Troubleshooting single sign-on (SSO) between IBM WebSphere Portal and IBM Lotus Domino

Charles Price
IBM Software Group
Advisory Software Engineer, Domino Portal Integration
Atlanta, GA

August 2009

© Copyright International Business Machines Corporation 2009. All rights reserved.

Editor's Note: This white paper is the third in a three-part series on SSO published over the past few months. See the previous papers, “Understanding single sign-on (SSO) between IBM WebSphere Portal and IBM Lotus Domino,” and “Configuring single sign-on (SSO) between IBM WebSphere Portal and IBM Lotus Domino.”

Summary: This document is designed to help administrators who have configured SSO in their environment but find it is not working correctly. In this paper, we explain exactly how to test, configure, and read debug so you can determine and resolve the root cause of whatever SSO blocker you are encountering in your environment.

Table of Contents

1 Introduction ............................................................................................................................... 2
2 Enabling debug when SSO fails to Lotus Domino................................................................. 2
   2.1 Analyzing the debug ......................................................................................................... 7
3 Troubleshooting error messages on the Domino console.................................................... 11
   3.1 No debug written to console ......................................................................................... 11
   3.2 Unable to decrypt token passed by WebSphere Portal .................................................. 16
   3.3 Realm values do not match ........................................................................................... 17
   3.4 Token has expired ........................................................................................................... 19
   3.5 No errors in Domino console, but SSO continues to fail ............................................. 21
4 Conclusion ............................................................................................................................ 23
5 Resources ............................................................................................................................. 23
6 About the author ..................................................................................................................... 24
1 Introduction

Assuming you've read the previous two white papers in this series on SSO, you should have a good understanding of how SSO works and have gone through the steps to configure and test SSO in your environment. This third of three papers walks through the steps to debug SSO, giving detailed examples of how SSO could still fail, and what actions are necessary for resolving those issues.

If SSO is currently failing, debug must be collected to understand the issue. You can enable SSO on either the IBM® WebSphere® Portal server or the IBM Lotus® Domino® server; it's only necessary to enable the debug on one of the servers, and this should be the server you access after signing in.

So if you sign into WebSphere Portal, then change the browser to Lotus Domino, and that fails, then follow the steps below in “Section 2, Enabling debug when SSO fails to Lotus Domino.”

If, however, you sign into Lotus Domino first, then change the browser to the WebSphere Portal server, and that is the only way SSO fails, skip to “Section 3, Enabling Debug when SSO fails to WebSphere Portal.”

If SSO fails in both directions, which is the most common case, it is recommended to debug with the Domino debug described in Section 2. Most likely this will resolve the issue both ways and, of the two debugs, it's the easier to enable.

2 Enabling debug when SSO fails to Lotus Domino

To enable and collect SSO debug on Lotus Domino, follow these steps:

1. Enter the following commands on the Domino console:

   > set config debug_sso_trace_level=2
   > set config websess_verbose_trace=1
   > tell http q (causing the following messages to appear on the Domino console:

      06/26/2009 01:29:13 PM  Domino Off-Line Services HTTP extension unloaded.
      06/26/2009 01:29:14 PM  HTTP Server: Shutdown
   > lo http

   If the debug was enabled correctly, you should see the following debug prints, showing how Domino reads the current SSO configuration:
Here we read the WebSSO document, whose settings are all described in detail in the white paper, “Understanding single sign-on (SSO) between IBM WebSphere Portal and IBM Lotus Domino”:
Now let's go through the process of creating an LtpaToken. Notice how the values of the document are used to populate the token with the three pieces of information found in every token: that is, realm value, followed by the user's Full DN (CN=null in this case as Domino is just confirming that it can create a token), and finally, the expiration time:

06/26/2009 01:30:36.27 PM [2930:0002-2CA0] SSO API> Setting token name parameter [LtpaToken]
06/26/2009 01:30:36.27 PM [2930:0002-2CA0] SSO API> Setting token domain parameter [.ibm.com]
06/26/2009 01:30:36.27 PM [2930:0002-2CA0] SSO API> Creation time not specified, using current time [06/26/2009 01:30:36 PM].
06/26/2009 01:30:36.27 PM [2930:0002-2CA0] SSO API> Expiration time not specified, using current time plus config expiration [06/26/2009 05:30:36 PM].
06/26/2009 01:30:36.27 PM [2930:0002-2CA0] SSO API> Encoding WebSphere style Single Sign-On token (LTPA).
06/26/2009 01:30:36.27 PM [2930:0002-2CA0] SSO API> -LDAP Realm=WMMRealm
06/26/2009 01:30:36.29 PM [2930:0002-2CA0] SSO API> -Username=null
06/26/2009 01:30:36.34 PM [2930:0002-2CA0] SSO API> -Expiration Ticks = 1246051836666 [06/26/2009 05:30:36 PM].

Below is the actual token. Again, notice the realm, followed by user's Distinguished name and expiration time:

06/26/2009 01:30:36.34 PM [2930:0002-2CA0] SSO API> Dumping memory of constructed token before encryption step [211 bytes].
00000000: 3A75 7375 7265 3A5C 4D57 524D 6165 6D6C   'u:user'
00000010: 432F 3D4E 756E 6C6C 3125 3432 3036 3135   '/CN=\nul1246051'
00000020: 3338 3636 3636 4325 4464 506D 4179 5150   '36666%CdDmPyAPQ'
00000030: 414D 5233 4952 426D 4J71 4431 632B 6570   'MA3RRImBnJqG1D/c'
00000040: 5649 4276 614E 7330 7737 6C67 5730 5675   'IVvBNa0s7wgl0WuV'
Now we encrypt the token using the private keys imported from WebSphere Portal into the WebSSO document in Lotus Domino:

06/26/2009 01:30:36.35 PM [2930:0002-2CA0] SSO API> Dumping memory of constructed token after encryption step [216 bytes].

06/26/2009 01:30:36.37 PM [2930:0002-2CA0] SSO API> Dumping memory of encoded token [288 bytes].

Finally, we run the encrypted key file through a Base64 encoder and send the token to the browser:
At this point the Domino server is able to generate LtpaTokens and complete the load of HTTP:

> 06/26/2009 01:30:37 PM  HTTP Server: Started

2. Open a browser and sign into WebSphere Portal (see figure 1).
3. On the Domino console enter:

   > start console log

4. Change the URL in the address bar to the database with which you are testing (the mail file, in our example). You should see a log-in screen, but DO NOT log in. Just stop here and review the debug.

5. Open the console.log file located in the the <domino_data>/IBM_TECHNICAL_SUPPORT directory.

6. Search in the log file for the last entry of start console log. This should be the last line before the SSO debug is generated.

There are four possible cases that we see when the debug is collected, which we discuss in detail below:

2.1 Analyzing the debug

Before we look at the four cases, it's useful to first discuss what the debug should look like when SSO works, which should make the error messages easier to understand. When SSO works, the debug will print out the following:
First, the browser sends the Domino server the token:

06/26/2009 02:08:35.57 PM [3130:000E-3680] HTTP Sessions> Validating single sign-on session cookie (LtpaToken) (wPnjASwXYx117KOTpDjd4DCypKWXscnjjqg+VlTrcl9gee9cs3as9IVPD/ySXbK/K8+Uf+0a7QN+8MeUPsoBzWk4gbDkj9XVoXX1y4L9lQDcF68e4auOfh/ ZPnXht7I6pgqosawX1YLwhA

Once the Domino server reads in the LTPAToken, it determines whether it has already validated the incoming token by checking the cache:

06/26/2009 02:08:35.59 PM [3130:000E-3680] HTTP Sessions> Single sign-on session cookie not found in cache, decoding. [ORG=, CFG=LtpaToken]

Because it is not in the cache, we read in the token and prepare to decrypt it:

06/26/2009 02:08:35.59 PM [3130:000E-3680] SSO API> *** Retrieving Extra Token Info (SECTokenValidateAndGetTokenInfo) ***
06/26/2009 02:08:35.59 PM [3130:000E-3680] SSO API> ConfigName specified [LtpaToken].
06/26/2009 02:08:35.59 PM [3130:000E-3680] SSO API> Dumping memory of encoded token [312 bytes].
06/26/2009 02:08:35.59 PM [3130:000F-2540] HTTP Sessions> Validating single sign-on session cookie (LtpaToken) (wPnjASwXYx117KOTpDjd4DCypKWXscnjjqg+VlTrcl9gee9cs3as9IVPD/ ySXbK/K8+Uf+0a7QN+8MeUPsoBzWk4gbDkj9XVoXX1y4L9lQDcF68e4auOfh/ ZPnXht7I6pgqosawX1YLwhA
00000000: 5077 6A6E 5341 5877 7859 4931 4B37 544F  'wPnjASwXYx117KOT'
00000010: 4470 646A 4434 7943 4B70 5857 6373 6A6E  'pDjd4DCypKWXscnj'
00000020: 6771 562B 546C 3637 396C 6567 3965 7363  'qg+VlTrcl9gee9cs'
00000030: 7333 3961 5649 4450 792F 5853 4B62 4B2F  '3sa91VPD/ySXbK/K'
06/26/2009 02:08:35.60 PM [3130:000F-2540] HTTP Sessions> Single sign-on session cookie not found in cache, decoding. [ORG=, CFG=LtpaToken]
00000040: 2B38 6655 302B 3761 4E51 382B 654D 5055  '8+Uf+0a7QN+8MeUP'
06/26/2009 02:08:35.60 PM [3130:000F-2540] SSO API> *** Retrieving Extra Token Info (SECTokenValidateAndGetTokenInfo) ***
00000050: 6F73 7A42 6B57 6734 4462 6A6B 5839 6F56  'soBzWk4gbDkj9XVo'
06/26/2009 02:08:35.62 PM [3130:000F-2540] SSO API> ConfigName specified [LtpaToken].
00000060: 5858 7931 4C34 6C39 4451 4663 3836 3465  'XX1y4L9lQDcF68e4'
06/26/2009 02:08:35.67 PM [3130:000E-2540] SSO API> Dumping memory of encoded token before decryption step [232 bytes].
06/26/2009 02:08:35.67 PM [3130:000E-2540] SSO API> Retrieved global static cache memory for config [LtpaToken].
00000000: F9C0 01E3 172C 1D63 EC48 93A3 38A4 E0DD   '@yc.,.c.Hl#.$8`'

Now we decrypt the Base64-encoded token:

06/26/2009 02:08:35.67 PM [3130:000F-2540] SSO API> Dumping memory of encoded token [312 bytes].

Finally we decrypt the token, using the public keys imported from WebSphere Portal into the WebSSO document. Notice how we write out the three values written into the token (realm, followed by the user's full DN, then the expiration time):

```
06/26/2009 02:08:35.68 PM [3130:000E-3680] SSO API> Dumping memory of encoded token after decryption step [225 bytes].
00000000: 3A75 7375 7265 3A5C 4D57 524D 6165 6D6C   'u:user\:WMMRealm'
00000010: 432F 3D4E 6F64 696D 6F6E 4120 6D64 6E69   '/CN=domino Admin'
00000020: 396F 7376 6D39 426B 644F 2F68 505A 586E   '9svm92b4of/H/ZA/8N'
00000030: 7468 6C37 364E 6770 3D3D                       'tlh76Ng=='
```

Now that we've decrypted the token we read the values and authenticate the user. These next three lines are typically where you look to determine the problem with SSO, and this is also where error messages are written:

```
06/26/2009 02:08:35.78 PM [3130:000F-2540] HTTP Sessions> Validating single sign-on session cookie (LtpaToken) (<LtpaToken>)
06/26/2009 02:08:35.78 PM [3130:000E-3680] SSO API> -LDAP Realm = WMMRealm
06/26/2009 02:08:35.78 PM [3130:000E-3680] SSO API> -Username = CN=domino Admin/O=ibm
06/26/2009 02:08:35.78 PM [3130:000E-3680] SSO API> -Expiration Ticks = 1246053815666
[06/26/2009 06:03:35 PM].
06/26/2009 02:08:35.78 PM [3130:000E-3680] HTTP Sessions> Decoded single sign-on session cookie, logging in (CN=domino Admin/O=ibm)
06/26/2009 02:08:35.78 PM [3130:000E-3680] HTTP Sessions> Validating single sign-on session cookie (LtpaToken) (<LtpaToken>)
```
So let's look at the cases in which problems can occur, and what the error messages will be.

3 Troubleshooting error messages on the Domino console
There are four errors typically seen on the Domino console:

- **No debug written at all**

- **Unable to decrypt token:**
  07/01/2009 08:56:09.98 PM [12DC:000A-14A0] SSO API> ERROR: when decoding LtpaToken [Single Sign-On token is invalid].

- **Realm values do not match:**
  07/02/2009 08:49:48.70 PM [1634:000B-17E0] SSO API> LDAP Realm does not match config setting [Single Sign-On token is invalid].
  07/02/2009 08:49:48.70 PM [1634:000B-17E0] SSO API> ERROR: when decoding LtpaToken [Single Sign-On token is invalid].

- **Token has expired:**

In addition, the issue may occur in which there are no errors, but you still do not have access to the database.

Let's delve into each of these in the next five subsections.

3.1 **No debug written to console**
If there is no debug written to the console, it means the LtpaToken was not passed by the browser to the Domino server. This is a result of the Internet domain set for the token not being used in the browser's address. For more details on how the domain is used by the browser and what it's used for, refer to Section 2.2 of “Understanding single sign-on (SSO) between IBM® WebSphere® Portal and IBM Lotus® Domino®”.

To fix the issue, check the following settings in the browser address bar:

First, verify that the URLs you are using in the browser can access each server. The URL must be a fully qualified name. In our example the WebSphere Portal server is on a
machine called dpi-dev (see figure 2). If we were to access the WebSphere Portal server at http://dpi-dev/wps/myportal, SSO would not work to our Domino server. Instead, we must use

http://dpi-dev.atlanta.ibm.com/wps/myportal

OR


NOTE: There must be a domain in the URL you use in the browser.

**Figure 2. Example URL**

Second, the domain you use must be the same as what's specified in the WebSphere Application Server Administration console. To check what is in the Administration console:

1. Open a browser to go to the WebSphere Application Server Admin console (for example, https://dpi-dev.atlanta.ibm.com:10041/ibm/console, in our environment).

2. Select Security > Secure administration, applications and infrastructure.

3. Expand Web security, and select Single sign-on (SSO), as shown in figure 3.
4. In the Domain name field in the next screen (see figure 4), specify the domain with which the cookie will be set (.ibm.com, in our environment).
In this case, since the Domain name is set to .ibm.com, we could therefore use dpi-dev.atlanta.ibm.com or dpi-dev.ibm.com as the URL to access WebSphere Portal. Neither Dpi-dev, dpi-dev.com, nor even dpi-dev.lotus.com would work because they do not contain .ibm.com.

If you need to change this value, simply change the setting. Click OK, save the changes, and restart WebSphere Portal for the changes to take effect.

The third place to check whether the token is being set in the browser is the cookies in the browser itself. The easiest way to do this is with Firefox by selecting Tools > Options; in the Options dialog box, click the Privacy tab and select the Show Cookies button (see figure 5).
Click the Remove All Cookies button in the next screen and then sign into WebSphere Portal. You should see the cookies as listed in figure 6.
The key cookie to look for is the one called LtpaToken. In the site or domain you should see the value you specified in the WebSphere Application Server Admin console described earlier. If it's not there, restart WebSphere Portal and run the test again.

Up to this point we've discussed why the token would not be set correctly into the browser. However, it's also possible that the token is set correctly in the browser, but it is not being passed to the Domino server. If this occurs, the only reason would be due to the URL you are using to access the test database.

Let's say the Domino server with which we are testing is on a server with hostname dpi-mail. In our example, the LTPAToken is set with site ibm.com, so the URL used to access the databases on Lotus Domino must also contain ibm.com.

Therefore, we could use http://dpi-mail.atlanta.ibm.com/mail/idsuser1.nsf or http://dpi-mail.ibm.com/mail/idsuser1.nsf; however, http://dpi-mail/mail/idsuser.nsf or http://dpi-mail.lotus.com/mail/idsuser1.nsf would fail, as the token would not be passed from the browser to the mail server.

3.2 Unable to decrypt token passed by WebSphere Portal

The most common issue is the Domino server being unable to decrypt the LtpaToken passed in from WebSphere Portal. When this happens, the following error will display at the end of the SSO debug:
There is only one reason this issue occurs; namely, the key used to encrypt the LtpaToken in WebSphere Portal is different than the one imported into Domino used to decrypt the token.

To fix the issue you must export and import the key file into Domino. Most likely this has already been done, and it seems unlikely it will help to do it again; however, the fact remains that the only way this error can occur is if the tokens are somehow different.

The most common way for the tokens to be different is if you used the Generate Token button in the WebSphere Application Server Admin console. Specifically, if you click Generate Token, then immediately export the key files, the tokens used seem to change on the WebSphere Portal server but are not updated in the key file you export.

The best way to resolve this issue is to restart WebSphere Portal and then follow the detailed steps in Sections 2.1, 2.2, and 2.3 of "Configuring single sign-on (SSO) between IBM WebSphere Portal and IBM Lotus Domino" to export and import the key file into Lotus Domino.

### 3.3 Realm values do not match

Generally, if we are able to open the token, SSO works; however, if it fails, it is usually due to the realm values not matching in the token. When the realm values are different, you will see the following error at the end of the debug:

```
07/02/2009 08:49:48.70 PM [1634:000B-17E0] SSO API> -Raw Token Username = uid=idsuser1, cn=users, dc=ibm, dc=com
07/02/2009 08:49:48.70 PM [1634:000B-17E0] SSO API> -LDAP Realm = null
07/02/2009 08:49:48.70 PM [1634:000B-17E0] SSO API> LDAP Realm does not match config setting [Single Sign-On token is invalid].
07/02/2009 08:49:48.70 PM [1634:000B-17E0] SSO API> ERROR: when decoding LtpaToken [Single Sign-On token is invalid].
```

If you step back in the debug just a bit, and look at everything written after “Dumping memory of encoded token after decryption step [255 bytes]”, you can see exactly what the issue is and how to fix it.

Just under the “Dumping memory of encoded token...” line in the debug you can see exactly how WebSphere Portal is setting the realm. It appears in the last column after
Before the DN passed in the token. In the example below, this is a WebSphere Portal 6.1 server on which security was enabled with realm support, so the realm is

```
defaultWIMFileBasedRealm```

At the end of the debug, we write what the Domino server read from the WebSSO document. In this case we read that the realm was null:

```
> 07/02/2009 08:49:48.68 PM [1634:000B-17E0] SSO API> Dumping memory of encoded token after decryption step [255 bytes].
00000000: 3A75 7375 7265 3A5C 6564 6166 6C75 5774  'u:user:
00000010: 4D49 6946 656C 6142 6573 5264 6165 6D6C  'IMFileBasedRealm'
00000020: 752F 6469 693D 7364 7375 7265 2C31 6E63 '/uid=idsuser1,cn'
00000030: 753D 6573 7264 2C3D 6329 2C6D 6364 633D '=users,dc=ibm,dc=com'
00000040: 633D 6D6F 3125 3432 3536 3938 3634 3232 '=com%12465894622'
00000050: 3138 4L25 4D54 4D2C 4C65 61 4E66 766b 81%LEMldjtyeiZFO'
00000060: 7361 3156 3668 2F4D 4F41 6867 5145 4463 316SfOZVh68M/AO'
00000070: 744A 3D 3465 4B6K 5W6c8ywjR'
00000080: 7361 752F 6468 5839 4C25 4D54 4D2C 4C65 31%LEMldjtyeiZFO'
00000090: 4A 3D 752F 6469 693D 7364 7375 7265 5W6c8ywjR'
000000A0: 766B 4365 7A4F 6B56 4D38 3863 7779 526A  'kveCOzVk8Mc8ywjR'
000000B0: 6350 4636 6A62 5978 346F 384B 5A6E 5034  'PccMbixYo4K8nZ4P'
000000C0: 6479 5534 7252 7974 5534 314E 4A30 3063  'yd4URrtYHUN10Jc0'
000000D0: 5064 4947 5334 7062 5444 5736 726D 7459  'dPGI4SbpDT6WmrYt'
000000E0: 4669 466B 7568 5839 6339 4334 4B65 576C  'iFkFhu9X9c4CkKIW'
000000F0: 4153 5450 7133 6969 3349 6D62 564L 3D  'SAPT3qiil3bmtw='
07/02/2009 08:49:48.70 PM [1634:000B-17E0] SSO API> -Raw Token Username = uid=idsuser1,dc=ibm,dc=com
07/02/2009 08:49:48.70 PM [1634:000B-17E0] SSO API> -LDAP Realm = null
07/02/2009 08:49:48.70 PM [1634:000B-17E0] SSO API> LDAP Realm does not match config setting [Single Sign-On token is invalid].
07/02/2009 08:49:48.70 PM [1634:000B-17E0] SSO API> ERROR: when decoding LtpaToken [Single Sign-On token is invalid].
07/02/2009 08:49:48.70 PM [1634:000B-17E0] SSO API> *** Freeing Single Sign-On Token List (SECTokenListFree) ***
07/02/2009 08:49:48.70 PM [1634:000B-17E0] SSO API> *** Freeing Single Sign-On Token (SECTokenFree) ***
```

As you can see, defaultWIMFileBasedRealm does not match null, so the token is invalidated. You have a few options to fix this issue. The easiest is simply to open the WebSSO configuration document and manually update the realm value to match what's passed in from WebSphere Portal.
In this case, let's simply edit the SSO document we're using and set the LDAP Realm field as defaultWIMFileBasedRealm (see figure 7).

**Figure 7. LDAP Realm field**

<table>
<thead>
<tr>
<th>Web SSO Configuration for: ltpa1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Basics</strong></td>
</tr>
<tr>
<td><strong>Token Configuration</strong></td>
</tr>
<tr>
<td>Configuration Name:</td>
</tr>
<tr>
<td>Organization:</td>
</tr>
<tr>
<td>DNS Domain:</td>
</tr>
<tr>
<td>Map names in LTPA tokens:</td>
</tr>
<tr>
<td><strong>Participating Servers</strong></td>
</tr>
<tr>
<td>Domino Server Names:</td>
</tr>
<tr>
<td><strong>WebSphere Information</strong></td>
</tr>
<tr>
<td>Token Format:</td>
</tr>
<tr>
<td>LDAP Realm:</td>
</tr>
<tr>
<td>LTPA Version:</td>
</tr>
</tbody>
</table>

In some configurations of WebSphere Portal, it is also possible to change the realm value that it sets. For more details on how to do this and to understand what the Realm value is likely to be set to, see Section 3.3.1 of "Understanding single sign-on (SSO) between IBM® WebSphere® Portal and IBM Lotus® Domino®."

### 3.4 Token has expired

The final error you are likely to see when testing SSO between WebSphere Portal and Lotus Domino is that the token has expired. Typically you would only see this error if the time specified in WebSphere Portal has expired since you signed in (2 hours by default). However, this can happen when testing and it's only been a few minutes, if setting up SSO for the first time.

On the Domino console, you'll see the following at the end of the SSO debug:

```
07/02/2009 09:59:31.24 PM [1048:000A-1174] SSO API> -Raw Token Username = uid=idsuser1, cn=users, dc=ibm, dc=com
```

19
From the above errors, you can easily see that the WebSphere Portal server set the token to expire at 8:58 PM (Expiration Ticks = 1246593530666 [07/02/2009 08:58:50 PM]); however, the current time on the Domino server is 9:59 (07/02/2009 09:59:31.26 PM), a full hour after the token expired.

Typically, this error occurs if the time set on the OS is incorrect. For example, if the time on the WebSphere Portal server is set to 1:00, but the time on the Domino server is set to 4:00, this problem would occur.

The WebSphere Portal server would set a timeout of 2 hours after you signed in, 3:00. When the Domino server opens the token, it only knows the time when the token expires, 3:00, and compares that to the current time on the Domino server. Because it is an hour after the token was set to expire, Domino invalidates the token.

Another case in which this issue occurs is if the time zones are different between the two servers. Many times we look at the time on both servers, and they are both set to 3:00, but SSO continues to fail. However, when looking more closely at the times on both servers we see that the WebSphere Portal server is set to EST (-5 GMT), but the Domino server is set to PST (-8 GMT).

What happens in this case is that you may sign into WebSphere Portal at 3:00 EST, and WebSphere Portal sets a timeout in the token of 5:00 EST. When you switch to the Domino server, Lotus Domino opens the token and calculates the timeout, which is 2:00 PST (5:00 EST = 2:00 PST). So when the Domino server compares 2:00 PST to the current time, 3:00 PST, the token is invalidated. In most cases you can resolve the issue by simply setting both systems to EST and to the same time.

If the Domino server is actually located in PST and WebSphere Portal is in EST, then that is fine; the actual time on both servers just needs to be correct. So in this example, if it's 3:00 EST, then the time on the Domino server should be 12:00 PST. As long as the date, time, and time zone are correct, the issue will be resolved.

If you need to correct the time or time zone, shut down the WebSphere Portal or Domino server, fix the time on the operating system, and then start the server and test again.
3.5 No errors in Domino console, but SSO continues to fail

It's very rare, but every once in a while you may see the situation in which the SSO debug is exactly correct, but you are still unable to access the test database user. In this case, the SSO logs are not very helpful; they simply show you the user's DN that was passed through the LtpaToken:

08/02/2009 08:57:02.63 PM [0744:0012-0E14] SSO API> -Raw Token Username = uid=idsuser1,cn=users,dc=ibm,dc=com
08/02/2009 08:57:02.63 PM [0744:0012-0E14] SSO API> -LDAP Realm       = defaultWIMFileBasedRealm
08/02/2009 08:57:02.65 PM [0744:0012-0E14] SSO API> -Username         = uid=idsuser1/cn=users/dc=ibm/dc=com
08/02/2009 08:57:02.65 PM [0744:0012-0E14] SSO API> -Raw Token Username = uid=idsuser1/cn=users/dc=ibm/dc=com
08/02/2009 08:57:02.65 PM [0744:0012-0E14] SSO API> -Expiration Ticks = 1249263072666 [08/02/2009 09:31:12 PM].
08/02/2009 08:57:02.65 PM [0744:0012-0E14] HTTP Sessions> Decoded single sign-on session cookie, logging in (uid=idsuser1/cn=users/dc=ibm/dc=com)
08/02/2009 08:57:02.65 PM [0744:0012-0E14] SSO API> *** Freeing Single Sign-On Token List (SECTokenListFree) ***
08/02/2009 08:57:02.65 PM [0744:0012-0E14] SSO API> *** Freeing Single Sign-On Token (SECTokenFree) ***
08/02/2009 08:57:02.65 PM [0744:0012-0E14] HTTP Sessions> Looking for single sign-on session cookie in session cache
(nHrxIukUwZzcSRjNtrVfThV0OgZri8jYfI8Ii6C5c8X1PEjN1AY27dubszE3ywJG6k2VD7rt1C2h1uHbSbQRoSk6NHlz8mkliwJ4sa/ tg5WTXaex6WXMneCLnS6py3yXA7CdE1uV

To troubleshoot the issue we need to compare the name passed in from the token with the name of the user in the ACL of the database you are testing. If both directories are authenticating with Lotus Domino, then this step is quite straightforward, and the name from the LtpaToken will be something like the following:

08/02/2009 08:57:02.65 PM [0744:0012-0E14] SSO API> -Username         = uid=idsuser1/cn=users/dc=ibm/dc=com

When you compare this to the ACL of the database, it will be in the Domino canonical format, so you will see “Ids User1/ibm” (see figure 8).
If you have access to this database via a group, then you should first add the user explicitly, and make sure it works correctly that way. Once that works, remove the user, and look at the member list of the group on the server you are testing. Ensure the user is listed as a member of the group using the full Domino canonical format (ids User1/ibm).

The more likely case, in which SSO debug works correctly, but you do not get access to the database, occurs with dual-directory environments in which WebSphere Portal authenticates with a non-Domino LDAP directory.

In this case, something was not completed correctly in mapping the directories. Use the SSO debug to find out exactly what the WebSphere Portal server is setting in the LtpaToken, and then use this value to synchronize the directories, as explained in Section 2.5 of “Configuring single sign-on (SSO) between IBM WebSphere Portal and IBM Lotus Domino.”
4 Conclusion

At this point, after reading the three articles in this series on single sign-on—“Understanding,” “Configuring,” and now “Troubleshooting”—you likely know more than you ever wanted to know about SSO. Most importantly, though, you are now able to quickly configure SSO between your WebSphere Portal and Lotus Domino environments, as well as enable debug and pinpoint problems in SSO, no matter the five possible issues:

- No debug written at all
- Unable to decrypt token
- Realm values do not match
- Token has expired
- No errors but you are still unable to access the database

5 Resources

developerWorks white paper, “Understanding single sign-on (SSO) between IBM WebSphere Portal and IBM Lotus Domino.”

developerWorks white paper, “Configuring single sign-on (SSO) between IBM WebSphere Portal and IBM Lotus Domino.”

Knowledge Base document #1158269, “Troubleshooting WebSphere Portal, Domino Extended Products, and Domino SSO issues”:
http://www.ibm.com/support/docview.wss?rs=899&uid=swg21158269

developerWorks Lotus Notes and Domino product page:

developerWorks WebSphere product page:

Lotus Notes and Domino wiki:
http://www-10.lotus.com/ldd/dominowiki.nsf

WebSphere Portal family wiki:
http://www-10.lotus.com/ldd/portalwiki.nsf

developerWorks article, “How to configure SSO with LTPA for IBM Lotus Connections 2.0”:
6 About the author

Charlie Price is an Advisory Software Engineer in IBM's Software Group. He has six years of experience in technical support for IBM Lotus software, and two years in the test organization, specializing in cross-product integration with Lotus, IBM, and other third-party products.

He is an IBM Certified Associate System Administrator - Lotus Collaborative Solutions (administering QuickPlace®), a Principal Certified Lotus Professional for Domino system administration, and an IBM Certified System Administrator for WebSphere Portal. He holds a degree in Mathematics Education from the University of Georgia and taught high school mathematics for three years before joining IBM. You can reach him at charles_price@us.ibm.com.

Trademarks

- Domino, IBM, Lotus, Notes, and WebSphere are trademarks or registered trademarks of IBM Corporation in the United States, other countries, or both.

- Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

- Other company, product, and service names may be trademarks or service marks of others.