RUNA WFE. Task Notifier Administrator's Guide.
Version 2.1

© 2004-2008, ZAO Runa.
RUNA WFE is an open source system distributed under a LGPL license (http://www.gnu.org/licenses/lgpl.html).

Configuring Username and Password Authentification

To configure authentification through 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

Configuring Kerberos Authentification

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

Configuring TaskNotifier (an application that notifies users about new tasks)

Configuring the client

Note. The client application (TaskNotifier) is built with preset server principals, WFTestUser. To change principals, it is necessary to rebuild the application.

1. 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.
2. Create a folder on the client computer with no spaces in the name of the folder. In this document, this folder will be referred to as $(NOTIFIER_ROOT).Extract files rtn.jar and runa_tasks.exe from the Workflow_comp\releases\notifier\task-notifier-2.0<...>.zip archive file into this folder.
3. Configure authentification. TaskNotifier can be configured to use either Kerberos or username and password authentication. See below.
4. After configuring the server part, you can activate the client application by executing file $(NOTIFIER_ROOT)\runa_tasks.exe (this starts a Java Virtual Machine and passes it the links to classes, located in rtn.jar).
5. (Setting up **MS InfoPath** forms). To display InfoPath forms (xsn type) on a client on Windows, it is necessary to install a COM component for InfoPath. To install this component, register it in the system by the **regasm** tool (which is included into the .Net Framework SDK) and then install DLL in GAC. Note. If this component is not installed, the client will work but will not be able to display InfoPath forms.

Note. Settings for a TaskNotifer build are described in the document "RUNA WFE. Developer's guide. Version 2.0", "Configuring a Thick Client" section.

Steps for Configuring Kerberos Authentification

1. Set `authentication.type = "kerberos"` (see the "Developer Guide")

2. 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".

3. 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
```

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***.[1]

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. 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:${{java.ext.dirs}}/*" {
        permission java.security.AllPermission;
};

// default permissions granted to all domains
```

1 An example of a krb5.ini configuration file is attached.

```
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;
};
```

Configuring the Server Part

Steps:

1. Include Kerberos login module into file $(DIST_ROOT)/server/default/conf/login_module.properties.

For example, ru.runa.af.authenticaion.KerberosLoginModule=SUFFICIENT.
Set server and user principals for the user, performing authentification, in the $(DIST_ROOT)/server/default/conf/login_module.properties file.

- principal is a name of the service, performing authentication (WFTestUser@RUNA.RU in the current implementation)

- serverPrincipal is a user principal for a user performing authentication of the server (actually, the user login name)

2. Configure the server, just like you configured the client computer.

3. Turn on DES cryptography for the user whose name is used for authentification in AD. See http://www.microsoft.com/windows2000/en/advanced/help/default.asp?url=/windows2000/en/advanced/help/dsadmin_concepts_accounts.htm.

4. On the Workflow server, create Kerberos keytab with information necessary for authentification of this user. Do it with ktab tool that comes with JDK 5.0. For example,

ktab -k file:///c:/winnt/krb5.keytab -a WFTestUser@RUNA.RU

(In the current implementation use name WFTestUser@RUNA.RU)

5. Synchronize time on the domain controller, workflow server and client.
See a detailed description of the ktab tool at http://java.sun.com/j2se/1.5.0/docs/tooldocs/windows/ktab.html.

Configuring NTLM Authentification from a Browser (Windows only)

To enable passwordless authentication, based on the user account, registered in a Windows domain, do the following:

- Add NTLM login module into the $(DIST_ROOT)/server/default/conf/login_module.properties file. For example, ru.runa.af.authenticaion.NTLMLoginModule=SUFFICIENT.
- Set domain name in $(DIST_ROOT)/server/default/conf/ntlm_support.properties file. For example, domain=MyDomain.
- Enable NTLM support in this file. For example, ntlm_supported=true.

By accessing the following page on the server on which the system is installed: http://<servername>/wfe/ntlmlogin.do (NTLM authentification can also work over HTTPS well), the users, registered in the specified domain and authorized to log into the system, will be authenticated.

To disable NTLM authentication, just turn off the parameters specified in steps 1 and 3 above.

Reboot the server after changing NTLM parameters.

How to Start Task Notifier

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

Run run.bat or run.sh.