MARCO MIRANDOLA: So good morning, everyone. Thanks for joining us in this session. For the next 90 minutes, we're going to talk about data standard. Before we get started, just raise your hand quickly. Who has never seen data standard-- is new to data standard? All right. OK, good.

Actually, the topic is about what's new about data standard. Which, I suppose you know what is data standard, but I thought that we're going to see a lot of people here, so I have a little introduction about data standard to give an overview first. So sorry for the others, but I will try to make it short and still hopefully can provide all the information for you. And then we're going to go into the topic about what's new in 2017.

My name is Marco, Marco Mirandola. I'm co-founder and CEO of coolOrange. We have a long history with data standard. Data standard is a product that we developed years ago under the name MyView. And then three years ago got acquired by AutoDesk. And now it's part of Vault.

So let's go and cover quickly again. So as I mentioned before, we're going to have a very brief introduction about data standard, so I'm not willing to bore anyone, but just to bring all the people on the same level. We're then going to see what is really new in the data standard 2017. And we're also going to see what's changed in that data standard 2017.

Now with data standard, one of the topic-- and I will emphasize this a little bit later-- is about the compatibility. So the ability to update the configuration of customization that you made in the context of data standard, hopefully without effort. With 2017, there have been some changes that affect the compatibility. It's not too dangerous, not too complex. I'm going to cover that. So that you know when you move from 2015, 2016, to 2017, what you have to do and what you have to pay attention for. and I can also explain why these changes have been introduced.

The session is about 90 minutes. Therefore, we will definitely have enough time for Q&A, which for the other sessions the other years were a little bit short. In the presentation, I have actually a lot of little animations, this sort of video-- it's actually animated gif-- which are going in loops, so don't be surprised if you missed one point, you just have to wait another 10-15 seconds, and then it's coming over again. And you can see the word flow again.

This is just intended to save a little bit of time during the presentation. I have a life environment here, so during the Q&A we can also go on the life environment, and then check the topics again in more depth. Last point is then, I summarize a little bit the sources where you can get more information. After this session where you can get help. On the AutoDesk pages, on the forum, and so on and so forth. And by that, I hope that we covered basically all the topics.

So let's go into a brief introduction about data standard. So data standard is actually a product. What it means, it has a setup, you can download it. Actually, you can download, so it's been around since 2014. I think the 2014 version can be downloaded just under the subscription center-- subscription portal. I'm not sure about 2015, but I think the 2016 and 2017 are available publicly if you just Google Vault Data Standard 2017, you probably find quickly the link where you can download the thing.

It's a free AutoDesk Vault extension. So it's no cost, but it's not coming with the Vault setup, and you have to install it separately. I will talk to this even later, but basically you have three setups. One for Vault, one for AutoCAD, and one for Inventor. I'll explain why, but you basically have these three setups, and you have to install them separately. Or basically, you can install that piece of data standard that you're looking for. I will show you why in a few seconds.

It works with Vault Workgroup and Vault Professional. So no Vault Basic. You need a Vault Workgroup and a Vault Professional. And from 2014 onward.

The basic purpose of data standard is to help you enter data in vaults of metadata in a more standardized way and help you maintain the data. An example, you have in Vault set up a couple of properties. You can configure the properties to be obligatory. So that before you're going to release a file, the engineer has to fill these properties. But there is not much support in entering these properties. You have in Vault the Property Editor, that's fine. But, for instance, within Inventor, there is not much of this stuff. Vault Data Standard provides a set of dialogues that you will see in a few seconds. It helps you, basically, enter in a more comfortable, standard way, the data. And the database that you will see, so what you will see is basically the out-of-the-box, the standard 2017 dialogue. But it can also be configured and customized. And I will talk about this in a few minutes.

So this is, for instance, the dialogue for creating a new folder. So if you go into Vault, you can create a new folder. You have usually this little dialogue where you can basically just enter the folder name. You can select the category and enter the folder name. That's nothing more. But if you install Data Standard, this is the dialogue that you get. Now in this dialogue, you can select the category, the [? numerance ?] key, you can set the data. But more important, on the right-hand side-- you will see it again, it's going into loop-- on the right-hand side of this dialogue, you see these sets of properties. And if you pay attention, as soon as I switch to category, here to product, you see some properties on the right-hand side. If you go, for instance, to market, you see auto properties.

So this is an important concept because what data standard does is reflecting the configuration of your Vault. There is nothing you have to do. You install data standard-- so this is just really setup, installed, and you get this dialogue. And immediately, it recognizes, basically, which categories you have configured-- in this case folder, folder. We will see it for the fun in a minute. It recognizes which categories you have, and it recognizes which user-defined properties you have configured for this category.

So by selecting the correct category, you get immediately the properties that belongs to that category, and you can fill them up. Here we have also all the concepts of Vault, so if a property is obligatory, you have to fill it up. Otherwise, you cannot move on. With the OK button, you will see it in the next little animation. So this is the concept.

So with the dialogues of data standard, you can create form, and you can't create it even without data standard. But with the data standard you have this dialogue for creating a little bit more comfortable way to folders. And, of course, even the editing is using the same dialogue. So you can say Edit Folder, and you get this dialogue again, with all the properties at once that belongs to you One topic about configuration-- the sequence of the properties that you see here, so the order, is configurable as well.

So there is a bunch of stuff you can configure, and I'm emphasizing the word configure, not customize. It's really just about setting up, maybe, a little bit some XML files, and so on, and you're set. But again, this data comes immediately without additional effort.

Now, on the far side, this is brand new because usually you can not create a file from within Vault. So if you want to create a file in Vault, you have to go through the application. You have to fire up AutoCAD, Inventor, Word, you name it. And then you created the file with the [INAUDIBLE] tool, and then you check in the [INAUDIBLE] into Vault.

Now, with data standard, you get this ability to create files from within Vault natively. Now, for the CAD site, you can create CAD files as well, but it doesn't really make a lot of sense. So I don't think that people are going to create the AutoCAD file or the Inventor file from within Vault and then starting from Vault. But especially for the Office document, that might be interesting. If you pay attention on this dialogue, you have, again, a category. But the more important thing is the template. The third from the top. You can basically select a template, and as soon as the dialogue closes-- we'll just have to wait a few seconds-- you will notice here on the left-hand side a folder called Templates.

Let's just wait a second. Do you see here the templates folder? So in the templates folder, you can basically place all your Word, Excel, PowerPoint, AutoCAD, Inventor templates. You name it. And then from within the dialogue, by selecting the document type-- so out of Office document, Inventor, AutoCAD, and so on-- you can then pick the template that belongs to that type of templates.

Now even this part, this is actually something that has been changed a little bit in 2017, so we'll talk about it. But there is a configuration file behind that allows you to define even where the templates should be. So by default, you need this folder called Templates, but it doesn't have to be called Templates, and it doesn't have to be there. It could be somewhere else inside in Vault. It's configurable. I'm going to show you later where you can configure this stuff.

Otherwise, the concept is the same. So you select the category. You select the numbering scheme if you have one. You select then the template that we want to use. You're going to fill up the properties that you need. And maybe you have seen here that the title here had a red border. If you open up the dialogue again. You see that this red border, which indicates that this property is obligatory, and if you see the OK button is disabled until you're going to enter the valid data.

So again, data standard is about helping the people to enter the data in a little bit more comfortable way. And also, make sure that all the data that has to be entered are going to be entered right away. And it supports you with combo boxes, selection lists, all the things.

Even properties on the right-hand side, as usual, to find properties. If you have them configured as a date in Vault, then you get a date picker. If you have configured them as a selection list, then you get the selection list. So there is nothing more you have to do. Really just configure your Vault-- and this is really the key thing-- configure your Vault, create the category, define the properties, and data standard goes along with those configurations without additional effort.

Now let's move on with the custom object. How many of you are actually using custom objects? Just a few. Not that many. OK. But you know what-- who doesn't know what custom object is? OK, just a few. So custom object is basically an additional custom object. No, it's an additional object that you can configure inside in Vault. Where you can basically store information, which are not item, files, folder, change order, and so on.

So let's say you want to manage transmittals. I think we have-- yeah, we have here, for instance, this example with transmittals. You have to collect a bunch of files and send them out to someone, and you want to track when the file has been sent out, and which file has been sent out, and so on. You can handle it with your Excel file if you want, or you can just configure inside in Vault a new custom object called Transmittal, and it just becomes an additional table-- an additional list where you can basically store information, linked them with our stuff, and so on.

And you can create a bunch of it. You can manage tasks if you want. you can manage birthdays of your colleagues, whatever makes sense to you. It's basically an additional placeholder, or bucket, or table, where you can basically store specific information. In this example, we have this example with this custom object. And here, again, we have-- so if you work with custom object when you create one, by default, you get, again, a little dialogue that looks like the dialogue for the folder creation. So it's really just one line where you can just enter the name or the idea of the customer object. And then you have to go through the Property panel of Vault for filling up additional properties.

As soon as you install data standard, you get this additional dialogue, which collects all the properties that belongs, in this case, to the category transmittal. So it's way easier to create custom objects, edit custom objects, and so on. And again, it comes for free. It's an AutoDesk tool. You can install it without additional cost. It's developed by AutoDesk, supported by AutoDesk. So it's really a Vault product. It's a Vault add-on that you have a free, and I think it's worth giving a try. So you have this addition, even on custom objects.

Now, additionally, what you get with the stand-up of data standard are additional taps. So for instance, if you click on a folder, you get this additional data sheet tap, which summarizes, basically again, like the dialogue, all the information at once. Yeah, even this is configurable, and so on. You can define which properties should go up and which should go down. So you can configure it the way that it's more appropriate for you.

So everywhere you have this data sheet, you have it for folder, for file, for items, for change order for custom object. This data sheet comes with all of the types of elements. On the file, you see this data sheet here, the last one, the data sheet tab. But additionally, you also have this CAD BOM sheet. And this one, is showing you, basically, the BOM that is inside the CAD file. So it works with AutoCAD, Mechanical, Inventor, and all the other applications. Actually, I'm not sure. AutoCAD Electrical? No, I'm not sure.

So the point is this, When you check in a file, a CAD file with the add-in-- could be AutoCAD Mechanical, Inventor, I think SolidWorks ProE and Solids it's the same-- when you check in that file, then with the checking process, the CAD BOM of the file is also checked in. So behind the record, in this case, the blue line here-- the record of this assembly-- we don't have just the file. The BOM of the file is also stored already in Vault.

And for those that are using Vault Professional, when you do an Assign Item, when you say right-click Assign Item, then this dialogue basically reads out the BOM from the CAD file. It gives you a preview in order to create the items. Well actually, it's not reading the BOM from the CAD file. But it's reading the BOM that during the last check-in has been stored already. In Vault. So the CAD BOM is already in the Vault database. And we took, basically, this advantage to say, well, if the CAD BOM is there, why not showing it? This works with Vault Workgroup as well. So even if you don't work with items, even with Vault Professional. So if you don't work with items, either with Vault Professional, Vault Workgroup, you still have this dialogue which shows you-- which give you a preview about the BOM.

AUDIENCE: You can modify the item number?

MARCO MIRANDOLA: No, so this is read-only. This is read-only. The only way to change the content here is to open up the file in AutoCAD in Inventor or make the change and recheck it in. Yes, so this is really just read-only. It's for previewing only.

It's an interesting thing because if you think about, you have this information, so faking that this is the standard, with a little bit of customization-- not too much, a couple of lines of code-- you could, for instance exports this bill of material as an Excel file or a CSV, and hand it over to the [INAUDIBLE] system. So the fact that you have this information already involved, it's a cool thing, and it can be useful for other scenarios.

So the next one is, with the items, again you have the data sheets tabbed, the last one, and additionally have this associated file. We have to list all of all the files associated to this item. And, again, even on the custom object, you have this additional tab that summarizes all information, and so on. So these are the things that basically comes out of the box without effort, just install data standard, and you have them. You don't have to do anything. You just configure correctly your Vault, categories, numbering schemes, properties, and so on. If you do that-- and it's something you should do anyway-- but if you do that correctly, the data standard will basically just follow the rules and help you even more to enforce the rules and the configurations that you have applied in Vault.

Now, we have also data standard inside in CAD. In this case, you see AutoCAD Mechanical. Well actually, it's also AutoCAD Vanilla, so it's AutoCAD and Inventor. And in both, the very first time that you're going to save the file, instead of becoming the usual save dialogue of AutoCAD and Inventor, you basically get this additional dialogue here. It's the same logic. You select the category based on the category you get. A bunch of properties that you can fill up right here from within AutoCAD and Inventor. You can select the numbering scheme. One additional thing is, if you give a look on the top of this dialogue, you have the ability to select the folder where this file should go in. And this is one of the additional concepts that you can take away from data standard. so without data standard, it's anarchy. Everyone can check-in a file where he thinks it's appropriate. And by just creating, locally, a folder-- I can create, locally, a folder called Marco. Save the file there. And the check-in dialogue says, hey, a new folder. It creates the new folder inside in Vault.

Now, for some cultures this is OK. For all the cultures, especially so the German ones, this is very, very bad. You need to have a rule, and it has to be an ISO 9000, whatever. But anyway, no, it's not a joke. I mean, the idea behind these, for instance this older picker, it's actually not the folders that you have locally. It's the folders that you have inside in Vault. So that means, that if you want to create a new folder, you have first to create the folder inside in Vault. Not everyone is allowed to do it. And so you can, and then you can use it right away.

Now again, this is also configurable and customizable, so it could be a mix. It could be a mix between Vault folders and local folders. We have seen some companies doing-- at the root of the folder is, for instance, set by the company rules. So you can just pick folders up to this level from Vault. But then underneath that folder, you have the ability to create your own stuff. So in order to keep a little bit of structure in the folder tree and still provide the freedom to the engineers to create their own stuff underneath that. So it's really highly flexible. You can set it up as you think it's appropriate.

But again, the idea is to make sure that all of people in the companies are working toward one system, and instead of creating chaos inside in the system just because it's Friday morning or

Monday morning whatever, the data remains consistent and becomes easily more manageable. Even for the future.

So both Inventor and AutoCAD. Now in the case of AutoCAD, it's pretty simple. It's one file. And it basically has this one dialogue. By the way, here in this case, in the CAD, of course, everything you do here is connected to the title block. And in the case of AutoCAD, the data standard wins. Yes.

AUDIENCE: How does this work with frame generator where, say, you might have 1,500 components?

MARCO MIRANDOLA: Right. So the question is how does this works with frame generator. It's a good point. So this is why. So on the AutoCAD side, one file, pretty simple. You have a title block. Whatever you enter here is going into the title block. That's it. You just have to name the title block and that's it. If you make changes on the title block in AutoCAD via double clicking the title block, everything you do is going to be lost the next time you save because data standard wins.

Therefore, in the data standard menu, you see that there is a top-- on the top there is somewhere data standard just ribbon. You have the ability to read-- to edit again. And so, go again into this dialogue, make the changes. And again, the changes that reflected. Now the question is about frame generator, design accelerator, every sort of additional Inventor tool that generates a bunch of files at once. So in the case of Inventor, for these kind of things, you don't see it here. But there is another dialogue, which looks similar, and you basically have to list of all involved files on the left-hand side, and then on the right-hand side you have the properties. So that case, if you use design accelerator, frame generator or whatever, the moment that you generate the thing-- you basically generate at once a bunch of files-- and then when you save it, it's handled as one-save process. And you basically see the phantom components underneath the real components. You can make a difference between phantom and the other components but within one dialogue, you basically handle all the files at once. So that's supported.

I think that's it for the usage. Now, I mentioned that data standard is a product, but in reality, it's also a customization framework. So the idea behind is this, since, I think, Vault 2010, or whatever, they introduced an API, and with the API, you could do a lot of stuff with Vault. So adding additional tabs, create dialogues. You had the ability to do really a lot of interesting things. However, visibility was limited to developers. So raise your hand. How many of you are .net developers here? OK, a handful.

How many of you have ever made any sort of script files in the past? Lisp, or PowerShell, or VBA, or iLogic or this kind of stuff. So the half of the room. And this is the difference. So when Microsoft introduced .net, everything becomes easier. Yes, for developers. Those of you that were able, in the past, to do a little bit of cool thing-- and you were proud of it, right? When it was working-- got completely lost, thanks to Microsoft. Well, no. It's not really true. I mean, they actually went back because they introduced PowerShell a couple of years ago for the IT folks. And PowerShell is the new batch-scripting, if you want. But in reality, it's built on top of the .net, so it's super powerful. You can do basically everything you do with .net, but it's a script language. It's way easier to consume. It has been developed for IT people, not for developers.

And actually recently, Microsoft made it open source, and now PowerShell is even available for Auto platforms, like Mac, and Unix, and so on, and so forth. So it's pretty interesting what they're doing now. Anyway, so why is PowerShell relevant here because data standard is built on top of PowerShell. So creating additional tabs, modifying the dialogues, creating additional menu items-- and I will show you a little bit of this-- is basically all done with PowerShell. And maybe, if you meanwhile don't feel comfortable, again, to create your own script, I bet that if you read one of these you can follow. You can understand the flow and at least understand this is the line that is responsible for creating the folder, or doing the property mapping, or whatever it is. So the idea of data standard is to lower the complexity bar for the non-developers and allow them-- those that want to do the stuff themselves, or at least give them the ability to follow what the heck is going on. And maybe make a little bit of tweaks and changes, and so on and so forth.

So the idea is basically to make customization simpler, more accessible even to non-developers, and compatible. Compatibility is a big word because compatibility requires a lot of rules before and after. So I would say 99%-- give me 1% of excuse when something is not going to be 100% compatible-- but it works quite well.

Now, let's give a look. So let's suppose you want to create a new menu item inside in Vault. There is this configuration file, it's called many definition of XML, and this is actually new in 2017. We'll talk about it later in the section What's New. But within this XML file you can define new menu item. It's called menu items, so I think it's pretty intuitive. You're going to set the label and description a hint, and you point to a PS of PowerShell file, so this has PS1 in this case, and in that file you will have the logic. But just configure in this file and basically defining where this menu item should look like, where it should show up-- in this case on the file context-- which is the right click on a file. That just by doing this couple of things, when we switch back to Vault in a second, there you go. You have the menu item.

Well, you actually have to restart Vault. In the video, I just cut it out in a few seconds, but basically just by configuring the XML file, you just restart Vault and you have your menu item. This could be done since Vault 2010, 2011 with .net. But it was a mess. You had to start up-- fire up your Visual Studio, register the DLLs, implementing to phase, do all this. And by the way, I take a bet with any skilled developer. I'm faster doing it this. So the time it takes to start a visual studio, I'm done with creation of the menu item.

So this is just to give you a feel of where, or who, is data standard addressing. It's basically you, here in this room. Not every one of you has to do it, but it's important. It's possible to do it. So when the developer comes back and say, I don't know. Maybe it's too complex. And I have to look. You can say, I have seen a fancy Italian talking about this stuff in Las Vegas. So it's doable, and just go ahead and do it.

So this is about menu items. Let's give a look to dialogues. Here, for instance, a little customization I made for the creation of the folder, I want to select a template. So when I'm going to create a new folder, I actually want to create a folder based on another selected folder, so that all the subfolders underneath get carried over. Let's suppose you have different type of projects, five, four, six, seven, typical projects in your company. You created a template folder somewhere. You sketched up all your sub folders, the way you want with the categories and [INAUDIBLE] and so on. You're going to create a new project. You don't want to do it again. You want to pick one of the existing ones, and start it over. It's kind of a copy designed for folders, if you want.

So in this case, the simple solution was, let's create a new folder. So we're going to create a new folder. We're going to pick here one of the existing folders. And I could filter and so it just made it very simple. I put in a name. And when I save it, you will see that this new folder has immediately also this subfolder components. If my folder would have contained even more folders, then all of the folders would have been recreated. These are the little things that can be done very easily, and very simple.

Actually, this is something that I have posted on our coolOrange blog, so if you are interested in this stuff, you can just go there and read in more detail how this has been done, and so on.

I'll give you, then, the source at the end. But this is the kind of things that drove us, back in the days, to develop MyView, and I think this was one of the reasons why AutoDesk decided to purchase this stuff. Because it's one of the typical scenarios-- I see a lot of heads here nodding, and so I say yes, we need this, or we know this, or whatever. So it does have to be easy to be done. And I don't want to invest here three weeks of work for doing this. It should be fairly simple.

So ability to customize, not configure. In this case, it's no longer a configuration. It's a customization. But it's fairly compatible. So, for instance, this example here has been done, I think, with Vault 2015. So on the block, this is the thing we did in the past with 2015. Just last week, as I prepared here the stuff, I just did exactly the same thing like I did three years ago, and it worked. There is no change here.

So let's talk about tabs. Under the data standard extension, data standard configuration, there are soft folders for file items, and so on and so forth. Again, it's a loop, so it will come up again in a few seconds. And windows folders, you can basically create new XAML files. So here configuration, you see a lot of folders for SEO folders, items, files, and so on. In the file, we have an additional XAML, and you see the CAD BOM on the data sheet. And just by creating an XAML file in that folder, If you restart Vault, you get an additional tab. The file could be even empty. I don't know what happened if it's empty, but still, the tab would be created. Of course there will be nothing in.

Now in this case, for instance, we used it for doing an integration to [INAUDIBLE] This is also interesting. The data that you see here are not coming from Vault, they're coming from the ERP system. So I'm in Vault. I clicked on a file, and based on the part number, I can basically connect to my ERP system and see stuff that comes from the ERP system. Life. And I can write it back, and so on.

So think about this tab as a possibility for you to summarize information that may be related to Vault. As an example, you're working with items. You click on a file. You may want to see the item information of that file. As a tab. You don't want to switch from the file to the item to see more information. You could show this information directly underneath in a tab like this.

In this case, we went even beyond. Instead of showing data that is within Vault, we show data that is outside of Vault. It could be data that comes from SharePoint. Or even other systems. If you're having a company-wide data management system, Team SANTORIN, ENOVIA, MATRIX ONE, SharePoint, whatever. It could be an idea to say, OK. Instead of switching between two applications, I could basically extend Vault to show me that information is relevant to me as a designer within Vault, in a tab, for instance. And it's bi-directional. I can read data, but I can also write it back.

So creation of the tab, pretty simple. Go in a folder. Just create an XAML file. And I can show later how this XAML file if can be also configured.

So, creation of new tabs. So, in summary, the purpose of data standard is, as a product, to simplify standardized data entry, you see the [INAUDIBLE] before. Nothing to do, configure your Vault, install data standard. You're good to go. But of course, and this-- and the dialogues reflects and respects the Vault configuration. So not tired, to stress this enough, configure your Vault. Making sure that the Vault is configured properly. The rest will go along the way.

Then the other thing is to think about-- so it's every version of data standard-- the focus is really to think about more configuration of router customization. To make it always simpler for you to tweak a little bit to dialogues without getting your hands dirty. But on the other hand is, if it comes to configuration, then try to make it as simple as possible. Now, the definition of simple is different for every person, whether you are a developer or not. But again, I bet that all of you that are not developers, if you spend a little bit of time and give a look to the scripts and the PowerShell files, you at least can read them. At least you can follow the structure.

So finally, let's come to the real topic. What's new in data standard 2017. So a new feature that has been introduced is the-- so is this Save Copy As. This is a pure inventor feature. And what it does, it gives you the ability for drawings and for assembly, to export a different format directly into Vault. So if you look on the top right side, you see this little picture. It's a screenshot from in Inventor. You haven't the ribbon, this additional function called Save Copy As. And by pushing that function, you basically get this dialogue. And if you are in a drawing, then you have the ability to export the file as a DWG, DXF, and PDF. These are the three formats that are supported. If you are in an IPT or an AIM, you have step NJT as a format.

By doing this and selecting the folder of Vault, you basically immediately saved this export file directly into Vault. Now, the file that you're going to export is not linked in any way to the original file. It's just there, so you can save it where you want, it's up to you. You can name it if you want. It can have two properties that you want. And by doing so, it takes over the properties that you have already in the file, so you don't have to retype everything. So it suggests you the same properties, if you want, but you can change them. You can pick a different category. It's a really independent file, and it has no relation to the original one.

So that means that let's suppose you're going to create a PDF of a drawing. So you select the folder, select the name, save the PDF in Vault. A week later, you change the drawing. You want to re-export this drawing as a PDF. You have to do the operation again. And if you're going to select the same folder and the same file, then the dialogue will tell you there is a ready one. Would you like to overwrite it? And you can say yes. But if you select a different name, he doesn't have any memory. He doesn't have any history. So every export is independent from the previous one. It's up to you to save it in the right place, and the right way.

This is the standard. Of course, knowing that we can customize this thing, of course this can be even customized to the specific work [INAUDIBLE] that you have. Question?

AUDIENCE: The options still work the same way as the Copy As?

MARCO MIRANDOLA: Right. Good point. Yes, the option dialogue. Right. On the dialogue here, you see basically this option dialogue, and it opens up the original Inventor option dialogue. So it's--

AUDIENCE: So you want to do one-to-one?

MARCO MIRANDOLA: It's really the same. It's the same function like the original Save Copy As function in Inventor. And if you click on the option dialog, you really get the Inventor option dialog. So the settings that you have-- and by the way, if you don't open up the dialogue, it takes the settings of the last of the previous configuration. So you wanted to save a DWG format in AutoCAD 2000 format because you want to give it out, whatever. If you did it before, it still keeps this configuration.

Sir, one thing. I noticed-- I don't know if it's just a problem of my [INAUDIBLE] machine, I noticed that this dialogue is showing behind the data standard dialogue. So if you try it and you don't see it immediately, try just to move away the data standard dialogue. The data standard dialog is always on top of everything. And my feeling is that it's also on top of this option dialogue. So just moved windows aside and then you should see it.

AUDIENCE: Can you put the option to attach it to the [INAUDIBLE]?

MARCO Sorry, the question is?

MIRANDOLA:

AUDIENCE: Can you have it attached to the [INAUDIBLE]. So like it becomes part of the parent file.--

MARCO MIRANDOLA: So the question is, basically, can I make an attachment--

AUDIENCE: Like a step-file.

MARCO MIRANDOLA: Right, can I attach, basically, the exported file to the original file. Technically, yes. There is just a but. And there is one logistical problem. I'm here on a file in Inventor, which is checked out from Vault. And while it's checked out, I can do nothing within Vault. So the problem is that I have the file checked out. I created the STEP file of the PDF, or whatever. That file is checked in. And in order to create the attachment, even the original file would have to be checked in, which is not. It's currently checked out. So this is why at the moment, this does not supported.

Now this is something that in the case should happen than later. So when the original file gets checked in, then technically, we have the ability to say, oh a file gets checked in. Let's see whether there is a dependent file, whatever. And create the [INAUDIBLE] connection. So I would say, yes, it's possible. But you have to take it from another side because the file is checked out.

AUDIENCE: Would it have to be post-processed?

MARCO MIRANDOLA: Right. It would have to be post-processed. Question?

AUDIENCE: [INAUDIBLE]?

MARCO MIRANDOLA: That's correct. You can. So the question is, does it only work when it's checked out? No, it does also work when it's checked in. So you can keep the file checked in, and just do a get and open it up. And you can still do a Save Copy As. That's right. But this is the point. So I remember we had this conversation with the product management team, and it was-- the thing is this. How can you do something without confusing the people. And in this case, the decision was to say, OK, let's do nothing because we are then consistent. Otherwise, we would have somehow to either not permit to do the Save Copy As while it's checked out, or bring [INAUDIBLE] and say, hey, you can do this, but be aware that because the file is checked out, you cannot create the link and so on and so forth.

So yes, I can do this even when a file is checked in , and if this is the rule, then it's pretty simple to customize this and make the attachment. but there are some things that need to be just thought about it and then take the decision.

AUDIENCE: So [INAUDIBLE] fully customizable?

MARCO MIRANDOLA: The dialogue, yes. Yes, so the dialogue is basically this XAML file-- I can show it to you later. So you can configure the data the way you want. Not only this, every data standard dialogue. It is configurable and customizable. And of course, all the functions behind like, OK, cancel, and so on and so forth, are also customizable. So you could basically connect to the OK button and say, well, if the is checked in, then check in the file and do the connection, and all that stuff. So yes. Yep?

AUDIENCE: Is there a way to keep the PDF file from automatically go into Vault to do this. Because I don't want my users to put a PDF of a drawing in the stack in this update. Only PDFs allowed [INAUDIBLE].

MARCO MIRANDOLA: So the question is basically, can I do this automatically with every check in. So can I generate a PDF automatically--

AUDIENCE: No. I don't want anything-- any PDF files that are generated from [? inventors ?] in my Vault.

MARCO MIRANDOLA: So your question is, can I do disable this function.

AUDIENCE: Yeah.

MARCO MIRANDOLA: OK, yes. Well, the easiest way is just to remove the bottom from the ribbon. There is in Inventor-- there is a configuration file where you basically can configure which buttons should be visible and not. The simplest thing would be really-- it's the cheapest one. Go into configuration file, remove the button, then you're good to go.

AUDIENCE: It would still be nice [INAUDIBLE]

MARCO MIRANDOLA: Oh, well that's the other thing. If you want to do it, be more selective, than I'm not sure if this can be done by just removing the file for certain time, but another thing that could be easily done is, in the PowerShell script that is behind this dialogue, just make it very stupid if. If File Extension equals IPT, go ahead. If File Extension is not IPTA, stop there. And this is, again, the type of logic that I bet that most of the old-fashioned script guys-- maybe you cannot do it, but you can read it. So it's-- by saying this, I see people nodding heads, so I think you can.

AUDIENCE: [INAUDIBLE] is there a way to set a [INAUDIBLE] mode so if we did?

MARCO MIRANDOLA: Sorry, say it again.

AUDIENCE: Is there a way to attach [INAUDIBLE]?

MARCO MIRANDOLA: So the question is, can I customize the PDF, or can attach paint table or this kind of stuff to the PDF. Well, I am not sure. Well, I will answer the question this way. Whatever Inventor can do-- so this is the native Save As PDF function-- then this dialogue can do the same. Not more, not less. So I'm not sure what Inventor can--

AUDIENCE: [INAUDIBLE]

MARCO MIRANDOLA: This is just Inventor. All right. Another thing that has been changed, and actually, this change came in with the update 1 of data standard 2017. This is something that we waited for a long time. In the Inventor Vault options of the Inventor add-in, you have this ability to define whether the numbering scheme with the number generators should be used for drawing or not.

So, typical example, you have a component, which has the number one, two, three, four. You want to have the drawing with the same file name. You want to have also the drawing to be called one, two, three, four. That ID [INAUDIBLE]. So this is something that you could always have been configured inside the regular Vault add-in. And now it's also supported by data standard. If you look into this video, it's just looping. It's set in this option. Say yes, generate me a number for the drawing. Once I save it, I do have the ability to select the numbering scheme I prefer. And so generate a new number, a different number for the drawing. If I switch these off, as is usually the case, if I do this, then I can't select a numbering scheme, and the drawing gets the same number like the model that you have inserted. Right

This was something that in the past was always possible with data standard, but it had-- It was connected with some coding, with some customization. We have an example on our blog that explains how to solve this problem. And, now basically, it's part of the standard. You don't have to think about. But it's update one. So data standard 2017 update one supports this-- finally this option.

Another thing, is with 2017, the logic between category and numbering scheme has been revisited. So if you have a category called Engineering, which you do. And you have a numbering scheme called Engineering, then by picking a category Engineering, the numbering scheme is already pre-selected. You can't pick a different one if you want, but by selecting that category, the summary scheme is already pre-selected. Just based on the name. There is no more logic behind it.

So here again, configure your Vault. Create your categories. Create your numbering schemes. And if you created the number schemes with the same name as the category, then within data standard, by selecting the category, the number scheme is already pre-select. You can select something else if you want, but at least it's already pre-selected.

Now for those that want to impose this logic, there is a very pretty simple workaround, which is basically just set the numbering scheme to be read-only so that the user can not select something else. Still, by changing the category, the logic will basically change number scheme behind, so you will see the change in the dialogue, but the user is not able to select something else. And in this way, very simple, just a little configuration in each XML file. You can basically impose this rule for the company.

So this is one thing. And another thing which is very nice, with a custom object, it goes even a step further. So if you create a custom object, in my case it's called transmiittal. And if you go here into configuration, I have a category called transmittal. And I also have a numbering scheme called transmittal. So the custom, object, the category, and the numbering scheme are all called the same. What happened is that, the moment that I create a new transmittal object, the category and the numbering scheme is already pre-select.

So here again, the idea basically is to make it as simple as possible to leverage the potential of your Vault to make it simple for you to go beyond the typical file check-in and checkout stuff. And do more stuff with Vault, even with custom object. And then use data standard as an out-of-the-box free extension, AutoDesk extension of Vault, to make it very, very simple to work with custom objects. So think about it. If you work with custom object. Yeah, you have a question?

AUDIENCE: If we've got the ability, say, in one of those [INAUDIBLE] Save As, can we access user names [INAUDIBLE]

MARCO MIRANDOLA: Right. So the question is-- I'm just repeating for the recording-- the question is, is it possible within the data standard dialogue, to access data sources that are either involved or accidentally [? SEAM ?]

MARCO MIRANDOLA: As an example, if I have a list of custom objects, supplier names, or project names, or even, for instance, some customer using custom object for the multi-language issue. I have drawings that I have to deliver in different languages. And they basically how this text catalogue. We have different columns. You enter the English one, and then someone is going to translate in different languages. So can I basically use this type of information within the data. The answer is yes.

So you could, for instance, have-- actually, like the example I showed before-- the one with a folder where I copy a folder. So that one is doing the same. It's actually a combobox. But that combobox is connecting to the Vault server and ask for the list of projects. And here is the same. I can have a combobox which points to specific custom objects and shows me the list of custom objects. But it's not limited to Vault only. It could be also something external as well. I could have a very [INAUDIBLE] XML file, and have data standard, some comboboxes pointing to that XML file and show the data and allow you to select. But it also can go beyond. You can't even connect to your ERP system.

So you create a new folder, or a new product in Vault. And you want to pick the project number from the ERP system. No problem. You create a drop-down box to connect your PowerShell to the ERP system. Get the list of projects, and by picking the product from the ERP system-- maybe title, supplier name, cost center, or whatever it's important-- is automatically taken over to the dialog. So absolutely.

Another-- and this is related customization, so this is really the point where we are 100% percent of the customization space. But it's all doable, and it's not rocket science. Another common thing is, for instance, cascading combo boxes. So I have two, three, four, combo boxes. I select the machine type, and by selecting the machine type, the next combo box should show me just the machine models. even these kind of things can be easily done in data standard. It's absolutely doable.

So again, if you use custom objects, or if you want to use custom objects in the future, then basically just create a custom object with a name. If you create a category and a numbering scheme, either one, category and number scheme, or just category-- that is up to you. But if it's all three, if it's custom object, category, and numbering scheme all with the same name, then there is basically nothing you have to do. We just go in-- well, there is actually just one thing to do. You have to configure the menu item files, so the menu file behind, so that this dialog shows up for your custom object. But once you do that, then the data is there and you can use it.

So next one. For those that are actually willing to do a little bit more fancy stuff with data standard, with 2017, especially for AutoCAD and Inventor, within PowerShell, the AutoCAD and Inventor API is exposed. So now, when you are inside in AutoCAD or inside in Inventor-- an example with Inventor. We know resellers and customers that want to access to an iLogic rule. so they save the file. I don't remember exactly why. I'm not an iLogic expert. But correct me if I'm wrong. There is something about having the ability to react on the first save, or that-- I'm not sure. So there is something about there. There is a trigger, but you have to load to the iLogic rule before, and-- yes, anyway.

So the scenario here, that I heard is, I want actually to run that iLogic rule the very first time this file has been saved. And now, they can use-- they can leverage data standard so that when the data pops up and you say OK, they can basically be in Inventor or API, access the iLogic rule and say run. This is just one example, but it could be anything.

It could be, for instance, you have a drawing, and you want to not just only test whether the properties are fine, but even test if the title block has been introduced. Or, let's say you have introduced a new type of title block, and you have legacy files with old title block type. And you may want to double-check that if someone is going to save an old file as a new version, the title block has to be replaced. You can access the Inventor API and just make a test whether the block is the right one. And if not, you can just prevent the OK button to be hit. So the guy cannot save the file.

If you give a look here in this video, this is the variable inspector of data standard which gives you the ability from within PowerShell to stop the code and get a printout, a dump of the current state of the variable. So this is a little bit more developer stuff. But you have here this [INAUDIBLE] document and this [INAUDIBLE] application. And this [INAUDIBLE] document is the active document that you have within AutoCAD and Inventor. And the [INAUDIBLE] application is the application itself. Is Inventor or AutoCAD.

So again, in this case, we are really in the deep developer space. And you have to be aware about the Inventor API and the AutoCAD API, and so on. For those that are not developers, just take it with you. Take it with your ideas, with your thinking. If you have a certain situation, be aware that within data standard of you have now access to the API, and you can do cool stuff. Question?

AUDIENCE: Do you know [INAUDIBLE]?

MARCO MIRANDOLA: AutoCAD OAM. Support for AutoCAD OAM.

AUDIENCE: [INAUDIBLE].

MARCO MIRANDOLA: Put it in this way. I would say that AutoDesk probably do not officially support VDS for AutoCAD OAM, like it does not for AutoCAD Electrical and so on. So the official applications are AutoCAD Vanilla, AutoCAD Mechanical, and Inventor. However, because it's all based on AutoCAD, there is a configuration file behind. I don't remember the name. But if you want, you can just drop me an email, I can send you the information.

There is a configuration behind data standard-- the plug-in, the add-in for AutoCAD. In that folder, there is a configuration file, and there you can configure which application types are supported by this add-in. and if you enter-- so if you change the configuration file, and you'd add additionally, also, the name-- the application name, like AutoCAD Electrical, AutoCAD OAM, or whatever it is-- then the add-in is loaded, and technically there should be no limitation.

AUDIENCE: The binding for the AutoCAD OEM might lock your ability to--

MARCO MIRANDOLA: Oh, that's a good point. Yeah.

AUDIENCE: So you should check with the tech-soft guys or the OAM team, because it may be the mechanism that exposes the API to data standard is locked in OAM when you buy your OAM app. So that's worth checking. It may or may not.

MARCO MIRANDOLA: Right. So the point is that, it could be that, technically, the add-in could be loaded, but the AutoCAD OAM is not allowing you to load this add-in because of the OAM restriction.

AUDIENCE: I'm not up-to-date on how they do that now. It used to be a [INAUDIBLE] process. It might be different now.

MARCO MIRANDOLA: No, it's a good point. It's a good point. So, but again, officially, no it's not supported. From a technical point of view, there is this configuration file where you could, basically, allow the add-in to be loaded by different types of AutoCAD. Whether AutoCAD OAM then prevents the loading or not, this is something that needs to be discussed with the OAM guys.

Another thing, with Update 1 of 2017, they introduced some additional debugging dialogs. So before Update 1, if you screwed up data standard-- you made an error in the XML file, or in the PowerShell file, and so on-- you did have a log in the log file. But raise your hand. How many of you are reading log files? Really? One?

AUDIENCE: [INAUDIBLE]

MARCO MIRANDOLA: OK, good. So all these errors have been reported always in the log file. And some good, some bad. But they have been logged. But no one really looked through the log file. The first thing is, you clicked the Save button, you expect the dialog to come up. The dialog is not showing up. And you immediately go to your code and think about, what is the thing that I changed in the last five minutes. So what are the things that I screwed up? And you try and try to revert them back to the point where it's working. And then you work again forward.

Now, to simplify this thing, now you get basically nice data. So the dialog comes always up. But it comes with this message box. Yes, so a red, ugly icon to say you made the mistake. So you are the one that screwed this up. But I tell you where. So this, for instance, is an example of an error message of a XAML. XAML stands for the dialog. It's a Microsoft language. XAML-- "ex-am-l." And this is about that the dialog has been configured, but screwed up. And here's an example on line 258, the second one from the top. We have here a text box. I think most of you understand what a text box is. And if you see the definition goes with text, blah, blah, blah, and so on, in the next line, 259, it ends with a forward slash. But if you look to the next one, we actually missed the closing bracket. So by typing, someone opened up the bracket, wrote the stuff, but forgot to close the bracket. Typical situation.

So the dialog comes up, and says, hey, you screwed up the file. It's the XML file. I tell you where. I'm friendly. I tell you which file is affected, and I'm telling you also the line which causes me headache. In this case, it's like 259, position 57. And the missing token is-- or the missing character is basically the closing bracket. So this error message is very good actually. There might be other messages which are not that precise. For instance, if you open up the quotes, but you forget to close it, then it could be that the next opening close is interpreted as the closing quote of the previous one. And so it might be that the line is not 100% accurate. But if you look to this line, it's probably exactly there or slightly below. So it's not that far away. It really helps you a lot to understand where you basically created this problem.

This is about the dialogs. If you screw up the PowerShell, it's pretty much the same. So here, for instance, on line 89, I do access to a property called Does Not Exist. So this property does not exist. But I want to access the property, and because it's not existing, the code doesn't know what to do with it. And you get this error message saying, pay attention in the function, initialize window. In the file, default PS 1, in this location, I found on line 89, character 2, this line that caused me problems. Please fix it.

So it's actually very good for those that are doing customization. Now they basically get immediate feedback that they screwed up something, and they, more or less, really know where to go. This, again, requires Update 1 for data standard 2017.

So if you remember the things that I mentioned before, the purpose of data standard in the evolution of the last version, and I think even future version, is one hand to make configuration more and more and more easy. To reduce as much as possible the need for customization. But on the other hand, if you really need to do customization because you want to access the [? OP ?] system, and this is not something that AutoDesk can leave out of the box. But if you want to do this stuff, the purpose is-- or the aim is to try to make it as easy as possible, as transparent as possible, to help where Vault data standard can, to make it as transparent, as easy, as simple as possible, And so on.

And I think this is-- well actually, I spoke with a lot of customization guys, and they actually liked this a lot. Because it introduces this it's half an hour, hour, one and a half hour of error finding where you basically are going backward to your code. And it tells you immediately, hey. It's not that friendly. It could be maybe a little bit more. But it's precise.

So what's changed in data standard 2017. One thing I mentioned before, the ability to create these new menu items. So before 2017, the menu item was actually an MNO file called My Menu MNO. And this comes back in the days when we developed originally data standard. And it was some kind of cryptic language. Don't ask me why it didn't choose, right away, to make XAML, I think-- or XML, I think. That one of the developers was just willing to try something new. And then they made the syntax.

Anyway, with 2017, the old-fashioned has been removed, and now we have a very clear and clean XML file where you can configure the menu items. This is how the XML file looks like. I personally prefer the XML Notepad as an editor. It's a free Microsoft editor. It's like the Notepad, but it's just the XML Notepad. You have to download it from the website of Microsoft. Just Google it. It's free, you can download. And it's way easier to configure.

So if you did not configure menu items before, you don't have to care about this. You just update your stuff to 2017, your dialogs, your PowerShell scripts, and you good to go. If you created a menu items before 2017, this is one thing we really have to sit down 10, 15 minutes, and basically recreate the menu items that you made before. But again, there is really-- they pay a lot of attention to make sure that all the customizations you do are carried over, that the money that you have invested for customization is not lost. You don't have to reinvest it again.

In this case, actually for several reasons, there have been problems with special characters, here in this format. With blanks, with special characters, and other stuff. There were even some additional other boxes and so on, that came up in the years. And this year, basically, the idea was we are either going to fix this logic, or we finally get rid of this stuff, and we go through the simple XML file, which can handle all these special characters, and blank spaces, and so on and so forth. And I think it was a good decision to get it to go this way. So XML file, pretty simple.

AUDIENCE: To do the coding on thi this XML?

MARCO MIRANDOLA: No. So the coding is not done in XML file. In this XML file, you basically just configure the menu item. If you look here, we have this menu item called Hello World in this example. You have the description, the hint, and then you have this PS file, this actually with PS with the third one from top, which points to this Hello World PS 1. The Hello World PS 1 is actually the PowerShell's crypt that will be executed once you click on this menu item.

So the logic, the coding, will be in the PowerShell file. This is just for the [INAUDIBLE] Vault. This just allows you to define a menu item, and then decide here in this section common sites-- it's the internal code name-- you can basically define where this menu item should show up. On the toolbar, on the context menu, on the File menu, on the Help menu. So you have here all these folder, content, standard toolbar, Tools menu, Help menu, and so on. You can basically say, OK, these menu items should show up here, here, and here. when you click on it, it runs the PowerShell script. I'll just finishing this. If you want, we can give a look also to the life system, and give a look here and in detail.

Another thing that has changed, as I mentioned before, when I showed before how to create a new file within Vault via data standard, you saw the ability to select a template, a PowerPoint, or a word document, or whatever. Now-- and I mentioned that you need these templates folder, where you're going to place the templates. Now that folder is configurable, and this is where you can configure it. Now, before 2017, this piece of XML was inside the dialog-- inside the dialog definition XML file. Now, it's external, so we have this file called File XAML, which is the dialog itself, and the file XML, which is basically this configuration here.

The idea is that, before 2017, if the administrator was willing to add additional template, he had to go into this magic dialog file and change the code there. If he screwed up something, then the whole dialog was screwed up. And in this case, I say, no, no, look. The dialog is one thing. And here you have the simple XML file where you can configure where the templates are located, and so on. And if you screw up that, it's not a big deal. That's the idea. So actually, nothing new, in this sense. The only thing is that this file has been extrapolated from the dialog, and now it's a separate file you can use for editing.

So another thing that has been added, you have now additional-- so this is for the coding guys-- you have now additional properties. An underscore category and underscore numbering scheme. And actually, especially the first two ones, they will be also stored as user-defined properties.

So with 2017, if you create a file and you save it via data standard, if you give a look to the files properties of that Inventor and AutoCAD file, you will notice new user-defined properties. They are just there, if you want them or not. And they are called VDS_Category and VDS underscore numbering scheme. Why this? You create a new file in AutoCAD or Inventor. You have this nice dialog. You select the category Engineering-- whatever you want. You saved the file, and then when you check in, you go into Vault, and the file is not under category Engineering, but it's under category whatever. How is that possible? I selected the category Engineering the dialog before. I just checked in the file.

Well, you have in Vault a rule for the category that says if the file has the extension DWG, and the property, whatever, is set to whatever, the category must be this. So whatever you select before in AutoCAD or in Inventor, when you check in the file, it could be that because of a rule inside in Vault, the category is basically switched-- changed. So who is right? Well, both are right, because in Inventor, before check in, I'm in the lead. I select whatever I want. But when I check in, Vault is in the lead, and he decides to do differently.

Now to overcome this problem in the past, we basically created an own user-defined property, and we based our rules on that user-defined property. So that when I select in Inventor and AutoCAD that category, then that rule in Vault says, oh, if this property has this category, well, then I'm going to apply that category. And so the rules between Vault and CAD were matching.

To simplify this, now the category that you select in the dialog will be automatically stored as a user-defined property in Inventor and AutoCAD. So if you do a mapping in Vault, you can basically place or put the rule based on this mapping. Way simpler. No coding, no PowerShell fancy stuff, no workaround. Pretty simple. Create an additional user-defined property in Vault. Make a mapping to this user-defined property in AutoCAD or Inventor. And then you can basically base a rule on top of it. Super simple.

So you see, again, the purpose is, again, to try to remove all the things that in the past have been done manually with PowerShell. AutoDesk learns about from you doing and listens about your feedback, and basically generates or improves the product in this sense.

Another thing is that, in between the PowerShell code in AutoCAD and Vault, there have been some differences between create mode and edit mode. Now, across AutoCAD, Inventor, and Vault you have these internal properties called create mode and edit mode. What these properties are doing, they are telling you whether you are in create mode-- so it's the first time the file has been created-- or the file has been already created and you're just editing. And based on that information, you can decide to say, OK, it's the first time that the file has been created. Then I want to set this and this property. If it's just a sequential edit, then I don't care about it, and so on. So this is more for the coding guys. Good.

So what do you have to think about when you do an update? The MNO file has been replaced to an XML file. If you did not play with the menu so far, I don't care. If you did, take this 10-15 minutes and re-edit your XML file. The templates are now in an external file. Nothing that you have to do specifically, but if you have specified templates in the past within the XML file, you can still keep it there if it's OK with you. But maybe it's a good idea to do the same like AutoDesk and basically put it in a separate file.

Dialogs have been simplified. One thing that you will notice if you worked with data standard before, is that the XML code was quite long, and there were a lot of logic, even inside the

XAML code, all this logic has been removed and is now in PowerShell. So the XAML file is now very, very simple, very compact. And all the logic is in PowerShell , so even here a lot of simplification came in. And again, the category number is [INAUDIBLE] logic. I think this is a cool move. To make it very simple for an exiting customer it might be a problem. Because if the name is not the same, then it doesn't work, and renaming it, it's maybe not that easy. But especially for new implementation, keep this in mind. Start from the beginning to think about category, numbering schemes, custom object, name them in one way, and then data standard will follow.

So technically, you can take what you did with 2016 and just copy, paste it in 2017. Whatever you did, you take it and remove it. We have customers that started, for instance, back in the days with Vault 2013. So with our version-- with MyView. Back in the days, not data standard. And they are now moving to 2017, and it still works.

Now in those cases, the dialogs were highly customized because they had a bunch of properties, a bunch of fields, bunch of rules. So they did not care about the new implementations, the new functions that came in and 2017. It was all not usable for them because if they have so much customizations they need for their purposes. And in that case we said, why bother? Let's just take what you have, copy, paste them to a 2017 environment. Check if everything works. I think we had to tweak a little bit, but it did work.

On the other hand, if you like the stuff with 2017, and you want to apply it, well, in that case, you know you have to start with what 2017 offers. And then little by little, reapply the changes that you made in the past to the 2017 dialogs. But it's really a decision what makes sense. If it's highly customized and the new stuff, it's not interesting. Don't care, take the old stuff, take it over. If the new stuff is interesting, well, then you have to sit down and rethink how can you leverage the new stuff? And what should you do with the old stuff?

So questions. Again, I have a live system here. We have another good 18 minutes time. I can show you some live stuff if you want. I have just one more slide with the references, so that you know where you can get help and so on. Any questions?

AUDIENCE: Is all the data standard stuff [INAUDIBLE] you know what I'm saying? Is there any way to centralize the setting so--

MARCO So the question is, if the stuff is client-side. The answer is yes. The add-in is client-side. There

MIRANDOLA: are not good ways to centralize it. The way that this is solved is by deploying the configuration, the customization, across all the clients. The tool from Doug Redmond, the Thunderdome, I think it's supporting data standard. I've actually never tried it, but what I heard-- it's working, and I see some guys saying yes. So I get confirmations. But yeah, it's client-side and needs to be deployed.

AUDIENCE: [INAUDIBLE]?

MARCO MIRANDOLA: A plugin for Microsoft Office. This is a good point. No, there is no plugin for Microsoft Office. And this is why, actually. Vault, a data standard for Vault, came in so that, basically, especially for Office, the way to treat it is by creating a file within Vault and then start from Vault and open the application. The reason why there is no plugin for Office is that, Office is managing the plugins in so-called apartments. So every plugin is in an own apartment and doesn't know about the other plugin. So when one plugin crashes, it's not bringing down the whole application, just that one plugin crashes.

Now why is this important. Because actually, the data standard ends the Vault plugins are two plugins that are working together. So inside in Vault and AutoCAD, you don't have to login twice. It basically uses the connection of the Vault plugin to communicate back to Vault. Now, with Office, this is not possible. So for Office, basically, the Vault plugin and the data standard plugin would have to become one. And in that moment then that would be possible. So the answer is no. And the technical explanation is this apartment thing. OK. Another question.

AUDIENCE: [INAUDIBLE]?

MARCO MIRANDOLA: It's-- yeah, it's the same question. So can I have, basically, the VDF code somewhere else-- somewhere on a network.

AUDIENCE: [INAUDIBLE]

MARCO MIRANDOLA: Even locally, no. So--

AUDIENCE: [INAUDIBLE]

MARCO MIRANDOLA: Yes. Yes and no. Put it this way-- I can show it to you here on the system. So this is where you have the data standard code, for instance, here for Vault. And here is basically where you have the PowerShell files. So this the default PS1, is one of the central files. And this is how it looks like. What you could do, is within this file, load a different PS1 file, which is located completely somewhere else. Yeah. So here within this file-- because this file is read by data standard-- what you can do is, in this file, or even you can place another file here, so basically all the files in this folder are read. So you can create a custom file called Import PS1, and in that file, the only line you have is Import Module from that path. And that part could be everywhere.

So it's not 100% what you want, but it goes in that direction. So you still have to have a little-- at least one file here-- that basically then loads the file from a different location. That would be possible. Yeah. Other question?

AUDIENCE: The CAD BOM, is it a flat BOM or can it be structured with other [INAUDIBLE]?

MARCO MIRANDOLA: So the CAD-- well, the CAD one you see is a flat one. But it's not limited to that. So the dialogue how it is shown is just the one, the single, the flat single-level BOM, but in reality, in Vault, you have all the information. So, for instance, when we do ERP-- I can show it quickly, when we do the ERP integrations-- I have here this tab, as I mentioned before. So this one, this Navision tab. This one here is showing me the data from live from the vision. Just to give you a feel, I have here Navision running items. If I'm going here on my suspension, and for instance, I just change here the description-- changes back to subscription-- and I go back into Vault, you still see here, the product XYZ, I basically just have to refresh. And here you go. You see it and you change. So you see that this project XYZ has gone away.

Now if I want to write back, I have here this update item, and it takes the description from Vault and writes it back into Navision. So if I'm going back to Navision again. And I just do here a refresh. I see here, again, my old description.

Yeah so, this is what-- why should this, because the question was about the CAD BOM. What we do, for instance, when we do the ERP integrations with PowerGate is we have this transfer BOM function. It opens up this dialog-- this is something that we developed-- and within this dialogue, we are now reading the CAD BOM, multi-level. Yeah, but this is something that we do, so we basically go to the assembly and view the API-- say give me the file BOM. And then we get the children, and for each children [INAUDIBLE] give me the file BOM, give me the file BOM. And we basically build up the structure of the whole assembly.

So now that we have this structure, we can-- and this is then specific now for this integration thing. But once we have this structure, we have done functions like check and transfer, where we can, basically, check the structure against the ERP system and say, is this BOM already existing? Yes, no. Is it identical? What has been changed? Is the quantity-- has been the quantity changed? The position changed? Additional position removed back and forth and so on. But it all starts by this CAD BOM object, which is beautiful.

And the good news is that every CAD application that, during the check-in, also checks in this CAD BOM object. For us, it's irrelevant. We don't have to be experts about SolidWorks, [INAUDIBLE], Inventor, or whatever. We just have to be the expert of this file BOM object. Wherever it comes from, we don't care. We take it, and we transfer to the ERP system. So multi-level, yes. It's possible, absolutely. In this case, this dialog here, it's really just showing a single level. But with PowerShell, you could, basically, transform it into a multi-level.

AUDIENCE: Tell them what the power gauge is that you were using. You should tell them what the power gauge is that you were using for that transfer.

MARCO MIRANDOLA: Yeah, the transfer is done via powerGate. And by the way, if you play with PowerShell, there is an additional tool that we provide for free, which is called PowerVault. And it's basically an extension, and we've wrote to simplify the way to deal with the Vault API. The Vault API can do everything. It's super huge, it's super-- you can do everything, basically. But you have to do everything. So, for instance, reading the BOM from via the API, it's a tough job because you have to handle the [? REIFF ?] component, a virtual component. You have to handle merged components, overwritten quantities. All this fancy stuff that you can do with Inventor. It's all things that you have to do.

Now, what we did, we basically created PowerShell command-lets-- very simple, very easy-- which allows you to get to the BOM. I can quickly show to you, for instance. If I'm going to create a new PowerShell file, I can just say open Vault Connection. And without any arguments, I basically connect to my Vault. So because developers are lazy, if you don't provide any parameters, it goes to localhost, the Vault is called Vault, the user is administrator, and the password is empty. Which is my case. Probably not your case. In your case, hopefully not. In your case, you have to provide information like server and Vault and user and so on.

But that's not the point. What I was going to show you is, I can say dollar BOM equals get Vault file BOM. And I just have to provide a path to my file. Now in my case, it's designs-- well, let me see, what is it. It's design, suspension, and then one AIM. Let's see if I did it right. If I execute this one line. Let's see if I have a BOM. Yes, I have a BOM. So you see I have 19 rows, and here, basically, I have all the properties.

So why did we do this? Because the first time that we, via API, wrote 50 lines-- 150 lines of .net code for reading this BOM, we said no. We have to invent something simpler. And because we're doing a lot with PowerShell, we came up, basically, with this simplified version of the old API. The other thing is that we take care about compatibility. So you can use this get Vault file BOM for Vault 2014 2015, 2016, 2017. We're working now on 2018. And it always works the same, even though the API has strongly changed.

And this was one of the other thing. We created this customization for the customer XYZ with Vault 2014. He moved to 2016, we had to put our hands on and redo the customization because the API changed. This was one of the things that we said, no, we have to invent something that makes it easier to use, simpler to understand, and compatible across to the older version. And we came up with this PowerVault.

So again, PowerVault is free. You can just go on the coolOrange site, and download and start playing with it. It's not covering everything from the Vault API. It covers those areas where we run into, and it covers those areas where recently came back to us and say, hey, I have this, and this, and this problem, and we basically come back and say, look, give me a couple of weeks of time. I'm going to create this command-let for you, and then you're good to go.

So as you see, you have Vault, get Vault file, association for getting the references of an assembly, and so on. Something about item, item BOM, and so on. The typical things that usually are used in this context. Any more questions? Anything you want to see?

AUDIENCE: Is this the CAD BOM feature, is that requiring a customization [INAUDIBLE]?

MARCO MIRANDOLA: You mean-- so this tab here comes for free. Nothing to do. You just install data standard, you get this tab. Now, if you don't like this tab because you want to see more properties, less property, or a different way, whatever, you can do configuration, which is going into the XML file and remove and add columns as you wish. If you wanted to do more stuff like, exporting this content as a CSV file, Excel file, in all that you can hand it over to your ERP system, then you have to do a little bit of customization. Is this answering the question?

AUDIENCE: Yeah, just [INAUDIBLE]

MARCO MIRANDOLA: Well, just to give you a feel, here in this folder, we have this configuration folder, and you see here, basically, the sub-folders for change, or for file, for folder and so on. Let me go here in the file, and you see here this XML file. So I'm just going to copy, paste the file. Rename it coolOrange. And just to give you a feel for editing these files, usually we use Visual Studio. Not for coding, just because it has a very nice designer where you can just play with the dialogs. Now, in my case, I need to add here to my project the coolOrange XML file. Vault, configuration, file, XAML. So this is the new coolOrange XAML file.

Now, of course, this file is identical to the data sheet because I just made a copy, paste. so the content is exactly the same. As you can see here, it's basically the same thing. So one easy thing I can do is, I can just here change the name to coolOrange, for instance. And without doing any further changes, I'm just going to close and restart Vault.

So I copied the file, I just changed the title of the tab. And by restarting here Vault-- it's called, again, data standard, why? Oh, my mistake. No, I edit as the original file. So I created, actually, a copy of that file. My mistake. I need to edit as a reference. Let me do this again. CoolOrange. So pay attention, here. I can say add, but then it makes a copy, or I can say Add as Link. And then it retains the link to it. So my change was not affected. So let me do this again. Come on, close the window. Restart Vault.

Because, actually, for this change, I need to restart Vault. But I want to show you something else which is quite cool. And there you go. Voila. [SPEAKING FRENCH] You see here the new tab. This is-- you could-- this was possible for the last six years, but you had to do it in Visual Studio. This is pretty simple. Just copy, file, do it, and you have the tab.

Now the other cool thing is, now that the tab is there and the name is right, so what I can do is, I can say, well, this should not be called Category. I'm going to call it-- let me see the label on this one. I am going to call it coolOrange-- is label. So I change it here. Go back into Vault. I just have to click in, and then there is a keystroke control F5, which reloads the XAML. So without restarting Vault, you see immediately, basically, what it looks like. At least with Visual Studio you have to debug, and then pause, and then stop again, and close Vault, and restart Vault for every little single change. So you see this is-- it's not all simple. You have to spend your time with data standard. But I hope that you get a feel about, the aim about the simplicity that it can have. And still how powerful it can be.

Simple things are done easily. Complex things are doable. It's not simple, but it's doable, and it's doable in a fairly simple time. We do use this a lot, especially for integration things. When it comes about ERP integration, SharePoint integration, other stuff. It's all general customizing.

You have a question?

AUDIENCE: After you do these customizations, how do you print them to the client?

MARCO MIRANDOLA: I have my little dollar, five years I give them on USB stick. Two dollar, and she runs around-- no, seriously.

(LAUGHTER)

I do that. No, you need to go around, basically, deploy this kind of configuration, customization, across all the clients. You don't do it with little kids. Officially, not. But you basically, either do it-- talk with the administrator. They usually have some server-side scripts that they run when the machine goes up. This is one option. Another option is a PowerShell script that basic does this. The Thunderdome solution from Doug Redmond is doing this, but yes, every change needs-- but there is also an advantage with this. We just had a conversation yesterday.

So how many of you have a test environment? OK, that's a good sign. So it gives it the ability to do the thing locally, make all the tests, test it, and when it's fine, then deploy it. So there is this good and bad. But I think, to some extent, at least for this purpose, it's good that you can do this stuff locally. And it's even something, if you have multi-sites, and so on. You can have different configuration for different sites. So again, there are arguments for it and against it.

AUDIENCE: [INAUDIBLE]?

MARCO MIRANDOLA: Yes. But you have to do this via PowerShell code. But yeah, you can access it. Basically, within the PowerShell code, you have access to the full Vault API. And for AutoCAD Inventor, also to the AutoCAD Inventor API. So really, you can do everything in PowerShell again as .net language. It's a script language, but it has the full power of .net, so you can connect to whatever you want. OK, the nice lady tells me that the time is over. One more question?

AUDIENCE: Does this work with Vault Office?

MARCO MIRANDOLA: Yes, it works with Vault Office. Actually, for Vault Office there is a special dialog. There's a special dialog here, this one. You see, file Office because this data basically doesn't provide you the ability to create CAD files. So specifically, for the Vault Office, you can have a simplified version of your dialog because the Office guys have a simplified version of Vault.

All right, if there is nothing more, I'd like to thank you for the time. I hope it was interesting. Oh, last thing before I close off is the sources so that you have them as well. If you have any question about data standard, the first link here is the link to the knowledge base. So you find there are a whole custom configuration-- sorry, documentation for configuration and customization. Go there, give a look, and so on. If you have any questions about customization and so on, go to the Vault customization forum. The AutoDesk guys are there, and even we-- so even we respond to your customization questions and so on.

If you want to see some examples, some ideas, stuff that we did, go, to the blog-- blog.coolorange.com, and if you need special support, if you need someone that looks over your shoulder, or if you need someone that takes one or two hours with you to go through some specific things, and so on, we provide that as a service. The simplest thing to do is drop an email to support@coolorange.com. It's not for free. I have kids. They need new shoes every year. But their shoes are cheap, so the kids are reusing the shoes, so we're not that expensive.

But asking the question is for free, the answer is for free. If we notice stuff right away and it takes 30 seconds to answer, we're not boring. We are more interested to have you working with this stuff. If it really comes to a point where you need a two-hour, three-hour, four-hour sessions, where we go through the stuff together, whatever. We can do that, and that of course will not be for free, anyway. But I would say, if you have a need, drop an email, and then we will tell you whether we would like to be paid by shoes, or we would do it for free.

If you liked the session, and even if not, please fill out the survey. Let me know what you thought about the session. Again, I hope it was helpful and interesting. Thanks for being here, and enjoy.