

Hands-on Python PDFs: Using the pypdf Library To Programmatically Design, Complete, Read, and Extract Data from PDF Forms Having Digital Signatures

Troy Martin Hughes, WUSS 2024 Academic Chair

ABSTRACT

The pypdf Python library (<https://pypdf.readthedocs.io/en/stable/index.html>) facilitates the programmatic creation, completion, cropping, and merging of PDF forms. Form data—including both dynamic text and field values—can be programmatically written to a PDF using pypdf, and data manually entered into PDF form fields by users can be programmatically extracted. With this combination of functionality, pypdf is a powerful tool that can build dynamically generated PDF forms that simplify user completion of forms, as well as subsequent form validation. This text introduces users to the pypdf library, and demonstrates a single use case in which copyright grant forms (CGFs, aka copyright-release or permission-to-publish forms) were automatically generated for the Western Users of SAS Software (WUSS) 2024 conference proceedings. This automation eliminated confusing language and components of the CGF—for example, by removing language specific only to US government employees (unless the author was a US government employee). Thus, in 2023, an author submitting a paper to WUSS had to navigate more than 50 unutilized fields in the CGF!! Moreover, an author completing the WUSS 2023 CGF had to enter his name, paper number, paper title, job title, and organization—information that the conference already had collected and which it should have been using to prepopulate forms for authors. Thus, the revised form and process now only requires each author to digitally sign the CGF; errors are eliminated and efficiency is maximized. This solution was developed and run on Python 3.11, and the full code is included in Appendix A.

INTRODUCTION

This text illustrates how the pypdf Python library automates aspects of PDF form creation, completion, and evaluation. *Form creation* can include the merging, cropping, and dynamic population of text into a PDF. *Form completion* can programmatically enter text into PDF form fields, which can forego the need for users to enter those same data manually. In some cases, the fields that are programmatically entered are locked thereafter so users cannot accidentally modify the form. Finally, *form evaluation* describes the programmatic extraction of values from fields, and enables Python to utilize user-supplied data from PDFs to populate Pandas DataFrames, spreadsheets, CSV files, and other data structures and files.

The pypdf solution that was implemented could have been developed using other Python libraries or other languages. At a high level, the solution is as much one of process improvement as it is of technological improvement. Outmoded processes and lack of vision had unfortunately stymied how CGFs were designed and how their data were collected by WUSS. Thus, some overarching best practices and principles that drove this project can be inferred, and these are described briefly.

Master Data Management (MDM) prescribes data governance, stewardship, and management principles, and one objective is to reduce—if not eliminate—data redundancy. For example, MDM prescribes that if you capture a customer's full name in one table, you shouldn't maintain separate versions of the customer's name in other tables. Thus, maintaining a single record (aka, the "golden record") militates against unwanted variation in data. Past CGF forms and their data collection violated MDM principles because authors were required to enter data (including name, job title, organization, presentation number, and presentation title) that the conference already had collected, which thus create duplicate information.

Automation prescribes that processes that can be defined, and which can execute predictably given predictable inputs, should be performed by software, not people. This working-smarter-not-harder mentality improves not only resource management, as burdensome rote is shifted to software, but also resource attitude, as drudgery is eliminated from those doing the work. Although form completion is commonly viewed as painful, it is especially agonizing when unnecessary. Thus, in automating the completion of nearly all fields in the CGF, the burden of data entry was removed from conference presenters and the burden of data validation was moreover removed from conference organizers.

THE DREADED COPYRIGHT GRANT FORM (CGF)

Copyright grant forms (CGFs) have been the bane of my existence for the past decade. Software conferences typically require each presenter to "grant" publication rights to the conference so the presentation slides and/or white paper can be published in the conference proceedings. These non-exclusive copyrights do not prohibit the author from publishing the work at subsequent conferences—or in blogs, papers, books, or other written products—but are required to demonstrate that the conference has each author's permission. And among SAS regional conferences like WUSS, SESUG, SCSUG, and MWSUG, CGFs have amounted to heinous, unnecessary work that burdens both presenters and the volunteer cadre of Section Chairs who must validate that CGFs have been completed correctly. Spoiler ahead—they rarely have been!

In 2014, at my first-ever WUSS conference, the CGF was a MS Word document that had to be printed, completed by each author (often handwritten, but occasionally typed), signed, scanned, and emailed to the conference. That is, the conference prepopulated no data into the form, and the author had to enter his name, paper number, and paper title. Even a mail merge in 1990s MS Word or Word Perfect could have prepopulated a separate form for each author, but no SAS conference invested the effort to automate this process. Figure 1 demonstrates an excerpt of my first-ever CGF for WUSS 2014—and even then, my titles did not fit in the allotted space!

Copyright Grant

Publisher: Western Users of SAS Software

Publication: Western Users of SAS Software Proceedings (WUSS 2014)

Work (Paper Title exactly as it appears on the final paper to be published):

*When Software Development is Your Means Not Your End:
Abstracting Spikes Methodologies for End User Development and
Analytic Application*

Author(s):
Greg Martin Hughes

1

Figure 1. WUSS 2014 Copyright Grant Form, Page 1 Excerpt

In 2015, WUSS had switched to using PDFs for its CGFs, but these were no more "forms" than the prior MS Word documents. For example, although the CGF had two lines in which to enter the presentation title, the text had to be entered separately (effectively into two "fields") if the title extended past the first line. And because these forms all required a wet signature, authors almost exclusively handwrote the entire form as opposed to typing it. This CGF modality continued until 2018, when WUSS regressed and provided only MS Word CGFs for presenters to complete. In 2019, WUSS CGFs returned to PDF format—a slight improvement, but nevertheless a demonstration of the lack of continuity across the years.

After taking a two-year hiatus during the COVID-19 pandemic, WUSS returned to an in-person conference in 2022 and ushered in actual PDF *forms*, which it retained in 2023. These forms featured actual fields into which data could be typed, checkboxes that could be clicked, and a digital signature block for all presenters. Finally, the CGF could be completed electronically, digitally signed, and returned via email without the need to print and scan a paper form! And with this advancement, for the first time in conference history, the majority of WUSS CGFs were typed as opposed to handwritten! Figure 2 demonstrates an excerpt from my 2023 CGF for the award-winning *Make You Holla' Tikka Masala*.

Publisher: WUSS
Publication: WUSS 2023 Conference Proceedings (online at www.WUSS.org)

Work Paper Number (shown as ###):

167

Work Paper Title:

(Must match title of
submitted paper)

Make You Holla' Tikka Masala: Creating User-Defined
Informats Using the PROC FORMAT OTHER Option
To Call User-Defined FCMP Functions That Facilitate

Author(s):

(Must match authors of
submitted paper)

Troy Martin Hughes

Copyright Owner:

N/A

If your company is the owner, put your company's name above and have the company's representative sign in the corresponding spot in Section A. For a paper with multiple authors, it is the responsibility of the first author to collect the signature(s) of co-author(s) and submit the copyright grant form. Only the representative of the first author's employer need to sign in the second page and it is assumed that the agreement is reached by the employer(s) of the co-author(s). In the case, there is no copyright owner declared by the author (or the first author of multiple-author paper), marking N/A in Copyright Owner row, i.e., the copyright belongs to WUSS, the publisher.

Page 1 of 4

Figure 2. WUSS 2023 Copyright Grant Form, Page 1 Excerpt

Despite the inarguable CGF advances in 2022, these 2022/2023 PDFs still offered several areas for improvement. For example, all presenters were still required to manually enter their presentation number, presentation title, full name, job title, and organization—information that WUSS already possessed and with which it could have prepopulated the form. Form data integrity also remained abysmal, and was actually worsened due to the complexity of these PDFs, with nearly half of 2023 CGFs completed incorrectly.

Much confusion resulted from convoluted business rules that prescribe who must sign the form. For example, Figure 3 illustrates the signature portion that each presenter must complete. However, note that presenters whose employer requires authorized release must complete both the left and right blocks, whereas presenters whose employer do not require release must instead complete the left block and check the box to the right.

Name of Author (please type or print)	Authorized Representative (please type or print)
Signature	Signature
Job Title	Job Title
Company Name	Date
Date	☐ If an authorized representative of the author's employer is not required to sign, then the author endorses that no further permissions are required within their company.

Figure 3. WUSS 2023 Copyright Grant Form, Page 2 Excerpt

Another obstacle in the WUSS 2023 CGF was the section only applicable to US government employees, as depicted in Figure 4. In 2023, some presenters accidentally signed and dated this portion, which then required a revised form to be completed.

WUSS 2023 Proceedings Copyright Grant Form

Section B:

The undersigned certifies that he/she prepared the Work as an employee of the U.S. government and prepared the Work as a part of his or her employment and that the Work is therefore not subject to U.S. copyright protection.

Signature	Date
----------------------	-----------------

Figure 4. WUSS 2023 Copyright Grant Form, Page 4 Excerpt

Magnifying these complexities was the reality that many presentations are coauthored, and thus require two (or more) presenters to complete and sign the same CGF. Figure 5 demonstrates the sections for first author and second author on the WUSS 2023 CGF.

Name of Author (please type or print) Signature Job Title Company Name Date	Authorized Representative (please type or print) Signature Job Title Date ☐ If an authorized representative of the author's employer is not required to sign, then the author endorses that no further permissions are required within their company.
Name of Author (please type or print) Signature Job Title Company Name Date	Authorized Representative (please type or print) Signature Job Title Date ☐ If an authorized representative of the author's employer is not required to sign, then the author endorses that no further permissions are required within their company.

Figure 5. WUSS 2023 CGF First and Second Author Blocks

With presenters failing to correctly complete nearly half of the CGFs in 2023, this required significant review by Section Chairs as well as rework by presenters, all of which could have been avoided. One stellar WUSS 2023 Section Chair, who vowed never again to endure the ignominies of manually completed CGFs, was voted Academic Chair for 2024, and finally overhauled the insufferable forms and processes.

SOFTCONF START/V2 SOFTWARE

The journey toward successful, intelligent CGFs began with an overhaul of SoftConf START/V2, the SaaS software leveraged by WUSS and other SAS conferences to maintain presenter and presentation data, including biographical information, white papers, slide presentations, and completed CGFs. MDM principles prescribe a single version of critical data should be maintained, and SoftConf was designated (for 2024) as the database of record for all data used to populate CGFs. This shift required the creation of a new "Update User Profile" portal in which WUSS presenters (including both first authors and coauthors) would be required to enter presenter-specific data, as demonstrated in Figure 6.

Copyright Grant Form (CGF)

Each accepted proposal must have an accompanying copyright grant form (CGF), a PDF form that is digitally signed by the author (and all coauthors, if applicable). The purpose of the CGF is to grant WUSS (the Publisher) the "nonexclusive right" to publish your paper or presentation. Note that this right in no way infringes on your right as an author to publish or republish your work, to include at other conferences, in blogs, vlogs, webinars, textbooks, or any other medium.

No paper or presentation will be included in the conference proceedings until the CGF is completed and uploaded into SoftConf, and delay in CGF filing may result your submission being removed from the conference. Please complete the following questions, which are required to automatically generate the specific CGF for your proposal(s). If one or more of your submissions is accepted, WUSS will communicate further instructions on how to download, complete, and upload your CGF(s).

Some companies and organizations require employees to obtain the signature of a manager (or other authorizing release agent) before publishing a paper or slide presentation in the conference proceedings. Please indicate whether your paper or presentation requires a signed release, and if so, an additional signature block will be added to your CGF so an agent can authorize release of your document(s) to WUSS. In these cases, the company or organization you enter will also become the Copyright Owner of the paper or presentation you publish, whereas WUSS retains rights as Publisher. *Please note that your response in no way affects your eligibility for or likelihood of proposal acceptance.*

Employer Release (*)

Papers or presentations prepared by US government employees in the furtherance of their official duties are not copyrightable in the United States (17 U.S.C. 105), and are therefore in the public domain. For this reason, WUSS requires all federal employees to self-identify to ensure the appropriate CGF is utilized. *Please note that your response in no way affects your eligibility for or likelihood of proposal acceptance.*

Federal Employee (*)

SAS Institute employees require a separate CGF for their papers and presentations, so WUSS requires all SAS employees to self-identify to ensure the appropriate CGF is utilized. *Please note that your response in no way affects your eligibility for or likelihood of proposal acceptance.*

SAS Employee (*)

The name of your company or organization is required on the CGF. If you are a part-time or full-time student, please indicate the name of your school. If you are unemployed, retired, or otherwise not working, please indicate "N/A" in the required field.

Company/Organization (*)

Your job title is required on the CGF. If you are a part-time or full-time student, please indicate "student." If you are unemployed, retired, or otherwise not working, please indicate "N/A" in the required field.

Job Title (*)

[Submit Data](#)

START Conference Manager (V2.61.0 - Rev. 7122)

Figure 6. SoftConf "Update User Profile" Portal Added To Support WUSS 2024

With these new fields, and coupled with existing presentation-specific data maintained in SoftConf (including presentation number and title), the groundwork was laid for CGFs to be dynamically created and prepopulated.

BUSINESS RULES

Business rules for the creation and completion of CGFs were derived from past CGF usage; that is, the rules were not changed. However, rather than these rules *describing* how presenters would need to complete CGFs, as had occurred in all past years, the rules were codified in Python to *prescribe* how CGFs would be dynamically built and populated for each presentation.

Note that one of the major advantages of dynamically created forms is the ability to remove form sections that are irrelevant for a specific presentation or user. For example, if, on average, 99 percent of presenters are not federal employees, why was WUSS subjecting all presenters to an entire page of the CGF applicable only to federal employees? Similarly, as most presentations are submitted and delivered by a single author, why was WUSS dedicating multiple pages of the CGF for irrelevant coauthor information, which only served to confuse presenters?! Thus, business rules prescribe not only how form data can be prepopulated but also how the form components are built and which are included.

Business rules prescribing form creation and content blocks include:

- The number of author blocks now matches the number of presentation authors. *The WUSS 2023 CGF had blocks for five authors, irrespective of the actual number of authors.*
- The "Authorized Representative" block now is only shown when a presenter has indicated (in SoftConf) that organizational release of the work is required. *This eliminates the need for the "Authorized Representative not needed" checkbox, and removes this signature block for presenters who do not require organizational release.*
- SAS employees must be listed as an author, but their author block no longer contains a digital signature field. *Per WUSS policy and an agreement with SAS Institute, SAS employees have not historically signed CGFs, in part because SAS Institute—not WUSS—maintains the copyright of all work products authored solely by SAS employees.*

Business rules prescribing the prepopulation of form data include:

- Where an author is a US government employee, the following text is dynamically written above the author's block: "The author certifies that he/she prepared the Work as an employee of the U.S. government and prepared the Work as a part of his/her employment and that the Work is therefore not subject to U.S. copyright protection."
- Where an author is a SAS Institute employee, the following text is dynamically written above the author's block: "The author acknowledges that he/she is a SAS Institute employee."
- Where an author's organization requires review and release, the following text is dynamically written above the author's block: "By signing, the author acknowledges that an authorized representative of his/her company or organization is required to sign to publish the Work."
- Where an author's organization does not require release, the following text is dynamically written above the author's block: "By signing, the author acknowledges that no further permissions are required by the author's company or organization to publish the Work."
- For presentations in which any single author is a US government employee, the copyright owner is shown as "public domain."
- For presentations in which all authors are SAS Institute employees, the copyright owner is shown as "SAS Institute."
- In all other cases, the copyright owner is shown as "Western Users of SAS Software."
- In all cases, the publisher is shown as "Western Users of SAS Software."

DYNAMICALLY POPULATING FORM DATA USING PYPDF

At the heart of this solution is the pypdf functionality that facilitates aggregating and concatenating pieces of various PDFs into a final, composite PDF. For example, the first page of the CGF contains all presentation-specific data (as opposed to author-specific data), and includes the presentation number, presentation title, and list of authors, as well as text that explains the form and its deadline for completion. Figure 7 shows an excerpt from the one-page PDF "template" or "skeleton"—designed in MS Word or Adobe Acrobat, and saved as a PDF—that forms the foundation of the final CGF that is dynamically created.

WUSS 2024 Copyright Grant Form (CGF)

Dear Author(s),

Congratulations once again for being accepted into the WUSS 2024 conference proceedings, held at the Hyatt Regency Sacramento, September 4 to 6, 2024. We look forward to hosting your presentation in person!

This Copyright Grant Form (CGF) applies **only** to the following paper/presentation, hereafter known as **"the Work"**:

Presentation ID:

Title:

Author(s):

Note that the preceding title must match the Work title that is submitted to WUSS for publication, and that the preceding author name(s) (and order thereof) must match the author name(s) that appear in the Work. If you detect a discrepancy in the title or in the author(s), please immediately notify your Section Chair, who will modify these data in the WUSS Submission System, and provide an updated CGF to complete.

Your presentation will not be scheduled until and unless this CGF is completed, and any delay in submission of this PDF could result in your presentation being removed from the conference proceedings. Only one CGF should be submitted per presentation, so all author(s) must digitally sign the same form, after which the completed CGF must be uploaded to the WUSS Submission System (<https://softconf.com/n/wuss2024>) no later than

Figure 7. Template for First Page of WUSS 2024 CGF (Presentation Number, Title, and Authors)

This one-page template additionally depicts the publisher and copyright owner, as Shown in Figure 8, which are dynamically populated using pypdf.

Publisher and Copyright Owner

The **Publisher** of the Work is:

The **Copyright Owner** of the Work is:

Figure 8. Template for First Page of WUSS 2024 CGF (Publisher and Copyright Owner)

For each presentation, the following code populates the fields in the first page of the CGF, using presentation-level data maintained in a Pandas DataFrame (i.e., `presentation_id`, `title`, `authors`), data defined in the code itself (i.e., `date_deadline`, `publisher`), and data computed by applying the preceding business rules to the SoftConf author table (i.e., `copyright_owner`):

```
# assign values on first page
```

```

presentation_id = row.sub_id
title = row.title
authors = ', '.join(row.fullname_list)
date_deadline = r'August 16, 2024.'
publisher = r'Western Users of SAS Software'
if row.any_govt == True:
    copyright_owner = 'public domain'
elif row.all_sas == True:
    copyright_owner = 'SAS Institute'
else:
    copyright_owner = r'Western Users of SAS Software'

```

Subsequently the `update_page_form_field_values` pypdf method is used to assign (or bind) these values to the first page of the PDF, as indicated by the `pages[0]` designation:

```

pdf_writer.update_page_form_field_values(
    pdf_writer.pages[0],
    {"presentation_id": presentation_id,
     "title": title,
     "authors": authors,
     "date_deadline": date_deadline,
     "publisher": publisher,
     "copyright_owner": copyright_owner},
    auto_regenerate=False,)

```

Figure 9 demonstrates how the dynamically created first page of the CGF appears for a presentation.



WUSS 2024 Copyright Grant Form (CGF)

Dear Author(s),

Congratulations once again for being accepted into the WUSS 2024 conference proceedings, held at the Hyatt Regency Sacramento, September 4 to 6, 2024. We look forward to hosting your presentation in person!

This Copyright Grant Form (CGF) applies **only** to the following paper/presentation, hereafter known as "the Work":

Presentation ID: 109

Title: Performing with Python Pandas: Taming Your Inner Snake

Author(s): Troy Martin Hughes, Pigsly McMurphy

Note that the preceding title must match the Work title that is submitted to WUSS for publication, and that the preceding author name(s) (and order thereof) must match the author name(s) that appear in the Work. If you detect a discrepancy in the title or in the author(s), please immediately notify your Section Chair, who will modify these data in the WUSS Submission System, and provide an updated CGF to complete.

Your presentation will not be scheduled until and unless this CGF is completed, and any delay in submission of this PDF could result in your presentation being removed from the conference proceedings. Only one CGF should be submitted per presentation, so all author(s) must digitally sign the same form, after which the completed CGF must be uploaded to the WUSS Submission System (<https://softconf.com/n/wuss2024>) no later than **August 16, 2024**.

Figure 9. Excerpt of First Page of WUSS 2024 Dynamically Created CGF

Note that style elements such as font family, style, size, color, and other attributes can be varied, and this is done by formatting these values in the PDF template used to build the final PDF. For example, highlighting the deadline for CGF upload appears in **bold red**, but this is accomplished in Adobe Acrobat, not programmatically in Python.

In this CGF example, the first page of the CGF is always built from the same PDF template and its structure always identical for all presentations; the only dynamic elements are the preceding data that are prepopulated. Conversely, all subsequent pages of the CGF are presenter-specific; for example, the second page applies only to the first author, and if applicable, the third page applies only to the second author.

Thus, the following code demonstrates how each author's author-specific data are dynamically bound to a second or subsequent page of the CGF:

```
pdf_writer.update_page_form_field_values(
    pdf_writer.pages[pg],
    {p + '.author_number': ' '.join([author_number_dict[pg], 'Author']),
    p + '.author_text': author_text,
    p + '.author': row['fullname_list'][pg-1],
    p + '.job_title': row['job_title_list'][pg-1],
    p + '.author_company_org': row['company_org_list'][pg-1],
    p + '.company_number': ' '.join([author_number_dict[pg],
    "Author's Authorized Representative"]),
    p + '.company_text': company_text,
    p + '.rep_company_org': row['company_org_list'][pg-1]},
    auto_regenerate=False,)
```

For example, the second page of a CGF that does not require organizational release is demonstrated in Figure 10, with blue text populated dynamically by pypdf.

WUSS 2024 Copyright Grant Form (CGF)

First Author

By signing, the author acknowledges that no further permissions are required by the author's company or organization to publish the Work.

Author Name: Troy Martin Hughes
Job Title: Director of Camelid Satisfaction
Company/Organization: Data Llama Analytics, LLC



Figure 10. Dynamically Populating Author-Specific Data in the CGF

Note, however, that some of the information that is dynamically populated is less obvious than the blue text. For example, "First Author" is dynamically written based on the author number. And the "By signing..." text, as previously described in the "Business Rules" section, is also built dynamically from business rules. Thus, quite intricate and customized forms can be designed solely by dynamically writing text. However, for even greater form control, PDF components—not only text—can be altered or aggregated dynamically, as demonstrated in the next sections.

DYNAMICALLY BUILDING FORM COMPONENTS USING PYPDF

As described in the previous business rules, most authors are required to digitally sign the CGF; however, SAS Institute employees are not required to sign the CGF, although their information still must appear on the CGF. The pypdf library offers methods to crop a PDF page to hide components; however, these methods fail in some cases, and digital signatures are one component that cannot be hidden. To overcome

this limitation, as well as to demonstrate how PDF components can be dynamically concatenated into a final form, consider the following simplified business rules:

- The author block of a SAS employee should not contain a signature block.
- All other author blocks must contain a signature block for the author.
- When organizational release is required, a second signature block is required.

That is, for a SAS author, the software must select a one-page PDF template that does not include a signature block (WUSS_2024_CGF_p2_0sig_BLANK.pdf), as shown in Figure 11.

WUSS 2024 Copyright Grant Form (CGF)

Author Name:
Job Title:
Company/Organization:

Figure 11. PDF Template Used for SAS Employees (No Signature Block)

And for all other authors, a one-page PDF template is used that *does* include a signature block (WUSS_2024_CGF_p2_1sig_BLANK.pdf), as shown in Figure 12.

WUSS 2024 Copyright Grant Form (CGF)

Author Name:
Job Title:
Company/Organization:

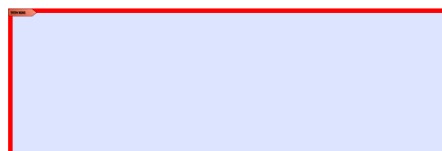


Figure 12. PDF Template Used for Non-SAS Authors (With Signature Block)

Although not demonstrated, a second signature block is required when an organization must review and release the white paper and/or slide presentation. Thus, the second (and subsequent) pages of the CGF can be built from different PDF templates, as the parenthesized PDF filenames demonstrates.

The following code reads and initializes all templates so they can be appended to the final CGF that is being dynamically built:

```
loc_base = r'C:\cosas\python\SAS_conference_copyright_form'  
loc_forms = r'blank_forms'
```

```

completed_forms = r'completed_forms'

# first page always has the same components
blank_form_p1 = r'WUSS_2024_CGF_p1_BLANK.pdf'
pdf_obj = open(os.path.join(loc_base, loc_forms, blank_form_p1), 'rb')
pdf_reader_p1 = PdfReader(pdf_obj)
# NO signatures required for SAS employees
blank_form_p2_0sig = r'WUSS_2024_CGF_p2_0sig_BLANK.pdf'
pdf_obj = open(os.path.join(loc_base, loc_forms, blank_form_p2_0sig), 'rb')
pdf_reader_p2_0sig = PdfReader(pdf_obj)
pdf_reader_p2_0sig.add_form_topname('temp')
# 1 signature required when organization not required to release
blank_form_p2_1sig = r'WUSS_2024_CGF_p2_1sig_BLANK.pdf'
pdf_obj = open(os.path.join(loc_base, loc_forms, blank_form_p2_1sig), 'rb')
pdf_reader_p2_1sig = PdfReader(pdf_obj)
pdf_reader_p2_1sig.add_form_topname('temp')
# 2 signatures required when organization is required to release
blank_form_p2_2sig = r'WUSS_2024_CGF_p2_2sig_BLANK.pdf'
pdf_obj = open(os.path.join(loc_base, loc_forms, blank_form_p2_2sig), 'rb')
pdf_reader_p2_2sig = PdfReader(pdf_obj)
pdf_reader_p2_2sig.add_form_topname('temp')

```

Thereafter, data maintained in SoftConf are used to dynamically select the correct PDF template to be used for each author and, if applicable, subsequent coauthor(s). Thus, for a CGF representing a three-author presentation, the first author might be a SAS employee who requires a PDF template having no signature blocks; the second author might be a non-SAS employee who requires no organizational release (and thus only one signature block); and the third author might be a non-SAS employee who requires organizational release (and thus two signature blocks). In this example, the first page of the CGF is built from the first-page template, which is used for all CGFs, and each of the subsequent three pages of the CGF is built from a separate PDF template. In total, four separate templates would be used to build the three-author, four-page CGF!

When an author is a SAS employee (indicated by a value that is not "NO"), the following code selects the correct PDF template, adds it to the CGF being built using the pypdf **append** method, and binds the values dynamically using the **update_page_form_field_values** pypdf method:

```

if row.sas_employee_list[pg-1] != 'NO':
    pdf_reader_p2_0sig.rename_form_topname(p)
    pdf_writer.append(pdf_reader_p2_0sig)
    pdf_writer.update_page_form_field_values(
        pdf_writer.pages[pg],
        {p + '.author_number': ' '.join([author_number_dict[pg], 'Author']),
         p + '.author_text': author_text,
         p + '.author': row['fullname_list'][pg-1],
         p + '.job_title': row['job_title_list'][pg-1],
         p + '.author_company_org': row['company_org_list'][pg-1]},
        auto_regenerate=False,)
    # reset required in case the same blank PDF has already been imported/cloned
    pdf_writer.reset_translation(pdf_reader_p2_0sig)

```

Note that the **reset_translation** method is required after the template is added and values bound, in the event that a second SAS author (a coauthor) exists for the same presentation, which requires the same zero-signature PDF template to be appended a second time in building the CGF.

TESTING AND DEMONSTRATING THE PROCESS

Given the number of business rules (enumerated in the preceding "Business Rules" section) and the interaction thereof, myriad versions of final CGFs can be dynamically created. A spreadsheet containing a representation of SoftConf data (including both presentation-specific and author-specific data) is shown in Table 1, and provided the metadata substrate with which this software solution was tested.

1	description	A	B	C	D	E	F	G	H	I	J	K
2	one author - 1455946 bio	101 Five Reasons to Sweep Right on PROOC FCHP: the SAS Fundco	data@vendude@gmail.com	Troy Martin Hughes	data@vendude@gmail.com	NO	NO	NO	NO	NO	NO	NO
3	two authors - 1455946 bio	102 Performing with Python Pandas: Taming Your Inner Snake	data@vendude@gmail.com	Troy Martin Hughes	data@vendude@gmail.com	NO	NO	NO	NO	NO	NO	NO
4	two authors - 1455946 bio	103 Five Reasons to Sweep Right on PROOC FCHP: the SAS Fundco	data@vendude@gmail.com	Troy Martin Hughes	data@vendude@gmail.com	NO	NO	NO	NO	NO	NO	NO
5	one author - no employer release	104 Five Reasons to Sweep Right on PROOC FCHP: the SAS Fundco	data@vendude@gmail.com	Troy Martin Hughes	data@vendude@gmail.com	NO	NO	NO	NO	NO	NO	NO
6	one author - employer release	105 Five Reasons to Sweep Right on PROOC FCHP: the SAS Fundco	data@vendude@gmail.com	Troy Martin Hughes	data@vendude@gmail.com	YES, my employer must authorize this	NO	NO	NO	NO	NO	NO
7	one author - US govt	106 Five Reasons to Sweep Right on PROOC FCHP: the SAS Fundco	data@vendude@gmail.com	Troy Martin Hughes	data@vendude@gmail.com	YES, I prepared this	NO	NO	NO	NO	NO	NO
8	one author - SAS employee	107 Five Reasons to Sweep Right on PROOC FCHP: the SAS Fundco	data@vendude@gmail.com	Troy Martin Hughes	data@vendude@gmail.com	YES, I am an employee of SAS Institute	NO	NO	NO	NO	NO	NO
9	two authors - 1 no release 2 no release	108 Performing with Python Pandas: Taming Your Inner Snake	data@vendude@gmail.com	Troy Martin Hughes	data@vendude@gmail.com	NO	NO	NO	NO	NO	NO	NO
10	two authors - 1 no release 2 release	109 Performing with Python Pandas: Taming Your Inner Snake	data@vendude@gmail.com	Troy Martin Hughes	data@vendude@gmail.com	YES, my employer must authorize this	NO	NO	NO	NO	NO	NO
11	two authors - 1 US govt 2 US govt	110 Performing with Python Pandas: Taming Your Inner Snake	data@vendude@gmail.com	Troy Martin Hughes	data@vendude@gmail.com	YES, I prepared this	NO	NO	NO	NO	NO	NO
12	two authors - 1 US govt 2 no release	111 Performing with Python Pandas: Taming Your Inner Snake	data@vendude@gmail.com	Troy Martin Hughes	data@vendude@gmail.com	YES, I prepared this	NO	NO	NO	NO	NO	NO
13	two authors - 1 US govt 2 release	112 Performing with Python Pandas: Taming Your Inner Snake	data@vendude@gmail.com	Troy Martin Hughes	data@vendude@gmail.com	YES, I prepared this	NO	NO	NO	NO	NO	NO
14	two authors - 1 US govt 2 SAS employee	113 Performing with Python Pandas: Taming Your Inner Snake	data@vendude@gmail.com	Troy Martin Hughes	data@vendude@gmail.com	YES, I prepared this	NO	NO	NO	NO	NO	NO
15	two authors - 1 SAS employee 2 SAS employee	114 Performing with Python Pandas: Taming Your Inner Snake	data@vendude@gmail.com	Troy Martin Hughes	data@vendude@gmail.com	YES, I am an employee of SAS Institute	NO	NO	NO	NO	NO	NO
16	two authors - 1 SAS employee 2 no release	115 Performing with Python Pandas: Taming Your Inner Snake	data@vendude@gmail.com	Troy Martin Hughes	data@vendude@gmail.com	YES, I am an employee of SAS Institute	NO	NO	NO	NO	NO	NO
17	two authors - 1 SAS employee 2 release	116 Performing with Python Pandas: Taming Your Inner Snake	data@vendude@gmail.com	Troy Martin Hughes	data@vendude@gmail.com	YES, my employer must authorize this	NO	NO	NO	NO	NO	NO

Table 1. Simulated SoftConf Submission and Author Data

The point of the table is that the complexity of these business rules and their interactions preclude a full demonstration of all possible CGF outcomes that could be imagined. Table 1, for example, only shows one- and two-author presentations, and omits the logarithmically infeasible display of three-, four-, and five-author presentations.

A first example demonstrates an author who requires organizational release, and which thus necessitates two signature blocks, as shown in Figure 13.

WUSS 2024 Copyright Grant Form (CGF)

First Author

By signing, the author acknowledges that an authorized representative of his/her company or organization is required to sign to publish the Work.

Author Name:
Troy Martin Hughes

Job Title:
SAS Ninja

Company/Organization:
Data Llama Analytics, LLC

First Author's Authorized Representative

By signing, the author's employer's representative authorizes publication of the Work.

Authorized Representative Name:

Job Title:

Company/Organization:
Data Llama Analytics, LLC

Figure 13. Second Page of CGF for a First Author Requiring Organizational Release

At this point, the presenter is able to download the customized CGF, having been provided a URL to the PDF from WUSS conference leadership. The first author only has to digitally sign the form, and is not required to enter any additional information. Thereafter, a representative from the organization must digitally sign the form, which is subsequently returned to WUSS. This revised process bypasses all the prior confusion that resulted when authors were required to enter information manually, or when irrelevant components of the PDF were unnecessarily conveyed to authors.

As a second example, consider a presentation for which the second author, Pigsly, is a SAS Institute employee (a Pork Data Scientist), in which case the author-specific information is printed but no signature block is required, as shown in Figure 14.

WUSS 2024 Copyright Grant Form (CGF)

Second Author

The author acknowledges that he/she is a SAS Institute employee.

Author Name: Pigsly McMurphy
Job Title: Pork Data Scientist
Company/Organization: SAS Institute

Figure 14. Third Page of CGF for a Second Author Who Is a SAS Employee

As a final example, the full text of a three-page, two-author CGF concludes this text.

NEXT STEPS

Although this process improvement represents several tremendous steps forward in how WUSS builds, disseminates, and handles CGFs, inasmuch as the programmatic underpinnings of the data processing and data management components that support these improvements, even more streamlined processes can be envisioned. For WUSS 2024, the author of a single-authored paper who does not require organizational release now only has to navigate to a customized URL, download the CGF, digitally sign and save the PDF, and upload the form to the SoftConf Submission System. No information must be entered!

Notwithstanding this progress, it would be folly to suggest this is the end of the road. For example, solutions have existed for years that allow users to digitally sign a document online without the need to download and save the file, or to upload or email the signed file. Thus, in years future, this author and conference chair hopes that even greater minds will build upon this solution to engineer even greater CGF functionality.

CONCLUSION

The inescapable benefits of automating the dynamic creation and completion of copyright grant forms (CGFs) has been shown, with customizable forms created for each of the 100+ presentations showcased at the WUSS 2024 conference. For the majority of presentations, the author is required only to digitally sign the form, with no other data input required! In other cases, only minimal data entry is required, such as by an authorized release agent of the author's organization. In all cases, automation and master data management (MDM) are embraced to deliver a solution that exhibits far greater functionality, configurability, and data integrity than past CGF solutions.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name: Troy Martin Hughes
E-mail: troymartinhughes@gmail.com

APPENDIX A. FULL CODE

```
# created and copyrighted by Troy Martin Hughes, 2024
from pypdf import PdfReader, PdfWriter
import os
from ast import literal_eval
import pandas as pd

''' SETUP

A folder (loc_base) is created: python_CGFs_HOW_code_data_forms

Within the folder, the following subfolders must be created:
    blank_CGF_templates
    completed_CGFs
    SoftConf_export
'''

loc_base = r'c:\python\cgf'
loc_blank = r'blank_CGF_templates'
loc_completed = r'completed_CGFs'
loc_softconf = r'SoftConf_export'

''' SPREADSHEET

The following spreadsheet is created (not shown) by joining two exported spreadsheets from
SoftConf:
    Submissions (downloaded as Submission_Information.xlsx) - one record per presentation
    Authors (downloaded as Author_Information_20240823_1400.xlsx) - author and coauthor data

At a minimum, the joined spreadsheet must contain the following fields:
    sub_id - numeric field depicting presentation ID, an incremented integer
    title - presentation title
    email_list - comma-delimited list of author and coauthor emails;
                not integral to CGF creation, but if a presenter is missing one or more
                biographical fields, the author's email is printed so he can be contacted
    fullname_list - comma-delimited list of author and coauthor names,
                used as the names to print on the CGFs
    copyright_employwer_list - comma-delimited list of author and coauthor release
requirements;
                (NO - indicates employer release and signature not required)
    copyright_us_govt_list - comma-delimited list of author and coauthor US govt affiliation;
                (NO - indicates author is not US govt employee, so copyright is not in public domain)
    sas_employee_list - comma-delimited list of author and coauthor SAS employment;
                (NO - indicates author is not a SAS Institute employee, so digital signature is
required)
    company_org_list - comma-delimited list of author and coauthor employer
    job_title_list - comma-delimited list of author and coauthor job titles
'''
```

```

presentations_xlsx = r'TEST_presentations_and_authors.xlsx'

df_subs = pd.read_excel(os.path.join(loc_base, loc_softconf, presentations_xlsx),
                        keep_default_na=False,
                        usecols=['sub_id',
                                'title',
                                'email_list',
                                'fullname_list',
                                'copyright_employer_list',
                                'copyright_us_govt_list',
                                'sas_employee_list',
                                'company_org_list',
                                'job_title_list'])
df_subs.email_list = df_subs.email_list.apply(literal_eval)
df_subs.fullname_list = df_subs.fullname_list.apply(literal_eval)
df_subs.copyright_employer_list = df_subs.copyright_employer_list.apply(literal_eval)
df_subs.copyright_us_govt_list = df_subs.copyright_us_govt_list.apply(literal_eval)
df_subs.sas_employee_list = df_subs.sas_employee_list.apply(literal_eval)
df_subs.company_org_list = df_subs.company_org_list.apply(literal_eval)
df_subs.job_title_list = df_subs.job_title_list.apply(literal_eval)

''' READING THE PDF TEMPLATES

Four templates exist:

    The first page of every CGF is identical (i.e., its structure, fields,etc)
    Subsequent pages can differ, and can have either zero, one, or two signatures required

Because multiple copies of the same template might be concatenated into the same CGF
(e.g., if two coauthors each requires only one signature, the same template is used twice),
the variables must be distinguishable across the pages, so the ADD_FORM_TOPNAME method
must be used to name (and differentiate) only the three templates that can have multiple
copies included in a CGF.

Later, as the form is incrementally built, the RENAME_FORM_TOPNAME method is used to
incrementally rename the topname; thus, the initialized value "temp" is overwritten.
'''

# first page always has the same components
blank_form_p1 = r'WUSS_2024_CGF_p1_BLANK.pdf'
pdf_obj = open(os.path.join(loc_base, loc_blank, blank_form_p1), 'rb')
pdf_reader_p1 = PdfReader(pdf_obj)
# NO signatures required for SAS employees
blank_form_p2_0sig = r'WUSS_2024_CGF_p2_0sig_BLANK.pdf'
pdf_obj = open(os.path.join(loc_base, loc_blank, blank_form_p2_0sig), 'rb')
pdf_reader_p2_0sig = PdfReader(pdf_obj)
pdf_reader_p2_0sig.add_form_topname('temp')
# 1 signature required when organization not required to release
blank_form_p2_1sig = r'WUSS_2024_CGF_p2_1sig_BLANK.pdf'
pdf_obj = open(os.path.join(loc_base, loc_blank, blank_form_p2_1sig), 'rb')

```

```

pdf_reader_p2_1sig = PdfReader(pdf_obj)
pdf_reader_p2_1sig.add_form_topname('temp')
# 2 signatures required when organization is required to release
blank_form_p2_2sig = r'WUSS_2024_CGF_p2_2sig_BLANK.pdf'
pdf_obj = open(os.path.join(loc_base, loc_blank, blank_form_p2_2sig), 'rb')
pdf_reader_p2_2sig = PdfReader(pdf_obj)
pdf_reader_p2_2sig.add_form_topname('temp')

''' BIOGRAPHICAL DATA QUALITY CONTROL

The biographical questions that feed the author-specific data are user-defined and were
added in SoftConf by the Academic Chair for WUSS 2024. As they are optional, some users
might be missing these fields, in which case they must be contacted so their information
can be added, without which their CGFs cannot be printed.
'''

# miss_any = TRUE if any of the biographical fields are missing
df_subs['miss_govt'] = df_subs.copyright_us_govt_list.apply(lambda x: '' in x)
df_subs['miss_sas'] = df_subs.sas_employee_list.apply(lambda x: '' in x)
df_subs['miss_emp'] = df_subs.copyright_employer_list.apply(lambda x: '' in x)
df_subs['miss_job'] = df_subs.job_title_list.apply(lambda x: '' in x)
df_subs['miss_org'] = df_subs.company_org_list.apply(lambda x: '' in x)
df_subs.loc[(df_subs.miss_govt == True)
            | (df_subs.miss_sas == True)
            | (df_subs.miss_emp == True)
            | (df_subs.miss_job == True)
            | (df_subs.miss_org == True), 'miss_any'] = True
# print list of authors/coauthors if biographical data are missing
print(df_subs.loc[df_subs.miss_any == True].email_list)

# filter to remove these authors
df_subs = df_subs.loc[df_subs.miss_any != True]

''' BUSINESS RULES

Business rules prescribe the following, each of which is applied in order:
    if any author is a US govt employee, the copyright is held in public domain
    if all authors are SAS employees, the copyright retained by SAS Institute
    in other cases the non-exclusive copyright is retained by WUSS
'''

# any_govt = TRUE for submissions where ANY of the authors are US govt
def any_val_yes(lst):
    for x in lst:
        if x != 'NO':
            return True
    return False
df_subs['any_govt'] = df_subs.copyright_us_govt_list.apply(any_val_yes)
# all_SAS = TRUE for submissions where ALL of the authors are SAS employees

```

```

def all_val_yes(lst):
    if '' in lst or 'NO' in lst:
        return False
    else:
        return True
df_subs['all_sas'] = df_subs.sas_employee_list.apply(all_val_yes)

''' PDF PAGE HEADERS

The second and, where applicable, subsequent pages are prefaced by the author number,
displayed as an ordinal number (i.e., First Author, Second Author, etc.)
'''

author_number_dict = {1: 'First',
                      2: 'Second',
                      3: 'Third',
                      4: 'Fourth',
                      5: 'Fifth',
                      6: 'Sixth',
                      7: 'Seventh',
                      8: 'Eighth',
                      9: 'Ninth',
                      10: 'Tenth'}

''' DYNAMICALLY CREATING CGFs

Given the number and complexity of the operations performed for each CGF, the DataFrame
is iterated to create one CGF per row
'''

for idx, row in df_subs.iterrows():
    pdf_writer = PdfWriter()

    # FIRST PAGE

    pdf_writer.append(pdf_reader_p1, pages=[0])

    # assign values on first page
    presentation_id = row.sub_id
    title = row.title
    authors = ', '.join(row.fullname_list)
    date_deadline = r'August 16, 2024.'
    publisher = r'Western Users of SAS Software'
    if row.any_govt == True:
        copyright_owner = 'public domain'
    elif row.all_sas == True:
        copyright_owner = 'SAS Institute'
    else:
        copyright_owner = r'Western Users of SAS Software'

```

```

# write first page
pdf_writer.update_page_form_field_values(
    pdf_writer.pages[0],
    {"presentation_id": presentation_id,
     "title": title,
     "authors": authors,
     "date_deadline": date_deadline,
     "publisher": publisher,
     "copyright_owner": copyright_owner},
    auto_regenerate=False,)

# SUBSEQUENT PAGES

# loop over authors
for pg in range(1, len(row.email_list) + 1):
    p = 'p' + str(pg) # used as grouping top-name for fields

    # assign values on subsequent pages
    show_company_signature = False
    if row.copyright_us_govt_list[pg-1] != 'NO':
        author_text = (r"The author certifies that he/she prepared the Work as an employee
of the U.S. government and prepared" '\n' "the Work as a part of his/her employment and that
the Work is therefore not subject to U.S. copyright protection.")
    elif row.sas_employee_list[pg-1] != 'NO':
        author_text = (r"The author acknowledges that he/she is a SAS Institute
employee.")
    elif row.copyright_employer_list[pg-1] != 'NO':
        author_text = (r"By signing, the author acknowledges that an authorized
representative of his/her company or organization" '\n' "is required to sign to publish the
Work.")
    show_company_signature = True
    company_text = r"By signing, the author's employer's representative authorizes
publication of the Work."
    else:
        author_text = (r"By signing, the author acknowledges that no further permissions
are required by the author's company or" '\n' "organization to publish the Work.")

    # write subsequent pages
    if row.sas_employee_list[pg-1] != 'NO':
        pdf_reader_p2_0sig.rename_form_topname(p)
        pdf_writer.append(pdf_reader_p2_0sig)
        pdf_writer.update_page_form_field_values(
            pdf_writer.pages[pg],
            {p + '.author_number': ' '.join([author_number_dict[pg], 'Author']),
             p + '.author_text': author_text,
             p + '.author': row['fullname_list'][pg-1],
             p + '.job_title': row['job_title_list'][pg-1],
             p + '.author_company_org': row['company_org_list'][pg-1]},
            auto_regenerate=False,)
    # reset required in case the same blank PDF has already been imported/cloned

```

```

        pdf_writer.reset_translation(pdf_reader_p2_0sig)
    elif show_company_signature == True:
        pdf_reader_p2_2sig.rename_form_topname(p)
        pdf_writer.append(pdf_reader_p2_2sig)
        pdf_writer.update_page_form_field_values(
            pdf_writer.pages[pg],
            {p + '.author_number': ' '.join([author_number_dict[pg], 'Author']),
             p + '.author_text': author_text,
             p + '.author': row['fullname_list'][pg-1],
             p + '.job_title': row['job_title_list'][pg-1],
             p + '.author_company_org': row['company_org_list'][pg-1],
             p + '.company_number': ' '.join([author_number_dict[pg], "Author's Authorized
Representative"])}),
            p + '.company_text': company_text,
            p + '.rep_company_org': row['company_org_list'][pg-1]},
            auto_regenerate=False,)
        # reset required in case the same blank PDF has already been imported/cloned
        pdf_writer.reset_translation(pdf_reader_p2_2sig)
    else:
        pdf_reader_p2_1sig.rename_form_topname(p)
        pdf_writer.append(pdf_reader_p2_1sig)
        # print(pdf_reader_p2_1sig.get_fields())
        pdf_writer.update_page_form_field_values(
            pdf_writer.pages[pg],
            {p + '.author_number': ' '.join([author_number_dict[pg], 'Author']),
             p + '.author_text' : author_text,
             p + '.author': row['fullname_list'][pg-1],
             p + '.job_title': row['job_title_list'][pg-1],
             p + '.author_company_org': row['company_org_list'][pg-1]},
            auto_regenerate=False,)
        # reset required in case blank PDF has already been imported/cloned
        pdf_writer.reset_translation(pdf_reader_p2_1sig)
    filled_form = r'WUSS_2024_copyright_grant_form_' + str(presentation_id) + '.pdf'
    with open(os.path.join(loc_base, loc_completed, filled_form), 'wb') as output_stream:
        pdf_writer.write(output_stream)

```



SEIZE THE DATA
SACRAMENTO, CA

WUSS

WESTERN USERS OF SAS SOFTWARE

SEPTEMBER 4TH - 6TH 2024 | WWW.WUSS.ORG

WUSS 2024 Copyright Grant Form (CGF)

Dear Author(s),

Congratulations once again for being accepted into the WUSS 2024 conference proceedings, held at the Hyatt Regency Sacramento, September 4 to 6, 2024. We look forward to hosting your presentation in person!

This Copyright Grant Form (CGF) applies **only** to the following paper/presentation, hereafter known as “**the Work**”:

Presentation ID: 109

Title: [Performing with Python Pandas: Taming Your Inner Snake](#)

Author(s): [Troy Martin Hughes](#), [Pigsly McMurphy](#)

Note that the preceding title must match the Work title that is submitted to WUSS for publication, and that the preceding author name(s) (and order thereof) must match the author name(s) that appear in the Work. If you detect a discrepancy in the title or in the author(s), please immediately notify your Section Chair, who will modify these data in the WUSS Submission System, and provide an updated CGF to complete.

Your presentation will not be scheduled until and unless this CGF is completed, and any delay in submission of this PDF could result in your presentation being removed from the conference proceedings. Only one CGF should be submitted per presentation, so all author(s) must digitally sign the same form, after which the completed CGF must be uploaded to the WUSS Submission System (<https://softconf.com/n/wuss2024>) no later than **August 16, 2024**.

Adobe Acrobat Reader is required to digitally sign this PDF, and free software can be obtained here: <https://get.adobe.com/reader/>. If you do not already possess a digital signature, you can create a “self-signed digital ID” for free. Instructions for creating your digital ID are here: <https://helpx.adobe.com/acrobat/using/digital-ids.html>. **Only digitally signed documents will be accepted by WUSS.**

Publisher and Copyright Owner

The **Publisher** of the Work is: [Western Users of SAS Software](#)

The **Copyright Owner** of the Work is: [Western Users of SAS Software](#)

The undersigned, being all of the authors of the above captioned Work, hereby grant(s) a perpetual, worldwide, royalty-free, nonexclusive right under the copyright in the Work to the Publisher to publish the Work in the WUSS 2024 Conference Proceedings and to take the Work, as published, and reproduce it in sales and marketing literature produced by the Publisher, effective if and when the Work is accepted for publication by the Publisher. A copyright notice in the name of the Copyright Owner is to accompany every reproduction of the Work if Publisher’s publication or reprint does not include an overall copyright notice protecting the Work. This grant is subject to the reservation of the Copyright Owner’s rights to (1) copyright ownership, including use or publication of all or any part of the contents of the Work in the course of their business or in any publication authored or published by or for WUSS or its designee; (2) reproduction and/or distribution of the Work but not the Publication as a whole as published by Publisher, and (3) approval of any changes in the Work by the Publisher. No proprietary rights other than this grant under copyright are included, and no transfer of any rights in the Work may be made by Publisher to a third party without the prior written approval of the Copyright Owner.

FIRST AUTHOR

By signing, the author acknowledges that no further permissions are required by the author's company or organization to publish the Work.

Author Name: Troy Martin Hughes
Job Title: Director of Camelid Satisfaction
Company/Organization: Data Llama Analytics, LLC

A large, empty rectangular box with a red border, intended for the author's signature.

SECOND AUTHOR

By signing, the author acknowledges that an authorized representative of his/her company or organization is required to sign to publish the Work.

Author Name: Pigsly McMurphy
Job Title: Pork Data Scientist
Company/Organization: Sizzlin Pork Loins, LLC



SECOND AUTHOR'S AUTHORIZED REPRESENTATIVE

By signing, the author's employer's representative authorizes publication of the Work.

Authorized Representative Name:
Job Title:
Company/Organization: Sizzlin Pork Loins, LLC

