LUCY KUHNS: And it's 8 o'clock. So welcome. Let's start. This is the Data Lab. I mentioned it was supposed to be a hands on lab, but they turned it into an instructional demo. So hopefully, with looking at these examples, I will also post the examples. So if you want to practice or try this on your own, you're welcome to do so.

The agenda and the class description is I want you to be able to see the possibilities of data within the Autodesk product line of Civil 3D, Navisworks Infoworks. Why is data so important? Because we need to have data for the BIM process. And it's all about the data. You've heard that term over and over again.

Now, I started my career as a land surveyor. And when I collected data, when we collected it, we took all the attribution that we could. We marked everything that we had. And that data then became more valuable as it lived in our database and in our drawings. It's a requirement for BIM readiness, especially for infrastructure.

The data exchange. The more information you have, again, the more possibilities for data exchange amongst different applications and different specialties. And of course, the future possibilities. You might acquire data and put attribution on it that you don't need now. But in the future you possibly could do it.

Now, about BIM. When we have a BIM model, we need to know things about the pieces and parts in that BIM model. It has to be attributed. It has to be located correctly. But we have to know what that data is, and what it's about, and what it's used for. Here's an example of a virtual construction simulation. The simulation is done so that we can make the overpass, the bridge through the median, without disturbing any of the traffic side by side. So if we model the elements in the computer, we will be successful in the real world.

Now, here's another model where we're doing time timelining, and a model where we're doing an automated clashed detection. Whether we're seeing where we're going to have some problems. Now here is an interesting-- is mike OK? Oh, dear, that's no good. Much better? OK. Thank you.

In this example that you saw here, what we did is we had a federated model, and we did a quick clash detection. Now the beautiful thing about a proper BIM for infrastructure model is

that we're able to switch back to the authoring tool, make changes, and then go directly back and forth into the models. It's called switchback.

And the requirements for a BIM ready design is, of course, geometry in a precise location. That geometry has to have attribution. And that's the main purpose of this class, is to show how we put attribution on the various forms of geometric data. When that geometry has attribution, then it's smart. It has dynamic behavior.

Here's an example of dynamic behavior. And we saw it. I'm a pipe. And I have to live one meter beneath the surface. So when the surface changes, I need to change. That's dynamic behavior. I'm a pipe. I have to connect to a structure. That's intelligence and dynamic behavior. If that structure moves, I have to move. Then the criteria and rules that we apply to that dynamic behavior, the criteria I mentioned was one meter beneath the surface. That's an example of criteria.

Now, when we exchange data, we want that data to be agnostic. In other words, that data has to be able to be used in Autodesk products, in Bently products, in any other products across the BIM portfolio. We have to be able to comply with standards and things like industry classes and CoBie. And the example applications that I'm going to be using today are the Autodesk applications.

So the attribution, and, again, this class is all about attributes. And I mentioned that we have geometric attributes, which are called shape attributes. And we have characteristic attributes. Now you guys that have done AutoCAD, the characteristics of CAD are what? Layer color, line type, things like that. Layers are very, very important attribute. It's one of the few attributes that carries across all applications. And if you're smart about using layers, that can be a very powerful attribution.

We're going to talk a lot about these smart extended attributes. That's the extra data that you put on the geometry to make it work in a BIM federated model, and have that sense to work as a system. For example, the cost, the material, and the phasing. If I had the phasing, then I can put it into a system of timelining for virtual design and construction.

So spatial geometry and attributes. So we have the shape and the geometry, and that's what we see in a drawing, in the drawing editor. Those are the things that I can grip and I can dimension, and things like that. Now, the tabular data, that is the attribution that we're going to attach to the geometry. And the tabular data it can be automatic. If I change the area in the

geometry, that tabular data should-- the actual numbers and values should change. So the tabular data is smart and it's also those static characteristics.

OK. Here's a process table. How many of these have you seen in this week? When we are doing BIM work, we are consumed with creating process maps, manuals, procedures. That's all within a BIM application in a BIM modeling process. Well, I've cut this down into what we're going to show today.

I'm going to show you, and we're going to start with, the very, very basics of adding attribution to data. And that's going to be layers, like I mentioned, hyperlinks, and notes. And then we're going to start to talk about blocks with attributes, very, very important. Then we're going to get into object data, and then property set data.

Now, the reason I have this form up here is to show you which type of data you could put on which type of objects. So for example, 3D solid objects, you cannot put object data on them. You have to put property set data. When I say object data, you're going to see what I mean. And so that's why we have this snapshot here.

So let's begin with the basic attributes, the basic attributes of layer, hyperlink and notes. These are available on all AutoCAD Civil 3D, Map 3D, any kind of AutoCAD vertical. And these are available on all of the geometry, and they carry through the entire process. They carry through from your authoring tool to your BIM federation tool, which might be Navisworks, and also Infoworks carries this information. And like I said, layers are one of the easiest types of attributes that you can use.

So here is a hyperlink attribute. Here is a lay out for a new station, for the Grand Paris on the outskirts of Paris. So that is a hyperlink. And the hyperlink, this will live throughout all of my process in the authoring tool, in Navisworks as we see here. And there we see the data table that shows that this has carried through. Will it carry through to Infoworks? Yes, it will carry through to Infoworks as well.

So hyperlinks-- now, hyperlinks don't always have to go to a URL. Where else can they go? A hyperlink can hyperlink to a document? It doesn't have to always go to URL. In Naviceworks this is how we expose them. And you can see in the properties column over here how we can get to those. And we can see there's a hyperlink that goes to a document, not to a URL. All right?

Now, when we use multiple hyperlinks here is a caveat. It will only-- work if you can put as many as you want on, but if you need to carry it through outside of the authoring tool, it will only take the top one. All right? So just a word of caution. We can also use notes. That is another valuable type of attribution.

Here isn't very basic. Now I like to use super basic examples because I want you guys to think out of the box. And if you understand how a basic attribute works-- and this is a note-- then you're going to be able to apply this as we get to more complicated attribution examples.

So this is super simple. Three boxes. I'm going to put the notes on them. And the note is simply construction 1, 2, and 3. And then you can automatically create a time line. Did you know that you can have a very complex drawing and hit the Automatic, and it organizes them? And I'm organizing them by the number in the attribute. And so then it automatically makes a timeline. So that gives me a rough Gantt chart that I can either edit outside the application, which might be easier, or inside the application.

For those of you that are in infrastructure and in Civil 3-D, and understanding what that picture is. And that is a very simple assembly. And as part of that assembly I'm showing what's called a shape. And in the shape name, the shape name can carry through vis-a-vis a layer. Now, this is an important thing to remember. How long can a layer name be? Anyone know what the limit is on a layer name? How many characters?

AUDIENCE: 256.

LUCY KUHNS: Yeah, 255. One under 256. So that means you can put a whole lot of things in there. And the shape codes carry through. So now this is interesting because when you design this assembly to use over and over again, what if in that shape code you mentioned things like material? And you mentioned things like, oh, it has to come after the base course. It has to come after this and this. And then we can pull out those.

So here's where we take it. In the name, template. And I make it a code. So that's the shape code. Now, remember you use your assemblies over and over again. So now, when I look at that pavement group 03, I can use that number three in a construction sequencing process. All right? It's a really nice little thing to add in your assembly work.

Does that make sense? Anyone do that? When you make your shapes-- I know you do, you certainly have to use logical naming. That's incredibly important when we add attribution.

So now, we started super, super simple. So this is going to be a progression. We're going to get a little bit more interested in more complication. And you might say blocks with attributes. Oh. Blocks with attributes is one of the most important things that you can leverage. And look at the blocks with attributes works on all of the primitive AutoCAD entities.

The blocks with attributes, here's the pros. 2D, 3D. They're held in the drawing database. They're transportable across all of the portfolio, also inside and outside of Autodesk products. You can easily export those to ASCII tabular files. When I say ASCII, that's good, because it's not compiled. It's not binary. It's readable. Again, across the broad spectrum.

It's scalable. Blocks are perfect for levels of development, levels of detail. I could start with a simple block as you're going to see, and as I start to understand my design more and more, I can refine that block. So it's a perfect use for level of detail. Automation. Let's say that I have to place 1,000 things in the drawing, and they're all the same, and they're equally spaced or equally gridded. Blocks you can automate this to make this very quickly.

Here's a con. You can't use parametric blocks in this attribution because, if you use a parametric block when you share it, you're going to see those asterisk U blocks. How many have seen those? Have you seen those where you do the Purge command? All right. Those are the results of parametric blocks.

So here is how we can take advantage of blocks for automatic placement. Here is a block, and the block has lots of attributes. I don't know what type of lamp this is going to be. I do know that it has to be placed along a path. So I have some placeholder attributes, auxiliary 1, 2, and 3 for future use. There is a pathway. And let's automatically-- I'm going to use tools within AutoCAD that will allow me to automatically place that block along that path.

It also can sense the rotation if you want. And that has saved me a little bit of time. The block is in 3D. That's one of the primary requirements of BIM for infrastructure, is that it has to be in 3D. So we've got the block information or the block itself in there. But all of those blocks are block 100. That's not going to do.

So what we're going to do is we're going to leverage the sink-- my favorite name of an AutoCAD command. It's called BATTMAN. And we're going to use the BATTMAN command. And the BATTMAN command is going to slap these attributes on all of those blocks. So if you had 1,000 blocks in the drawing, this would automatically attribute 1,000.

But like I said, they're all the same number. Now that's not going to work for us. So that's light 100. The next one's light 100. So I'm going to show you how we can automate this by using this export attributes. So ATT out. Now, when I export the attributes, this is a nice thing, because I have now done what? I have geometric attributes. And look, I have the tabular attributes.

So instantly, from just plain AutoCAD, I have a pretty nice system here. So this is Excel, and all I'm doing is I'm just importing those attributes into Excel because I want to take, and I want to make all of light number 100. I'm just going to start a sequence of 101 102. Once I get that sequence down in Excel, you know you can just pull down on the little right black box. Just pull down. And look, it's automatically numbering those.

All right. Now think about this. If you had 1,000 things, this would automatically do it. All right. So I've got external data, and I have the blocks in the drawing. We save this file, and then we have a command attribute in that brings the data back on the blocks. OK? So there we go. I use the Attribute In command, pick the text file that I just made, and now I have the blocks automatically numbered.

So sometimes it's easier to do things automatically externally, and blocks allow this process to happen. So just a quick review on these commands that you can use to automate lots and lots of geometry in a drawing. The attribute dialog, which turns the attribution on and off, and of course the BATTMAN command. I'm just splashing these up. I'm not going to go into them. I just want you to know the names of these commands.

When you create attributes on a block, and you can have many, many, many attributes. And as you see here at the bottom, I have three auxiliary attributes. That's for future use. I'm thinking that these blocks are going to get more sophisticated and more intense. So I just have temporary placeholders.

The reason I'm showing you this is when you create a block, in the process of making a block it says, select the objects to make a block. If you use a crossing window, you don't know what order the attributions are going to be sequenced in. So here's a tip. When you make a block, you pick the order of the attributes that you want to happen.

Why? Because later on, when I automate this, I generally want ID to happen first, type second, so on and so forth. OK. So just a tip, pick the attributes first, up, up, up, up, up, and then you can pick the geometry.

OK. The LOD concepts. That's why we have this auxiliary future placeholders. And the BIM advantages, like I said, it's in 3-D. They can take many, many attributes. They can be georeferenced. And we can use the block replace function-- it'll keep the attributes, but the LOD changes. And you're going to see these advantages right now.

So here are the blocks. They're on a path. One of the requirements of BIM for infrastructure is that these items have to be georeferenced. And they have to be georeferenced because these are going to be built on Earth. So there's the lat and long. OK? And for the lat and long, and we're also in the British National Grid, so I'm on a projection as well.

And let's see. I'm going to use the BIM Maps. You can tell me where this is. I said British National Grid, so I gave it away, but let's see where this project is.

AUDIENCE: [INAUDIBLE]

LUCY KUHNS: How did you guess? In honor of Jack' [INAUDIBLE]. OK, so you can see that this is in England. All right. And this is just-- because if I'm connected to the internet, and I have my Autodesk account, I can use the big BIM maps as my underlay. So that's requirement number one for BIM for infrastructure. Well, other than it's 3D. It has to be georeferenced.

Now, there's the block replace. I'm demonstrating LOD aspects with a block. We didn't know what this thing is going to be. But now I have an idea that the shape envelope is going to be that. This thing that I'm going to pick out later is going to take that shape and, oh, it also has a globe on top. And so, do you see how easy that I can take something I have no idea what it's going to look like, but I can switch it out?

Now, here's another interesting thing. Those blocks are smart enough to live on a surface. Blocks know that they can move up to a surface. And there is the end result after I know what the block is. It's georeferenced. Blocks know that they can live on a surface. I didn't have to assign the elevation. I just said, block, go to the surface. Bam. And it does it. OK.

So why do I love blocks? They do all the work for me. It's an amazing, amazing thing. So here's the surface awareness. If you have a surface, and if you have a block, there's the tool. Move the box to the surface. OK? Anyone ever use that? Yay. OK. Lots of hands go up.

Now, I can also take advantage of this. Here are some 3D houses. There is the grading, the pad grading for the houses. They come right up onto the site. And now I'm just going to project

them onto the profile view. So those are 3D blocks coming up to the surface, and then being projected to the profile view.

Update and date awareness. So I'm going to put an attribute on this block. And this block is going to have the date. And that's the date of the last inspection. All right. So now I'm thinking about maintenance. I'm thinking lifecycle on my block. Even though I'm just starting the design, I'm thinking to the future. And now we can see that I have a date. And depending upon how old the last inspection was, I'm going to assign whether it's stable, critical, or if it needs to be checked.

Here's a little simple LISP routine that is literally going to check the date on the computer, and run against the data on the block, and change its color. So the color of the block then will change, whether it's critical it needs to be checked, whether it's stable. Whatever it needs to happen. OK? So I'm purposely thinking about the lifecycle use of-- the future use of this block.

So now when I have this as a real tool, I can-- it's georeferenced. It's colored, which means it's one of the characteristics. That's super simple. I can put it on a mobile device. And now I know which one of these light posts to click Go To with the lat and long, and check for maintenance. OK? Very, very simple, but very effective. And again, starting with blocks with attributes.

I just put this in the slide deck just to remind you of the commands that you can use so that you can download and use the slide deck later. The data extraction that I've shown you and I've shown you this because there's another form of data extraction that we used to use, and it was called ODBC. Has anyone ever heard of that, where I have external databases, and I try to connect them to my drawing? Used to work pretty good, very easy.

Now, with the advent of Microsoft drivers and 64-bit Office applications and Office 365 and all that, this is becoming-- the ODBC thing is becoming-- harder and harder to do. But I just showed you how to do it very simply by sidestepping the ODBC thing. And that's with these commands right here.

So here is a different extraction. I showed you the one that started with an A. This is the EATTEXT, which means extruding the attributes. Now I'm taking this out as a Excel spreadsheet directly. This is the one that we used to use with the ODBC Connect. I'm just showing you this in case you do want to do ODBC Connect.

You've probably-- this has been in AutoCAD for ever. So who's used this? OK, excellent,

excellent. This also allows me to instantly put tables into the drawing. When I put a table into the drawing, you know that table is live, and you can do things like sum columns and add rows and columns and do formulation in AutoCAD.

So here I'm just taking the time to set up this table, and then decide whether I want to put it in the drawing as well as have it externally, and the output format, whether it's CSV, whether it's a Excel spreadsheet, or whatever, or the old fashioned Microsoft Database.

Now, when the table is in the drawing, unfortunately it's not a live link. OK. I'm going to show you in our next example how we make a live link to something like a table and the geometry in a drawing. And there's this table. So instant bill of materials or whatever.

All right. Now we're going to take that block example, and we're going to turn it into something with object data. Now, that means that the geometry is going to have a tag on it that has object data. And you're going to see how much more powerful this is. Remember, I'm saying we're doing this as a progression. We're starting very simple. I'll show you blocks and the simple extraction of the data from the blocks. Now we're going to up it to a shape file, and we're going to use the Map Export.

So I'm starting with my 3D block. It's got the attributes. Now, this is the trick that we have to do with Map Export. That's the command. Who's used Map Export? Excellent. OK. So more than half of you. Ubiquitous shape files. So most of your software-- your spatial software, for sure, and a lot of your BIM software-- can read a shape file.

So I have to understand that these are points. I simply collect the points. After I collect them, make sure when you do the command that you make sure that they collect and that are not filtered out. So I look at the bottom there. And now, by clicking in the attribution-- these are all the physical attributes, the characteristics of the CAD entity. But here are the block attributes. So we have lots and lots of automatic attribution with the characteristics, and then the ones-- the attribution that I added in

OK. And so I've created that shape file. And the shape file in any of the CAD software I can drag and drop that shape file information in. I can go to the Map task pane. And in the Map task pane, now here is where we have-- this is the data, the tabular data, and the geometry. And now they are linked automatically simply. OK. So that is the method to do because it's a good linkage from the external tabular data to what's in the drawing.

OK. So in summary, the attribution order, how you pick them, is important. Don't use parametric blocks. Remember, this is good for LOD techniques. And there's several different ways for extraction. The best way of linking is using the object data and making shape files. Data connection strings are going to be a little bit difficult. Data connection strings mean ODBC. The shape files are good in AutoCAD and Infoworks, but shape files do not work in Navisworks, but regular blocks do.

OK. Now, we saw object data on blocks, which is pretty interesting. Object data, the pros. And you've seen it, a little glimpse of this. They're held in the DWG drawing database, so that means I can share them with all of you. And you get those data tables. They're transportable externally. We saw that with a simple shape file.

They're scalable. I can take object data from many, many sources and, as long as the attribution and tags are the same, I can join them. I can merge them. That means that they're scalable. So that's a very, very important thing to think about. And I'm going to show you some automation methods of how you get object data on hundreds and hundreds of little pieces of geometry.

Con is you have to set up an object data table. You have to make a list of what the attributes are. But that's a one time setup. The other thing is with the object data we call it 2 and 1/2 d. Depending upon what application you're using, sometimes shape files are not full 3D. But did you know that Civil 3D pipe networks can be held as shape information?

And then there are some geometry limits, some sophisticated geometry spirals and things like that would be difficult to do. And it does require-- you can't use base AutoCAD. You just have to use Map 3D or Civil 3D or Geospatial. And in GIS, when we're making this object data information-- again, I said you cannot use it in Navisworks.

OK. So the first thing we have to do. Remember I said there's a little bit of setup required. Now, I know many of you are pros at this. And I'm sorry to make you sit through to see how you make an object data table. But an object data table, I'm going to switch to the workspace for Map. And go to my Map set up.

And I am making this video in case you want to-- if you've never done this before, and if you want to try this-- I'll give you the, like I said, the base drawing files. I'm going to name this table. And then I need to give the field names. These are the attributes that I'm going to attach to the geometric shapes.

And once I have all of this data done, and you have to give it the data type, whether it's a double, whether it's an integer, whether it's a string. A string is a piece of text. And give it a default value and things like that. So here I'm going to make a single attribute of a building and the height of the building.

OK. Now we have to put the data on the buildings. So I need to attach this object data to the buildings. So now, what happens if I have 1,000 of these buildings? Years ago, there used to be advertisements for geospatial analysts. And guess what they did? They sat all day and put object data on things like the building. So this could take forever. But we need a way to make this happen really, really fast.

So now you see that with one of these I have the object data. You see that I put the building height of 25 that's held on that piece of geometry. Well, this can be very painful. And I don't want you to sit the rest of the class while we put object data on every one of these. So let's look at a fast way to do this.

So if I have a little piece of text, and you can do this for different types of attribution. So there's the text. That's a label. And that's-- in this case it's the height of the building. And as long as it's inside that shape, I can cast it to the perimeter. All right. This is a really old trick in Map 3D, where you picked the data table, and it will sense the text inside, and now it's been applied to that object.

So if we look at it, we can see that all of our buildings now have instant object data attached to them.

AUDIENCE: [INAUDIBLE]

LUCY KUHNS: The question is, can I erase the text afterwards? Yes, you can. Because that text was just cast in its place. Now, on the object we don't need that text anymore. And that's an interesting point because, once you erase that text, what happens if you put a different text in there? Like the number of floors or the name of the building? Guess what? You just run this again. And all of a sudden you have 1,000 things automated and someone didn't have to do it. So thank you for the question. Yes, it's not directly linked to that text. Once you've used it, it's done.

So now we can extract that data using the Map Export command. You saw me do the Map Export command earlier. And this time on the Map Export, remember GIS. You have certain

things. You have a point object or you have a polygon object. And in the middle you have a linear object.

So we're going to make these shape files. We give the shape file a name. It's going to create the shape file set. So if you're asking for data and someone sends you one shape file, are you in business? No, you need the shape file set. OK. So there I'm picking that this is a polygon. And I'm going to collect all of those. And now we have a shape file.

What can I do with the shape file? It won't go into Navisworks. What will it go into? It will go into Infoworks. Here again, I can take full advantage of the default characteristics of the object. And then we'll also look at the attributes that we did apply.

AUDIENCE: Can it do this to any object?

LUCY KUHNS: It has to be a closed polygon.

AUDIENCE: Can it be like a pipe part?

LUCY KUHNS: If the pipe part is a closed polygon. If it's a 3-D pipe part, no. We have to-- we're going to look at that in a little bit with property data sets. OK. So there is that. And let's move that forward. And we've got, in this case, 38 of these done.

OK now let's see where we take this. Of course, this will link back and forth, as you saw before. We drag and drop the shape files in. And now, again, we go to the Map workspace. And we zoom to the objects. And we show the data table. And we see that the data table is properly linked. And we have the area attribute that was pulled from what? Pulled from the physical characteristic of that. OK. So there's a good link between geometry and a tabular data form.

Now we can see that this is in 2D. And now let's take this into Infoworks. So we have the 2D shape file. Now, when we bring it into Infoworks, what are we going to do to the shape file, to the footprint? We're going to use the attribute of a height to do what? Exactly. So we're going to tell it that it is a building because we want it to have characteristics of a building.

And we'll just drag that shaped file in. And when we attach it, look, it has within the properties the-- not only the ID, the custom or unique ID that we attach to it, but there is that attribute that we threw on it from that very simple text. And I'll give it a rule style of nothing, just the plain white shape mass, just so we don't offend any architects with a facade that we don't like.

In the geolocation, if I want to, I can drape it onto the surface again, so it's surface aware. I don't have to think about the elevation. And we'll close and refresh. And we'll zoom in there. And you can see that this is quite effective for making our little neighborhood with the correct building heights.

And I'm going to do the Zoom To. I select on them. And the Zoom To in Infoworks. I'm showing you this because it's not as easy as the Zoom To in any of the other CAD products. And there we have straight from super, super simple lines to something in Inforworks. OK. So that is a very interesting primitive workflow that ends up in some pretty nice results.

OK. We can combine multiple data tables. Like I said, I can put in exactly the question you asked. If I take out that one text, I can replace it with another text. And in this one, I'm going to turn 2D in our drawing into 3D. OK. So this is the mapping shape files.

Now I'm taking these out and I'm going right back to our original data. And what I'm going to do is I'm going to drag in a little lisp code. And it's like seven lines of lisp. And I'll show you what this lisp routine looks like. And I'm just going to drag it in and drop it. And what is it going to do? It's going to look at the text and, in my CAD drawing, it's going to pop up those shapes to--

I'm going to drag in this lisp routine. When you drag in the lisp routine, it's going to tell you what command it makes. Those are the southern lines of lisp, very, very simple. Look what it's looking for. It's looking for that tag called HT. The reason I'm showing you this is you can take this lisp routine and use it for your own stuff, and make it do things to your object data.

The trick there is how you pull object data with a lisp code. Sometimes that's not straightforward to do. So that's the reason I want to show you this Lisp routine. And now I run the lisp routine, drag it in. Tells me that the name of the command is Height. I type in height. And it's going to automatically do that.

OK. So some more automation. And there it is. Put it in conceptual view. OK. The best of all worlds, in CAD and in Infoworks. And again, I show this to you because that one line to get the object data record table, that code in lisp is something that you might need to use for other purposes.

OK, just a reminder. When you have object data in a drawing and you do edits, be careful who's ruined the data table before with improper edits. Be really careful if you start to--

remember we have the geometry in that data table. And if you mess them up, then you've ruined your data tables.

So just the last bullet point. Word of caution. Do not ruin the data table. You have to do the proper check out of the geometry, extract it, and then check the changes back in. All right. Now let's up the ante a little bit. Let's get to property sets, which is a little bit more sophisticated. These are where we can put property sets.

Now, why are property sets more valuable? We saw object data was only limited to the core or the vanilla AutoCAD entities. With object data, we can put them on to our specialty objects, like the architectural objects, the civil objects, and things like that. In property data sets, this is the place where we find them.

The definitions. We have definitions that are manual, that we have to type. In I'm going to show you ones that are automatic, and a few little lines of VBScript so that you can do automatic property set definition. We can also use AutoCAD Fields. Who remembers AutoCAD Fields? Does anyone use them? OK, good. Good. I use them in title blocks all the time. That's probably why my main use is in title blocks.

I'm going to show you how you can save and share property sets, how we can use property set information for labeling. And again, at the bottom, the property data set transfers through to Navisworks, and hopefully some they'll get through to Infoworks.

So for example, here are objects that can hold property data. I can have my COGO information, surfaces, corridor solids, plain old AutoCAD solids, and other Civil 3D objects. So this additional property dataset information, it's a great addition to the things that we can do. And you've probably, if you've ever-- who's applied property data in here? OK. Oh, not so many people. All right. You're going to be amazed at the list of items that you can put property data on. And I'm going to give you a reason why you don't check all.

OK, so here is an example of where I want to use property data. So there is the way that we can create property data. There's a new property data set that I'm going to create. And I'm going to call this construction information. Now, when it applies to so many people do, they just select all. Now I want to understand where I want to put this property data. And I want to clear all and be very succinct, and know what I'm going to apply this to.

And I'm going to apply this to my 3D solids primarily. And I'm going to do it to solid 2D What's a

solid 2D? It's a body. Sometimes when we do corridor extraction you end up with bodies instead of solids, right? Because it's a link and not a shape. So here is a manual property applied-- I'm only applying these to 2D and 3D solids. This is a manual property. It's simply the ID.

Now, I am go to autoincrement this ID. In other words, one, two, three, four is going to automatically put a ID number-- a unique ID number-- on it. Now the next manual one is simple text. And that's going to be just an ID. And that could be-- as we develop our BIM libraries that ID is going to be very important. Correct? It's going to be our Unicode ID or something.

And now we have these two manual IDs. Let's put in a third one. This is just the description of our object. So that's pretty straightforward. Putting in these manual things. And you can put as many, of course, as you want. But it gets more interesting when-- and I'll put in the default here for the description, this is a single family residence, a single house. And we'll see how this is going to look.

OK. Now, this is the automatic attribution. Now, this list here is the reason that we only picked 2D and 3D solids. If I picked everything, this list would be really, really long. And it's difficult to select everything in those lists. So here I am pulling the automatic characteristic of volume from that 3-D solid. And that's going to give me the volume of the house. And then I'm going to put in a date. And the date is going to be in here a field.

So even though that's a manual, it is going to be automatic because I'm going to use the field property to pull the date out of the computer. OK? So there's an interesting place where a manual attribute that we think of as being static, if we put an AutoCAD field in there, it can actually be of benefit to us.

And so now, when we put this information on here, I'm selecting all of the solids. Here's how we apply. We have to put the property data set onto these. So in the lower left corner-- this is hidden quite nicely, so tiny-- in the lower left corner, you select-- after you select the items. I'm selecting all the solids in my model. And I'm going to make sure that they can use and make use the property data called construction information. Maybe some of the solids I'm not going to use, you know, put on there. But it's just easy for me to cast that information on all of the solids.

Now I can be very specific. And in this case, I'm selecting on a solid. There is the property

data. And I can edit it within the properties data box. Or-- that's probably the easiest way to do it. OK. So now I'm changing that name to utility duct. I can select many of these all at the same time. If I select them one at a time, and I have 100 utility ducts, that's not very efficient, is it?

So we select all of them. And I can cast that property data onto those objects. OK. Now, only a handful of you have used property data? Correct, how many? One, three, OK.

Now, here I've selected some of the extractions from my corridor. This is a sheet piling that was part of my corridor design. And this would be like a retaining wall. In this case, it's a big sheet metal piling that runs along a riverbank. And we used that same property data that I originally created for the 3D solids of the house. But since I checked that it was applied to any 2D or 3D solid type object, I can apply that to any of the corridor data extraction objects.

All right. Now, what happens if, by mistake, or you need to change something, you need to remove property data from the objects? Right? It's not straightforward. But let's look at how we would remove property data once we have the property data attached.

So I come in here to the construction information. And that's the one that I'm going to eliminate. I'm going to select my objects. And, in the properties, I'm going to make sure that I get-- and again, this is why I want to assign property data and not to all of the entities, but in this case just the solids. And there I have the property data-- or these are the objects that had that property data. And I can remove those. And then I'm going to use--

So I'm going through each of those solids. And I'm looking to make sure that there's no property data attached to them. And now I'm going to use the Property Set Clean command. And once you have the Property Set Clean command, you name that property set that you wish to delete. And now you can see that I can purge it from the list. OK? So that's how you get rid of property data sets. You have to take it off of all the objects that you apply it to. Again, a reason not to use that Select All button when you have the option to apply property data.

OK, how do we share property data sets? This is kind of a nice utility that they have in there, is I can send them, once I make them, or the other way you can do it, you saw the bullet point was a DWT that I can save out. So sharing the property data. So here is the property data that I have on the sheet piling. There's the property data set.

We can see that I have several here, that I've put in some corridor shape properties. And in my property dataset manager, you right click on that particular collection that you want to, and

you just send it. Right? It sends it out as a DWG. It's a blank DWG. You send it to yourself. And then all I need to do is just drag and drop that DWG into any other drawing. And now you have that property set information in there.

When you drag and drop it in, it'll ask you for an insertion point, just type 00. It won't put any physical objects in your drawing. So it won't affect your Zoom Extents. But it will give you now access to all of those property data sets. And you have to do them one at a time. Of course, you can save your drawing as a DWT, you know, erase all of the geometry, and it will also hold the property data sets.

OK, to insert, here is where I take that drawing, drag, and drop it in. In this case, I'm just using the Block Insert command. OK? Just showing you that you can put it at 00. Doesn't do anything to the drawing. And there I'm putting in-- notice that it doesn't give them a name that is highly recognizable. Maybe when you send them out, you want to change the name.

And there are two property data sets put into a drawing vase. And they show up in the property data sets. OK? So no geometry, just the information. So that's how we get the property data sets. All right. Now, let's look at how we create a formula in the automatic collection, or the automatic category. And here I'm going to take a feature line from a corridor, and we're going to apply property data on that feature line.

So I'm going to list the feature line. And one of the unique things that we see in the feature line is it has a 3D length and it has a grade range. So let's see how we can slap that onto the feature line. So I've got to go to my property datasets. And I'll make a new one just for this one. And I'm going to call it a formula example.

And again, this is a skeleton example I'm using a feature line. Here, again, I'm careful not to select all of the things, the possible items I can use, because I want to limit the type of entity that this is going to go on. And I'm going to put, of course, a unique ID on there. I'm going to have it automatically increment.

And then there's another ID that we must have, and that is the entity handle. OK. Everyone here knows what an entity handle is. When you create something in an AutoCAD-based product, it has a unique identifier. And that's held from draw-- every time you open and close the drawing, it's the same unique identifier.

So in my-- I'm going to back this up just a little bit so you can see where we started. So now

I'm making an automatic formula called length. Now, this little code snippet we're going to look at-- and I'll give it to you. You can download it off of the site for this class with the PowerPoint and the other information. I'm going to use the handle in this to make this work. And then I'll go back and show you this in more detail.

So the properties-- remember, we have to select the item. I have to make it work in the lower left hand corner. I'm going to attach just the formula example. And there is the 3D length of that item. OK. So a very simple code snippet. Let's add a little bit more to this.

Now, in the code snippet, you have to know which version of Civil 3D that you're using, because it reads the document object. So 10.5 for 2016 and 11.0 for 2017. And this is the code snippet that we're going to use. And here is the characteristic, the object property that I can pull from that entity.

I'm going to show you how you can use a little command in Civil 3D or an AutoCAD to find the list of everything that you could possibly pull out to put in these snippets. The trick is, when you're creating these, that you have to put the active link to the handle in there.

So when you copy clip, I'm not going to expect you to type this information in. You're going to copy clip these snippets in. And once you copy clip the snippet into the editor window, you need to highlight the word handle with the quotation marks, and then you need to go in the lower window, and you need to double click on a handle. That makes a live connection to that handle. Because if you don't do that, your automatic formula won't work.

So let's look at this example again. I'm going to put in this little helper tool called Dump. It's a little lisp command. Now, the reason I'm going to show you this helper tool is because you can pick on any entity-- type in Dump, pick on any entity, and you go to the top of this list. And it tells you, any one of these items is the thing that you can make an automatic property from.

OK. So length 3D, length 2D. Max grade, min grade. Well, guess what I'm going to do? I'm going to use-- we've already done the length 3D. I'm going to grab one of the grades, the min and the max grade, and I'm going to put that in as a property data. But you need to know what the name is. And you need to know the objects-- these items that you can use.

So here I go. I'm going to do the formula. This is the window that shows up. I copy clip that snippet in after I name this, and we're going to do maximum grade. Then you take the code snippet from this class. There is that magic OBJ period and max grade. And I got max grade

from that list after I dumped the object information. Here is where I had to click on the handle to make it work. And now, when we look at the object properties for this, we've got the max grade in there. OK.

Now, if I look at the surface properties, here is some extra information that we can get from a different type of object. So I can create an automatic label on a surface object. I need to create a new set. But let's look at this because it's a little bit of a different code snippet and some different lines in the routine.

So we'll call this surface information, applies to it. And again, not everything, is only going to apply to what kind of surface? TIN Thank you. Because there's also just AutoCAD solid type surfaces. So make sure you get that. Who knows what TIN stands for? Thank you. Triangular Irregular Network. Does Revit work with Triangular Irregular Networks? Does Revit? Anyone?

AUDIENCE: [INAUDIBLE]

LUCY KUHNS: It does? But does it have cut outs and curves?

AUDIENCE: No.

LUCY KUHNS: That's the problem, right? It goes across. So we always need Civil 3D. Right? All right. So now here's the snippet for surfaces. Notice that the bottom line is OBJ point, and not the name of the object. But we had that extra word in there. Statistics. OK. So that's how we're going to grab that data from a surface. So I'd given you an example to use with Civil 3D objects. And ones that are little more sophisticated that have the statistics for the surface.

So let's apply that to a surface. And see what it looks like, surface information. Get the handle. There is that OBJ with the statistics. The area 3D. And we're going to apply it. And let's see if it works in our properties. OK. All right. We have the area 3D, automatic properties and the data sets, the property data sets.

Again, just to review. I showed you you can get the Dump lisp routine from the class site. And these are the things that you can apply.

AUDIENCE: [INAUDIBLE]

LUCY KUHNS: Yes.

AUDIENCE: [INAUDIBLE]

LUCY KUHNS: Oh, yes. In the applications. 10, 10.2. OK so there is the one for surfaces. And again, this is the key. The rest of the snippet is pretty much the same. This is the key. Additional information. You can get as sophisticated as you want in these. And if you know VBScript really well, you can really, you can do a lot. And it's amazing. I'm just giving you the training wheels or the skeleton examples. And I hope that, if you need this, that you can grow it on your own, and develop it on your own.

OK, let's look at corridor extraction in Civil 3D. So corridors are the main way that we get linear design out of a project. And in this case, I have linear design that this corridors that I'm designing not only has the road, it has the yard for the building sites. It has the beach along the river. And it's got the sheet piling retaining along the river. So it's a pretty-- it's a big corridor. Does a lot. But it saves me a lot of time. And I'm going to extract the corridors solid.

Now the reason that we want to extract the corridors solid in-- I know I went through that kind of fast. But in the setup you can use the code name for the layering. Remember that from very earlier on today. And it's, again, it's incredibly important. I have property data sets for the corridors information. And it's the construction data.

Now we also have the corridors shape information, which in 2017 is automated. We have, just by default, we have some corridors data that is applied. Oh, here's a little trick. When you extract information from a corridors that's a link, it turns into a body, right? Now, a body is just like a sheet of paper and it's not good for us in a BIM model. So you have to use the convert to surface, and give it a little thickness. And now it becomes a 3D solid, a proper 3D solid, OK? So convert to surface, and then the Thicken command. All right. And that will get rid of those pesky bodies, and give you real objects to perform BIM. So that's a little side tip there.

AUDIENCE: When I click it, does it go inward or outward?

LUCY KUHNS: It depends if you do positive or negative. OK. And so now, since it's a 3D solid, I can put the attributes that I have assigned to be available for 3D solids. So a little tip. All right. Now, using property data. We can use property data in labels. And for example, here's a screenshot of this.

And this is-- using them in labels is an enhancement, a version one product enhancement that got sent out. [INAUDIBLE] went like two days ago, two weeks ago. It's been very recent, I think.

And here is the rundown on how to do it. OK. So let's read this together. No, no. Basically, again, understanding that what these code formats are, and the things that I pull out, PS, property set, the name of the property dataset that I created, like construction information, length, surface information, whatever.

And then that the name of the attribute, the name of the thing, like if I put the house number or whatever, the attribute, is very simple. And that's how it shows up in your labeling. So now we can pull this. And this is only in 2017 we can do this. If you just hack that in by hand in 2016, it won't work. OK. You have to have the enhancement tool that just came out for 2017.

All right. How do we use property data? We were going to use it in Navisworks clash quantities. And we can use the find items to create these sets that we do clash to create sets where we timeline this. And they show up in Navisworks. If I do NWC out, that's a Navisworks works cache file from Civil 3D, we can see that in our standard or on our Properties panel, we get all of the information. And we can expand on that and look at that.

OK. IFC Export. What does IFC stand for? OK, Industry Foundation Class. And this is a wonderful thing to use. What kind of entities-- what are the only entities that it works on out of Civil 3D? They have to be 3D solids. Can I have a pipe and a structure as a 3D solid? Yeah, if I turn up and do have a model view, it's a 3D solid, isn't it? if it's in a planned view, it's not a 3D solid. If it's in a 3D view, if I have it styled, it's a solid, isn't it? Oh, so IFC will work with that if it's in a model you.

But it the IFC only works on 3D solids. The IFC carries along the property date information. Does IFC carry along geospatial information? Yeah, it does. But does Infoworks recognize it? We'll see.

All right. So using property data in Infoworks 360. Ah, Dave Simeoni was standing in the back of the room. He's one of the product developers. I wish he was here to see this. Some day, some day, because all this data is important. It's important. We need to harness it and leverage it. So the stuff coming in to Infoworks, we'd have to hand-put all that information back on.

So let's see how this looks in Infoworks. OK, what's the most basic object in any infrastructure project? COGO points. 101, basic stuff. There's no easy way to get it in Infoworks, is there? I can't take a COGO point from a Civil 3D drawing and put it in the Infoworks, unless I write an

exotic script.

OK, so here's the trick. You convert it to COGO point blocks. You create a shape file. Remember how I said shape files are incredibly important? And then you import it as a shape file. So I'm going to show you how we can take some trees, and leverage them in Infoworks. All right. And-- these we're not playing full full size. Sorry about that.

OK. So here is my Civl 3D drawing. I have a tree inventory. And I have a tree type and a tree drip diameter. OK? So that's the information on the COGO point. I'm going to take the COGO point, and I'm going to turn it into a point block. So here we have the information. There's the point group. We can see it's a point group called trees.

I'm going to select all of the trees. I'm going to turn those into-- or we can see that they're in 3D. Oh, they're not on the surface. Do I care when I put them in Infoworks if they're on the surface? I probably don't want them to have an elevation. Why, in Infoworks? The drape option. So that so they can go to any surface I want. So I don't care if these are on the right elevation or not.

If I do in my Civil 3D, that's the way we can move blocks to the surface. Remember that's one of their neat characteristics OK so now, I'd move those up to the surface so they do have an elevation. You can see that change in the elevation from zero to a number. So I can use that number in Infoworks or elect to use Drape and not use that number, the best of both worlds.

So I'm selecting these. You could see they popped up to the correct surface elevation. And now we need to turn them into blocks. OK. I select the COGO point tools, drop it down, create blocks from the COGO points, select that group, the point group of trees. This is by default, it's a point block called point. And you can see it didn't erase the points. It's just put block information on top of it. Brings the number, brings the description, brings the elevation. All right.

So I have that. Those are blocks. So how easy is it to make a shape file? We use the Map Export command to do the shape file. And we create the trees. Again, we have to use point. Select them automatically. They're already selected. So just verify that nothing is filtered out. Make sure that you check on the proper data. And create a unique feel.

It's going to have a coordinate system attached to it. We do OK. All right. So now, let's import and configure this in Infoworks. OK, here is our site. We've got the new engineering view. If

you haven't seen that-- all right, now go to import the shape file. And the configuration is going to be interesting. In the Configuration tab, we'll tell it what type it is at the top. We're going to tell it that it's trees.

And now we're going to start to style them. First of all, the name is going to be the point number. So we drop down the list and we see those attributes that we have. And the description is the description of the tree. So, so far so good.

Apply the co-ordinate system. Here's where we have the Drape or Don't Drape. So we can either use the elevation off the point or, in this case, I don't want to because my surface might be a little different in Infoworks.

All right. Now in the table, this is where we can put in a user tag, or start to develop this a little further. So here are the trees. Are they impressive? It's pretty depressive. At least there are little trees. OK. So what we want to do is I'm going to go in and style the trees. And we're going to use a style. OK.

Now we've made them all the same tree the same size. But someone made the work to tell us that there are different kinds of trees and different sizes of trees. So at least we got trees in. That's pretty easy. Let's configure them. And see if we can make them look a little different.

So I'm using style rules. And I'm going to add a style rule. And I'm going to make a style for elm trees, oak trees, apple trees. And we're going to parse out in the expression editor. So we're going to look at the expression editor here. And we're going to build an expression. I'm going to use the description. And I want to see what the description property looks like. And I want to parse out the last part of that. So I have to use a string length calculator.

And so I'm going to come in here, and I'm going to look at my comparison functions. And in this case, I'm going to see if the word elm is contained anywhere in that description. And I'm going to say, is the description property elm located in that item? OK. So that's pretty-- that's an easy one. And so we can go ahead and the description in that is equal to elm. All right.

So now-- but, I'd put in four inch elm. But what happens if I have a 4 meter diameter elm? What happens if I have a different size? I'm only getting one in there. Let's see if it works. You always want to try these. Before you make a big huge expression, try the first part of it and see if it works.

OK, we have a slightly different tree. I don't know if-- it's definitely a different tree. OK. So what

we need to do now is we need to add to our style rules. We need to add for different trees. So we've done the same thing. I've made one for elm, apple, and oak. OK. So now we definitely have three different styles of trees because we used that description, and we used the comparison called [? in. ?]

Now, let's see how this works. Look in the categories for comparison. And when you do this, here's a trick, because there's not very good documentation on expressions anywhere in the help. So hover and this gives you the help. So it says, it tests to see if that property is contained. And then you can also look in your list of all of those properties that you have in the particular model.

And so we're using description in matches now. If I want to get all of the different diameters, you can see that I just put a comma separated. But what if I have 25 of those. That's going to be a pain, isn't it? So let's use some string manipulation tools. And let's just say anything that is called an elm is going to work. OK.

So in this case, let's look at how we would do this expression. OK, so this is finding an instance of a string-- a string is text-- and this is the explanation here that's telling us that it returns an integer number of where that word shows up in the list. OK. So if I have the word elm in the list, that integer is going to exist, right? If I don't have elm in the list, what is that entered you're going to be? Zero.

So I just put in a test condition that says anything greater than zero has got to be an elm tree. All right. So I do that with all of my different property-- with all my different trees. And that then accommodates any other description that might be. So we'll do that. And now-- I've got a different set in case I'm in the Middle East.

And there is my elm, oak, and apple trees. OK. And I can change it. See that you can toggle on and off the different style rules for those. So again, hover over the definition. And you saw the explanation. If that word exists in there, I'm going to get a positive value.

OK. Last thing we're going to do is the size. Now, remember we are going to have to think about how we get the size out of this because it's part of also that big long description. So in our trees, I want to go into our scale. All right. Now, I'm going to put the expression directly here, not through a style rules, but directly in the scale department. I want to parse out the number. And this is easy because the number is the first thing in the list.

So I'm going to use the Substring command. And recognizing that the very first position is always zero. And so I know that I-- the number is never going to be greater than the two digits. So I'm just going to take the substring of the first two. So zero in one, zero, and I'm going to stop at the position three, in other words-- not position three, but position two. In other words, I'm going to take three picks, 0, 1, 2, and then 3 is going to be a space.

So that's what the expression looks like. And I have to change it from a string to a number. And that's what we use that thing called ToFloat. Who knows lisp in here? If you know lisp, then this makes sense. But you have to turn it into a real number. And now I put that in for the x and the y and z. And we can see you just copy those over. And now I think we're going to have--

Whoa, we've got a forest, the giant's forest. Us And that's because we're not really sure what the size is of those little models in Infoworks. So in this particular case I just measured them and found out that it was one eighth. OK so I divide everything by an eighth. And there we have from COGO points in Civil 3D to a nice model in Infoworks for your tree. And we can have them as traditional trees, or trees in a different type of-- part of the world. OK.

Colors don't show up. I put the slide in just to show you about that string [? categorization, ?] the substrings, and how that code works. All right. Custom schema creation. Wow, we're kind of at the bottom of the hour here. This is like a two minute video. OK. On how we create a custom schema. This is important because when I bring a tunnel in the Infoworks, I don't want it to be a barrier or a street lamp or a house. I want it to be a tunnel.

First thing I need to do is I need to look where I'm putting my model on my hard drive because, remember, your model lives a SQLite file and a whole bunch of auxiliary files, so I put them-- I put them on a place where I've got a lot of space in my computer. Now, when you're doing custom schemas caution, caution, caution, make a copy of this so you don't ruin your model. Because you can ruin a model really easily, right, Vincent?

Not that you do. I just know you and I have heard some horror stories. All right. Now, this is a tool that, again, you can download off the web or from the class site. And you execute this tool. And there is a detailed video with the tool that shows you how to do this. I'm just running through this very quickly. The best way to do it is to take an existing schema as the default.

So in this case, I'm using city furniture. I'm wiping city furniture out, renaming it tunnel, and then I'm adding extra attribution. I'm doing the name of the tunnel as a string, the chainage,

which is a string, because it has a plus sign in it, and then the length of that. So I'm adding some attribution that's going to be part of that schema.

Now, when you save it, again, you have to know where your local model is. And it's called [? IM.Jason. ?] And it will just show up in that particular directory. And now, when we have something to bring in-- a 3D object to bring in-- we have it as part of the list. So that is the 2 minute overview on how to make custom schema. Again, there's more detailed instructions and the executable that you run outside of Infoworks that makes that schema.

Who needs the extra schema? Anybody in here? I know I've wanted it for a long time. All right. So we're almost at the bottom of the hour here. And in summary, we started with some very, very primitive things, and worked our way up to doing expressions and a little bit of VBScripting in our applications, to hopefully make your data more powerful and more valuable.

And I will be up here for questions for a little bit afterwards. But thank you very, very much. And have a good rest of the day, rest of Autodesk University. Thank you.