the ingest job will automatically be cancelled.

Inspect the ingest and compare the items flagged as duplicates. If **all of the items** are the same, you have two options:

- 1) Manually delete the duplicate items set to be ingested and resubmit the ingest request

2) Resubmit the ingest request, this time checking the “Allow overwrite” button.

Add files

Up Toggle Directories Toggle Files Ingest Cancel

Allow overwrite

Roots:

AVPres

Medusa will again perform its md5 check, flagging the same duplicates but this time leaving the job on the dashboard with a red highlight (shown below):

Pre-ingest report	Status	Actions
	Awaiting approval	
	Awaiting approval	

Filter results:

First < 1 > Last

Double check that the files being overridden really are alright to be overridden before proceeding. **OVERRIDING MATERIALS IS A VERY, VERY RARE NECESSITY.**

Document Version History

Date	Version	Author	Notes
2019-12-13	0.1	Willson, Christine; Popp, Tracy	Initial draft

2020-2-25	0.2	Willson, Christine	Christine's notes incorporated to top, 'Software Acquisition' and 'Acquiring and Imaging Software for Preservation' merged
2020-04-26	1.0	Willson, Christine; Popp, Tracy	Christine's notes and other notes at end of document were deleted. Addition of a table of contents. Document edited for submission to the FCoP