Scott Haselton: both of the webinar section that you see here and the time, we talked about issues such as how to handle service codes, multiple plans per file, providing an external and internal, and we talked about that have been seen while putting the data required for these files. Such as modifiers dealing with situations that a provider may be using. Their Social Security number, tax identification number. Service codes across claims.

Scott Haselton: If there are still questions that your organizations are running into that cover any of those topics, I would encourage you to visit the actual websites where you can download, not only the transcript, but some websites that provide examples to walk through some of those topics with more depth. We are going to be doing a recap of webinar four.

Scott Haselton: There is a lot to cover today and we want to leave time to answer the questions you have.

Scott Haselton: Provider networks and other in network files that can be reached by a table of content. Using these provider networks, initially this is how the schema was laid out where you have your network provider or provider group that was associated to whatever was being recorded -- reported on. This worked out conceptually for making sure those items were linked to the provider groups. While this worked, it did lead to some issues for the provider, many items and services. Where you could move the provider groups to the top of the schema and have the developers of the files come up with a number or an ID that could be referenced within the negotiated objects. This is why it is called provider references.

Scott Haselton: This left-hand box is where it started out and is blue blocks on the right-hand side -- this blue block on the right-hand side is where it ended up. Provider networks can be quite large. There could have been leased networks across multiple plans. The potential to find one's at that many times was certainly a thing. That is when the external provider network were introduced. On the left-hand side, this is where the internal provider network, this is where we left off from the last slide, where we were referencing the provider ID. The provider groups can be pushed to an external URL. It would be under this location attribute within the provider references found at the top of the file. That allows the definition of these provider groups to be moved outside of the file. Here you would still be referencing that same ID.

Scott Haselton: Now the benefit is many files could reference that location and take advantage of some of these optimizations. The URL would have to .2 a valid provider group. Which is nothing more than what we have already defined. These three things we just covered, if you're interested in clicking any of these links, I would encourage you to go to the CMS webinar page and go a little bit deeper.

Scott Haselton: The second thing we covered last webinar was allowing for multiple in network files to be included within a table of context instead of having just that single in network file. That table of context was created to soften the issues we were seeing why having multiple plans that have the exact same negotiated grace for the exact same providers, for the exact same items and services. Avoiding having those multiple plans have to report each dependently. This allowed for combining those plans together and the in network file. This is where that single in network file would be for reporting plans. For example, Medicare and Medicaid. The changes here would be to allow multiple in network files by changing this object into an array of objects. What you end up with is having the ability to split some of these in network files to wear just like external network groups, you can have multiple plans that use some subset of common items and services across many files. This just cuts down on the size of the files and allows for reuse. It is extremely beneficial if it is something that your organization can leverage.

Scott Haselton: External in network files need to be valid. They are just like any other type of in network file. That is it for external files. We covered on our last webinar billing code files. This is straightforward. There are times when a negotiated rate is different and independent of a modifier. We have added the ability to add modifiers. That attribute that was added was an array of strings. You can have multiple modifiers tied. If you are interested on what that looks like and a potential sample, go ahead and get on this material from the webinars and you will be able to dive a little bit deeper.

Scott Haselton: That brings us to today's this -- that brings us to today's discussion. A lot of negotiation types. We have added this concept of the recode and ultimately today, was the adding of this attribute called additional information. A text bill within the manipulation of a file. The negotiated rate, two new types were added since the last webinar. Through the feedback we have received, the percentage of billed initiative type was fairly difficult, if not impossible to fit with a dollar amount. Adding types to support reporting made sense. We had a new type called per diem which is really straightforward. This information has been available for quite some time so we don't need to spend much time.

Scott Haselton: Originally there were three bucket types derived. As we continue through this process, we found contractual arrangements don't necessarily fit neatly into one of these three buckets. So that triggered the addition of these new types to help drive down of those trying to more closely measure the types. The new negotiated types covered, this is pulled directly. If you want some more information, you can click through. The percentage mention to prove for -- to support the percentage type for a covered item or service for billing charges arrangement. The per diem is that dollar arrangement for that daily rate for each item and surveys. This is how one might use the percentage type for their percentage of bill arrangements. An example we have seen, in this case it is 35.7 whatever the negotiated rate is. To reflect the schema, you notice no new attributes were added to support this. Values were added to the initiated type to support this. Technically not a heavy lift in terms of having to worry about reformatting the schema if you are at a certain place in development. Pretty straightforward reporting.

Scott Haselton: It is implicitly referred that if a type is percentage than the rate itself is percentage. We have gotten questions around the percentage number, where should it be reflected -- or should it be reflected as a whole number? You should aim to be reflecting that percentage as a whole number. Taking whatever that percentage was and multiplying it by a whole number. You can click on this pool request link and that will give information around this change. With the introduction of percentage to support percentage of billing charges, whenever you want to do something, there are follow-up questions. Adding percentages for billing charges is something we definitely wanted to do to help those developers and cut down internal developments and complexity on figuring out how do we translate that percentage to an actual number.

Scott Haselton: There are other attributes that typically follow for an arrangement. We purposely left those out with the thought or an eye towards the future of continuous development where typically these three attributes that are seen in percentage of billed charges can be added on later. What typically is seen and some of these contractual arrangements, is the relative priority. Those are usually a single number. A multitude of -- multiple services and arrangements. This is usually an outweighed number. It is usually one to 999. Not to exceed for -- not to exceed threshold, this is a singer -- this is a single dollar value that is the maximum paid in percentage. Heard a threshold is a dollar amount and once a percentage of dollar passes that threshold it changes to pay the bill. Usually at a discounted or lower percentage to control costs. These three were not added. We will be looking to add somewhat shortly.

Scott Haselton: Of those scenarios and looking ahead of what possible implementation would be, we have seen this scenario. The numbers have changed but we have pulled this directly from contractual arrangements. This is the current limits is Asian -- this is the current limitation. There is some additional data left out that is very important to this negotiated price. That context is last.

Scott Haselton: Future implementation may look like this where you have a priority, an additional priority, and exceeded limit. That would be optional attributes if the implementation type was. You can see how the negotiated text above. You have your priority or not to exceed. In terms of supporting a deeper level of reporting for these percentages of bill arrangements. We will cover in the next couple of slides, this is not what it looks like for the current implementation, some of the additional attributes for today's schema.

Scott Haselton: The next was Per diem. There is not too much to speak to by adding the per diem. That was for the greater understanding of prices. That was a daily amount. The negotiated rate is a dollar amount. The context rate seen commonly through negotiated arrangements. In a few slides we see how we can represent that. To make sure we are not losing that context. Only the values have been added. That is it for the new negotiated types.

Scott Haselton: We have introduced this concept called custom billing codes. This bore out of contracts that we had seen where 75% of all of inpatient and outpatient services. Rather than having them repeat x-amount of billing rates, we wanted to get this schema closer to what the actual context says. To really help alleviate some of that burden for those that are going to be producing these files. We have introduced two new values. One of them, CSC, means custom mall. The new billing type is custom all. That is meant to represent all custom types under the contractual arrangement. The billing code value or the billing code is CSTM-00. This is intended to reduce file size. Under certain contractual arrangements for producing these files.

Scott Haselton: Here is a situation we ran into, a negotiated arrangement of 35.7% for Bill of patient -- inpatient and outpatient services. You can record these billing codes in single value. All inpatient and outpatient services means we are going to look at all coding types. That is going to record -- represent all coding types. Here we see the percentage is 35.7. One thing to pay attention to, since the billing code type version is the billing -- depending on the version code type, that actually has this contractual arrangement. In that case, if it was produced today, it would be 2022. If you have an arrangement only for a subset of codes, like all HCPCS codes, you don't need to do every billing type. You can do a subset of type. Represented here, you will use the custom billing code as well. You would hit all codes possible and have the negotiated price of percentage and 30's -- 35.7. We have gotten pretty positive feedback on this.

Scott Haselton: We are starting to look where we can -- on further billing codes. One good candidate that has been brought up is how they represented a code that represents all service codes. That is something on the radar that can benefit from some of these past as well. If you have any ideas or you think a certain billing code

Scott Haselton: lastly, the additional information text field. This one is meant. We have this last one please let us know if you want to open up a discussion. We are definitely interested in how this is planned to be used. We want to get a better idea of other common negotiating scenarios would help inform those situations. There could be, when it said there please open up a discussion. There also may be a discussion where a has been discussed and you think there is no implementation possible without using the additional information attribute. It has really helped to help lessen some of that development determined that developers may be going through when building some of these files.

Scott Haselton: Getting back to the previous examples that we saw, this would be a way to use the additional information text field until some of those additional attributes that we talked about previously are added into future iterations of the scheme. The current implementation for this potential scenario. This is pulled directly from an arrangement that we have seen. Members change the spirit of the arrangement holds true.

Scott Haselton: How does one report on the scenario. We have that 35%, but there was a lot of additional context here that would be very important in understanding. This is how one might add that contextual information. You still have that negotiated type that is tied to a percentage, the 35%. The additional information is going to be unstructured data. That is purely for providing context to a negotiated price. Another scenario, the per diem that we have seen that is very common. Maybe you pay one number for a set amount of days. The current implementation would look like this. Not too much additional information there to go off of and have a true understanding of what is applied and one. Leveraging that additional information attribute would really help provide that information on what negotiated rate got supplied one. You can see the additional information is for the first three days . Those are some examples of why we added the additional information, how you can leverage it. Ultimately, what is actually being put into that additional information. We can turn that data unstructured and turn it into structured data. It is going to be easier, not only for the consumer, but also for the producers as well.

Scott Haselton: With that said, those were the changes that we introduced since the last webinar. I would like to run through a tool that was developed to help support the development process and it is this validator tool. Which is pretty exciting. With the version now available and we set it a lot, we plan to continue to develop beyond if the discussions continue. Introducing this transparency coverage, this tool is actually downloadable tool so it is meant to run on local machine. It looks for errors and bugs against the actual transparency. This tool will help produce compliant files by checking for syntax errors as well as for attributes that are required in -- under version 1.0.

Scott Haselton: The validator tool is good to talk about what it does not do. It does not test for the accuracy of the data within the schema. It only test for those required attributes contained within the file. Some of the things we were looking at before embarking on building a tool that could be leveraged by the community, there were two main guiding principles that we wanted to make sure we were heading. One was, since these files are fairly large, having a tool that could validate files quickly was important. We did not want to build a tool where you would start the validation of the file and then go on lunch break and then come back and it did not timeout. We are keeping an eye on that feedback. Feedback was important. With the size of these files, there was the complexity of loading a lot of this information into your local system or your local computers memory. Not everybody has 10 gigs of free RAM available to actually run these files. Making sure that everybody this program can stream inputs that may be sitting on your local file system or internetworks file system -- your network's file system.

Scott Haselton: I am going to go ahead -- and this was just released today. I am going to go ahead and run through how you could use this program, and then the same thing applies if you enjoyed this or you want to put this as part of your CICD pipeline. Free field to do so. -- feel free to do so. I will load this here. Nothing more exciting than a live coding exercise.

Scott Haselton: The first thing you will want to do is clone and that brings the project to your local computer. There is some documentation on prerequisites. There are two core dependencies -- there are two core dependencies on using this. We wanted to write a program about the proliferation of macs and their chip. We wanted to write this once and have it work across multiple systems. It is providing that type of functionality. Lastly, NPM. Those are the only two requirements.

Scott Haselton: The first thing you want to do is make sure docker is running. We will go into the folder that we just downloaded. We will use the docker command to build out the container that is going to be required to fill out these files. Next, just to show, I am copying and pasting to make sure I am not cheating at all. We are going to go ahead and build project. The project is now ready to be used.

Scott Haselton: We have to health command to provide information on what information is available for using the spirit the commands that we have our, I do want to validate file. The first thing to do once the tool has been downloaded is to make sure we have a copy of the schemas local to this directory so that the validator can reference them against your files. The first thing to do their is run this command and that goes out and fetches. Now you are ready to actually validate file. How to do that is, we have included some test files. These are pretty common that we pulled from the other rebuttal. This should not look too strange. These example files are just being pulled from the other one. It follows a basic set of inputs that will be required. We want to validate file. You need to point to the data file. In this case, I am going to point to the test file and the allowed months. -- amounts. This is going to be your locally developed file. You have to indicate what version of the schema you want to validate against. What this does it spends up in the background a docker container to stream this file that is local to your system into the validator. There are a few of them. These are the available schemas that you can validate against.

Scott Haselton: By default, we have the in network rates. What happens when you try to validate against a version that does not exist, what you will get is a list of all versions that the schema currently has. That could help correct or fix your mistake. Lastly, I think it would be good to have an idea on what an error output looks like. And baby some of the did -- and maybe some of the design decisions around that. What we have here, there is an error in this file.

Scott Haselton: I am testing it against version 1.0. We have an error here that the input is invalid. There is an integer when we are expecting a string. We made the design decision on this to fail fast, so when the first error is encountered, fail right away. Some of these files are large and we do not want the developer sitting for dozens of minutes at a time. In this case, we will look at this file. We will see within the lifetime network, it was the first one. The service codes, it will be the fourth one. We expect to see an integer at the fourth value where the schema says it needs to be a st ring. If we go ahead and look at that file, we will see if it holds true. So this is the allowed amounts. We are in the out of network. The service code, we are looking for the fourth. As you can see, this is an integer. That is an error and the validator picked that up. That is the job of the validator , to make sure when you add certain attributes, so if you have a billing class with no service codes provider, this will pick that up. It is to help your development and be injected into your development process.

Scott Haselton: Hopefully there is some sort of continuous development on that set up with the process. And this could easily be plugged into that to provide you with a better sense of what you are producing. With that said, I want to take a step back.

Scott Haselton: We have a few questions for you. When they pop up on the screen, please select the answer that best describes your particular situation. I believe this open for 30 seconds. We have a few questions that we have to get through. Thank you.

Scott Haselton: Next question. It is around the validator tool. While this is going on, it is helpful to know the validator tool performance considerations. Testing on their local machines which are not the most powerful, but is still pretty good. What streaming this data we were able to achieve roughly 25 megs per second for validation. It goes to 100 megs for. That is some basic benchmarks that you can expect from this tool. If you are interested on actual performance benchmarks, there are some links here that have some of those benchmarks.

Scott Haselton: Lastly, question three. More around the strategy of development and what that looks like.

Scott Haselton: I will try to answer as many questions as possible. Yes, they would need to be referenced. The answer is yes, they can be. The separate file needs to be referenced within that main table of contents in network file. But, yes. With Perdiem not available in the past, lots of the rates, and still OK to continue using negotiating type even though it is a Perdiem arrangement? I would say that the only thing that has really changed between those two types is the value negotiated in per diem. I do not want to make any assumptions, but does not seem it would be too heavy to lift to update those types. I would encourage you to look into that. And hopefully you can use the additional information. If you must stick with the negotiated type, I would say leverage that additional information text field to provide that extra context. But I think it will be easier if you just update to using the per diem type.

Scott Haselton: Can you please clarify the use of plan name? Does this represent the health plans actual name? Or is it being used to represent the applicable products at a higher level? We have some information on that . On the price transparency guide, it talks a little bit about that. I think that is what you are asking. Maybe you are asking about the plan name. Right, the plan name. This would be the name of the plan and the plan sponsor. Maybe you could reach out to me so we can get some further information about that.

Scott Haselton: Could percentage be used for the percentage of Medicare billed charges? No, that cannot be used. Percentage used against the percentage of Medicare, you can still get to the dollar rate. It is important to know that in situations you can get to a dollar rate, you must get to a dollar rate. There are situations where you cannot -- could not. It was a retrospective look back, whereas these contractual rights look forward. If you have a percentage that is used for a percentage of Medicare, there is a pretty good example in the roles preamble on how you would be able to get to a dollar amount.

Scott Haselton: On per diem example, how would one manage an arrangement that pays out flat rate and then after X days goat -- goes to a per diem. I think this would be excellent usage of the additional information. You would look at what you could report currently with the permit deal and then you would provide the additional context within that additional information attribute. The percentage of belt charges is meant to support developers that are pulling the information out of contracts. To get as close to the contract as possible in an accurate way. The allowed amounts is that retrospective look back at what actually happened and what was paid. When changes are made, they are methodical and I am trying to think of all parties that are impacted. With the in network rate, it is supporting those that are producing these data elements, these files. Getting that data into a structured way, then those that are going to consume the file can use leverage and build services and products on top.

Scott Haselton: Can the billing code custom all the used to extract the rate for all other items and services, like a catch all? What I would say to that is, open up a discussion and let talk through what an additional billing code can be. I would not say let's try to have one billing code to represent multiple scenarios, because then it gets muddy. That would be my recommendation.

Scott Haselton: Does the additional information text field have [INDISCERNIBLE] . Currently, it does not. You are in charge of how large these files can be. While it is great to provide that additional information, I would be technical -- technical with it. Would we be able to use the validator tool [INDISCERNIBLE] Now come out right now currently the only files supported are json. There is hope to get xml support as well. Additional formats, probably not on the immediate roadmap. A lot of future road mapping would be decided on where we are seeing these files after July 1.

Scott Haselton: Does the validator tool have a file size limit? No, it does not. That was one of the design considerations that was really important. It will only be limited based on what your local system or your network system is hosting. The validator, it is an open stream buffer that reads in as a stream. It does not load the whole file in the memory. No theoretical file size on that. That was really important. If the file fire compliant? That is a tricky one. Right now, no. But you can add one attribute to the file and it can be fire compliant. That could be a future iteration.

Elissa: Dines: I am aware of the time and wanted to –

Scott Haselton: I am sorry.

Elissa Dines: Thank you for answering so many questions. There are number of questions that we will not be able to get to today. As usual, we are taking these back and we will extract the questions, provide answers as technical clarifications on the transparency in coverage website. So you can find us there. We will work as quickly as we can to get answers to questions. We will continue to work through specific cases through github. I think but that come out we can wrap up for today. We really appreciate everybody's feedback. And thank you all and have a pleasant day.