

Geodatabase Essentials Part 2 - An Introduction to ArcSDE Geodatabases

A geodatabase is an object-oriented geographic data model that uses modern relational data storage techniques. Building on the Geodatabase Essentials Part 1, this session provides an introduction to the advanced functionality of multiuser geodatabases. Topics include multiuser editing workflows, such as editing through versioning, versioned editing with the option to move edits to base, and nonversioned editing; archiving; data distribution with replication; and access to spatial data using SQL.

http://video.esri.com/watch/101/geodatabase-essentials-part-2-_dash_-an-introduction-to-arcsde-geodatabases

Video Transcription

00:01 My name is Brent Pierce, and to my left is Gary McDougall.

00:03 We both are members of the geodatabase development team.

00:06 We work in Redlands, and we've been working with enterprise geodatabases for quite some time now.

00:13 And we know a little bit about them.

00:15 And we're pretty excited about them, so we thought we'd just share some of this information with you folks.

00:20 And just a quick poll, a show of hands.

00:22 I know it's early, but how many people are currently using the geodatabase?

00:26 Just raise your hand.

00:29 And then of those...keep your hands up, can't get away that easy...

00:33 ...how many of you are using enterprise geodatabases?

00:36 ...so ArcSDE geodatabases?

00:38 So I'm getting more hands going up, I don't know how that works.

00:44 So it seems like about half of you are already using ArcSDE geodatabases.

00:47 So I'm hoping to convert the other half of you after this presentation.

00:50 So it just depends how well me and Gary do up here.

00:54 So this session is really an introduction specifying on ArcSDE geodatabases.

00:59 How many of you attended the Essentials Part 1 session yesterday?

01:04 Okay.

01:05 So this is really building on top of that.

01:07 And we'll really concentrate on ArcSDE geodatabases in this session.

01:11 So we won't spend much time talking about personal and file geodatabases.

01:14 We're only focusing on the technologies and the functionality specific to ArcSDE level geodatabases.

01:21 So just off the bat, some assumptions we're making when we wrote the slides...

01:25 ...is that we assume that you have a little bit of working knowledge of relational databases.

01:30 Not a...you know, you don't need a lot to understand any of these slides.

01:32 Just to kind of understand what they are and what they can do at a very basic level.

01:37 And also, a basic knowledge of the geodatabase data model.

01:40 And judging by the, all the hands I saw go up in my previous questions...

01:43 ...I'm assuming you know a little bit about the data model.

01:45 About some of the functionalities are available for a geodatabase...

01:49 ...and some of the datasets that are in there.

01:51 We won't focus on these datasets.

01:53 There are specific sessions for these.

01:55 And if you're working with, say...

01:57 ...a topology or geometric network or any one of these special data models within the geodatabase...

02:03 ...I'd definitely encourage you to attend any of these sessions...

02:05 ...because these are really a deep dive into that specific dataset type.

02:11 And I'm going through that.

02:12 And one last thing before we get started.

02:14 If we could hold all the questions 'til the end...

02:16 ...me and Gary will be around to answer any of those questions.

02:19 So just write them down if any questions come up...

02:22 ...and we'll just try to get a flow going in the session.

02:23 So we'll answer any of those questions that you have in the end...

02:26 ...and stay as long as it takes to get those questions answered.

02:30 One last thing, if you could turn off any...

02:32 ...I've got to take paging devices out of there, it just looks old.

02:35 But if you could turn off any cell phones...and possibly get a new icon for a cell phone...

02:39 ...but if you could turn off any of your cell phones...

02:41 ...put them on vibrate we'd really appreciate it.

02:46 So a little bit about where we're going today.

02:48 I'll start explaining, just to give a brief introduction to the ArcSDE geodatabase...

02:52 ...explaining kind of what it is, and some of the considerations...

02:56 ...we, we made when, when creating the enterprise geodatabase...

03:01 ...when first creating this thing called the geodatabase.

03:04 And then we'll go on to versioning, and I'll talk about editing after that...

03:07 ...and then a couple slides on geodatabase archiving...

03:09 ...which is a technology to archive any edits that are made to the geodatabase.

03:14 And then Gary will take over and talk about distributed geodatabases...

03:18 ...and some of the replication solutions that are available through ArcSDE geodatabases.

03:25 So what exactly is the geodatabase?

03:28 And you know, we're definitely guilty of throwing around that term quite a lot...

03:32 ...and using it to refer to a couple different things.

03:37 So I have those things that we typically use to refer to a geodatabase up on the, on the slide here.

03:43 But basically, at its simplest level, you could think of a geodatabase as a physical store or a storage place...

03:51 ...for all geographic information.

03:54 And this storage space is scalable to support many different platforms...

03:59 ...depending on how or what constraints you have as an organization...

04:03 ...as to what platform you can store your data on.

04:05 So the really basic way of thinking about a geodatabase is just a physical store of all

geographic assets...

04:11 ...or geographic information you have.

04:13 But it's not just that physical store, it's also this core data model.

04:18 So it's not just a simple store of the data, but it's also the storage of that data plus this data model...

04:24 ...that allows you to comprehensively model your real-world entities.

04:28 Which is essentially what you're doing.

04:29 You're storing this information in a storage device of some sort...

04:34 ...a database, a DBMS of some sort, but there's also this geodatabase component...

04:40 ...that allows you to model real-world entities better than just a bunch of tables...

04:44 ...and a bunch of joins between those tables.

04:46 So sometimes we use the term geodatabase to refer to that data model...

04:49 ...and that logic that lives within that geodatabase tier of the ArcGIS stack.

04:56 We can also use the term geodatabase to refer to that transactional model.

05:00 So as we know, geoda...geographic information is rarely ever static like the real-world entities that they're modeling.

05:06 This is changing all the time.

05:09 So we have a transaction model in the geodatabase which allows you to manage and edit that information...

05:14 ...to keep it current, keep it up to date.

05:17 And the last thing a geodatabase is, is a set of com, components that allows you to access that information.

05:23 So there's a, there's a pretty robust API involved with the geodatabase that's part of ArcGIS...

05:29 ...and developers can leverage this API to write custom applications and, and do things with their geodatabases...

05:35 ...and support custom workflows.

05:38 So really, the geodatabase is a collection of all these points.

05:41 So it's that physical storage, it's the more complex behavior, that geodatabase/ArcGIS data model....

05:49 It's the transaction model for mod... managing GIS workflows, and it's also a set of APIs that can be used to extend that programmatically.

06:00 So when we first started creating or thinking about creating a thing called the geodatabase...

06:05 ...we took a couple of things into consideration, and the first thing was, you know...

06:09 ...when we were creating this thing, there's already databases out there that, that did some things very well.

06:14 So what they did very well is they had a very good base relational model.

06:18 They had a very solid model for having tables and relating these tables...

06:23 ...and enforcing that referential and relational integrity down in the, in the relational DBMS.

06:31 They also had a really solid short transactional model.

06:34 So the transactional model is really a way to have multiple people edit the same information in a consistent way.

06:40 So DBMSs, databases were doing this very well at the time.

06:45 And DBMSs are also notoriously...

06:47 ...maybe not notoriously, but they're also very reliable.

06:51 They're very flexible...

06:52 ...and they scale from very small setups to really, really large setups across multiple servers.

06:58 So we knew DBMSs, databases were doing this very well, so we just chose to extend that.

07:03 And that's really the approach we took when developing the geodatabase.

07:07 So we extended those relational models by building simple features and then adding logic to these simple features.

07:14 So the simple features you can just think of as tables in a database.

07:18 And really the functionality that we add is the logic.

07:22 So, you know, we have this thing called a feature class, which is just a simple feature at table...

07:27 ... plus logic so that table knows it is a river or it's a collection of rivers or a collection of streets.

07:35 Also, what this does is allow for this functionality to be consistent across multiple DBMS platforms.

07:41 So this was a key consideration when we were building the geodatabase.

07:44 We wanted all that functionality...

07:46 ...to look the same whether your feature class was actually stored on Oracle or it was stored on Access...

07:51 ...or a personal geodatabase or on disc in the file geodatabase.

07:55 The end user's experience, we... really was key to the, that this feature class would behave the same way...

08:01 ...independent of how that feature class was stored.

08:09 Some other things we had to consider when building a geodatabase was editing and data compilation.

08:15 You know, this is a very key task that many GIS professionals do on a daily basis.

08:20 So we really needed to provide a very rich set of editing tools, which ArcGIS currently has.

08:26 And since it's a geodatabase, we had to maintain both attribute and spatial integrity.

08:34 So databases were already maintaining attribute integrity very well.

08:38 We had to build a whole tier in the geodatabase to maintain that spatial integrity.

08:42 So as you changed feature's geometry, as you changed the path of that river or the...

08:47 ...added a new street, we had to maintain that spatial integrity in a very consistent way.

08:52 We also want to provide a nicer editing experience in providing Undo/Redo to that editing...

08:58 ...so you can undo and redo edits that you make within ArcGIS or in a geodatabase...

09:04 ...just like you were undoing/redoining edits to a Word document or a PowerPoint presentation.

09:10 We also needed to support multiple users editing the same information as the, at the same time.

09:15 That was a very large part about what we needed to build the geodatabase to do.

09:20 And we do that through supporting a thing called versioning...

09:23 ...and I'll talk a little bit more, quite a lot about versioning later in the presentation.

09:28 We created this thing called the long transaction...

09:30 ...which really just extends the database's short transaction.

09:34 And based on versioning and that ability for multiple users to edit the same data...

09:39 ...at the same time in a relational database...

09:41 ...we also extended that versioning model to support replication...

09:44 ...or support distributed data models and data management through replication...

09:50 ...and also archiving, which is the ability to archive edits.

09:55 Lastly, what we needed to build into the geodatabase was a way to customize or extend what the geodatabase can do.

10:03 And we did so by building a API that can be extended and leveraged by developers.

10:08 So there's a pretty comprehensive API associated with the geodatabase...

10:13 ...and we, we took effort to make that API very comprehensive as we were creating the geodatabase.

10:19 So people can take the geodatabase and extend it to meet their custom workflows.

10:28 So there are three different types of geodatabases that are available.

10:32 There is the personal geodatabase, which is on the Microsoft Access...

10:35 ...built on top of the Microsoft Access platform.

10:38 And this is really a single editor/multiple readers type geodatabase.

10:42 So you're going to have a single editor at any given time with the Access geodatabase.

10:46 At 9.2, we added the file geodatabase, which is also a single editor/multiple reader type geodatabase.

10:53 But instead of being based on top of Microsoft Access...

10:55 ...this is really just a folder of binary files on disc.

10:59 And what this means is since it's a folder of binary files, it's cross-platform...

11:03 ...so it can be supported both on Windows platforms and non-Windows platforms...

11:07 ...which isn't necessarily true with the Microsoft Access geodatabase or personal geodatabase.

11:12 But for this presentation, I'm really going to concentrate on ArcSDE geodatabases.

11:16 And really what differentiates these types of geodatabases...

11:18 ...is that these are built on top of enterprise commercial DBMS platforms.

11:28 So as I mentioned before, really what differentiates an ArcSDE geodatabase...

11:33 ...is the fact that it's built on top of an RDBMS.

11:37 And because it is built on top of a relational database...

11:40 ...it allows for some fairly unique functionality.

11:43 So this allows us to be open and interoperable across many different DBMSs.

11:49 So currently, we support five different DBMSs.

11:52 We support Oracle, SQL Server, Postgres, Informix, and DB2.

11:58 And this data, data in one, that built on top of one DBMS is easily portable to data on another DBMS.

12:04 So this makes it very open and inner, interoperable across an organization that might be using multiple...

12:09 ...different types of, of DBMSs to store their information.

12:14 Because it's built on a relational da... DBMS and those all support an open SQL standard.

12:19 You can use SQL to access any aspects of your database.

12:22 And we've taken on great strides at 10 to open up the geodatabase to SQL developers.

12:28 It's a very big thing when you have third-party applications that...

12:31 ...just want to use a standardized SQL specification to get at information within a geodatabase.

12:37 So we've, we've definitely changed the schema around to provide it...

12:41 ...make it more open for people that need access to information in a geodatabase...

12:45 ... directly through SQL.

12:47 What this also means is that some...one...

12:49 ...some of the key unique functionalities are versioning, replication, and archiving...

12:53 ...in ArcSDE geodatabases.

12:54 And these technologies are really only available in this type of geodatabase...

12:58 ...and they aren't available in file and personal geodatabases.

13:06 So this is the classic, you know...

13:08 ...what we call the layer cake of, of the ArcGIS stack.

13:11 At the top, you have ArcGIS, the application...

13:15 ...so you have ArcGIS, you have things like ArcGIS Server in that top tier.

13:19 You have ArcGIS Desktop.

13:20 So ArcMap and ArcCatalog are there.

13:24 Sitting below that is the geodatabase tier.

13:27 So in the geodatabase tier, that's where all the logic lives to define what a feature class is...

13:32 ...define what a relationship class is...

13:34 ...define how a geometric network traces and gets created and gets managed within the geodatabase.

13:42 Now between the geodatabase tier and the DBMS tier...

13:46 ...is what we call the ArcSDE tier...

13:48 ...and this is, this is a tier.

13:50 It's not a geodatabase in this diagram.

13:52 And it is really just the layer between the geodatabase and the DBMS.

13:57 And what the ArcSDE tier really does is...

14:00 ...it provides a kind of DBMS agnostic way of interacting with the databases.

14:07 So the geodatabase tier does not need to know anything about the specific database...

14:12 ...it's going against.

14:13 ArcSDE knows that information...

14:15 ...and it formulates the queries that are necessary to work with that DBMS down in the ArcSDE tier.

14:21 So it means the geodatabase can just create a feature class...

14:24 And the ArcSDE tier says, oh, this is an Oracle database...

14:28 ...so I need to create this feature class in this specific way.

14:32 And then of course, on top of the operating system, all this stuff sits...

14:36 ...so it's, it's standing on top of a specific OS.

14:39 When we use the term ArcSDE geodatabase...

14:41 ...we're really incorporating the geodatabase tier, the ArcSDE tier, and the DBMS tier...

14:46 ...all in one.

14:47 And that's, when we refer to an ArcSDE geodatabase...

14:49 ...it's really a functionality available across those three tiers.

14:53 When you connect to and work with an ArcSDE geodatabase...

14:57 ...you would see this view in the Catalog, ArcCatalog, or the Catalog window.

15:01 So you see this view which you see objects such as rasters.

15:05 You see objects such as feature classes.

15:07 You don't work with the tables directly.

15:09 So you see this geodatabase behavior any time you connect to an ArcSDE geodatabase within ArcGIS.

15:21 So of the ArcSDE geodatabases, there's actually three different editions that we added at 9.2.

15:27 And why we did this was to provide a solution for end users that can scale from very small,

personal systems...

15:34 ...all the way up to large enterprise geodatabases.

15:38 So when choosing between editions, there's a couple key factors that you should take into consideration.

15:44 And the first one is, how many users are currently using your system?

15:47 So this is a big difference between the three editions.

15:50 So ArcSDE for desktop really has a max of three users with only one concurrent editor.

15:57 So if you're going to need multiple people to edit at the same time...

16:01 ...you probably wouldn't want to use a desktop edition of ArcSDE.

16:06 Now with Workgroup edition, you can have up to 10 clients or 10 users connected at any given time.

16:13 And those users can be editing, those users can be reading, it really doesn't matter.

16:16 Just 10's the magic number here.

16:18 Now with Enterprise edition of ArcSDE, there's no limits on the number of users could be connected.

16:22 Those limits are really only at the database level.

16:25 So if your database... the specific DBMS can support 500 users at any given time, then...

16:33 ...the geo, enterprise geodatabase can support that many users.

16:37 One other factor, taken into consideration is which DBMSs are supported.

16:41 So for both Desktop and Workgroup, those are built on top of SQL Server Express.

16:47 This is a free edition of SQL Server.

16:51 Now if you have needs to use maybe a more robust or more scalable geodatabase...

16:57 ...then there is, sorry, DBMS...

17:00 ...Then Enterprise edition is a very good choice...

17:02 ...because in this edition, they support the five DBMSs that are supported...

17:05 ...and for SQL Server, this is the full-blown version of SQL Server and not the, not the Express edition.

17:13 And a little more on that Express edition and something to keep in mind is that...

17:17 ...because both Desktop and Workgroup are supported only for SQL Server Express...

17:23 ...SQL Server Express has some built-in throttles in that free database...

17:27 ...and those throttles are that the max database size can only get up to 4 gigabytes, so...

17:33 ...something to take into consideration when you're trying to choose which edition...

17:37 ...is how big is your data going to get, and is it going to exceed this limit?

17:42 Also, there's some memory or virtual memory throttles here, so there's...

17:46 ...it'll only ever use 1 gigabyte of RAM that you have on the server machine.

17:51 And if you have a multiple CPU machine, it only ever uses single U, CPU at any given time.

17:56 Now with Enterprise, since it uses the full-blown version of SQL Server...

18:01 ...and supports a full-blown version of SQL Server...

18:03 ...as well as the fuller versions of all the other DBMS platforms...

18:06 ...there are really no limits to the database or the database's performance.

18:11 So those are three real key factors to take into consideration when you're trying to choose...

18:16 ...which actual version or edition of ArcSDE to implement.

18:22 So once you've chosen which version of ArcSDE to implement...

18:26 ...you really want, need to decide how you're going to make that connection...

18:30 ...to the ArcSDE geodatabase.

18:32 And there's really two different ways to make a connection to an ArcSDE geodatabase.

18:36 There's what's called an application server-type connection.

18:39 This is sometimes referred to as three tier.

18:42 And it's referred to as three tier because the GIS client...

18:44 ...let's just say that's ArcMap or ArcCatalog...

18:48 ...connects through what's called an application server.

18:51 I have some shorthand on the screen app server.

18:53 And this is a service running that really brokers connections to the geodatabase.

18:58 Now this application server is actually doing all the heavy lifting involved with...

19:03 ...connecting to and working with datasets in an ArcSDE geodatabase.

19:08 So wherever that application server is running...

19:11 ...wherever it's living...

19:12 ...that will take the most overhead of all the connections.

19:15 And all application server connections will go through that one app server.

19:21 And now a different type of connection is what's called the direct connect...

19:25 ...sometimes referred to as two tier because there's really only two tiers with a direct connect.

19:29 There's the GIS client, say, ArcMap or ArcGIS Server service...

19:33 ...and that's connecting directly to the DBMS, directly to the ArcSDE geodatabase.

19:38 So what this means is all the heavy lifting will be done on the GIS client.

19:44 But you're spreading out the ... you're spreading the... the work that needs to be done...

19:50 ...to work with the ArcSDE geodatabase to each individual client.

19:53 You're not using a single application server here, so...

19:56 ...each client does their own work and, and connects directly to the geodatabase.

20:02 So depending on how many people you'll have connected...

20:05 ...depending on how your system is architected...

20:09 ...you want to decide whether to go with an application server or go with a direct connection.

20:17 Just a little note on backwards compatibility.

20:20 I get a quite a few questions of this every year, especially when we come out with a new release.

20:24 So ArcGIS 10.0, we just released, will be backwards compatible with, with these versions...

20:32 ...of ArcSDE.

20:34 So 9.1, 9.2, and 9.3 ArcSDE geodatabases.

20:38 So you'll be able to connect from ArcCatalog, ArcMap, ArcGIS Server 10.0...

20:44 ...to any of these types of geodatabases.

20:46 And by 9.3, I also mean 9.3.1.

20:49 So technically, there really isn't a 9.3.1 geodatabase.

20:52 We didn't, you didn't have to upgrade the geodatabase to 9.3.1...

20:56 ...so there really is just 9.1, 9.2, 9.3.

20:59 And of course, from ArcGIS 10, you'll also be able to connect to 10 geodatabases.

21:04 It'd be a pretty big oversight if we didn't allow that.

21:10 So that's really just a quick introduction to ArcSDE geodatabases.

21:14 The next thing I'd like to talk about is versioning, which really is one of the key technologies...

21:17 ...the key reasons why you might want to use an ArcSDE geodatabase.

21:24 So versioning is a technology really that allows multiple users to edit and work with the same data at the same time.

21:33 To those users, it really appears as if they have...

21:37 ...they have their own version, their own snapshot of the data.

21:41 And it does so without applying locks or duplicating any data.

21:45 So you see here in the graphic I have, those readers in blue are working with the default version...

21:52 ...of the geodatabase, and they see a certain representation of the data...

21:56 ...where editors are working with the Analyst version.

22:00 The editors in yellow on the screen are working and editing the Analyst version.

22:04 Those readers, since they're connected to the default version, will never see the edits...

22:08 ...the analysts are putting, or the editors are putting within the Analyst version...

22:13 ...until those editors decide to push those changes into the default version.

22:18 So it really provides users with an isolated view of the geodatabase...

22:22 ...depending on which version they choose to interact with.

22:29 In ArcGIS, you connect to a version.

22:32 So at the bottom of the slide, I have a screen grab of the version manager.

22:36 And this just shows all the different versions in that geodatabase.

22:40 And a version has a bunch of different properties.

22:44 So all versions have an owner.

22:45 They have to be created by somebody.

22:47 They usually have a description.

22:49 They have a permissions level, which controls who can edit it, who can work with it...

22:54 ...who can see it within an ArcSDE geodatabase.

22:57 And versions are created from other versions.

22:59 So all ArcSDE geodatabases have a single version to start with, called the default version...

23:05 ...and then all versions will be created from that default version.

[23:10](#) So you can see here, I'm looking at the Parks feature class...

[23:13](#) ... and its representation within the default version.

[23:16](#) If I was to change to the QA version, I'd see a different view of what the Parks feature class looks like in that version.

[23:27](#) So why would you ever want to use versioning?

[23:30](#) One of the key reasons is that if you have multiple editors accessing your data...

[23:35](#) ...versioning is a really nice solution for you because ...

[23:39](#) ...those editors can edit their own copy of the data...

[23:41](#) ... and only merge those changes into the version that everyone else is working with...

[23:46](#) ...after that editing has been completed.

[23:49](#) Also, if you need to have different, different views or different snapshots of the data.

[23:53](#) So say you're working on a couple different proposals.

[23:56](#) You might want to have a version for each proposal...

[23:59](#) ... and just connect to those different versions...

[24:01](#) ...when you want to see proposal 1 versus propose 2 versus proposal 3.

[24:06](#) Also, if you're editing any complex data within ArcGIS...

[24:09](#) ...that data has to be what's called register as versions.

[24:11](#) So it has to be version aware...

[24:14](#) ...and you have to use the versioning infrastructure to edit any complex data such as geometric networks.

[24:21](#) Also replication and archiving really are extended from versioning...

[24:34](#) So this is an example of editing with what's called the long transaction...

[24:38](#) ...which is really just what versioning creates.

[24:40](#) So this is where you want to isolate work across multiple sessions.

[24:44](#) So if you're going to edit geographic features, often this takes a little while.

[24:48](#) This might take a week to put all the edits in.

[24:51](#) You don't want to put those in piecemeal.

[24:52](#) You want to actually isolate those in an Analyst version.

[24:55](#) And then when that actual work order is complete...

24:58 ...then you can push it into the version that the public is consuming, the public is reading.

25:03 Another case here is if you want to model what-if scenarios.

25:06 So you got all your infrastructure created in your public version.

25:10 But maybe in some flood version, which was created from the public version...

25:14 ...you want to model a what-if scenario.

25:16 What if this levy wa, were to break?

25:19 What if this pipe was to burst?

25:20 How would my infrastructure handle this?

25:22 And you could use all the data modeling tools to be able to, to model this out and, and map any...

25:27 ...kind of what-if scenarios through versioning.

25:31 Last real ... a case here is a workflow management.

25:35 So I've seen versioning used to manage the different stages of a project.

25:39 And the example I have here is a residential development, going from farmland...

25:43 ...and then creating residential parcels.

25:45 You might create a version for each specific stage of that project.

25:52 So, really versioning lends itself well to editing, and there's really three types...

25:56 ...three different ways you can edit geographic information.

26:00 So we'll talk a little bit about editing.

26:06 So in the geodatabase, there's really just two types of transactions.

26:09 There's a short and a long transaction.

26:12 And really as the name would suggest, a short transaction is designed to be completed very quickly...

26:18 ...and this is what DBMSs have been doing forever and doing quite well.

26:22 So a good example of a short transaction is an ATM transaction...

26:26 ...updating library records, you know, punching that time card at the end of the day.

26:30 This just goes into the database very quickly, makes an update, and then commits it.

26:34 It doesn't leave the transaction open very long.

26:37 And what this means is that transactions are isolated, so until those edits are committed...

26:42 ...I lock the data, and no one else can see my edits until I decide to either commit them...

26:48 ...or roll them back.

26:51 The opposite of a short transaction is a long transaction.

26:53 And this is really a large number of operations over very long period of time.

26:58 So this, this is more GIS-type editing.

27:00 So if you're updating parcels, if you're just updating general geographic information...

27:04 ...typically you can't lock up that data for long periods of time.

27:08 So you have to use what's called a long transaction to edit that information.

27:13 And if you use a long transaction, you could have multiple users editing this...

27:16 ...and they're not going to lock the data.

27:19 They're not going to duplicate that information...

27:21 ...and editors can work with this unique and isolated version or view of the geodatabase.

27:26 But often what we're seeing is GIS editors need both short and long transactions...

27:31 ...so we've provided solutions for both different types of transaction models in the geodatabase.

27:40 So there's three different ways of editing information within a geodatabase.

27:44 You can use a long transaction with what's called version editing.

27:48 You can use a short transaction with nonversioned editing and editing nonversioned data.

27:54 And the third way is editing directly through the Structured Query Language, or through SQL.

27:59 And this really uses the database's native short transaction model here when updating information.

28:04 I'll talk a little bit about each, each of the three different models here.

28:09 So version editing, you're actually deciding which version to make your edits. So...

28:16 ...so it's editing done directly through a version in your geodatabase.

28:20 And because it's done directly through a version...

28:22 ... it supports concurrent editing of the same data at the same time...

28:27 ...through use of long transactions.

28:29 So this means, like, editor 1 and editor 2 in the graphic...

28:33 ... can work with that information as long as they want.

28:35 They can edit those, their own design versions and work order versions...

28:39 ...until that work order or design has been completed.

28:43 And then they can push these edits into the default version.

28:45 They can be consumed by the readers in the graphic.

28:49 So this also allows for a nice undo/redo editing experience within ArcGIS.

28:54 And it doesn't lock up that information.

28:55 So as editor 1 and editor 2 are working with the data...

29:03 So they can read the latest as-built representation of the features...

29:08 ...and that information isn't locked from any readers who might be consuming the geographic information.

29:18 So how does this black magic of versioning really work?

29:21 Well, the first thing that you have to do is you have to register a, a dataset...

29:25 ...register a feature class or, or a table as versioned within ArcGIS.

29:31 And this can be done through ArcCatalog, and...

29:34 ...and if you register a class as versioned, all we do is we create two tables.

29:38 Collectively, these are referred to as the Delta tables.

29:41 But they're really an Adds table and a Deletes table.

29:43 And these tables will be associated with that feature class, with...

29:47 ...what's called the base table.

29:49 So when you register a class as versioned, you're saying...

29:52 ...I want to make versioned edits on this feature class.

29:55 And we, behind the scenes, create the Delta tables...

29:57 ...and we use these Delta tables to track the changes that are made on versions to that...

30:04 ...that class or that dataset.

30:06 So once these Delta tables are created, any edits that are made are tracked.

30:10 So here I have an example of adding feature 6.

30:14 So you go into ArcMap and you just insert a new feature.

30:18 Well, what that does is it adds a records to, adds a record to the Adds table...

30:23 ...and references which version that edit was made on.

30:27 So notice how it's not inserting into the native base table at all.

30:31 It's actually putting these changes into the A and D tables, into the Delta tables.

30:38 Now, instead of inserting a feature here, I'm deleting feature 5.

30:42 So as you could have guessed, it wouldn't put a record into the Adds table.

30:45 It's actually going to put a row into the Deletes table.

30:48 So it just says I deleted feature 5, and this is the version, this is the version state ID...

30:54 ...where this record was deleted.

30:56 And finally, updating a feature really involves an insert into both the Adds and Deletes table.

31:03 So we first have to write into the Deletes table to see, say, that this old representation...

31:09 ...of feature 2 has been deleted at this version.

31:13 And then we add a row to the Adds table, saying this is the new shape...

31:17 ...or the new representation of feature 2...

31:19 ...and this is the version where this feature 2 has been updated or modified.

31:25 So that's why we're not duplicating data.

31:27 As we're just tracking changes in these Adds and Deletes tables.

31:30 We're not doing wholesale copies of the feature class at all.

31:32 We're just tracking any changes that are made to the versioned records.

31:37 So how does this whole thing work together?

31:39 So you have all these edits in the A and D tables...

31:42 ...and really a versioned representation of a feature class is a combination of the base tables...

31:46 ...those core tables, filtered through any edits that are made on the A and D tables.

31:52 So you're connected to a version in ArcGIS.

31:55 We look at the base table, and then we look at any adds or deletes that are made on that version...

32:01 ...and then we return the results to ArcGIS.

32:04 That's when, that's why when you connect to a version...

32:07 ...you see that version's representation of the Parks feature class...

32:10 ...or that version's representation of the owner's table.

32:14 Because we're filtering it through the A and D tables that are associated with that particular class.

32:22 At 9.3, we added some new functionality to make it easier to see the differences between versions...

32:29 ...and we call this, this dialog the Version Changes Viewer...

32:33 ...and it's available off of the Versioning toolbar in ArcGIS.

32:37 And really what it allows is to see how a version has changed relative to another version.

32:43 So you can view these changes and you don't have to do what is called a reconcile...

32:47 ...which is a way to merge changes between versions to see how one version is different...

32:51 ...than another one.

32:53 And you can not only see how the attributes are different in that grid on the top right...

32:57 ...but you can also see in the two map controls in the bottom how the geometric...

33:01 ...or geographic representation of that feature is different.

33:08 So you can use versions to, to edit information as I've explained...

33:13 ...but there's always the case of merging changes between versions.

33:16 So you have all your editors that are editing their...

33:19 ...edit versions, say, QA in this case, in the example.

33:22 But after they're done, their work order, they're going to want to merge their changes...

33:26 ...into the default version because everyone else in the organization is looking at the default version.

33:30 I have a bunch of map services that are pointed to the default version.

33:34 So you can do this merging through a process that we call reconcile and post.

33:39 And the first part of the process, reconcile, takes any changes that are made in the default version...

33:45 ...and merges them with the QA version. So...

33:48 ...so the workflow here is, I want to take the changes made in QA and I want to push them up to default.

33:53 Well, we have to first reconcile with the target, default, in this case...

33:58 ...just to determine, has default changed at all?

34:00 Because I don't want to override any changes that were made independently in the default version.

34:05 So we first reconcile, and we try to detect any conflicts.

34:08 And what a conflict is, is if a feature has been changed both in the default and the Edit version, or sorry...

34:15 ...the target and Edit version.

34:17 So in this case, if one of those parcels had been changed in both the default by some other editor...

34:22 ...and the QA version by, by another editor...

34:26 ...we would flag this feature as being in conflict, and we do this during a reconcile.

34:31 And this is to avoid any overriding of changes, so we'll, we'll...

34:36 ...we won't lock the data at all.

34:38 So there's a case where you could have where the feature changed in both the target and Edit version...

34:44 ...and we really want to notify that editor that's doing the reconcile, that's doing the merging process...

34:49 ...hey, these features are in conflict...

34:51 ...you have to pick which representation of this feature is correct.

34:55 So we tried to make this as easy as possible by providing the Conflict Resolution dialog in ArcGIS...

35:01 ...and this is just a way to resolve conflicts.

35:04 So you see all the features that are in conflict during a reconcile...

35:07 ...and you can choose to pick the version of the feature that I've edited in the QA version or, no...

35:13 ...actually, whoever edited the default version is actually correct.

35:16 I'll take their representation of the feature.

35:18 So this is just a way to graphically go through all the conflicts that you encounter during a reconcile...

35:24 ...and resolve these in, in a very efficient way.

35:28 So after all the conflicts have been resolved... oh, got a little ahead of myself.

35:32 At 9.3, we've actually augmented the conflict resolution process by allowing you to merge geometries.

35:40 So what a lot of people were asking for was for very large features...

35:50 One editor edited up in Northern California...

35:53 ...I'm editing the coastline down in Southern California around San Diego...

35:57 ...and that coastline feature will always be in conflict, but really, both of these edits are valid.

36:02 I don't want to take one or the other, I want to take both.

36:05 So at 9.3, we added the ability to merge both these edits.

36:09 So you can take both coastline edits as long as they're not directly overlapping.

36:13 So this is really the ability to merge geometries.

36:16 And this is a, a requested feature by a lot of people that are editing very large polygons...

36:20 ...or polyline features and run into conflicts.

36:24 So once you have reconciled and dealt with any conflicts that might arise during the reconcile process...

36:31 ...the next stage is to push those changes into the default version, and we call this process a post...

36:37 ...and really you're posting any of those changes from the QA version into the default version.

36:42 So after you've posted from QA to the default version...

36:46 ...both of these versions point to what's called the same database state, or they point to...

36:52 ...the same state in the database.

36:53 So it means from the end user's perspective, these versions look completely identical.

36:57 So all the changes that were made in QA are now in default...

37:00 ...all the changes made in default got pulled into the QA version during reconcile.

37:08 So probably the best way to show this is through a quick demo, so I'll just jump into...

37:13 ...into ArcMap here and show you a quick versioned editing scenario.

37:19 So here I have a geometric network and I have a bunch of buildings.

37:23 All this information has been registered as versioned.

37:26 So I already went into ArcCatalog, registered all this information as versioned.

37:30 The A and D tables have been created.

37:32 And I've connected to my QA version through the Change Version dialog.

37:37 So I'm looking at the QA, the representation of the buildings and the network...

[37:42](#) ...as it exists in the QA version, and the QA version is what I'm going to use to do all my edits.

[37:47](#) So I'll first start an edit session here.

[37:52](#) And we have this new Create Features Pull Right at 10...

[37:55](#) ...which is the new editing experience.

[37:56](#) So I can select which type of feature I want to create.

[37:59](#) So I just want to quickly digitize a building in.

[38:01](#) So I'll select Building and ... let me make some quick edits.

[38:07](#) I'm not much of an editor, so bear with me here.

[38:12](#) So I'm digitizing in this building and inserting this feature here.

[38:17](#) So I've inserted that feature, I'll delete this shed here.

[38:22](#) And then I want to modify geometry, so I'll double-click on this feature...

[38:26](#) ...and just delete a couple vertices.

[38:37](#) So currently, these changes are all isolated to my own version.

[38:40](#) But take the scenario of the fact that I've been editing the QA version maybe for two months...

[38:45](#) ...and it's finally time for me to merge my changes in.

[38:48](#) Well, I can use the new Version Changes viewer to merge these changes in, so...

[38:53](#) ...in the Versioning toolbar, I have the workspace selected.

[38:58](#) I'll open the Version Changes viewer, and I can compare with the default version...

[39:02](#) ...because that's the actual version I want to merge with right now...

[39:05](#) ...but I've kind of forgotten all the different things I've edited within this, this QA version.

[39:11](#) So selecting OK will compare with the default version and show me all the different things...

[39:16](#) ...that have changed so you could see here.

[39:18](#) I have an insert, an update, and a delete that I just did in Map.

[39:22](#) And if I were to open the Change display, I could actually see what these look like.

[39:26](#) So I could see what these look like in the QA version and, of course...

[39:29](#) ...it's not even going to exist in the default version right now.

[39:33](#) Also, updates, you can see, this is the update I made in the QA version...

[39:37](#) ...but it also looks like it's been updated in the default version...

39:41 ...so I'm guessing this might be a conflict when I go to reconcile.

39:45 And then finally I delete.

39:46 You can see it doesn't exist in the QA version, it exists in the default version.

39:49 So the Version Changes viewer is really just a...

39:51 ...a very quick way to get all the changes in a version and compare those with another version.

39:57 So I'm done with my edits, I now want to just merge these in the default version.

40:01 So the first thing I do is click the Reconcile tool, which reconciles with the default version.

40:07 I'll just accept the, the default properties which really help me define what I consider to be a conflict...

40:15 ...and how it de, gets resolved by default.

40:17 So selecting OK will now pull these changes from the default version into the QA version...

40:24 ...and look for any conflicts.

40:25 And then it, it found one, so it prompts me to go into the Conflict Resolution dialog.

40:30 And if I pop open that Conflict Resolution dialog, I can see that I have one conflict and, shockingly...

40:38 ... it's that feature that I edited or someone edited in the default version and I edited but, in the QA version, but...

40:45 ...you can see here that they've been modified in different parts of the geometry.

40:49 So I can use that new functionality, which I explained a little bit earlier to merge these geometries.

40:55 So both these changes are actually valid.

40:57 So selecting Merge Geometries allows me to take both solutions or take both edits...

41:02 ...if I was to close the Conflict Resolution dialog...

41:05 ...you can see here, I get the changes both from the default version and also from the ver...

41:09 ...the changes I made in the QA version.

41:12 So I've gone ahead and I've resolved all my conflicts.

41:14 The next thing to do is post with the default version.

41:18 So I've posted with the default version.

41:19 Now I use the, let me just stop my edit session here...

41:25 ...and I can use the Change Version dialog to change to the default version, see what that

version looked like.

41:31 And since I've posted to the default version, it's going to look exactly the same as the QA version.

41:37 So that just really quickly shows how you might edit a version and do simple edits on a version...

41:42 ...and some of the tools we've provided at 9.3 and earlier releases to make those workflows very quick and efficient.

41:59 So let me quickly talk about the, the other types of versioned editing we added.

42:03 At 9.2., we added the Move to Base option, so...

42:06 ...so when you're editing a version or deciding to register a feature classes version, you can...

42:13 ...you can choose to register it with the Move to Base option enabled.

42:16 And what, what really this is, is it's versioned editing...

42:19 ...with the ability to push any changes that are made to the default version directly into the base tables.

42:25 So typically, any edits go into the A and D tables, and those aren't available in the base table...

42:31 ...the actual database table until you do a thing called compress.

42:34 And compress takes all those edits from the default version and pushes those down into the, the base table...

42:41 ...so all other applications can see those changes.

42:44 And we really wanted to promote versioned editing...

42:47 ...but we also didn't want to lock up all the edits in the Delta tables until they compressed.

42:51 So we added the Move to Base option to register a feature class...

42:55 ...and then any edits made in default will go directly into the base tables.

42:58 And this is really designed to better integrate with other IT applications.

43:04 So such as third-party applications that might be directly accessing those feature class base tables...

43:09 ...so those applications have no knowledge of versioning...

43:11 ...they have no knowledge of A and D tables.

43:14 They're just hitting that Parcels table in the de, database.

43:17 So we really wanted to provide a way to get those edits directly into the database tables as

quickly as possible.

[43:23](#) And that really is what the Move to Base option provides.

[43:27](#) Some caveats to this is that it only works on simple data.

[43:30](#) So if you have points, lines, polygons...

[43:32](#) ...even annotation, some relationship classes...

[43:36](#) ...those will all work, and you can register these with the Move to Base option enabled.

[43:40](#) But it won't work with some of the more complex data models such as topology and geometric networks.

[43:46](#) So just something to keep in mind, depending on which type of data structures you have in your geodatabase.

[43:53](#) So why would you ever want to use this?

[43:56](#) Well, of course, if you have third-party applications which are accessing those database tables directly...

[44:00](#) ...you might want to use the Move to Base option to get those versioned edits into the base tables quicker...

[44:05](#) ...so you don't have to rely on doing a compress every time you want to merge those changes in.

[44:09](#) Another reason you might want to use the Move to Base option is...

[44:12](#) ...if you have a large number of database constraints on the actual database tables.

[44:16](#) So if you have check constraints, if you have not null constraints...

[44:18](#) ...unique constraints on the database table.

[44:21](#) These constraints will fire when we push or do the edits to the base tables.

[44:26](#) So if you want to control an editing experience...

[44:28](#) ...where the editors are actually seeing these database constraints bubbling up...

[44:33](#) ...and they can do something about those when they're editing...

[44:36](#) ...then the Move to Base option is a very good solution.

[44:42](#) The last type of editing and maybe the easiest type of editing or simplest type of editing...

[44:46](#) ...is nonversioned editing. It's something we added at 9.2.

[44:49](#) And it really is just editing to data that hasn't been registered as versioned yet.

[44:54](#) And all you're doing is you're editing the database tables directly here.

44:58 So you're not editing in a version at all.

45:00 You're not using versioning to edit this information.

45:03 You're just directly using the database's short transaction model to interact with that information.

45:09 So this is again designed for IT integration because you're directly working with those database tables...

45:14 ...that maybe some other third-party applications are also accessing, reading from, writing to...

45:19 ...and this also is only supported on simple data, so points, lines, polygons, things like that.

45:27 The last type of editing is editing directly through SQL...

45:29 ...and at 10, we've taken great strides to open up the geodatabase to SQL developers, but...

45:35 ...so SQL can be used to directly update geodatabase information and also feature class...

45:41 ...and table information through database applications such as SQL Plus or...

45:45 ...or any other method you use to execute SQL against the DBMS.

45:50 Also since the advent of spatial types in the database, if you store your spatial information...

45:55 ...in a spatial type in the database, you can actually update the geometries using spatial functions...

46:00 ...native spatial functions for that spatial type in the DBMS.

46:04 So really a spatial type is just a data type like a, like a text field or a integer field.

46:11 It's just a geometry field that stores spatial data natively in the geodatabase.

46:16 And they're really useful because...

46:19 ...you don't need to use Esri clients to edit spatial information with spatial types.

46:23 You can do that directly through the database using what's called constructors...

46:26 ...and functions in the DBMS platform that you choose.

46:31 This also provides SQL access to geometries, so...before, we always hid the geometry away...

46:37 ...in a big blob and it was really hard to access through SQL.

46:39 But now it's open, so you can get all the coordinate information of a se...

46:44 ...geometry directly through SQL using a spatial type.

46:46 And all the databases that we support have their own native implementation of the spatial types...

[46:52](#) ...and for some of the DBMSs that we support, we also support SDE geometry...

[46:57](#) ...Esri spatial type, which is fully open. It can be used directly at the database level.

[47:03](#) So, quick summary here. There's three ways to edit data in ArcGIS...

[47:07](#) ...versioned editing, using long transactions...

[47:10](#) ...nonversioned editing, short transactions...

[47:12](#) ...and editing through SQL, using the short transaction model.

[47:16](#) And which one do you use? It's always the big question.

[47:20](#) It really depends on the behavior you desire.

[47:22](#) Do you have long edit sessions? You're editing a lot of data over a long period of time.

[47:26](#) You might want to use versioning.

[47:28](#) If not, short transactions are a good model.

[47:30](#) Also, is the data being accessed by third-party applications?

[47:34](#) If it is, you might want to use the Move to Base option or use nonversioned editing...

[47:38](#) ...to get those edits directly into the base table as quickly as possible.

[47:41](#) And also, how many editors are editing the same data at the same time?

[47:44](#) If a lot of editors are editing or need to edit that data at the same time...

[47:47](#) ...versioning and long transactions is really the best solution.

[47:51](#) All right. A couple quick slides on archiving.

[47:55](#) Try not to rush through these...

[47:57](#) ...but try to give Gary a little bit of time to show all the cool stuff in replication.

[48:00](#) So archiving is something we added at 9.2.

[48:04](#) And it's really just a way to archive the history of any edits made to the default version...

[48:09](#) ...in a versioned geodatabase.

[48:11](#) So really, we maintain a record of feature classes over time and we time stamp any of those edits.

[48:16](#) And then you can perform queries against this information...

[48:21](#) ...so you can see what a feature class looked like 10 days ago.

[48:24](#) You can see what a feature class looked like at the end of 2007.

48:28 And this is built on top of versioning so it extends the versioning model...

48:33 ...so a feature class has to be registered as versioned before it can be enabled archiving.

48:37 So since it's registered as versioned, it's going to have those A, A and D tables...

48:41 ...and when you enable archiving on a feature class...

48:44 ...we actually take that base table and all the A and D edits for the default version...

48:50 ...and we create a new table, a new nonversioned table...

48:54 ...in the database with all those changes.

48:56 So it's going to be a complete copy of the feature class...

48:58 ...and we're going to add some columns to that.

49:00 :So we're going to add a From and To date, which we use to track dates...

49:03 ...to track time stamps.

49:05 So that's what happens when you enable archiving.

49:07 You create this archive table in your DBMS.

49:10 And then as you make edits to the default version...

49:13 ...really the editing workflow doesn't change.

49:15 You make changes to the default version.

49:17 Those changes go into the A and D tables...

49:19 ...depending on what changes you make.

49:21 And then when you save your edits, we cha... we take those edits...

49:24 ...and we actually append them to the Archive table.

49:27 We insert them into the Archive table ...

49:29 ...and time stamp them with the time that you've made those changes.

49:33 So this allows you to work with archived data and see historical representations of that information...

49:39 ...and you can do so in two ways in ArcGIS.

49:41 You can do so by specifying a specific date.

49:45 Here I'm specifying a specific date from a date picker.

49:48 So I want to see how that version looked at that specific time...

49:52 ...while there's also a concept of historical versions or historical markers...

49:55 ...which are just really named moments in time.

49:56 So you can go ahead and create a bunch of historical markers...

49:58 ...so you don't have to remember all those dates...

50:00 ...and those historical markers might be significant to your organization.

50:03 And then you can just connect to those by name.

50:07 So now I'll throw it over to Gary and he'll, he'll give a quick demo and a couple slides on replication.

50:22 Okay, so what is geodatabase replication?

50:25 Can you guys hear me back there okay?

50:27 Yeah, okay.

50:28 Geodatabase replication allows you to distribute data across two or more geodatabases.

50:33 So you can take your, your data and have it in multiple locations.

50:38 And what replication will do, is it'll keep those geodatabases in sync.

50:42 I can independently make changes to all of those geodatabases...

50:45 ...and then run processes to sync them all together.

50:50 This next slide shows some typical use cases...

50:53 ...where we've found people have been using geodatabase replication.

50:57 Upper left, we have this idea of a central hub...

51:00 ...we have multiple offices in different locations...

51:02 ...and you want to keep them all in sync.

51:04 A one way to do it is to create a central geodatabase that replicates out to all of those offices.

51:10 Then when any one office makes a change...

51:12 ...you can synchronize to the central hub...

51:14 ...and then have those changes pushed from the hub down to the target office.

51:19 On the upper right, we have mobile users, people in the field.

51:23 Maybe your field-workers need to have a geodatabase on their laptops in the field to do their work...

51:35 Well, you can do that with geodatabase replication.

[51:37](#) You can create a replica of a subset of your central office geodatabase out to a...

[51:43](#) ...geodatabase on the laptop...

[51:44](#) ...have the field-workers make edits, and then come in at the end of the day...

[51:48](#) ...plug in and synchronize, post those edits back up to the central database...

[51:51](#) ...and then go out the next day and do it again.

[51:54](#) The lower left, we have multiple levels.

[51:57](#) Maybe you have federal, state, and city offices.

[52:02](#) And you want to keep them all in sync.

[52:04](#) Each one of them has their own data...

[52:06](#) ...but they also have the data that is being edited at the level below.

[52:10](#) So for example, if I have a city office and I make all my edits for that city...

[52:14](#) ...I can post those changes up to the state-level office...

[52:18](#) ...and if every city does that, then the state has all the data for all the cities in the state...

[52:22](#) ...and you can run analysis across all of it.

[52:25](#) Then I can take it a step further and I can take all those state-level data...

[52:29](#) ...and post it up to a federal view geodatabase...

[52:31](#) ...and then I can run countrywide analysis.

[52:37](#) The lower right shows production publication.

[52:39](#) That's a case where I have editors editing a production geodatabase...

[52:43](#) ...and have separate people accessing a read-only publication geodatabase.

[52:48](#) This allows me to tune each geodatabase for the type of user...

[52:52](#) ...and also give some independence so that I can take down the production geodatabase...

[52:56](#) ...for maintenance without affecting others that are reading the geodatabase.

[53:02](#) Okay, so how does replication work?

[53:04](#) Well, you start off with data in a source geodatabase.

[53:06](#) And when you call a Create Replica...

[53:08](#) ...it copies that data down to a target geodatabase.

[53:11](#) That process also registers on those geodatabases a replica.

53:15 A replica is, has metadata.

53:17 In this case, the replica is called Replica A.

53:21 The source geodatabase, the replica gets labeled as a parent.

53:25 On the, on the target geodatabase, it gets labeled as a child.

53:28 Together, they make up a replica pair, and once that's established...

53:32 ...you can connect to them later and do synchronizations.

53:35 So I can make edits at the parent or at the child, and then I can synchronize from child to parent...

53:39 ...or parent to child, and then do that continuously.

53:43 Now when you create a replica, you base it on a specific version.

53:47 So rep, geodatabase replication only replicates versions...

53:51 ...and you only replicate a specific version per replica.

53:59 You can just pick the specific datasets you want to replicate.

54:03 And then within those datasets, you can use filters to define what features you want to replicate.

54:08 Maybe I only want to replicate the southwest corner of my data.

54:13 We have three different types of replication.

54:15 We have checkout/check-in, where you start off with data on the parent.

54:18 You check it out, make edits on the child, and then check that in.

54:22 And then once that's done, the process is over.

54:25 To do more edits, you have to create a new checkout.

54:28 We have one-way replication, where you create a replica from parent to child...

54:33 ...make edits on the parent, and then sync multiple times to the child.

54:37 Now at 10, we've added a new option, which is one-way child-to-parent replication...

54:41 ...where you can make edits on the child and send back to the parent multiple times.

54:45 So you can choose at creation time which direction you want to send changes.

54:49 And the final type is two-way replication.

54:51 With two-way, you can make changes at either the parent or the child...

54:54 ...and sync from one to the other multiple times.

55:01 So a little more information on the types of replicas.

55:04 The source geodatabase must always be ArcSDE, no matter what of replicating you're using.

55:09 The target geodatabase can be, can also be ArcSDE for all types.

55:14 The target geodatabase can be a file or personal, as well, for one-way and checkout/check-ins...

55:19 ...but not for two-way.

55:21 And then one-way replicas at 10, like I mentioned, we have a one-way...

55:24 ...child-to-parent replication option that's new...

55:26 ...and we also have an option where you can use archiving instead of versioning to track changes.

55:31 That reduces the footprint of replication on your system.

55:37 We support connected or disconnected environments.

55:41 Connected means that you're always on the same network.

55:43 You connect to the geodatabases and you call synchronize.

55:46 Disconnected means you're, you're not physically on the same network...

55:50 ...and we allow you to do synchronizations and creations by exporting, transfer...

55:55 ...transporting a file, and importing.

55:58 So all workflows are supported in both environments.

56:00 Just in disconnected, you need to do a few extra steps.

56:03 We have desktop commands, GP tools, and an SDK.

56:07 Our synchronization is based on message exchange...

56:09 ...and we send messages back and forth when we sync.

56:12 And it's also fault tolerant, so if the databases were to go down...

56:15 ...or an error were to happen, we roll back to the previous state.

56:19 So all you need to do if there's an error is run the synchronize again...

56:22 ...and it'll get you back in shape.

56:26 We can also apply schema changes across replicas.

56:28 So I can add a field on one replica geodatabase...

56:32 ...and then we have some tools to move that field across all of my other replicas.

[56:39](#) We support LAN and WAN environments, so...

[56:43](#) ...geodatabase replication can work within my office network...

[56:48](#) ...or it can work over the Internet using geodata services in ArcGIS Server.

[56:54](#) A few best practices.

[56:56](#) Always anticipate your future needs when you create a replica.

[57:00](#) Once you've defined the replica, you can't change its definition.

[57:03](#) I can't make my replica area larger.

[57:05](#) So if you know down the road that you're going to be expanding the area you're replicating...

[57:09](#) ...create that extent for the, for the whole area initially.

[57:14](#) Develop a well-defined data model before creating replicas

[57:17](#) We support applying schema changes, but it's extra work you need to do, so...

[57:23](#) ...the, the more well-defined your data model is before you create replicas, the better.

[57:27](#) Automate creation synchronization.

[57:30](#) Good way to do that is to use GP medal... mo, models in the Windows Scheduler.

[57:35](#) Consider replicas when you're managing your ArcSDE geodatabase.

[57:38](#) So just like you managed versioning and versions, you'll also want to consider your, the replicas that are there...

[57:44](#) ...and there, there's a link to a white paper shown here that talks about that...

[57:47](#) ...and I recommend reading that for anyone using replication.

[57:51](#) Use the appropriate technique for the type of network.

[57:54](#) If you have a fast network, then you can use connected synchronization.

[57:57](#) If you have a slower network, you may want to look into geodata services.

[58:00](#) If you have a really, really slow network or no network, go disconnected.

[58:05](#) And then always upgrade to the latest release and the latest service pack.

[58:08](#) If you're going to version 10 and you're planning on upgrading your geodatabase...

[58:12](#) ...and you already have ve... replicas in that, in that 9.3 or pre-10 geodatabase...

[58:17](#) ...there's a patch that we've just released that you, that you, that you should use.

[58:22](#) So that, this is just, that's just for information.

58:26 So what are some of the FAQs, what are the, some of the more common questions we get about replication?

58:31 One of the things people want to do is this reverse one-way workflow...

58:34 ...basically, the new feature we've added in 10 to support child-to-parent replication.

58:39 How you do that in earlier releases is described in our technical articles.

58:43 Sometimes people work with multiple projections.

58:46 They have different coordinate systems in, in their parent and child.

58:49 You can get that to work. There's a couple extra steps you need to do...

58:52 ...and we have a KB article for that.

58:56 One of the main questions we get when people work with geodata services...

58:58 ...with ArcGIS Server is about time-outs and data size limits.

59:02 So there's an article that talks about how to adjust that appropriately for replication.

59:08 We have, we log information whenever we create or synchronize replicas...

59:12 ...and we log it to a file called replicalog.dat in your user's temp directory.

59:17 When you want to look at that file, it, an easier, an easier way to look at it is to look at it through a style sheet.

59:23 We provide a style sheet in a KB article.

59:26 When you use an ArcGIS Server with SQL Express, there's a couple things you need to know about permissions.

59:31 That's also des, that's also worth noting.

59:34 If you want to add a feature class to a replica, you can do it, but there's no out-of-the-box tools.

59:39 You need to do it with the SDK, and we have a sample for that.

59:43 And then, when you have relationship classes based where the ObjectID is the primary key...

59:48 ...it's supported, but there are a couple of caveats you need to know about, and that's also in the doc.

59:55 Okay, so another question we ge...we get often is...

59:56 ...what's the difference between DBMS replication and geodatabase replication?

1:00:01 Well, geodatabase replication is built on top of a geodatabase, so it's DBMS independent.

1:00:06 I can replicate Oracle to SQL Server to Postgres...

1:00:09 ...as long as they're all geodatabases, geodatabase replication will work.

[1:00:14](#) Geodatabase replication also knows how to properly synchronize more complex data structures in the geodatabase...

[1:00:20](#) ...so like networks and topology, it knows how to maintain the integrity of those.

[1:00:25](#) It's version specific, so when you replicate something, you're only replicating one version.

[1:00:29](#) And it's loosely coupled.

[1:00:31](#) I can make independent edits and then synchronize when I want.

[1:00:34](#) The edits I make aren't dependent on the synchronization.

[1:00:37](#) They're completely independent operations.

[1:00:39](#) DBMS replication is replication at the DBMS table level.

[1:00:43](#) It's below the level of the geodatabase.

[1:00:45](#) When you're using DBMS replication tools, they, it doesn't know anything about networks or topologies.

[1:00:51](#) It just knows that there's a set of tables here that you have in your geo...

[1:00:54](#) ...in your database.

[1:00:55](#) So to, to use that effectively, you really need to understand what those tables are and how they work.

[1:01:00](#) The main workflow for DBMS replication is, is useful with geodatabases is for high availability-type applications.

[1:01:07](#) I have a database. It needs to be up 24/7.

[1:01:10](#) I can use DBMS replication to replicate that so that if one goes down, the other one pops up...

[1:01:15](#) ...and then, and you don't miss a beat.

[1:01:19](#) At 10, we've added some things. I've talked about that. I'm just going to review them here.

[1:01:22](#) We have the one-way replication using archiving, which I mentioned...

[1:01:26](#) ...and also one-way child to parent.

[1:01:28](#) We allow you to do schema mapping at 10, so...

[1:01:32](#) ...the names of your feature classes don't need to match.

[1:01:33](#) They can be different, and we can map them.

[1:01:37](#) We have simple checkout and two-way replication.

[1:01:41](#) Simple is, simple enfold a concept that we had it one-way in earlier releases...

1:01:45 ...and we now have it across all of the geodatabases.

1:01:48 You could replicate to a named version on the child at 10.

1:01:52 We used to require it always be default on the child.

1:01:55 And we support nonversioned data for checkout/check-in.

1:01:59 All the other replication types support, require versioning.

1:02:04 One of the targets, goals for 10 was to better support some of the workflows we've been hearing from customers...

1:02:09 ...and one of them is showing in this diagram.

1:02:11 That's the case where I've got a bunch a, of offices, say, in different states...

1:02:15 ...and I want to centralize a lot of that information up to one central geodatabase...

1:02:19 ...so that I can run analysis across all of the states.

1:02:22 So things like schema mapping, one-way child-to-parent, and even replicating a ...

1:02:29 ...a named version on the child all help and make that workflow much smoother for people.

1:02:35 Okay, so I'm going to do a quick demo of replication...

1:02:37 ...just to give you an idea of how it works.

1:02:46 Okay, so in this demo, I'm a field-worker, and on my laptop I have a geodatabase...

1:02:51 ...that I've made some edits to in the field.

1:02:53 I've come back in the office, I've plugged in, now I'm ready to synchronize.

1:02:57 So what I do is I just connect to my geodatabase, I right-click...

1:03:01 ...I see Distribute Geodatabase Synchronize Changes.

1:03:03 Now before I run that command, let's take a look at what I'm about to synchronize.

1:03:08 Here's the enterprise geodatabase. This is the main geodatabase in the office.

1:03:13 Take a look down here and notice that there's some buildings digitized here.

1:03:16 These have been digitized after I created the replica.

1:03:19 So if I look at my field geodatabase, those building don't exist.

1:03:24 However, in the field, I made some edits.

1:03:26 I added some underground service lines to these two buildings.

1:03:30 So I want to take those service lines and apply them to my enterprise geodatabase and at the same time...

1:03:38 ...I want to pull down those buildings to my field laptop so I can unplug, go back in the field, and do some more work.

1:03:41 So let's kick off that synchronization process.

1:03:46 I call Synchronize Changes, I pick my re... my replica, which is Rep Demo...

1:03:50 ...I'm going to go in both directions and hit Next and I'm going to hit Finish.

1:03:55 So this kicks off the process.

1:03:57 So this is actually doing two synchronizations at once.

1:04:00 It's collecting all of the changes on my field laptop and sending them to the parent...

1:04:05 ...and integrating them into the parent on the enterprise...

1:04:09 ...and then it's collecting all the changes that have been made in the enterprise...

1:04:12 ...and sending them back into the field geodatabase.

1:04:16 So with that done, let's take a look at what happened.

1:04:19 If I call Refresh, you'll see the buildings have come into my field geodatabase.

1:04:25 And if I go to the enterprise and I call Refresh, you'll see that those...

1:04:31 ...those two service lines have been added

1:04:33 Now I, I have a, a, I didn't set this up correctly.

1:04:35 The old one should have been deleted as well.

1:04:38 But, that was just, that was my fault in the setup.

1:04:41 But you'll see that the, the secondaries have come in, as well.

1:04:45 So now what I can do is I can unplug my laptop, take it out in the field, and do some more work.

1:04:51 So that's just a taste of what you can do with replication.

1:04:53 Tomorrow at 10:15, we have a full replication session, if you want to learn more...

1:04:57 ...where we go into lots of different details.

1:05:07 So that's pretty much the, the end of our presentation.

1:05:10 So thanks very much for attending.

1:05:12 We really appreciate all the support.

1:05:14 And if you could, have any questions, we can answer them up front right now...

1:05:17 ...or you could come up front a little bit later.

1:05:20 There should be some session surveys on your seats.

1:05:22 We really appreciate any feedback we get on these presentations.

1:05:25 We're always trying to improve

© Esri 2013 <http://www.esri.com>