Lotus. Developer Domain
The Technical Resource for Lotus software
LDD Today

Level: All
Works with: QuickPlace 3.0
Updated: 01-Jul-2002

Going (Quick)Places with Charlie Hill

Interview by
Tara Hall

In this, the first of a two interviews with the QuickPlace development team, we talk with Charlie Hill, product designer for QuickPlace, to find out what you can expect in the upcoming Release 3.0. Next month, we'll speak with three more members of the QuickPlace development for an in-depth discussion of some of the Release 3.0 features, such as Search Places, PlaceTypes, and QuickPlace directory integration.

QuickPlace 3.0 has been re-architected to include Java and XML. Can you tell me how these components are being used?
In this release, we're introducing the first APIs to QuickPlace. In fact, we're doing some very important internal work on the architecture of the product to create a Java and XML-based object model of all of the objects that you might want to access. Basically, all of the objects that you want to program are exposed as XML entities, though we are not exposing the entire XML object model. We want to prepare the product for more APIs by building a complete XML object model internally. Fundamentally, we want to appeal to the pool of developers who are programming to standards-based technologies, which obviously include Java and XML.

More specifically, we have two goals in this release. One of them is to expose some much needed APIs to integrate QuickPlace with Web portals—for example, the ability to present forms for users to help themselves to new QuickPlaces or to manage the membership of places in a custom Web portal. The APIs enable the portal developer to access QuickPlace for these purposes. Second, system administrators can use the administration commands that we're exposing in 3.0. These commands are exposed as Java APIs, so if administrators want to automate administration functions, such as an expiration policy or a departmental cross-charge for each place created in the creator's department, they can automate those functions. One way to do that is through the Java APIs, which use XML-based parameters. When you write an API call in Java, you actually pass all the data through those APIs in XML that conforms to the QuickPlace object model.

QuickPlace 3.0 is a Domino overlay, so in this release, customers are required to install QuickPlace on top of a Domino server. Why the change?
We're not newly requiring Domino. It's really a packaging change that we're making to 3.0. Previously, you could install QuickPlace stand-alone, which included a canned version of Domino behind the scenes. Or you could buy Domino separately, then install QuickPlace on top of it. In both cases, Domino was required for QuickPlace to run. The real difference is that we've changed the packaging. We no longer provide a stand-alone server. We did that because we really want to expose the full power of Domino administration to all current deployments of QuickPlace, which is important for two reasons. One of them is scalability. If you have a small deployment on one server and you want to move to multiple servers, using the layered install of Domino and QuickPlace provides you with a cleaner way of scaling up that deployment. You can take advantage, for example, of the high availability features using Domino clustering.

Second, in terms of manageability, we want our deployments to take advantage of Domino's powerful administration client. The APIs with XML and so on, are a continuation of QuickPlace's standards-based philosophy, which isn't dependent on Domino. We only use the Java Virtual Machine in Domino for the APIs. We're a consumer of that service.

Is clustering new in this release?
We introduced clustering in QuickPlace 2.0.8 for Windows NT servers only. In 3.0, we support clustering on all server operating systems. QuickPlace includes a special clustering feature that automates the creation and deletion of replication stubs between clustered servers; however, this feature isn't new to 3.0. Our clustering enablement relies on native Domino clustering and replication.

There are factors you have to take into account when installing a QuickPlace server on top of Domino, like performance. How can you ensure that installing a QuickPlace server on top of your Domino server won't degrade performance on your Domino server? Do you have any deployment recommendations?
Generally, in a successful deployment of QuickPlace, you want to dedicate at least one server to QuickPlace. We recommend that you follow the dedicated application server model. In other words, don't mix the scaling characteristics of QuickPlace with the very different scaling characteristics of the Domino mail server or the Sametime server or any other application server. There are a couple of reasons for that. One is that the scaling characteristics are different. For example, QuickPlace is typically CPU limited, so you want to take full advantage of tuning the CPU performance without having to deal with the characteristics of other applications. Second, in many deployments, you reach a point where you want multiple QuickPlace servers. It's much easier to scale your QuickPlace servers separately from other application servers.

What were your intentions for this release and how did you determine what to implement in this release?
We had a number of different intentions that came directly from customers. We have many active deployments, and we've spoken to many customers who've told us what they wanted next in QuickPlace. At the beginning of the release, we divided the design problem up into three audiences: the end users of QuickPlaces, the developers who are building custom solutions on top of QuickPlace or integrating QuickPlace with other solutions like portals and so on, and the system administrators.

We had a number of goals for our end user audience. There is a lot of customer interest in adding real-time collaboration capabilities to QuickPlace. Before 3.0, there was a very modest capability to do what we called a "team chat" using a scaled-down version of the Sametime server built into QuickPlace. In 3.0, we've taken some major steps toward providing a rich range of real-time collaboration functions. First, we removed the scaled-down version of the Sametime server and replaced it with support for connecting your QuickPlace server to an existing Sametime server already deployed in the environment. We also provide two features. The first is awareness and chat, which allows QuickPlace users to see in real time who else is online and to chat with those people. The second is the ability for users to schedule Sametime online meetings on the QuickPlace calendar.

Do users see who is online in Sametime or who is in the QuickPlace?
Many customer with whom I've spoken about QuickPlace 3.0 expect us to show "who is here" in the place. In fact, my initial assumption was to implement the feature that way. However, it became clear during the design process that because users don't spend much time in QuickPlace—and we don't really want them to spend any more time there than they need to—the "who is here" model is very lonely. For much of your time spent in QuickPlace, your team members may be absent. So we focused instead on "who is online" to maximize the availability of team members. I expect that with the availability of the Sametime 3.0 toolkits, customers will add the "who is here" feature to QuickPlace 3.0.

What other end user features did you add to this release?
Related to the Sametime online meetings and QuickPlace calendar, we added another feature: calendar subscriptions. If you use a Notes calendar or a Microsoft Outlook calendar to track your appointments, you can subscribe to QuickPlace calendars to add to your personal calendar all of the events published in any given QuickPlace. You can keep track of those events on your personal calendar.

Another area of improvement for end users is usability. There are a number of usability fixes throughout the product, including a new default theme applied to all new places created. The new default theme is designed to provide better readability and to handle more content. There are a number of other feature enhancements that make it easier for users to find things in QuickPlace.

Search Places lets you search across multiple QuickPlaces in a domain. An enhancement to the existing search function also enables you to limit the scope of your search to a specified room or folder.

Another feature for end users is a room map that allows you to navigate a place with a large hierarchy of rooms. You can open a little window that shows you an outline view of the place, so you can quickly navigate from point A to point B.

Another very important area that also has to do with information access is My Places. This feature provides a listing of places that you're a member of. You can bookmark My Places and open that page at any time to find all the places you're a member of. The Search Places and the My Places features are also exposed as APIs so that developers can present each of them through a Web portal instead of through the built-in user interface, if they prefer that approach.

Users have asked for better and faster user look-up. We are standardizing our support for directories on LDAP. At the same time, we're adding a better user interface for searching for people in a very large enterprise directory to ensure that you can find people even if there are very many similar names in the directory. It also makes it easier to add many names at a time to a place.

We also added support for blind users and users with low-vision impairments by improving access to QuickPlace. We also added a special Accessibility Mode to allow someone who's spending a lot of time in QuickPlace to see an optimized user interface that is screen-reader friendly and high contrast.

And for developers, what changes can they expect?
We're taking two major steps forward for developers. In the area of customization, we've done a major rework of our PlaceType application template architecture, adding the ability to perform a central refresh of all places on a given server. This makes PlaceTypes much more manageable for administrators. It introduces the idea of a central push of updates from the administrator to the end user. Developers also have a more robust refresh mechanism underlying the central refresh feature. Any type of change, whether to content within the place, to the layout of folders and rooms, to design changes—like logic changes to Placebots or changes to forms or to themes—can be robustly refreshed through the new central refresh mechanism. We think that this will enable not only a kind of application upgrade capability, but also a kind of content-push feature. New kinds of applications are possible where libraries of content can be periodically updated for teams who consume that content.

Another area of improvement includes the APIs that I mentioned earlier. There are some new features for portal integration. At a high level, you can provision Places—creating, updating, and deleting Places through a Web portal or, for that matter, through an automated system, as well as control membership of Places through a portal or through another automated system.

An area related to that is listing Places and searching across Places in a Web portal. By presenting this programmability using Java APIs, there is a wide range of "remoting" mechanisms available to the developers. In other words, if you have a portal server that makes requests to a QuickPlace server, there is a range of different ways to make those requests. We didn't want to pre-require WebSphere in this release of QuickPlace. For that reason, we didn't build these services as Web services out-of-the-box. But anyone already committed to Web services can easily wrap the QuickPlace Java APIs as Web services. Because the parameter passing is already XML-based and SOAP-friendly, you can get the benefits of Web services, WSDL, and SOAP to make remote server programming standards-based. If you aren't ready for Web services yet, you can use traditional ad hoc methods, like servlets or any other proven method, for server-to-server communication. And, by the way, referring back to something I said earlier, all of the administrator functions that we added to this release are programmable through exactly the same APIs.

What's new for administrators in this release?
There are some additional, very interesting, developer opportunities for administrators to develop automated systems that track the status of the QuickPlace service. Based on when a place was last accessed, the size of the place, or other parameters that they want to track, administrators can automate policies around those parameters and make calls to other management systems or back to QuickPlace servers to archive places, delete them, and so forth. That's one very important feature that we're introducing spanning both developers and administrators.

We've added a lot of value for administrators in this release. There are three important things that administrators need to know. I already mentioned the first one: the packaging decision to separate the QuickPlace application server from the other layers and services that it depends on, such as Domino and Sametime. And this is, as I described, a scalability and manageability strategy for the product.

The second major new innovation in the product is the Place Catalog, which is a listing or directory of all of the Places that currently reside on a collection of servers that you choose to treat as a unit of service. We're introducing this concept of the QuickPlace Service, which is a collection of servers administered by the same team on the same network, without communication barriers between them.

What are some examples of Services?
For a large scale deployment, we recommend at least two QuickPlace Services—one for the extranet and one for the intranet—because these services use different networks and directories. In addition, some companies may choose to manage their servers on a divisional or regional basis, which may lead them to organize their servers into more than one Service, even within an intranet deployment.

Services can span more than one domain under two conditions: the servers in the Service must be cross-certified with the Place Catalog server, and they must have at least editor access to the Place Catalog database. Servers belong to only one Service at a time.

How do the QuickPlace Service and Place Catalog work together?
Administrators coordinate the servers in the Service through the Place Catalog, which assures that place names are unique between the servers. The catalog also keeps an up-to-date record of all places and servers. The Place Catalog aggregates all of the information about places that administrators need to know to perform their functions. We provide a number of administration tools that work with the Place Catalog, based on a new command-line tool called QPtool. This changes the world for QuickPlace administrators to enable them to see—in an integrated view—a collection of servers that they consider an integrated service that they need to provide to their users.

The Place Catalog sounds like a very scaled-down directory. Is that an accurate description?
Technically, the catalog is not a directory. At one point, we said that we would provide the Place Catalog as an LDAP directory. However, in the process of developing the catalog we discovered that LDAP didn't provide the reporting capabilities that we needed for the Place Catalog. Instead, we implemented it as a Domino application, but we abstracted the service behind an API so that long-term, it doesn't make any difference that it's a Domino application. It's basically a set of views of place data that applies to places on any number of different servers. There are ways of adding custom data to the Place Catalog. For example, if you have a custom field, like cost center, that you want to track for every place, you can add a cost center entry to the Place Catalog. Subsequently, when you run a report against the Place Catalog, you can derive the value of the cost center field in the report. That provides a lot of interesting administration possibilities.

So it's not a tool that you would use to look up a QuickPlace member?
It tracks the membership of each place, but that's membership, not identity. The directory is responsible for centrally defining the identity of people as the basis for authentication, in particular. The Place Catalog manages the membership of places, but that's more of an implementation detail. The important thing to know is that the Place Catalog assures the integrity of places across multiple servers and tracks QuickPlace usage. QuickPlace already manages membership locally without having a Place Catalog.

Earlier you mentioned QPtool. What is it?
In release 2.0.8, the previous release of QuickPlace, we provided a tool called the Admin Utility, a single-server administration tool. In 3.0, we replaced the Admin Utility with a command-line tool called QPtool. QPtool includes

all of the capabilities previously available in the Admin Utility and includes some new capabilities. In particular, you can use QPtool to register places with the Place Catalog and subsequently generate reports from the Place Catalog about basic usage information—about the Service that is currently deployed.

QPtool is a Domino console command, so people familiar with using the Domino console can get all of the Domino console benefits when using QPtool. For example, you can use the Domino Administrator client to access a remote server and to call QPtool using the remote console feature in Domino.

Administrators can schedule QPtool commands using Domino scheduling. They can also script QPtool from the operating system shell, whether it's the C shell or the DOS prompt. We provide administrators with a variety of ways to access administration functions and with different ways of programming them, ranging from simple programming in the operating system shell or inside Domino to full programmatic access through Java.

We talked a little about portals. Could a developer build a WebSphere portal to access QuickPlace?
Yes and no. In 3.0, we're making it easy for you to integrate QuickPlace with portal applications, in particular, operations that you want to perform outside of the place itself and bring into a portal, like creating places and managing membership. We provide portal developers with the option to create custom user interfaces for these functions in the portal, so there's a more graceful relationship between the portal and QuickPlace. Our work on PlaceTypes strengthens developers' ability to build custom templates that share the look and feel of a portal application. However, when users want to collaborate technically speaking, they'll leave the portal and go to the QuickPlace workspace.

In general, our message to portal developers is that it's very easy to expose the management functions of QuickPlace in any portal, whether it's WebSphere or a Domino Web application, and then, using HTML links, to take users to the browser-based workspace of QuickPlace where they can actually work. So, there's already a very nice level of integration in terms of fluidly switching between QuickPlace and the portal and vice versa.

The Place Catalog may bring some additional possibilities for portal developers because it allows some custom data to be associated on a place-by-place basis. There may be some additional possibilities for exposing catalog data in a portal.

This release has tighter integration with DOLS and Sametime. Any plans for integration with iNotes as a mail client for QuickPlace?
No plans directly. Typically, QuickPlace is designed to be neutral to the messaging infrastructure of the customer. We are providing the calendar subscription mechanism that I mentioned that, because iNotes is compatible with the Notes Calendar schema, iNotes will take advantage of it, and calendar events can show up in iNotes. Aside from that, we typically find that some customers deploy several Lotus Web applications alongside each other, and we focus on interoperation of QuickPlace with such an environment and with other technologies from other vendors.

ABOUT CHARLIE HILL

Charlie Hill is the product designer for Lotus QuickPlace and a member of the Lotus Design Council and the Lotus Architecture Board. Before joining the QuickPlace project, he designed Instant!TeamRoom 1.5. In his checkered career before joining Lotus, Charlie was a user experience designer at Apple Computer, working on next generation personal computing environments and integrated hardware/software appliances. Prior to that, he was a researcher and visiting tutor at the Royal College of Art in London, where he designed a workspace for fashion designers, and before that he spent way too long at various educational institutions in the UK, studying engineering and design. Charlie lives with his wife and daughter near Boston, MA.