CD SWAT

DevTest 9.5.1

How-To

LDAP Integration with DevTest

Prepared by: Surya Suravarapu

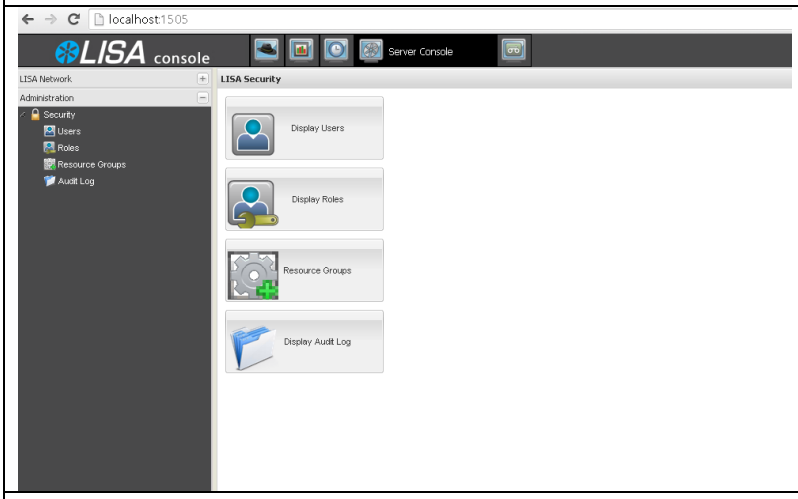
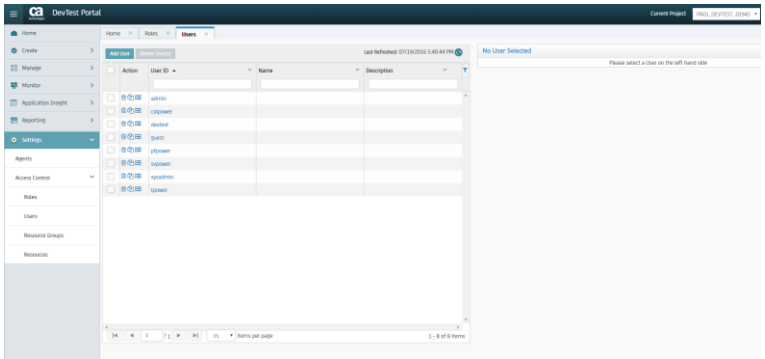
Date: January 2017

Table of Contents

Purpose

User authentication is an important aspect of DevTest. It helps control access to the various features of DevTest by the users. This is especially critical when multiple teams are involved. This document talks about LDAP integration for authentication in DevTest. This helps integrate the LDAP user's / user groups into DevTest for authentication.

User Authentication

	User authentication in DevTest is achieved via role based access control. It is also known as ACL. ACL comprises of Users and Roles. This information can be found under Settings -> Access Control in portal (DevTest 9.5) and under Administration tab of Server Console (for previous versions) as shown below.
	ACL in older DevTest versions
	ACL in DevTest 9.5

ACL

	ACL comprises of list of users and roles. Each role has permissions defined for various action / activities within DevTest. Some of the examples of activities are as given below. • VSE Service Deployment • Stage Test • View VSE Dashboard ACL is enabled by default from DevTest 8.0 onwards. For the older versions, we have to explicitly enable ACL for it to take effect. ACL can be enabled in versions older than 8.0 by adding the below property to local.properties file. lisa.acl.auth.enabled=true ACL data is stored in internal Derby database by default unless we configure an external database.

LDAP Integration

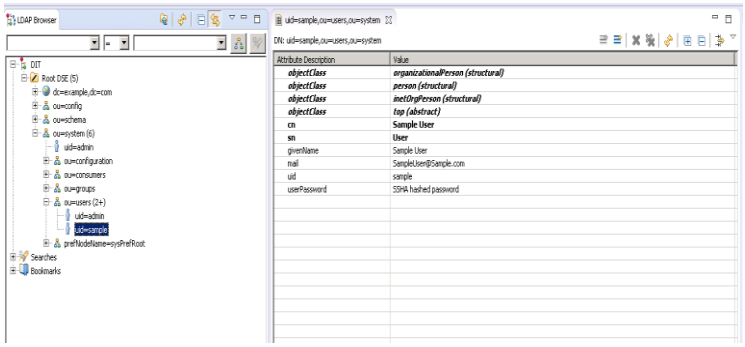
	User authentication can also be managed through LDAP integration with DevTest if an LDAP or Active Directory based system is already available. This enables LDAP administrator to manage users and their passwords. User passwords can no longer be managed through ACL. This integration will ensure integration into an already existing user management system. LDAP integration system has evolved over the different versions of Devtest. We will learn about LDAP Integration in older versions (7.5.2 in this document) which is LDAP user based and in newer versions (8.4 onwards), which has the ability to integrate LDAP user groups.

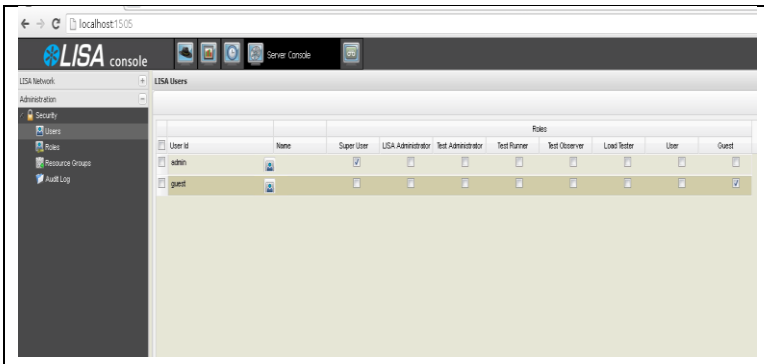
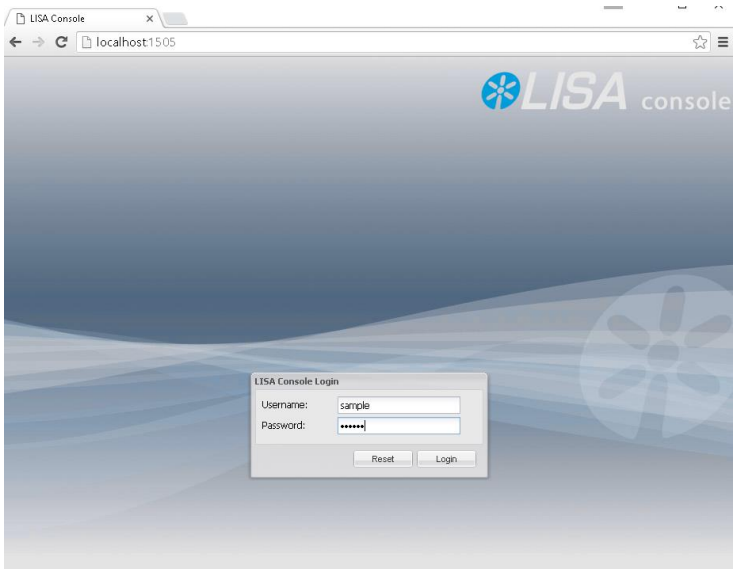
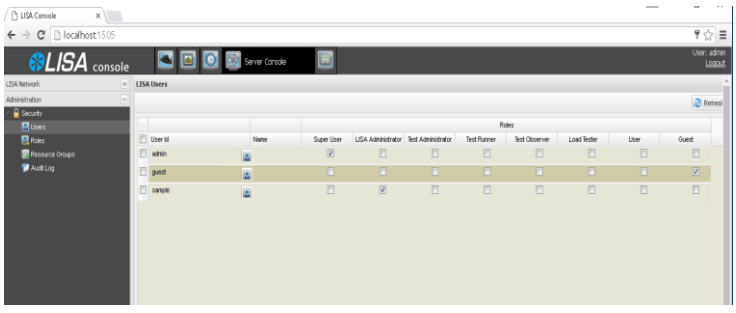
LDAP Integration in LISA 7.5.2

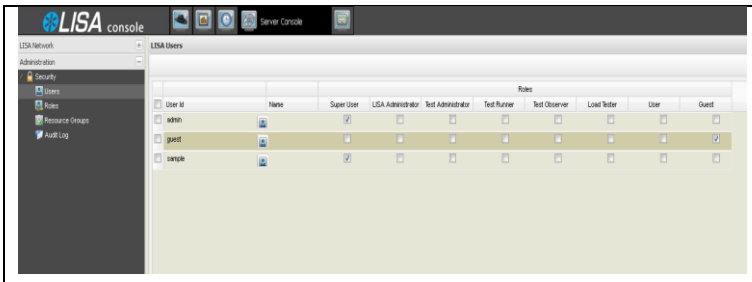
	LDAP Integration in LISA 7.5.2 involves configuring LDAP settings in local.properties file. The authorization process will authenticate the user and if user is not present in the LISA database, it will automatically add the user to the database.

Configuration for LDAP Integration

	For the configuration, the below properties need to be added. lisa.acl.ldap.ldapUrl – URL for the LDAP server. lisa.acl.ldap.securityPrincipal – The name of securityPrincipal (LDAP admin user) lisa.acl.ldap.securityCredential – The password for security principal. When we start the registry, the password will get encrypted and _enc will be added to property name. lisa.acl.ldap.securityAuthentication – security level to be used. We can set this to be either simple or none. If we set the value as none, the LDAP authentication will ignore the password and just validates the user name. In addition, we do not need to include the lisa.acl.ldap.securityPrincipal and lisa.acl.ldap.securityCredential properties. If we set the value to simple, the username and password will be validated. lisa.acl.ldap.baseContext - The distinguished name of the node where the user search begins. lisa.acl.userSearchFilter – The user search filter that specifies the object class for the user entries.

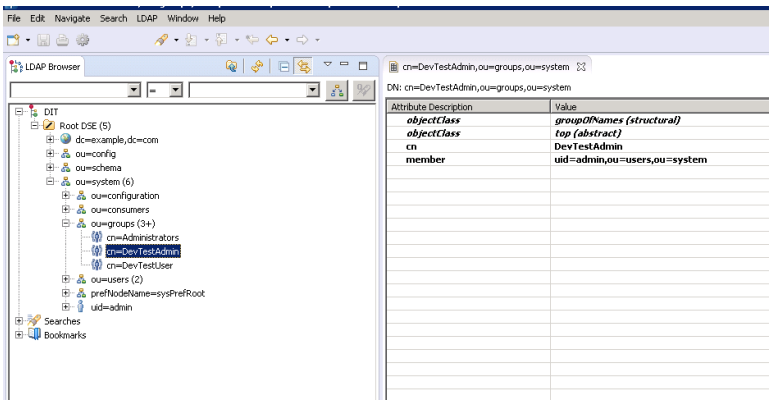
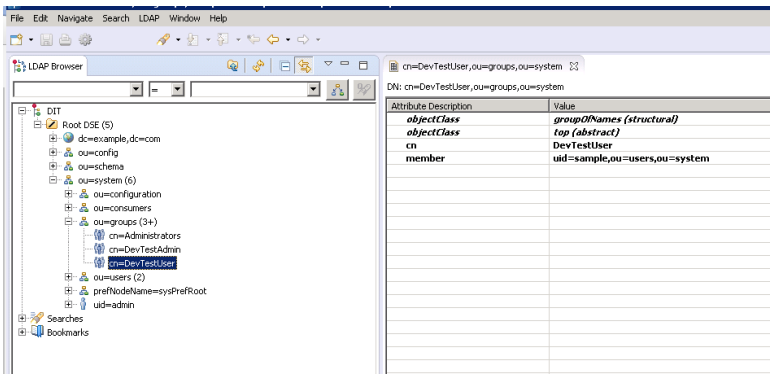
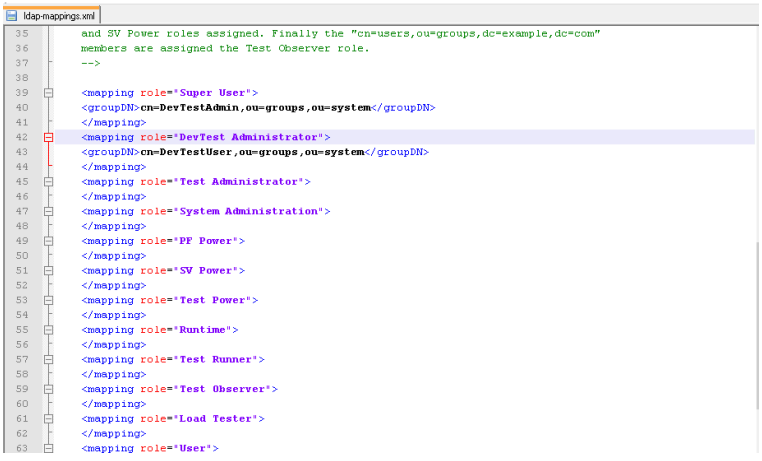
	lisa.acl.ldap.usernameAttribute – The attribute that specifies the user name. lisa.acl.ldap.userSearchAllDepths – This specifies whether to search subnodes (true) or not (false). lisa.acl.ldap.lisaDefaultRole – This specifies the default role to be assigned to users added to Lisa DB after being authenticated successfully. If not specified, the default role is Guest. Note: For LDAP authentication to work properly, remove or comment out the lisa.acl.auth.enabled property.
local.properties #lisa.vse.protocol.ims.header.length=80 # ========== # LISA Enterprise Dashboard # This is the URL to use for sending component and metric information # to the LISA Enterprise Dashboard # ========== #lisa.enterprisedashboard.url=tcp://somehost:2003/EnterpriseDashboard lisa.acl.auth.module.impl=com.itko.lisa.acl.custom.BaseLDAPAuthenticationModule lisa.acl.ldap.ldapUrl=ldap://cie-ldap.ca.com:10389 lisa.acl.ldap.securityPrincipal=uid=admin,ou=system lisa.acl.ldap.securityCredential_enc=b2e93d5ac89ecf680920df5b3be5d64b lisa.acl.ldap.securityAuthentication=simple lisa.acl.ldap.baseContext=ou=users,ou=system lisa.acl.ldap.userSearchFilter=(objectClass=inetOrgPerson) lisa.acl.ldap.usernameAttribute=uid lisa.acl.ldap.userSearchAllDepths=true lisa.acl.ldap.lisaDefaultRole=LISA Administrator	Fig 1. Sample local.properties file with ldap settings configured.
	Fig 2. Sample LDAP user configuration using Apache Directory Server.

	Fig 3. ACL screen showing default user configuration.
	Now, when you try logging in with an ldap user, the user will get added to the ACL database with the default role specified. This is because we have configured ldap integrated authentication.
	Fig 4. Login to LISA console using LDAP user.
	Fig 5. Lisa console showing the new user added as a LISA Administrator.

	Fig 6. Lisa console with the user role changes as Super User for him to be able to login successfully.
	Fig 7. Lisa console with the login successfully for the user.

LDAP Integration in DevTest 9.5

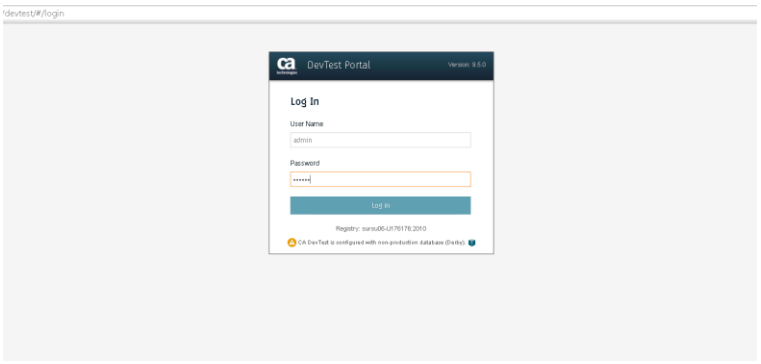
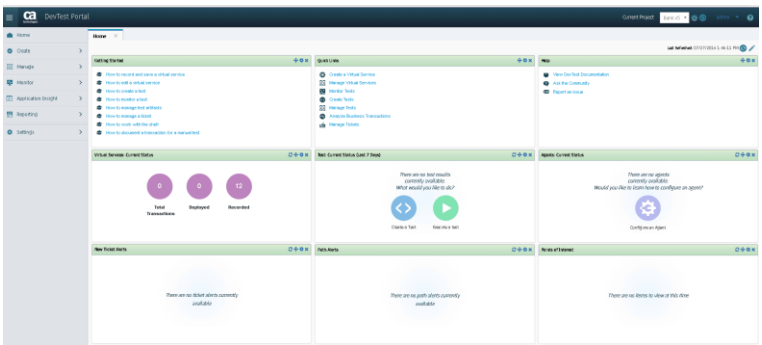
	We can configure ACL (starting version 8.4) so that user authentication is based on information in an LDAP server, multiple LDAP servers, the database or LDAP servers and the database. If the LDAP user is successfully authenticated and is not present in the database, the user will be added to the database. LDAP Integration (starting version 8.4) supports integration with LDAP groups. The configuration of LDAP involves configuring settings in authentication-providers.xml and ldap-mapping.xml files. More details given below.

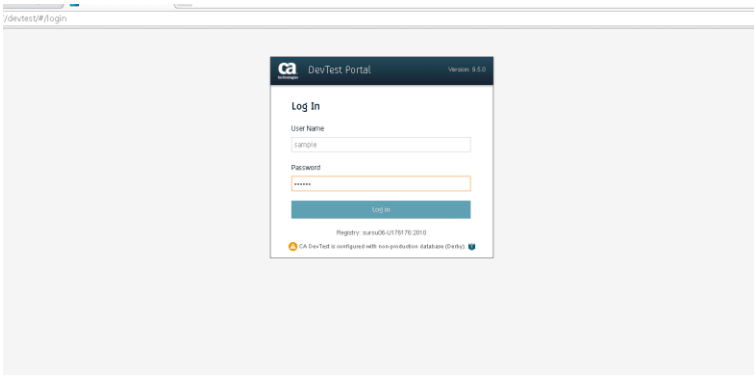
<table>
<tr>
<td> 
</td>
<td>Fig 8. LDAP Browser showing sample LDAP group configuration with two DevTest specific groups</td>
</tr>
<tr>
<td> 
</td>
<td>These LDAP groups can be mapped against DevTest ACL user roles as shown below via ldap-mapping.xml file. We can map more than one ldap group if needed against the Devtest user roles by adding an additional <groupDN> element.</td>
</tr>
<tr>
<td> 
</td>
<td>Fig 9. Ldap-mappings.xml file showing a sample mapping of LDAP groups against DevTest user roles.</td>
</tr>
<tr>
<td></td>
<td>The LDAP user and group configuration settings need to be made in authentication-providers.xml file as shown below.

Note: To use this type of LDAP authentication, both Devtest Workstation and Registry need to be upgraded to atleast 8.4 version.</td>
</tr>
</table>

	Authentication providers need to be configured in authentication-providers.xml file. We can configure as many authentication providers as needed in the order they need to be used.
	The following fields need to be filled for authentication providers. **name** – This is a name to identify the authentication provider **type** – This is to specify the type of authentication provider. Some of the providers are LDAP, ActiveDirectory, Embedded, Custom, or Legacy. These values are case sensitive, hence, need to specify in the case as specified. Legacy is deprecated and will be removed in a future release. **autoAddUsers** – This is to specify if successfully authenticated users are automatically added to the database or not. We can specify either true or false. This property is true by default. **authenticateOnly** – Controls whether LDAP / AD is only used to authenticate a user. If this value is true, then we need to explicitly create a user account in ACL database or configure autoAddUsers to true. This is false by default. **enabled** – This is to specify whether this authentication provider is available or not for authenticating users. The default value for this property is true. **defaultRole** – This is to specify the default role to be assigned to a successfully authenticated user if no other role is already assigned. Default value for this property is 'Guest'.

	rejectUnmappedUsers – This is to prevent users with no LDAP group mapped to roles from logging in. The default value for this property is true. If we specify this value as false and a user doesn't have an entry in ldap-mapping.xml file, the user is given a role associated with defaultRole parameter. The below are configuration settings specific to LDAP. **url** – The url of the ldap server. **user-dn** – the distinguished name of the LDAP user to be used to connect to server. user-password – The password associated with the user that is used to connect to the server. Passwords will automatically be encrypted once the registry starts running. **user-dn-pattern** – This is the pattern that is used to generate a distinguished name for the LDAP / AD user who is used to bind to the server. This element can be specified o or more times as needed and the patterns are tried in the order in which they occur. The pattern argument {0} will contain the user name. **user-search-base** – The relative name to be used while searching for users. **user-search-filter** – The LDAP filter used to find user entries. The search filter can include a placeholder '{0}' that contains the user name of the user who is trying to login. If the filter also contains the objectClass value as follows, the filter will not only look for an attribute value match, but also look for entries of the object class specified. (&(objectClass=interOrgPerson)(uid={0})) **group-search-base** – The relative name to start search for groups.

	group-search-filter – The LDAP filter user to search for group entries.
authentication-providers.xml 100 <authentication-provider 101 name="ITKO Authentication Module" 102 type="Legacy" 103 enabled="true" 104 defaultRole="Guest"/> 105 106 --> 107 <authentication-provider 108 name="Corp. Active Directory Server" 109 autoAddUsers="true" 110 authenticateOnly="false" 111 enabled="true" 112 type="LDAP" 113 defaultRole="SW Power" 114 rejectUnmappedUsers="true"> 115 <url>ldap://cie-ldap.ca.com:10389</url> 116 <user-dn>uid=admin,ou=system</user-dn> 117 <user-password>{cry}13ec2589b1b2f0e0637a0de4a37ffc04772182c821b5bba4c94c7e42c324772d436ad2a52194e</ 118 <user-dn-pattern>uid={0},ou=users,ou=system</user-dn-pattern> 119 <user-search-base>ou=users,ou=system</user-search-base> 120 <user-search-filter>(&(objectClass=inetOrgPerson)(uid={0}))</user-search-filter> 121 <group-search-base>ou=groups,ou=system</group-search-base> 122 <group-search-filter>(member={0})</group-search-filter> 123 </authentication-provider> 124 125 </authentication-providers> 126	Fig 10. authentication-providers.xml file showing a sample configuration of LDAP authentication provider.
 	Fig 11. Login with user 'admin' who is part of LDAP group DevTestAdmin, which has been assigned to SuperUser role.



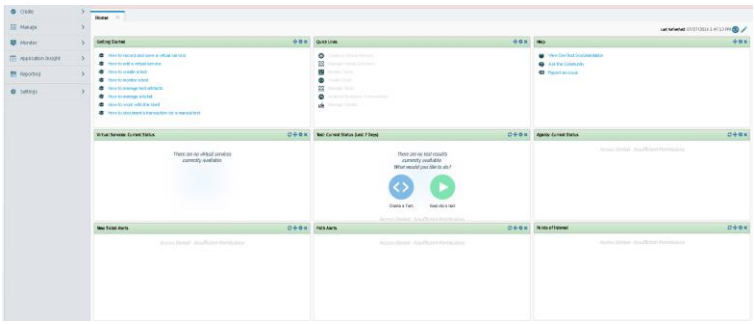


Fig 10. Login with user 'sample' who is part of LDAP group DevTestUser, which has been assigned to DevTest Administrator role.

Summary

Using LDAP integration helps better manage DevTest users. The recent changes have made it even easier to configure users especially from large organizations. Organization who have an existing LDAP / ActiveDirectory installation can benefit immensely from this feature.