



## Creating expiring registration

by  
Lynda  
Urgotis

**Level:** Intermediate  
**Works with:** Domino 5.0  
**Updated:** 10/03/2001

*Editor's note: The Notes.net Exposed series provides you with a behind-the-scenes look at the Notes.net Web site. This time, we explore how we set up user registration for the [iNotes Web Access demo](#) so that user accounts expire after 48 hours.*

It's no secret that something's going on behind the scenes. You go to the iNotes Web Access demo on Notes.net, register, and you're in business, doing things with your Internet Explorer browser that you never thought possible—gleefully sending e-mail, scheduling meetings, and making progress against your To Do list.

Seven days later, it's like you never existed.

So what's going on? How did Monday's demo-persona "Jesse Ventura" get bounced out of the ring a week later? Read on for the true story of how the iNotes Web Access demo came to be, and how the backstage maneuvers of a couple of agents helped make it all possible.

## To be or not to be

It was a serendipitous moment. Our content developer, Christie Williams, was working with Jason Dumont (director of client strategy and the iNotes client) and Vinod Seraphin (lead software architect) on an interview for the November issue of *Iris Today*. Hmm, they wondered, would a live demo of iNotes Web Access work on Notes.net? They took the idea to Barb Mathers, site manager for Notes.net, to see what she thought. She liked it, and the timing was right. Over the past year, Barb had started campaigning to get more demos on Notes.net. There was also a precedent. About a year ago, we launched our first live demo on the site, the very successful DOLS (Domino Off-Line Services) demo. And actually, you could say we've been hosting demos even longer, because as Barb points out, Notes.net is really one big live demo of Notes and Domino technology. We've been "demoing" since the inception of the site four years ago.

The next question was, "If we build it, will they come?" After all, the beta version of iNotes Web Access was available on Notes.net. Was there enough interest and value in putting up a live demo? We decided that the answer was yes, because the demo would let you get the iNotes Web Access experience almost instantly, with none of the administration or hardware overhead that's necessary to set up the full beta version in your own environment.

With everyone convinced that it was a good idea, we needed to decide what exactly the demo experience was going to be like.

## One account or more?

We had two goals: one was to keep things simple on our side, and the other was to provide a real-world experience of using iNotes Web Access to anyone who registered for the demo.

Our plan to keep things simple began with the notion of having one account on the server, that everyone would use. Simple, yes, but it turned out to be

an unworkable solution. For one thing, there would be no way of controlling how many simultaneous users were active. And with each person customizing iNotes Web Access to fit their own requirements, it could be a confusing first-look. Your environment would be constantly changing, and you'd be looking at someone else's idea of "cool" preference choices. We didn't want to have to warn every new user that the user interface they saw would mysteriously change every time they logged on.

Since one major feature of iNotes Web Access is being able to send and receive Internet e-mail, everyone needed their own mail account and database—another good reason to adopt the "one account per user" model.

Then we started thinking about numbers. Being optimistic types, we envisioned that lots of people would want to try out the demo. (It turns out that we were right—there were more than 20,000 registrations in less than three months.) But just imagine thousands of registered users and their mail files, all hanging out on the server—forever. Clearly, we needed to set some limits or we would be the victims of our own success. So, we decided that 48 hours after registration, the account and e-mail access would expire. Later, we extended the time limit to seven days. But we're getting ahead of the story.

### Registration: Thank you King Kong for signing up

Check another item off the list. We now had a general sense of what the overall iNotes Web Access demo experience should be. Next on the list was how to handle registration. Kevin Giles (application developer for Notes.net) came up with a registration process that was very easy for people to use. All they have to do is go to the Notes.net Web site, click to get to the demo, and enter their name and password. The registration process takes over from there, completing three basic tasks:

- Creating a Person document in the Domino Directory
- Adding the User Name to the iNotes demo Registration database
- Creating the user's mail database based on the iNotes Web Access template

Here's what happens during the registration process, which takes only a minute or two from start to finish. When someone comes to the demo site to register, we prompt them for a first name and last name.

The screenshot shows a web browser window titled "http://showcase2.notes.net/ireg.nsf/iregform - Microsoft Internet Explorer provided by Lotus/IBM". The address bar shows "http://showcase2.notes.net/ireg.nsf/iregform". The page features the "Notes.net" logo and "Lotus iNotes Web Access Demo" text. Below this is a navigation bar with buttons: Home, Demo, Site Map, Site Info, All About Domino, My Domino, and Site Library. The main heading is "iNotes Web Access Demo Registration". Under "User Name Information", it states "All fields are required." and lists four input fields: First Name (containing "King"), Last Name (containing "Kong"), Password (containing "password"), and Verify Password (containing "password"). A red note at the bottom reads: "\* Note: After entering your registration information click the Create Mailbox button. Please wait until you are redirected to the login screen. Do not click the button more than once as this will result in an error." At the bottom are two buttons: "Create Mailbox" and "Clear Entries".

When we get that value, we look it up in the iNotes demo Registration database to see if it's unique. If it's not unique, we display a message (from the Registration Error database) and ask them to give us another name.

We want the user accounts to contain only legal characters, so we check the characters using a view called CharLookup that has a list of all the illegal characters that we don't allow in the user name. For each user name passed in, we do a lookup to make sure that none of those characters are present.

Once we determine that the user name is valid, we create the Person document and populate the fields that are going to be useful for setting up the iNotes Web Access account, such as the e-mail address. That's done by the CreatePersonRecord agent.

Once the Person record is set up and all the fields are populated, we then create the mail database (CreateMailFile) based on the iNotes Web Access template. And then we set up the ACLs on the database.

From there, we send them to the confirmation page that tells them their user name and password, and generates a link back to their mail database.



You can take a closer look at the [iNotes demo Registration and Registration Error databases](#), which are posted in the Iris Sandbox.

We decided to retire each user account name after it had been used once. For us, it was mostly a matter of maintenance, since we didn't want to run into conflicts between similar names and mail files on the server. You might want to customize this registration process for your own use, and allow the same person to re-register under the same name after the original account is disabled.

So, having welcomed the Notes.net community to the iNotes Web Access demo, the next part was figuring out how to say goodbye. And that's where the agents come in.

## Expiring registration: Secret agent man

Some agents do their best work secretly, in the background. The iNotes Web Access demo features two agents who are the strong, silent type. One agent is called the Timekeeper, with virtual stopwatch in virtual hand, and the other is called the Housekeeper, taking a vigorous approach to database

cleanup. These agents, the creation of Moe Peabody (system administrator for Notes.net), are about to share a moment with user King Kong, who registered seven days ago.

Let's recap what's happened up to now. King Kong is officially registered. He has a Person document in the Domino Directory that contains a "create date" timestamp. He's enjoying iNotes Web Access. But in the background, his time is running out. The "create date" timestamp is soon to become very important as the agents wake up and go to work, each with their own special job to do.

Here's a summary of what happens. The Timekeeper runs in the Domino Directory. It moves seven-day-old Person documents into a separate Archive database and deletes them from the Domino Directory. The Housekeeper runs in that Archive database. It looks up the mail files for each of the archived Person documents, deletes them, and then moves the Person documents into an Processed view, which we use to track usage.

Persistence is an important, interesting, and extensible part of the process. In the iNotes Web Access demo, we're using an "on/off" variation of the persistence feature. The Owner field of the Person document contains a persistence value that specifies how long the user's registration will be enabled. If the Owner field is empty, the account is disabled after seven days. If the field contains 999, the account lasts indefinitely. Other values could be used for intermediate time periods.

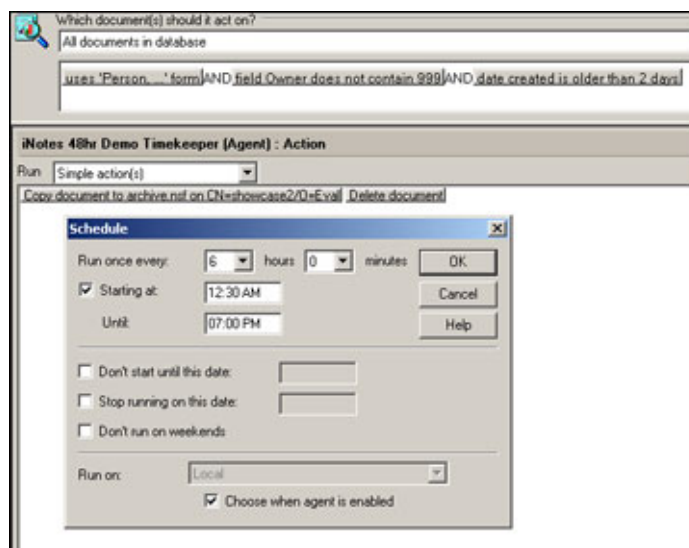
The following sections discuss details about the Archive database and the two agents. You can take a look at the [Archive database](#) (with the agents), which is posted in the Iris Sandbox. (Note that this posted version includes the Timekeeper agent, which belongs in your Domino Directory and should be copied there if you intend to use these agents.)

### The Archive database

The Archive database is a scaled-down Domino Directory containing only two views, People and Processed. The People view is where the expired Person documents are moved by the Timekeeper. The Processed view is where the documents go after the Housekeeper deletes the mail files.

### The Timekeeper agent

The Timekeeper is a simple agent that is put in the Domino Directory.

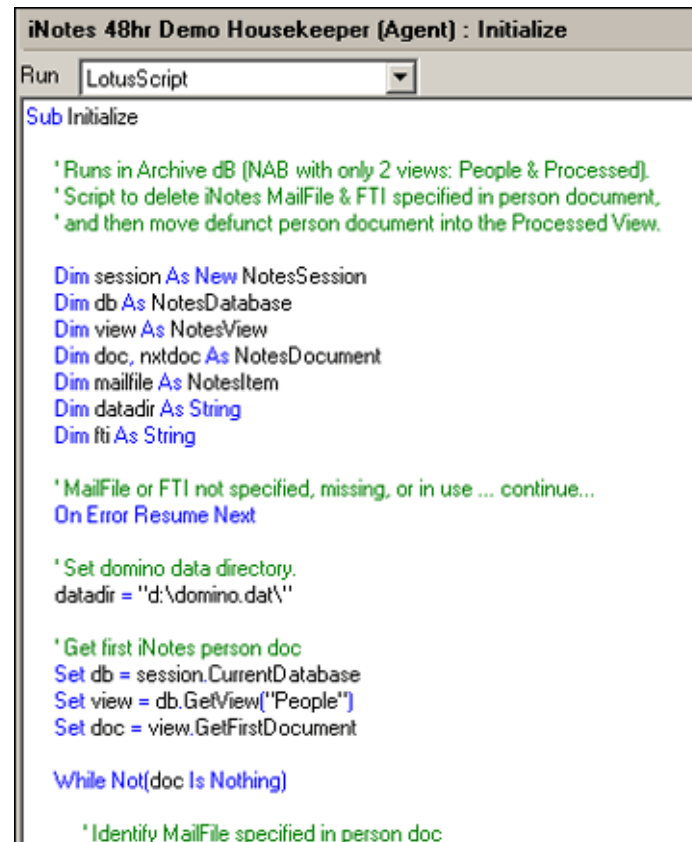


Timekeeper runs on a schedule on all documents in the Domino Directory

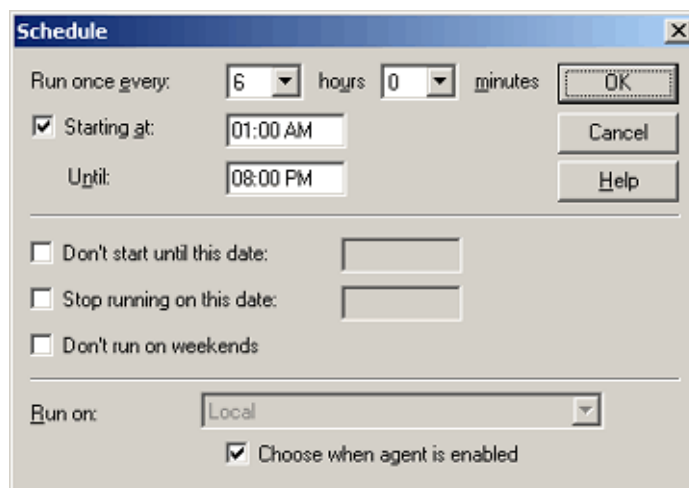
four times a day. When it finds a Person document older than seven days without a persistence value in the Owner field, it deletes it from the Domino Directory and copies it into the Archive database. The account is now disabled.

### The Housekeeper agent

The Housekeeper agent resides in the Archive database. It is a LotusScript agent (which is important to note because you can't delete files on the disk using formula language).



Housekeeper runs on a schedule on all documents in the Archive database four times a day. The agent scans through all the Person documents moved into the People view, looks up and deletes the corresponding mail file along with any full-text indexes, and then moves the old, archived Person document into an Processed view.



### More about persistent accounts

Here's a tip if you decide to create and use persistent accounts. Rather than modifying the Domino Directory, use an existing, unused field in the Domino Directory. We use the Owner field in the Person document. It's best not to modify the forms because then you don't have to redo the changes when you upgrade the Domino Directory. So if you want someone to have access for 30 days, you can enter "30" or some other predefined value in the Owner field of the Person document. The Timekeeper agent can then look at the value and compare it against the "create date" timestamp.

That's how we use it in our demo, but you can also adapt it for different circumstances. Perhaps you have a similar short-term e-mail requirement that could use an automated timekeeper and housekeeper. You can extend the functionality by using different persistence values for different user requirements. For example, an administrator at a college could use it to set up mail accounts for a professor's courses. In this scenario, the persistence value would depend on whether the student was signed up for one or two semesters. The professor could distribute the URL to each class. Then the students would register and get a mail database. In the one-semester course, the mail account would be automatically disabled after 100 days, while the two-semester course wouldn't expire for 200 days.

### Fine-tuning expiring registration

As the iNotes Web Access demo took shape, there were some changes to the agents along the way. First, we noticed that people were creating full-text indexes on their mail files. A look at the server showed all these full-text indexes hanging around after the mail files had been deleted. So we updated the Housekeeper agent so that when the mail file is deleted, the agent deletes any associated full-text index.

We also noticed that some of the mail files were not getting deleted. This happened because they were in use when the Housekeeper tried to delete them. Since we were not allowing same-name registrations (even after an account has expired) the solution was simple—just rerun the Housekeeper later to clean up the stray files. Another solution would be to modify the Housekeeper's error handler to leave the Person documents in the People view if the mail file was in use. Then, when the Housekeeper ran again, it would try to delete the file again, and keep trying until it was successful.

Take note, that if you do modify the registration process to allow same-name registrations you will also need to add checks to the process to deal with stray mail files and full-text indexes. We found it easier to just eliminate same-name registrations altogether. It simplified registration as well as cleanup.

## **A happy ending**

Now that the demo has been available for a while, Jason Dumont notes that it provided even more benefits than anticipated. It allowed the iNotes Web Access team to make available more recent versions of the software than they would otherwise have been able to do because of the testing necessary to post a downloadable build versus a hosted build. It gave the team a chance to get some real deployment numbers based on the use of the site. And it also demonstrated the reliability and scalability of the product in a real-life deployment environment.

For the Notes.net team, the success of the iNotes Web Access demo helps blaze the trail for our next live demo.

The iNotes Web Access demo continues to attract hundreds of users a day since its initial rollout in November. It's interesting to monitor the number of registrations, watching it peak after Lotusphere, when everyone wanted to get their hands on the software. After all, it's only a seven-day commitment. And even if you forget to watch the clock, the agents remain vigilant, attending to their chores, methodically checking credentials and making room for the next batch of iNotes Web Access users.

## **ABOUT THE AUTHOR**

Lynda Urgotis began her career during the Paleolithic era writing about chipped-stone tools. She has documented her way from Data Resources to PSDI to Software House. She is rapidly approaching her thirteenth anniversary at Lotus where she contributed to Lotus Improv, SmartSuite, eSuite, and now, Notes. She wrote the Que book, "Quick Reference for Improv" as well as chapters for the Improv Que book. Gardening delights her spirit. Her daughter, Megan, and husband, Michael, are just plain delightful.