



## **\*\*NEW\*\* Cizer.Net Reporting 4.0** **CNR Security Web Service & Sample Application**

---

### **Introducing Cizer.Net Reporting Security Web Service!**

In expanding our Cizer.Net reporting toolset, we have released several web services with the new version of our reporting software, version 4.0. This Cizer .Net Security web service will allow other software products and services to communicate and interface with Cizer.Net Administration features from their own custom applications and solutions.

To help understand the functionality of this new CNR Security Web Service, a pre-built application has been developed that incorporates the various aspects of the web service. It should provide developers with clear examples of how to use it and how to implement it in some of their own solutions.

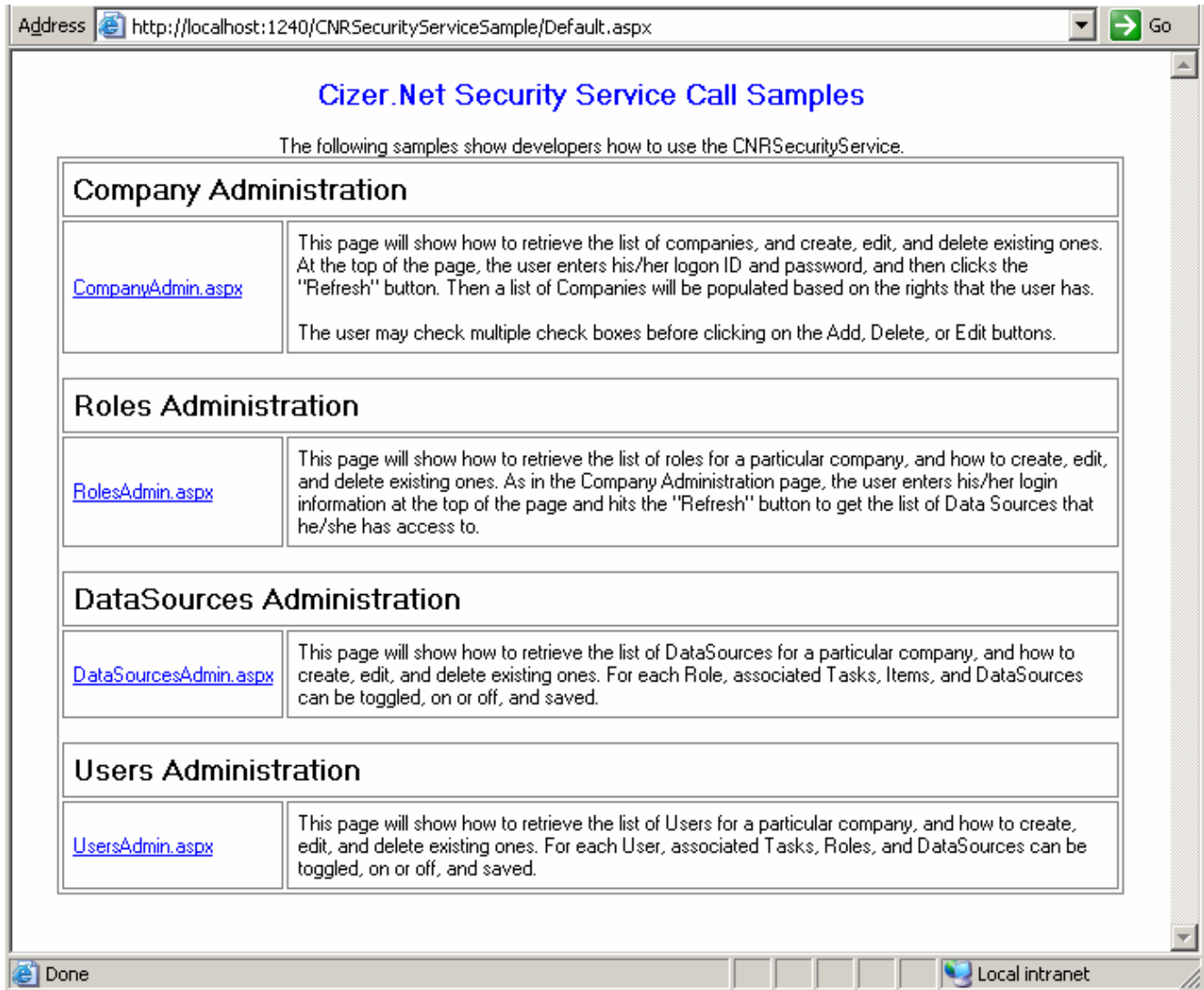
This document will act as a guide through the various functions of the web service.



### **Contents:**

- [The Sample Application of the CNR Security Web Service](#)
- [Initial Setup Code](#)
- [Administration of Companies](#)
- [Administration of DataSources](#)
- [Administration of Roles](#)
- [Administration of Users](#)
- [Conclusion](#)

## CNR Security Web Service – The Sample Application

The main page of this sample web application is the **Default.aspx** page (shown below). It incorporates all of the functionality of this web service condensed into four types of examples. These examples highlight common calls to the service and mirror the Administration pages in the Cizer .Net Reporting tool.



Address  http://localhost:1240/CNRSecurityServiceSample/Default.aspx  Go

### Cizer.Net Security Service Call Samples


The following samples show developers how to use the CNRSecurityService.

Company Administration	
<a href="#">CompanyAdmin.aspx</a>	This page will show how to retrieve the list of companies, and create, edit, and delete existing ones. At the top of the page, the user enters his/her logon ID and password, and then clicks the "Refresh" button. Then a list of Companies will be populated based on the rights that the user has.  The user may check multiple check boxes before clicking on the Add, Delete, or Edit buttons.

Roles Administration	
<a href="#">RolesAdmin.aspx</a>	This page will show how to retrieve the list of roles for a particular company, and how to create, edit, and delete existing ones. As in the Company Administration page, the user enters his/her login information at the top of the page and hits the "Refresh" button to get the list of Data Sources that he/she has access to.

DataSources Administration	
<a href="#">DataSourcesAdmin.aspx</a>	This page will show how to retrieve the list of DataSources for a particular company, and how to create, edit, and delete existing ones. For each Role, associated Tasks, Items, and DataSources can be toggled, on or off, and saved.

Users Administration	
<a href="#">UsersAdmin.aspx</a>	This page will show how to retrieve the list of Users for a particular company, and how to create, edit, and delete existing ones. For each User, associated Tasks, Roles, and DataSources can be toggled, on or off, and saved.

Done  Local intranet

## Initial Setup Code

Before calling any of the functions in the web service, an instance of CNR Security Service must be set up and a user must be authenticated. In order to administer Companies, Roles, DataSources, and Users, the user must be an *Administrator* in the Cizer .Net Reporting tool (usually with the logon ID *admin*). Below is the example C# code:

```
CNRSecurityService objService = new CNRSecurityService();
AuthHeader authentication = new AuthHeader();

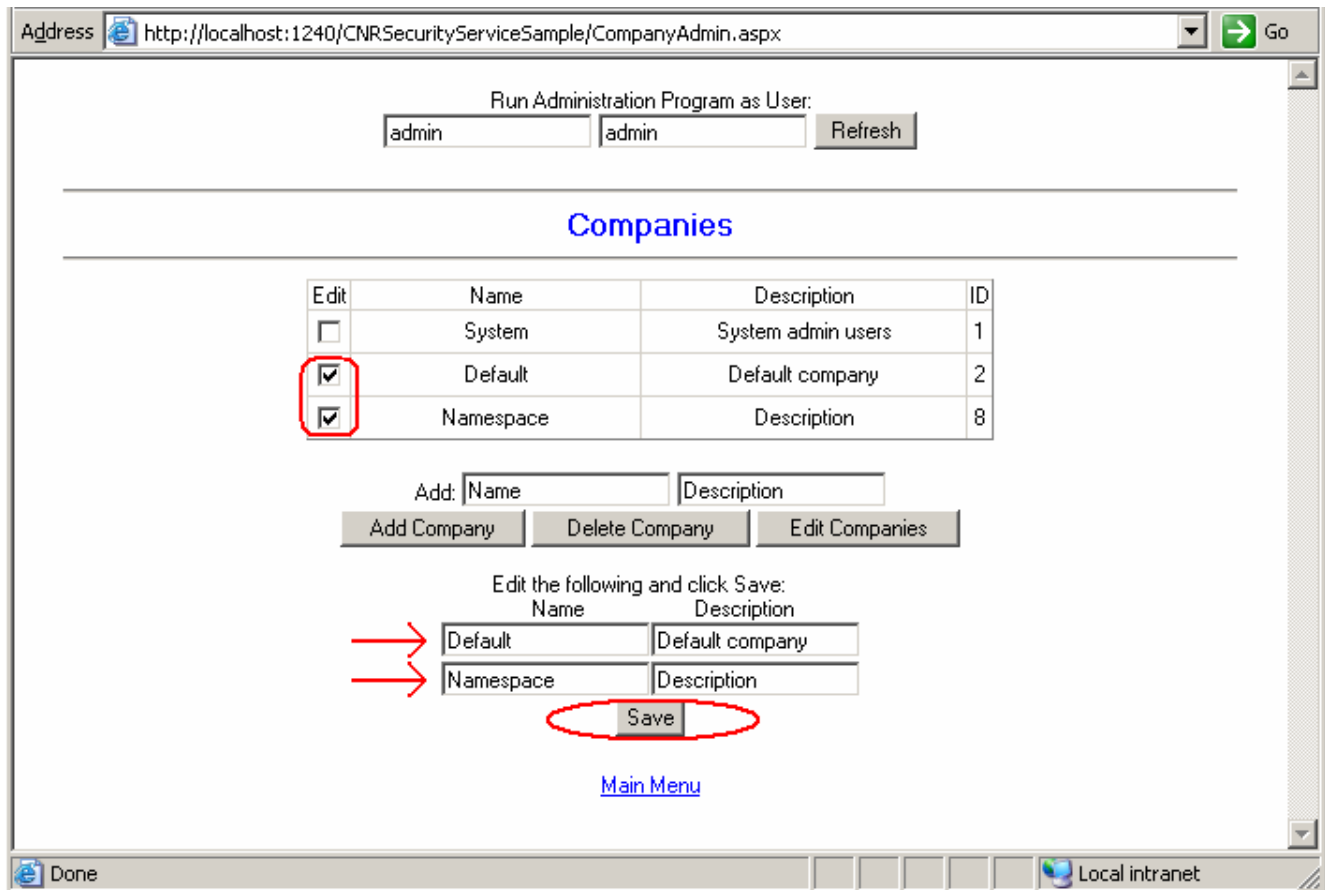
// Authenticate the user and assign the Token to the Authorization Header
authentication.Token = objService.AuthenticateForms(userid, password);

// Store the Authorization Header in the CNRSecurityService to validate the session
objService.AuthHeaderValue = authentication;
```

Now calls to the web service can be performed. Examples of these calls are seen below.

## Administration of Companies in Cizer .Net

As an Administrator, several functions provided by this CNR Security service will help maintain the list of Companies that exist in the Cizer .Net Reporting Tool. In the sample application, on the page **CompanyAdmin.aspx**, the list of all Companies is retrieved and several options allow an Administrator to Add, Edit, and Delete them.



To edit a Company, or several Companies at once, check the appropriate check boxes. Then click the *Edit Companies* button. Below the Edit button, the selected Companies will appear in editable text boxes. Once all changes have been made, click the Save button and the changes will be seen in the List of Companies above.

Let's take a look at the code that allows these functions to work. Remember to authenticate the user first:

```
// STEP 1: Initialize the service and security
CNRSecurityService objService = new CNRSecurityService();
AuthHeader authentication = new AuthHeader();
```

```
// STEP 2: Login as an Administrator with appropriate username and password
authentication.Token = objService.AuthenticateForms( userid.Text, password.Text );
objService.AuthHeaderValue = authentication;
```

Then Add Company:

```
// STEP 3: Add the new company (returns the ID of the new company)
int id = objService.CreateCompany( addCompName.Text, addCompDesc.Text );
```

Or Get a Company:

```
// STEP 3: Call the service with the ID of the Company  
Company comp = objService.GetCompany( CompanyID );
```

Or Get a List of Companies:

```
// STEP 3: Get all Companies ( returns Company[] )  
Company[ ] compList = objService.GetCompanyList();
```

Or Update the Company:

```
// STEP 3: Call the service with the updated company  
objService.UpdateCompany( Company[i] );
```

Or Update the Company's attributes:

```
// STEP 3: Call the service with the new values to assign to the company  
objService.UpdateCompanyMembers( CompanyID, name.Text, description.Text );
```

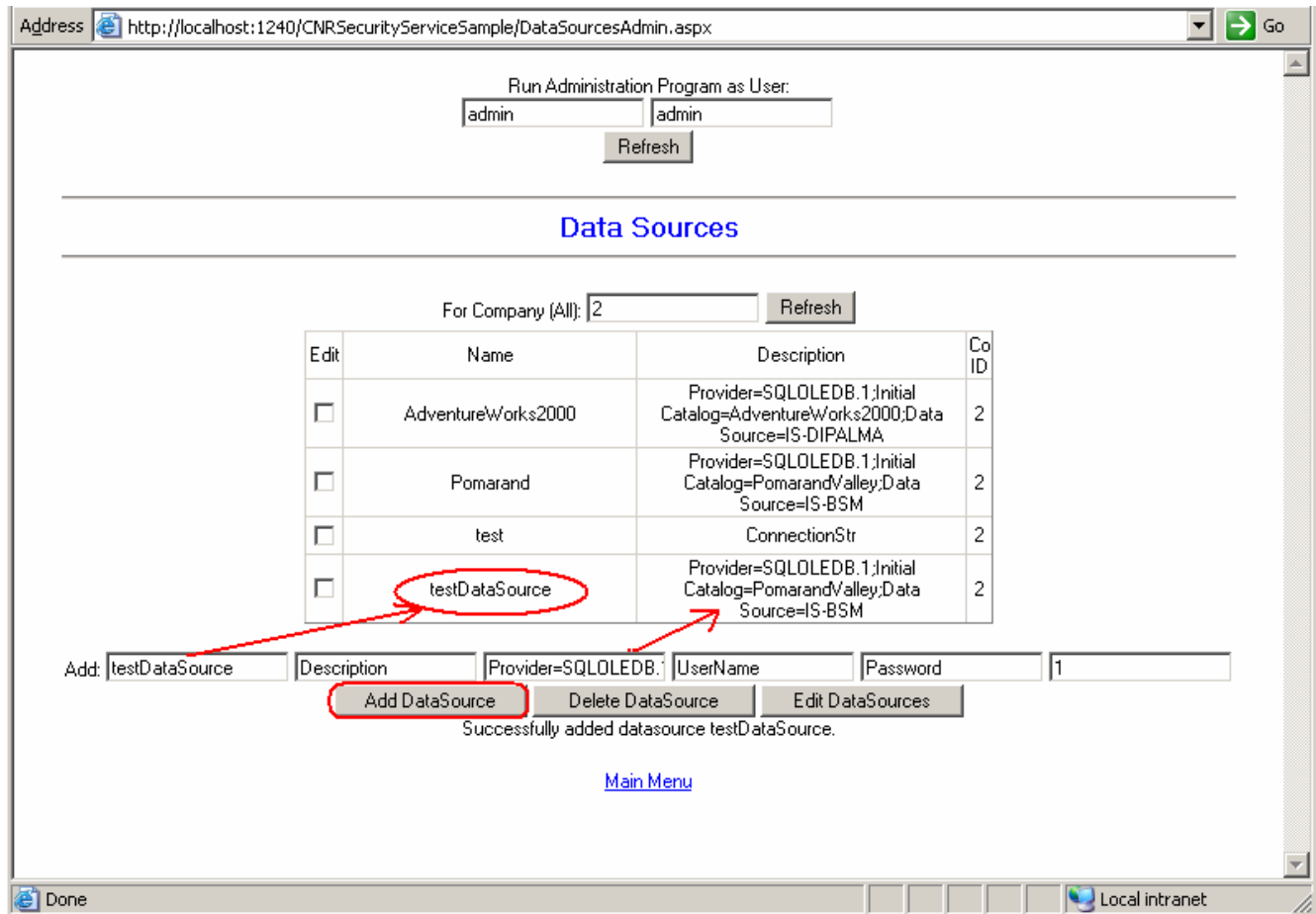
Or Delete the Company:

```
// STEP 3: Call the service with the ID of the company to delete (where ID > 1)  
objService.DeleteCompany( CompanyID );
```

Refer to the code in the sample application webpage, **CompanyAdmin.aspx.cs** to see more detail of how these functions are used.

## Administration of DataSources

Several functions are provided to administer the list of DataSources as well. They can be created and assigned to certain companies, and the list can be retrieved based on the Company ID. In the sample application, on the **DataSources.aspx** page, the list of DataSources for Company with *ID=2* is retrieved, and several options allow you Add, Edit, and Delete them. In the screenshot below, the new DataSource *testDataSource* has been added to the list under Company 2.



Let's take a look at the code that allows these functions to work. Remember to authenticate the user, as seen in the Company administration example.

To Add a DataSource:

```
// STEP 3: Add the new DataSource (returns the ID of the new DataSource)
int id = objService.CreateDataSource(companyID, DSName.Text, DSDesc.Text,
DSConnString.Text, DSUser.Text, DSPassword.Text, (int)spExecFormat,
enumDelimiterType.Bracket, enumDatabaseType.SQLServer,
(enumDSNType)dsnInt, dsnParentID );
```

Or Get a DataSource:

```
// STEP 3: Call the service with the ID of the DataSource  
DataSource ds = objService.GetDataSource( DataSourceID );
```

Or Get a DataSource List:

```
// STEP 3: Get all DataSources for a specified Company  
DataSource[ ] dsList = objService.GetDataSourceList( CompanyID );
```

Or Update the DataSource:

```
// STEP 3: Call the service with the updated DataSource  
objService.UpdateDataSource( DataSource[i] );  
// OR the updated members  
objService.UpdateDataSourceMembers(companyID, DSName.Text, DSDesc.Text,  
DSConnString.Text, DSUser.Text, DSPassword.Text, (int)spExecFormat,  
enumDelimiterType.Bracket, enumDatabaseType.SQLServer,  
(enumDSNTType)dsnInt, dsnParentID );
```

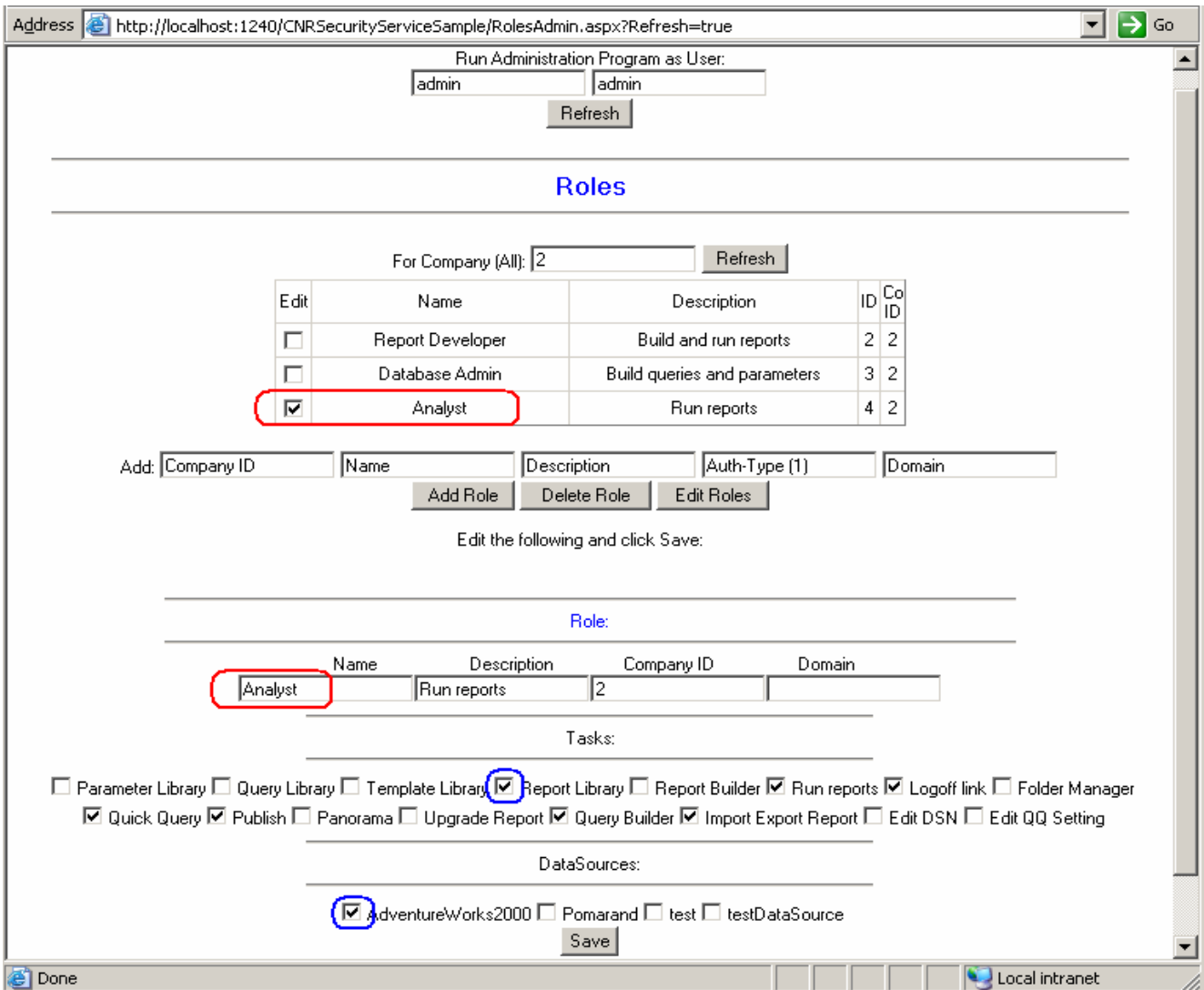
Or Delete the DataSource:

```
// STEP 3: Call the service with the ID of the DataSource to delete  
objService.DeleteDataSource( DataSourceID );
```

Refer to the code in the sample application webpage **DataSourcesAdmin.aspx.cs** to see more detail of how these functions are used.

## Administration of Roles

Roles and Users have several attributes associated with each one. The functions of this service allow the administration of Roles as well as the Role's associated *Tasks*, *DataSources*, and *OtherItems*. In the example below, in the **RolesAdmin.aspx** page, we see the Role *Analyst* is being edited. In addition to the basic attributes of the Role (Name, Description, Company ID, and Domain), there are also Tasks and DataSources that can be edited. In this particular example there are no OtherItems associated with this Role.



The screenshot shows a web browser window with the address `http://localhost:1240/CNRSecurityServiceSample/RolesAdmin.aspx?Refresh=true`. The page title is "Roles". At the top, there is a "Run Administration Program as User:" section with two input fields containing "admin" and a "Refresh" button. Below this is a table of roles for Company ID "2". The "Analyst" role is selected with a checked checkbox. Below the table are fields for "Add:" (Company ID, Name, Description, Auth-Type (1), Domain) and buttons for "Add Role", "Delete Role", and "Edit Roles". The "Edit the following and click Save:" section shows the "Analyst" role details in a table with fields for Name, Description, Company ID, and Domain. Below this is a "Tasks:" section with a list of checkboxes: Parameter Library, Query Library, Template Library, Report Library (checked), Report Builder, Run reports (checked), Logoff link, Folder Manager, Quick Query (checked), Publish, Panorama, Upgrade Report, Query Builder (checked), Import Export Report, Edit DSN, and Edit QQ Setting. The "DataSources:" section has checkboxes for AdventureWorks2000 (checked), Pomarand, test, and testDataSource. A "Save" button is at the bottom.

In order to update the attributes, the checkboxes are checked or unchecked. When the Save button is clicked, the Tasks, DataSources, OtherItems, and basic attributes are updated.

Let's take a look at the code that allows the Roles to be administered. Remember to authenticate the user, as seen in the previous administration examples.



To Add a Role:

```
// STEP 3: Add the new RoleItem (returns the ID of the new Role)  
int id = objService.CreateRole( CompanyID, RoleName.Text, RoleDesc.Text,  
    (int)RoleAuthType, RoleDomain.Text );
```

Or Get a Role:

```
// STEP 3: Call the service with the ID of the RoleItem and the Company  
RoleItem role = objService.GetRole ( RoleID, CompanyID );
```

*(This DOES return all of the Role attributes: Tasks, DataSources, and OtherItems)*

Or Get a List of Roles:

```
// STEP 3: Get all RoleItems for a specified Company  
RoleItem[] Roles = objService.GetRoleList( CompanyID );
```

Or Update the Role:

```
// STEP 3: Call the service with the updated RoleItem  
objService.UpdateRole( Roles[i] );  
// OR the updated members  
objService.UpdateRoleMembers( CompanyID, RoleName.Text, RoleDesc.Text,  
    (int)RoleAuthType, RoleDomain.Text );
```

*(This will only update the basic attributes. The next three functions should be called to update all of the Role attributes: Tasks, DataSources, and OtherItems)*

Or Update the Role's Tasks, DataSources, or OtherItems:

```
// STEP 3: Call the service with the updated RoleItem  
objService.SetRoleTask( Roles[i] );  
objService.SetRoleDataSource( Roles[i] );  
objService.SetRoleItem( Roles[i] );
```

Or Delete the Role:

```
// STEP 3: Call the service with the ID of the RoleItem to delete  
objService.DeleteRole ( RoleID );
```

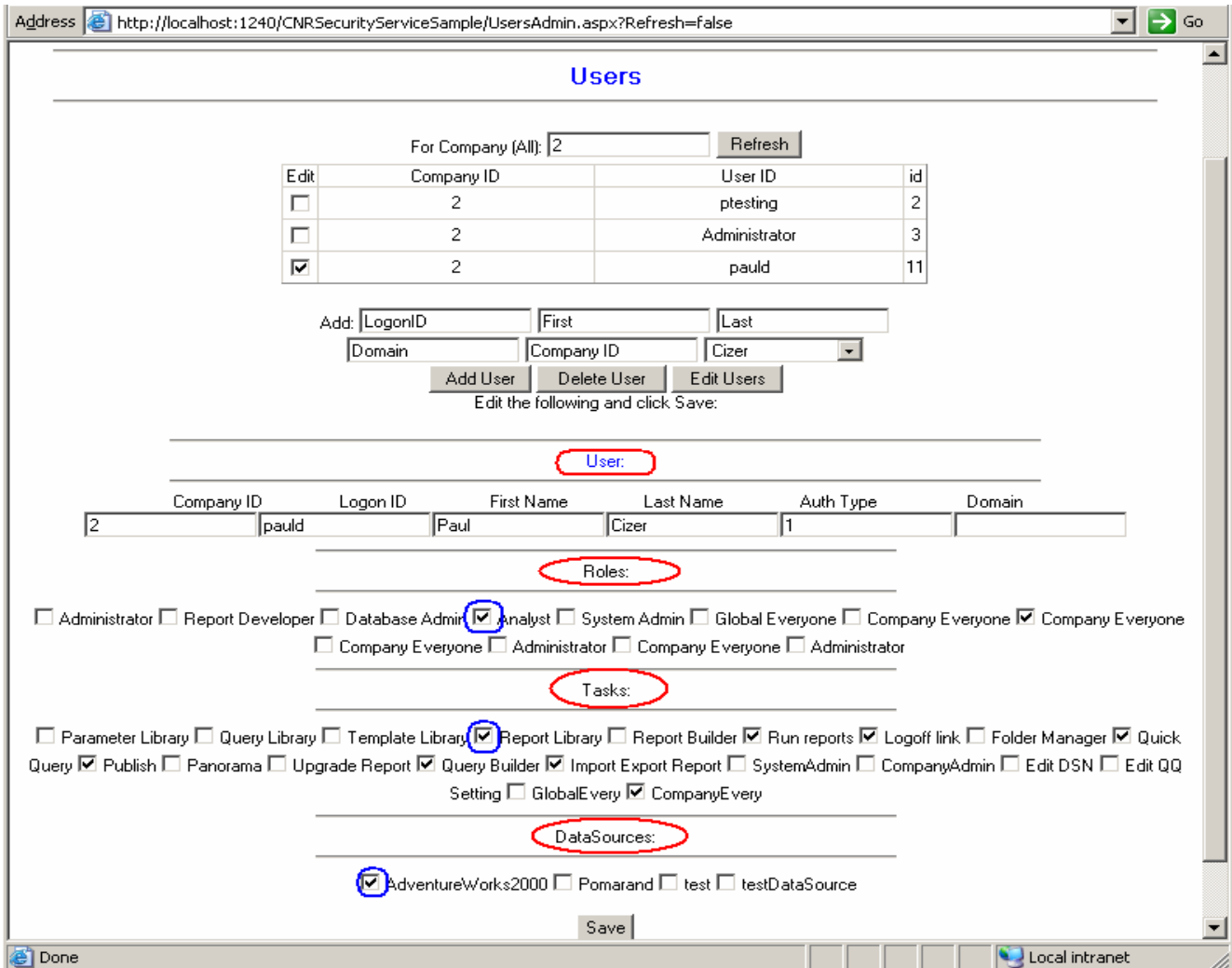
**Example:** Select a particular Task item for a Role, and update the Tasks of this Role.

```
RoleItem updRole = objService.GetRole ( RoleID, CompanyID );  
updRole.Tasks[i].Selected = true;  
objService.SetRoleTask( updRole );
```

Refer to the code in the sample application webpage **RolesAdmin.aspx.cs** to see more detail of how these functions are used.

## Administration of Users

Users have several attributes associated with each, as well. The functions of this service allow the administration of Users as well as the User's associated Tasks, DataSources, and Roles. In the example below, in the **UsersAdmin.aspx** page, we see the User "pauld" is being edited. In addition to the basic attributes of the User (Company ID, Logon ID, First Name, Last Name, Authorization Type, and Domain), there are also *Roles*, *Tasks*, and *DataSources* that can be edited.



The screenshot shows a web browser window with the address `http://localhost:1240/CNRSecurityServiceSample/UsersAdmin.aspx?Refresh=false`. The page title is "Users".

At the top, there is a filter: "For Company (All):" with a dropdown set to "2" and a "Refresh" button.

Below this is a table of users:

Edit	Company ID	User ID	id
<input type="checkbox"/>	2	ptesting	2
<input type="checkbox"/>	2	Administrator	3
<input checked="" type="checkbox"/>	2	pauld	11

Below the table are input fields for adding a new user: "Add: LogonID", "First", "Last", "Domain", "Company ID", and a "Cizer" dropdown menu. There are "Add User", "Delete User", and "Edit Users" buttons. A note says "Edit the following and click Save:".

The "User:" section contains a form for editing the selected user "pauld":

Company ID	Logon ID	First Name	Last Name	Auth Type	Domain
2	pauld	Paul	Cizer	1	

The "Roles:" section has several checkboxes, with "Analyst" checked:

Administrator  Report Developer  Database Admin  Analyst  System Admin  Global Everyone  Company Everyone  Company Everyone  
 Company Everyone  Administrator  Company Everyone  Administrator

The "Tasks:" section has many checkboxes, with "Report Library" and "CompanyEvery" checked:

Parameter Library  Query Library  Template Library  Report Library  Report Builder  Run reports  Logoff link  Folder Manager  Quick Query  Publish  Panorama  Upgrade Report  Query Builder  Import Export Report  SystemAdmin  CompanyAdmin  Edit DSN  Edit QQ Setting  GlobalEvery  CompanyEvery

The "DataSources:" section has "AdventureWorks2000" checked:

AdventureWorks2000  Pomarand  test  testDataSource

A "Save" button is at the bottom of the form.

In order to update the attributes, the checkboxes are checked or unchecked. When the Save button is clicked, the Tasks, DataSources, OtherItems, and basic attributes are updated.

Let's take a look at the code that allows the Users and its members to be administered. Remember to authenticate the user, as seen in the previous administration examples.

To Add a User:

```
// STEP 3: Add the new UserItem (returns the ID of the new User)  
int id = objService.CreateUser( CompanyID, LogonID.Text, UserFirst.Text, UserLast.Text,  
    (int)UserAuthType, UserDomain.Text );
```

Or Get a User:

```
// STEP 3: Call the service with the ID of the UserItem  
UserItem user = objService.GetUser ( UserID );
```

*(This DOES return all of the User attributes: Tasks, DataSources, and Roles)*

Or Get a List of Users:

```
// STEP 3: Get all UserItems for a specified Company  
UserItem[] Users = objService.GetUserList( CompanyID );
```

Or Update the User:

```
// STEP 3: Call the service with the updated UserItem  
objService.UpdateUser( Users [i] );  
// OR the updated members  
objService.UpdateUserMembers( CompanyID, LogonID.Text, UserFirst.Text,  
    UserLast.Text, (int)UserAuthType, UserDomain.Text );
```

*(This will only update the basic attributes. The next three functions should be called to update all of the User attributes: Tasks, DataSources, and Roles)*

Update the User's Tasks, or DataSources, or Roles:

```
// STEP 3: Call the service with the updated UserItem  
objService.SetUserTask( Users [i] );  
objService.SetUserDataSource( Users [i] );  
objService.SetUserRole( Users [i] );
```

Or Delete the User:

```
// STEP 3: Call the service with the ID of the UserItem to delete  
objService.DeleteUser ( UserID );
```

**Example:** Select a DataSource item for a User to use, and update the DataSource attributes of this User.

```
UserItem updUser = objService.GetUser ( UserID );  
updUser.DataSources[i].Selected = true;  
objService.SetUserDataSource ( updUser );
```

Refer to the code in the sample application webpage **UsersAdmin.aspx.cs** to see more detail of how these functions are used.



## **\*\*NEW\*\* Cizer.Net Reporting 4.0**

### **CNR Security Web Service & Sample Application**

---

#### **Conclusion**

In customizing your solutions, Cizer .Net reporting Administration can now be seamlessly embedded in any of your applications. The Sample application will help you understand the Cizer .Net Security Web Service to provide custom Administration of all of your Companies, DataSources, Roles, and Users in your Cizer .Net Reporting tool.