Notes.net Iris Today

by
Dwight
Morse

Level: Intermediate
Works with: Domino 5.0
Updated: 02/01/2001

More and more companies are looking at browsers as a viable client option in place of bigger, more fully functional clients like Notes. Browsers have a smaller footprint, making them easier to install. There is less to configure when setting up browsers, and end users generally are familiar with their user interface, lowering training costs. Their lack of functionality has always been a problem, but technologies like iNotes Web Access and Domino Off-Line Services (DOLS) are starting to bridge that gap.

What does this mean if you are a Notes/Domino administrator? Suddenly, you need to research what large-scale groups of browser clients mean in your world.

This article takes a look at browser access to Domino services from the old guard, reviewing new, browser-specific features in Notes/Domino terms. It covers everything from registration, to mail, to security, in terms that you, the Domino administrator, understand.

This article assumes a general knowledge of Notes clients and Domino servers. More specifically, it assumes that you are comfortable registering Notes users and setting up their mail, and that you have a basic understanding of Notes security.

Some basics

Rather than assume that everyone has their PhD in acronyms, let's start out by briefly stating some things that many of you may already know. This will fill in any gaps some of you may have so that I don't lose you later. You can skip this section if it's too basic for you.

Browsers communicate with Domino via a number of Internet protocols. Domino does not "talk" all of these languages by default. For Domino to communicate via any one of these protocols, that service must be started on the Domino server. The most common of these Internet protocols is HTTP (Hypertext Transfer Protocol). With Release 5, you have the option of using Domino’s HTTP stack or running with Microsoft’s IIS (Internet Information Server) stack. You can start the HTTP service every time the Domino server starts by adding HTTP to the ServerTasks= line in the server’s NOTES.INI file. For example:

```
ServerTasks=Replica,Router,Update,,AMgr,Adminp,Sched,CalConn,
Event,HTTP
```

The HTTP service allows browsers to open any Notes database. The HTTP task converts the Notes views and documents into HTML (Hypertext Markup Language) for the browser to display. Some forms and views translate better than others. At Iris, we’ve spent a lot of time figuring out the balance between form (what looks cool) and function (what loads fast). We’ve applied what we’ve learned to the existing Webmail template and the new iNotes Web Access client, which we recommend you use for browser access to mail.

Registering browser clients

You don't need to install anything on a computer to get a browser to work with Domino, but an "account" must be set up for authenticated access. In this case, by "setting up an account" I mean creating the familiar Person document in the Domino Directory and creating the user's mail file.

Many people believe that the registration tool in the Domino Administrator is only useful for creating Notes IDs. That isn’t true. You should use the registration tool to register any and all users in the Domino domain, including all browser users.

Using the registration tool has several distinct advantages over creating Person documents, one-by-one, for each browser user. First, the registration tool lets you import existing users from directories (for example, NT, cc:Mail, Netscape, and so on) or text lists. This saves you from having to hire an intern to create thousands of Person documents. Second, the registration tool triggers the creation of the user’s mail file. Additionally, the registration tool lets you configure a number of options—such as password quality, Internet address format, mail template, mail file quotas, and group membership—for multiple users, all at one time.

When registering browser users with the Domino Administrator, you don’t need to create an ID file. You can uncheck the boxes asking where you’d like to store the user’s ID. Beyond that, you need to specify the mail system as IMAP, POP3, or Lotus Notes (for Webmail and iNotes Web Access). There is one issue that you need to be aware of when registering Web users from the Administrator. When specifying Lotus Notes as the Mail System, the Administrator will force you to create a Notes ID. To work around this issue, you can allow the Administrator to create the Notes ID and then delete it or you can just ignore the ID.

Registering browser users using the Administrator registration tool will allow you to create Internet passwords for them, resulting in a challenge/response security model that is much different than the public key infrastructure you are used to with Notes clients. If you want to get closer to the Notes/Domino security model, you’ll have to take another step beyond registering the users and set up SSL for your Domino server. Though it is important to note this fact here because registering Notes users creates the public and private key pair, details about setting up SSL appear in the **security** section later in this article.

Understanding browser access to mail

The Extended Mail template (mail50ex.ntf) affords browsers with Notes-like access to mail. The mail file is an NSF that gives browsers much the same UI as they would have through a Notes client, while providing the same mail back-end you are familiar with. Browser users can send and receive mail through the NSF, which means that you, as the administrator, can manage, monitor, and track mail usage using the same set of tools used for Notes clients. There are some differences, however.

For instance, you can set mail quotas in the same way you do for Notes clients, but you can only manage quotas on the server. Users will have local mail files only if they are using Domino Off-Line Services to bring their mail NSFs locally to access them through a browser when off-line. Since browser users cannot compact these local replicas of the mail file in the way a Notes client can, users can manage space in the mail database on the local copy only by limiting the size of what is replicated and deleting old messages so new messages can reuse the freed space in the NSF. On the server replica, however, you can run COMPACT to retrieve white space.

Also, the way mail is sent from browsers is slightly different than the way it is sent from Notes clients. Users compose the message with the browser, but when the message is sent, it is deposited into the server’s MAIL.BOX using the HTTP protocol. From there, the Domino router is used to forward the message to the recipient using the Domino router via SMTP or the native

Notes routing facility. At this point, the message is the same as if it were sent by a Notes client.

You can monitor mail usage and throughput on the server using log files and statistics. The same can be said for tracking messages. You can track all messages sent by browsers by implementing the Message Tracking Collector. In other words, browsers are not capable of tracking their own messages the way a Notes client can. You again have to rely on the server's capabilities.

Another difference in the way that mail works through a browser is that contact information is stored in the mail file rather than in a separate Personal Address Book. This adds some bulk to the mail file, depending on how many addresses are added to the user’s Personal Address Book. Of course, a perfect way to cut down on the size of the file is to employ a retention policy that necessitates the deletion or archiving of older documents. Obviously, you don’t want to rid the database of contact information, but you don’t need to worry about that. These documents have fields ($NoPurge and PROTECTFROMARCHIVE) set to keep them in the database.

Because the mail file is an NSF, it’s possible to move the mail file using the new Administration Request in Release 5.0. This is initiated from the Domino Administrator client, in the Mail Users view. This request automates the steps needed to move a user’s mail from one server to another successfully, ensuring that no mail is lost during the move. The Administration Process (AdminP) creates separate requests for each of these steps and reports on their successful completion. In general, these steps include:

- Replicating the mail file to the new server
- Modifying the Person document to reflect the new mail file location
- Pushing down changes to the user’s Location document
- Replicating the old mail file with the new mail file
- Deleting the old mail file

However, not all of these steps are applicable to browser clients. While they can benefit from having the replica created and Person document changed, browsers will not have Location documents to modify. This poses a small problem. Moving a mail file is an ordered process. One step must be completed before the next is started. The Location document modification will never be completed with browser clients, which means that you’ll need to complete the last two steps of the process. You'll need to replicate the old mail file with the new one, one last time, after the change to the Person document has replicated throughout you domain. Once the replication has completed successfully, you’ll need to delete the old mail file.

There is an alternative to deleting the mail files yourself. You can download some LotusScript called “**Create Push Change Request Agent**” from the Iris Sandbox on Notes.net. You can use this LotusScript to create an agent in the Administration Request database that sends mail to the user whose mail file has moved and creates a request to replicate the two mail files one last time along with a subsequent request to delete the old mail file. Instructions about using this agent appear on its download page.

Providing browser access to databases

The Domino server is able to serve all of your databases through the HTTP service. The views and forms are very usable, but they do not provide the same user experience that browser users are accustomed to. Domino Designer allows you to modify databases to provide a more Web-like look and feel, supporting such functionality as frames and JavaScript, and you can add that functionality to any existing Notes database. If you have a mixed environment, where both browser and Notes clients will be using a database, you can specify different experiences depending on which type of client is accessing the database; for example, you can design one field for the Notes client and another for a browser and then use the Hide from Notes/Hide from Web options so the proper fields appears for each client.

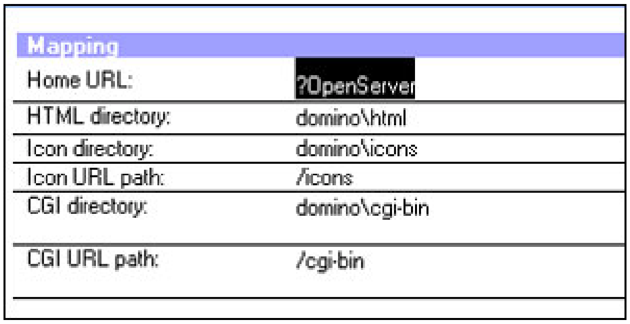
To prepare for browser access to your Domino server, you should also think of what you'd like browser users to see. Keep in mind that using a browser for access is a very different experience than the experience Notes users are accustomed to.

For instance, where Notes users choose File - Database - Open and are presented with a list of servers and then databases, browser clients connect to a specific URL. While this may seem limiting at first, it actually gives you the opportunity to build the user experience. You may choose to create a portal (Web page) that has links to all the databases that your users need. This can significantly cut down on your administration costs, because you can manage the Web page centrally and affect all users who are accessing the databases. For example, if you move a database to a different server, you only need to edit the link; whereas for Notes clients, you probably send an e-mail telling them how to remove the old bookmark and replace it with the new one.

This simplicity breaks down when we start to talk about user-specific databases, specifically mail. Since a unique URL is needed for each user's mail file, you might think that you would need as many portals as you have users. But remember that this is software, and that means that anything is possible. I've seen a Web server running on "potato power," so who's to say that we can't redirect users to their individual mail files given a name and password? In fact, you'll find such an example application in the Iris Sandbox on Notes.net. Look for "**MailJump**."

Given all this functionality, maintaining a page that has links to all the databases that your browser users might need can still be a large task. As an alternative, you might decide to present a list of the databases available on the server when the server is accessed. Such a list is not what appears by default when you enter a server's IP address into a browser, however.

When a browser accesses a Domino server, a default page comes up showing links to Notes/Domino Help, Release Notes and the Lotus, IBM, and Notes.net sites. While this is helpful, it is a far cry from the Notes experience. There is no list of databases to access nor is there a list of servers in the domain. The reason these lists do no automatically appear has to do with security. You may not have reviewed all the database's ACLs to make sure that their default access is appropriate. If you *have* reviewed the ACLs and feel comfortable with providing a list of your server's databases automatically, you can do so by modifying the Server document. Simply go to the Internet Protocols tab and edit the HTTP section so that the Home URL field says "?OpenServer."



Mapping	
Home URL:	?OpenServer
HTML directory:	domino\html
Icon directory:	domino\icons
Icon URL path:	/icons
CGI directory:	domino\cgi-bin
CGI URL path:	/cgi-bin

It is important to note the difference between browser and Notes client access in terms of capacity planning. The number of concurrent browser users that a Domino server can support is less than the number of concurrent Notes users. This number varies depending on the database, but in general, you can expect a 3 to 1 ratio for Notes client to browser client users, and you should plan accordingly. This ratio will continue to improve and will improve considerably in Rnext for the iNotes Web Access client.

What about private views and folders?

Browser users cannot create private views of databases, a fact you must take into consideration when creating or modifying templates. You may have more requests to create views for browser users than you receive in a Notes client environment. Many users may prefer a particular type of view that is not in the standard template, and you will have to assess its popularity to see if it should be added to the template.

Browser users, however, can create their own folders, so you won’t have worry about their ability to categorize their mail.

Calendar & Scheduling features

Browser clients will be able to take advantage of the Calendar & Scheduling functionality available in Notes. They can schedule appointments, meetings and the like and their freetime will be available to other browser clients. If you implement Domino Off-Line Services, they’ll be able to schedule appointments while they are disconnected, just as they would with a Notes client, and then replicate their freetime back to the server. You can set up resources (rooms and equipment) for reservation for browser clients just as you have for Notes clients.

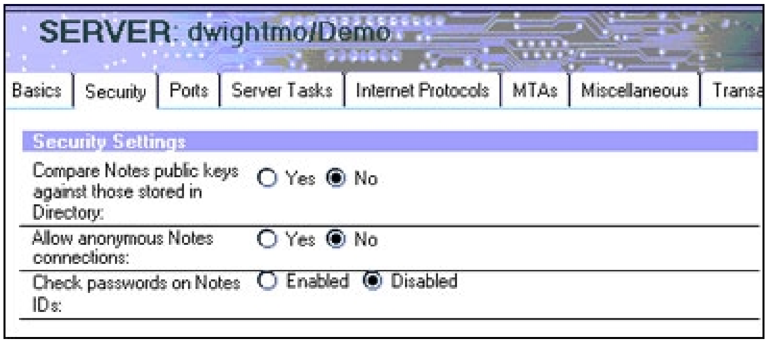
Using delegation

Browser users are able to take advantage of mail file delegation in the event that someone else needs access to their mail or calendar. They will not, however, be able to modify the ACL. (This is not necessarily a bad thing. The user will not be able to change the default ACL that you have assigned from a browser, which may cut down on problems with agents, archiving, replication, and possibly mail routing. Most importantly, as anyone who has experienced this will agree, browser users will not be able to delete their mail files accidentally.)

Last but not least: Security

Security is possibly the biggest concern for Domino administrators moving from Notes clients to browser clients. When you set up a Person document for a browser user, you can include an Internet password. This password is used for a challenge/response security model. When a browser accesses a database (mail or otherwise), Domino asks for a name and password. It checks for the name in the Domino Directory and compares the password given against that provided in the Person document. Many customers feel that this is adequate security when setting up their Domino intranet. Domino allows for variations in security when accessing databases. You can provide either less or more secure connections to your databases than this challenge/response model.

If secure connections are not a concern, you can allow anonymous access to a database, by specifying Anonymous as a user in any database’s ACL. There are two things to consider when implementing this feature. First, the Domino server must allow anonymous access. This is configured in the Security tab of the Server document.



Security Settings

Compare Notes public keys against those stored in Directory: ○ Yes ● No

Allow anonymous Notes connections: ○ Yes ● No

Check passwords on Notes IDs: ○ Enabled ● Disabled

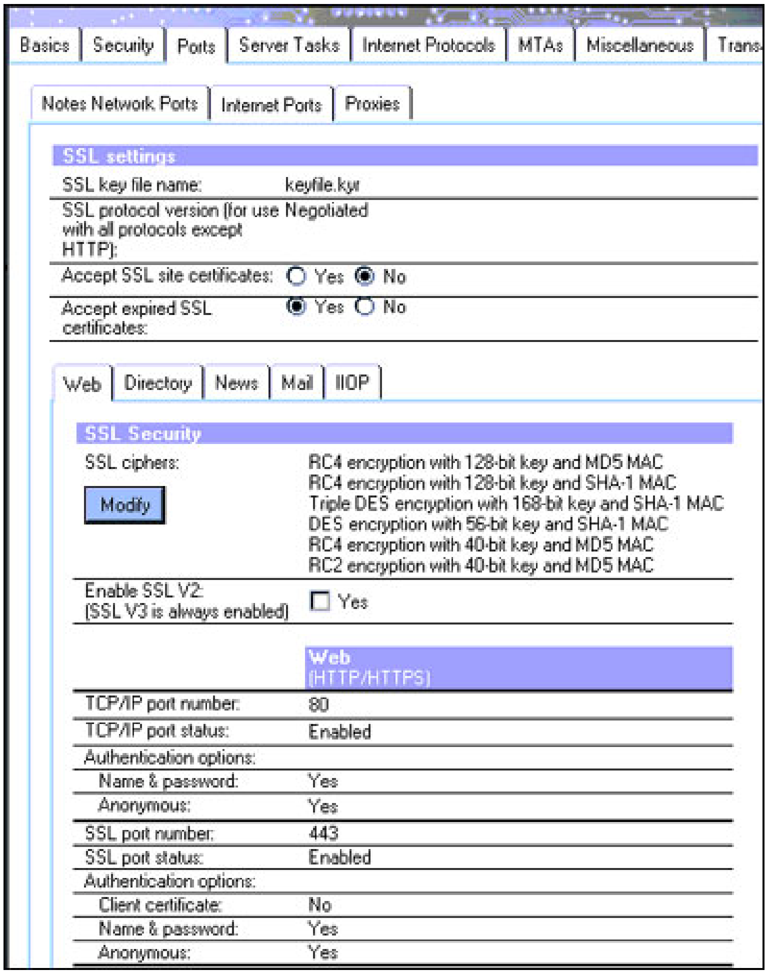
Second, all anonymous users will be seen as the same by the Domino server. This means that you will not be able to distinguish anonymous users. When tracking usage of the database, all these users will be grouped together as the user “Anonymous.” You will not be able to determine who, exactly, is using the database. This makes it impossible to be certain that users are accessing the replica that is located closest to them. This is a concern for administrators that want to ensure the best user experience, in terms of performance, for all users. Think of it this way—you want your users in your New York office to access the replica in New York and your users in London to access the replica in London because the database will be served more quickly over a local LAN connection. The way to monitor this is to track database usage in the server log. It is impossible to do this if a user is seen as Anonymous instead of John Doe/New York/ACME. Authenticated server access, using the challenge/response model precludes this issue.

While the Internet password presents a more secure model than anonymous access, it is still short of the public/private key pair that you are accustomed to with Notes clients. Having a private key locally on the client goes one step further to ensure that the person accessing the server is who they say they are. Instead of just needing to know a user’s name and password, private keys require that the user has a key in their possession that matches the public key located in their Person document on the server. For Notes clients, this key is located in their ID file. You can achieve this same type of security for browsers by issuing X.509 certificates. The X.509 certificate becomes the browser ID file.

Notes IDs are created using the certifier ID that was made when you set up the first server in your domain. The Certificate Authority was assumed. With X.509 certificates, you can choose an outside Certificate Authority like VeriSign. If your users are purely intranet, and we’ll assume that they are here, there’s no reason to bring an outside Certificate Authority into the picture. To mirror the Notes ID model, you will become your own Certificate Authority. The Certificate Authority is the equivalent to the Notes certifier. If both the user and server have X.509 certificates created from the same Certificate Authority, private key authentication can occur.

To better understand how X.509 certificates work, we can think of them in Notes terms. The Certificate Authority is like the Notes certifier, the key ring is like the Notes ID, and the X.509 certificate is like certificates contained within the Notes ID. You should reference the **Domino R5 Administration Help** for assistance in setting up a Certificate Authority and creating X.509 certificates for SSL-enabled Domino servers. Once that is done, you are ready to distribute X.509 certificates to browser clients for (private key) authenticated access to Domino servers.

To distribute X.509 certificates, you need to use the Certificate Authority functionality provided with the Domino server. The Domino server must be configured to use SSL. To do this, make sure that the SSL port status is enabled in the Internet Ports tab of the Server document.



SSL settings

SSL key file name:	keyfile.kyr
SSL protocol version (for use with all protocols except HTTP):	Negotiated
Accept SSL site certificates:	○ Yes ● No
Accept expired SSL certificates:	● Yes ○ No

SSL Security

SSL ciphers:	RC4 encryption with 128-bit key and MD5 MAC RC4 encryption with 128-bit key and SHA-1 MAC Triple DES encryption with 168-bit key and SHA-1 MAC DES encryption with 56-bit key and SHA-1 MAC RC4 encryption with 40-bit key and MD5 MAC RC2 encryption with 40-bit key and MD5 MAC
Enable SSL V2: (SSL V3 is always enabled)	□ Yes

	Web (HTTP/HTTPS)
TCP/IP port number:	80
TCP/IP port status:	Enabled
Authentication options:	
Name & password:	Yes
Anonymous:	Yes
SSL port number:	443
SSL port status:	Enabled
Authentication options:	
Client certificate:	No
Name & password:	Yes
Anonymous:	Yes

Although we can use comparative terms, actually creating X.509 certificates for browser clients is quite different than creating Notes IDs. A better comparison would be recertifying Notes IDs. When a Notes user needs to recertify their ID, they need to request a certificate. The request is sent to the Certificate Authority. The Certificate Authority creates the certificate and sends it back to the user via e-mail. The user then accepts the certificate, thus ending the process. When browser clients wish to authenticate using X.509, they go to the Certificate Authority database on the Domino server and request a client certificate. That request is approved by the certificate authority and the certificate can then be picked up at the Certificate Authority database.

The main difference between certificate distribution for browser clients as opposed to Notes is the uncoupling of client and certificate. It is assumed that if you use the Notes client, you have an ID, and further that that ID contains a certificate capable of authenticating you with servers in your domain. The same cannot be said for browsers. In fact, you can almost be certain that the browser was installed long before any client certificate request from a Domino Certificate Authority. While the distribution of certificates for Notes clients is coupled with the installation of the client software, we must now think of a different way to get certificates to existing browsers. As mentioned above, a URL to their certificate can be sent via e-mail. But this is not a good method if your mail is on a Domino server and you need the X.509 certificate to access it. That basically leaves you two options.

- You can have the user request their own certificate by accessing the

Certificate Authority database with their browser and then notify them by means other than e-mail when their request has been approved so that they can revisit the site to pick it up.

- You can request and pick up the certificate yourself (the same browser must be used for both tasks) and distribute it to the client by means other than e-mail. You need to export the key and provide the user with instructions on how to import the key.

Since the only real advantage to this second option is that it obviates the need to explain to the user how to request and pick up a certificate (there is no option for batch creation), I recommend using the Certificate Authority functions provided with Domino (unless you think your users will not be able to successfully complete the process). The first option also provides a more secure requisition in that only the client requesting the certificate ever has possession of it.

Now that we know the choices for authentication, let’s talk a little about maintenance. First of all, password checking and expiration for Internet passwords does not exist in Domino. Further, Internet passwords cannot be synchronized with NT passwords in the same fashion that Notes passwords can. You can choose to trust NT, however, and get virtually the same outcome; a one-password challenge. A browser user changes their password by editing the Internet password field in their Person document, so you need to give them access to edit that field. Keep in mind that because the password is contained in the Domino Directory, any change must be replicated before taking effect on any given server.

Conclusion

If you came into this article with a clear understanding of how to create and manage Notes client users, you're hopefully leaving with a better understanding of how that translates to the browser client world. You should feel more comfortable about setting up browser access to servers and databases, understand the mail and calendar expectations, and know the differences in the security model associated with browser access. You're now ready to roll out that iNotes Web Access and/or DOLS pilot.