

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

So let's talk about how Typefi automates InDesign Server. What we're looking at here, this PDF that I have open right now inside Acrobat, this is a health insurance plan document, and this is a sort of a stereotypical type document in this class. And how we might automate this is very much built around the content itself. The content is going to drive how this is automated. So before we get into the actual sort of mechanics of how we automate it, I wanted to talk about, well, what do we see in this document? So the first thing we see is that there's a hierarchy. There's a visual hierarchy of this content from the understanding your plan of benefits to these other headings of where to find information about your specific plan. And in this sidebar box where we're looking for special information that's more important to the subscriber.

([01:01](#)):

And so there's a hierarchy that's created or implied in that structure. And as we move through the document, we can continue to see that hierarchy repeated with the different heading levels in that content. That sort of lends itself to start thinking about how we might break this content down into component parts. Now there are other things within this document when we get towards the end where we get into these sort of examples of, well, here's what your coverage might look like if you select plan A, B, or C. And so this is a different kind of page of information that we have. But again, we can start to think about how we can break this down into these components. And the way that we might approach that first off is in the actual design of the template itself. So when I look at the InDesign template, that matches up to that output that I was showing earlier.

([02:01](#)):

This template is set up with a series of prototype page designs. And so on the right hand side we can see my pages panel that shows I have sort of this opening page with a table of contents and information about the plan. I have internal pages that switch to the two column view. I have this sort of disclosure just as the opening page for this comparison of what your prices might be. And as it continues on, it flows to a different type of page. And within this document, if we think back to that sidebar object that we can start to create these objects within InDesign that become responsive. And so as we add or remove content to a frame, it resizes automatically and their behaviours attached to each of these things that sort of guide how the content flows around them or past them. So this is where we start thinking about, in the same way that our content is being sort of broken down into these fragments or components, we're also looking at the design itself as being very component based. Now the question comes, okay, how do I get my content into this InDesign layout in a fully automated fashion? So let's switch over into AEM. And right now I'm looking at AEM Guides, which is a component content management system for AEM. And what this gives us is this architecture to structure and join my content that is, here is a view of it, looking at it in Assets where it's a folder collection of files. But here within Guides I can see the relationships between all of those files. I can see the visual map of my DITA map that connects all these different elements together.

([04:09](#)):

And so if we think back to that opening page of my PDF where we have this grey box that that's going to correspond to this figure that we've defined within this DITA file. And this is a child to the "Understanding your plan benefits." And again, that matches up to this idea of sort of a heading level one to a heading level two in my content and how we start thinking about these sorts of structures within guides. We can start to tease out and break apart our content into discrete pieces. We can organise that, we can rearrange it and structure it in this simple DITA map that sort of provides almost like a bill of materials for how to build my content. And the nice thing about this is that some of these pieces might be reused across many different disclosures. And so this allows me to write it once and use it many times.

([05:14](#)):

So the next question then like, okay, how do I now tie this into InDesign? Because InDesign is not a normal sort of output preset for Adobe Experience Manager. And that's where Typefi comes in. We provide a connector. So if I look back at this content on the Assets view and I come down here to the bottom and I find my DITA map and I select it, in addition to the regular tool set we have across the top, there's a new Typefi button. And so if I engage that, this moves me into the Run Typefi Job interface. This is going to use the web services API that Typefi provides with our connector in AEM to package and resolve all of the con refs and links within that DITA map, pull all that content from all the different locations it might be, package it together, send it over to Typefi, on to InDesign Server, and I can choose what I want to get back.

[\(06:18\)](#):

Do I want just the PDF output or do I want to get the actual InDesign file itself that remains fully editable that I can then continue to modify. I can pick up additional files and combine it with it and build a more complex document in that sort of fashion. Now from a demo perspective, when I click Run here, this packages the file up, sends it over, but I can't see what's happening because InDesign itself is a black box. It's a headless server. So as an alternative to that, what I'm going to use is this equivalent workflow that's set up on a special version of Typefi that's running locally on my system. And so when that job is sent over, the first thing we do is we're going to pick up the fonts and make sure we have the fonts that are used for that design so that we maintain your look and feel and your brand compliance.

[\(07:15\)](#):

And then from there we actually start composing InDesign layout. And so we're going to pick up that template that I was showing earlier, and then we're going to run that against my local version of InDesign 2024 and then export out an interactive PDF. So the cool thing about this is when I run this and I switch over to InDesign, I can close out my template and very soon InDesign will be taken over by Typefi and we start to compose the layout. And you can see the decisions as it starts building out those boxes, laying things and moving things around and positioning them on the page, building the tables from the XML, and then we are done. So in about 20 seconds for a 24 page document, we can now see our final output. The nice thing about it also because I'm running it locally is it actually opens it for preview within Acrobat.

[\(08:12\)](#):

If we switch over to the job that I ran earlier from Adobe Experience Manager, we can see that well that completed as well. And it returned that InDesign layout and that PDF right back over to AEM. And this allows me to sort of integrate this into additional workflows so that if I want to now take this content that I've composed and built, I can pass it into a workflow for approval and review or even distribution. So this allows you to decentralise the composition or the layout of your content and that you still have the benefits though of the ease of use of building these templates visually within InDesign. And so if I want to rebrand something, I want to change the colour scheme, change the fonts. Ee get acquired, it becomes something else, that's very easy to take these templates and with a few clicks, I can completely re-skin it, check it back in, and now all the new content comes through, takes on that new look and feel. You can't get that kind of speed to market making those changes with other systems that are built around CSS and HTML or FO or other sorts of just raw PDF generators. By using InDesign and InDesign Server as our rendition engine, it gives us a tremendous amount of flexibility. Now the other thing about this is that I'm showing you this built around the integration that we have with Adobe Experience Manager, but there's nothing to stop you from integrating this with other solutions, whether it's Documentum or Alfresco or MarkLogic, anything.

[\(09:59\)](#):

The Typefi Server itself just provides you with a Restful API to drive the composition from whatever XML you might have. So it gives you a really powerful platform to build publications and enable your content creators to self-serve to publish on demand with all the look and feel that you want from your

documents. And we can also handle all the sort of complexities around mathematics and equations and custom rules that you might have because when we think about these workflows, we can start to extend and modify the workflows. We have a huge library of different actions that you can build and add into these workflows. And even within the action of say, Create InDesign Document, we can also trigger individual scripts that use the InDesign ExtendScript model to augment and go beyond the native capabilities of InDesign. And so that altogether is the way that we approach this automation around InDesign Server to try to make it as flexible and useful as possible. That it's how do we take these fragments, this components of content that you're using and whatever your CMS is, and marry them up and arrange them and lay them out in the best possible fashion with InDesign and give you something that is flexible and useful as the output. And that's why we say Do More.