

Mastering your Enterprise with Lotus Domino

Author:

iCore effective GmbH

Can Uenal

Tel: +49 6151 360780

Email: can.uenal@e-core.com

© Copyright iCore effective / Proprietary Information / All Rights reserved. All rights reserved. No part of this publication may be reproduced and/or published, stored in an information retrieval system or transmitted in any form or by any means, print, photo print, microfilm, audiotape, electronic, mechanical or otherwise, without the prior written permission of the copyright owner. The information in this document is subject to change without notice and organization unity not to be considered as a commitment of iCore effective. No responsibility is accepted for any error, which may appear in this document.

Domino6 LDAP	4
How an LDAP object class relates to a Domino form	4
How an LDAP attribute relates to a Domino field	5
LDAP-standard attributes on Domino forms	5
Integration of MS Active Directory & Lotus Domino	7
Creating users and groups in Active Directory	7
WebSphere Application Server 5.1	12
Global Security	12
WebSphere Administrative roles	13
Creation of the LTPA Token	14
Creating a Domino Single Sign On (SSO) Document	17
WebSphere Portal & Domino 6.5.3 LDAP	21
w pconfig.properties	23
Enable LDAP security for WebSphere Portal	23

Domino6 LDAP

In most companies where IBM Lotus applications are employed, end-users generally have at least 3-4 separate usernames and passwords for the Windows NT workgroup or domain, their LDAP directory, and Lotus Domino Web applications. As a result, the system administrators face a number of tasks related to password management, such as resetting passwords in several places because end-users forget their passwords, synchronizing several sets of password quality rules that may or may not overlap, and creating and disabling accounts in multiple places when someone joins or leaves the organization.

The default Domino LDAP schema includes:

- Domino-specific schema elements defined by the default forms in the Domino Directory
- All LDAP-standard schema elements defined in RFCs 2252, 2256, 2798, 2247, and 2739. The LDAP service uses the file LSHEMA.LDIF to build these elements in the default schema.

You can extend the schema to add custom schema elements that your organization needs.

To see detailed information about the Domino LDAP schema, open the Domino LDAP Schema database (SCHEMA.NSF) on any server that runs the LDAP service.

Name	Value	Type	Size
cn	cn ouenal	text attribute	10
mail	cn_ouenal@icore-icore.com	text attribute	36
objectclass	dominoPerson	text attribute	12
objectclass	inetOrgPerson	text attribute	13
objectclass	organizationalPerson	text attribute	20
objectclass	person	text attribute	6
objectclass	top	text attribute	3
dominocertificate	30 33 36 30 32 46 30 32 20 31 44 33 39 42 46 43	binary attribute	1194
givenname	cn	text attribute	3
sn	ouenal	text attribute	6
uid	ouenal	text attribute	7
maldomain	icore	text attribute	5
mailserver	CN=ouenal7,O=icore	text attribute	22
mailfile	mail/couenal	text attribute	12
creatorname	CN=ouenal7,O=icore	operational attribute	22
modifiersname	CN=ouenal7,O=icore	operational attribute	23
createtimestamp	20050307075807Z	operational attribute	15
modifystamp	20050307075808Z	operational attribute	15
subschemasubentry	ou=schema	operational attribute	9
dominouid	C70815C6A4H76E8DC1256FB0002BC62F	operational attribute	32

Pic 1: Domino LDAP Schema (v. R7)

How an LDAP object class relates to a Domino form

An LDAP object class is similar to a form in the Domino Directory, in that each defines a set of information for a directory entry. A Domino-specific object class -- whose name usually begins with *domino* -- always maps to a form in the Domino Directory. For example, the

object class *dominoPerson* maps to the form *Person*, and the object class *dominoGroup* maps to the form *Group*.

An object class that is not specific to Domino, for example a standard LDAP object class defined in the *LSHEMA.LDIF* file, *maps to a form only if you create such a form*

For example, the object class *residentialPerson* is part of the default Domino LDAP schema, but it has no corresponding form in the Domino Directory. Therefore by default you can use only LDAP operations to add, search, and modify, *residentialPerson* entries. To give Notes and Web users access to these entries, you must you create a corresponding form following a specific procedure. If you create a corresponding form, *residentialPerson* entries are created as documents that are visible to Notes and Web users.

How an LDAP attribute relates to a Domino field

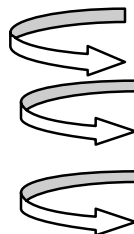
An LDAP attribute is similar to a field in the Domino Directory in that each define a piece of information about a directory entry. An LDAP attribute defined for a Domino-specific object class always maps to a field in a form in the Domino Directory. The name of the attribute and the name of the field may not be identical. This difference occurs when a preexisting field in Domino has a purpose similar to an LDAP-standard attribute.

LDAP	Domino
uid →	ShortName

LDAP-standard attributes on Domino forms

If a Domino object class inherits from an LDAP-standard object class, the fields that represent the inherited attributes may be hidden in the Domino Directory document.

For example, the *dominoPerson* object class inherits the attribute *employeeNumber* from the LDAP-standard object class *inetOrgPerson*. On the other hand accessing data from Lotus Domino the object class *dominoPerson* is the “content provider”.



LDAP	Domino Form
Top	-
Person	-
OrganizationalPerson	-
inetOrgPerson	-
dominoPerson	Person

LDAP SCHEMA - Object Class: dominoPerson			
Names		Object Classes	
LDAP name:	dominoPerson	Object Class Type:	Structural Object Class
OID:	2.16.840.1.113078.2.2.2.1.1	Superior Object Class:	inetOrgPerson
Notes mapping:	Person	Auxiliary Object Classes:	
Schema:	Lotus Domino Directory	Description:	
Attribute Types			
Mandatory Attribute Types:	cn objectClass sn	Optional Attribute Types:	AllFullname AllFullnameLanguage AllFullnameLanguageDisplay AllFullnameSort audio AvailableForDirSync BinaryFile businessCategory c CalendarDomain caL_course ccMailLocation ccMailUserName CertificateDisplay ChangeRequest CheckPassword Children

Pic 2: LDAP entry for the Object Class "dominoPerson"

There are some syntaxes in the default Domino LDAP schema that map to Domino field types. For example, the LDAP syntax *Integer* maps to the field type Number. To see whether a syntax maps to a Domino field, find the document for the syntax in the Schema database (SCHEMA.NSF), and compare the LDAP name field to the Notes mapping field.

Integration of MS Active Directory & Lotus Domino

Domino administrators working in a Windows 2000 environment with ActiveDirectory can now administer users and groups from a single administrative interface of their choice:

A new feature of the Domino 6 server, **ADSync** enables the administrators to integrate both LDAP servers without having to manually update both with changes.

This synchronization feature allows a Domino administrator to securely and precisely delegate the responsibility for Domino user and group management to the network administrators who manage these details in Active Directory.

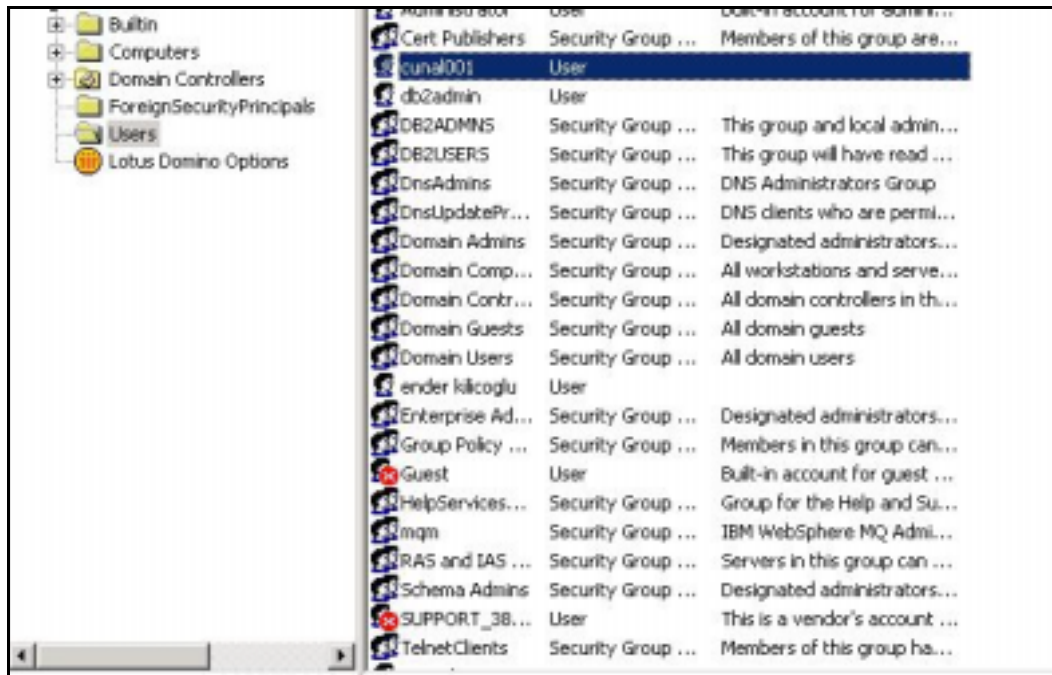
An Administrator can create new users and groups in Active Directory and have those changes reflected in the Domino Directory, including the creation of person or group documents, Notes IDs, passwords, and mail files for the users. In order to accomplish these tasks, the Active Directory administrator must have a properly certified Notes ID and appropriate access to make changes in the Domino Directory.

The registration server must be Domino 6 or later and the Domino Administration client must be a 6 or later client. Additionally, policies must be created that contain subpolicies, either implicit or explicit, for all Domino certifiers where users will be created. Finally, you must have the appropriate rights in Active Directory to add users and groups, and synchronize passwords.

Creating users and groups in Active Directory

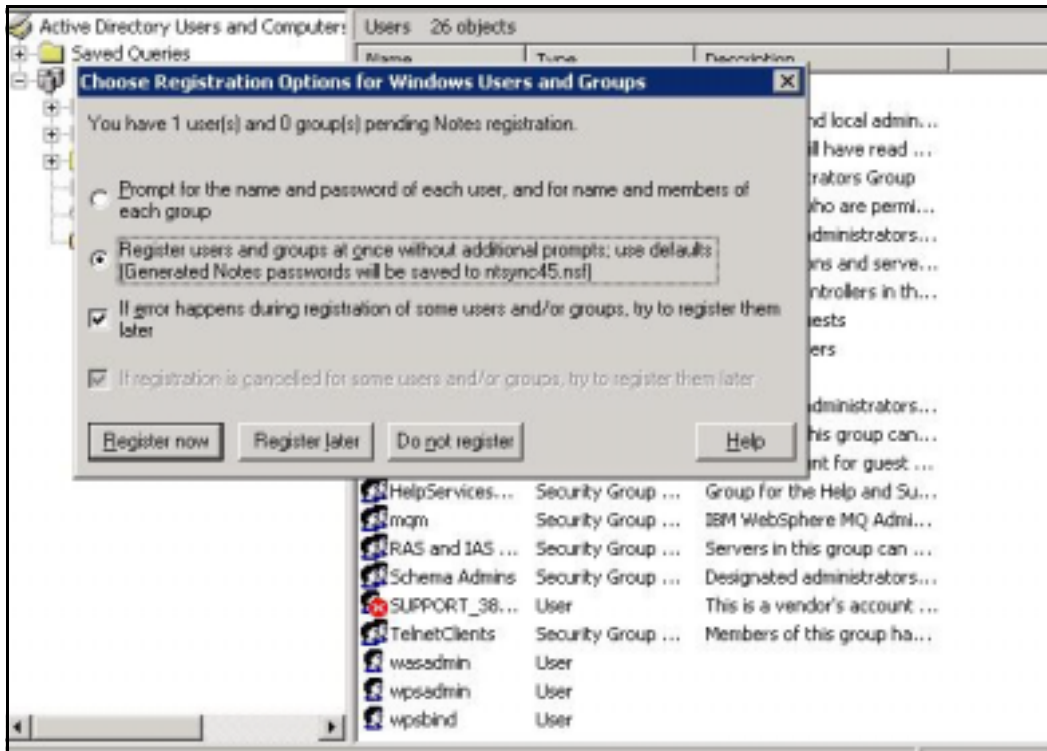
To access Active Directory Users and Computers from your Windows workstation click **Start -> Programs -> Administrative Tools -> Active Directory Users and Computers**. You may initiate Active Directory "actions" in the right-hand results pane, or in the left-hand navigation pane. Domino users and groups are created by either of two methods:

1. In the left pane, right-click an entry and choose your action from the pop-up menu.
2. In the results pane, select one or more users and groups, then select "Register in Domino" from either the context menu, the toolbar, or by right-clicking the entry and using the pop-up menu.



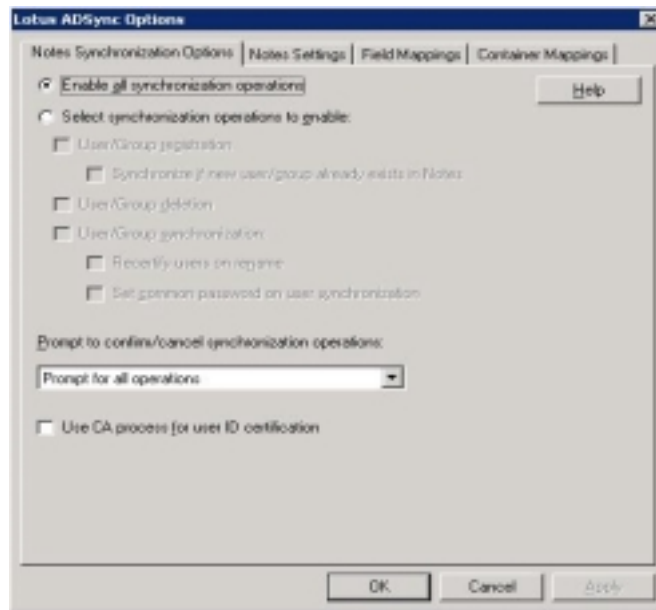
pic3: Integration of Domino and Microsoft Active Directory

Before starting registering users and groups from Active Directory, the Lotus Domino option must be enabled.



pic 4: registering an existing user ("db2admin") to Lotus Domino

With ADSync initialization complete, several synchronization options, can be chosen as shown in the next four windows.



pic 5: Options to configure Domino with ADS

From the Notes Synchronization Optionstab you can:

- Enable or disable all synchronization operations
- Customize synchronization options with "Select synchronization operations to enable."
- Configure prompting options from the drop-down selection box
- Choose to use the CA process for user registration

On the Notes Settings tab you can specify:

- Registration server (which Domino server will be used for registration)
- Administration ID (which user ID will have administrative privileges)
- User deletion options (From the drop-down selection box, choose which actions should take place when a user is deleted.)
- Default certifier and policy
- Group type mappings

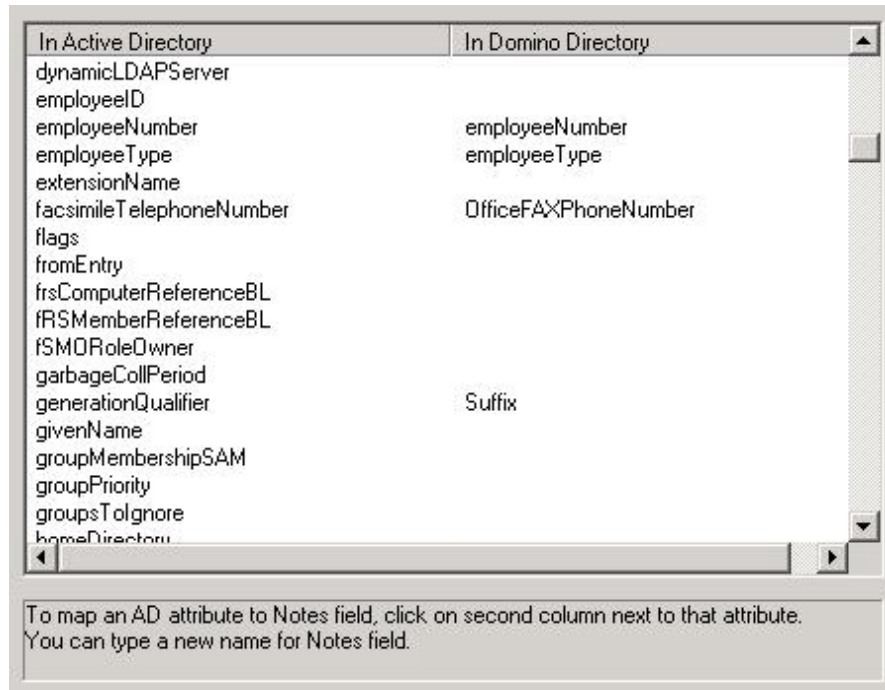


Fig 6: Field mappings between ADS and Lotus Domino

The Field Mappings tab is where you select which Active Directory fields are to be mapped to Domino Directory fields. During ADSync tool initialization, the schemas from Active Directory and Domino are mapped based on default settings. If additional field mappings are needed, left-click in the right column under “In Domino Directory” and a drop-down selection box with Domino directory fields is presented.

The Container Mappings tab is where Active Directory containers are mapped to Notes Certifiers and Policies. Active Directory containers are a special class that has both a namespace and attributes. The container does not represent anything real or concrete, but rather holds one or more objects.

Objects, on the other hand, are the underlying principle of everything in the Active Directory. Servers, workstations, printers, users, documents, and devices all represent objects. Each object has its own access control list (ACL) and attributes.

By design, the synchronization tool allows to preserve the hierarchies in Active Directory and Domino using mapping. An extended ACL is an optional directory access control feature available for the Domino Directory, an Extended Directory Catalog, and the Administration Requests database.

Websphere Application Server 5.1

There are several mechanisms to combine a Domino Server and Websphere Application Server (WAS). Within this paper Single-Sign-On (SSO) for a common authentication of a mixed Domino and WAS environment and the Domino DSAPI for using the Domino HTTP stack by WAS will be introduced.

WebSphere, and indeed J2EE, do not implicitly provide a secure means of communication but rather rely on an additional service, typically a transport-layer digital encryption algorithm, called *Secure Sockets Layer (SSL)* and *Transport Layer Security (TLS)*.

In this chapter the WebSphere Application Server (WAS) security will be introduced and based on the WebSphere Security model the WAS environment will be configured for the LDAP use. For this step the required actions are:

1. enabling the global security of WAS
2. creation of the LTPA Token
3. configuring WAS for LDAP use

Global Security

The Security section is the focal point for the configuration of WebSphere security. It is accessible from the Admin Console. After logging in, click the **Security** link in the navigation pane. WebSphere security can be enabled and disabled in its entirety by selecting a single switch. This is the Global Security *Enabled* switch which is accessible from the Administrative Console under **Security -> Global Security**.

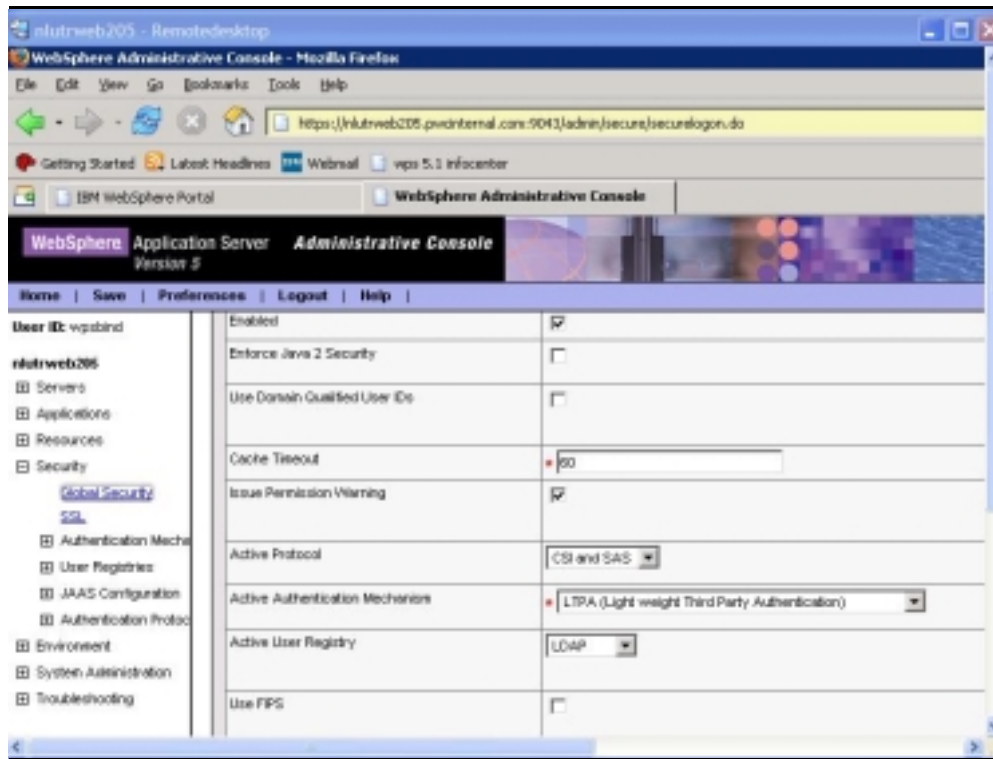


Fig 7: Global Security of WAS

To enable Global Security, certain criteria must be met.

- An authentication mechanism must be selected. The Application Server supports two authentication mechanisms by default, SWAM and LTPA.
- A user registry must be selected. The Application Server supports the concept of a custom registry, which makes the integration of WebSphere with any type of appropriate registry fairly straightforward. LocalOS and LDAP are the two types of registry provided by default and LocalOS is selected initially.

The J2EE role-based authorization concept has been extended to protect the WebSphere Administrative subsystem. Four roles are defined for performing administrative tasks

WebSphere Administrative roles

The identity that is specified when enabling Global Security is automatically mapped to the Administrator role. Therefore, it is not necessary to manually add this identity to the administrator role.

Users and groups, as defined by the user registry, may be mapped to administrative roles. To enable a new mapping, it is necessary to save the changes to the master configuration

and restart the server. For this reason, it is advisable to map groups to administrative roles so that users may be added to the groups appropriately (and hence the users are mapped to administrative roles) without the need to restart the WebSphere server.

Creation of the LTPA Token

IBM Lightweight Third Party Authentication (LTPA) tokens, or cookies, provide a means to share authentication information between Lotus, WebSphere and Tivoli application (Web) servers. A user who has been authenticated once by an application server will be automatically authenticated on other application servers in the same DNS domain providing the LTPA keys have been shared by all the applications. LTPA utilizes a token which is stored as a cookie in the user's browser.

The LTPA token contains data that uniquely identifies the user, such as the user's Distinguished Name (DN), and an expiration date that effectively limits the session time before the user is forced to re-authenticate. Special notes related to use of LTPA include:

- All application servers using LTPA tokens must reside in the same DNS domain.
- All application servers must share the same user registry (LDAP directory). Supported directories include Lotus Domino (configured as an LDAP directory), IBM Directory Server, MS Active Directory, and iPlanet.
- Browsers accessing application servers must be configured to accept cookies, which are used to store a token containing authentication information.
- Optionally, SSO may be set up to work only on encrypted HTTPS connections.
- LTPA is an IBM-specific solution; other application server vendors provide limited (or no) support.

With LTPA as the authentication mechanism, a trusted third party server is used to authenticate the user. Depending on whether a token has already been issued to the user, there are two possible actions a Web server might perform. The two actions or mechanisms are:

1. Creation (encoding) the LTPA token by the initial server the user logs into
2. Interrogation (decoding) of an LTPA token provided by the browser in the HTTP request to a server

Domino provides a cryptographic token-based mechanism to provide single sign-on support between protocols such as HTTP and IIOP, and also with the IBM WebSphere application server. The servers that participate in single sign-on use an encrypted "Web SSO Configuration" to share secret data in the Domino Directory used for generating and validating single sign-on tokens.

Note: In an environment with WebSphere and Lotus or Tivoli products, or both, the LTPA keys must be generated by WebSphere and imported into the other products

- When interoperability with WebSphere is not required, Domino uses its own format for the Single Sign-On token that is slightly different from the one implemented by WebSphere.

If you are using single sign-on, you will want to generate a set of keys for export. These keys are intended to be imported to the other servers participating in single sign-on.

The following steps show you how to generate the LTPA keys for the Express Application Server:

1. In the LTPA Configuration panel, click the button **Generate Keys**. This will launch the key generation process in the background. You will be prompted to save the configuration after the process is completed.
2. Save the configuration to store the generated keys stored in the WebSphere configuration; they will appear in the security.xml file.
3. Re-open the LTPA configuration page.
4. Specify the Key File Name which is the name of the file where LTPA keys will be stored when you export them. You need to export the keys in order to enable Single Sign-On on another server. Specify the full path name for the key file. We have used c:\WebSphere\Appserver\LTPAkeys. This file will be created for you if it doesn't exist.

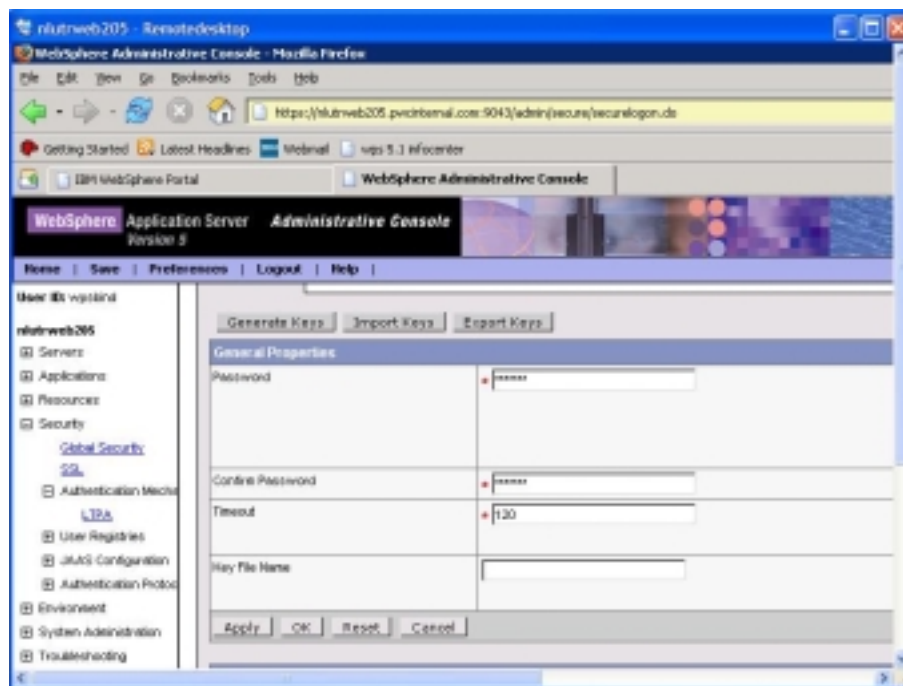


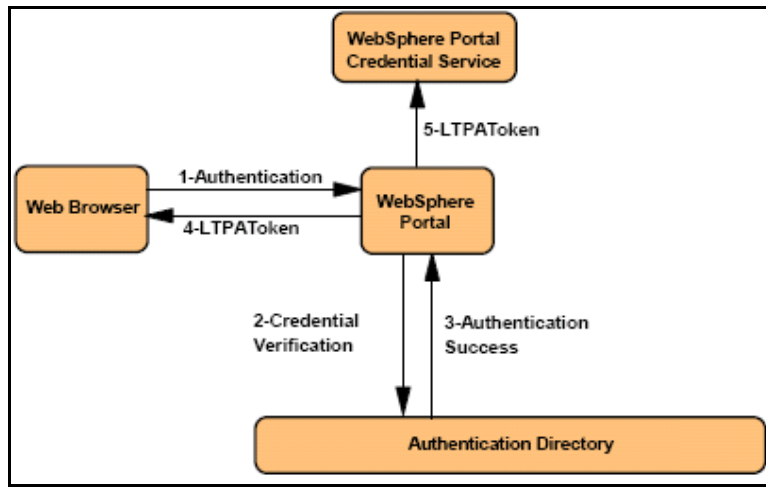
Fig 8: creating a LTPA Token with Websphere Application Server

To configure WebSphere to use Domino as its user registry, follow the steps below.

1. Start the WebSphere Administrator's Console.
2. Expand the tree **Security -> User Registries -> LDAP**. You will see the LDAP configuration panel open in the main window.
3. Fill in the following configuration settings
 - Server User ID: this field must contain the value specified in the Short Name/User ID field in the Person Document of the Domino Directory created in the steps above for the WebSphere administrator; this is the user ID that will have to be used for login to start the WebSphere Administrator's Console once security is enabled, for example: wasadmin.
 - Server User Password: enter the Internet password set for the wasadmin user in this document.
 - Type: Domino
 - Host: name for the Domino (directory) server, for example: dominosrv
 - Port: 389
 - Base Distinguished Name: this is the base distinguished name of the directory service, indicating the starting point for LDAP searches of the directory service. As we defined all our users and groups under /pwcinternal, we have entered o=pwcinternal for this field.
 - Bind Distinguish Name: distinguished name used when WebSphere binds with the Domino server. If no name is specified, the administration server will bind anonymously.
 - Bind Password: if a user is specified in the Bind Distinguished name, include the corresponding password here.

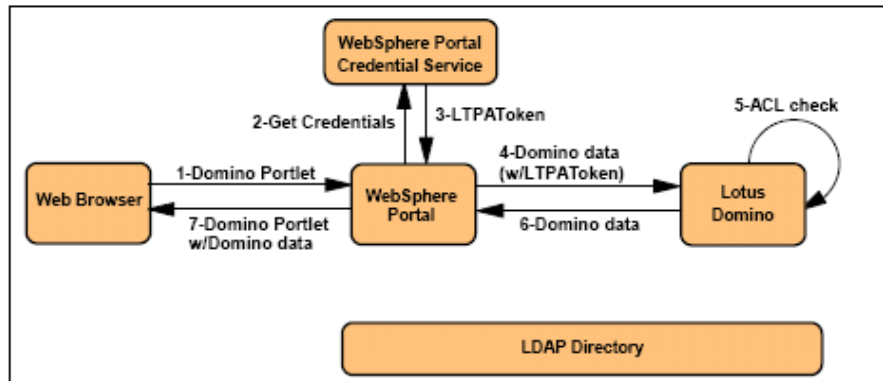
Creating a Domino Single Sign On (SSO) Document

Domino 6 can have different SSO configurations for different services (Web, POP, etc.), even on the same server, using Internet sites. However, configuring SSO in a Domino 5/6 mixed environment, potential problems can occur because R5 does not recognize the Internet Site documents. The next picture shows the interaction between the different applications and the related tasks:



Pic 9: Browser/Portal interaction with LTPA SSO

1. In “1-Authentication”, the user makes a request to the portal and provides a set of authentication credentials
2. The portal server then verifies the credentials (2),
3. and assuming successful authentication (3), it creates an LTPA token.
4. This LTPA token is then not only sent back to the client browser (4),
5. but is also placed into the Portal’s credential service (5).

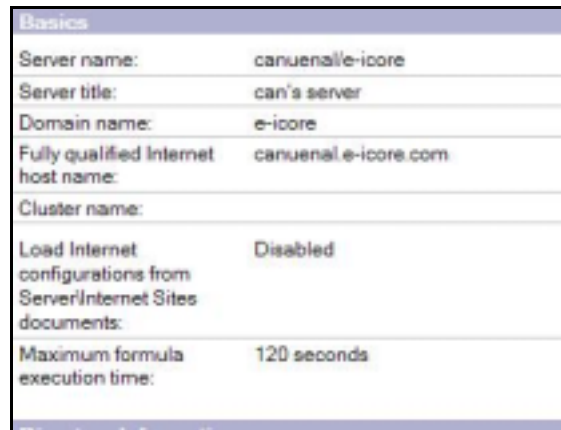


Pic 10: Portal/Domino interaction with LTPA SSO

1. In this set of interactions, the browser-based client requests a Domino portlet from the portal server (1).
2. The portal server knows that it must access Domino on behalf of the user to get data for the portlet, so it goes to its credential server and fetches the LTPA token it cached for the user at original login (2 & 3).
3. The portal then sends the request to Domino with the LTPA token (4).
4. Domino would trust the LTPA token and perform an ACL check on the requested resource based on the user's name in the LTPA token (5).
5. Assuming the user is authorized, Domino would send the data back to the portal server (6), which would then render the data back to the user as part of the originally requested portlet. Note that no communication takes place to the authentication server in this interaction. However, this assumes that the user's name is directly listed in the ACL, with perhaps Domino name mapping enabled. If the ACL contains groups for which membership must be verified, then some communication with the authentication server would take place.

As Domino 6 still supports the “R5 Web config,” there is still a possibility to enable SSO in a mixed release environment. The important points are:

- Make sure the Domino 6 servers have Internet sites disabled on the server document Basicstab.



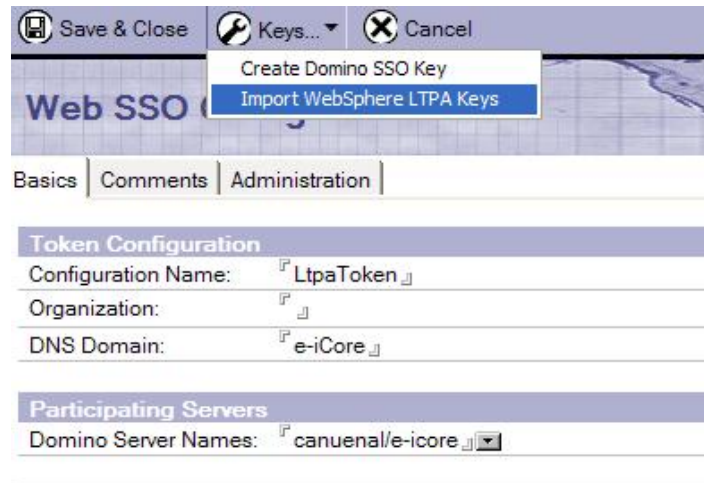
Pic 11: Domino Server configuration

- Create the SSO config doc using the Domino admin client and opening one of the server documents. From the action bar, select “Create Web (R5)...” then “SSO Configuration”, and call it “LTPAToken.”



Pic 12: creation of a Lotus Domino SSO document

- Do not use an Organization name on the Web SSO config doc (this field is only used to support Internet Sites). If you do, the SSO doc will not be visible from the server Internet protocol tab.



Pic 13: imparting the WAS LTPA with Domino

WebSphere Portal & Domino 6.5.3 LDAP

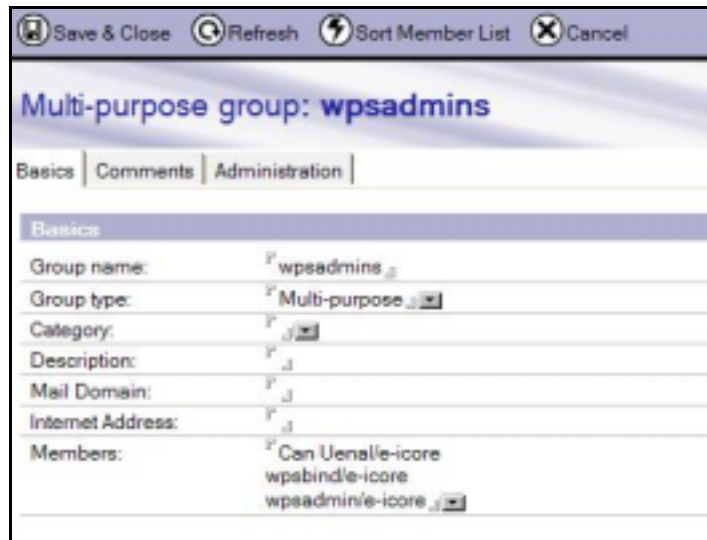
There are several mechanisms for configuring WPS with Domino. Before you can configure WebSphere Portal to work with any LDAP directory, the directory must have some specific user and group information already populated.

- A minimum of one group and one user is required for WebSphere Portal.

Depending on the software you already have deployed and configured, you may need to set up to two additional user accounts. These can either be existing user accounts that you want to use in WebSphere Portal, or you can create new user accounts.

The required group is **wpsadmins** or an equivalent. This is the first administrator group for WebSphere Portal. Members of this group have administrative authority within WebSphere Portal. It is expected that the first WebSphere Portal administrative user, be a member of the **wpsadmins** group in the directory. The following points describe the one required and two possibly needed user accounts:

- **Required.** WebSphere Portal administrative user. This is the first administrator account for WebSphere Portal. This account is also a member of the **wpsadmins** group and is typically called **wpsadmin**.
- **Optional:** Websphere Portal server is an application that runs on top of Websphere Application Server. Websphere Application Server handles the security for Portal, however you can choose to have WebSphere Portal configure WebSphere Application Server security for you. In this case a Security Server ID with an account name and password must be specified. This account is configured into WebSphere Application Server. It becomes the ID that is used to administer WebSphere Application Server. During my testing, I have created a user account called **wpsbind**.
- **Optional:** An LDAP access account for WebSphere Application Server and, by extension, WebSphere Portal. This identity is used by WebSphere Portal to access the LDAP directory. As you will see in next chapter, both LDAP directory and security configuration for Websphere Portal involves modifying values in the **wpconfig.properties** file. If you keep the default values for the "Bind Distinguished Name" in this properties file, the user name "**wpsbind**" will be used for this LDAP access account.



Pic 14: required group (wpsadmins) and the administrative users

The WebSphere Portal administrative user is wpsadmin, and the LDAP access account and the WebSphere Application Server administrative user is wpabind.

Important: In pic16 the group wpsadmins has been manually edited to be wpsadmins/e-icore. In other words, it now has a fully distinguished LDAP name of *cn=wpsadmins/o=e-icore*.

This change has to be made when using a Domino LDAP directory, because Domino does not store groups in the hierarchical form that WebSphere Portal expects.

wpconfig.properties

The wpconfig.properties file that is used to configure WebSphere Portal for an LDAP directory requires that certain password information be specified in it.

```
WPSconfig validate-ldap -DPortalAdminPwd=password -  
DLDAPAdminPwd=password  
-DLDAPBindPassword=password -DWasPassword=password -  
DLTPAPassword=password
```

The values in bold represent the various LDAP user entries that both WebSphere Application Server and WebSphere Portal use for security.

These entries are as follows:

- **PortalAdminPwd**: The password assigned to the WebSphere Portal administrative user (in our case, the user wpsadmin).
- **LDAPAdminPwd**: The password assigned to the LDAP directory administrative user (in our case, the user wpsbind).
- **LDAPBindPassword**: The password assigned to the LDAP directory user that will be used by WebSphere Portal to connect or "bind" to the LDAP directory (again, in our case, the user was wpsbind).
- **WasPassword**: The password assigned to the WebSphere Application Server administrative user (in our case, the user wpsbind).
- **LTPAPassword**: An LTPA token is the mechanism used to achieve single sign-on between the Domino 6.5.1 products and WebSphere Portal. As part of the configuration of the wpconfig.properties file, the token is automatically.

Enable LDAP security for WebSphere Portal

This section describes how to enable LDAP security for WebSphere Portal. On the Portal Server node, there are pre-configured templates that can be customized to configure WebSphere Portal for LDAP.

- Open a command prompt and navigate to the <wp_home>\config directory.
- Change to the <was_home>\bin directory and enter the following commands:
 - startServer server1
 - stopServer WebSphere_Portal
- Change to the <wp_home>\config directory and enter the following command:

WPSconfig.bat validate-ldap

If an error occurs, review the values in the `wpsconfig.properties` (typographical errors are quite often the cause of an error on this step) and the settings in the LDAP server. Also, ensure that the LDAP server is actually running.

- If the validation was successful enable security by issuing the following command:

WPSconfig.bat enable-security-ldap

- Change to the `<was_home>\bin` directory and enter the following commands:

stopServer server1 -user wpsbind -password <password>

startServer server1 -user wpsbind -password <password>

stopServer WebSphere_Portal -user wpsbind -password <password>

Where `wpsbind` is the WebSphere Administrator user ID and `<password>` is the password for the user ID.