**Level:** All
**Works with:** Sametime 3.0
**Updated:** 03-Sep-2002

# Same place, Sametime with Chris Price

Interview by
Tara Hall

*Chris Price is a software architect with the Sametime team whose primary focus is the server-side architecture and specifically, for this release, the Enterprise Meeting Server (EMS). We spoke with Chris, who's located in Lexington, Kentucky, about EMS.*

**Can you tell me about what the customers can expect with the Sametime 3.0 server?**
There are two releases in Sametime 3.0: there's a standalone server, and there's what we call the Enterprise Meeting Server, which is an add-on product. In the standalone server, we have a few new tools for Meeting Services. We added a dynamic whiteboard file attachment feature, so you can add content to your active meeting.

We also enhanced our accessibility in release 3.0 to meet ADA requirements. We have screen-reader support in all the HTML pages where you schedule meetings, and keyboard accessibility is enhanced on the client.

Scheduled alerts tell you when certain people come online now. If you're waiting for someone to come in the office, you can now be buzzed when they come in. There are also some mobile features that extend Sametime Everyplace to Sametime Connect. We've got more work to do there. We are setting up more profiles, and so far, it integrates nicely with the 3.0 client, so that you can see when someone's online on a mobile device. They have a different icon above their name.

On the server, we enhanced the manageability and administration capability. For instance, we have additional reports generated when users attend meetings, and the reports tell you which tools are used in meetings. We also enhanced our diagnostics and client connectivity. In the past, we had problems with clients connecting and staying connected during meetings. So we made some enhancements in that area for diagnostic purposes. We have new support for the single sign-on authentication model from Domino. We're using LTPA tokens for single sign-on between multiple servers within your domain. Also new is the HTTP polling connection method. We still support HTTP tunnelling, but polling helps keep more clients connected when they use proxies that sever connections.

We also have support for some new cameras that are coming out. They're a couple of 360-degree cameras available right now that can take a fish-eye view of all the faces around a conference table. You can see a wide-angled view of all the people in a conference room.

We also made some improvements to support high bandwidth video. Some customers want to improve the video in a meeting, so we added support for turning up the video bandwidth and getting high quality videos at a meeting.

And Sametime 3.0 supports sending a file through the Sametime Connect client now. You can send files to someone on your buddy list.

**How does this send file feature work? How is it that you can deliver a file so quickly through an instant**

**messaging client?**
An Instant Message is a virtual channel between two parties. The chat that you type is nothing more than data that is being sent between two parties on this special channel. So, to send a file, we invoke the message type on the special channel, which carries binary data. That channel carries along with it the metadata, like the file name.

We built in some administrative controls. Sending files is like opening yourself up for a denial of service in which people can grab huge files and send them. There's an administrative level control that allows you to send files only of a certain size. Sametime can weed out certain file types too, so you can't send sick or bad files. We don't provide any virus protection, but customers are going to ask us to do that at some point.

**Does sending files through Sametime require a whole new protocol or do you use your existing protocols?**
It's essentially the existing protocols, but a different message type. It is just an extension of the message type.

**Earlier you mentioned Sametime Everyplace. What plans do you have for future integration?**
To integrate with Sametime Everyplace, we need the ability to allow users to set their preferences more specifically. For instance, imagine that you'd like to set a server profile that says normally within the hours of 8 AM and 10 AM I want you to log me in, but, log me in mobilely because that's usually when I'm commuting to work. I know that my colleagues who need to get in touch with me—because they're already in the office by 8—can reach me while I am in the car on the way to work. I want my name to be green (active), so that if they need to send me a quick message, they can do so. Maybe set my status to "I'm on the road." To integrate better with Sametime Everyplace, we must add both types of features to have it set your profile up, instead of just the notion of I'm either online on my desk or I'm not at my desk.

**Howard Berman, the product manager, tells me that you are the EMS expert. So, can you tell me what is EMS?**
The Enterprise Meeting Server is a centralized model for delivery of meeting services in an enterprise ready fashion. What we mean by "enterprise ready" is that from a manageability standpoint, there's a single place to go to look at what's been going on in an environment across many servers. From a manageability or a serviceability standpoint, it's failover ready, so if a server that's hosting a meeting goes down, the system can schedule that meeting and set that meeting up on a different server that is available. We say enterprise ready because it has the attractive features that large corporations, large enterprises, want to see.

Typically with the standalone Sametime server, if you want to look at log information, for instance, you have to go to each server to look at the log. With EMS, we centrally store all that information using IBM enterprise tools, like WebSphere as the front-end to the system. As a back-end datastore, we use DB2. And we use MQSeries for workflow between the components in the system. EMS is essentially a front-end on top of a group of Sametime standalone servers. There is an administrative feature that allows a standalone server to be added to the group of servers that EMS provides access to. So the Sametime 3.0 server essentially becomes managed when you add it to this group. All the administrative functions now belong to, and are accessed by, the Enterprise Meeting Server. You don't go to the standalone server to schedule meetings and to attend meetings any longer. You go to the Enterprise Meeting Server. It directs clients to the servers where meetings are being hosted.

It's a very centralized delivery model, very much like what IBM does with their current e-meetings offering. We work with the e-meetings group and took what they had done with the concept of the room server and the managed server and productized that. We want to offer to other large corporations with similar deployments as IBM the facilities and the features to do that kind of thing themselves.

**With EMS, do you get failover in a video meeting? And do you get failover in a meeting where there's a whiteboard, or screen sharing, and presence?**
Yes, all of the above with EMS. If a server fails while you are at a meeting, basically the meeting is rescheduled on a different server. Right now, we don't do a good job of transitioning the state to pick up where you were in a meeting, but the clients get reconnected. And if users had hit, for instance, the Save button in whiteboards, we save the content. Then, that saved content is available for them in the rescheduled meeting. If they've gone through and made a lot of annotations, and they don't get saved, then we cannot extract that content if a server crashes. But the idea is to keep meetings available if something catastrophic happens on a server. That's the notion of failover. If our telephone call were to fail right now because a switch some place died, we can still pick up the phone and call each other. Our call might be routed to a different switch, but our current phone call is lost. We have to re-establish the connection and continue where we left off.

As far as community and awareness failover go, a clustering feature exists in Sametime 2.5 that the Rehovot team had done a lot of work on to improve. So, what you have is a subset of your managed servers designated as community cluster servers. The servers that are in a subset of those named servers are considered to be part of that community cluster, and if one of those servers goes down, then the user's Sametime Connect client automatically reconnects to the next server in the cluster.

**This release of Sametime requires Domino. Does EMS require Domino?**
For each room server, you must have Domino installed. The Enterprise Meeting Server is a non-Domino application. It's a J2EE WebSphere application. We require Domino on the room servers, but not on the EMS.

**If you don't require Domino, how do you cluster the servers?**
The meeting clustering support is not dependent on Domino. Clustering for those servers is based on clustering logic, or a failover logic, and it's native to the EMS application. The logic that runs on the EMS server is logic that determines when a meeting server fails and how the meeting gets re-routed to another server. We do use some Domino clustering from a community services standpoint, but not for EMS.

**You mentioned the technologies that are going into EMS, like J2EE. Can you tell me more about how those technologies are being used?**
When we started, we knew that Lotus had a focus now to deliver applications and to deploy in the WebSphere environment. So we wanted to use a standards-based mechanism for designing EMS. Obviously, J2EE is the way to do things now. We're pretty much using standards in terms of all the JSPs. Any of the back-end, like databases and messaging services—JMS (Java Message Service) that we use—follow standards. When you purchase EMS, you get a license to run WebSphere, DB2, and MQSeries. We designed J2EE applications so that if a customer has an existing investment in, say, Oracle, we provide at least the SQL schema in such a way that the database administrator can install that schema into a datastore and configure the schema to use the Oracle data engine, instead of DB2.

We try to design things with J2EE to allow people who have existing investments to better use existing technologies. We're IBM, so we're building these things, testing them, and running them against all the IBM products in those spaces as well. We chose a model that separates presentation from application logic. There's an additional Web application that runs in WebSphere that manages things, that collects information from the room servers for load balancing, and that schedules meetings also. They're all J2EE-based standard applications that are packaged together in an enterprise application that gets deployed on a WebSphere server.

This model is different from the Domino NSF model. But I think J2EE is a technology that much of the development community is becoming aware of and becoming more familiar with. It provides a very flexible environment. We're trying to keep our presentation layer as simple as possible so that customers, as well as developers, can easily customize. We've got cascading style sheets in EMS so you can do simple things like change text color. We're keeping Java-code essentially out of the JSP pages. And we're focusing all that Java code into JSP tag libraries that can be re-used. If you have developers who want to rewrite HTML pages, they can reorganize the page and reference the tag libraries that we're providing without having to copy and paste a bunch of Java code from one page to the other.

**Do you now have, or do you have plans for, an API toolkit for customers to extend EMS?**
We don't provide a toolkit in this release, but tools are available to extract our Web application to get the framework and the HTML. Then you can start changing things.

**EMS is an application and an administrative tool. Is that correct?**
It's both an administrative tool and a user front-end. Users see the Sametime interface as presented by what we call the Sametime Web application. There's another, separate Web application called the Sametime Administrator. The admin features include some canned reports that show administrators which types of tools people are using in meetings as well as how many people logged in, what the peak log-in time was, and how many users were there at that time. There are a couple of areas of EMS that are user-facing and some that are administrative-facing. If you compare the out-of-the-box UI of a Sametime standalone server with the UI of Sametime EMS, they're very similar. But there are some additional search and organization features that you see inside of EMS.

You can imagine that if you're deploying at an enterprise level there are probably a lot of meetings going on in the environment. It becomes important for users to find the meetings that they're interested in. We have some integrated search bars on the EMS user interface. There's also an integrated calendar picker on most of the views, so you can click on a day to see the meetings on that day. Those views are searchable with the search bar. You can click on a day's meeting and perform a full-text search.

**EMS includes different components. What are they?**
With J2EE, there are two types of application resource bundles: EARs and WARs. A WAR is a Web Application Resource bundle that may contain HTML, JSPs, and Java code. WARs also describe the associated role mappings for allowing access control and resource mappings to assign which back-end resources should be available to the Java code. An EAR is an Enterprise Application Resource bundle that contains one or more WARs as well as the role definitions for this set of WARs.

EMS is packaged as a single EAR that contains three WARs. The Sametime UI WAR contains the user-facing components required to schedule and attend meetings. The Sametime Admin WAR contains the administrator-facing components required to view the log and report data and to configure aspects of Sametime features. The Sametime Server WAR contains the back-end meeting server components that handle the clustering and failover of the group of managed Sametime servers.

The back-end meeting server components consist of the scheduler, logger, load balancer, and health monitor. The scheduler manages starting and stopping meetings. The logger collects log information from the managed servers. The load balancer collects statistics from the managed servers to determine their current load. The health monitor is checking the managed room servers to make sure they are still functional, and when it detects nonfunctioning servers, it alerts the other components.

**With EMS, am I limited to a cluster of 3.0 servers only or can I have a mixed-release cluster?**
At this time, the cluster must contain Sametime 3.0 releases. Of course, as we move forward, we want to support hybrid environments in which you have both 3.0 and say 3.5 or 4.0—whatever the next release is. But at this point, everything has to be 3.0.

EMS is an offset release. I'm not sure how long it will be before we release EMS, but there will be a service pack that you install on top of a Sametime 3.0 server for anything that we find between the release of Sametime 3.0 Gold and the release of EMS to update the Sametime server to a managed server. We should say that it requires Sametime 3.0 + service pack one.

**You mentioned WebSphere. Is it required to run EMS?**
WebSphere is required for the EMS Web application. What I should say is that WebSphere is the J2EE container

that we test with. This is a J2EE application, so by definition, it is a standards-based application. Theoretically, it can run on other Web application servers.

**So, if I have a Tomcat server, I could run it on that?**
Yes. In fact, some of our developers have been doing their testing on that server because Tomcat is a smaller environment. It requires fewer machine resources. Maybe it's not as enterprise-ready as WebSphere, but it's quicker to get up and running. Often times, developers will choose that server because it is just easier to get going.

**You mentioned single sign-on earlier. How does this work in EMS?**
We require the LTPA token for single sign-on. If you have a third-party sign-on provider, like a Netegrity Siteminder or IBM Tivoli Identity Manager, with available plug-ins for WebSphere, the provider can generate the LTPA tokens. We're requiring LDAP to connect to a directory. So, your user information—users lists, but not necessarily your passwords—must exist in an LDAP-compliant directory. Then, the Domino server can accept that LTPA token when users go to a room server to attend a meeting. The Domino room server accepts the LTPA token when users log into our community services and meeting services. So, they're using LTPA as our single sign-on.

**Were there any directory changes made to support EMS?**
EMS will not support multiple LDAP directories. In Domino, you could use directory assistance to point to different LDAP directories. For community services on a standalone Sametime server, you can set up multiple LDAP directories, including the WebSphere directory. But WebSphere only supports a single LDAP server. So, we had to relax our multiple directories feature. Now we support a single LDAP directory, but it doesn't necessarily limit you. One of the big features of EMS—or one of the big things that EMS is going to do—is fit the application-hosting model in which a single service provider can provide meetings services to a bunch of different organizations.

We like to use the Coke and Pepsi example. You can service both Coke and Pepsi on one site, and employees from both companies will not necessarily know that the other company is using that same service. At the directory level, we're segregating our directories by organizational unit. We're still looking at one LDAP server, but because of the tree-based, hierarchical nature of LDAP, you can separate things out on this particular Web application.

Company A looks at the tree that has Company A in the base DN (distinguished name). So, that's how we're doing things. In-house we're testing with the Netscape, iPlanet, IBM SecureWay, and Domino as an LDAP directory.

**You're testing the same directories that QuickPlace is using, which may not be a coincidence because Sametime is becoming more integrated with QuickPlace.**
We're working more with QuickPlace team. In fact, in Purple—the code name for our next release—you're going to see a lot of integration there at least at a components and tools level. I'm not sure product-wise what the plans are, but we'll be working more and more with the QuickPlace team. In this release, we've been doing a lot of testing in the lab where QuickPlace has been using a couple of our APIs, like the meeting management API to schedule meetings and attend meetings. So, we're working more and more with the QuickPlace team on a regular basis.

**What benefits does Sametime receive from integrating with QuickPlace?**
One of the things that we like about QuickPlace is its content management. If you've used the meeting server to schedule a meeting, you can upload your file or your presentation. Then users can click on a file to download it, but you have to go someplace else to upload that file. You had to get away from the scheduled meeting to do it. It would be really nice to start a meeting from a QuickPlace and to have access to the materials that are in that QuickPlace inside your meetings.

QuickPlace has some great content creation tools. We'd like to take advantage of those kinds of tools in a meeting environment. It's more of a marrying of the synchronous and asynchronous and accessing some of the rich content in starting off our meeting.

**Do you think there's ever going to be a time when people schedule and host a meeting in a QuickPlace?**
Yes, I think so. It's really up in the air right now the direction we want to go with this stuff because we don't want to force ourselves into any sort of model. Is it just going to be that way? Or, will QuickPlace wrap around a meeting in some way? We're in the planning stages right now, so we're not sure how it is going to be. But starting a meeting from a QuickPlace is the type of thing that you're going to see. And, we'll integrate features like the Domino 6 Calendaring and Scheduling feature that lets you schedule a meeting from your calendar.

**Do you have any EMS deployment tips for customers?**

I participated in the Sametime and QuickPlace Customer Council, and I've heard from a lot of customers that EMS is a shift. It's a change in the deployment models that we've been pushing with the previous releases. So, it will be interesting to see how this first release is received by the companies that currently use the previous releases because the current model is a little bit different.

Again, we have aligned with this model of centralized service delivery. The out-of-the-box features of EMS should be considered network geographically equivalent. So server A and server B pretty much have the same network connectivity to the clients. However, we have the options of some additional programmatic interfaces to the system. That's what we call a load policy interface that can be augmented to allow customers that have disperse environments to align meetings with remote servers. For instance, if you've customized your meeting center, you can ask users if they want meetings to be scheduled in Europe or Asia or the United States because you have the infrastructure in all three of those locations. A user can select a checkbox to schedule the meeting in Asia, which becomes part of the meeting data; then when the custom load policy starts up, it schedules the meeting on the server in Asia. Then the meeting center would pick a server in Asia to start that meeting on.

Everyone's environment is going to be different. We're providing the facility to write policies for complex and arbitrary environments. Consider EMS as a centralized server model out-of-the-box. There are efforts under way to work in some more of this location-based functionality in the product, probably shortly after we ship release 3.0. We want to provide customers with as much flexibility as they can take. We'll do the best we can to make the functionality that we ship solve most of the problems. But there are always special cases in which you'll want to customize your environment.

**Does EMS replace the Meeting Center?**
The Meeting Center is basically where you go to schedule meetings or where you attend meetings. We've optimized it, or customized it, a little for EMS with the thought that there are a lot more meetings that are available because you're an enterprise environment. We include a search bar, for instance, where you can find meetings that you've scheduled or were invited to. Something that makes it a little easier to see which meetings are available.

In previous releases in which you had multiple standalone servers, the big question was which server is the meeting going to be on? Certainly in Lotus, we've got different servers that we work with. Which server did you put the meeting on? You don't have that model any more with EMS. We just go to the Sametime server, and which room server is actually hosting the meeting doesn't matter to a user. Users always go to one place to attend the meetings.

I suppose that you would say that if you have a standalone Sametime server, it's got a Meeting Center. And now, if you manage it, you're essentially replacing its Meeting Center with the Meeting Center than runs on EMS. So, if you look at it that way, it's a replacement of that feature because when you use EMS to manage the server, you no longer go to the standalone server to manage it.

At another level, all the logging and monitoring of data in previous releases are all NSF-based. That data would go into the ST log database, but in 3.0, data is stored in DB2. DB2 uses a relational model, so the data model has changed. It's no longer NSF. That sort of thing you would say is a replacement feature. As far as the meeting features, there is no replacement of anything there. Basically the Meeting Center now runs on EMS and not on the standalone server. But the actual meeting still takes place on the room server.

As an administrator, you know which capabilities each of your room servers have. Some servers may have more RAM. Some servers may have less. At an administrative level, you cater the resource levels of each of those servers to your liking. For instance, if server A is a big server that can host 200 meetings simultaneously and server B is a small server in the same cluster, maybe you can host only 100 meetings simultaneously. EMS takes those issues into consideration when you schedule meetings. The notion of booking resources is new in EMS.

In previous releases of Sametime, you would schedule meetings, but there wasn't a mechanism for capacity planning or understanding what the future load would be on a server. Now, when you schedule a meeting, you specify how many participants you want—how many you expect in the meeting. We can put those things in a database, and an administrator can use planning tools to see that he needs to add some capacity to this cluster because there is a higher-than-normal meeting demand. Those types of tools are available in an administration back-end to set the level of capacity for these servers.

In addition, with a subset of the managed servers that are your community servers, you want to manage those in such a way that you don't put scheduled meetings on those servers. It's all about how many resources you have. Do you have enough capacity on the machine? Maybe you do. But a lot of people want to separate meeting functionality from other functionality. So they go to the community servers and say I don't want any scheduled meetings on this server. The administration tools in EMS allow you to split services across the machines in your environment.

**You mentioned capacity planning. Do you have any hardware recommendations for people?**
There are some hardware recommendations that are part of our shipping documentation. I think a standalone or managed server has minimum RAM configurations of half a gig. But EMS requires something more because WebSphere is a pretty big application. I would say that you probably want a gig or more of RAM on one of those machines.

You've also got the back-end things to think about too like DB2. You could put all your services on one box. We do it in testing a lot. What we find is that a lot of customers do it for pilot deployments. If you want the best in scalability and reliability, you want to separate these services out. Each of those services, like DB2, has its own requirements to tell you how much RAM and how much disk space one of these boxes should have.

**Do you have any network bandwidth requirements or recommendations for people?**
You want to grow your bandwidth with your meeting usage. You can never have enough bandwidth. Because we centralized service delivery, you have all your servers in one network location. Customers may have machines that work better across different wires. For instance, you may have a machine with two network cards on different sub-nets. If you managed to route your user population to this server with different routes, you have split your bandwidth impact in half. You could also put a few servers on one sub-net and other servers on a different sub-net. By default, a meeting will be active on only one of the servers in your cluster. So, by splitting the servers across different networks, you have reduced the bandwidth impact on each of the separate networks.

I can't really give you a hard and fast answer, like you really need a 100 megabit LAN to run this. Your requirements are going to grow. If you offer more and more services from a central location, you want to support a central location and to make sure that you're delivering services to the client in optimal fashion, which may include using multiple sub-nets with multiple NICS in your servers.

**What's in the future for EMS?**
The future for EMS is going to be pretty interesting. The next release of Sametime is code-named Purple. What you'll see in this release is that EMS and Sametime standalone merge back together. The best way to think of it is that Sametime 3.0 plus EMS was our first attempt at moving towards a J2EE infrastructure. When you think of Sametime Purple, you can think of the complete Sametime Web application running on a WebSphere box. It's up in the air right now. It's possible that there may be two offerings: an enterprise versus a single server edition. Maybe the single server release will have embedded DB2 and a pinned-down version of MQSeries that doesn't have clustering support. But both versions will be built with the same componentry.

You can think of limited licenses or light-weight licenses to run Sametime as a standalone environment. But then, when you want to deploy it and license it for the enterprise environment, it's really the same Web application that's running, but on less restricted versions of WebSphere, DB2, and MQSeries. So, maybe for Sametime Purple, the standalone version of Sametime runs on the WebSphere Single Server Standard Edition. But for the Enterprise version, you use WebSphere Advanced Enterprise Edition. It's sort of fuzzy right now. We're just in the planning stages of what we're going to do for Purple.

**ABOUT CHRIS PRICE**
In 1995, Chris started working at Databeam, the company that developed the neT120 Conference Server that caught the eye of Lotus/IBM. Databeam was acquired in 1998 and began the development of Sametime 1.0 along with the Israeli-based Ubique, Ltd. Chris has a degree in physics from Transylvania University and another degree in electrical engineering from the University of Kentucky. He's been doing software development since high school and considers it as much a hobby as an opportunity for employment. His other interests include home theater/HDTV and gadgetry in general. Chris also likes to race other cars at stop lights and has an extensive collection of traffic tickets to show for it.