1

00:00:24.840 --> 00:00:33.000

Marvelyn Davis: Good afternoon, ladies and gentlemen, and welcome to the transparency and coverage webinar before we begin, I have a few announcements.

2

00:00:33.450 --> 00:00:44.640

Marvelyn Davis: If you are a member of the press, you may listen in, but please refrain from asking questions during the webinar members of the press can email CMS.

3

00:00:45.180 --> 00:01:01.440

Marvelyn Davis: At press at CMS that hhs.gov for those that need clothes capturing the instructions and the link are located in the chat function in the webinar I will now turn the call over to Elissa Dines.

4

00:01:02.640 --> 00:01:05.880

Marvelyn Davis: In the consumer support us that scenario.

5

00:01:07.380 --> 00:01:07.860

Marvelyn Davis: Elissa.

6

00:01:08.580 --> 00:01:17.460

Elissa Dines: yeah thanks Marv on behalf of society Oh, thank you for attending our third webinar on the transparency and coverage Final Rule machine readable file required.

7

00:01:18.600 --> 00:01:26.160

Elissa Dines: As a reminder, the goal these webinars are to drive the development machine readable schemas to a finalized version 1.0.

8

00:01:26.880 --> 00:01:34.440

Elissa Dines: CMS recognizes that the representation of information required from the transparency and coverage rule and differ from plan to plan.

9

00:01:34.860 --> 00:01:42.690

Elissa Dines: And implementations can lead to various challenges such as file size concerns or how to represent information and unique situations.

10

00:01:43.410 --> 00:01:52.650

Elissa Dines: So, we hope to iterate on the development of information format or scheme of the required data elements to produce a more optimized and consistent file structure.

11

00:01:54.090 --> 00:02:03.480

Elissa Dines: will be answering questions related to today's presentation at the end of the webinar you may ask a question by typing it into the Q amp a function at the bottom of the screen.

12

00:02:03.990 --> 00:02:20.310

Elissa Dines: And we'll do our best to get to as many questions as possible today there are folks that are unable to join the webinar today will be posting the recording of this webinar to our transparency and coverage web page. the web page is located in the chat box.

13

00:02:21.450 --> 00:02:22.320

Coming up shortly.

14

00:02:24.150 --> 00:02:29.820

Elissa Dines: reminder This call is being recorded and we will post the recording on the transparency pepperidge website.

15

00:02:30.660 --> 00:02:36.240

Elissa Dines: Please also remember we're posting a list of Q and a's the transparency pepperidge website.

16

00:02:36.750 --> 00:02:42.030

Elissa Dines: there's been some great discussions on GitHub and questions asked in our previous webinars.

17

00:02:42.360 --> 00:02:59.190

Elissa Dines: And to ensure everyone has access to their responses from CMS we've compiled these technical clarifications and posted them online easy for everyone to find you can find these 10 book clarifications on the resources page of the health plan price transparency rule website.

18

00:03:01.080 --> 00:03:09.450

Elissa Dines: And now I’m going to turn it over to CMS is technical advisor Scott Hamilton to delve into what's been happening on GitHub.

19

00:03:11.970 --> 00:03:12.870

Scott Haselton: Thank you Alyssa.

20

00:03:14.010 --> 00:03:21.360

Scott Haselton: So, for today's agenda it's pretty packed and pretty exciting as well because we're going to be talking about.

21

00:03:22.620 --> 00:03:33.180

Scott Haselton: Some of the larger concerns around the file size and some mitigating efforts that could be taken along with some of the changes that have been made to address that.

22

00:03:33.810 --> 00:03:54.300

Scott Haselton: Directly so really excited to talk about some of the changes and tips and tricks that could be employed to help kind of really push down the size of that file, but today's agenda is a quick recap, of what we covered on the webinar number two the second webinar.

23

00:03:56.220 --> 00:04:14.730

Scott Haselton: What changes have been energy introduced from the last webinar until now and then ultimately going to finish off today's webinar with a discussion on the file size and some implementation implementations that could be taken to really drive down.

24

00:04:16.230 --> 00:04:17.850

Scott Haselton: Some of these larger files.

25

00:04:19.290 --> 00:04:29.340

Scott Haselton: So, with That said, a recap, of what we covered in webinar to I we went through the change process and that covered.

26

00:04:29.970 --> 00:04:45.420

Scott Haselton: What is it look like when smaller and both smaller and larger changes are introduced to get hub, and the process around that and the Community feedback that can go into to some of these changes.

27

00:04:46.890 --> 00:05:05.910

Scott Haselton: We talked about GitHub discussions and get branching incorporating some of these changes, we talked a little bit about pull requests and what those look like and we're going to see a couple examples of that today, with some of the changes that have been introduced from the last webinar.

28

00:05:06.990 --> 00:05:15.090

Scott Haselton: And then we talked about the different types of changes that that could take place, such as adding removing or moving attributes around.

29

00:05:15.780 --> 00:05:25.710

Scott Haselton: And the real idea here and the hope here is that we want to iterate or be iterative and have this process be iterative through these changes, these small changes.

30

00:05:26.850 --> 00:05:35.880

Scott Haselton: And that's why we recommend getting if you want to have the most up to date information really being a participant on GitHub.

31

00:05:36.690 --> 00:05:44.730

Scott Haselton: Where you could see transparently, some of the decisions that are being made and some of the technical changes that are being made.

32

00:05:45.420 --> 00:05:59.880

Scott Haselton: And if you missed any of that we do have recordings of previous webinars here@cms.gov health plan price transparency and you could find that under the resources tab by going to webinars.

33

00:06:01.170 --> 00:06:11.730

Scott Haselton: And you can download the webinar right here and there is a pass code to open it up, or if you'd rather just read, we also have the transcript as well.

34

00:06:13.380 --> 00:06:19.050

Scott Haselton: So, in addition to the change process that we talked about, we also talked about.

35

00:06:20.220 --> 00:06:31.650

Scott Haselton: reporting the NPI or smaller providers that use their social security number as their 10 number and introducing a new format, in which.

36

00:06:32.790 --> 00:06:46.950

Scott Haselton: Those that still do that can report, ultimately, instead of reporting the social security number the provider that still uses their social security number for their tax identify identification number.

37

00:06:48.060 --> 00:07:02.370

Scott Haselton: The providers’ NPI would be used would be used instead and We walked through what some of those changes and examples look like and then at the end we also chatted about.

38

00:07:03.480 --> 00:07:06.900

Scott Haselton: The testing suite that we've introduced.

39

00:07:08.490 --> 00:07:22.620

Scott Haselton: That is hopefully going to provide a little bit of structure when making changes to the code or to the schema or the examples within GitHub also.

40

00:07:23.310 --> 00:07:40.380

Scott Haselton: If your organization wishes to leverage some of the testing suite you can do so as well to get quick feedback on whether some of your sample files that have been generated actually are valid against the schema.

41

00:07:43.200 --> 00:07:52.830

Scott Haselton: So, moving on to what has actually changed within GitHub since the last the last webinar till now.

42

00:07:57.090 --> 00:07:57.690

Scott Haselton: This.

43

00:07:59.760 --> 00:08:04.560

Scott Haselton: We want to cover a couple things we want to cover the naming Convention.

44

00:08:06.510 --> 00:08:08.490

Scott Haselton: And then we want to cover.

45

00:08:09.900 --> 00:08:14.340

Scott Haselton: This this concept of provider references are factored way

46

00:08:15.660 --> 00:08:27.390

Scott Haselton: to represent provider groups that are associated with negotiated rates for items and services, those are the two main things that we want to, we want to cover today.

47

00:08:28.410 --> 00:08:40.080

Scott Haselton: With the naming Convention this actually came about, to provide a little bit of background and I plan to we could go ahead, this might be easier if we just go through this.

48

00:08:40.800 --> 00:08:48.840

Scott Haselton: Originally, when our naming Convention was made, we had the concept of only one plan being reported profile.

49

00:08:49.620 --> 00:09:04.230

Scott Haselton: So the naming was pretty straightforward where you would have a year, or you would have a date the public pairs or issues name the plan name the file type name, so that would either be allowed amounts or in network.

50

00:09:05.310 --> 00:09:15.540

Scott Haselton: And then the actual extension for whatever extension your organization and chooses to produce these files so I’m for CMS for an example of.

51

00:09:16.080 --> 00:09:24.420

Scott Haselton: CMS reporting on Medicare, this is what we would expect the date and the payer and the plan, and then the name of the file.

52

00:09:25.050 --> 00:09:43.080

Scott Haselton: And then, this is essentially the data that would be required at the root node of the file to identify the actual plan so here, you would see the plan name, which is Medicare some of the identifying characteristics of that, and this is all dummy data.

53

00:09:45.060 --> 00:09:51.900

Scott Haselton: There through discussions, it became quite clear that the uniqueness of plans.

54

00:09:53.130 --> 00:10:03.240

Scott Haselton: That there is a lot of overlap and actually there are quite a few plans that are not unique with one another, where they have the same exact.

55

00:10:04.350 --> 00:10:21.030

Scott Haselton: negotiated rates for the same provider groups for the same items and services, and this would lead to having a report those multiple plans with effectively 99% of the same information.

56

00:10:22.110 --> 00:10:29.730

Scott Haselton: And that that is that's quite duplicative and actually not really helpful, so the idea was to take this.

57

00:10:30.690 --> 00:10:36.510

Scott Haselton: The plan name identifiers and then allow for reporting have multiple plans within a single file.

58

00:10:37.320 --> 00:10:44.580

Scott Haselton: So, then you could really cut down on that duplicative reporting or multiple plans reporting the same data.

59

00:10:45.150 --> 00:10:55.050

Scott Haselton: So the idea here was we're going to go ahead and create a new attribute called reporting plans and then just move this this.

60

00:10:55.680 --> 00:11:13.590

Scott Haselton: This plan identifier information into this attribute and here we could have multiple plans we could have as many plans as we want that actually have the exact same rates same providers and same items and services, and this was this was a fantastic opposite optimization.

61

00:11:15.000 --> 00:11:22.050

Scott Haselton: Unfortunately, this led to some file naming issues, we can no longer just I.

62

00:11:23.280 --> 00:11:40.290

Scott Haselton: We can no longer rely on the plan name being unique to be a part of the actual the actual file name so here, this is what we would expect from the first example of Medicare being a part of the file name, but unfortunately it doesn't capture.

63

00:11:42.030 --> 00:11:46.920

Scott Haselton: The Medicaid plan that's also included in this file so.

64

00:11:48.720 --> 00:12:04.830

Scott Haselton: There were a few iterations that were tossed around do we continue to add plans to the actual file name, unfortunately, that doesn't scale and there are limits to how long you can have a filename also it's not very helpful to look at.

65

00:12:06.090 --> 00:12:08.250

Scott Haselton: Initially, from a user's point of view.

66

00:12:09.330 --> 00:12:23.220

Scott Haselton: So, moving towards are moving down that route didn't was not going to be feasible, so what this is really the crux of the file name issues that we're we were encountering.

67

00:12:26.100 --> 00:12:37.860

Scott Haselton: So the idea here is if you choose the option to include multiple plans within the single file, which is optional.

68

00:12:38.430 --> 00:12:47.850

Scott Haselton: it's a great optimization we would certainly recommend it, but if you choose to report multiple plans per file there is this concept.

69

00:12:48.390 --> 00:13:00.690

Scott Haselton: of a new table of contents file that will effectively capture all of the same information that was in the single in network or a lot amounts file and then.

70

00:13:02.220 --> 00:13:04.170

Scott Haselton: make a reference to.

71

00:13:07.080 --> 00:13:12.570

Scott Haselton: And then make a reference to where that file can be downloaded for the.

72

00:13:15.120 --> 00:13:19.530

Scott Haselton: For both for the multiple plans that that are identical.

73

00:13:21.510 --> 00:13:22.080

Scott Haselton: and

74

00:13:23.550 --> 00:13:24.450

I’m sorry.

75

00:13:42.750 --> 00:13:43.710

sorry about that glitch.

76

00:13:46.830 --> 00:13:50.460

Scott Haselton: So we have we have this new concept of a table of contents file.

77

00:13:51.750 --> 00:14:04.020

Scott Haselton: And it really contains much of the same information that we found in our original file so there's not there's not really any new information that's going to be required outside of.

78

00:14:05.310 --> 00:14:16.680

Scott Haselton: Where to find the file, so we have these reporting plans and reporting entities, these are all the same we're just moving this data into a new table of contents file.

79

00:14:17.790 --> 00:14:25.020

Scott Haselton: What is new is this new attribute called the end network file and the idea here is that there is a description.

80

00:14:26.010 --> 00:14:41.820

Scott Haselton: Of that file and then the file location so for these reporting plans this in network file attribute is telling the user or the computer where to actually go and download the file for these plans.

81

00:14:42.840 --> 00:14:53.730

Scott Haselton: So this information, right here would actually be pointing back to the network negotiated information for the plans that it represents.

82

00:14:56.910 --> 00:15:18.210

Scott Haselton: So we could see that this file location would be, we would be wherever you would want to define it and it would contain the information not the plan information anymore, because that would be within the table of contents, it would just contain the in network negotiation information.

83

00:15:20.040 --> 00:15:32.040

Scott Haselton: So this is the network file and this, this means that the in network file itself Medicare and network file, you can name this whatever you would like, because it is actually being defined within.

84

00:15:32.520 --> 00:15:41.730

Scott Haselton: Within this object, and there are there doesn't need to be a standard on a naming the specific in network or a lot about files.

85

00:15:42.450 --> 00:15:53.970

Scott Haselton: Now this shifts the burden of naming over to the table of contents file, so that is what we need to come up with for standard and not the actual individual files themselves.

86

00:15:59.940 --> 00:16:11.580

Scott Haselton: So the new naming standard if you choose to report multiple plans in a single file and the creation of a table of contents file will be required.

87

00:16:12.030 --> 00:16:28.830

Scott Haselton: by the new naming standard for the table of contents file will be the date, much like we saw with the individual files, the payers name and underscore index, and then the file extension, so in this case.

88

00:16:30.030 --> 00:16:37.380

Scott Haselton: we're going to see this being the date, this is going to be CMS and it will be underscore index.

89

00:16:41.040 --> 00:16:46.470

Scott Haselton: Now the naming if you choose not to report multiple.

90

00:16:47.850 --> 00:16:55.590

Scott Haselton: If you choose not to report multiple plans per file and that's fine you could still report, a single plan profile.

91

00:16:56.940 --> 00:17:07.290

Scott Haselton: The name is going to be the same as it was if you choose to if he if you don't choose to leverage the multiple plans profile so nothing will actually be changing there.

92

00:17:08.250 --> 00:17:24.450

Scott Haselton: And the addition of the table of contents file itself that is the naming that will be different and you'll see the updated documentation actually at the at the root of the GitHub page, so we have examples of.

93

00:17:25.590 --> 00:17:34.740

Scott Haselton: Both the naming right here, and then we also have examples if you're curious on what some of those table of contents files may look like.

94

00:17:36.270 --> 00:17:37.620

Scott Haselton: To help provide am.

95

00:17:39.480 --> 00:17:40.770

Scott Haselton: Just provide an example.

96

00:17:45.660 --> 00:17:50.370

Scott Haselton: So here we have this is would be the table of contents sample.

97

00:17:52.020 --> 00:17:58.500

Scott Haselton: We have a reporting entity, we have a reporting structure that contains multiple plans.

98

00:17:59.760 --> 00:18:10.470

Scott Haselton: That we have the in-network file for that specific for these groups of plans, we have the allowed amount file for those groups of plans and then we have another object here let's say that.

99

00:18:12.630 --> 00:18:30.240

Scott Haselton: Both Medicaid and Medicare had the same negotiate negotiated items and services, but ship doesn't it has let's say different provider groups or different items and services we go ahead and include that single plan as part of the array.

100

00:18:31.620 --> 00:18:35.730

Scott Haselton: And then we just point to the file on where that would be located.

101

00:18:45.660 --> 00:18:57.330

Scott Haselton: So that was that was file naming and you could right now we have a pull request open that you can make if you could provide feedback on.

102

00:18:58.590 --> 00:19:11.160

Scott Haselton: And there was this was this this proposal was it was by and large proposed from the Community, and it went through some iterations of this is a fantastic suggestion.

103

00:19:11.580 --> 00:19:19.770

Scott Haselton: And it went through some iterations and the biggest change from what was proposed to the Community to what is actually being incorporated.

104

00:19:20.550 --> 00:19:31.080

Scott Haselton: Right now, with this specific pull request is taking a smaller iterative step versus the initial proposal that can be found in in the discussions.

105

00:19:31.770 --> 00:19:53.310

Scott Haselton: The initial proposal actually had the idea of taking multiple in network files that could then be combined together from remote sources and we just wanted to take the first initial or first initial step iterative step to solve the file naming issue.

106

00:19:54.840 --> 00:19:59.010

Scott Haselton: While continuing to look at potential optimizations around.

107

00:20:00.270 --> 00:20:17.190

Scott Haselton: referencing external files that could then be combined, so that is, that is something that CMS is still looking at and very much interested in, but the idea here really was to figure out or drive a solution towards.

108

00:20:18.240 --> 00:20:24.780

Scott Haselton: file naming for those that wish to take advantage of reporting multiple plans in a single file.

109

00:20:25.830 --> 00:20:41.190

Scott Haselton: So continue to observe or keep an eye on potential changes coming down the pipe for the table of contents that may or may not allow for the referencing of external files.

110

00:20:43.620 --> 00:20:49.560

Scott Haselton: Next, I would like to move on to the provider reference provider referencing.

111

00:20:51.240 --> 00:21:06.720

Scott Haselton: And this change, so the table of contents that allows for multiple plans to be reported in a single file if that changes that cuts down on the actual storage concerns and bandwidth concerns is fairly massive.

112

00:21:08.100 --> 00:21:10.440

Scott Haselton: I would go ahead and lump.

113

00:21:11.490 --> 00:21:15.660

Scott Haselton: The provider reference change or proposal to be.

114

00:21:17.190 --> 00:21:22.560

Scott Haselton: Almost as large as including multiple plans profile.

115

00:21:24.210 --> 00:21:31.020

Scott Haselton: So, right now, currently for all items and services, we have the concept of.

116

00:21:32.910 --> 00:21:42.300

Scott Haselton: here's an item for service, and we have the negotiated rates and then we have this concept of provider groups which basically.

117

00:21:43.980 --> 00:22:08.310

Scott Haselton: pairs up all the providers with their place of business or where they actually provide the service and then the negotiated rate for that actual service, and this, this was this initial step was fine in terms of making sure that all the required information was available and formatted.

118

00:22:09.630 --> 00:22:13.200

Scott Haselton: In a way, that could be used as a composite key but.

119

00:22:15.480 --> 00:22:22.770

Scott Haselton: As one would expect some of these provider groups can get significantly large extremely large.

120

00:22:24.270 --> 00:22:33.450

Scott Haselton: And I created a little example here, and what I mean by that is there are situations where the amount of providers that are associated with the single on.

121

00:22:33.810 --> 00:22:49.710

Scott Haselton: A single place of business can be massive so what one would expect would be this NPI array to be quite large and it just goes on and goes on and goes on, and there are we've talked to folks that have some of these provider networks.

122

00:22:51.030 --> 00:22:59.580

Scott Haselton: that are in the thousands so it's not insignificant and the idea here is, we are, we are.

123

00:23:01.290 --> 00:23:20.790

Scott Haselton: requiring the reporting of some of these larger networks for every single item in service for each negotiated price so there's the potential of quite a bit of duplication that could happen there without actually providing that much additional value.

124

00:23:22.560 --> 00:23:38.040

Scott Haselton: So in providing in looking into provider references the idea here is to allow for a reference to that provider group on the actual root node so.

125

00:23:38.700 --> 00:23:49.860

Scott Haselton: We have this provider reference object, right here in the documentation which takes the provider group the provider group that you just saw me outlining or highlighting.

126

00:23:51.330 --> 00:24:10.080

Scott Haselton: And then, it adds this concept of a provider group ID which is pretty much a primary key so it's interesting here because flat files typically our data dumps and they're typically the normalized or something called D normalize.

127

00:24:11.760 --> 00:24:14.940

Scott Haselton: because all the data needs to live in a single in a single file.

128

00:24:16.350 --> 00:24:22.230

Scott Haselton: By introducing this concept we provide a little bit of normalization, which is what you would typically find.

129

00:24:23.550 --> 00:24:33.840

Scott Haselton: In like a normal database development database schema or even in rest API services, so we put this at the actual root node.

130

00:24:34.680 --> 00:24:48.270

Scott Haselton: And then, this allows for the defining of those provider groups once and only once, and then referencing those group ids within the negotiated item or service, so an example of what that might look like.

131

00:24:51.780 --> 00:25:02.070

Scott Haselton: Is here we have a single plan that's being reported, because we have all the plan data here, and then we have the provider reference that is at the root.

132

00:25:02.610 --> 00:25:18.630

Scott Haselton: Of all of this plan data and this provider references an array of objects of those provider groups like the one I just highlighted previously and then a provider group ID which this is going to be a numeric value that must be defined.

133

00:25:19.890 --> 00:25:29.490

Scott Haselton: by those that are wishing to employ this option, and this is optional, by the way, this is not a requirement using provider references.

134

00:25:30.510 --> 00:25:34.230

Scott Haselton: is more of a suggestion, if your organization.

135

00:25:35.310 --> 00:25:44.760

Scott Haselton: finds that using continuing the use provider groups within those iterations are those items in services loop.

136

00:25:46.260 --> 00:25:47.790

Scott Haselton: You can continue to do so.

137

00:25:49.590 --> 00:25:53.730

Scott Haselton: But here we have provider references at the root node.

138

00:25:55.320 --> 00:26:06.120

Scott Haselton: And there is there is one group and here's a whole other group let's say it's the same group of providers it's just at a different level, so we give her different ids.

139

00:26:07.620 --> 00:26:10.380

Scott Haselton: And then within the actual in network object itself.

140

00:26:11.430 --> 00:26:20.520

Scott Haselton: Instead of having provider groups, we have provider references, and this is going to be an array of ids that have been defined up here.

141

00:26:21.420 --> 00:26:33.360

Scott Haselton: So we have one, so this suggests or this points back to this I’ve been provider group within ID one is a part of this negotiated rate for this item or service also.

142

00:26:35.160 --> 00:26:36.660

Scott Haselton: also provide a group to.

143

00:26:38.430 --> 00:26:40.440

Scott Haselton: And to give you an idea on.

144

00:26:41.640 --> 00:26:47.730

Scott Haselton: If you implement this some of the benefits of file size reduction.

145

00:26:50.190 --> 00:26:52.980

Scott Haselton: I have this example where.

146

00:26:54.300 --> 00:26:56.670

Scott Haselton: If you were planning to use provider groups.

147

00:26:58.710 --> 00:27:04.650

Scott Haselton: I repeated the same item 10 times with the same groups.

148

00:27:06.510 --> 00:27:09.570

Scott Haselton: Just to get an idea on how large of that file is.

149

00:27:11.160 --> 00:27:17.520

Scott Haselton: doesn't seem significant, but it, it has a possibility to become extremely large.

150

00:27:18.300 --> 00:27:36.810

Scott Haselton: And I did the same thing with the provider reference so those provider groups and that previous file there this in groups here one group to group, and then I repeated that process for the 10 the same 10 items and services so here we're just using the reference.

151

00:27:40.590 --> 00:27:52.740

Scott Haselton: And the given idea of these files both of these files are logically equivalent, but the file size between them are significantly different.

152

00:27:54.690 --> 00:27:56.460

Scott Haselton: And how different that is.

153

00:27:58.800 --> 00:28:05.610

Scott Haselton: I met her I easy my command line I’m we're going to go ahead and look at some of the sizes, the sizes comparative those two files.

154

00:28:11.640 --> 00:28:20.250

Scott Haselton: So we have the provider group, which is the larger if you choose to continue using provider groups, and then we have the optional provider reference.

155

00:28:21.300 --> 00:28:28.890

Scott Haselton: For those 10 the same 10 items and services with those provider groups 8585 K.

156

00:28:30.600 --> 00:28:33.570

Scott Haselton: For that same logical provide a reference.

157

00:28:34.710 --> 00:28:44.010

Scott Haselton: it's actually 19 K so just with 10 items and services if you if you're reporting more than 10 items and services.

158

00:28:44.550 --> 00:29:00.090

Scott Haselton: With provider groups of comparable size that you saw just between these two that's already a seven that's roughly like a 70% decrease in file size and the file size itself is only going to get smaller and smaller.

159

00:29:01.950 --> 00:29:19.950

Scott Haselton: Based on the amount of provider groups and the amount of items and services that are being reported it wouldn't be unusual to see file sizes decrease up to 90% are some larger plans even decreasing like 9595 to 98%.

160

00:29:21.600 --> 00:29:23.400

Scott Haselton: So very, very significant.

161

00:29:25.350 --> 00:29:36.060

Scott Haselton: optimization and we recommend that both a you take a look at some of the documentation, this also has a pull request.

162

00:29:38.280 --> 00:29:42.330

Scott Haselton: That you could look over and see what files have changed.

163

00:29:43.650 --> 00:29:51.120

Scott Haselton: Calm comment or provide an iterative update on what has been proposed here.

164

00:29:52.380 --> 00:29:53.610

Scott Haselton: And from there.

165

00:29:55.680 --> 00:30:02.340

Scott Haselton: We can expect this change to be integrated in sometime within the next week and.

166

00:30:03.630 --> 00:30:16.020

Scott Haselton: When I say optional the provider references themselves you don't need to do that, you could still use provider groups, but the idea here is that you have to choose either a provider reference or a provider group.

167

00:30:16.590 --> 00:30:26.820

Scott Haselton: Because we still need to have that ability to point both the providers and their associated tend to the item in service.

168

00:30:27.930 --> 00:30:30.540

Scott Haselton: To the negotiated item or service so.

169

00:30:31.980 --> 00:30:43.560

Scott Haselton: We recognize that organizations have had been in development for a while, so that's why we, we continue to make some of these optimizations optional I’m, just in case.

170

00:30:45.780 --> 00:30:51.450

Scott Haselton: You like how things are currently already set up and we wouldn't want to change that or.

171

00:30:52.950 --> 00:30:57.450

Scott Haselton: or some of these optimizations just are irrelevant to you.

172

00:30:58.500 --> 00:31:00.450

Scott Haselton: So really excited to get this in.

173

00:31:01.710 --> 00:31:05.580

Scott Haselton: This also has the possibility, just like.

174

00:31:06.870 --> 00:31:16.980

Scott Haselton: Just like the table of contents, with the possibility of referencing external files, this also has that possibility of referencing external provider files.

175

00:31:17.520 --> 00:31:30.030

Scott Haselton: I wouldn't actually cut down on the total size of the files because you're one would be splitting the provider file into a separate file.

176

00:31:31.260 --> 00:31:41.760

Scott Haselton: You would still have to download that along with your in network for allowed amounts file, so it would still be that the size would still be the same.

177

00:31:43.110 --> 00:31:59.100

Scott Haselton: split up over multiple files, it would just be a cleaner way to organize it and we we're still we're looking at that possibility as well, but what you can expect is that the schema for the provider reference.

178

00:32:00.330 --> 00:32:11.280

Scott Haselton: Here the schema for the provider reference this probably won't change if this were to be an external file, it would still be the same, it would just be put into a different file.

179

00:32:11.760 --> 00:32:18.720

Scott Haselton: So you could rest assured that no large changes would be made if we do allow for.

180

00:32:19.650 --> 00:32:35.850

Scott Haselton: External references to provider files, but in the spirit again in the spirit of small iterative steps we wanted to first refactor how providers are being represented within the same file itself before taking that additional step.

181

00:32:37.050 --> 00:32:40.530

Scott Haselton: In in allowing for referenced files.

182

00:32:42.630 --> 00:32:48.330

Scott Haselton: So those are some of the changes that we have we've implemented both of them really have to do.

183

00:32:49.140 --> 00:32:57.510

Scott Haselton: With file size, with one of them actually trying to button up some of the naming Convention questions that that have been swirling around.

184

00:32:58.500 --> 00:33:15.390

Scott Haselton: Again, if you see a problem or have a suggestion or just a general question, please go ahead and go into the poll request and make your voice heard there and we will try to respond to all relevant.

185

00:33:16.500 --> 00:33:17.910

Scott Haselton: Discussions or questions.

186

00:33:20.010 --> 00:33:25.680

Scott Haselton: So with that said and moving on to some file size.

187

00:33:27.420 --> 00:33:29.280

Scott Haselton: optimization recommendations.

188

00:33:31.440 --> 00:33:32.490

Scott Haselton: This is really.

189

00:33:33.990 --> 00:33:40.380

Scott Haselton: This is really some things that you could do, if you have not employed them yet to really address.

190

00:33:41.940 --> 00:33:43.380

Scott Haselton: ballooning file sizes.

191

00:33:45.120 --> 00:33:50.220

Scott Haselton: And so, there are various opportunities there's we're going to be covering.

192

00:33:50.940 --> 00:34:03.840

Scott Haselton: Examples of provider grouping service codes multiple plans per file the reference provider group some of this stuff we've already we've already touched on, because the changes were just introduced.

193

00:34:04.470 --> 00:34:11.880

Scott Haselton: And then file compression and obviously depending every organization is unique so depending on your situation.

194

00:34:12.450 --> 00:34:30.270

Scott Haselton: Your mileage may vary on all of these are one of these or any type of combination of these so with provider groups on the use case here, is it providers that have negotiated rates for items and services under a single 10, and this is what it used to look like.

195

00:34:31.980 --> 00:34:41.250

Scott Haselton: providers and 10s weren't actually grouped up in an individual object I so that actually allowed that caused a little bit of complication.

196

00:34:41.910 --> 00:34:55.320

Scott Haselton: Around having to display multiple negotiated rates, even though the negotiated price may be the same for different groups of providers and in the think about it conceptually.

197

00:34:56.490 --> 00:35:02.790

Scott Haselton: Some pseudo code is what you want to be doing is grabbing all the providers and grouping them by the tax ID.

198

00:35:04.560 --> 00:35:06.720

Scott Haselton: And this is what the previous implementation looks like.

199

00:35:08.520 --> 00:35:24.000

Scott Haselton: And with the updated implementation, we could have multiple groups that can be a city associated with a single to a single negotiated rate so that's great, but still keeping in mind that you want to be able to group all those providers.

200

00:35:25.170 --> 00:35:28.050

Scott Haselton: group I’m by an actual tax ID.

201

00:35:29.160 --> 00:35:36.900

Scott Haselton: So we have noticed this scenario, this particular scenario, with a few people that we've talked with, as well as actually this came up and get hub to.

202

00:35:38.460 --> 00:35:55.710

Scott Haselton: Where provider groups are being split across the actual tax ID but the tax ID might be the same, and so again thinking back to that grouping by tax ID we really only want to have one representation of this tax ID and then put.

203

00:35:56.790 --> 00:36:16.260

Scott Haselton: All the relevant MP eyes into a single array so this means taking these two these two APIs or lists of NPIs and putting them into a single list and then collapsing, a single tax ID into this value, right here, or this representation.

204

00:36:17.490 --> 00:36:26.760

Scott Haselton: So if you find yourself not grouping by tax ID there this, this is a great way to start to cut down on some of those provider group objects.

205

00:36:27.930 --> 00:36:30.210

Scott Haselton: and potentially saving quite a bit of space.

206

00:36:33.900 --> 00:36:38.670

Scott Haselton: So the takeaway a group all your providers associated with a single tenant to single provider group option.

207

00:36:40.320 --> 00:36:41.580

Scott Haselton: So, with the service code.

208

00:36:42.630 --> 00:36:53.250

Scott Haselton: Initially, the service code was represented as a string and it was pretty apparent early on that negotiated rates are negotiated prices.

209

00:36:55.290 --> 00:37:00.960

Scott Haselton: Many of the service codes themselves have a single negotiated rate or single negotiated price.

210

00:37:01.470 --> 00:37:10.290

Scott Haselton: And, and if that were the case, you would run into a situation like this, where the only thing that was really different between this object.

211

00:37:10.680 --> 00:37:25.290

Scott Haselton: And this object it's all the same that has all the same providers the same 10 being negotiated prices exactly the same, the only difference is the service code itself, and this leads to I mean roughly what 90% of that's duplicative.

212

00:37:27.420 --> 00:37:43.080

Scott Haselton: So that that that led to quite a bit of waste, so the previous implementation here was again that that service code is an individual string and we have now, the current implementation allowing, not only did we move the service code.

213

00:37:44.730 --> 00:37:54.000

Scott Haselton: From the negotiated rates object into the negotiated price object, but now we allow for an array So the idea here is don't break up service codes.

214

00:37:54.840 --> 00:38:05.310

Scott Haselton: That that are part or that have the same negotiated rate put all of them into the array so you don't have to repeat the negotiating prices object over and over.

215

00:38:07.290 --> 00:38:13.290

Scott Haselton: So yeah the takeaway here is making sure that you group all the relevant service scripts together for this specific negotiated price.

216

00:38:16.830 --> 00:38:23.730

Scott Haselton: I we've covered this previously, not only in the last webinar but earlier today.

217

00:38:25.020 --> 00:38:41.970

Scott Haselton: so quickly just moving through this one, this is an optional we highly recommend it if you find your use case being that plans have the same exact multiple plans that have the same exact negotiated rates for all items and services for the same provider networks.

218

00:38:43.200 --> 00:38:55.770

Scott Haselton: If you fit that use case this option is, I think a fantastic way to cut down on some of that that storage footprint and then ultimately the bandwidth that's going to be needed to serve these files so.

219

00:38:56.970 --> 00:39:01.860

Scott Haselton: yeah the impact is the to avoid the possibility of producing hundreds of large duplicate files.

220

00:39:03.840 --> 00:39:08.670

Scott Haselton: And with that we had we talked about our table of contents.

221

00:39:09.990 --> 00:39:20.280

Scott Haselton: That would basically capture all of the reported plans that are the same and then have this concept of where to find that actual file itself.

222

00:39:22.170 --> 00:39:25.470

Scott Haselton: And yes, this was just like earlier in the webinar.

223

00:39:28.410 --> 00:39:46.620

Scott Haselton: So really taking a good look at the plans that are being offered and how identical those plans are in relation to each other and then really trying to push those plans into a single file and leverage that table of contents to point to what that where that file lives.

224

00:39:48.000 --> 00:39:53.190

Scott Haselton: will be huge we will absolutely decrease the total footprint size.

225

00:39:54.570 --> 00:39:56.430

Scott Haselton: Of the amount of files that are going to be needed.

226

00:39:59.910 --> 00:40:02.730

Scott Haselton: A reference provider group so we talked about this.

227

00:40:04.350 --> 00:40:15.240

Scott Haselton: just a moment ago, but the idea on really trying to use these or reference them is, if you have large provider groups per 10.

228

00:40:15.900 --> 00:40:34.890

Scott Haselton: Or, if you have lots of different provider groups or if you just have small amount of provider groups, this really is a fantastic way to cut down on the actual size like we mentioned, even with 10 items and services with a relatively small provider group.

229

00:40:35.940 --> 00:40:42.060

Scott Haselton: We saw that 70% decrease in size and it's only going to grow larger.

230

00:40:43.170 --> 00:40:46.890

Scott Haselton: Depending on how large that network and how many items and services that you offer.

231

00:40:48.150 --> 00:40:59.100

Scott Haselton: And it looks a little strange and that's because the flat files themselves are typically D normalized views of data, and what that means is just.

232

00:40:59.640 --> 00:41:11.640

Scott Haselton: Everything is put into a single file and there's going to be a lot of redundancy, a lot of duplicate of data, but the idea is that you don't have to go anywhere else everything's just there for you.

233

00:41:13.110 --> 00:41:19.890

Scott Haselton: And this takes it almost a normalized this reference provider group homes takes a normalized representation.

234

00:41:21.330 --> 00:41:25.980

Scott Haselton: That you would typically find in a database or an API call.

235

00:41:29.460 --> 00:41:30.420

Scott Haselton: As.

236

00:41:31.860 --> 00:41:33.150

Scott Haselton: As we looked at earlier.

237

00:41:34.530 --> 00:41:41.760

Scott Haselton: you're going to take converting your provider groups let's say that you want to go through this process of using provider references.

238

00:41:42.270 --> 00:41:47.730

Scott Haselton: And that conversion is you know look roughly like this process right here you're going to take your provider groups themselves.

239

00:41:48.300 --> 00:41:57.090

Scott Haselton: And you're going to put them in the provider group object which is going to be the same exact thing and so these two different groups get put into these two different objects.

240

00:41:58.140 --> 00:42:01.740

Scott Haselton: Under provider references the provider references array.

241

00:42:02.850 --> 00:42:11.130

Scott Haselton: The only real difference here is that now your organization will need to provide whatever this numeric value this this.

242

00:42:12.270 --> 00:42:17.310

Scott Haselton: it's called a primary key but whatever this value is your organization will need to provide that.

243

00:42:18.600 --> 00:42:21.180

Scott Haselton: In addition to the actual provider group itself.

244

00:42:24.600 --> 00:42:30.900

Scott Haselton: And then, this this is typically what you'll This is where you'll see the file size reduction here.

245

00:42:31.860 --> 00:42:42.600

Scott Haselton: So in this old implementation or this I shouldn't say old, but this implementation that's using provider groups, if we if we want to go ahead and start using provider references.

246

00:42:43.590 --> 00:42:52.320

Scott Haselton: We could see that this particular provider group is represented over here same with this provider group over here, but this.

247

00:42:52.950 --> 00:42:57.480

Scott Haselton: This actual object, right here is the provider reference objects, so we do have some ideas.

248

00:42:58.410 --> 00:43:17.790

Scott Haselton: So instead of representing these provider group objects within the negotiated rates object, we go ahead and update that to say provider references and then we use the ids that are associated with those references, so you could see just within.

249

00:43:18.960 --> 00:43:32.160

Scott Haselton: The example that had to provider groups itself it's I mean it's significantly smaller and it will this particular change will scale and quite dramatically, the larger the file is.

250

00:43:36.030 --> 00:43:37.710

Scott Haselton: And these are where those groups are.

251

00:43:38.730 --> 00:43:51.090

Scott Haselton: So the idea here is we're going to normalize the provider groups and put them on that route object and then ultimately we're going to reference them within the network on the network object itself.

252

00:43:53.490 --> 00:44:05.730

Scott Haselton: So I again I alluded to this a little bit earlier, but something to think about is those provider reference objects those probably will not change.

253

00:44:06.270 --> 00:44:19.830

Scott Haselton: If external files are implemented, so you could feel pretty secure, knowing that any additional logic that you might need to include in this provider reference objects, you probably want me to include anything.

254

00:44:21.480 --> 00:44:22.350

Scott Haselton: And then lastly.

255

00:44:23.370 --> 00:44:32.490

Scott Haselton: file compression there are some questions around Am I allowed to compress the file after they've been produced, and the answer is yes, you can and recommend it's recommended.

256

00:44:34.650 --> 00:44:53.040

Scott Haselton: Sometimes compression doesn't get you very much, depending on the type of file that you're compressing So if you were to compress like an MP3 or a video file you're not going to see a lot of significant gains, but you will see quite significant gains for it for text files.

257

00:44:54.840 --> 00:44:57.240

Scott Haselton: And, here are some common compression formats.

258

00:44:58.530 --> 00:45:05.160

Scott Haselton: I would just recommend don't use anything super exotic that is that.

259

00:45:06.330 --> 00:45:15.210

Scott Haselton: Consumers have the file will actually have to pay for to uncompressed and there are so many options or free compression options out there that are already loaded on.

260

00:45:16.230 --> 00:45:22.740

Scott Haselton: Most operating systems that I choosing any one of these will get you quite far.

261

00:45:24.450 --> 00:45:37.410

Scott Haselton: We CMS doesn't have any official recommendation on what compression algorithm should be used, but you are free to go ahead and compress and we've received feedback that.

262

00:45:39.180 --> 00:45:50.310

Scott Haselton: From various organizations that just the compression step alone when you take away all the other optimizations just the compression step alone has reduced file size by.

263

00:45:51.060 --> 00:46:00.360

Scott Haselton: Potentially up to 60% so that is pretty significant and it's one of those good last steps that that you, you couldn't attend should take.

264

00:46:04.410 --> 00:46:06.090

Scott Haselton: So with that said.

265

00:46:08.910 --> 00:46:12.000

Scott Haselton: I am going to put up poor.

266

00:46:14.010 --> 00:46:22.710

Scott Haselton: we're going to be asking a couple questions here and they dovetail into some of the discussions that we've been having around file size and some.

267

00:46:23.880 --> 00:46:36.660

Scott Haselton: Some opportunities that could be taken based on your actual individual organizations DNA, if you will, so the first poll, that we want to ask is or get a sense of is.

268

00:46:38.190 --> 00:46:46.740

Scott Haselton: how large the actual health plans that you are working with how many Members, those are covering to give a rough.

269

00:46:47.760 --> 00:46:54.780

Scott Haselton: gut check on what some of these provider groups could look like in size so i'm going to go ahead and watch this right now.

270

00:47:28.380 --> 00:47:30.810

Scott Haselton: i'm going to leave this on for another 30 seconds.

271

00:48:00.840 --> 00:48:02.250

Scott Haselton: Great Thank you um.

272

00:48:03.330 --> 00:48:10.830

Scott Haselton: Interestingly, it's pretty uniformly distributed across those as for which is, which is fascinating because.

273

00:48:11.910 --> 00:48:14.250

Scott Haselton: Each one of those options are going to have their unique.

274

00:48:15.720 --> 00:48:20.490

Scott Haselton: Their unique constraints and challenges when building up the file.

275

00:48:21.810 --> 00:48:23.940

Scott Haselton: And here's a second polling question.

276

00:48:25.890 --> 00:48:37.710

Scott Haselton: That would provide some valuable insight if you started developing which I would like to take this opportunity again to really recommend start your development now or yesterday.

277

00:48:38.550 --> 00:48:55.290

Scott Haselton: A lot of the information that's within the Final Rule that information hasn't changed identifying all of your items and services, identifying your negotiated rates identifying your provider groups, all of this could be done right now, without having to wait.

278

00:48:56.640 --> 00:49:13.140

Scott Haselton: With without having to wait for these iterative changes to be to be implemented and much of these changes that we've been seeing they're out there optional as well, so there's really there really is no reason not to get started, but for those that have started.

279

00:49:15.480 --> 00:49:19.320

Scott Haselton: Here is a question around your estimated size for the network file.

280

00:49:20.370 --> 00:49:22.560

Scott Haselton: we'd really love to get some thoughts on that.

281

00:50:36.570 --> 00:50:50.040

Scott Haselton: Okay, thank you, thank you for responding to that this is, this is all over the board as well with a lot of unknown at the moment to certainly some larger representation of the 50 gigabit as well.

282

00:50:51.750 --> 00:51:02.850

Scott Haselton: And, and that that's large and we definitely want to, we want to solve that so thank you for the feedback and with that said i'm going to hand it on over to Alyssa.

283

00:51:05.730 --> 00:51:20.730

Elissa Dines: Thank Scott and yeah before moving on, I want to just remind folks of the departments issue guidance on August 20 and different enforcement of their requirement to publish machine readable files related to prescription drug prices.

284

00:51:21.510 --> 00:51:30.090

Elissa Dines: Ending further rulemaking a link to the guidance is available on GitHub side as well as the transparency and coverage website, and this is a website.

285

00:51:30.690 --> 00:51:50.160

Elissa Dines: And I just want to note that CMS will not be promulgating any form and manner guidance related to the rx file at this time, so any material currently or previously published on the rx file repository page should be considered null and void and not guidance of the department.

286

00:51:52.200 --> 00:52:03.480

Elissa Dines: Okay, before we take a quick look at questions we're hoping to get your feedback on what topics, you would find most useful for future webinars.

287

00:52:03.960 --> 00:52:20.880

Elissa Dines: we've covered some of the big ones at this point so are there other topics we haven't discussed that you would like us to if you could submit your answers in the Q amp a box, we can take them back for consideration so I’ll just.

288

00:52:21.930 --> 00:52:24.660

Elissa Dines: pause for a few moments, so that folks can.

289

00:52:25.680 --> 00:52:27.930

Elissa Dines: Let us know what they want to hear from us about.

290

00:52:45.060 --> 00:52:47.820

Elissa Dines: Okay, seeing some good ideas here.

291

00:52:54.750 --> 00:52:57.720

Elissa Dines: Thank you, more moments, because these are.

292

00:52:58.830 --> 00:53:00.150

Elissa Dines: Great suggestions.

293

00:53:05.640 --> 00:53:05.970

yeah.

294

00:53:07.080 --> 00:53:08.610

Elissa Dines: percentage charge.

295

00:53:10.290 --> 00:53:11.730

Alternative payment.

296

00:53:16.890 --> 00:53:19.470

Elissa Dines: Security issues.

297

00:53:24.270 --> 00:53:24.690

Okay.

298

00:53:31.710 --> 00:53:32.430

Okay.

299

00:53:38.970 --> 00:53:41.160

Scott Haselton: Unless I can jump in and start answering some of these.

300

00:53:41.670 --> 00:53:51.540

Elissa Dines: Okay, that sounds good um thanks everyone for your feedback feel free to keep submitting suggestions and we will take them back.

301

00:53:53.490 --> 00:53:55.380

Scott Haselton: And yes, so.

302

00:53:56.640 --> 00:54:10.710

Scott Haselton: This one actually came through the GitHub discussion, and I saw it here as well, it's around the billing code type version, what if there is no specific version like icd has versions right they have nine and 10.

303

00:54:12.030 --> 00:54:23.430

Scott Haselton: What if you use something that doesn't have a version, could the year be used, and the answer is yes it's meant to be the versions meant to be a year if I if there is no specific version.

304

00:54:25.530 --> 00:54:28.440

Scott Haselton: associated with whatever the code that is being used.

305

00:54:32.100 --> 00:54:39.780

Scott Haselton: is using the provider reference objects required, no, no it's optional most of this stuff is going to be optional.

306

00:54:41.130 --> 00:54:43.800

Scott Haselton: Because we don't want to be too disruptive with.

307

00:54:45.060 --> 00:54:49.140

Scott Haselton: Some of the development that might already be in flight, but at the same time.

308

00:54:50.340 --> 00:54:51.930

Scott Haselton: These some of these.

309

00:54:53.130 --> 00:54:55.620

Scott Haselton: These implementations are pretty fantastic.

310

00:54:56.640 --> 00:55:03.150

Scott Haselton: So, we try to design the schema in a way that you could choose what's going to be the most relevant to you.

311

00:55:04.710 --> 00:55:19.290

Scott Haselton: Without being disruptive we of course would encourage you to stay plugged in with GitHub so you could see the changes that are being made and hopefully bear some fruit from some of those iterations on with your organization.

312

00:55:22.320 --> 00:55:34.140

Scott Haselton: Can the end network file being array so we could split the network file into smaller files, so this was this was in reference to the table of contents.

313

00:55:35.910 --> 00:55:37.770

Scott Haselton: table of contents file itself.

314

00:55:39.360 --> 00:55:50.820

Scott Haselton: And the question is really around this can this can this be an array so we could have multiple files that would be split up for the associated plans themselves and.

315

00:55:51.960 --> 00:55:55.770

Scott Haselton: I touched on this a little bit, we are taking this first iterative step.

316

00:55:56.850 --> 00:56:05.280

Scott Haselton: We wanted to really focus on solving the problem for file naming That was the first Internet of stuff.

317

00:56:06.030 --> 00:56:14.430

Scott Haselton: And because of that we still wanted to have a single file being represented for that table of contents with those associated plans.

318

00:56:15.360 --> 00:56:28.140

Scott Haselton: So, I hear I hear where you're coming from, and this has been asked a couple times, but for right now we're sticking with this, because this is that first initiative step, there may be.

319

00:56:28.980 --> 00:56:39.210

Scott Haselton: In the future, the ability to split some of these files up, but we have to think of a larger problem, or we have to come up with a largest solution if we are going to go down that road, such as.

320

00:56:41.550 --> 00:56:48.870

Scott Haselton: You know at what rate, do you split these files up a how many files or you can allow if you split it up into like 100 files.

321

00:56:50.040 --> 00:56:52.530

Scott Haselton: What does that mean for the consumer and the consumers.

322

00:56:54.090 --> 00:56:59.610

Scott Haselton: bond or spider or whatever automation that they set up they're going to be pinging your network.

323

00:57:00.930 --> 00:57:03.630

Scott Haselton: quite a bit so really coming.

324

00:57:04.950 --> 00:57:17.640

Scott Haselton: coming to terms with what some of those boundaries are with splitting up the files that is a separate pop problem, and that is why we didn't include it in this actual iteration because we weren't solving for that we were solving for the actual file name.

325

00:57:18.780 --> 00:57:19.170

Scott Haselton: But I.

326

00:57:20.520 --> 00:57:21.810

Scott Haselton: I hear where you're coming from with that.

327

00:57:31.290 --> 00:57:38.640

Scott Haselton: Why is the table of contents file approach optional for any entity that needs to consume the machine-readable files.

328

00:57:39.780 --> 00:57:42.660

Scott Haselton: They now have to devote multiple processes consumed to it.

329

00:57:44.070 --> 00:57:47.970

Scott Haselton: Yes, I mean i'm not every plan needs to be.

330

00:57:49.200 --> 00:57:54.960

Scott Haselton: reported in not there, there are unique plans that may be want to be reported in a single file.

331

00:57:56.340 --> 00:58:06.090

Scott Haselton: And we wanted to provide that flexibility if that's the way that you have been building out your files to continue to do so and.

332

00:58:07.080 --> 00:58:19.770

Scott Haselton: We by making this table of contents file optional, we also kind of sidestep this is, this is an interesting problem that that is buddying up of the life cycle of.

333

00:58:20.220 --> 00:58:34.080

Scott Haselton: Software in the lifecycle of policy and those don't necessarily align so by making things optional it actually allows for flexibility without having to invoke and potentially potential policy triggers so.

334

00:58:35.190 --> 00:58:38.220

Scott Haselton: And at the end of the day, it's providing flexibility to.

335

00:58:40.140 --> 00:58:52.860

Scott Haselton: And that's where we landed on that, and I hear you consistency would be key and we don't see anything going outside of those two naming formats.

336

00:59:10.200 --> 00:59:14.610

Scott Haselton: not quite sure what multiple reporting plans can be in different files.

337

00:59:15.900 --> 00:59:23.010

Scott Haselton: there's the new table of contents allow for that type of reporting, I think it does I i'm not 100% sure your question is.

338

00:59:26.190 --> 00:59:34.590

Scott Haselton: But here in the table of contents, the example that we have we have multiple we have multiple plans there with multiple files and.

339

00:59:36.960 --> 00:59:37.440

Scott Haselton: i'm sorry.

340

00:59:42.450 --> 00:59:49.440

Scott Haselton: And, and let me know if this answers your question, if not, maybe we could follow up offline but um.

341

00:59:50.700 --> 01:00:07.020

Scott Haselton: So, we have our table of contents and in this actual sample we have we have multiple plans these plans happen to be identical because they're a part of the same array but then we have an additional plan that's just all by itself that the chip plan and then it's referencing.

342

01:00:08.490 --> 01:00:10.110

Scott Haselton: it's referencing a single file Jim.

343

01:00:26.340 --> 01:00:26.640

Scott Haselton: Oh.

344

01:00:26.730 --> 01:00:38.820

Elissa Dines: My knowledge that we're at little past 430 but Scott has graciously agreed to stay on just a few more minutes to get to some more of these questions so.

345

01:00:39.060 --> 01:00:40.890

Elissa Dines: that's one stick around these.

346

01:00:43.650 --> 01:00:47.040

Scott Haselton: So, there were questions around or there have been questions around.

347

01:00:49.020 --> 01:01:00.480

Scott Haselton: And I see, I think I see it here as well, around service places of service or service codes were there were a single negotiated rate is actually associated with all service codes.

348

01:01:00.990 --> 01:01:11.070

Scott Haselton: And can we create a new code, if you will, that basically represents all new codes and we've been talking about this without any decision just yet, but.

349

01:01:12.390 --> 01:01:14.280

Scott Haselton: It makes sense, it makes sense.

350

01:01:15.810 --> 01:01:30.000

Scott Haselton: I the challenges there, though, on documentation is that you move away from the specific documentation of where the surface codes are on the CMS one side to an additional value that's not actually a part of that set.

351

01:01:31.170 --> 01:01:34.980

Scott Haselton: So just some things to consider and we are we're still considering that.

352

01:01:47.790 --> 01:01:54.450

Scott Haselton: So, for the table of contents update it's really only helping with the naming standard issue correct it is.

353

01:01:55.590 --> 01:02:13.560

Scott Haselton: It does allow it solves for the naming standard while still allowing for multiple plans to be reported for a single file and then I think it provides if looking towards the future provides flexibility for more iterative steps to be made on top of it, but this is that first step.

354

01:02:23.550 --> 01:02:29.790

Scott Haselton: Our first 50 gigabit file was based on the first machine readable iteration.

355

01:02:31.980 --> 01:02:48.180

Scott Haselton: Jeff this okay well that's I I’d be very curious maybe in a future webinar or Q and A if you're able to take some of these different optimization solutions and apply them.

356

01:02:49.650 --> 01:02:53.430

Scott Haselton: To see where you actually were that that 50 plus gigabit file lands.

357

01:02:54.900 --> 01:03:02.160

Scott Haselton: I think, by using or leveraging the provider references that should that should significantly help.

358

01:03:03.840 --> 01:03:06.000

Scott Haselton: Help some of those some of those concerns.

359

01:03:27.570 --> 01:03:40.170

Scott Haselton: yeah this is interesting, will you reduce the length of the attribute names they seem too long and verbose for the machine readable file and pick up at least 50% of all that's um that's a good call out and.

360

01:03:41.280 --> 01:03:52.050

Scott Haselton: yeah there's always that that balance the play between readability and not readability and fully understanding that you know these machine readable files are meant to be.

361

01:03:53.910 --> 01:04:03.720

Scott Haselton: consumed through the process of automation I know hope there is that compression will help out a lot, and with that I.

362

01:04:05.910 --> 01:04:11.490

Scott Haselton: mean the Community can come up with if they want to go through a minute fire, if you will.

363

01:04:13.410 --> 01:04:27.180

Scott Haselton: or Sera laser I There are options there that I mean if the Community is interested in coming up with us here laser that allows for that magnification and then DC realize or to.

364

01:04:28.710 --> 01:04:43.410

Scott Haselton: To bring that modification back to the initial naming I we wanted to make the schema itself actually human readable so that's kind of why we landed where we landed.

365

01:04:46.140 --> 01:04:50.040

Scott Haselton: yeah it's one of those it's one of those mini.

366

01:04:51.600 --> 01:04:57.090

Scott Haselton: Development tradeoffs that you, you have to consider there's no perfect answer in that.

367

01:05:12.630 --> 01:05:23.220

Scott Haselton: Lots of comments or questions comments really about really leveraging those external files and it goes back to what I said earlier, it's.

368

01:05:24.030 --> 01:05:38.070

Scott Haselton: it's coming up with the boundaries of what that looks like how many files can be allowed and do we allow for every provider group do we allow it, for every so it's really solving it's not necessarily a technical problem if you will.

369

01:05:39.360 --> 01:05:40.320

Scott Haselton: it's more of a.

370

01:05:42.600 --> 01:05:44.580

Scott Haselton: it's more of a like a definition problem.

371

01:05:47.640 --> 01:05:54.960

Scott Haselton: And Alyssa I think we could take the rest of these back and try to answer these as best as we can.

372

01:05:56.340 --> 01:06:04.530

Elissa Dines: Okay, that sounds good thanks to everyone for sticking around for a few extra minutes as Scott said we'll take some of these back and.

373

01:06:05.580 --> 01:06:08.100

Elissa Dines: review and as always post.

374

01:06:09.810 --> 01:06:18.420

Elissa Dines: answers to the extent we can on the transparency and coverage website under technical Claire patients call.

375

01:06:20.610 --> 01:06:33.030

Elissa Dines: Yes, technical clarification and, of course, for kind of longer or complex or personalized issues feel free to reach out to us at the transparency coverage.

376

01:06:33.810 --> 01:06:49.710

Elissa Dines: mailbox transparency and coverage at CMS that HHS don't go and take your questions that way and with that I want to thank you all for your thoughtful questions and input and we'll announce a date for our next webinar shortly.

377

01:06:50.790 --> 01:06:52.650

Elissa Dines: Okay, have a great day.