

Lukas ([00:01](#)):

All right, so today's webinar we're going to be talking about a product called RunScript that lets you generate PDFs online, and we're going to get into all the details of how that works and why it's so powerful. So the demo is going to be done by Ben Hauser, who's the VP of Engineering at Typefi, and I'm Lukas Kaefer, the Marketing Manager. And I'll just be kind of hosting behind the scenes. Quick little about Typefi. If you're not familiar, we're a software company founded in 2001 and based in Australia, and we're all about helping you publish faster and do more with the resources that you already have. So we have an automated publishing software for InDesign, a few InDesign plug-ins, and then obviously RunScript, which we are going to talk about today.

([00:56](#)):

Obviously we're doing intros right now. Then Ben is going to give us some background about how RunScript works, some of the underlying technologies and the capabilities of it. And then he's just going to show a demo, get into a little bit of detail, and then we'll have some Q&A at the end. In terms of logistics, if you're not familiar with Zoom, you've got mute buttons on the left for video and audio. The chat is somewhere in the middle, and then you can always raise your hand as well. We ask that you do please stay muted during the presentation, but during the Q&A at the end, please feel free to speak up, unmute if you have a question, you can also post in the chat and I'll be staying on top of that and going through those. And we are recording this obviously, so that will be available within a day and I will email that to everyone who registered. And so that's it for my bit. So without further ado, Ben, I'm going to turn it over to you.

Ben ([02:00](#)):

Great, thanks Lukas. I hope you can hear me okay.

Lukas ([02:04](#)):

Yep.

Ben ([02:04](#)):

Cool. So RunScript. It's a cloud service from Typefi and essentially it's InDesign Server in the cloud. And one of the most powerful features of InDesign Server is scripting. And we've found that pretty much anything a human operator can do using the InDesign user interface, you can automate that by writing scripts. Adobe has created a wonderful and rich and deep scripting API for InDesign. So yeah, RunScript enables you to use all that power of InDesign scripting in the cloud. Now here's a few key points to note about RunScript. So why would you use it? Well, firstly it's a cloud offering. So that frees you up completely from having to manage your own hardware, monitor the software that's running on it, employ technical people to perform upgrades and maintenance and so on, that hardware and that software, Typefi will manage that for you, one hundred percent in the cloud. The other good thing is that you don't have to worry about paying the annual InDesign Server licence. We've got a model where you pay per use. So we've worked out, the way the pricing works is that you pay, there's a fee per second that a job is running, so there's no upfront fee, there's no monthly fee, it doesn't cost anything unless your job is actually running. And then it's like a metre. We'll track how many seconds the job runs and then you'll only be charged for that.

([04:17](#)):

The price is five US cents per second. I did some analysis of one of our customer's jobs earlier this week and they run thousands of jobs every month. And the average price, I mean the average runtime of their jobs is 5.5 seconds, and they weren't trivial jobs either. So each one of those jobs, they would place multiple images, resize things, place text, apply formatting. There was quite a lot going on. Export a PDF

at the end and all that would only take five seconds per job run. So at 5 cents per second, that's about 25 cents to do one of those jobs. And it is super fast. The InDesign Servers are running hot in the cloud, they're ready for jobs 24/7, and from woe to go the average time takes about five seconds. So they're the sort of benefits of RunScript to keep in mind. I'll share my screen now. Alright. Can you see that? It should be a white page with RunScript on it. Yes, I see some nods.

Lukas (05:55):

Yep.

Ben (05:58):

Okay, so that's sort of the quick overview of RunScript. Now what I will show you is some of that in action. Alright, so what you're looking at here is the desktop of my MacBook. On the left you have, I've got Adobe InDesign, and then on the right I have Visual Studio Code. So what I'm going to show you here is just as a starting point, how to use InDesign scripting on the desktop firstly. So there, there's a document here in InDesign. It's a normal InDesign file. There's no plug-ins installed, nothing special about it. You can see it's highly designed, it looks very pretty. A graphic designer has created this. Another key point when working with InDesign is that you don't have to compromise on design. So we're sort of marrying the power of programming and automation with the pinnacle of design.

(07:14):

So if I wanted to change the name of the employee of the month, I mean one way to do it, the manual way to do it would be a human operator would go to their laptop, open InDesign, open the file and then manually select the text and change it. So that's one way you could do it, but I'll just very quickly show you another way to do it, which is to use a script. So if I select the name here and I just come up with a syntax, I've just typed in sort of double angle bracket. Over here on the right, I'll make it a little bit bigger. So over here on the right, this is JavaScript and it's going to look for that pattern and then replace it with a hard-coded string here, Lukas. So let's see if I can run that. And when I press play, keep your eyes over here in InDesign and you'll watch that name change there. So Lukas, you are our employee of the month. Congratulations, about time I say.

(08:29):

So that is InDesign on the desktop and running a script on the desktop, but if we want to move that into the cloud, that's essentially what RunScript is. So what I'll show you now is the same thing happening, but everything is running in the cloud using Typefi RunScript. So I've created this app here, which is it's a pretend app, employee of the month certificate generator. It's an HTML front end. It's a Node js backend, very simple app that I created. Well, there may have been some AI used in helping me create this. So let's enter another employee's name. So Caleb's on the call. When I click this generate certificate button, this little node app running on my MacBook is going to call RunScript in the cloud, send it the InDesign file, send it the name, and then it'll return back a PDF.

(09:47):

So click the button, shouldn't take too long. And here we go. Something's coming back here and we've got the PDF with Caleb's name on it. So what I'm trying to show there is that you use RunScript running behind the scenes, your own apps, whatever they are, can use it very quickly and silently as a backend service to very quickly create custom PDFs. And I'll just get into slightly more technical detail. I won't go very deep at all. So one of the things to keep in mind when you are using RunScript is that it's essentially it's InDesign Server in the cloud, and that's all it is. It doesn't do file storage, but it needs files to work. So you need to send it your InDesign files and whatever other files you need. And then it does its work and then it creates output files. And so the way that that works is we use a cloud storage service. It can use any sort of cloud storage service to manage files for you. The one that I've been using in these demos is just S3, which is Amazon's simple storage service. You can see I've created an S3 bucket. I've simply

uploaded files into this bucket, and then my programme tells RunScript to go and retrieve these files. And when it's done its job, write the files back here. So that's really the most technical thing you need to understand with RunScript. I'll just show you the backend code of my app.

(11:53):

It's only, what is it, 80 lines here. Most of this code sort of here to here, I'm just telling RunScript, go and get the files from this S3 bucket, and when you're done, write the files to this S3 bucket. Here's where I actually run the job. Then we get the response and then download it and open it in the browser. So a few lines of pretty straightforward JavaScript and you have added custom PDF capabilities to whatever your app is, using Typefi RunScript. So that's really all I wanted to show you. I think something to keep in mind with RunScript is though, this example was just super simple, like a toy example. And as I said, InDesign scripting is super powerful. You can place texts and graphics, you can apply styles, formatting, resize things, move things, thread text between frames, add pages, anything you can do in InDesign, you can automate with scripting. So it's a very, very sort of powerful tool that you have at your disposal. So that's my quick intro to RunScript. So back to you LuKas.

Lukas (13:28):

Yeah, thank you Ben. Yeah, so we're just going to take some questions now, and I see Chad has already raised his hand. So Chad, go ahead.

Audience (13:38):

So I love all the developers who are on this call. All you guys are amazing, but I love when developers are like, as you can see, it's only a couple of simple lines of code. To my brain, it might as well be a million lines of code because I can't do this. I'm really good at telling scripters and developers what I want done, but I can't do it myself. So do you guys offer any type of service which we could say, hey, this is what I'm trying to do, can you help me make this happen and then incorporate that into RunScript?

Ben (14:20):

Yeah, well, the answer is yes. When I'm thinking back through some of the customers that we have onboarded, we did work pretty closely with some of them and we actually wrote their scripts for them, in some cases. Oftentimes they would come with an idea of what they wanted to do. They were fairly technical people, but not expert in InDesign scripting. And we were able to help them and work very closely with them at a low level to get them going on RunScript. And I think that that will continue to be the case for quite some time. We're more than happy to engage at a pretty low level and help you build your scripts.

Audience (15:08):

So one follow up question I have, for a period of time, Adobe had released a service called Project Aspen, which was basically online data merge. Could we do that using RunScript? Could I feed it a CSV file and say run a data, I mean, I'm sure via scripting, right, you guys can do anything, but could I feed RunScript a CSV file and say create a data merge of this, of all these records and then that be the output. That would be doable using RunScript?

Ben (15:54):

You might want to chime in here, Caleb, but my answer would be yes, absolutely. If it's possible with InDesign scripting, then it's absolutely possible with Typefi RunScript. We don't...

Caleb (16:10):

The answer is

Ben ([16:12](#)):

We don't have a canned option where it's just completely sort of foolproof and you uploading and say data merge. There would have to be a script written to manage it, but absolutely possible.

Audience ([16:28](#)):

Okay.

Caleb ([16:33](#)):

Yeah, I think that the key difference is that Aspen exposed a singular endpoint for InDesign and it had some additional validation and a UI around the CSV to make sure that it was properly structured, whereas RunScript exposes the full nature of InDesign scripting as an API. And so you provide the plumbing around that. But absolutely, if you want to create an Aspen like solution with RunScript, it's entirely doable.

Audience ([17:20](#)):

Cool.

Caleb ([17:25](#)):

Chad, to answer your question again, because you have access to InDesign scripting, any output that InDesign can produce is a viable output from RunScript. So if you want to produce an EPUB or you want to produce JPEGs or PNGs of every page, or you want to select a text frame and export an RTF, all that is possible. Ken.

Audience ([17:55](#)):

Awesome.

([17:58](#)):

I was wondering about fetching images. So I would want to use images that aren't static. So could I pull in an image from perhaps another AWS location or, that wouldn't be part of the InDesign script, would be, or would? It would be part of a pre-RunScript script, would it?

Ben ([18:23](#)):

Yeah, that's right. So the way to specify input files is with those pre-signed URLs and you pass them into RunScript and as you said, yes, something runs before the InDesign script to pull down all those images. So I'd be fairly confident in saying that as long as you have a valid URL, you should be able to get any of those assets and pull it down to where RunScript is running the job.

Audience ([18:59](#)):

So customise the script before it runs, probably.

Ben ([19:05](#)):

Yeah. Well, we've written that code, so all you need to do is pass in the URLs.

Audience ([19:11](#)):

Yeah, well, until I know the URLs, they're either not in the script or I run another script to put them somewhere predictable.

Ben ([19:18](#)):

Yeah, there is a job descriptor file that is used, so you would create that for each job. So that can be a whole array of custom URLs for a particular job, and it can be different for every job.

Audience ([19:38](#)):

Thank you.

Caleb ([19:40](#)):

I think one of the ways to think about this is that RunScript is a very powerful task in a potential larger sort of product that you can do other things in that product before you decide, okay, now I've collected all this information, I've collected all these pieces, now we want to run it through and actually produce the output. And so you have absolute freedom on what you do on the front end and then just tie in to RunScript as that backend provider on the final sort of piece. Like if you create a really wet, rich web front end that gives you sort of pseudo drag and drop to position things, you're capturing information from that front end that you can translate into coordinates that then feeds into your RunScript side of things to drive the realisation of that in the InDesign context.

Lukas ([20:47](#)):

I actually have a question. So in general, is it more suited to shorter documents, one page, two page, or could it do a longer document? What's the level of complexity that you could go to with something like this?

Caleb ([21:10](#)):

I would say the shorter answer is your wallet.

Lukas ([21:15](#)):

And your time to write these scripts, I guess.

Caleb ([21:18](#)):

Yeah, I mean, it's time to create the solution that gathers all that and then the time to compose it. There's no constraint in many ways. InDesign Server is InDesign. The largest file size that you can produce with InDesign Server is going to match whatever you can produce with InDesign. I mean, I know that I have created workflows that have run a 1500 page dictionary as a single task. So yeah, the sky's the limit as far as what you want to do with it. It's just a matter of, well, what do you want to do with it? Ahh Ken, Apple Script. Yeah, I mean the challenge here is this is running in a Windows infrastructure through Azure AWS, so you're basically stuck with ExtendScript as that solution.

Audience ([22:47](#)):

Got it.

Caleb ([22:53](#)):

Ben, do you want to address that Roland's question?

Ben ([22:56](#)):

Okay. So the question is, can you publish a few examples around with RunScript? Specifically how the InDesign document is then opened on the server? I had difficulties with the path of the InDesign document. I think that if I share my screen, again, you got to click the share button. If we're talking about the path of these documents, so if you look in the S3 bucket, the InDesign document, the input document is this eotm.pdf, no, sorry, eotm.indd. That's the input document. And if you look in the code, this is the little bit that is getting that input document. The way this works is that the Key is the name or the path to the document at S3 and Path is the path to the document at RunScript, like the local path. So here, there's no folders or anything like that. It's just get it out of the bucket and use it directly. And you can see this sort of thing here because we have a special font that travels with this InDesign file, and you can see that it's in the folder of S3, in the root folder at S3. But then when I save that down at RunScript, it has to be in the document fonts folder. So you can see how to manage the paths at the different sides. Does that answer your question or have I answered a completely different question?

([25:12](#)):

Oh, great. Another question here. Do we set up our own S3 bucket or is that provided by Typefi? You set up your own S3 bucket. With RunScript we just, think of RunScript as compute, and we don't want to do any storage management at all. We just provide the compute and you BYO storage, and it doesn't have to be S3. It can be any sort of cloud storage, Azure blob storage. I think there's many different options out there.

Caleb ([25:51](#)):

And just an extension of that, one other question was, does RunScript support Adobe fonts? And the answer is more complicated, but the underlying thing is that the fonts that you get through a subscription, either through Creative Cloud or through the Monotype fonts, those are not supported with InDesign Server because both services, Adobe and Monotype, they need to understand the user that's attached to activating those fonts. And there's no facility within InDesign Server to bind you to a user for those. And so for fonts, you can provide the actual font file in your S3 bucket or whatnot, and those are loaded through the document fonts. And this is how you as the sort of the owner of the licence for those fonts stay in compliance with your font licence. We don't have the font, it doesn't exist in our font except for when you are running your job. And in talking with many different lawyers and font foundries and so forth, that's in compliance. If you have a licence that says you can have this font deployed on say, 10 different desktops, well, if one of those desktops is temporarily used with RunScript, but you're not using it internally, then that's okay. That's part of the licencing support. So it's just a matter of that font is never installed on RunScript. It's never able to be used by anybody else. It's only used for the duration of your particular job.

([27:55](#)):

But if you have a font that you like that you activated through Adobe fonts or Monotype fonts, you would need to reach out to the original foundry to obtain an on-premise version of that font licence and font file. And I guess sort of related to that, because it's InDesign Server, you can do composition in any of the languages that InDesign supports. If you have the, say the US or international English installation of InDesign, there's certain features within InDesign that you don't have access to. You don't have the panel for creating a composite font or doing the right to left composition and all that sort of stuff. But all of those features are fully addressable through InDesign scripting. So what you compose through RunScript, if you want to do a book in Hebrew or you want to do it in Korean or Russian, all of that is supported.

Ben ([29:21](#)):

So there's another question here. Is there detailed documentation available in order to get started? So yes, there's a couple of GitHub repos that we'll share the link with where all the code, all the documents, the

This transcript was exported on Feb 13, 2025 - view latest version [here](#).

scripts, everything that you saw in this demo and others is totally available. I think Lukas just added some links there. Thanks Lukas.

Caleb ([29:49](#)):

And the other thing is, if you want to get started, you sign up and we'll give you \$50 in credit to help you get started. So there's no cost to initially start playing with it and exploring how RunScript might work for you. So that's a nice little freebie.

Lukas ([30:11](#)):

So I just put the link to the signup and there's a GitHub repo with some example code as well.

Ben ([30:22](#)):

And if you get stuck, then feel free to reach out. The team is more than happy to help get you going with RunScript.

Lukas ([30:35](#)):

Yeah. Cool. So we are a couple minutes over time. If there are any questions, please feel free to speak up, but if not, I think we will wrap it up.

Caleb ([30:50](#)):

All right. Thank you all. It's a pleasure to see you.

Lukas ([30:56](#)):

Alright, thanks for joining everybody.

Ben ([30:58](#)):

Yeah, thanks everyone.

Lukas ([30:59](#)):

Have a good one.