Configuring Authentication Providers

The authentication providers you configure for your XNAT determine how your site's users will be able to log in. If you do not specify any authentication providers, XNAT uses the default authentication provider, which uses the local XNAT database. The **admin** account is the default user account in a newly initialized XNAT database (there's also a user named **guest**, but that user can't be used to log in).

Many teams using XNAT have existing user repositories such as university-wide user databases that they'd like to leverage for user authentication and account management. XNAT provides the ability to integrate authentication through external repositories through its authentication provider API.

As of the XNAT 1.7.5 release, LDAP authentication is no longer part of the core XNAT application but requires installing the LDAP authentication plugin. See Configuring LDAP Authentication Providers below for more information.

This page explains how to configure one or many LDAP authentication providers for your XNAT, and how to preserve or remove your local database account access while doing so.

Upgrading?

If you have configured LDAP for versions of XNAT earlier than 1.7, you'll need to migrate properties from your existing deployment.

If you have configured LDAP for versions of XNAT 1.7 *prior to* 1.7.5 (including pre-1.7), you'll need to modify some of the property names to work properly with XNAT 1.7.5 and later.

If you are running XNAT 1.7.0, you should upgrade to the newest XNAT release, but if you can't the procedure to configure LDAP authentication providers for XNAT 1.7.0 *only* differs from subsequent point releases of XNAT 1.7.

Managing Authentication Provider Configurations

To add an authentication provider, create a properties file and add it to your XNAT installation:

1. Go to the **config** directory in your XNAT home folder. The default location for this is **/data/xnat/home/config**, but can vary by deployment.
2. If it doesn't already exist, create a new folder named **auth**. For the default settings, this would be located at **/data/xnat/home/config/auth**.
3. Place your properties file(s) in this new directory. Your authentication provider properties files *must* be named ***something*-provider.properties**. The *something* part can be any legal file name but should indicate the particular authentication provider.

Once you have these properties files in the **config/auth**, restart Tomcat.

XNAT 1.7.0 required putting your provider configuration properties in a plugin jar file. This is not compatible with later versions of XNAT, so if you have your provider properties configured in that way, you should just extract the properties file from the plugin jar, then remove the jar file from the XNAT plugins folder.

All XNAT authentication providers share a common set of properties:

Property	Required?	Default	Description	
name			Defines a human-readable name for the provider. This should be unique on the system.	
provider.id			Defines the ID for this provider. This must be unique on the system. This value is to enable the provider in XNAT.	
auth. method			Indicates the method to be used for authentication. This basically maps directly to the provider implementation. For the LDAP authentication provider, this is always ldap.	
auto. enabled		false	Indicates whether user accounts that authenticate using the provider definition should automatically be enabled on the system. If true, users can use the system right away. If false, an administrator needs to review and enable the account manually before the user can access the system.	
auto. verified		false	Indicates whether user accounts that authenticate using the provider definition should automatically be verified on the system. If false, users must receive an email from the system and click the provided link before they can access the system.	
visible		true	Indicates whether the provider is visible to users (i.e. displayed on the login page)	

Most authentication provider implementations require other properties as well, but which properties and acceptable values for those is dependent on the provider implementation.

Prior to XNAT 1.7, all authentication providers were configured in a single file (**services.properties**), with each provider distinguished by a prefix on the property names. For example:

```
provider.providers.enabled=db, mainrepo

provider.db.name=Database
provider.db.id=db
provider.db.type=db

provider.mainrepo.name=Main
provider.mainrepo.id=mainrepo
provider.mainrepo.type=ldap
provider.mainrepo.address=ldap://ldap.miskatonic.edu
provider.mainrepo.userdn=cn=readonly,dc=miskatonic,dc=edu
provider.mainrepo.password=password
provider.mainrepo.search.base=ou=users,dc=xnat,dc=org
provider.mainrepo.search.filter=(uid={0})
```

In this example, the default database provider has the ID **db**, so its properties are prefixed with **provider.db**, while the LDAP provider has the ID **mainrepo** and its properties prefixed with **provider.mainrepo**.

For XNAT 1.7, each provider must be defined in its own properties file, but the prefix is no longer required (as of XNAT 1.7.5, you also don't need to define the default database provider).

Notice that both the **db** and **mainrepo** provider configurations have three properties in common: **name**, **id**, and **type**. *Every* provider configuration must have these three main properties but the names of two of these properties has changed in XNAT 1.7.5!

- **id** is now **provider.id**
- **type** is now **auth.method**

To migrate the properties for the **mainrepo** provider above:

1. Move all of the properties for the provider to a file named something like **mainrepo-provider.properties**
2. Remove the **provider.mainrepo** prefix from all of the properties
3. Rename **id** to **provider.id** and **type** to **auth.method**
4. You can also add new properties supported in 1.7.5 or later: **visible**, **auto.enabled**, and **auto.verified**
5. Move your properties file to the **config/auth** folder

The **mainrepo** provider configuration file would look something like this:

```
name=Main
provider.id=mainrepo
auth.method=ldap
visible=true
auto.enabled=false
auto.verified=true
address=ldap://ldap.miskatonic.edu
userdn=cn=readonly,dc=miskatonic,dc=edu
password=password
search.base=ou=users,dc=xnat,dc=org
search.filter=(uid={0})
```

Enabling and Disabling Authentication Providers

Earlier versions of XNAT used a value set in a properties file to determine which configured providers should actually be enabled at run time. XNAT 1.7.5 has moved this to the **Security** section of the **Site Administration** page. The specific setting is labeled **Enabled Authentication Providers**. All authentication providers that should be active and enabled should be specified by the provider.id value, with each provider separated by a comma. These changes go into effect as soon as you click the Save button, i.e. no Tomcat restart is required.

This is also scriptable through the REST API. The currently enabled providers can be retrieved through the REST path /xapi/siteConfig/enabledProviders. The enabled providers can be set by POSTing a JSON list of the provider IDs. The code below queries and sets the enabled providers setting:

```
$ http --session=admin --body --verify=no https://xnatdev.xnat.org/xapi/siteConfig/enabledProviders
HTTP/1.1 200 OK
[
"localdb"
]

$ http --session=admin --body --verify=no POST https://xnatdev.xnat.org/xapi/siteConfig/enabledProviders <<<
'["localdb", "xnatldap"]'
HTTP/1.1 200 OK


$ http --session=admin --body --verify=no https://xnatdev.xnat.org/xapi/siteConfig/enabledProviders
HTTP/1.1 200 OK
[
"localdb",
"xnatldap"
]


$ http --session=admin --verify=no POST https://xnatdev.xnat.org/xapi/siteConfig/enabledProviders <<<
'["localdb"]'
HTTP/1.1 200 OK


$ http --session=admin --body --verify=no https://xnatdev.xnat.org/xapi/siteConfig/enabledProviders
[
"localdb"
]
```

Testing Configurations

You can test provider properties against an LDAP server using the ValidateLdap.groovy script (running this script requires having Groovy installed and the plugin jar available). To run the test script, use following syntax:

```
$ groovy 'jar:file:path/to/xnat-ldap-auth-plugin-1.0.0.jar!/ValidateLdap.groovy' [properties-file]
```

On Linux or OS X, the '' characters are required to prevent the "!" character from being detected by the shell interpreter. You can prefix the "!" with a backslash ("\") instead.

If you don't specify a properties file, the validate script will use the same default values as specified in ldap1-provider-sample.properties, along with the default user asmith and password password. You can specify a properties file that only overrides a few properties in the sample configuration as well, otherwise inheriting the values for the default properties. The username and password properties aren't normally configured in the provider properties definition but can be specified for the LDAP validator with the properties user and pass (note that password is already used in the provider definition, but indicates the password for the LDAP binding account and stays the same regardless of the specific username and password being validated).

The output from a successful validation looks something like this:

```
$ groovy 'jar:file:build/libs/xnat-ldap-auth-plugin-1.0.0.jar!/ValidateLdap.groovy'
Dec 06, 2017 3:18:45 PM org.springframework.security.ldap.DefaultSpringSecurityContextSource <init>
INFO: URL 'ldap://ldap.xnat.org', root DN is ''
User asmith authentication state: true
```

Adding configurations via plugins

Related Documentation

XNAT no longer supports configuration of authentication providers by plugin.

Configuring LDAP Authentication Providers

As of the XNAT 1.7.5 release, LDAP authentication is no longer part of the core XNAT application but requires installing the LDAP authentication plugin as described in this section.

Installing the LDAP Authentication Plugin

As of the XNAT 1.7.5 release, LDAP authentication is no longer part of the core XNAT application but requires installing the LDAP authentication plugin. You can download the latest release of this plugin from:

- The **Downloads** page on the plugin's source page
- The LDAP plugin project on the XNAT build server

Once you've downloaded the plugin jar, copy or move it into the **plugins** folder under your XNAT installation's home folder. Restart the Tomcat service.

Note that adding, modifying, or removing LDAP configurations also require a restart.

Managing LDAP Authentication Configurations

XNAT uses the Spring Security library's LDAP integration and should support most LDAP implementations (we've actively tested against Active Directory and OpenLDAP providers). To authenticate against an LDAP server, or multiple servers, you must create a separate properties file for each LDAP server.

To connect to an LDAP repository, you must provide some information about the LDAP server you want to use. Here is an LDAP properties template which shows what an LDAP properties file should look like (you will need to change these properties to match those of your LDAP and name the file `PROVIDER_ID-provider.properties`, where PROVIDER_ID is the id of the provider you are configuring):

```
name=LDAP
provider.id=ldap1
auth.method=ldap
address=ldap://ldapurl:389/dc=my,dc=domain
userdn=cn=MyServiceAccount,ou=MyGroup,dc=my,dc=domain
password=MyPassword
search.base=ou=people
search.filter=(uid={0})
```

name	what you want your users to see on the login page, if they have a choice of authentication providers
provider.id	uniquely identifies the provider in case there are multiple providers of a given type. If you add a second LDAP provider, it should have a different ID ("ldap2" is fine). Before XNAT 1.7.5, this property was simply **"id"**.
auth.method	indicates what type of provider it is. The two types that are currently supported are "db" for the local XNAT database and "ldap". Before XNAT 1.7.5, this property was **"type"**.
address	the URL of your LDAP server. Note the trailing parameters in the example URL. These should be included.
userdn	the server login configuration script that grants site-wide access to your LDAP server
password	password for that user
search.base	configures where the LDAP server should look for user accounts

search.filter	the LDAP field that contains the user's login name. This may be different depending on your LDAP implementation.
	An Active Directory implementation will need something like the following:
	`search.filter=(sAMAccountName={0})`
	With OpenLDAP it might be more like this:
	`search.filter=(uid={0})`

order	an optional parameter to control the order in which the login options appear on the XNAT login page in the case of multiple authentication providers. This should be an integer, where the providers with smaller "order" values are listed earlier in the dropdown. This property is only used prior to XNAT 1.7.5. As of 1.7.5, the order of the authentication providers is determined by the order of the providers in the array defined here.

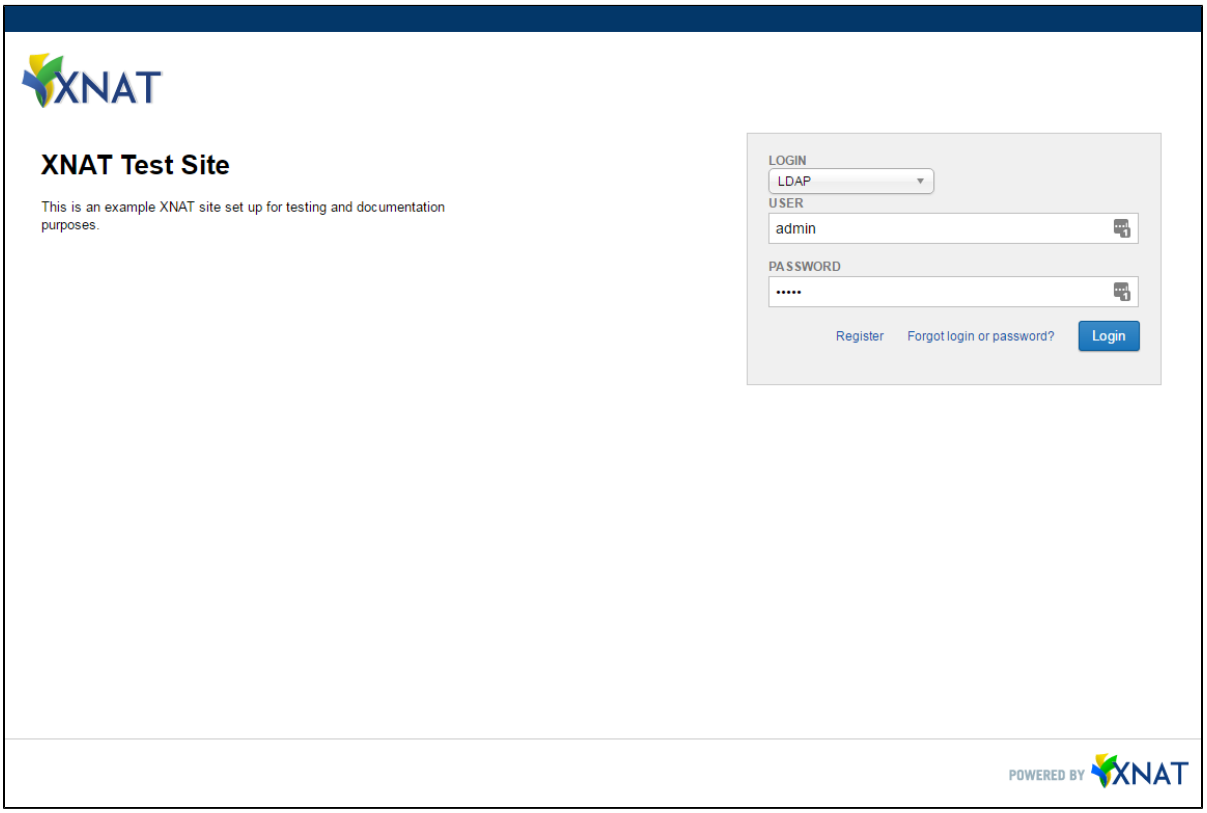
Preserving Local Database Accounts

This does not apply to XNAT 1.7.5 and later. Instead you can enable or disable each authentication provider by ID:

- Go to **AdministerSite Administration** (requires an administrator account)
- Click the **Security** tab on the left side
- Find the text box labeled **Enabled Authentication Providers**
- Enter the IDs of the authentication providers that should be enabled, separated by commas
- Click **Save**

User Logins With LDAP Providers

If you have configured more than one authentication provider, your XNAT login screen will give users the option of selecting how they want to log in. Otherwise, all logins will be checked against the only configured provider (regardless of whether that provider is LDAP or local database).



See XNAT User Management for more information about how to manage your user accounts. When someone with an LDAP account logs into XNAT for the first time, XNAT stores their user information and their user account can then be managed just like other user accounts. The one exception to this is that if you want to change passwords for LDAP users this needs to be done on the LDAP server unless your XNAT instance is customized to support modifying users on the LDAP server through the XNAT UI.