# Table of Contents

        - contact workiva support for issues with Workiva Scripting or Workiva APIs
        - check the Developers & APIs community for help with your code
    - Recommend resources to learn Python and APIs
7. Admin API Recipes
8. Platform API Recipes
    - Spreadsheets overview
    - Copy a spreadsheet (create + update)
    - Format a spreadsheet
    - Export a spreadsheet
    - Update embedded table
    - Refer where to learn more about Workiva and the Platform API (wrap-up)
9. Wdata API Recipes
    - Wdata overview: what is wdata and quick UI walk through… and you can do the same with Wdata APIs
    - Loading data
    - Extracting data
    - Updating a file
    - Refer where to learn more about Wdata product and API (wrap-up)
10. Scripting API Recipes
    - Overview
    - Push the latest changes to your script
    - Run a script, refresh token, and get status + output
    - Refer where to learn more about the Workiva Scripting API (wrap-up)
11. Workiva Public APIs and Workiva Scripting Tips and Tricks
12. Use Cases: Hide empty rows at scale
    - Explain challenge and process w/o APIs/Scripting
    - Provide an overview of the solution
    - Explain how to implement the solution
    - Provide a quick code walk-through
    - Explain how to execute the solution and what the results are
        - ex: manually run a IA/script execution once a quarter, the IA shows that run successfully, 100+ documents/entities don't display empty rows
    - Explain how to access this accelerator (today: contact partner/Workiva; tomorrow: marketplace)
13. Use Cases: Schedule of Investments (SOI) at scale
    - Explain challenge and process w/o APIs/Scripting

- Provide an overview of the solution
- Explain how to implement the solution
- Provide a quick code walk-through
- Explain how to execute the solution and what the results are
    - ex: schedule a chain once a quarter, the chain shows that run successfully, 10+ documents/funds contain now customized SOI
- Explain how to access this accelerator (today: contact partner/Workiva; tomorrow: marketplace)

## Intro

[Next slide - Speakers]

- Hi, my name is Jesus Bouzada. I am a Product Manager for the Workiva Developer Platform

- I am really excited to have you with me and I hope I can make the best of our time today

[Next slide - Agenda]

- This journey today will take us through the Workiva Developer Platform and its components: Workiva APIs and Workiva Scripting

- I will share exciting real life examples of how the developer platform is adding tremendous value to our customers

- And I will end by sharing additional resources for you to continue learning about these awesome capabilities we are building.

- Ready? ok, let's go

## Development Platform

[Next slide - Workiva Dev Platform]

- We are starting this journey today with an introduction of the Workiva Developer Platform

[Next slide - POV Opening]

- The Workiva developer platform is an instrumental part of opening the platform
- It equips you to extend the value of our platform with new regulatory, financial and ESG reporting solutions

[Next slide - Solutions]

- Many successful SaaS companies transitioned from a singular product to a technology platform that powers multiple solutions. And we went through that process in Workiva
- We started with an initial beachhead solution like SEC, and expanded to multiple other solutions to create a sticker value prop with our customers
- The Workiva platform now powers 10 different solutions from Global Statutory Reporting to ESG

[Next slide - Growth Flywheel]

- The flywheel starts to spin with the introduction of a new fit-for-purpose solution that solves a real problem for customers. So customers buy the solution
- This, in turn, generates resources for our company to reinvest into that solution or create new ones
- But Workiva can only build a limited number of solutions with the resources and the knowledge we have
- Left are many unique needs that are unmet

[Next slide - Growth Flywheel with builders]

- We have added an outer loop where builder tools and marketplace equip and enable you and sets that flywheel spinning even faster
- Delivering much more value to our customers

[Next slide - Automation Ecosystem]

- Our automation ecosystem is part of the builder toolkit
- There, we have a great set of tools for configuration and management of data and processes
- You are probably familiar with these tools. I am talking about Chains, Data Prep, wdata, and processes among others

- But sometimes you need to go just a bit further and extend those capabilities with a bit of custom logic

[Next slide - Automation Ecosystem 2]

- Our public APIs and Workiva Scripting provide that critical extension point that allows external builders to create and run custom logic that can augment the current automation capabilities.

- To be clear, Chains and Data Prep continue to be critical to extract and transform data that get loaded in wdata.
- And APIs and scripting complement those tools by helping to automate complex processes.
- For example, you wouldn't want to build a python script to connect an SAP source with Workiva when there is a Chain connector to do that already.
- But you may want to format the data based on specific requirements with a Python script

# Workiva Public APIs: Intro & Overview

[Next slide - Workiva Public APIs]

- Ok, we are going to transition now and put the lense onto the Workiva Public APIs

[Next slide - Platform API]

- We introduced a Platform API that gives you access to the data and services of the platform.
- This API lets you interact with Files, Tasks, The graph database and most recently we added support to format spreadsheets and tables via the API.
- These APIs are helping you integrate with other systems, drive automation and create custom solutions against our platform.
- Since introducing our Platform API we have seen incredible adoption with about 15% of our accounts leveraging these APIs.

[Next slide - Admin API]

- As many of you started to work with hundreds and even thousands of entities and users, you told us you needed the ability to automate the setup of Workiva for these customers.
- So we created an Admin API and I  am excited to tell you we are adding the ability to create Workspaces, Groups, Memberships and Roles through this API which will save you countless hours and give you confidence through automated control of the setup.

[Next slide - SCIM API]

- We also recognize many customers have systems that speak industry standards such as SCIM for identity management.
- Workiva is also committed to supporting these API standards so our platform can seamlessly integrate with Okta, Sailpoint, Active Directory and others.
- As these standards evolve or as new standards are introduced we will continue to evolve to support these.

[Developer Hub]

- If you haven't already, I'd encourage you to visit developers.workiva.com

- This is the documentation experience we've built for you to learn about these APIs, and there is sample code there too.

- The main head in there is the platform API, the primary API we've been working on
- And in that first row you can see the Admin API

- You see a few other API definitions. These are areas that we are going to be consolidating down some into platform, others into admin.
- But this is something we haven't got to yet

- In the next few minutes, I want to get more hands on with you and show you how you can interact with these APIs

[Authorized APIs - Add a Grant]

- What do you see first when you try to access wdesk?
- The login screen, right? You need to get authorized before you access the data in the Workiva platform

- The same happens with the Workiva APIs
- Your code will require access to user data in the Workiva platform,
- you need user authorization for your code.

- You can get user authorization in wdesk if you are Workspace Admin.
- Go to the Classic Account Admin and then go to People, and OAuth2 Grants
- Once there, click the button Add a grant

[Authorized APIs - Select Scopes]

- The next step is that you will need to fill information. And I wanted to focus on the scopes first. In the developer hub, you can find out what scope you need for an specific endpoint.
- For example, if I wanted to create a new task, I will need the scope to write tasks. As you see in this image here

[Authorized APIs - Create the Grant]

- As well as the scopes, you will need to provide
  - Grant Name
  - Your user name
  - And expiration date
- Once you fill that in, you can create the grant

[Authorized APIs - Generate Credentials]

- That generates your client secret. Your code requires a client id and a client secret to get authorized into the Workiva Platform.
- You can find them by clicking the edit button on your grant
- They will be at the top of the pop up form

- If you are not a workspace admin, ask your admin to generate the credentials and share the client id and the client secret with you

- One last thing here, you must not share this information with anyone else. Sharing client id and client secret is like sharing your username and password to login to wdesk. Don’t do it

- As you see in these screen shots, my credentials are hidden.

- Now, we are going to take a look at sample code that use these credentials, the client ID and the client secret,
- to generate an access token that gets your code authorized into Workiva Public APIs

- And the code we'll look at is in Python
- But you can use any language
- In the end, the HTTP calls generated that go across the wire, are the same regardless of which language you use.

[Authorized APIs - Code Sample 1]

- You provide the client id and the client secret to your code.
- Here, I store them as environment variables in my system and then read them in my code.

- Next step to generate the access token is to build the url of the API you want to call. In this case, I want to retrieve a token

[Authorized APIs - Code Sample 2]

- You do that by adding the end point you want to use to the base url
- You can find both in the developer hub.
- The base url is within the API Details section. If you are in US, you use the one at the top. That is the one I use in this code sample
- The end point to retrieve a token is in the section with the same name, under the identity management API

- We have now the url. Next, you find what type of request you need

[Authorized APIs - Code Sample 3]

- To retrieve a token you do a post request

[Authorized APIs - Code Sample 4]

- To retrieve the token,
    - you need a request header like this
    - and send your client id and your client secret as data in the request

- But, wait, what is this request thing I keep talking about?

- To answer that, we need to understand how HTTP-based REST APIs work.
- HTTP what?
- As most SaaS applications today, Workiva APIs are HTTP-based REST APIs

[HTTP REST APIs 1]

- Well, everything happens in a request-response workflow

- Down here, you can see sample code to access Workiva APIs, the tasks API in this case.

- Let’s do a code walk-thru

- This, in line 20, is how you build the url to call the API to create a task
- Any access to the Workiva Public APIs requires an access token. And I just showed you how to generate one. We use that token in line 25
- This API endpoint requires two data points, assignee and tittle
- With the token and the data I can make a post request to the url

[HTTP REST APIs 2]

- The Workiva platform will receive the request and confirm the credentials,
- The platform will create the taks and return a response that contains information about the new task.

[HTTP REST APIs 3]

- Then, we can use that response in our code and continue the process we want to automate.
- It's very much of a client-server model.

- I added down here the link for the task creation documentation in the developer hub

- Ok, by now, we've seen the types of APIs in the Workiva Platform, how to authorize your code, and how to build a request and receive a response

[Data Model]

- But let me step back for a moment and talk about the data types your code interacts with in the Workiva Platform

- Something you will hear when people talk about APIs is this concept of resources or objects
- What this is is the things in the APIs that you can interact with
- These resources and objects are the exact same thing that you talk about with our customers when you are describing the Workiva platform and the solutions you are building
- It's things like users, tasks, a document, a spreadsheet, a process, etcetera
- And we just use those resources to do different operations
- "Give me all the users"
- "Create a task"
- "Update a file name"
- They are the things we interact with

- And the way we interact with them is through what we call endpoints
- Endpoints are the location that allows you to interact with these resources
- The location is usually represented by a url type structure as we have seen in previous slides

- All of these endpoints and resources get rolled up into what we call an API definition.
- You can find Workiva API definitions on our developer hub as you see here on the right for the tasks API or the spreadsheets API

- And I used Python and the requests library. If you are familiar with APIs already, you may be wondering if there are language-specific libraries, or SDKs, that you can use to interact with Workiva APIs more easily

[Code Generation - 1]

- Unfortunately, as of today, we are not providing public SDKs for Python or Java or other popular programming languages.

- That is something we keep talking about, but no plans for now

- The good news is that we do provide Open API specifications for Workiva APIs so you can generate these libraries yourself.
- And I wanted to tell you how you could do that

- The first thing you need to do is to go to the Code Generation section of the Platform API in the developer hub, and download the yaml file that contains the Open API spec

[Code Generation - 2]

- Next, you can use the generator project to generate your client. I shared the link here. It is pretty easy to use
- In this example, I used it to generate a Python client as you see on the CLI Installation article I screen shot for you here
- The result was this folder on the right with all the artifacts I need in my Python project

[Code Generation - 3]

- Then, you import the clients in your code like I did here.

[Code Generation - 4]

- This code does the exact same thing I shared with you earlier but it uses the generated client
- As you see, the client simplifies your code and allow to more easily authorize into the tasks API and create a task

# Automate and Extend with Workiva Scripting

[Next slide - Workiva Scripting]

- We are moving on to Workiva Scripting now, let's see what this is all about

[Next slide - Why Scripting]

- To start with, I want to tell you why we are doing Scripting

- We had some conversations with partners and customers around how they were using our APIs and Automations tools
- They were struggling to efficiently automate customers' unique needs with our capabilities then
- And we were seeing how successful platforms were addressing this by offering builder capabilities including the ability to run custom code
- And you may think: "but code that leverages Workiva APIs can be run outside the platform"
- There are issues there
    - You would need to build dedicated infrastructure resources for customers. When they already have the Workiva Platform
    - And what is more critical, most customers will not be comfortable letting their data out of the Workiva Platform and back in. They trust Workiva to keep their data secured

[Next slide - What is Scripting]

- Our solution to this was to introduce a new platform capability called Workiva Scripting.

- Workiva Scripting is available today.

- What this does is to allow you to create and run Python scripts on top of our secure FedRamp compliant platform.

- When you get Scripting enabled in your workspace, you see a new icon in the create menu to create Scripts.
- Workiva Scripts are files hosted in the Workiva Platform that you can manage from the home experience the same way you do with other Workiva files like documents or spreadsheets.

- To build your scripts, you can use your own local editor or the Scripting editor in Workiva.

- Using our public APIs you can interact with the Platform from a Workiva Script and implement any custom logic or automation you need.

- Then, your Workiva scripts can be triggered from a chain, HTTP request or even an integrated automation within your Workiva files.

- Let's dive into this now

[Next slide - Running scripts in the Workiva Platform]

- You will need runner credentials and script credentials to run scripts in the Workiva Platform

- Runner credentials allow you to trigger the script run. There are three things you need
    - You will need to generate credentials to authorize into the Scripting API. We covered this earlier.
        - You go to the Classic Account Admin and Add a Grant. You don't need any scope to interact with the Scripting API
    - You will also need to have one of the Scripting roles.
        - Script Editor, which allows you to create, edit and run scripts
        - Script Viewer. With this you can see the code and run scripts, but not create or edit scripts
        - And Script Runner is the least privileged role that only allows you running scripts
    - Finally, you will need to have permission to at least view the specific script. This is file level permission.

- All of this will allow you to trigger the run, but you will probably also want your code to interact with the Workiva Platform.
    - That is the authorized APIs part that we covered earlier.
    - For example, you need an API grant with scopes to edit spreadsheets and permission to edit a specific spreadsheet if you want to edit that spreadsheet with a Workiva script.
    - You get that, right?

- Once you have these credentials, there are four ways to run Workiva scripts

[Next slide - Running scripts with the Editor]

- To begin with, you can run Workiva Scripts right from the Scripting Editor available in wdesk.

- This is a convenient option to develop and test your code

- You can use the Run Parameters form to enter your input parameters
- Workiva Scripting stores those parameters as environment variables in the system
- In your code, you get the parameters as you would do with any environment variable

- The editor also provides limited, but convenient functionality to track and troubleshoot past runs. I will expand on this in a moment during the demo

[Next slide - Running scripts from Workiva Files]

- Running scripts from Workiva files is probably the way you want to go to have your end users running your scripts.
- Providing that this option is enabled in your Workspace, you can setup a manual script execution with the right side panel, the lighting bolt
- It is early days for this capability. At this time,
    - you provide your script id and,
    - if you need to provide input parameters, you can provide the spreadsheet where the key, value pairs are defined
- Once it is setup, you trigger your automation manually as you see here

- Your code to dynamically identify the triggering document and the sheet where you define your input parameters would look something like this.

[Next slide - Running scripts from Chains]

- The next option you have to trigger your Workiva scripts is from Chains.
- And this is great if you have a super complex Chain and you want to extract some of the logic into more manageable Python scripts
- Or you may just want to leverage the power of Chains and its awesome features like scheduling or monitoring the runs

- We have built a connector especially for your Workiva Scripts.
- The connector provides great logging capabilities and ensures Chains tracks and reports  completion of even the longest runs

[Next slide - Running scripts from Scripting APIs]

- Another option you have to run scripts is by hitting the Scripting API directly.

- You will use this if you want to run scripts from outside Workiva, you may want to integrate with your systems for example.

- You do a HTTP POST request to the /runs endpoint and send input parameters with the body of the request. That starts the run and returns a run id.

- Then, you would add the run id right after /runs to create new endpoint that you use to check status. I am talking about this endpoint here.

- Once that endpoint returns the complete status. You can use this other endpoint to get the logs if you need that.
- That will give you the output that you see in the output panel of the Scripting editor.

- A note, you will need to refresh the access token if you run goes over 10 minutes. Access tokens expire after 10 minutes.

- All this magic is in essence what the Chains Scripting Connector does for you.

- These are all the options you have to run scripts

[Next slide - Demo]

- Let’s see these things now in action with the demo

- Home
    - Scripting is enabled to selected workspaces only.

    - Once it is enabled, you go to Settings and then, Members to assign Script Editor role to the members that need to create and develop scripts

- Settings
    - Like I did here for my user

    - And you also have Script Viewer and Script Runner roles if you want to restrict use of Scripting for some users

- Let’s go to Files and start working with Scripts

- Files
  - When you have the Script Editor role, you will see a rocket icon in the create menu
  - You click that icon to create a script

- Editor
  - When you create a script or open an existing one, you are directed to the editor
  - Before I take you through the editor, let me go back to my files to make one more point there

- Files

  - Scripting is integrated with the Workiva home experience
  - The same way you can create documents and spreadsheets, or search and open them, you can do that with scripts. You will have a new file type, scripts file type
  - Also similar to files, you can add script owner, editor, or viewer permission to a user

  - Let’s explore the editor now

- Editor

  - This is one of my scripts

  - What you see here is
    - an editor in the center that helps you with the syntax,
    - a menu bar to do things like saving and running the script at the top
    - side panels to help manage your source code (here to the left) and properties and see the history of the runs (here to the right)
    - and you can see the output of the run you select here at the bottom

  - It’s a good practice to modularize your code.

- init.py

  - In this example, I created one source file to get the auth token to authorize into our public APIs

- Spreadsheet.py

    - and another source file to interact with our spreadsheets API

- Main Method
    - Once you have your modules in place, you just need to orchestrate everything in the main module and you are ready to test your script.
    - From the editor, you can run your script, sending any parameters or data to that script as necessary. This is a great way to test your scripts

- Run Parameters modal
    - Here, you add the name of the parameter and the value, and click run to trigger a new script run
    - And let me share real quick what happens when you click run
    - Every time you click run here, or you run a Workiva Script with the other options we covered earlier, a container is created behind the scene and python is installed along with a few other drivers
    - That container is thrown away once the run is completed

- Editor
    - To read those parameters, your code will be getting values from environmental variables set in the thrown-away container.
    - Like right here in my code
    - You will also want to connect with the outside world and use libraries and call APIs
    - Workiva Scripting supports Python built-in libraries, so you can use those right away

- requirements.txt
        - If you need to use one of the libraries available in the PyPI Package,
            - You can manage those dependencies in the Scripting Editor
            - you will add a source with this exact name, requirements.txt, like I did here
            - Then, you add the library and the version of the library you want in a separate line

        - Any update to the txt file will rebuild a dependencies.zip file with your dependencies
        - That is used behind the scenes to install your libraries in the thrown-away container used to run your script
        - There may be some libraries that require additional drivers, those will not work in Workiva Scripting

    - You will also want to interact with the Workiva Platform, you can only do that via Workiva Public APIs. This way,
        - We keep scripts portable, the same Python code that works for you locally, works with Workiva Scripting
        - And we keep the execution of scripts in the Platform secured

    - You can also interact with non-Workiva APIs from Workiva Scripts.

    - At this time, you will need to send the API credentials as parameters for your code to get authorized into the APIs.

    - Everything you can do in the Scripting editor, you can do with the Scripting API, and more

- Swagger

    - For example, you can use the Scripting API to copy Workiva scripts from one Workspace to another

    - As you see here with this Swagger UI tool, the Scripting API offers endpoints
        - to interact with Scripts themselves,
        - to interact with the content of the scripts, with the actual code,
        - and to interact with the execution of scripts

- Files

    - This is a lot of information. Let me take you through a use case that you are probably familiar with

- Singapore spreadsheet - SFP

    - Imagine you have this GSR customer that comes to you and tells you:
    - “hey, the roll forward process for all my entities is taking too much time doing manual work”
    - and they explain the manual effort:
        - “this is the spreadsheet where the data rolls forward every quarter for the Singapore entity,

        - say that new data comes to the statement of financial position here

- Singapore doc - SFP original

        - I will need to take that data into the report and format the data accordingly to criteria specific to the singapore entity. In this case, the data should look like this”

    - And the customer goes on saying:
        - “and this is just one statement of many in a single report, and I have hundreds of entities that I need to generate reports for, and each with different criteria”
        - “I just cannot go on working this way”

    - Then, you tell your customer:
        - “what if I told you that
        - with Workiva APIs and Workiva Scripting
        - I can make it so you bring all the data from the spreadsheet to the report
        - and formatted based on the entity specific criteria
        - and just with the click of a button right there, in your report”

    - That is exactly what the script that I have been showing you does.

- Singapore doc - SFP empty

    - So let’s say that your customer needs to bring the data for the statement of financial position here, to this empty section and format it

- Singapore doc - IA

    - And I have created this integrated automation here that you see when I click this lighting bolt to access my integrated automations

- Singapore doc - IA Edit

    - I enter here the script Id and the information for the spreadsheet where the data is.
    - That is how you would create an automation to manually execute a script

- Singapore doc - IA

    - If I go back to the list of automations, I can manually run it from here

- Editor

    - When I do that, I will see a new run when I come back to the script

- Singapore doc - SFP empty

    - And back to the document, I will start seeing data coming up after a few seconds that take to create that thrown-away container I told you about

    - I am going to see how the data comes and get formatted in real time, and how the empty rows are hidden. Pretty cool

    - Now, say that I wanted to run this from chains instead of the report itself. I may want to schedule it, for example

- Chains Edit mode

    - That is very easy if you are familiar with Chains already
    - If I look for Workiva Scripting in my BizzApps I can find the new Scripting Connector (providing that that was enabled for you in advanced)

    - I can drop this Run Script node into my chain. I will not do that because I already did
    - But let's look how to set it up

- Chains edit Scripting Connector

    - It is very easy to setup, I just need to provide my script id and enter my input parameters
    - In this case, I want to send only the client id and the client secret.
    - But I could also send information for the spreadsheet where I have the data, and information for the document where I am creating my report
    - In this example, that information is hardcoded in the script

    - I will also need to setup a connection. Let me show you my connection quickly

- Chains Connection

    - I dont want to spend too much time here.
    - You just need to know that here is where you setup
        - the runner credentials as well as the location for the Scripting API
        - and the Authorization API that is used to generate the token
    - You may know this already if you have experience with Chains

- Chains Edit mode

    - Back to my chain
    - One thing I really like from Chains is that I can setup variables

- Chains Settings

    - This is really convenient to protect your secrets as you see here

- Chains Edit mode

    - let me execute now
    - It's running now, let's go to the scripting editor

- Editor

    - You can see a new run has initiated, so we should see some data coming to the report again, let's check

- Singapore doc - SFP empty

    - Here it is. I am glad that things are working well even if I am doing this in real time,
    - You know how things can go south sometimes when you demo in real time

- Chain Last run

    - Back to Chains, we see how the run completed

- Chain Last run - Logs

    - I really like how the Connector allows you seeing the script output as if you where in the scripting editor, that is really convenient

    - And this is all I had to demo today, but we can expand on this later during the Q&A if you have questions

    - I will keep going in the interest of time

[Next slide - Qualification Criteria]

- Let me tell you about the qualification prerequisites we recommend you to complete before you take on implementations that include Workiva Scripting
- Completing these prerequisites will increase significantly your chances to succeed with Scripting implementations

- We are recommending you that two or more people within your group complete these prerequisites
- And this image on the left side is our hand out with the details that we may have shared with you already since you are interested in supporting Scripting implementations

- So what are these prerequisites that we recommend your people to complete?
- We recommend they pass the Workiva Platform certification
- We recommend they complete this training that you are doing.
- We recommend they have intermediate Python skills.
- To validate they have intermediate Python skills, the Certified Associate in Python Programming certification from the Python Institute is a great resource.
- Finally, we recommend they have good understanding of working with REST APIs. Advance certifications from the Python Institute demonstrate that. But the online courses in this hand out are great too.

[Next slide - Deployment Scenarios]

- Last thing I wanted to cover for Workiva Scripting is the deployment scenarios that we are seeing being used

- On the left, there is the model where customers host scripts in their own workspaces.
    - Here, you may build the script in your workspace. But your end goal is to transfer ownership of the script and the whole solution to your customer

    - This allows to customize the solution to the specific needs of your customer.

    - Your customer may not be qualified to do Scripting implementations.
    - We recommend you to keep providing support services for that solution after you do the initial bulk of the work and deploy the solution.

    - We (Workiva) don't have the bandwidth and the expertise to support somebody else's code. You are experts of the code you build
    - However, we own a collection of scripts that have been used for real life use cases and have added value already.
    - We call those Accelerators. While they are not available via the marketplace yet, we are happy to share those.

    - We expect you make them your own and support your customers using them

- On the right, this model is more similar the iPhone AppStore model
    - You build a script and make it generic enough that can be used by multiple customers
    - You host that script in your own workspace. And, in the future, customers may be able to install this scripting-based automation in their workspace
    - In this model, your customer does not even know that Workiva APIs and Scripting power the automation

- There may be a hybrid model where you deploy in the customer's workspace but you keep the IP. That is also possible.

- And with that, we are concluding this intro video for Workiva Scripting.

## Use Cases

[Next slide - Use Cases]

- We have had some of you participate in the introduction of Workiva Scripting and the results have been amazing.

- Let's look into three real life examples that will give you more context

[Next slide - Zero Suppression Challenge]

- In this example, one of our customers is using us for GSR, they have hundreds of entities around the globe
- and each of these entities will have a document, in that document there will be linked tables
- when they do the roll forward process, the way the data result for an entity, those tables will often end up with empty rows.
- Very similar to what I showed you earlier in the demo
- Another manual process that is countless hours of work
- But, what's more important, customers are saying, "hey Workiva, we cannot expand to more entities if every time we add an entity we need to expand the manual process, it just doesn't scale"

[Next slide - Zero Suppression with Scripting]

- using our public APIs and scripting, you can completely automate that, so this document with this table with these empty rows, those go away
- Investing just a few hours, you can create a reusable script and show customers how that script could run, automatically go through their document, scan through the tables, and hide the empty rows
- In this example, the script was able to do this in just a minute, and only in one document we hid 770 rows automatically

[Next slide - Investor Correspondence Challenge]

- In this next example, a Fund Reporting customer needed to automatically generate over 400 investor PDF reports every quarter.
- Imagine if someone had to manually create and export every one of those reports

- Our services team tried to solve this with our traditional automations tools
- But the resulting automation was incredibly complex and difficult to maintain. Also, it wasn't meeting the customer requirements around formatting and speed, it was taking well over an hour

[Next slide - Investor Correspondence with Scripting]

- With our public APIs and Workiva Scripting, our services team was able to create custom logic that automated that process with a few simple lines of python code.
- That script can now be deployed and run on top of our secure, FedRamp compliant platform.
- What is great is that with that custom logic, we were able to meet the customer's unique formatting and speed requirements.
- Now, in a matter of a few minutes, this script generates over 400 investor reports.

[Next slide - Onboarding at scale Challenge]

- In this last example, onboarding customers some times involve an enormous amount of work creating organizations and workspaces.
- In some cases, we may be talking about hundreds or thousands of them

- Like with this MSP Partner that wanted to create 800 organizations and 5 workspaces under each organization. And they asked us for help

- Imagine if someone had to manually create those. That was happening before we built our Admin APIs

- We estimated that, at 3 minutes per org, manually creating all those 800 orgs would take 40 hours

[Next slide - Onboarding at scale with Scripting]

- Our Admin team saw a great opportunity here to show case the value of the newly created Admin APIs so they chimed in to help as an exception
- They built a script in a couple of hours that used Workiva Admin APIs to create all the orgs, workspaces, and memberships

- And ran the script with Workiva Scripting. Running the script took a few minutes

- A total of 38 hours were saved in this case.
- But, more importantly, this opens up the door for many more hours saved for all of you and all our customers

- These are just three examples, but there are more great examples I have been involved with that showcase the value of Workiva APIs and Workiva Scripting

- We can share some of these solutions as a starter kit for you to make your own, we refer to that as accelerators. Happy to talk some more about that in the Q&A section

- But let's cover the last part first

## Additional Resources

[Next slide - Additional Resources]

- In this last section, I wanted to share with you how you can get help and other learning opportunities around APIs and Scripting

[Next slide - Training Opportunities]

- You can learn more about workiva with the Workiva Essentials path available in Workiva Learning Hub.
- Understanding the Workiva Platform is critical for using Workiva APIs and Workiva Scripting effectively

- And, of course, you are already learning Workiva Public APIs and Workiva Scripting in this path that you are doing

[Next slide - Support Intro]

- When you contact Workiva Customer Support with a problem that you have, there is a decision that our Customer Support needs to make
- And that decision is:
    - is the problem with the Workiva products? the Workiva APIs and the scripting product that we create that allows you to run a script
    - or, in the other hand, is the problem with the code that you wrote

[Next slide - Support Tooling]

- We have created support tooling that will guide our Customer Support on the root cause of the issue
- The scripts, when they exit, will produce a status and the status will tell our Customer Support if
    - it is failed because of us,
    - or it is failed because of the Python code

- This tooling is not available to you at this point.
- But there are ways for you to know if the error is with the Workiva products.
- For example, if you see a 500 error code, that could mean that the error is with Workiva

[Next slide - Customer Support]

- If the error is with Workiva, you submit a request in the Workiva support portal and Customer Support will respond you promptly.

[Next slide - Customer Support 2]

- Customer Support will triage the issue and will escalate to Workiva engineers if the issue is with the Workiva APIs or Workiva Scripting.

- Side note here. While Scripting is available for early adopters only, you can report Scripting issues directly to me.
    - We are still training Customer Support to be able to support Scripting, so they will escalate directly to me while Scripting is for early adopters

- The last point I wanted to make here, and this is very important. Workiva can't debug your Python code. You are the experts on what your code is trying to solve

[Next slide - Community]

- If the issue is with your code, what Workiva Customer Support will tell you is to leverage our online support community
- There is a developers and API section in that community

[Next slide - Community 2]

- Workiva experts will check that community frequently and try to provide guidance and share best practices.

- And this is how you learn and get support on Workiva things

- What about general Python and APIs

[Next slide - Coding Resources]

- Well, you have a ton of resources I there

- I really like the certifications from the Python Institute. The third one here will validate you are intermediate at Python

- There are plenty of online courses too. I was recommended these two from Coursera
- And blogs are extremely useful too, check out these two here
- I was also recommended this book, apparently it is pretty good
- And you can always ask question in stack overflow. If you dont know about it, you will soon

[Next slide - Thanks]

- And this concludes Workiva Public APIs and Workiva Scripting learning path
- I really appreciate the time you invested with me and really hope this path is helping you getting started with the Workiva Developer Platform

- I am looking forward to hearing the amazing solutions you build on the Workiva Platform and how those are making your users' life easier

- Thanks again and see you soon