i-net Clear Reports 2012

Security Guide

Configuration of Report and System Permissions

i-net software

1 Content

Security is an important feature for a reporting software. There are many conceivable cases where it would be necessary to restrict the rights on the reports. You may not want all users to be able to execute just any report and to thereby gain access to almost all data of a database. This can be configured using the Report Permissions.

It might also be necessary to limit the locations from which it is possible to take reports and execute them. This will prevent that someone can execute unknown reports on the report server. This can be configured using the Report Locations.

With the System Permissions it is possible to control the access to the modules of the Remote Interface.

2 Concepts of the Report Permissions

I-net Clear Reports does not have its own login database or connectors to any login system. It instead uses the login information from the web server.

2.1 Security Mechanisms

There are the following 2 security mechanisms available.

2.1.1 Report Locations

With report locations you can limit the locations from which reports can be loaded for execution. Only reports from the given locations and their sub-folders will be permitted to be executed. This does not require any user information. The default option is that it is possible to open reports from all local (server) and file locations.

2.1.2 Report Permissions

With report permissions it is possible to set access rights for specific users or user roles. This requires the login information of the current user. There are 3 ways to receive this login information:

- Login in the current web server
- A third party Single Sign-On System
- A login script for a frontend server.

The concepts are described below.

2.2 System Requirements

If you want to protect your reports or parts of the reports from anonymous access then you will need a web server with a working login configuration. Based on a error in the Java VM to handle sessions in a web browser, we advise to use the latest Java VM (Java 6 Update 10 or later, or Java 7).

3 Installation

3.1 General

The first step is to decide from where i-net Clear Reports is to receive the login information.

I-net Clear Reports supports several authentication methods (login types):

- **Automatic**: This default value allows i-net Clear Reports to determine the available method automatically. The server tries to request the login in the following order: External web server (a Login URL must be set), LDAP Authentication (if the Login URL starts with ldap or ldaps), Windows Authentication (if the server is running on a Windows operating system), PAM Authentication (if the server is running on a non Windows server and if it was possible to load the PAM library), Internal web server, Master Password.

- **External web server**: With this setting, a Login URL to an external web server can be defined. Users will then have to authenticate against this web server. If no (or no valid) Login URL has been defined, a fallback to Master Password authentication is performed.
- **LDAP Authentication**: With this login type a LDAP server will be used for authentication. If the URL is empty, the LDAP server will searched in the DNS of the current domain. Sample login URL: `ldap://MyLdapServer:389/` or `ldaps://MyLdapServer:636/` (with SSL).
- **Windows Authentication**: If the server runs on a Windows operating system, users will automatically be logged into the system with their Windows accounts. If the server's operating system is not Windows, a fallback to Master Password authentication is performed.
- **PAM Authentication**: This login type can be used if i-net Clear Reports runs on Linux or Mac OS X. It can be configured using the file "/etc/pam.d/reporting". If it does not exists, then "/etc/pam.d/passwd" will be used as fallback.
- **Internal web server**: If i-net Clear Reports is installed and running in an application server such as Apache Tomcat, the authentication system provided by the application server is used. For example: the default user administration of Apache Tomcat takes place in the file: tomcat-users.xml. If there is no authentication system active in the application server, a fallback to Master Password authentication is performed.
- **Master Password**: If no Login URL or application server with active authentication system is available, the user can log into the Remote-Interface using a password that is defined by the administrator. The password should have been defined during setup or has to be set with the first access to the Remote-Interface. Using Master Password authentication users will only have access to restricted modules and reports when they log into the Remote Interface. A direct authentication for reports or interfaces is not possible.

If the authentication method is set to "Automatic" or "External Webserver", a URL to a Login Script can be defined. You can use the default LoginServlet (single server only) or an external login script. If you enter the address like:

```
http://localhost/clearreports/LoginServlet
```

or

```
http://localhost/YourPath/login.aspx
```

then the browser should request login information from you. How this occurs depends on how you have configured your login for your web application server (BASIC, FORM, etc.). After you login in the browser you will receive an XML file like:

```
<properties>
 <entry key="username">scott</entry>
</properties>
```

If you receive an error then your configuration is wrong. In this case and if you use the LoginServlet then changing the web.xml in the reporting.war can be necessary depending on the used application server. See the documentation of your application server or servlet engine for more details.

If you use an external login script then you will need to enter the URL in your i-net Clear Reports configuration, accessible through the Configuration Manager in the category "Login/IP Filter".

Please note: If the login script administrates the user by domain **and** user name (as in "DOMAIN/User"), the permissions have to be configured the same way.

3.2 Report Server (Listener)

Report Permissions are available for the Report Server (Listener) as standalone application or service. But the Report Server supports the external login script, only (see: chapter Login Script). It does not support the LoginServlet.

3.3 Tomcat & JBoss

First, you will need to configure the realm of your Tomcat or JBoss server. By default, it is a XML based user database. You can set up this in the server.xml file. See the documentation of your web server for more details.

Secondly, you need to enable "Single Sign-On". This is disabled by default. You can do it if you add the following line to the server.xml:

```
 <Valve className="org.apache.catalina.authenticator.SingleSignOn"
/>
```

In the default server.xml this line is commented out, and need only be uncommented.

3.4 Jetty 5.0

The standard installation of Jetty does not deploy applications automatically. You have to create a new folder with the same name as the war file (e.g. `reporting` for `reporting.war`), inside the Jetty webapps folder (e.g. "/usr/share/jetty/webapps") and copy the file "reporting.war" into it. To deploy the content execute the command: "jar xf reporting.war". Jetty has to be restarted afterwards.

In the file "/etc/jetty/jetty.xml" the server configures a section to load all Web Applications from the Web Applications location above. As in Tomcat you have to add a realm like the following:

```
<Call name="addRealm">
  <Arg>
    <New class="org.mortbay.http.HashUserRealm">
      <Arg>i-net Clear Reports</Arg>
      <Arg>$$PASSWORDFILE$$</Arg>
    </New>
  </Arg>
</Call>
```

Please be sure that the the first ARG section is exactly the same as above – this is the name jetty use to match the realm to the application.

The $$PASSWORDFILE$$ is a plain text file containing the user name, password and groups. An example is located at "/etc/jetty/adminRealm.properties". It has the format:

```
<username>: <password>[,<rolename> ...]
```

3.4.1 Jetty with MySQL Authentication Backend

The Jetty application server supports the authentication via an database backend too. We will show how to do it for MySQL here. The advantage is, that you can insert, remove and change the values inside the database on run-time.

At first you have to change the realm from the example above to the following:

```
<Call name="addRealm">
  <Arg>
    <New class="org.mortbay.http.JDBCUserRealm">
      <Arg>i-net Clear Reports</Arg>
      <Arg>$$PASSWORDFILE$$</Arg>
    </New>
  </Arg>
</Call>
```

Instead of the `$$Passwordfile$$` from above you need a file with the following content:

```
jdbcdriver = com.mysql.jdbc.Driver
# The following has to be replaced by your settings
# ##########################################
url = jdbc:mysql://$$SERVER$$/$$DATABASE$$
username = crystalJetty
# Replace the wollowing to your Password
password = $$PASSWORD$$
# ##########################################
# Leave this, those are the table setings
usertable = users
usertablekey = ID
usertableuserfield = username
usertablepasswordfield = pwd
roletable = roles
roletablekey = ID
roletablerolefield = role
userroletable = user_roles
userroletableuserkey = user_ID
userroletablerolekey = role_ID
cachetime = 300
```

The file defines the connection to your database (`jdbcdriver`, `url`, `user name`, `password`) and the layout of the database. It is important to make sure, that you have the .jar file of the `jdbcdriver` in your class-path. The simplest way is to copy the driver library (jar) into the `ext` directory of the Jetty installation (e.g. "/usr/share/jetty/ext/").

You can now create the database tables with the following Statement:

```
CREATE TABLE IF NOT EXISTS `roles` (
  `ID` int(11) NOT NULL auto_increment,
  `role` varchar(20) NOT NULL,
  PRIMARY KEY (`ID`)
);
-- --------------------------------------------------------
CREATE TABLE IF NOT EXISTS `users` (
  `ID` int(11) NOT NULL auto_increment,
  `username` varchar(20) NOT NULL,
  `pwd` varchar(255) NOT NULL,
  PRIMARY KEY (`ID`)
);
-- --------------------------------------------------------
CREATE TABLE IF NOT EXISTS `user_roles` (
  `user_ID` int(11) NOT NULL,
  `role_ID` int(11) NOT NULL,
  PRIMARY KEY (`user_ID`,`role_ID`)
);
```

The tables themselves should be self-explanatory. Only the linking of roles to user-names may be strange: Put the IDs (integer) of the desired user and role into one row of the `user_roles` table.

Please take care of the following: Jetty support plain-text and encrypted passwords from the users table. To Create an entry with an encrypted MD5 password, please use a command like the following:

```
INSERT INTO users (`username` , `pwd` )
VALUES ( '$$USERNAME$$', CONCAT('MD5:', MD5( '$$PASSWORD$$' )) );
```

3.5 Oracle Application Server

Before installing i-net Clear Reports, you must first make sure that the following three files are not write-protected, since i-net Clear Reports needs to edit these files during installation:

```
<Oracle Home>/j2ee/home/config/server.xml
<Oracle Home>/j2ee/home/config/application.xml
<Oracle Home>/j2ee/home/config/http-web-site.xml
```

Make sure your server is running, and run the i-net Clear Reports installation. Now, you must manually edit the file web.xml of i-net Clear Reports, found in:

```
<Oracle Home>/j2ee/home/applications/crystal/crystal/WEB-INF
```

In this file, you must add the roles of the users who will be using for i-net Clear Reports to the end of the web.xml file using the <security-role> tag. Since you will be defining the permissions for various users and roles from within i-net Clear Reports, it is recommended to use as global a group as possible here, but if you wish, you can also use more limited roles. The role(s) you enter here define which users may access the i-net Clear Reports servlet at all.

Assuming you have two groups, "users" and "guests", and you want to provide a certain amount of access to users from these two groups, an example would be:

```
   ...
   <security-role>
      <role-name>users</role-name>
   </security-role>
   <security-role>
      <role-name>guests</role-name>
   </security-role>
</web-app>
```

See the Oracle AS documentation for general information on how to set up users, groups, and roles.

3.6 BEA Weblogic Application Server

Before you configure the permissions for i-net Clear Reports, you need to specify the users for the sign on in the BEA Weblogic application server. You can manage the users in the "Users and Groups" tab of the security realms section. See the BEA Weblogic 10 AS documentation for general information on how to set up users, groups, and roles.

Configuring or changing the permissions for i-net Clear Reports requires a new deploying of the reporting.war archive in the BEA Weblogic AS. The BEA Weblogic AS provide four security models, which specifies the location where the roles and policies are defined.

- DD Only: Use only roles and policies that are defined in the deployment descriptors.
- Custom Roles: Use roles that are defined in the Administration Console; use policies that are defined in the deployment descriptor.
- Custom Roles and Policies: Use only roles and policies that are defined in the Administration Console.
- Advanced: Use a custom model that you have configured on the realm's configuration page.

At this point only the first model "DD Only", which is also the default security model in the BEA Weblogic AS, will be covered. For more informations on the security models see the security documentation of the BEA Weblogic 10 AS.

By using the "DD Only" model you must define the users and roles in the deployment descriptors.

You must manually edit the web.xml of i-net Clear Reports, found in the WEB-INF directory inside the reporting.war archive. In this file, you must add the roles of the users who will be using i-net Clear Reports to the end of the web.xml file using the <security-role> tag. Since you will be defining the permissions for various users and roles from within i-net Clear Reports, it is recommended to use as global a group as possible here, but if you wish, you can also use more limited roles. The role(s) you enter here define which users may access the i-net Clear Reports servlet at all.

Assuming you have two groups, "users" and "guests", and you want to provide a certain amount of access to users from these two groups, an example would be:

```
   ...
   <security-role>
      <role-name>users</role-name>
   </security-role>
   <security-role>
      <role-name>guests</role-name>
   </security-role>
</web-app>
```

Now you have specified the roles, but in the BEA Weblogic AS you must also specify which users are members in these roles. This must be done be editing the file weblogic.xml, also found in the WEB-INF directory inside the reporting.war archive. In this file, you must add the users to the roles using the <security-role-assignment> tag.

Assuming you have three users, "manager", "user", and "guest" specified in the security realm of the Weblogic AS, and "manager" and "user" are members of the "users" role, and "user" and "guest" are members of the "guests" role, an example would be:

```
   ...
   <security-role-assignment>
  <role-name>users</role-name>
  <principal-name>user</principal-name>
  <principal-name>manager</principal-name>
 </security-role-assignment>
 <security-role-assignment>
  <role-name>guests</role-name>
  <principal-name>guest</principal-name>
  <principal-name>user</principal-name>
 </security-role-assignment>
</weblogic-web-app>
```

The principal names must be the same as the user names specified in the security realm.

Keep in mind that only users assigned to roles are accessible to i-net Clear Reports. Any permission check on users not assigned to any role will fail.

After changing these two files you must create the reporting.war file again and deploy it to the BEA Weblogic AS.

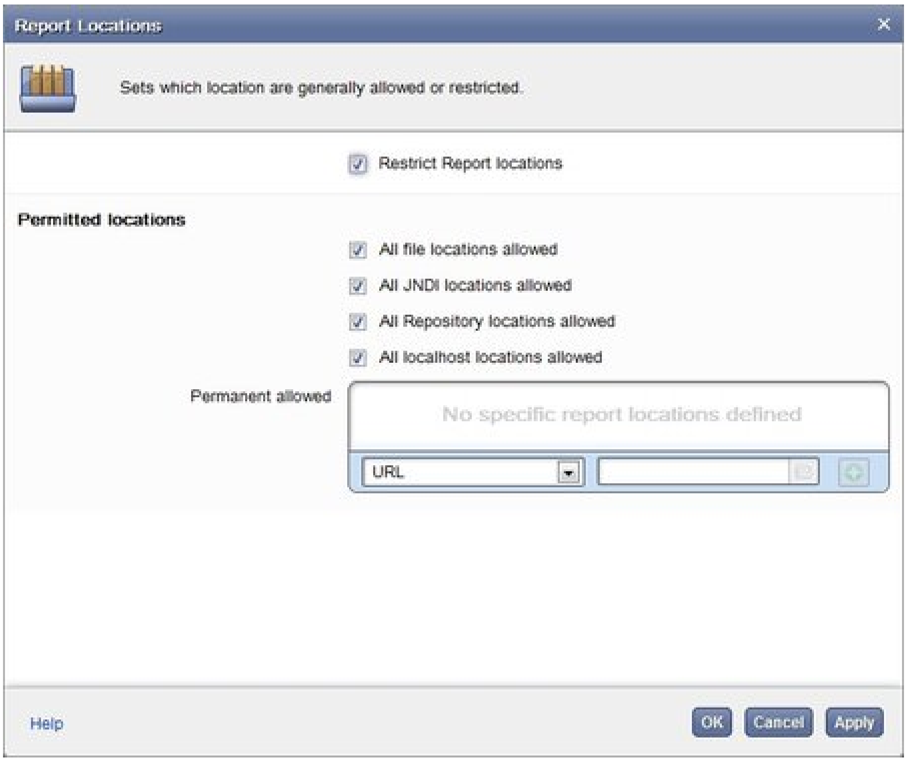
For more information on deploying web applications to the BEA Weblogic AS contact your BEA Weblogic application server administrator or check the

Deploying Applications to Weblogic Server documentation.

4 Activation of the Security Features

4.1 Activation of Report Locations or Report Permissions

You can enable the check of the Report Locations or Report Permissions in the appropriate i-net Clear Reports - Configuration Manager category.



Report Locations

Sets which location are generally allowed or restricted.

☑ Restrict Report locations

Permitted locations

☑ All file locations allowed
☑ All JNDI locations allowed
☑ All Repository locations allowed
☑ All localhost locations allowed

Permanent allowed: No specific report locations defined

URL

Fig. 1: Configuration Manager – Report Locations

If you enable the check of the Report Locations then you should enable "All file locations allowed", "All localhost locations allowed" or you must add at least one URL from that reports can be loaded by i-net Clear Reports.

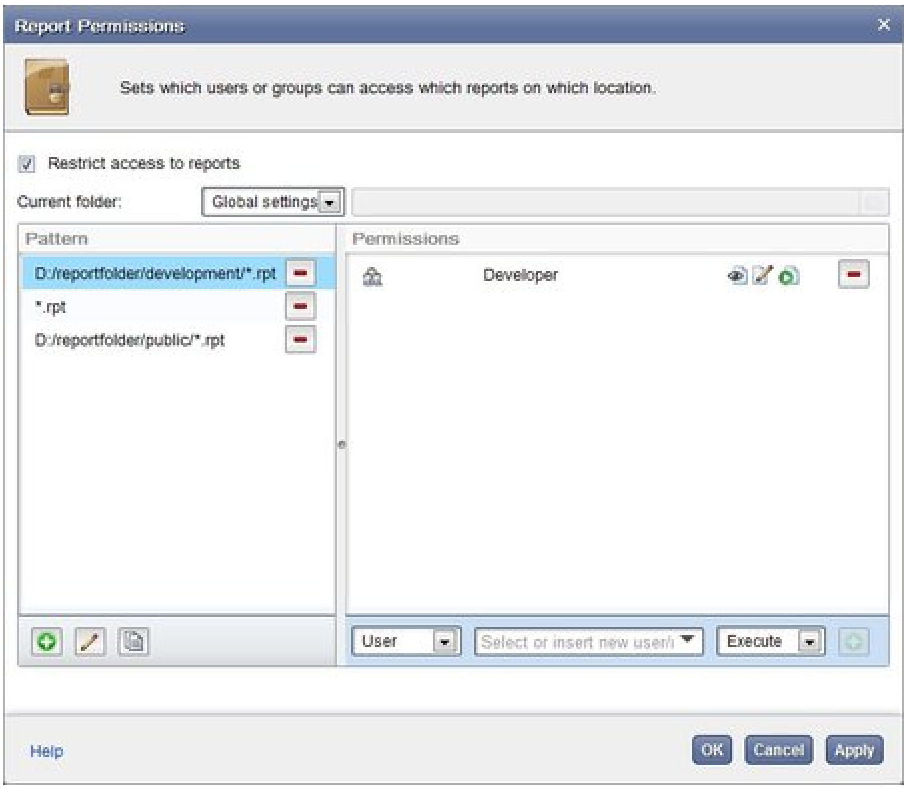
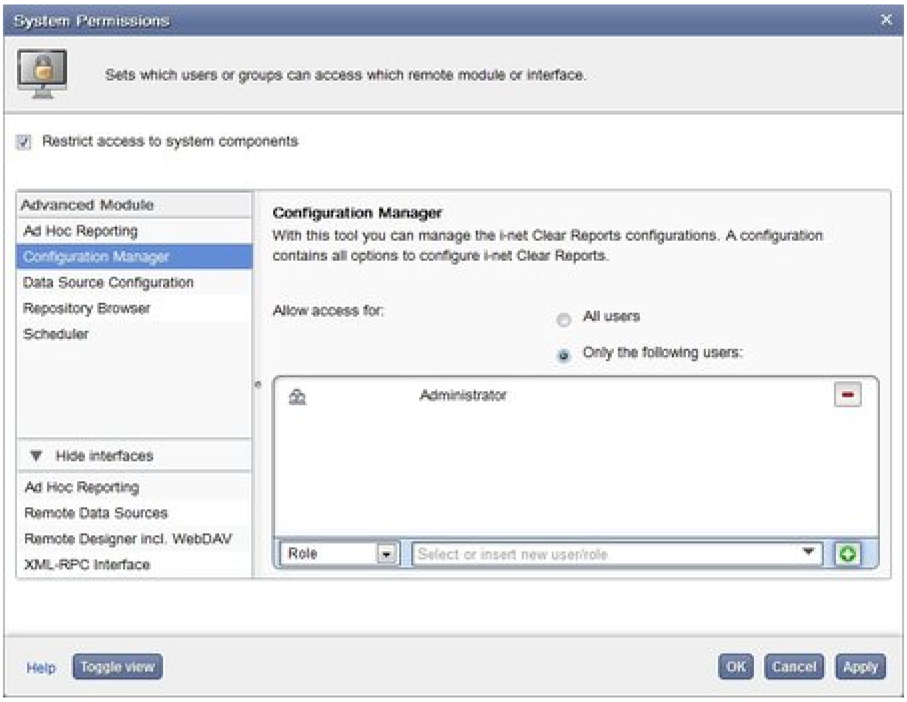

Report Permissions
Sets which users or groups can access which reports on which location.
☑ Restrict access to reports
Current folder:
Global settings
Pattern
D:/reportfolder/development/*.rpt
*.rpt
D:/reportfolder/public/*.rpt
Permissions
Developer
User
Select or insert new user/r
Execute
Help
OK
Cancel
Apply

Fig. 2: Configuration Manager – Report Permissions

If you enable the check of the Report Permissions then you should configure at least one report pattern (such as "*.rpt") and a user or a group (role) for this pattern so that it is possible to execute reports on the i-net Clear Reports report server using this configuration.

4.2 Activation of the System Permissions

You can enable the "Restrict access to system components" in the i-net Clear Reports - Configuration Manager category "System Permissions".



System Permissions

Sets which users or groups can access which remote module or interface.

☑ Restrict access to system components

Advanced Module
- Ad Hoc Reporting
- Configuration Manager
- Data Source Configuration
- Repository Browser
- Scheduler

▼ Hide interfaces
- Ad Hoc Reporting
- Remote Data Sources
- Remote Designer incl. WebDAV
- XML-RPC Interface

Configuration Manager

With this tool you can manage the i-net Clear Reports configurations. A configuration contains all options to configure i-net Clear Reports.

Allow access for: ○ All users ● Only the following users:

Administrator

Role | Select or insert new user/role

Fig. 3: Configuration Manager – System Permissions

If you enable the restrict access then you should configure at least one user or group for the module "Configuration Manager" so that it is possible to execute this module in the Remote Interface.

5 Features

5.1 System Permissions

With system permissions you can restrict the access to certain parts of the system such as modules of the Remote-Interface and server interfaces.

The system permissions can be administrated using the Remote Configuration module of the Remote-Interface. The permissions are divided into the following groups:

Advanced Module

The remote modules are web applications in the Remote-Interface of i-net Clear Reports. Currently the following permissions for the Remote-Interface are available:

- Ad Hoc Reporting: allows the user or role to access the Ad Hoc Reporting module.
- Configuration Manager: allows the user or role to access the remote Configuration Manager.
- Data Source Configuration: allows the user or role to access the remote Data Source Configuration Manager.
- Repository Browser: allows the user or role to access the remote Repository Browser.
- Scheduler: allows the user or role to access the Scheduler module.

Interfaces

I-net Clear Reports continuously provides new interfaces to administrate or use the report server. Currently the following permissions for interfaces are available:

- Ad Hoc Reporting: allows the user or role to access the Ad Hoc Applet (Context /adhoc) and to use the remote Report Wizard of the Designer.
- Remote Data Sources: allows the user or role to retrieve the remote Data Sources of the server via the Remote Designer or the ad hoc feature.
- Remote Designer incl. WebDAV: allows the user or role to access the Webdav interface (Context /repository) and the usage of the Remote Designer.
- XML-RPC Interface: allows the user to make use of the XML-RPC interface. For an overview over which XML-RPC methods are offered, refer to the XML-RPC API page (Context /xmlrpc).

If the configuration has no Login URL set, you can log-in to the Remote-Interface as system administrator. This user has no restrictions, each module will be visible and can be used.

5.2 Report Locations

By default, any report in any location on your file server can be executed on your i-net Clear Reports server. That may make things easy for your employees and customers, but it can be a security problem.

To solve the problem and reduce the security risks, you can grant access for certain report locations only. The report locations consist of a list of folders and URL's which contains report files that can be executed.

5.3 Report Permissions

With the report permissions you can restrict the access to reports depending on the login user. It is possible to store the report permissions either in an i-net Clear Reports configuration or as files in a folder, similar to an .htaccess file. Depending on the location the permissions are either specific only to the used configuration or specific to the folder for which they were created. Whether you use "Global settings", "URL", "Filesystem" or "Repository" depends on your needs.

The permissions are additive, which means if a permission is granted in any way (Filesystem or global settings, for a role or user, or all roles or users) the user will have access, no matter what other settings will say.

The following permissions can be set:

- read
- write
- execute

5.3.1 Global Settings

Advantages:

- Patterns are specific to the configuration. It is possible to deploy the permission settings together with the configuration.

- All configured permissions are displayed in one list.
- It is possible to give a user and/or a group the permissions for multiple folders or URL's in one pattern.
- With this you can easily grant permissions to admins or super-users and you don't need to grant permissions for multiple folders.

Disadvantages:

- It can be difficult to configure, because the patterns could be more complex if they include folder names.
- There can be security problems because a single report file can be accessed with different URL's which may not all covered by the specified pattern.

A pattern can contain characters, the asterisk symbol (*) as a wildcard for any number of symbols, the question mark (?) as a wildcard for a single symbol or slashes (/) as a separator for folders. After a pattern has been created, you have to define at least one user name or group for the pattern. For users and group you can use the asterisk symbol (*) as a wildcard, also. For instance if the group "admin" should have access to all reports, you must create a new global pattern "*" for all reports and a add the group "admin" to this pattern.

5.3.2 Folders

Advantages:

- The patterns are specific to the folder containing the permissions settings.
- If the permission settings are copied or moved to another folder, then the permissions are specific to that folder, also.
- Different URL to the same report template are no problem.
- The permissions are independent from the used configuration.

Disadvantages:

- It is not possible to see all permissions in one list. To see or modify the permissions, it

is necessary to open the folder or URL.
- If a user has the same permissions for multiple folders or URL's then it is necessary to configure the permissions for each folder or URL.

If permissions exist for the specified folder or URL they will be loaded automatically. You can save the changed permissions in the current folder or you can save the permissions into a different folder with the "Save As" button. This is necessary if you have created or modified permissions for an URL because this permissions file must be manually copied onto the web server.

Saving your permission settings in a folder causes a "reportPermissions.xml" file to be saved into this folder – this means the folders you wish to set permissions for must not be write-protected.

5.3.3 Formula Feature

The report permissions can be used within reports using formula functions. The following 3 functions are available in the node "Security Functions" of the Formula Editor:

- WebUserName
- IsWebUserInRole(String)
- FireAccessDenied

Using these functions in formulas, you could filter records in the Record Selection Formula, hide fields or sections, etc., depending on the WebUserName and/or the Role of the current user. With the FireAccessDenied function it is even possible to cancel the report execution if the current user is not logged on or is not a member of the specified roles (groups). It is possible to test these functions within the i-net Designer if the report permissions have been configured in the configuration used by i-net

Designer and if a user name and/or roles have been specified in the "Designer Options" dialog.

5.3.4 i-net Designer Properties

To test the security functions with i-net Designer, you can set the current web user name and its roles in the category "Virtual Permissions" of the "Designer Options..." (see menu "Options | Designer Options...") . Here you can enter the simulated user name and user roles.

5.3.5 Repository Permissions

The Report Repository is a feature of i-net Clear Reports Plus which allows streamlined and simple central storage and execution of reports on the report server.

The Report Repository makes use of every security setting mentioned above. In addition to configuring the report permissions in the Configuration Manager, users with appropriate privileges can also manage them directly from the Repository Browser.

For detailed information please have a look at the Repository Guide.

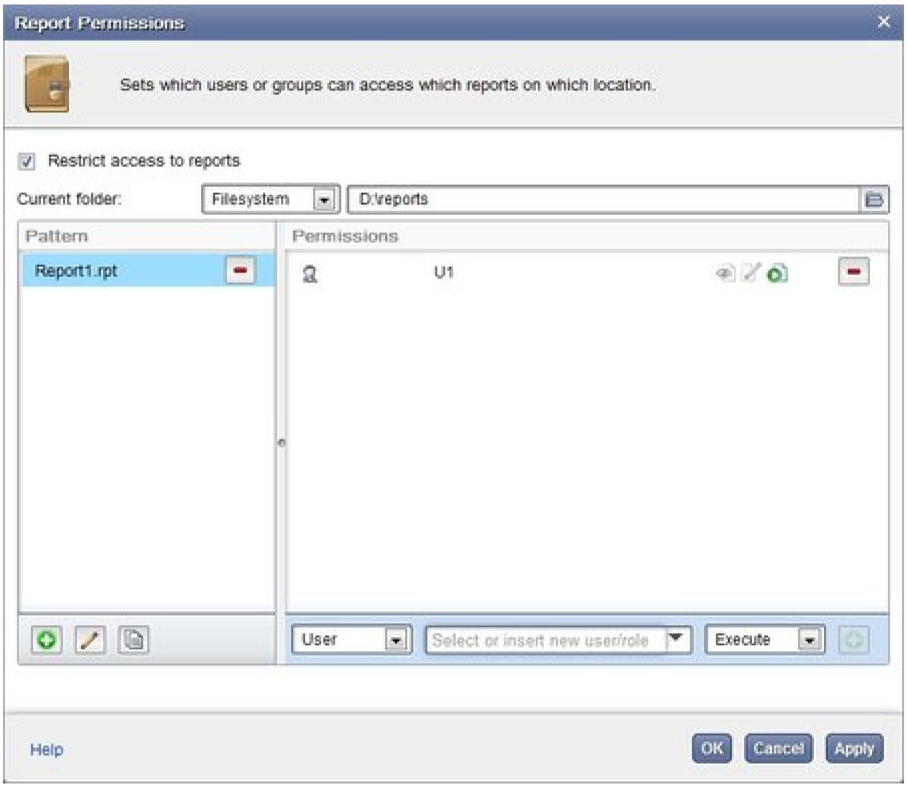
6 Example Scenarios

6.1 Security Settings for Specific Users

The first scenario describes a simple example to explain how to set report permissions for individual users.

Let's assume you'd like to give a user "U1" the right to execute a report called "Report1.rpt" in the folder "D:\reports". To do this, you first start the Configuration Manager and choose the category "Report Permissions".

In the case that you have not yet activated the option "Restrict access to reports", you need to do this now. By doing this, you are preventing all users with no access rights from executing any report.



Report Permissions

Sets which users or groups can access which reports on which location.

☑ Restrict access to reports

Current folder: Filesystem D:\reports

Pattern	Permissions
Report1.rpt	U1

Fig. 4: Report Permissions for user "U1"

To now give user "U1" the permissions to run a report called "Report1.rpt" in the folder "reports", you choose the "Filesystem", and enter the file location in the text box (in this case, "D:\reportfolder"). You can also browse the file system to choose the path.
If there are existing report permissions for the chosen path or URL, they are automatically loaded.

Because there are currently no report permissions set for this folder, the user does not have the permission to run the report. Click on the "Add Pattern" button and enter the name of the report in the list entry. After doing this, the user "U1" must also get the permission for this specific report. Select the list entry "Report1.rpt", select "User", enter the user's name "U1" and select the right "Execute". Finally, click on the "Add" button to store the entered permission. This causes the given user "U1" to have access to this report.

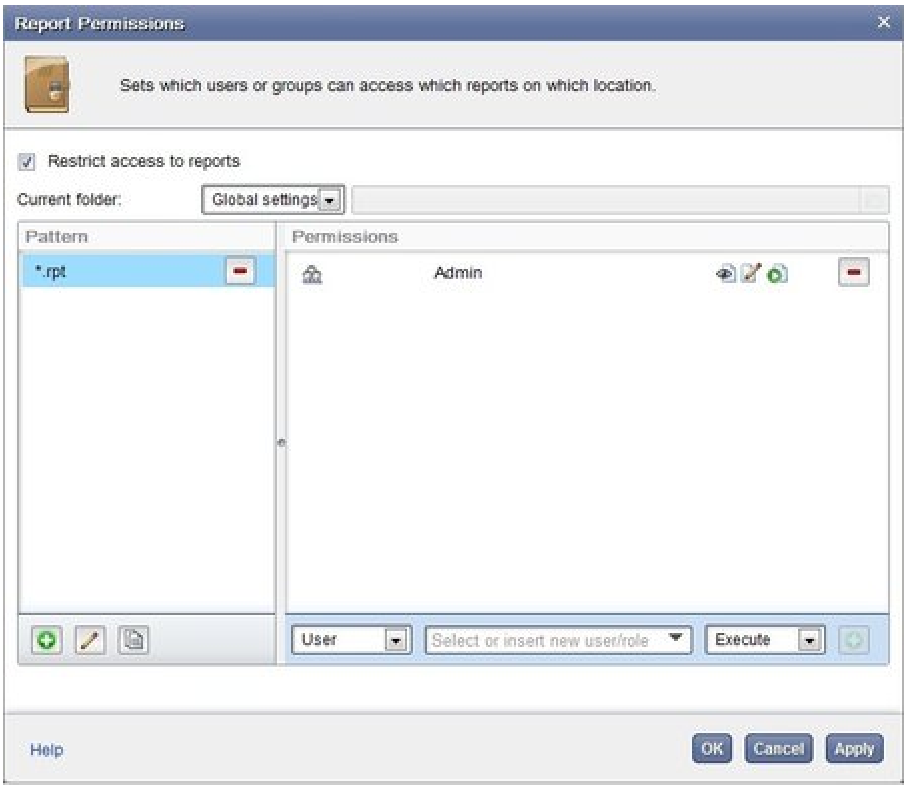
6.2 Security Settings for Groups

In the second example, we want to give report permissions to the groups "Admin" and "Developer". The reports are located in subdirectories of the folder "C:/reportfolder". The subdirectories are called "public", "development" and "private" and contain reports which are only supposed to be able to be executed by users from the various groups. The group "Admin" should have permissions to run all reports, while the group "Developer" is only supposed to be able to run reports in the folders "public" and "development". All other users are to be able to access reports in the "public" folder.

In order to give the "Admin" group unlimited rights, select the current folder "Global settings" and add the report patten "*.rpt". After this select "Role" and enter the name "Admin" to this pattern.

Give the group "Developer" the permissions to the folders "public" and "development", by switching to the "Filesystem" and entering the URL or path to the "developer" folder, and adding the pattern "*.rpt", followed by the group "Developer" for this pattern. Store the permissions in the folder.

Now enter the path to the "public" folder, add the pattern "*.rpt" again and add a user to this pattern name "*". This gives any user (symbolized by the "*") the permissions to run any report in this folder. Don't forget to save your changes by clicking on "OK" button.



Report Permissions

Sets which users or groups can access which reports on which location.

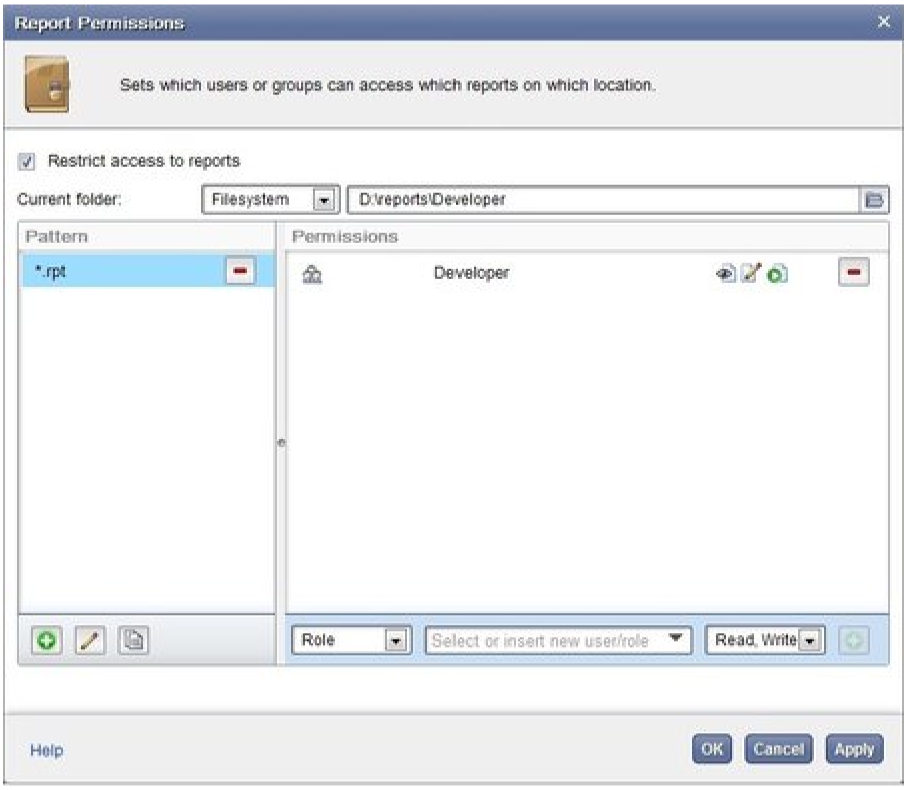
☑ Restrict access to reports

Current folder: Global settings

Pattern	Permissions
*.rpt	Admin

User | Select or insert new user/role | Execute

Fig. 5: Group "Admin" can run all reports



Report Permissions

Sets which users or groups can access which reports on which location.

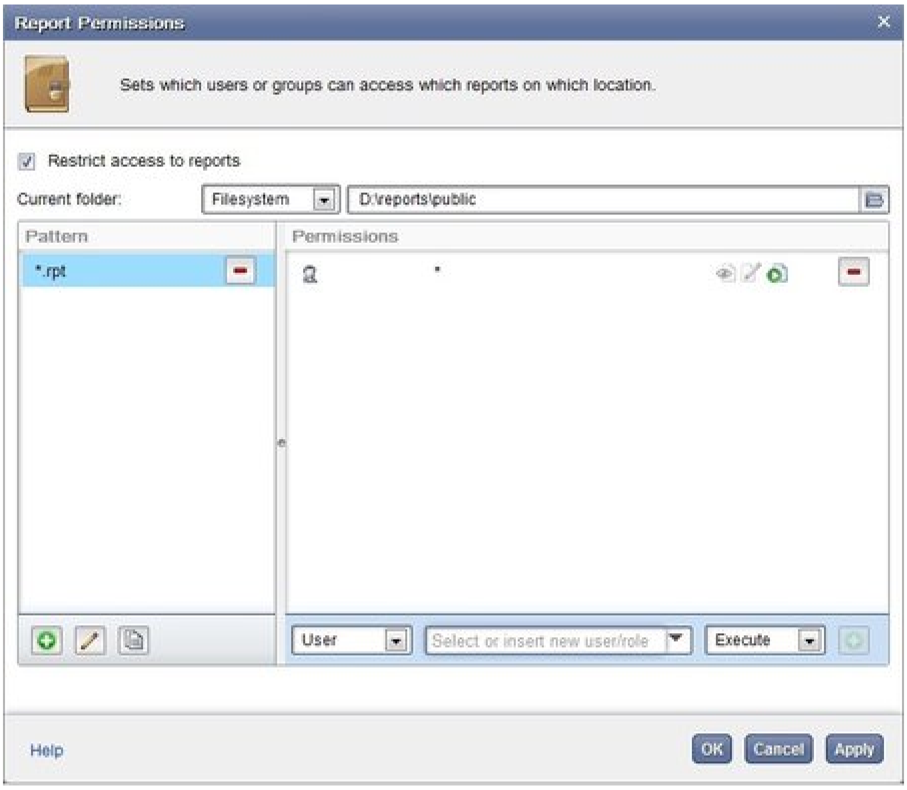
☑ Restrict access to reports

Current folder: Filesystem | D:\reports\Developer

Pattern	Permissions
*.rpt	Developer

Role | Select or insert new user/role | Read, Write

Fig. 6: Group "Develper" can run reports in the folder "Developer"



Report Permissions

Sets which users or groups can access which reports on which location.

☑ Restrict access to reports

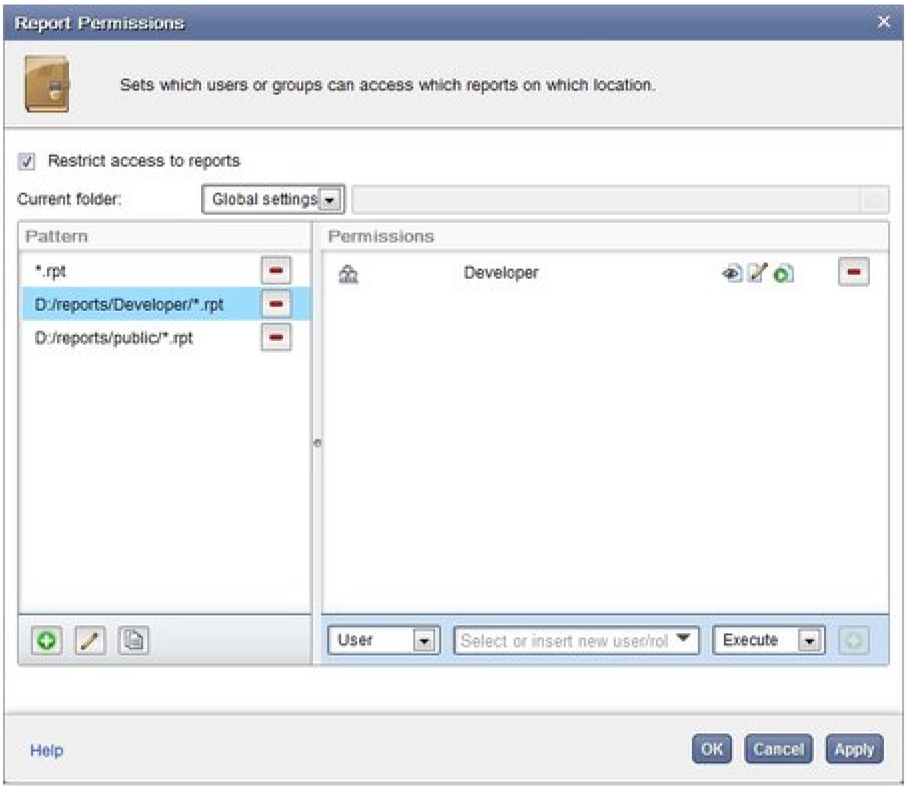
Current folder: Filesystem D:\reports\public

Pattern	Permissions
*.rpt	*

User | Select or insert new user/role | Execute

Fig. 7: All users can run reports in folder "public"

It is also possible to configure the same report permissions in the "Global settings". In this case you have to set the path of the folder in the pattern.



Report Permissions

Sets which users or groups can access which reports on which location.

☑ Restrict access to reports

Current folder: Global settings

Pattern	Permissions
*.rpt	Developer
D:/reports/Developer/*.rpt	
D:/reports/public/*.rpt	

Fig. 8: Group - Report Permissions as "Global Settings"

6.3 Filtering with a Record Selection Formula

If you have data in your database tables for different users and you want every user to only be able too see his or her own data then you can filter the data. A typical example is a ticket or billing system.

If you have a table column with the user names then you could set the Record Selection Formula to something like:

```
if WebUserName() = "" then
 fireAccessDenied()
else
 {YourTable.username} = WebUserName();
```

The call of the function fireAccessDenied is used to force a login for the user if they are not currently logged in.

7 Frontend / Backend Architecture

If you have a frontend / backend architecture (for example Apache with Tomcat Plug-In or IIS with any Servlet Engine) then login typically occurs in the frontend web server. The backend server with i-net Clear Reports has no access to the login information. Such architecture can be run on a single system or might span over multiple systems. There are 3 possible solutions.

7.1 Same Login Base

If your frontend and backend web server use the same login base (for example an LDAP dictionary) then the browser will resend the same login information if you switch between content directly from frontend server and backend server. A logout will not be possible.

As an alternative you can duplicate the login base if it is only a small number of users.

7.2 Single Sign-On Systems (SSO)

There are a lot of Single Sign-On systems on the market. Many of them support popular servlet engines like Tomcat. If you have no preference then you could have a look at JOSSO, an open source SSO.

7.3 Login Script

Another solution is a simple script that presents the needed login information. The requirement is that the backend server can access the frontend server (with no firewall problems). You need only:

- copy the script to your frontend server.
- Set up the security settings on this script (Authorization type, etc.)
- Test the script by calling it with a browser like:

```
http://<YourServer>/login.aspx?abc=
```

After you have logged in you should see an XML result like:

```
 <?xml version="1.0" encoding="UTF-8"?>
 <!DOCTYPE properties SYSTEM
"http://java.sun.com/dtd/properties.dtd">
 <properties>
 <entry key="username">YourUserName</entry>
 <entry key="abc">False</entry>
 </properties>
```

- Enter the URL of the script in the Configuration Manager of i-net Clear Reports.

In the following you can find a sample script for ASP.NET and JSP.

Attention: You have to make sure that the user name is always returned in way that is unique to the system (e.g. always make it lowercase - even if the user logs in with uppercase letters) - the reason behind this is: i-net Clear Reports supports case-insensitivity in every permission checking context, but it will respect folder names case-sensitive. Now if you had a user named "JonDoe" and he logs into the system with "johndoe" there would be two different home directories in the repository though you meant the same user.

7.3.1 ASP.NET

Create a file with the extension *.aspx (e.g. login.aspx) in the IIS and copy the following script into this file. Only the athentication method "Basic authentication" need to be enabled for this .aspx file in the IIS configuration.

It is required, that Microsoft .Net Framework 2.0 is installed.

```
 <%@ Page Language=VB ResponseEncoding="utf-8" %>
 <%
 If User.Identity.Name = "" Then
  Response.Write( "401 Access Denied" )
  Response.Status = "401 Access Denied"
  Response.End
 end if
 Response.ContentType = "text/xml; charset=utf-8"

 Response.Write( "<?xml version=""1.0"" encoding=""UTF-8""?>" & Chr
(10))
 Response.Write( "<!DOCTYPE properties SYSTEM "
"http://java.sun.com/dtd/properties.dtd"">" & Chr(10))
 Response.Write( "<properties>" & Chr(10))

 Response.Write( "<entry key=""username"">" & Server.HTMLEncode(
User.Identity.Name) & "</entry>" & Chr(10) )

 Dim Key
 For Each Key In Request.QueryString
  if Key <> "" Then
   Response.Write( "<entry key=""" & Server.HTMLEncode(key) & """>"
& Server.HTMLEncode(User.IsInRole(key)) & "</entry>" & Chr(10))
  End If
 Next

 For Each Key In Request.Form
  if Key <> "" Then
   Response.Write( "<entry key=""" & Server.HTMLEncode(key) & """>"
& Server.HTMLEncode(User.IsInRole(key)) & "</entry>" & Chr(10))
  End If
 Next
 Response.Write( "</properties>" & Chr(10))
 %>
```

7.3.2 JSP

Create a file with the extension *.jsp and copy it in any web context.

```
 <%@page language="java" contentType="text/xml; charset=utf-8"
pageEncoding="UTF-8"
 import="java.security.Principal"
 import="java.io.*"
 import="java.util.*"

 %><%!
 public static String encode(String s){
  StringBuilder out = new StringBuilder();
  for(int i=0; i<s.length(); i++){
   char c = s.charAt(i);
   if(c > 127 || c=='"' || c=='<' || c=='>'){
    out.append("&#"+(int)c+";");
   }else{
    out.append(c);
   }
  }
     return out.toString();
 }

 %><?xml version="1.0" encoding="UTF-8"?>
 <!DOCTYPE properties SYSTEM
"http://java.sun.com/dtd/properties.dtd">
 <properties>
 <%
  Principal p = request.getUserPrincipal();
  if( p != null ){
   out.write( "<entry key=\"username\">" + encode( p.getName() ) +
"</entry>\n" );
  }

  Enumeration e = request.getParameterNames();
  while(e.hasMoreElements()){
   String key = (String)e.nextElement();
   key = new String( key.getBytes("ISO8859_1"), "UTF8");
   out.write( "<entry key=\"" + encode( key ) + "\">" +
request.isUserInRole(key) + "</entry>\n" );
```

```
 }
%>
</properties>
```

7.3.3 PHP with Apache

Create the files .htaccess, .htpasswd and .htgroups. This file can look like this:

.htaccess

```
# dont allow htaccess and htpasswd
<Files ~ "^.(htaccess|htpasswd)$">
deny from all
</Files>

# .htpasswd contains the password and users
AuthUserFile /opt/lampp/htdocs/.htpasswd
AuthGroupFile /opt/lampp/htdocs/.htgroups
AuthName "Please enter your ID and password"
AuthType Basic
require valid-user
```

.htpasswd - A user test with password test.

```
test:WCt/yYmXR2kLA
```

.htgroups

```
admin: test
```

Create a php login file with the follow content:

```
 <?php
   // This is the .htgroups file - it needs read permission!
   $AuthGroupFile = file("/path/to/.htgroups");

   // If the Apache has AUTH Info, set them for PHP as well
   if (!empty($_SERVER['AUTH_USER']))
   {
      $_SERVER['PHP_AUTH_USER'] = $_SERVER['AUTH_USER'];
      $_SERVER['PHP_AUTH_PW'] = $_SERVER['AUTH_PASSWORD'];
   } else if (!empty($_SERVER['REMOTE_USER'])){
     $_SERVER['PHP_AUTH_USER'] = $_SERVER['REMOTE_USER'];
   }

   // Check if someone has authenticated - if not, do another Basic
Authentication
   if (!isset($_SERVER['PHP_AUTH_USER'])) {
     header('WWW-Authenticate: Basic realm="Server Authentication"');
     header('HTTP/1.0 401 Unauthorized');
     echo 'Access Denied';
     exit;
   }

   // here could be some checking if the user is OK agianst a
database or something
   // alternatively this can be done via .htaccess in apache

   $return = '';
   $return .= '<entry key="username">' . strtolower(htmlentities(
$_SERVER['PHP_AUTH_USER'])) . "</entry>\n";

   foreach ( $_REQUEST AS $key => $value ) {
     $status = !preg_grep("/$key:.*?\s" . htmlentities($_SERVER[
'PHP_AUTH_USER']) . "(\s.*?)?$/", $AuthGroupFile) ? 'false' :
'true';
     $return .= '<entry key="' . htmlentities($key) . '">' .
$status . "</entry>\n";
         }
```

```
    header('Content-Type: text/xml; charset=utf-8');
    print <<<OUTPUT
  <?xml version="1.0" encoding="UTF-8"?>
  <!DOCTYPE properties SYSTEM
 "http://java.sun.com/dtd/properties.dtd">
  <properties>
  $return
  </properties>
  OUTPUT;
  ?>
```

7.4 Scripting in the Frontend

An alternative to the previous methods is a script in the frontend server. If you use a proxy in your frontend server that is based on a script like ASP then you can extend the script to enforce report permissions. This is not possible with a ISAPI or an AP13 connector. If you prevent your reports with a frontend script then it is not possible to use the features of the report permissions. In general there are 2 mechanisms:

- folder permissions
- file permissions

We will demonstrate this using IIS as an example. By default the IIS sets the permissions on the NTFS level. It is not possible for the backend i-net Clear Reports server to check the NTFS rights. Therefore you will have to check it in the IIS itself. This is not possible with the ISAPI extension but with our ASP.NET solution. ASP.NET must be enabled in your web server, note that this is not the default option.

7.4.1 Folder Permissions with IIS

This our recommended solution for a frontend script. You have set the security on an folder, for example "management-reports". It should be possible for the logged in user to only execute reports that are included in this folder. You simply need do the follow steps:

- Activate ASP.NET for your IIS
- Install the i-net Clear Reports - Standalone Server as Service
- Create the folder "management-reports"
- Set the rights to the folder that only your management and the system can access. Copy the files default.aspx and clearreports.vb in that folder. Also, you can rename the file default.aspx if you already have a default document.
- Activate the security check in the default.aspx. By default it is comment out.
- Copy the report templates into the folder "management-reports" or in it's sub folders.

After this you can request "report1" as PDF with following URL:

```
http://<YourServer>/.../management-reports?report=report1.pdf
```

After the login you should receive a login request. After you have logged in you can see the PDF file in the browser.

7.4.2 File Permissions with IIS

The following is not recommended because you can easily make a mistake. Also it is not possible to work with format extensions. Instead you will need to set the "init" parameter.

The installation is like the previous sample without the requirements of a specific folder. In the file default.aspx you will need to test if you can open the file with the current user settings. You would need to program this yourself. This can look like the following pseudo code:

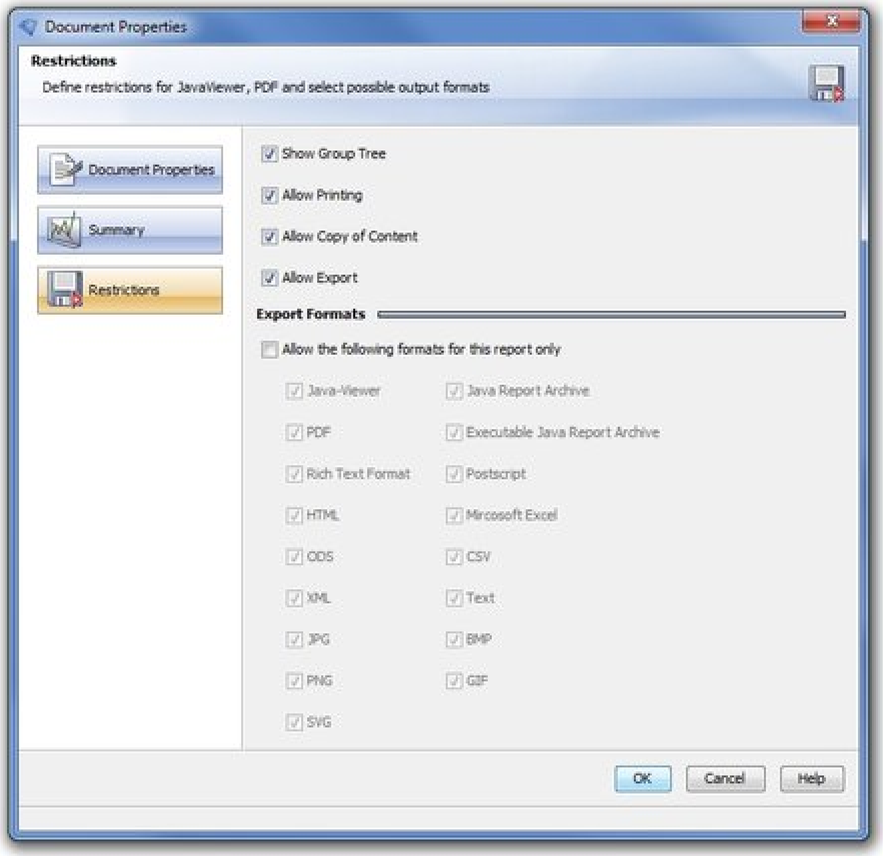
```
If report = "major.rpt" And username != "Admin" then
 // Access Denied
End If
```

8 Other Security Options

8.1 Restrictions

Note that i-net Clear Reports also has some other security options not related to a login. These are located in the category "Restrictions" of the dialog "Document Properties" in i-net Designer and include the following:

- Show Group Tree
- Allow Printing
- Allow Copy of Content
- Allow Export
- Allow the following output formats for the report



Restrictions

Define restrictions for JavaViewer, PDF and select possible output formats

☑ Show Group Tree

☑ Allow Printing

☑ Allow Copy of Content

☑ Allow Export

Export Formats

□ Allow the following formats for this report only

☑ Java-Viewer	☑ Java Report Archive
☑ PDF	☑ Executable Java Report Archive
☑ Rich Text Format	☑ Postscript
☑ HTML	☑ Mircosoft Excel
☑ ODS	☑ CSV
☑ XML	☑ Text
☑ JPG	☑ BMP
☑ PNG	☑ GIF
☑ SVG	

Fig. 9: i-net Designer - Document Properties

8.2 Allow unknown Data Sources

For security reasons the property "Allow unknown Data Sources" can be disabled. You can find it in to the configuration category "Behaviour".

If this property is disabled, then it is not possible to execute reports containing an unknown datasource. A data source is unknown, if no data source configuration exists with this name on the report server. This will prevent that each user has the possibility to load data from external databases.