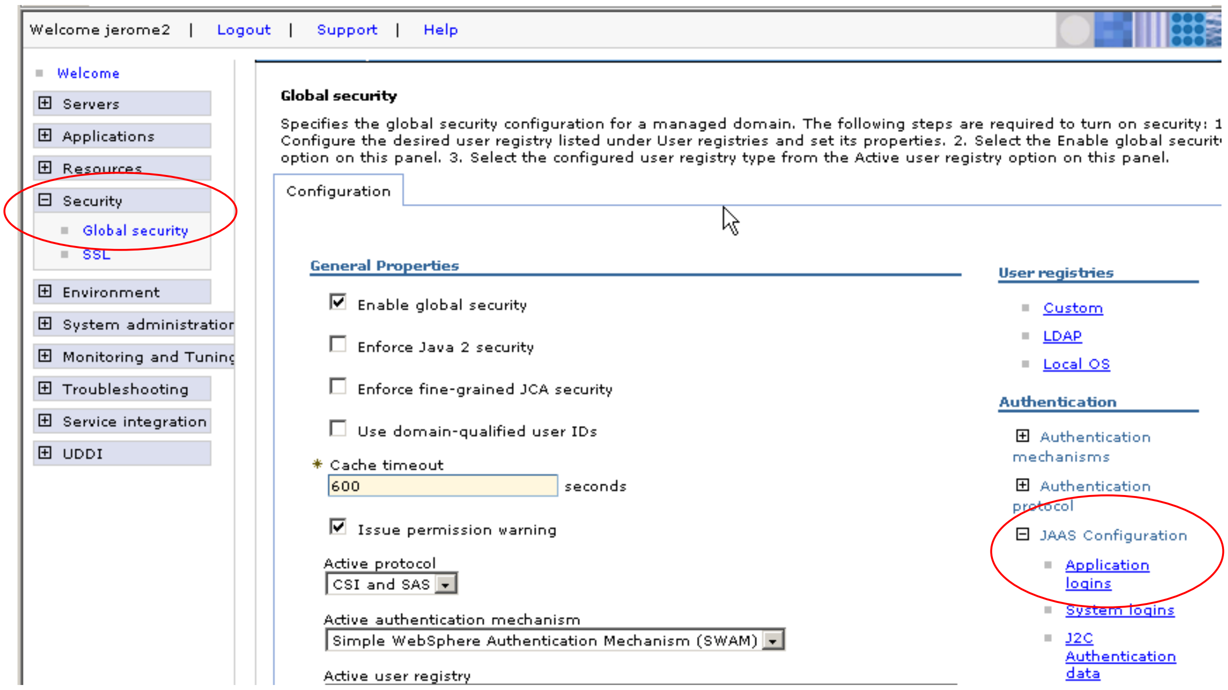
Using jaaslounge within websphere 6.0

The use of jaaslounge on a websphere server is done in three steps:

- add a new jaas configuration in the websphere “Application login configuration” mechanism, with desired jaaslounge login module;
- set the custom registry to the supplied *JaaslougeRegistry* adapter class, and configure it to use the previously declared application login configuration;
- notify websphere to use the custom registry

New application login configuration

Launch Websphere server and run the administrative console – usually located at http://*yourServer*:9060/ibm/console (you may have to log on with the previously defined authentication scheme).
On the *Security* → *Global security* pane, access the *JAAS configuration* → *Application logins* screen.



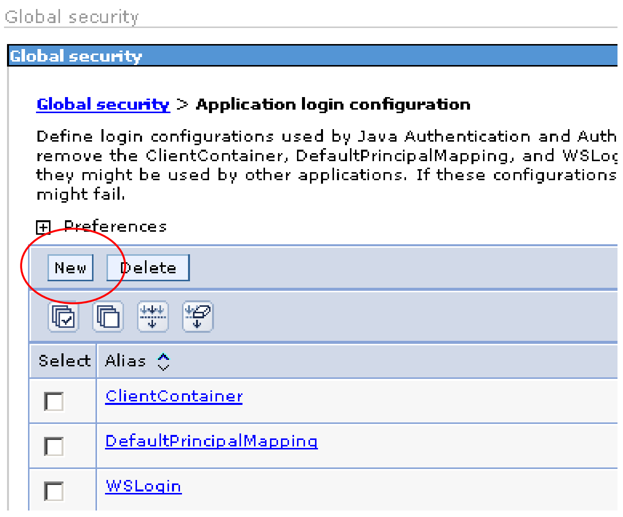
Global security

Specifies the global security configuration for a managed domain. The following steps are required to turn on security: 1 Configure the desired user registry listed under User registries and set its properties. 2. Select the Enable global securit option on this panel. 3. Select the configured user registry type from the Active user registry option on this panel.

General Properties

☑ Enable global security
☐ Enforce Java 2 security
☐ Enforce fine-grained JCA security
☐ Use domain-qualified user IDs
* Cache timeout: 600 seconds
☑ Issue permission warning
Active protocol: CSI and SAS
Active authentication mechanism: Simple WebSphere Authentication Mechanism (SWAM)
Active user registry

Create a new configuration:



Global security

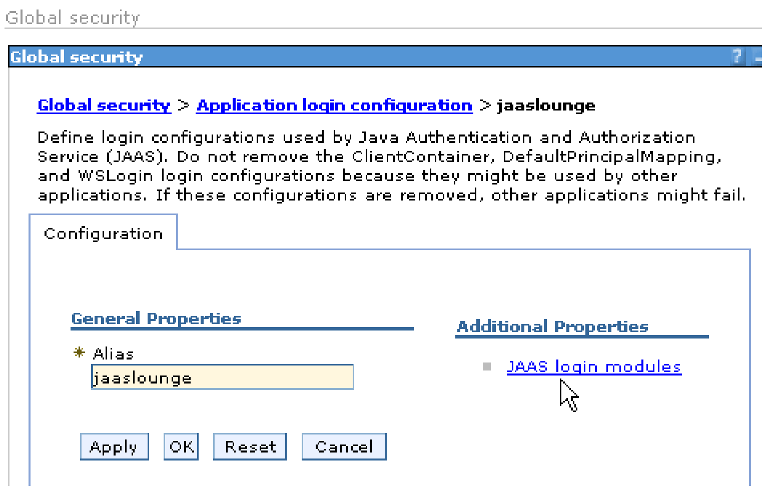
Global security

Global security > Application login configuration

Define login configurations used by Java Authentication and Auth
remove the ClientContainer, DefaultPrincipalMapping, and WSLog
they might be used by other applications. If these configurations
might fail.

Select	Alias
☐	ClientContainer
☐	DefaultPrincipalMapping
☐	WSLogin

Give an explicit alias to your configuration, click on *Apply*, and then access the *JAAS login modules* screen.



Global security

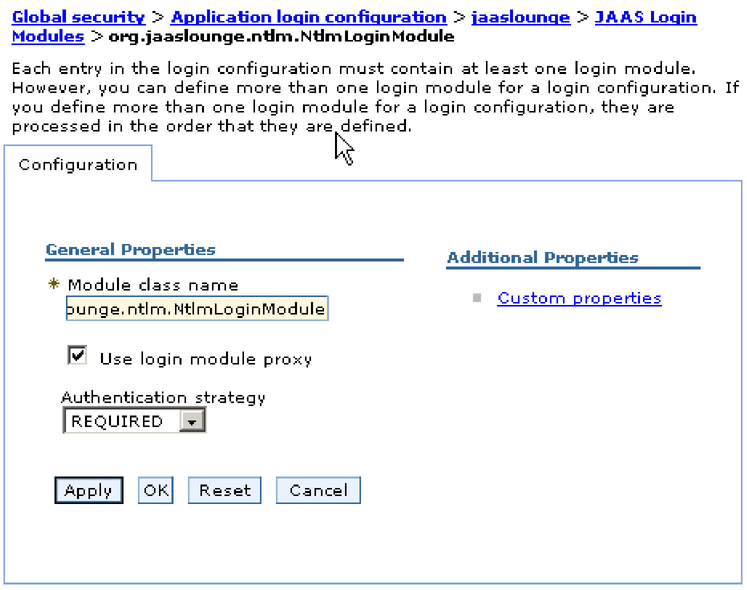
Global security

Global security > Application login configuration > jaaslounge

Define login configurations used by Java Authentication and Authorization Service (JAAS). Do not remove the ClientContainer, DefaultPrincipalMapping, and WSLogin login configurations because they might be used by other applications. If these configurations are removed, other applications might fail.

* Alias: jaaslounge

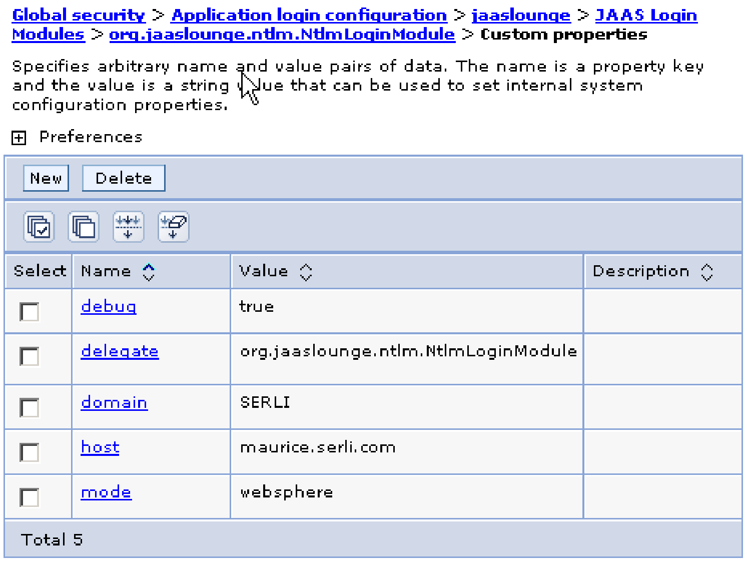
On the *JAAS login modules* screen, click on *New* button. The module window opens. On this form, specify the jaaslounge login module class you want to use, for instance org.jaaslounge.ntlm.NtlmLoginModule or org.jaaslounge.ldaplm.LDAPLoginModule. Select the “Use login module proxy” checkbox.


Global security > Application login configuration > jaaslounge > JAAS Login Modules > org.jaaslounge.ntlm.NtlmLoginModule

Each entry in the login configuration must contain at least one login module. However, you can define more than one login module for a login configuration. If you define more than one login module for a login configuration, they are processed in the order that they are defined.

* Module class name: ounge.ntlm.NtlmLoginModule
* [✓] Use login module proxy
* Authentication strategy: REQUIRED

Click on *Apply*, and then open the *Custom properties* screen to set necessary properties.
(Refer to the specific documentation of the jaaslounge module you want to use).
The *mode* property, common to all jaaslounge modules, must be set to "websphere".


Global security > Application login configuration > jaaslounge > JAAS Login Modules > org.jaaslounge.ntlm.NtlmLoginModule > Custom properties

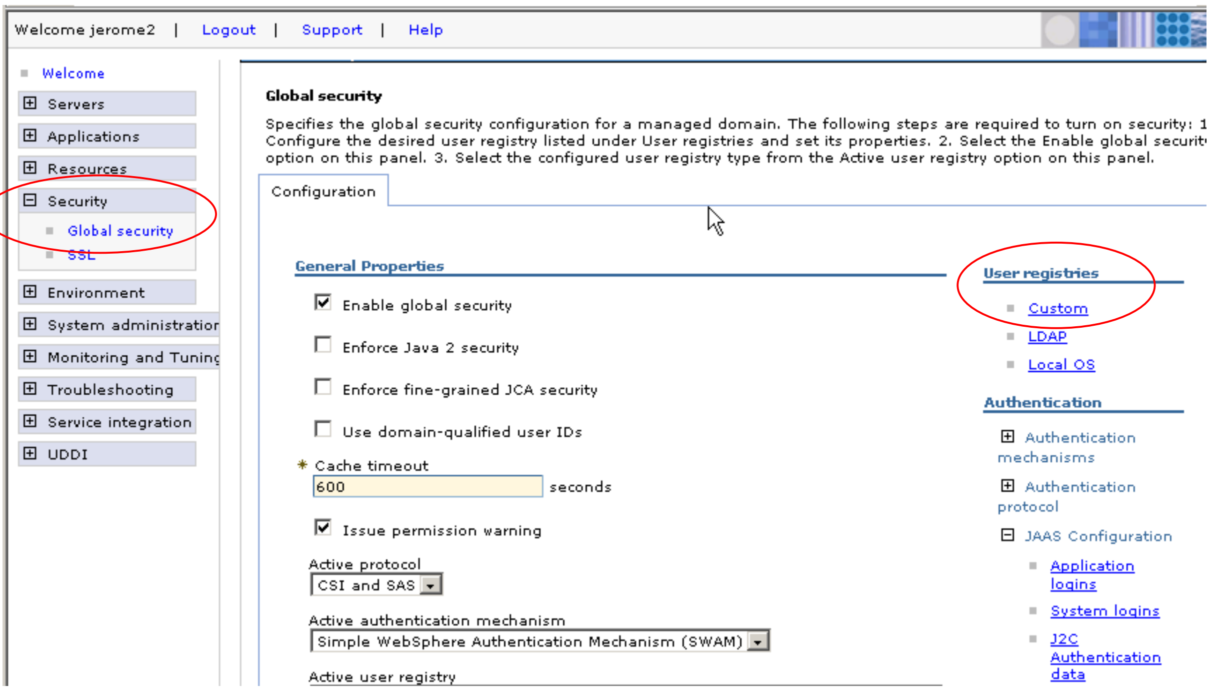
Specifies arbitrary name and value pairs of data. The name is a property key and the value is a string value that can be used to set internal system configuration properties.

Select	Name	Value	Description
☐	debug	true	
☐	delegate	org.jaaslounge.ntlm.NtlmLoginModule	
☐	domain	SERLI	
☐	host	maurice.serli.com	
☐	mode	websphere	

Total 5

Configure the custom registry

Now go back to the *Security* → *Global security* pane, and access the *User registries* → *Custom* screen.


Welcome jerome2 | Logout | Support | Help

Global security

Specifies the global security configuration for a managed domain. The following steps are required to turn on security: 1 Configure the desired user registry listed under User registries and set its properties. 2. Select the Enable global securit option on this panel. 3. Select the configured user registry type from the Active user registry option on this panel.

Configuration

General Properties

- [✓] Enable global security
- [ ] Enforce Java 2 security
- [ ] Enforce fine-grained JCA security
- [ ] Use domain-qualified user IDs

* Cache timeout: 600 seconds

- [✓] Issue permission warning

Active protocol: CSI and SAS

Active authentication mechanism: Simple WebSphere Authentication Mechanism (SWAM)

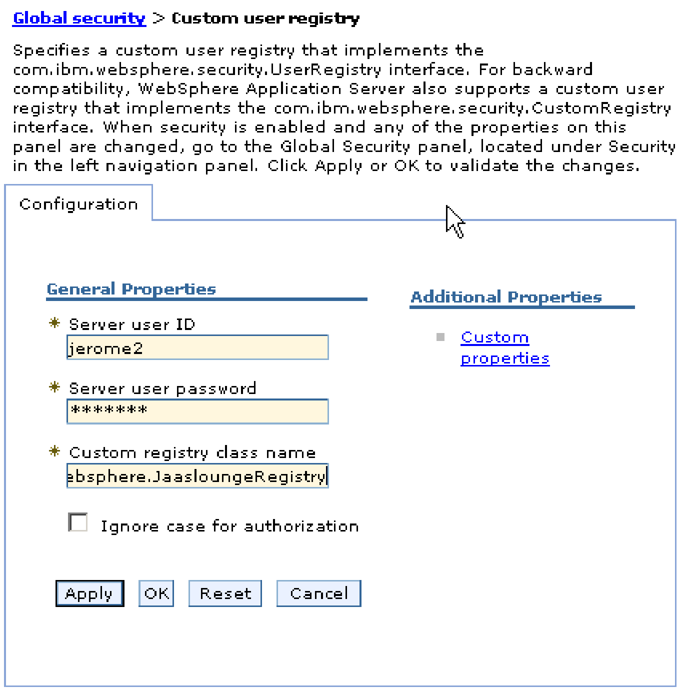
Active user registry

Set the class name for the user registry to :

org.jaaslounge.adapters.websphere.JaasloungeRegistry

This class will allow to call the application login configuration previously defined.

Note: Be careful that the values you give for *Server user Id* / *Server user password* actually exist in the target authentication system, because these values will be used to log on the administrative console once your new registry will be taken in acount.


Global security > Custom user registry

Specifies a custom user registry that implements the com.ibm.websphere.security.UserRegistry interface. For backward compatibility, WebSphere Application Server also supports a custom user registry that implements the com.ibm.websphere.security.CustomRegistry interface. When security is enabled and any of the properties on this panel are changed, go to the Global Security panel, located under Security in the left navigation panel. Click Apply or OK to validate the changes.

Configuration

General Properties

* Server user ID: jerome2

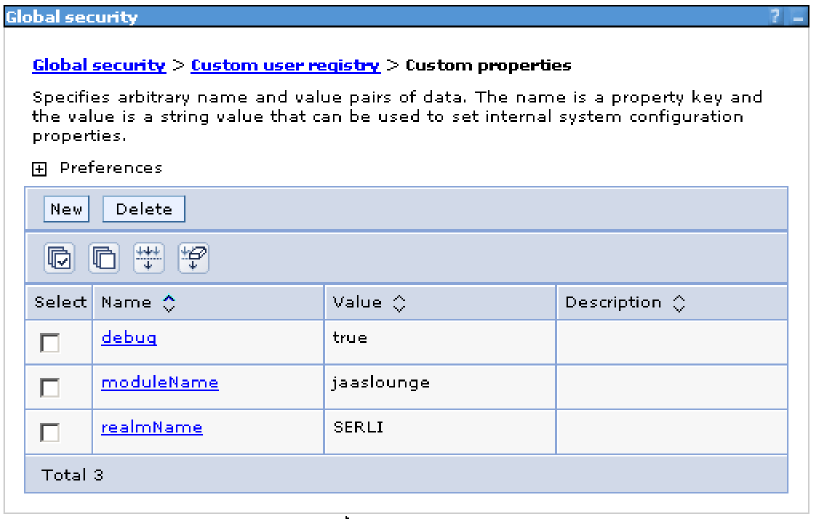
* Server user password: *******

* Custom registry class name: ebsphere.JaaslougeRegistry

☐ Ignore case for authorization

Open now the *custom properties* window for the custom registry. Three properties can be defined to use this class:

- moduleName (mandatory) must contain the alias you gave to your Application Login Configuration.
- realmName (mandatory) is the name of the custom realm you intend to receive. (this name is prefixed to the user and group names in the returned Principal and Credential occurences, it will be seen only by developers. It has no special impact)
- debug (optional) might be set to “true” to have some execution trace in the server’s log file.


Global security

Global security > Custom user registry > Custom properties

Specifies arbitrary name and value pairs of data. The name is a property key and the value is a string value that can be used to set internal system configuration properties.

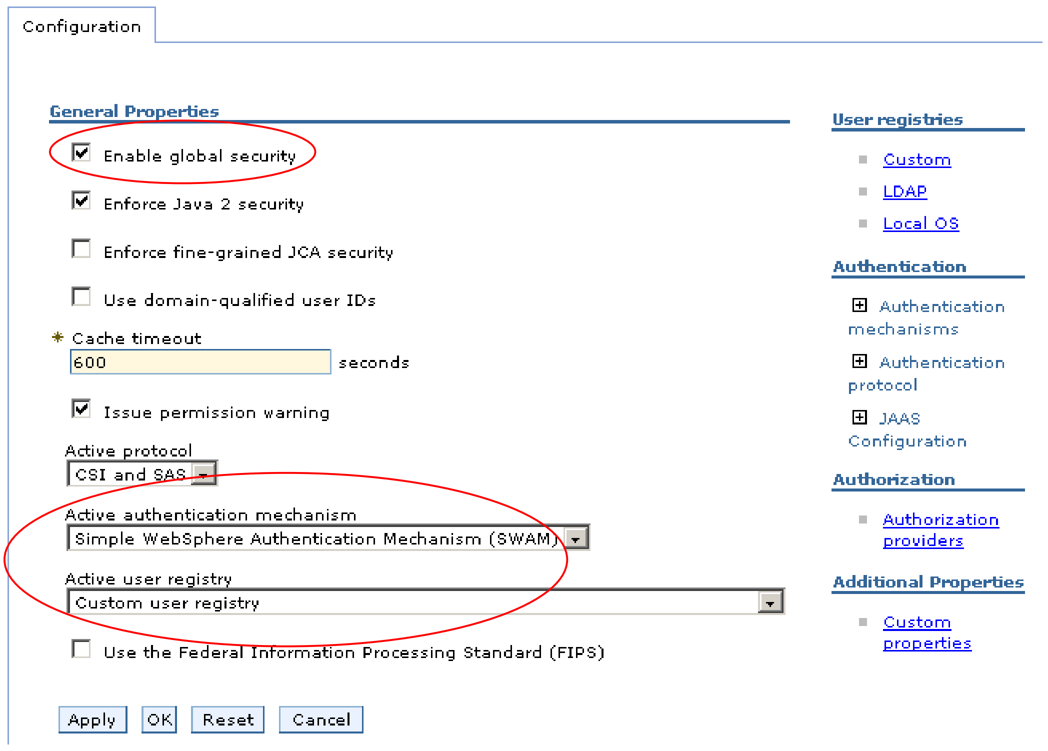
Select	Name	Value	Description
☐	debug	true	
☐	moduleName	jaaslounge	
☐	realmName	SERLI	

Total 3

Use the custom registry

Last step of our installation is to notify websphere to use the custom registry. Go back to the *Security → Global security* main window.

- set the *Active User Registry* listbox to “Custom user registry”
- set the *Active authentication mechanism* listbox to SWAM
- select the *Enable global security* checkbox



Configuration

General Properties

- ✓ Enable global security
- ✓ Enforce Java 2 security
- ☐ Enforce fine-grained JCA security
- ☐ Use domain-qualified user IDs

* Cache timeout: 600 seconds

- ✓ Issue permission warning

Active protocol: CSI and SAS

Active authentication mechanism: Simple WebSphere Authentication Mechanism (SWAM)

Active user registry: Custom user registry

- ☐ Use the Federal Information Processing Standard (FIPS)

Copy the jar files to your websphere server libraries directory:

ServerBaseDirectory/lib/ext

Copy into this directory *jaaslounge.jar* and the additional jar files depending on chosen module (*jcifs.jar* / *jcifs-ext.jar* for instance, for NtlmLoginModule)

Restart the server.

Note

Up to know, we set up a simple case, using one jaaslounge login module. But by mixing the power of websphere *application login configuration* mechanism with the flexibility of jaaslounge module, you can imagine to add different jaaslounge modules to your configuration.
In that case, the different modules will be called in sequence, depending on their result and the “authentication stategy” chosen for each one (*Required* / *Requisite* / *Sufficient* / *Optional* – see websphere documentation).
The different groups returned by all succesful modules are then merged in the list of groups of the final Subject.
You could for instance imagine to implement

- a very strict control that accepts the user only if it is registered with different authentication systems (many modules with REQUISITE option)

- a system that accepts a user if it is registered with at least one system (many modules with SUFFICIENT or OPTIONAL option)
- a system that authenticate the users from a fist REQUISITE module, and then enriches the groups list of the user from many OPTIONAL modules.

The only constraint is to use only jaaslounge login modules, because they return information in a form expected by the *JaasloungeRegistry* adapter.