

# Best Practices for Designing Effective Map Services

In this session, you will learn tips and tricks for creating maps that are optimized in terms of design and performance for online use. You will also learn guidelines and procedures for (1) organizing and referencing data, (2) presenting and symbolizing information clearly, and (3) optimizing the drawing performance of ArcMap documents. All these will be discussed in reference to optimization for online uses.

<http://video.esri.com/watch/78/best-practices-for-designing-effective-map-services>

---

## Video Transcription

**00:01** Welcome to the session Best Practices for Designing Effective Map Services.

**00:05** My name is Craig Williams and with me today is Ivonne Seler, and we'll be talking about not only...

**00:12** ... best practices, but kind of the details of, you know, what types of services you put together to make a really good Web map.

**00:22** So, to do this we'll talk about map service planning and design.

**00:27** So, how should you serve up different types of data?

**00:29** What fits in different categories of services.

**00:34** And that ties into ways to serve your map, so talking about cache tiles, dynamic map services...

**00:41** ...client-side graphics, when you should use each one of those.

**00:44** And then we'll follow it up with performance tips for map services, just things you should keep in mind when...

**00:51** ...you're publishing map services.

**00:53** And along the way we'll give a demonstration of using the Map Service Publishing toolbar, to publish an...

**00:59** ...optimized map service, and the steps that that toolbar can take you through to optimize the performance of your map...

**01:05** ...and rout out any problems that you may have in your service.

**01:11** So, one of the things we're trying to move people away from is what we call the Web 1.0 application style.

**01:19** So, to me, this is an ArcIMS map, you know, big map with tons of layers.

**01:25** You go through and you individually toggle them all, and it refreshes and draws dynamically...

**01:32** ...from the map service every time you make a change, and that's slow.

**01:37** So how many of you have seen map services that present a user interface like this?

**01:44** Yes; I don't want to see this.

**01:48** Essentially...I have a colleague at Esri who calls this the Swiss Army knife of maps.

**01:55** So, everything's in there that you need, but you can't find it, and when you go to use it, it doesn't really work that well.

**02:02** And essentially this is putting a lot of desktop GIS workflows onto the Web.

**02:06** Now that's okay if you're a GIS professional, but I imagine most of you are actually serving data out to people who aren't...

**02:13** ...GIS professionals, who maybe don't know the individual differences between all these layer types and things like that.

**02:20** So, why is this important?

**02:22** This is important because we're sharing data with more people, non-GIS professionals, basically that have...

**02:29** ...experience with consumer mapping sites.

**02:31** And these sites have raised the bar, right?

**02:34** Everybody is using Google Maps, or Bing Maps, or whatever they like.

**02:38** And, this one-size-fits-all web map of IMS layers that you turn on and off doesn't really work for...

**02:47** ...people who are used to this paradigm.

**02:50** You get slow performance, they get frustrated, they'll stop looking at your site.

**02:56** So, that's why today we're going to be talking about building a new style of web maps.

**03:03** And, essentially when we're talking about that, we're talking about the main principle is dividing up...

**03:07** ...your basemap data and your operational data into logical groups.

**03:13** You present a small number of layers to turn on and off.

**03:16** You might not even present that with a table of contents-like experience.

**03:19** It may be a button that just all of a sudden draws some extra data.

**03:24** And, you build a really good web map by using a high-performance blend of cache tiles for the data...

**03:31** ...that's not changing that often, or that you can manage the cache updates.

**03:35** Dynamic layers for, you know, data that you're interacting with, or data that's updating quite a bit.

**03:40** And then in some cases you may find the need to use client-side graphics, just to provide the functionality...

**03:45** ...that you may need in your web application.

**03:49** So, when we talk about organizing data into logical groups, this basemap/ operational layer split...

**03:54** ...is the most important thing to keep in mind.

**03:58** Basemaps, typically we're talking about cached map services here, and the idea here is to provide a geographic...

**04:06** ...frame of reference, containing vector data that's maybe not getting updated that often or that you can...

**04:12** ...manage the updates and be able to update the cache.

**04:16** And raster data.

**04:17** So, say you fly aerial photography every six years or something; you can do updates there.

**04:24** And one of the key things with the basemaps that we want to promote is the idea that, you know...

**04:29** ...don't try to put everything in a single basemap.

**04:32** Ideally, you want to reuse these in multiple web applications.

**04:37** You don't want to make that Swiss Army knife map, you want to make a lot of really good maps.

**04:43** And, to do that, you're going to need to reuse some of these basemap services.

**04:47** So you may have an imagery cached map service that you'll reuse in many applications.

**04:53** But you make the investment to build that cache once, and you can reuse it across the board.

**04:59** Operational layers, on the other hand, show a focused item of interest.

**05:03** This may be the item that you're actively working with, or you need to perform analysis on, and the idea here...

**05:10** ...is that they're the main focus of the application itself; they're what supports whatever you're doing in the application.

**05:16** And, typically, when we're talking about operational layers, they're displayed on top of the basemap.

**05:22** So, we'll get into a case study that you're all familiar with, talking about, you know, complex maps.

**05:28** Everybody uses Google Maps at some point.

**05:31** And, when you think about Google Maps, you know, there's the regular maps, street view, there's the terrain view...

**05:38** ...and then there's, like, the satellite imagery view.

**05:42** If you were to actually try to re-create that map, you would find yourself needing to use a lot of different layers.

**05:50** There's a lot of data in there.

**05:53** But you don't think about that when you're using it.

**05:55** It's a single cached service in most cases, and you're just viewing the map, and the cartography's good...

**06:02** ...and things make sense, and you don't have to think about the fact that, you know, there's probably a ferries...

**06:07** ...layer and golf courses and hospitals that someone's managing in a back-end database.

**06:13** The terrain view, for instance, uses a lot of the same data as the regular street map view, but adds things...

**06:21** ...like contours, changes the symbology, et cetera.

**06:24** And you know for the aerial satellite view, they got to manage imagery and source information...

**06:30** ...so that they can actively show you that information as you pan and zoom around the map.

**06:36** So, one thing that you might not think about with Google Maps is that they actually have operational layers.

**06:42** So, if you ever view the traffic information on Google Maps, that's an operational layer...

**06:47** ...actually brought in as vectors and drawn in the browser.

**06:52** There's also the Wikipedia. You know, georeferenced Wikipedia articles show up, photos, you know, the web cams.

**07:00** Even, say, whether or not there's street view available in your area.

**07:04** These are all operational layers that you turn on and off above the basemap.

**07:11** So, we talked about Google Maps.

**07:13** Let's look at some more modern ArcGIS Server examples of these paradigms.

**07:17** So, the first one I'll go to is this Orange County, Florida, property appraiser map.

**07:33** This is a typical map; when you first start viewing it, it shows an overview of the county itself.

**07:37** You can zoom in to it, though.

**07:41** And the drawing performance is fairly good.

**07:45** You can see that it's revealing more information as I zoom in further.

**07:50** You can get down to some parcel levels, shows you some text for the parcels.

**07:54** And then, you know, when you need to interact with a parcel, there's a bunch of tools.

**07:59** And one of the cool things here is they have this nice little view of the parcel information.

**08:06** And you can actually see a little photo of it.

**08:08** And, they actually have links to viewing the Google street view and an oblique view on Bing Maps.

**08:15** And then, some information here.

**08:17** Now look at these fields of information they're showing here.

**08:21** Do you see ObjectID?

**08:27** No.

**08:30** So, Clint Brown, who is director of software products at Esri, he'll say, if you have ObjectID field...

**08:36** ...in your info windows, you've failed.

**08:40** And it's pretty true.

**08:41** I mean, you don't just want a dump of your field values.

**08:43** One of the things that aggravates me is actually, when you have tax assessment values, or values here...

**08:51** ...that, you know, often they're not formatted.

**08:53** That's a monetary value.

**08:55** You know, slap a dollar sign on the front of it, so people actually know that.

**08:59** So, you know, this is just one example of, you know, an organization that's tried to present a more useful view of the information...

**09:08** ...not just dumping out all the attributes from the attribute table.

**09:21** So, how many of you have seen the Solar Boston site?

**09:25** We demonstrate it quite a bit because it's pretty good.

**09:31** It's another example of a different type of report where they're not actually showing you...

**09:36** ...necessarily attribute data. We'll zoom in.

**09:40** What this is about is, it shows you whether or not it would be a good idea, perhaps, to put solar panels on your roof...

**09:51** ...you know, if it'd be cost-effective due to the solar potential.

**09:54** So, you first go to this map and there's not that much blocking your view of the map itself.

**10:02** If you want to interact with it more, there's some tools that can slide in.

**10:05** You can pick these up and move it around.

**10:07** And the tools are pretty simple.

**10:08** You can click on a building, and it brings up this little report, and it's showing you the solar potential during...

**10:17** ...the months of the year, and whether or not this building would be good for, you know, solar panels.

**10:26** So there's some calculation values in here.

**10:28** Talks about the assumptions that this calculation is making and, you know, what you can do next.

**10:37** So these are just some examples of, you know, good sites.

**10:41** I won't necessarily go to the City of Greeley one, but if you look at the slides after User Conference...

**10:47** ...you can visit that site as well.

**10:50** So, okay, now we're going to talk about ways to serve your map.

**10:53** So, I kind of touched on this briefly - cached maps versus dynamically drawn maps versus client-side graphics.

**11:00** What we'll do is we'll talk about each one of these.

**11:04** So, cache tiles are predrawn images.

**11:07** There's some process that cooks the images, writes them to disk.

**11:12** And, because the map is already drawn, these will offer you the best performance and scalability.

**11:19** And these are the standard for all the commercial websites.

**11:21** So, Google Maps, Bing, all the ArcGIS Online services, are cached map services.

**11:28** And, you know, it's really nice that you get this fast and drawing map, but you may be sitting there thinking, okay...

**11:35** ...but I've got to invest in making this cache.

**11:38** And that is one thing to think about, it requires you to create and maintain the cache.

**11:43** So you need to balance that with your data updates in your organization.

**11:49** So, what should you cache?

**11:50** We are going to say that you should always cache basemap data.

**11:55** But, there's also the case of operational layers that you can cache.

**12:00** So, the Google examples I talked about before, these photos and Wikipedia links, they're cached...

**12:07** ...they're tiles that get downloaded to the browser.

**12:10** The other case that no one thinks about in this way, but is almost always cached, is weather data.

**12:16** You go to weather.com, they have this viewer where you can view, you know, the radar weather data on top of...

**12:23** ...Bing Maps basemap, and that's cached images.

**12:29** They're updating the cache very frequently, but they're still cached tiles.

**12:34** The cached tiles offer them scalability and also offer some other added benefits, such as making the ability to play...

**12:41** ...through an animation of those different time set caches easier.

**12:45** So, in some cases you may want to consider caching your operational layers...

**12:51** ...but you'll need to manage the updates and how often you can update that data.

**12:56** So, where do I create a map cache?

**13:00** Where is the cache stored?

**13:02** What if the data changes?

**13:03** There's a lot of things to think about with caching.

**13:07** I'm not going to cover them here.

**13:09** So, there's other sessions. Today is Wednesday, so the early one's already happened, so you got Thursday to think about.

**13:17** And, you know, there's some past presentations from Developer Summit and past User Conferences online...

**13:22** ...if you can't make it this week, that cover these topics.

**13:26** Caching alone is worthy of at least one presentation, if not more, because there's a lot of

different...

**13:33** ...workflows that people have to account for.

**13:38** So, we just talked about caching; well, what should you draw dynamically?

**13:43** Well, think about real-time data; I mean it would be nearly impossible to cache real-time data.

**13:50** The weather guys do it, but if you think about how weather data works, I mean, they're really only...

**13:53** ...updating once every five minutes or so.

**13:57** So they can handle it.

**13:59** Frequently changing data with a large scope, so if you run a national organization with editors editing data...

**14:06** ...all over the country, or if you're worldwide and you're doing it all over the world on a single day...

**14:14** ...managing the changes, and updating the caches may be very difficult.

**14:19** And then, one of the things we commonly see is that for internal map services, if you're serving up services...

**14:25** ...for intranet-based applications that only a few people in your organization are going to be viewing...

**14:33** ...you can probably draw it dynamically; and if that performance is good enough for you, then that's okay.

**14:42** A couple of things to keep in mind when you're talking about dynamically drawn map services.

**14:46** Every time you make a request through a dynamic map service, the server has to go and get the data, then it has to render...

**14:54** ...the data into an image, and then it sends that image back to the client.

**15:00** In the cache case, the only thing that needs to happen is you need to pull that image from the client.

**15:04** And in most cases, that's cached, so if you've been there before, it's already on your browser.

**15:10** So, dynamic map services are slower than caching, but at 9.3.1 we introduced a new optimized map service...

**15:20** ...that improves the performance of dynamic map services greatly.

**15:26** So, optimized map services use a new, faster drawing engine.

**15:31** And we serve up a different file type.

**15:34** So in the past you'd serve up a map document, and with optimized map services you serve up

a map service definition.

**15:41** And the end result is that you get a map service.

**15:45** So, for the web APIs and the REST endpoint, it looks the same and behaves the same, for the most part.

**15:55** Optimized map services offer faster, dynamic drawing than ArcIMS.

**15:59** So, in previous versions of ArcGIS Server, people were resistant to move because their IMS...

**16:05** ...applications drew faster, and that's one of the big items we rectified with optimized map services.

**16:10** The other thing to keep in mind is that optimized map services support a subset of the most common layer types.

**16:17** We chose layer types that we thought made the most sense for dynamic map services, and we continue...

**16:25** ...to add more to that list at each release.

**16:28** But, you know, when you analyze your map and work through the workflow of making an optimized map service...

**16:35** ...you need to think about what's in that service and whether or not it makes sense from a performance standpoint.

**16:41** And the way you create the map service definition, and you can create the service as well, is that you...

**16:46** ...prepare and publish with the Map Service Publishing toolbar in ArcMap.

**16:52** So, to show that workflow I'm going to hand it over to Ivonne.

**16:55** She's going to go through and analyze, fix errors and warnings, preview and then publish a map service.

**17:02** Right. What I... Can you hear me?

**17:10** What I have here is a map of Gulfport in Mississippi, and my goal is to publish it as an optimized map service.

**17:18** So, therefore, first, I need the Map Publishing toolbar and I find this in Customize > Toolbar and Map Service Publishing.

**17:30** In this toolbar I find tools for analyzing my map, so I can find all the issues that exist in my map.

**17:38** I can preview it and see and get a feel and touch for my map service that I want to create.

**17:46** I can then save it as a map service definition file, or I can publish it directly to ArcGIS Server.

**17:55** I can also change the options for my map service that I want to publish.

**18:01** So I'm going to start with analyzing my map.

**18:05** And this opens the Prepare dialog.

**18:08** We have three categories in this dialog; the first is errors.

**18:13** These are issues in your map that you have to solve before you can publish it as an optimized map service.

**18:20** We have warnings.

**18:23** These are changes you should do to your map that are highly recommended to improve performance.

**18:29** And then we also have messages that might help you with performance for your map service.

**18:38** So we have this list that you, for example, can sort by severity from high to medium to low.

**18:46** You see the status: it could be unresolved or resolved issue.

**18:50** You get a description of the error warning or message.

**18:54** And you can also search for specific issues in your map.

**19:00** I'm going to start with the errors.

**19:03** My first error tells me that the layer type is not supported.

**19:07** When I right-click this error, I get a list of solutions.

**19:12** The bold one is always the recommended solution.

**19:15** In my case here, it's "Remove the layer" or I can read the help.

**19:19** And I can also select the layer in the table of contents.

**19:23** So I click this first and I see that it's the buffer points.

**19:27** This is a geoprocessing layer, and geoprocessing layers are not supported in optimized map services.

**19:35** So I right-click again, and I remove the layer.

**19:40** And as you might notice, the status of my error changed to Reanalyze.

**19:45** So what I have to do is I click the Analyze button again, and now I'm down to just one error.

**19:52** And this one is "The layer data source is inaccessible."

**19:56** I can actually already see this here, because I have the red exclamation mark on my layer.

**20:02** So right-clicking this error, I get a longer list of solutions that I could apply to this error.

**20:09** I will choose the first one, Set Data Source, and what it does, it actually calls the repair.

**20:15** So even if I would have more broken layers in this map, it would automatically also repair all the other broken layers.

**20:25** And I already know where my data are, it's md\_watershape.

**20:34** So I will just apply this, reanalyze, and now I don't have any errors in my map.

**20:40** I could already go ahead and publish this map.

**20:44** What I am going to do is I will preview it first.

**20:48** So I click the Preview button, and this opens this Preview window.

**20:55** And very important in this window is the time; it tells me how long it will take to draw this map.

**21:03** And when I open it the first time, it's around a second.

**21:08** I also have navigation tools to zoom and pan around in my map.

**21:15** I'd see the scale, make it a little smaller, and I can also, for example, pan the map to the viewer or zoom the map to...

**21:27** ...the viewer, so I can synchronize what I see in my map document and what I see in the Preview...

**21:33** ...for my optimized map service.

**21:37** So I will zoom my map to the area that I'm looking in my map service.

**21:44** I can also preview it with specific settings for the image format.

**21:49** And see here, I have different formats that I can use. It's depending on what you want to publish; for example...

**21:58** ...if you want to create a cache, it's important which image format you're using, and here you can already get a view...

**22:06** ...how this specific image format would look like in your optimized map service.

**22:13** So after seeing this I will now move on and solve other problems in my map.

**22:24** The first high-severity warning that I have is "Layer's data source doesn't have a spatial index."

**22:33** The spatial index is important to quickly find features, to draw features fast, so I need the spatial index...

**22:43** ...to improve my performance.

**22:45** I right-click the warning, and I use the Add Spatial Index.

**22:53** So this is opening a geoprocessing tool for me that will add the spatial index to my specific layer.

**23:10** And it already selected the right feature class that I want to change, so I can just click OK, close it...

**23:23** ...reanalyze my map, and now this problem is solved.

**23:29** My next two high-severity warnings are for the same layer.

**23:34** It's the hillshade, and this is a raster layer, and this raster does not have pyramids...

**23:41** ...and it's also in a different projection than my data frame.

**23:46** We as ArcGIS Desktop user often very used to the projection on the fly, so for me it often happens...

**23:55** ...that I don't even recognize that I used a layer that has a different projection, because...

**23:59** ...the projection on the fly is just happening in the background and I can work with my map.

**24:05** But when you work with map services it takes a big hit on performance, because every time that you...

**24:12** ...draw the map, you have to do the projection on the fly.

**24:16** So we highly recommend that you have all your data in the same projection when you publish as a map service.

**24:27** For the pyramids here, the only option that I have is the help.

**24:32** The help is available for each and every error message, warning, or information, so that you can read first...

**24:41** ...what problem you have in your map, and then do the right steps to solve that problem.

**24:50** Here you can read everything about the pyramids, what it means, how to create them.

**24:59** I already created a dataset that has the pyramids as well as the projection, so what I'm going to do is...

**25:06** ...I go directly into the table of contents, I open the hillshade properties and I will set the new data source.

**25:20** And here it is, it's the hillshade\_p.

**25:26** Now I reanalyze my map to see if the problem is solved, and you can see that now only the medium warnings are left.

**25:35** So I want to show you that this had an influence on the performance.

**25:39** So I open the Preview window again, and this time I don't need a second to open it, it's down to 0.38 seconds.

**25:48** So that's the new neat thing about the Preview window, you really see that changes that...

**25:52** ...you did really have an influence on the performance.

**25:58** Now I'm going ahead and we'll solve the rest of my warnings.

**26:03** The first one is "Enabling the option to convert layer transparency to color transparency may improve performance."

**26:12** As Craig already mentioned, in 9.3.1 we used a new graphic engine for the optimized map service.

**26:20** And this new graphic engine is capable of drawing the color transparency instead of the layer transparency.

**26:27** And this will improve the performance severely.

**26:32** When I right-click this warning, I have the option Help, because this is a new, this is introduced newly in 9.3.1.

**26:44** We really want you to get informed first before you apply it.

**26:48** But what I'm going to do is I will use the color transparency.

**26:55** And I reanalyze the map.

**26:58** And now you see it's resolved in the map service definition.

**27:02** And what that tells me is that I didn't change anything on my map document itself.

**27:08** But, when I convert it into a map service definition, this changes...

**27:14** ...to "Color transparency will be converted and available."

**27:21** Because this is a problem that is not completely solved in the map document, it still has the...

**27:26** ...status Resolved in the map service definition.

**27:29** If I only want to see the unresolved problems, I can check the Show Only Unresolved Items...

**27:39** ...and now I will just see the five warnings that I haven't touched yet.

**27:46** And they are telling me all the same, "The layer contains a multilayer line symbol whose symbol...

**27:51** ...width may result in aliasing when displayed."

**27:57** Much as the color transparency is new with the new graphics engine, it also has specific properties...

**28:06** ...for the anti-aliasing.

**28:09** So aliasing can appear, for example, I'm not sure if you can see it from that far...

**28:18** ...but you see this jagged line, so this is an aliasing effect; I get this zigzag line.

**28:26** If I use anti-aliasing, I can improve the quality of my graphics.

**28:33** And you can change this in the map service publishing options.

**28:38** So here you can set it once for the feature layers.

**28:43** And this goes from None to Best, and you can do it separately for the text.

**28:50** And this is also from None to Four.

**28:53** When you change these options, be aware that this comes for cost.

**29:00** If you, for example, set the anti-aliasing to Best, it will not draw as fast as if, for example, you have None anti-aliasing set.

**29:15** I have different options here to solve these problems.

**29:19** One is, I could change the width of the symbol, so that I don't have this effect anymore, also on my map document.

**29:28** Or I can automatically correct the line width so that a specific line width will be drawn in the optimized map service.

**29:35** And that's what I'm going to do.

**29:38** So I'm selecting this option for all my warnings.

**29:51** And now basically I have no errors, no warnings.

**29:54** The only thing left are my messages.

**29:59** And the first message I see is "The layer uses a gradient fill."

**30:04** This message is also related to the new graphics engine.

**30:10** Here you would have the help for this, and it tells me basically that it will look slightly different in my...

**30:19** ...optimized map service than I'm used to in the desktop.

**30:24** So in this case I would just briefly look over it and see if the differences are really that big so that...

**30:33** ...I have to take action, but actually it looks quite similar to me.

**30:40** I would just leave it as is.

**30:44** The last messages that I have there, "Layer draws at all scale ranges."

**30:50** In my case, I will leave it that way because I really want all the layers to draw at all scale ranges.

**30:57** But if you create your map services, you really want to make sure that only the layers draw that you want to see at specific scales.

**31:06** Because it's very expensive to draw a layer that has a lot of features at all scale ranges...

**31:13** ...if you don't need it, if you can't even see it.

**31:17** So I would not do any changes here because I'm fine with my results.

**31:21** And I will now publish the map.

**31:26** I have two ways to do this.

**31:27** I can create a map service definition file.

**31:34** I go to my folder where I have my MXD, and I will save it together with my MXD.

**31:43** I do this because every time you want to do changes to your map document, the workflow will be, you do the...

**31:50** ...changes to your map document, you reanalyze the map to see if any introduction that you did actually...

**31:59** ...changed the performance or caused errors and warnings, and then you re-create the MSD file.

**32:07** And you would use this MSD file, for example, to create your cached map service.

**32:13** Or, you can also publish directly to ArcGIS Server.

**32:19** And this opens the wizard that is already familiar to you from ArcCatalog, so I have my server, I have my service name...

**32:31** ...I choose the capabilities that I want; for me, it's Mapping Only; and now I could finish the creation of my service.

**32:42** I want you to pay attention to this area, Map Service Definition.

**32:47** You see that, also here, a map service definition file is created.

**32:52** But I don't choose the location.

**32:55** The location is chosen by the system and it is on my server machine.

**33:00** So here's it's on c:\arcgisserver\arcgisinput.

**33:05** This is where my map service definition file will live.

**33:10** So I can finish this up and then I quickly want to show you in ArcCatalog the preview of my MSD that I can use later...

**33:19** ...to publish to the server and the service that I just published.

**33:31** And I will refresh and preview.

**33:38** And here you see my map and I can zoom it and can have a look, and I also have my map service here...

**33:51** ...and I will see the same map as I just published. Thank you. Craig?

**33:59** All right. Thanks, Ivonne.

**34:03** So, just to recap some of the things that Ivonne covered.

**34:06** There are some cartographic choices that you can make when publishing an optimized map service.

**34:11** So, anti-aliasing is a feature of the drawing engine in an optimized map service, and you...

**34:18** ...can choose it for both features and text.

**34:21** The text setting will also apply to any character or marker symbols in your map.

**34:27** Primarily, this is used to improve visual quality.

**34:30** It's almost always used for basemaps that are being cached.

**34:34** It slows performance, but in that case it's slowing the performance of the caching, which is okay...

**34:40** ...because it's going to produce a better quality product in the end.

**34:44** So, if you look at any of the main commercial, you know, map services that we've talked about, they're all anti-aliased.

**34:54** Since anti-aliasing is part of the drawing engine in optimized map services, it is available though for...

**35:01** ...dynamic map services, and this isn't available when serving a map document, because in that case...

**35:08** ...the anti-aliasing is applied for caching only as a post-processing step.

**35:13** So, one of the things you should think about though when you're choosing anti-aliasing is what image format...

**35:18** ...you're going to be using.

**35:20** So whether you're caching or you're going to be requesting dynamic images, you need to think about that format.

**35:26** Because anti-aliasing, for instance, doesn't mix well with PNG-8 because you're introducing...

**35:31** ...a lot of colors into the map with these gradients of the smooth drawing.

**35:37** One capability of the optimized map service is a true PNG-32 output, and this works really well...

**35:44** ...with anti-aliasing for overlay effects.

**35:47** So, if you've ever used the Google Maps hybrid view, where it's showing some street information on top...

**35:53** ...of the cached tiles of imagery, that's a PNG-32 with anti-aliasing, and that's how they can have smooth text over...

**36:03** ...the image and make it look really nice.

**36:06** There's also this option about choosing color transparency versus feature transparency.

**36:11** I would urge you to read the help document on this subject if it comes up, because it's pretty complex.

**36:18** You've got to make the decision whether or not color transparency will change the appearance of your...

**36:22** ...map, there's a chance that it could, but the reason we've even brought this option to the forefront is...

**36:28** ...that color transparency increases performance, and that's because color transparency can be handled faster...

**36:35** ...inside of the drawing engine, specifically.

**36:38** And then, keep in mind the symbology of your map.

**36:42** We used to recommend, when publishing map document-based map services, to use this Esri\_Optimized style.

**36:53** Despite the fact that it uses the term "optimize," you do not have to use this in an optimized map service...

**36:59** ...but keep in mind that, you know, more complex symbology will slow down your map.

**37:05** So keep it simple if you can.

**37:09** So, what types of data and layer types are available for optimized map services?

**37:15** Well, all geodatabase feature class types, shapefile, SDC, raster data types; you know, annotation layers...

**37:27** ...most of the common layer types that you'll be using.

**37:29** Most 2D symbols, callouts, legends, things like that.

**37:33** There are a bunch of items that aren't supported.

**37:35** How do you know if you have one of those in your map?

**37:38** Well, when you analyze you'll get told that you have one of these items in the map.

**37:41** You can also read the help because this stuff is thoroughly documented.

**37:46** So, each individual item that's not supported is documented in the [ArcGIS] Server Help.

**37:53** So, if you can't use an optimized map service, it's probably for one of these reasons.

**37:59** You may need fine-grained ArcObjects access through the map server API.

**38:05** This is not really a pattern that we're heavily promoting anymore.

**38:08** And, the most common use of it, actually, is the ADF Editor task.

**38:13** If you're heavily relying on the ADF Editor task, I would urge you to visit the feature service sessions this week...

**38:21** ...and learn about the new capabilities of ArcGIS 10.

**38:25** You may need to serve a map that has an unsupported layer type, or an unsupported renderer type.

**38:30** Classic case of this, actually, was the plenary on Monday.

**38:33** If you were at the keynote talk, there was a lot of dot density maps.

**38:37** We don't actually support the dot density renderer in optimized map services yet, so they served MXDs.

**38:45** But, you know, we're working on adding more each release.

**38:48** When possible, break out these unsupported types into separate services, if you can.

**38:56** And you know, a couple of other things to think about that may be in your map - Maplex labeling...

**39:01** ...cartographic representations, they were not supported in optimized map services at 9.3.1.

**39:07** They are at ArcGIS 10, so we added those types.

**39:11** But we really recommend both of those only for caching due to their performance.

**39:15** Maplex, specifically, can be really slow if you have really complex rules; it'll give you great text placement.

**39:21** We use it for all the ArcGIS Online basemaps, because we can't visit and edit all that text by hand.

**39:28** But to use it in a dynamic map service would be way too slow.

**39:34** Cartographic representations, you might be able to get away with that a bit more.

**39:37** And then 3D and chart symbols are another type we don't support.

**39:43** If you find yourself needing to serve a map document-based service, the traditional MXD...

**39:50** ...service, continue to use the Esri\_Optimized style.

**39:53** That will give you a performance benefit.

**39:56** Those are simpler symbols that draw faster in the MXD drawing engine.

**40:00** And, still use the Map Service Publishing toolbar.

**40:04** In this example, there's an error that you won't be able to rectify to serve your map, but you can at least...

**40:10** ...find all those issues like projection on the fly and broken data sources in a huge map, et cetera.

**40:16** So you'll want to use this just to find those deep hidden issues that may be in your map.

**40:24** Now, we've talked about map services, for the most part.

**40:27** Some of you may be going to imagery sessions this week and saying, hey, we haven't talked about...

**40:31** ...image services and how they fit into web maps.

**40:33** You know, when should I use an image service or, you know, cache my imagery.

**40:38** Well, image services are really great for making images available quickly, so you get a big drop of images...

**40:45** ...you need to get them up online fast.

**40:47** And, perhaps you need to modify the image display on a per-request basis; so do things like sharpening...

**40:54** ...or contrast changes, you may want to modify the compression.

**40:58** So, if you're serving out to a device that has low bandwidth, you can request a degraded image...

**41:05** ...but get it there faster and not take up as much network traffic.

**41:11** So, there's many things to think about with an image service, and there's many things to think about with performance.

**41:18** So, map caches are always going to be faster than an image service.

**41:22** But if you need a lot of the interactivity of an image service, you'll want to go down that image service path.

**41:28** And we have help documents, help topics, about kind of the tradeoffs of image services, you can find this...

**41:37** ...in the Image Server 10 documentation, it's called "Optimization considerations for ArcGIS Image Server."

**41:45** Talks about hardware, response time, and then when to kind of bail and cache your map.

**41:50** So think about that.

**41:57** So we've talked about caching, we've talked about dynamic map services, and this third item...

**42:01** ...is what we call client-side graphics.

**42:05** Client-side graphics are different in that we're not requesting images or, you know, requesting...

**42:11** ...the server to make an image and hand it back to us.

**42:13** What we're actually requesting is data. And the web APIs, you'll request this data, it'll come back with attributes...

**42:24** ...and then those features are drawn in the browser.

**42:29** So when should you use client-side graphics?

**42:32** Well, if you have really interactive operational layers for your mashup cases and your web maps...

**42:39** ...so you need to really see the geometry, interact with the geometry, you want to have interactive highlights...

**42:45** ...you'll probably use client-side graphics.

**42:48** Additionally, if you have layers that need to be thematically symbolized on the fly, I'll give a...

**42:53** ...demonstration of this in a minute, but the map server does not support this currently.

**42:59** So, if you want to make a thematic map and have people be able to change the attribute, for instance...

**43:05** ...that is being viewed, or change the color, you can do that with the client-side graphics.

**43:10** And then the other thing to think about is query or geoprocessing results.

**43:14** So, say you run a geoprocessing task and you get back, you know, a set of lines as the result.

**43:20** You may want to draw those as client-side graphics, just to be more flexible and not have to rely on a map...

**43:25** ...service that's tied with your geoprocessing service.

**43:29** So we'll looking at an example of thematic mapping web map.

**43:33** This is actually on the ArcGIS Resource Centers, County Thematic Representation.

**43:40** And I can view it live.

**43:44** What we'll see here is South Carolina, and what we have here is this simple choropleth map...

**43:51** ...shows population in the year 2000 by default.

**43:54** And what I have here is a drop-down that provides a list of different attributes we can show...

**44:01** ...and I can change those and the map will change.

**44:03** And I can also change the color.

**44:06** I can change these colors to whatever I want.

**44:09** And this is happening very fast in the browser because these features have all been brought down.

**44:18** So it's really flexible with what kind of attributes, if you want to explore data in an application like this.

**44:28** A couple of things to keep in mind when working with client-side graphics.

**44:32** You should probably generalize the geometries.

**44:34** That last application was pretty fast because it was South Carolina.

**44:38** If you're dealing with a really detailed geometry of Texas counties, it may not be so fast, because you're...

**44:44** ...bringing over and holding a lot of that geometry and memory in your browser.

**44:48** Be careful not to request too many features, and be aware of your server limits on the feature request.

**44:56** So we keep bumping up this number, each release of ArcGIS, as we get the query functionality...

**45:00** ...to be faster on the server.

**45:02** But you may reach a point where you're actually querying for more features than the server limit is set to return...

**45:09** ...and then you're going to get kind of a truncated list of those features.

**45:14** So, where can you learn more about client-side graphics and all these techniques we've been talking about today?

**45:20** A couple of different places.

**45:21** This session, in particular, is based a lot on a blog article that we wrote in 2008, but it's still really relevant...

**45:31** ...called Design Patterns for Web Maps.

**45:33** It's on the Resource Center.

**45:35** Sterling, who's one of the server documentation leads, posted this.

**45:39** It was really written by Sud Menon, Jeremy Bartley, and Sterling together...

**45:45** ...and it covers, you know, all these strategies for basemaps, client-side graphics, when to handle these different items.

**45:57** The implementation of something like client-side graphics is going to differ, depending on the web API you're using.

**46:03** So, see the web API sessions; there's tons of them this week, for JavaScript, Flex, Silverlight...

**46:09** ...whichever one you're going to use.

**46:11** Additionally, we do have a lot of examples on the Resource Center.

[46:14](#) I can't stress that enough.

[46:16](#) Go visit the Resource Center; each one of these APIs has a resource center that covers examples...

[46:20](#) ...of how to do all these things.

[46:24](#) Now, one of the questions I commonly get asked, after presenting about the web APIs is, which web API do I use?

[46:32](#) Well, this depends on your organization.

[46:34](#) Do you have developers who are going to be doing the work?

[46:37](#) If so, what language are they used to working with?

[46:39](#) What development environment do they have?

[46:43](#) You may have restrictions in your organization about whether or not you can deploy something that uses a plug-in...

[46:48](#) ...whether or not that plug-in is available for everybody.

[46:52](#) We get this question a lot, and to answer it we wrote a blog post, because we finally had to come up with an answer.

[47:01](#) If you're a developer - I mean, this kind of is an easy question to answer, because you know what you like...

[47:08](#) ...but what we saw was a lot of GIS folks getting into this space and starting to write web maps and...

[47:15](#) ...code against the web APIs, and they're not sure where to start.

[47:17](#) So, this topic covers - once again, posted by Sterling - covers all the questions that can help you make that decision.

[47:31](#) All right. Authoring a good web map.

[47:33](#) So, you see these cached map examples.

[47:36](#) You may have a lot of data, but you might not know where to start for making a really...

[47:41](#) ...high-quality example of the cartography.

[47:45](#) Well, look at the map template gallery on the resources site; once again, we've basically posted templates...

[47:53](#) ...for all the ArcGIS Online services. They're small areas.

[47:57](#) You can kind of remove that data, plug in your own data, and stand up these map services very quickly.

[48:02](#) This is how we're doing the Community Basemap work on the world topographic map.

**48:06** People are just downloading the template, plugging in their data, sending us back tiles.

**48:13** Most of these templates are designed for the caching basemap cases.

**48:18** But, there are user community-based templates, so we've worked with user groups and different industries...

**48:26** ...to come up with these industry-specific templates.

**48:29** Some of those templates cover both basemaps and operational layers.

**48:33** For instances, the water resources template covers both of those.

**48:36** And they have maps and applications in these templates.

**48:45** When you're authoring a cached map, you need to keep in mind that you need to design the map for each scale in the cache.

**48:52** So one of the easy ways to do this is to follow these key simple procedures.

**48:57** One is to add all the scales for the tiling scheme to the ArcMap drop-down for scales.

**49:07** You go to this customized list option, you can do that.

**49:12** Prior to ArcGIS 10 you actually had to import a text file of those scales.

**49:16** At ArcGIS 10 you can import a tiling scheme directly, and it'll just suck those scales out of the tiling scheme and put them in there.

**49:23** This will allow you to step through each scale just by using this combo box.

**49:28** The other step that we do, and you'll see this in the templates if you go and download them, is that...

**49:33** ...we create a group layer for each scale.

**49:37** We set up this scale dependency on that group layer, and we use those group layers almost as...

**49:43** ...independent maps to manage the map at each scale.

**49:48** Because you really have to design the map for the scale that it's going to be viewed at.

**49:51** Oftentimes, you will be using the same data between scales.

**49:56** The way you should do that is to copy the layers between the groups and actually have duplicates of those layers, because...

**50:02** ...while you may be using the same data, chances are the symbology's not going to be the same, because you're...

**50:07** ...going to need to filter; you're going to need to use definition queries and change symbol sizes as you move across scales.

**50:13** But, by keeping a single scale in a single group layer, it's going to greatly simplify your experience...

**50:18** ...of managing the data and the layers.

**50:23** Couple of things to keep in mind - less is always more; simple symbology's still faster.

**50:31** Obviously the more complex your symbols are, the more work is being done, slower things can get.

**50:39** People often want to bring print workflows into the web, so get as much information onto a...

**50:44** ...single scale of this map as possible.

**50:48** In many cases, you don't need to do that anymore because you can zoom in to the map.

**50:52** If you want to see more detail, you just zoom in.

**50:54** So you're not trying to cram everything on the image so that people don't have to zoom in and out.

**51:01** In the paper world you did that, because you probably only had one scale of the map.

**51:05** But that's not the case with web maps.

**51:08** Avoid the long TOCs and toolbars and anything desktop-application-looking in your web map.

**51:18** When dealing with client-side graphics, avoid requesting more attributes than you actually need in the client-side graphics.

**51:26** Don't request them all unless you actually need them all.

**51:30** And use intuitive info windows, so get rid of the ObjectID field, you know, put formatting on...

**51:39** ...say, numeric values that indicate a monetary value.

**51:45** And then we'll finish up with a few performance tips.

**51:48** One of the things we like to say is, precompute when possible.

**51:52** A couple common cases of precomputing, annotation, classic case. Annotation always draws faster than...

**51:59** ...labeling, because annotation is precomputed positions; labeling needs to dynamically...

**52:06** ...figure out the positions of the text.

**52:09** Another thing that people don't think about is to query, the precompute query or tool results...

**52:15** ...and I show an example of that.

**52:19** If I go back to this Solar Boston site. Where is it?

**52:25** So I calculated the solar potential for this rooftop, right?

**52:29** Well, if I bring up the tools again, I chose this Select Building option.

**52:35** But what I can actually choose is this option to draw an area.

**52:39** Say I want to put in a big solar field, right there.

**52:45** Well, if I had selected a building, the result would have already come up.

**52:50** And the reason why is that they cache the results for every building in their database, so that you could just select a building.

**53:00** Well, now this guy at the Esri User Conference 2010 in San Diego has just hammered the guy's server in Boston with a...

**53:08** ...geoprocessing request to calculate the solar potential for this crazy polygon, and it takes about a minute to calculate this.

**53:19** So precaching it for all those buildings is a big deal.

**53:24** It greatly improves the performance of the site.

**53:26** You can find your building, click on it, you get the result immediately.

**53:29** If you always had to calculate that result, it would take that long, so now it's up.

**53:35** So, think about things like this when you're designing your web map. Ah! Start over. Where was I?

**53:50** All right. Projection.

**53:53** Ivonne covered this when she gave her demonstration.

**53:59** Often, a touchy subject.

**54:01** People are very tied to their projections and the accuracy of their data, but then you want to serve up your...

**54:05** ...data on top of Bing Maps, and you've got to deal with web Mercator all of a sudden.

**54:10** There's a couple of different approaches to take for this; one of the things that we see often is that people...

**54:16** ...often end up making kind of a shadow copy of their database for serving.

**54:22** They'll do an extract every once in a while and serve that data.

**54:26** You can reproject when you do that.

**54:28** If you're using replication to make that copy of the data, you can actually reproject on a replica.

**54:34** The replica can be in a different projection than the base data.

**54:38** So, say you want to have a really fast map in web Mercator, you could actually define your

replica to be web Mercator...

**54:44** ...and the replication would take care of that.

**54:49** The ultimate precompute when possible is obviously caching, so there's nothing faster than delivering prerendered images.

**55:00** Now on this projection-on-the-fly subject, there's a couple of different ways to go.

**55:04** The best situation is that you have no projection on the fly in your example map.

**55:10** So, you serve up a map, all the data sources are in the projection of that map...

**55:17** ...and the web map that is using the map service is also in that projection.

**55:23** Now, the next step of that is actually projecting on the fly in the map service.

**55:28** So, the map server API actually supports requesting images in different projections, and as a service ...

**55:36** ...publisher, there's not actually anything you can do about that, because anybody can execute that request.

**55:40** You need to keep that in mind if you're developing a web map that's using that service to avoid doing it.

**55:45** But, in some cases it can be ok.

**55:50** Now, serving up a map that has projection on the fly already burned into the map.

**55:55** That's the case that Ivonne went through.

**55:57** We tell you when you're projecting on the fly; try to avoid it if you can.

**56:01** And then the worst thing possible that we always see is that people serve up...

**56:05** ...map documents with map services in them that are projecting on the fly.

**56:12** Now, we don't like people to project on the fly, and we definitely don't like people to put map services...

**56:16** ...in map documents and serve them, because it's like a Telephone game of services.

**56:23** I call this one service that then calls this other service. Who knows if that service is calling another service.

**56:30** It can get really slow.

**56:31** And we don't like it so much that you can't do it in an optimized map service.

**56:36** We don't allow you to embed one service inside of another.

**56:42** On data access side, if you're using SDE, remember to follow the SDE tuning tips.

**56:47** So, tune SDE; use direct connect, if at all possible, because you'll get increased performance.

**56:55** When working with file-based data, avoid UNC paths.

**56:59** UNC paths are really convenient, especially when you're like, using data across your organization.

**57:05** They are slower, though.

**57:06** The alternative is to map a network drive; it's a little bit faster, but for the ideal case you want to access that data locally.

**57:18** Indexes matter for the data.

**57:21** Spatial indexes are really important, and it's not just about keeping them up to date.

**57:25** Keeping them up to date is important, but you also want to have the correct size relative to what data you're requesting.

**57:31** You don't want to have some feature that is skewing the index grid of your spatial index...

**57:38** ...and causing you to return a lot of features on every query.

**57:41** That's just going to slow down map drawing performance.

**57:45** Attribute joins are important for a couple different cases.

**57:49** Joins is the main one, so when you have your join fields, those need to be indexed.

**57:56** And then common queries.

**57:57** So, you may not need the parcel ID for drawing at all, but you may have a web mapping application...

**58:06** ...that allows somebody to type in that ID and zoom to that parcel.

**58:13** Well, in that case you're going to want that field to be indexed, because it's going to make that search faster.

**58:22** So, just to review, we want you to organize your map services into logical groups.

**58:27** Basemaps, operational layers, and you deploy these basemaps and operational layers using these techniques...

**58:35** ...of caching, dynamically serving, or using client-side graphics.

**58:40** And follow all the performance guidelines of the tools in ArcMap that help you guide through...

**58:45** ...potential performance problems, and the other items we talked about today, and remember to precompute when possible.

**58:54** Resources here this week, Esri staff down in the showcase, Hall C, a little bit different island configuration...

**59:02** ...this year; we're in the back, but just look for all the new signs that identify the groups.

**59:08** There's a big server section in the middle, and then kind of the mapping side is on the side of that.

**59:15** Other sessions this week, the caching sessions, all the web API sessions, and there's a lot of other...

**59:20** ...ArcGIS Server sessions about scalability, performance, everything.

**59:25** And then after the conference remember to visit the Resource Centers for more information.

**59:29** If you have any questions, you can prod us for answers and we'll get back to you, hopefully with a blog post...

**59:36** ...if we receive enough traffic on a particular item.

**59:41** So, we'll take questions now.

**59:43** Please fill out the surveys if you can, and have a good day.

**59:56** [Audience question] How do you know what projection a certain service uses...

**59:59** ...and is there any efforts toward standardization?

**1:00:03** How do you know which projection a service uses, and is there any movement towards...

**1:00:08** ...standardization? That's your question?

**1:00:12** Well, there is a movement towards standardization, it's web Mercator, whether you like it or not.

**1:00:18** A lot of people don't like it.

**1:00:20** An example I gave before was if you're in northern Sweden, you don't like web Mercator, 'cause it's ...

**1:00:25** ...a horrible projection for that area.

**1:00:27** But in the United States it's okay, I guess.

**1:00:32** If you're dealing with ArcGIS service, you can go to the services directory and see the projection of the service.

**1:00:39** So, yeah.

**1:00:45** [Audience question] You're talking about specifically, well, on what I call graininess in web map [inaudible].

**1:00:53** I don't know how to describe it any more than that.

**1:00:56** But a lot of times when I'm publishing map services, I get a very grainy display.

**1:01:03** Okay.

**1:01:04** [Inaudible audience question]

**1:01:07** I know exactly what...

**1:01:08** So he's saying that he's seeing a grainy image when he has a map service that he's publishing.

**1:01:15** What that is, is that's called dithering, and you commonly see that with a PNG-8 image format...

**1:01:22** ...when you have more than 256 colors in your map.

**1:01:27** And with PNG-8, it depends on the algorithm that's used to make the palette of 256 images.

**1:01:36** The algorithm that optimized map services uses is different than the one that regular map document-based services use.

**1:01:44** And the optimized map service one is better.

**1:01:46** So you will see that less often; you will see it, occasionally.

**1:01:52** It's sort of a problem because a lot of people, say you dump a hillshade into your map...

**1:01:57** ...you're going to see that problem almost immediately, because the gradient of the hillshade is essentially...

**1:02:02** ...causing more than 256 colors in your map.

**1:02:06** And then, the image creation algorithm needs to figure out, oh, how do I try to make this map look the same...

**1:02:15** ...and that's this dithering algorithm, and it can look quite bad in a lot of cases.

**1:02:20** It's sort of a nasty problem, too, because in most cases in our web APIs, the default format it requests...

**1:02:26** ...if we don't specify it, is PNG-8, and the reason that they do that is because PNG-8 is...

**1:02:30** ...typically smaller, so it comes down faster.

**1:02:33** So if you want to use PNG-8 as the format, you need to make decisions in your map, turn off...

**1:02:42** ...anti-aliasing; you'd want to turn off text anti-aliasing too, probably...

**1:02:46** ...and make sure that you have a limited color palette in your map.

**1:02:52** The other option is to use a different image format.

**1:02:55** So, use PNG-24.

**1:02:57** [Audience question] What about 32?

**1:02:59** Thirty-two would be ok, but 32 can be larger.

**1:03:03** You need to really test the performance and look at the image size of what's being returned.

**1:03:07** If you actually use the Preview window in the Map Service Publishing toolbar, if you look at the drop-down...

**1:03:16** ...the default that we always choose for you is BMP, which is crazy if you think...

**1:03:20** Like, it's crazy to think about, but there's a reason we did it.

**1:03:24** So, the reason it's BMP is because BMP provides a really good baseline for how fast the image...

**1:03:31** ...can be created with no compression, and then as you choose other formats, like say you choose...

**1:03:37** ...PNG-8, you can compare the results from BMP to PNG-8, and actually look at the overhead of...

**1:03:46** ...what the PNG-8 palettization and compression is causing on your map service.

**1:03:52** And if you're dealing with really high performance map services, stuff that draws in, you know...

**1:03:56** ...like point one seconds, that can actually make a huge difference.

**1:04:02** So you need to use the Preview window, and then the other thing, the little secret, is that when...

**1:04:06** ...you deal with the Preview window, the images are written to your temp directory, so you can actually...

**1:04:10** ...look at the image sizes there.