RunaWFE. TaskNotifier. Administrator guide

Version 3.0

Changing a RunaWFE server associated with rtn client

By default rtn client is associated with RunaWFE server that is supposed to be located on a computer named wfe_server. In order to change this setting you can use do the following:

1. On the client computer (where you have rtn client installed) make an association between RunaWFE server IP address and wfe_server name. On Windows OS you can do it by adding a string following the pattern "ip_address wfe_server" to 1. C:\WINDOWS\system32\drivers\etc\hosts file contents. On Linus OS the same is done in /etc/hosts file.

2. Change the setting in rtn client to another RunaWFE server name. This can be done in af_delegate.properties and application.properties files. You can put a new name for RunaWFE server or its IP address in the place of wfe_runa server name. If RunaWFE uses ports that differ from the default (1099 and 8080) the port numbers should be replaced by the actual ports numbers.

Authentication

Choosing the authentication type

The authentication process in rtn client consists of 2 parts:

- RMI authentication. It is used by the rtn client to obtain the information about user's tasks. When a new task arrive the rtn icon in the system tray changes and a popup message appears.
- Authentication in the rtn built in web browser. This is necessary for the correct work of the RunaWFE system web interface.

There are 2 available ways to perform RMI authentication: with the help of RunaWFE user login and password or via kerberos. You can set the chosen authentication type in authentication.type property of the file application.properties. For the first type set the value of property to "userinput" and for the second to "kerberos". If you choose kerberos authentication type no additional input is required from the user during the authentication of the rtn client. While if you choose the login and password authentication type user will be prompted to enter the login and password during the authentication process.

The type of the built in web browser authentication is set via login.relative.url property in application.properties in the form of an url relatively the RunaWFE web-interface address. There are 3 urls available for the rtn built in web browser authentication:

- /login.do - Use it to set the authentication via entering login and password. This requires that RMI authentication type is also set to the "userinput" type.
- /ntlmlogin.do - set the ntlm protocol to be used for the authentication.
- /krblogin.do - set the kerberos protocol to be used for the authentication.

Configuring Username and Password Authentification

To configure authentification via username and password, open application.properties file in the Task Notifier folder, find a string containing authentication.type parameter and set its value to userinput (authentication.type=userinput).

Note. It is also possible to set a default username and password. Edit the following parameters in the above file:

- userinput.default.login – default login name
- userinput.default.password – default password

If you want to login with set above login name and password automatically (without dialog window appearing every time during login) set *userinput.login.silently* to **true**.

Configuring Kerberos Authentification

Note. In this section, all user and server names and principals are case sensitive.

Configuring Kerberos on the rtn client computer

1. Set the following parameter values in the registry key:

- For Windows Server 2003 and Windows 2000 SP4

key: HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\Kerberos\Parameters

parameter: allowtgtsessionkey=dword:0x01

- For Windows XP SP2

key: HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\Kerberos

parameter: allowtgtsessionkey=dword:0x01

Note. After setting the parameter, it is necessary to reboot the computer.

A description of the issue, solved by this step, is available at: http://java.sun.com/j2se/1.5.0/docs/guide/security/jgss/tutorials/Troubleshooting.html, chapter "javax.security.auth.login.LoginException: KrbException: KDC has no support for encryption type (14) - KDC has no support for encryption type".

2. Create/edit Kerberos configuration file krb5.ini

This configuration file must be in the %SystemRoot% folder and have the name of krb5.ini.

It is necessary to specify the following cryptographic algorithms:

```
[libdefaults]

default_tkt_enctypes = des-cbc-md5 des-cbc-crc des3-cbc-sha1

default_tgs_enctypes = des-cbc-md5 des-cbc-crc des3-cbc-sha1

permitted_enctypes = des-cbc-md5 des-cbc-crc des3-cbc-sha1
```

3. Install JRE5.0.10 or higher on the client computer. You can download it from http://java.sun.com/j2se/1.5.0/download.jsp.

4. After the RunaWFE server is configured the client application can be activated by executing $(NOTIFIER_ROOT)\run.exe (run.sh).

A detailed description of a Kerberos configuration file

A detailed description of a Kerberos configuration file see at: http://web.mit.edu/kerberos/www/krb5-1.4/krb5-1.4.3/doc/krb5-admin/krb5.conf.html [2].[3]

Configuring the Server Side

A detailed description of RunaWFE configuration see in RunaWFE. Guide for installation and configuration [4]. Keep in mind that for the correct work of the rtm client via RMI with kerberos it is necessary to set in the file kerberos_module.properties (in rtn folder) the same serverPrincipal as it is set in RunaWFE server (in the same named property in the kerberos_module.properties in jboss-root/server/default/conf).

Configuring JVM Security

Enable the security manager. To enable the security manager for all locally executed applications, define the environment variable _JAVA_OPTIONS and set its value to -Djava.security.manager .

The default security manager restrictions will then be applied to all locally executed applications. These restrictions are defined in file $JAVA_HOME\lib\security\java.policy.

The format of this file is described in http://java.sun.com/j2se/1.5.0/docs/guide/security/PolicyFiles.html [5].
For a list of permission types, used in the security manager, see http://java.sun.com/j2se/1.5.0/docs/guide/security/permissions.html.

Here is an example of a policy (from java.policy file) which grants all permissions to the classes contained in the D:\tmp folder and grants no permissions to the classes contained in any other folder of the file system (including those from JAR archives).

```
// Standard extensions get all permissions by default
grant codeBase "file:$/*" {
permission java.security.AllPermission;
};
// default permissions granted to all domains
grant {
// Allows any thread to stop itself using the java.lang.Thread.stop()
// method that takes no argument.
// Note that this permission is granted by default only to remain
// backwards compatible.
// It is strongly recommended that you either remove this permission
// from this policy file or further restrict it to code sources
// that you specify, because Thread.stop() is potentially unsafe.
// See "http://java.sun.com/notes" for more information.
permission java.lang.RuntimePermission "stopThread";
// allows anyone to listen on un-privileged ports
permission java.net.SocketPermission "localhost:1024-", "listen";
// "standard" properies that can be read by anyone
permission java.util.PropertyPermission "java.version", "read";
permission java.util.PropertyPermission "java.vendor", "read";
permission java.util.PropertyPermission "java.vendor.url", "read";
permission java.util.PropertyPermission "java.class.version", "read";
permission java.util.PropertyPermission "os.name", "read";
permission java.util.PropertyPermission "os.version", "read";
permission java.util.PropertyPermission "os.arch", "read";
permission java.util.PropertyPermission "file.separator", "read";
permission java.util.PropertyPermission "path.separator", "read";
permission java.util.PropertyPermission "line.separator", "read";
permission java.util.PropertyPermission "java.specification.version", "read";
permission java.util.PropertyPermission "java.specification.vendor", "read";
permission java.util.PropertyPermission "java.specification.name", "read";
permission java.util.PropertyPermission "java.vm.specification.version", "read";
permission java.util.PropertyPermission "java.vm.specification.vendor", "read";
permission java.util.PropertyPermission "java.vm.specification.name", "read";
permission java.util.PropertyPermission "java.vm.version", "read";
permission java.util.PropertyPermission "java.vm.vendor", "read";
permission java.util.PropertyPermission "java.vm.name", "read";
};
grant codeBase "file:/D:/tmp/*" {
permission java.security.AllPermission;
};
```

How to Start Task Notifier

Set a reference to RUNA WFE server In af_delegate.properties.

Put swt-win32-3232.dll to a directory which is included in Path environment variable value. Run

```
javaw -cp .;rtn.jar ru.runa.notifier.PlatformLoader
```

Note. If you don't want to use Path variable put swt-win32-3232.dll to the same directory with the rtn.jar. And run

```
javaw -Djava.library.path=. -cp .;rtn.jar ru.runa.notifier.PlatformLoader
```

or

```
runa_tasks.exe
```

Run run.bat or run.sh.

Setting Up Email Notification

In order to enable email notification

1) The executors who are supposed to receive emails must have their email address set in executor properties and have "Active" status.

2) Set smtp.SendNotifiction=true in server\default\conf\emailTaskNotifier.properties. In the same file set the smtp server parameters and valid smtp server user name, that will be used to send emails from.

References

[1] http://www.gnu.org/licenses/lgpl.html

[2] http://web.mit.edu/kerberos/www/krb5-1.4/krb5-1.4.3/doc/krb5-admin/krb5.conf.html

[3] An example of a krb5.ini configuration file is attached.

[4] http://wf.runa.ru/doc/WF-system_Installation_guide

[5] http://java.sun.com/j2se/1.5.0/docs/guide/security/PolicyFiles.html

Article Sources and Contributors

RunaWFE. TaskNotifier. Administrator guide *Source*: http://wf.runa.ru/doc/index.php?oldid=668 *Contributors*: Natkinnat, WikiSysop