Guy van der Kolk (00:00):

So before we start, I need you all to say a little prayer to the demo gods because I'll be showing off new stuff, some of it still in development. Yeah, I like living on the edge. Everything that I'll be showing you is all about enabling you to do more and about asking, okay, this is the thoughts that we have, but what do you need for this to be successful in your organisation if relevant? Now I am not a presentation designer, so this will be my second to last slide, but I'm the one that's lucky enough to be here with you in front of this room to show off some of this stuff. But ultimately I am not the one that is doing a lot of the actual work. So a lot of the people that are on the leadership team, you're seeing them here, but the crew that actually makes a lot of what we do possible is our colleagues in Sri Lanka who are not here today, but almost everything that I will be showing you today is thanks to their hard work. So I wanted to make sure that, show some appreciation for everybody that can't be here today, but that made a lot of what I will be able to demonstrate to you possible. Now, what will we be talking about today? A fifth thing was added, which I'll talk about first, but I'll be showing you a little bit of what work we've been doing in the 64-bit version of Typefi Writer. I'll be showing off a little, some parts of our web-based authoring integration with our partners Fonto and Oxygen. As Caleb announced earlier, I will be demonstrating a SharePoint integration that we've been working with where I'm really interested to hear your feedback. Can I get a little show of hands, is anybody using SharePoint in their organisation?

(02:09):

As I expected, about half the room. And some work that we've been doing with HTML5. So some very exciting things. Now all of the things that are on this slide are still in preview and are going to be coming out at some point in the future. Chandi nudged me a little bit on, when Caleb was talking about our partnership integration with RecoSoft to also show off our native export to DOCX because we're going to be obviously looking at what we can do with RecoSoft, but we have a native export to DOCX module as part of Typefi. So that's something that I wanted to demonstrate. A lot of work that we did here was done for one of our customers, ISO.

(02:57):

So this is an example of a PDF that has been generated using Typefi for ISO, and this is the equivalent, let me grab the right one. Nope, that's not the right one. This is the corresponding round-tripped DOCX file from that same input XML. So it's got all the stuff that you would expect. It's got a table of contents with all the clickable hyperlinks. Obviously it's got the heading base navigation on the left-hand side. It has MathML, which is, ISO uses MathML. So that gets converted into the native Microsoft equation editor, or if you're using MathType, that can be used as well. Tables are all taken care of. Let me actually switch to a different area. Here you see an example of some tables. Here you see an example of some figures that have been extracted out of the content and put into this.

(03:59):

So this is something that is possible by default with Typefi using the, for those of you that have access to it, and if this is something you're interested in. The Microsoft Word plug-in, let me actually zoom in a little bit so that you can see. You can see that the Microsoft Word plug-in has an Export to DOCX action, which takes CXML and turns it into DOCX natively. And then we also have a Modify DOCX action as part of that which uses transforms to take the DOCX file that was generated and modify it. So what's an example of a modification that was done in this case using the Modify DOCX action. That is, if you're looking at the Word file for example, you can see that we have different sections. So for example, if I go all the way back to the beginning, there is no heading, there is no...oh there was in this case.

(05:00):

But you can see that we've added stuff like headers and footers and things like that. Very specifically, ISO also wanted these to be double page spreads. So a left hand footer, right hand footer. this example doesn't have it, but ISO has tables that are, so we have modifications to the sections so that the tables are

horizontal instead of vertical for landscape tables. So those are all things that are possible. What obviously we'll be investigating with the partnership with RecoSoft is those things where customers who may want a DOCX file. Not so much for taking it back into Typefi, because that was one of the requirements from ISO was this DOCX file needed to be able to be sent to authors where they can do modifications. And then it needed to be taken back into eXtyles so that XML could be taken from it again with the minimum of effort. So it's more about making it easy to take this content back into a Typefi process and not so much about making the DOCX file look like the PDF, which is something that the RecoSoft solution might be, might be very good for, but we'll be doing a lot more testing on that integration in the future as Caleb said.

(06:22):

And if you want to know anything about, because this is like I said, part of the native functionality. So if you want to know anything about that, just reach out to your solutions consultant or me and we'll be happy to talk you through the possibilities. Alright, now we go into the stuff where we need the demo gods. Alright, so one of my areas of responsibility is Microsoft Word and the integration we have, which is called Typefi Writer. And one of the things that the team and I have been working on for one of our customers is index validation. So some of our customers write their indexes as part of the Word file and one of the things that Microsoft Word does very well is it's sort of if your index, people may manually modify index entries. And as one of the previous speakers said, whenever a human touches something, there's a high likelihood that we make mistakes.

(07:28):

But what Word does, it just covers up the mistakes. So then when you take the Word file and you try to transform it into XML, you've got to account for all of these variations, which is fine, but I'm a firm believer that where your content is concerned, it is best that your content is right. So how do you know that your index, if Word covers it up, how do you know that your index entries aren't right? Well the team and I based on some customer feedback, because pretty much everything we do is based on your feedback from the day-to-day use, is how do we fix that? As I said, this is a pre-release, so it's going to come up with an error in a moment, but that is, we'll just ignore that. It'll be fixed before the release. We have added a new index entries, let me zoom in because I don't think you can see that very clearly.

(08:18):

We've added a new Index Entries Validation to Microsoft Writer in an upcoming version that should hopefully be released between now and the end of the year, where we, as the name implies, we look at the index entries in the Word file, that was the error, and we tell you if there's something structurally wrong with the index. Now looking at this, you might think that somebody did some horrible work coding the index entries, but this is a testing file where we put all of the possible variations in. And one of the things that I'm very excited about, obviously the index entries helps. So here we've got an example of where it's more of an informative thing where if you're going to an InDesign workflow, you can only have three sub-entries. InDesign only supports three sub-entries. So this is more of a informational message, but the thing that one of the pieces of feedback we've been getting consistently from customers over the years is, the warnings and the messages. Like how do you, making it clearer how to solve the problem.

(09:26):

So that is something that we are going to be putting a lot of effort into the next period, but this is an example of one of the first places where we implemented that. So as you can see, you've got the warning message at the bottom and it has a direct link, thanks to my wonderful colleague Toni. It's still a draught so we're working on it because it's not ready to be released yet. But you will now get the ability in the future as we deploy this to more and more warning messages, when you're using Typefi Writer, you will be able to get A, the warning but B, be linked directly to an article that will provide you with steps on how you can solve it yourself. Which is always something that is important. Solving it yourself is not needing to create a help desk ticket for it, which always takes you out of your workflow is better than. So

this is something that's going to be happening more and more in the interface as we clone Toni and get more documentation out of that.

(10:38):

Now there is one thing. Are there Typefi Writer users in the room? There's one. Alright, do you happen to know if you're using 64-bit already in your organisation? Okay, it doesn't matter. One of the things Microsoft announced that they were moving from 32-bit to 64-bit natively about three years ago. And now we are going to, and obviously this is going to be communicated with a proper email to everyone in the near future, but Microsoft Word in 32-bit will remain available to be downloaded, but it will be unsupported. So we will not be doing any more bug fixes. We're seeing that the adoption rate is almost, nobody uses 32-bit anymore and not needing to worry about fixing bugs for it means that we can spend more time in getting you the stuff that you need. So that is a little bit about what's coming in Microsoft Word in the near future.

(11:42):

Alright. Alright. Next thing. So one of the things that we're noticing is that no matter what the kind of content you produce, Word is generally still ubiquitous. It's the place where a lot of people start writing whatever they need to write, but a lot of our customers obviously use XML. And one of the things that we really wanted to do was we wanted to, with this partnerships with Fonto and Syncro Soft's Oxygen, we wanted to investigate what could be possible natively within Typefi Server. Because a lot of our XML-based customers also store their files on the Typefi Server and keep their XML files there so they are able to work with them relatively easily. One of the things that we have chosen to do, and I'll be demonstrating a little bit of both of them, is we decided to make it possible to work with either Oxygen or Fonto—why both?

(12:45):

I'll explain that a little bit more in a minute—natively from Typefi Server. So I'm going to start with Oxygen just because I think Oxygen is a lot more familiar to the people in this room. A lot of us work with Oxygen on a day-to-day basis. So it was a logical choice for that kind of person. The person that's already familiar with Oxygen regularly, and that is just looking to make quick edits. So the entire purpose of this was not so much to do an entire authoring phase like creating, in this case I'm showing it with DITA, but it can be customised for JATS, BITS, whatever XML flavour that you need. We wanted it to be so that, okay, you're storing your files on the XML, on the Typefi Server. And let me double check, I don't remember, I think I clicked Oxygen integration, but I just. Yes, now I'm sure.

(13:36):

So let's say that we're going to, at the last minute there is a need to make a change to an item on the cover. So you don't need to download the XML file from the Typefi Server, make your changes and then upload it back with the. No. Right now, when you enable that feature or when that feature is enabled, you can select, in this case your DITA file, you can click edit and that will open it up in Oxygen and Oxygen Web Author. And at that point you will be able to make your modifications. Now I am not by any means an Oxygen Web Author online user expert, so I'm not going to go too much into that. But you see the interface of Oxygen Web Author, and what we decided to do was take it one step further so you can go in here and make your modifications and insert new images and do whatever you need to do, in this case in a DITA file as part of a DITA map.

(14:34):

But we also added integrations so that, let me zoom in a little bit at the top left, so that you can attach workflows. So that you can run your job directly from either Oxygen or Fonto web author, directly from the interface. Obviously you could also go ahead and make your modification, close it, and run your workflow natively. That obviously depends on your preferences, but we wanted to make sure you had the option to run the job directly from here without needing to make any changes. And you can see that we've

added the Workflow button and the Publish button. So what I'm going to do is I'm going to close this one and briefly address the, show you the same file, but then opening it up in our Fonto integration. And just to give you a brief, why did we choose both Oxygen and Fonto? Well, because they are in, at least what we think, they're addressing two different customer demographics.

(15:41):

Oxygen Web Author is for those people, a lot of us in this room, that are already familiar with an XML editor. But Fonto is much more an authoring environment that feels familiar to Word. So that means for people who are maybe not as familiar with XML as you and I are, it's a much, it's a friendlier environment. So you have choices depending on the requirements that you have. I've already prepped this so that the workflow is already attached. So how would it work? You would go ahead and cover tests, and I would go ahead and click the Publish button. This is a pretty quick test. So in a moment, if the internet continues to work, which it has been doing so far, this job is now being sent to the Typefi Server and in a moment the folder will go green and then we'll be able to go into the folder and get the job.

(16:39):

One of the things we've done is we've kept this very simple because we want to know from you, if you choose to use this, what you expect with these PDFs, what you expect with the output, are you happy with just closing this? That's the kind of thing, the kind of questions obviously that we want to address. So if you click view job details, what's going to happen is it takes you to the Typefi Server job queue where you can see your job and when it's done, you'll be able to download your PDF from here. The internet is a little bit slower than it was was yesterday. Like I said, by default this works with DITA but it can be customised for JATS or BITS or whatever kind of flavour you have. So if you see this and you're interested, by all means reach out and ask any questions. Are there any questions about this at this time? It's better to, if there are, to just address them now then to get back later.

(17:42):

No. Awesome. Alright. Oh yeah, it's done. There we go. So the job is completed and if I wasn't already opening it up, I could go ahead and look at the generated PDF directly from the authoring environment. Alright, cool. Going to close this off. I'm going to close this off. I don't need that anymore. SharePoint. So a customer reached out to us and they said, we like Typefi very much. It helps us produce a lot of things a lot faster, but we use SharePoint. And every single time now we have to go and either go to the job folder or if we're using Typefi Writer, we have to download the file and then move it somewhere else. So could you do something more intelligent than that? And we're like, huh, let's see what we can do. So the team and I did some investigations, and we took a look and Microsoft has a very expansive API solution related to SharePoint.

(19:02):

So there's a lot of things that are possible in that regard. What we went ahead and did is, I shouldn't need Word anymore. Let me close this all up and get it out of the way for you. I have a folder here. This is an XML-based workflow right now. We're still looking and that's one of the things that we really, especially for those that are using SharePoint, I really want to pick your brain on how you would, especially if you're using and Writer and SharePoint, how you would see that integration working. But for now, this is an XML-based customer. So a lot of things are like file paths and things like that are already natively in the XML. So there's a folder here and right now if I go into that folder structure, I want you to observe that there's only images in here. So this is one of our native files and I'm going to go back to the browser and I am actually going to make sure that I'm in the right folder so that you can see it happen. And I'm going to open a demonstration file. So I'm going to run the workflow right away and then I'll talk a little bit about what we did. It takes a couple of minutes to run and for the things to magically appear. So let's go ahead and run that workflow.

(20:24):

So that's one of the first things. SharePoint, it's a Microsoft integration. So what do I need to do? Well, I need to log in with my Microsoft credentials. Right now, in the version of Server where this is going to be a part of, it's a two step process because as Caleb said, we're still working on Auth0. With Auth0, this is going to be a seamless single-sign-on process where you log into Typefi and if you have corresponding Microsoft credentials that will, I don't know how the magic works, but it talks to each other and then it just works. So right now it's telling me that I've been logged out of my SharePoint integration, so I'm going to, which means that I won't have access to the SharePoint shared libraries where I need to work with. So I'm going to go up here and disconnect from SharePoint, and reconnect, and let's see if it asks me for credentials. Alright, so this is giving me a list of all the SharePoint share libraries that I have access to that I could potentially interact with. And then I'm going to go back to the SharePoint demo and run the workflow. Let me see.

(21:38):

There we go. So it's running and then I'll go back and talk about what we did. So the SharePoint integration involves two steps, an update to Microsoft Server to allow you, sorry, to Typefi Server to allow you to connect to your Microsoft SharePoint, I forget the technical term, thingy, to be able to know what kind of credentials you have access to or what SharePoint libraries you have access to. And then we've created two actions at this point where you can, so one is Copy from SharePoint. Let me zoom in on that a little bit. So the Copy from SharePoint action allows you to define what shared library you're going to be working with. In this case, I've defined the public shared library. And it allows you obviously to define the input file because that's something that's universal to Typefi, and it allows you to define what the destination is.

(22:29):

So in this case, what's happening is that images folder that I just showed you on my local desktop that's stored on SharePoint, is going to be copied into the job folder. So that's available to InDesign for composition. So that's one part of the process. But the other part of the process, which is actually the thing that I still find magical to this day, is at the end we have a Write to SharePoint action. Right now that uses a properties file with, a properties text file, because that allows you to control some things while we gather feedback from you on how do you use this, how would you use this so that it's easier to configure using the buttons. And you can see that I'm writing to the shared library. So let me take a quick, the job should be finished. Okay, it's still running a little bit. Let me actually show you how that, SharePoint, it goes a little bit into the details, but most of us are technical.

(23:27):

So here's an example of that SharePoint property. So you can see that I've defined this to download the PDF, the CXML, and the generated InDesign files from the job folder. It's taking some, so it's automatically going to be creating the folders. And in the case of the CXML, you can see it also has a variable. Let me zoom in again because that's a little bit, you can see that it has a variable to only copy the last file, right? In the future we're probably going to do this with UI options so that it's easier for you to select this, but for this first iteration, the customer was like, we want to get this to work. And that allows us some time to talk to other customers and say, how would you use this? So let me see if the job is done. It is still running. I think it's the copying back and forth that is a little bit slower on the, ah, it's done. Alright, so the magic. Fingers crossed, demo gods.

(24:36):

There we go. The CXML, InDesign, and PDF files were not there before and now they are. So there we have our generated PDF file that was copied from the Typefi job folder into your SharePoint shared library without any interaction from a user, which I think is very exciting. If you are using SharePoint, please, whether it's now or if you want to set up a call with me separately after you've maybe talked about it internally, because I really want to know how we can expand this to other SharePoint customers. Alright, that's SharePoint. Any questions about that?

(25:21):

If not, you know where to find me later. Alright, cool. So one of the pieces of feedback that we did gather was, when I demonstrated this later, which I thought was, the last time, which I thought was really good, is, okay, this is all well and good, but now you're copying all those images every single time. Can you configure it so that you have a storage space on the Typefi Server for your images and that we're just copying the images that we need? Anything that's changed. I don't know, we'll have to add it to the backlog. It sounds like something that should theoretically be possible in the same way, because that would prevent a lot of back and forth. And one of the things with InDesign Server is it requires the images to be in the same location as where InDesign Server runs. It can't grab it from other locations, especially on the cloud. Makes sense. Because anyway, that is the SharePoint integration. The last thing, and I think it's very fitting, especially talking about the HTML that, I'm trying to find them, from IGI group earlier. Typefi has been able to export HTML natively for years. I've been part of the team that takes care of that, but somehow we were never able to make it compelling. You can export HTML, fine, but then what do you do with it? So the team and I did a little bit of work on that over the last few months in preparation for this conference. And pretty much everything that I'm demonstrating here today, and that's also going to be obviously shared online and talking with other customers, is all about this idea of, okay, I have the thought that this is important, but I need to know from you: where do we take this? What is missing? What else can we do? So I've got my, we call it the Marketing Monkey file internally, it's just a simple file, headings, just some things to play around with, some images, as a thing to, like I said, to play around with. But one of our customers asked us, we produce similar content to your Marketing Monkey file, how about making it so that we can take that article and publish it to the web, have a little table of contents and things like that, and make it easy for us so we don't need to worry about that.

(28:01):

And they gave us a little example of what they would like to achieve and it's like, huh, that sounds like something we can do. So let's build it and see what's possible. So I have a Word file because in this case that's the place where I'm starting from, and it's called the Sustainable Development Goals. And what you'll probably observe is that one of the things that I'm still working on is, how would the covers work? Like a cover on the PDF, but on the HTML, you're not going to have a PDF A4 sized cover. So that's one of the things where we're looking at, how would you do that? You'd probably have a different section or a different graphic just for your HTML, and you'd probably use conditions like you're seeing me use here. In this case I've made it so that you can see that the section cover and the chapter opener image have a condition of not in HTML. So that means that, I'm for now ignoring them while we figure out what we do with that, as customers may want to use this. So I'm going to publish this, and right from Microsoft Word in this case, and I'm going to choose a different workflow. I'm going to choose my HTML workflow and I'm going to click Publish. This should be pretty quick because there's no PDF involved in this case. And then let me open up the website so that you can have a peek of what is going on.

(29:34):

It is running, so it should appear anytime. Now one of the things, while this is going to be running shortly, if all goes well, one of the things that we needed to do was have heading structures. Because headings is one of the most important tags in, okay, we could argue about that over either lunch or a cocktail tonight. But the heading structure is one of the most important things about making your content visible, because no matter how you render something, if something is tagged as an H1, H2, H3, H4, H5, H6, any browser in the world is going to have native support for showing one heading bigger than the other even without CSS. So for your end users, having that basic H1, H2, H3 tag structure is vital for making it easy to read.

(30:33):

So that's one of the things that we needed to do. But in CXML, that was not easily doable. We didn't have that tag. We just had P tags and then usually we call it "type equals heading one" or that kind of thing. You're not going to be surprised that that's going to change. We are going to, in a future version of

CXML, currently the version is 3.2. So 3.3, which is going to come out sometime at the same time as the HTML. You can now see that not only, let me zoom in again, lemme put this in.

(31:11):

Not only do we have P tags, but we now have H tags as well. So we've got H1 through H6, we're not doing H because H is, H1 through H6. Those are the six levels of support in HTML, so it makes sense to support them in CXML as well. So the way this is going to work beyond HTML is that it's going to work in Typefi Writer. If you're building your templates, anything that's H1 in the CXML, if you generate a DOCX file from that, it's going to automatically apply the proper heading styles in Word from your template to InDesign. If you've got it in InDesign defined as a heading style, that's all going to override anyway. It's going to be very smart once we're done with all the testing. But for now, for this demonstration, I did a manual transform so that the heading one tags are mapped to H1.

(32:12):

The job is finished. So I'm going to go ahead and go into the job and I going to download, I'm going to select the CSS file, the JavaScript, the links, and the HTML. Manual process right now, but pre-release and all that. I'm going to download it. That should be pretty short. Well, 22 megabytes. What do you mean, 22 megabytes? We haven't implemented shrinking images and all that stuff yet, so that's, obviously we can give you 22 megabytes worth of images, but I felt the images was not the most important part of this demonstration. So does take a little bit of time to download though, that's why you want to shrink your images.

(33:03):

While that's downloading, let me take a step back and actually show you the workflow for this. So what, this is still work in progress, so the UI might change, but the most important thing is that we've got a table of contents area where you can select which headings you want to make appear. And we will automatically look at H1 through H6. And if you only select that you want your H1 through H3 to appear, we will automatically parse that from the CXML and only show those levels of headings in the HTML. And then, it should be downloaded.

(33:46):

Is it? Yes, it's downloaded. So let me open that up. And so one of the things that obviously we are definitely looking for feedback from you. Let me extract it first because otherwise Edge can't open it. Here we have an article. Please don't rate me on my CSS and design skills, not why I'm here. You can make it look however you want. But the basic functionality is that out of the box we have the table of contents. Very basic table of contents. We've got the ability to go back to the top if that's something that you want. You can hide the table of contents or show it if you want. And the other thing obviously that we built into it is, if my track pad will work, let me actually go out of full screen mode. If my track pad will work, is obviously a mobile version where the table of contents gets hidden at the bottom and you can pick from it.

(34:56):

One of the things that bugs me obviously is that the table of contents is serif and the other entries are still sans serif. I'm like, that's not that important for the demonstration. The thing is that I wanted to have something that's a conversation starter with customers around HTML to say, this is something, you're producing articles, this is something we can support natively, and then obviously it can be expanded to do whatever else you may want. Okay. That's it from me. Thank you so much for your attention and I'm looking forward to hearing from you how you want to use some of this stuff in your workflows. Thank you so much.