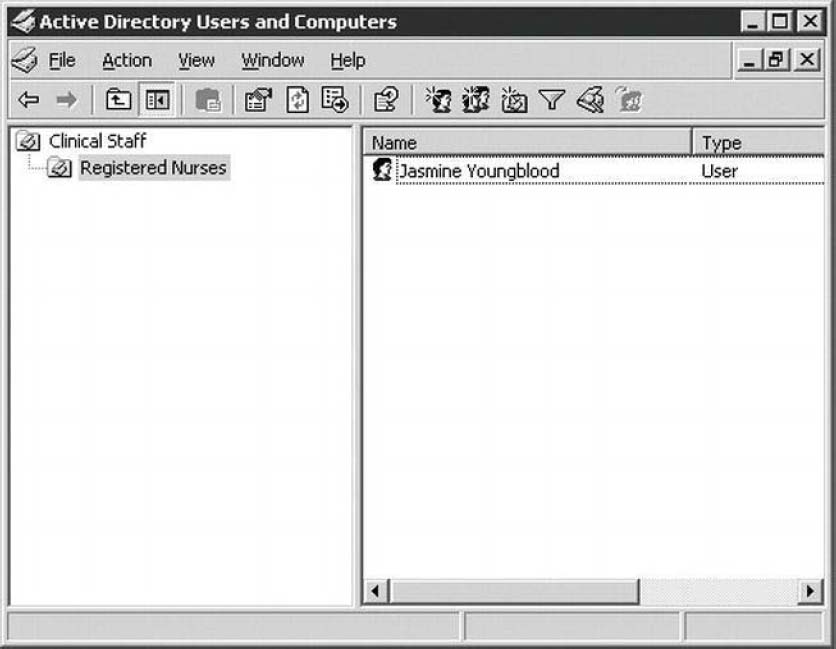
**Introducing SSRS Roles**

By default, the installed SSRS Web service uses Windows integrated authentication to access reports and report content. Windows user or group security accounts stored in Active Directory must be associated with an SSRS role before they will have access to the SSRS server. Administrators can assign the Windows accounts to SSRS roles with Report Manager. In the test scenario for our health-care application, we have set up a test Windows account, named jyoungblood; you will assume jyoungblood is a registered nurse in a health-care organization who makes home visits to patients.

All the clinical staff, including nurses such as jyoungblood, are associated with security groups within Active Directory for the domain. So, you will make jyoungblood a member of the RNsecurity group. In addition to the security group RN, all registered nurses, including jyoungblood, will be contained with an organizational unit (OU) inside Active Directory, as shown in the Active Directory Users and Computers window in [Figure 11-10](https://cdn2.percipio.com/1619501603.1168ce432708e24d1fee24e0c8ce3c1ea8258f87/eod/books/49277/OEBPS/section-102.xhtml#ch11_fig10). Although you will not use OUs when assigning a user or group to a role in SSRS, it is important to note that you can use OUs to configure Group Policy settings that apply to security as well, such as locking down the user's desktop or Internet Explorer.

Figure 11-10: Test Windows account in Active Directory



Before assigning the test Windows user to an SSRS role and testing the permissions settings, first look at these five predefined roles:

* *Browser:* Users assigned to the Browser role may only view reports, folders, and resources. They may also manage their own subscriptions.
* *Content Manager:* Administrators are assigned to the Content Manager role by default. Users assigned to this role can perform every task available for SSRS objects, such as folders, reports, and data sources that they manage.
* *My Reports:* This is the default role automatically assigned to a user when the My Reports feature is enabled on the SSRS server, discussed later in this section.
* *Publisher:* Users assigned to this role, by default, have enough privileges to publish reports and data sources to the report server. Typically, this role is used for report authors who work with Report Builder or Visual Studio to create and deploy reports.
* *Report Builder:* The Report Builder role is used primarily for assigning the required permissions to users who will use the Report Builder application launched from Report Manager, which is covered in [Chapter 09](https://cdn2.percipio.com/1619501603.1168ce432708e24d1fee24e0c8ce3c1ea8258f87/eod/books/49277/OEBPS/chapter-9-81.xhtml#ch009).

SSRS roles are defined by the tasks that users assigned to each role may perform. SSRS tasks provide content management permissions and define which SSRS objects are viewable by the user. Users can perform the following tasks:

* Consume reports
* Create linked reports
* Manage all subscriptions
* Manage data sources
* Manage folders
* Manage individual subscriptions
* Manage models
* Manage report history
* Manage reports
* Manage resources
* Set security for individual items
* View data sources
* View folders
* View models
* View reports
* View resources

Each predefined role is configured by default, with a specific combination of allowable tasks. Users assigned to the Publisher role, for example, may manage folders, reports, resources, models, and data sources as well as create linked reports.

SSRS 2012 does not allow administrators to edit the tasks available to roles or to add new roles through the Web-based Report Manager. That work is relegated to the SSMS 2012 interface. Open SSMS and connect to the Reporting Services instance by using the Connect button and selecting Reporting Services. The dialog box will look similar to the normal connection screen, but the server type will be Reporting Services. Enter the name of the SSRS server and click Connect to view the available options. [Figure 11-11](https://cdn2.percipio.com/1619501603.1168ce432708e24d1fee24e0c8ce3c1ea8258f87/eod/books/49277/OEBPS/section-102.xhtml#ch11_fig11) shows the connection screen.

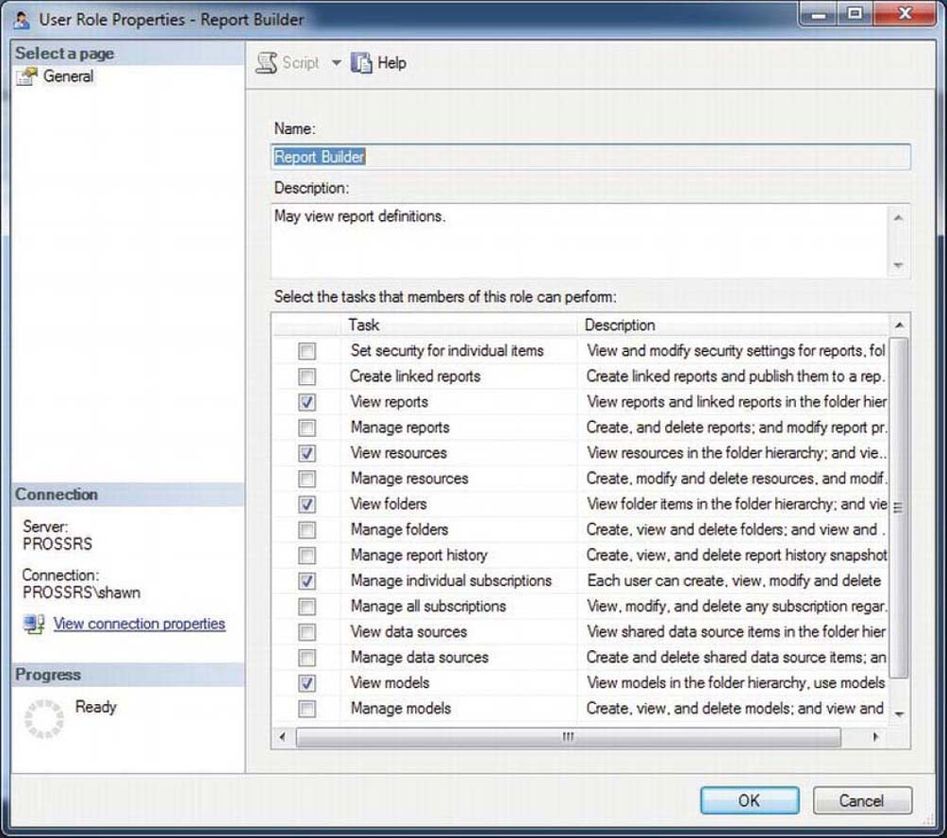
Once connected, you will find three sections listed for the instance. The section we will focus on here is the second, Security. Expand the Security node to reveal two subnodes, Roles and System Roles. The Roles section is used to manage user access on a reporting level. This section handles regular users who need to view, browse, and create reports. The second section, System Roles, is used to administer privileges for overall system tasks. We will focus on the Roles section here.

Figure 11-11: Connecting to SSRS via SSMS



Expand the Roles node to see the five default roles that we discussed earlier. To see the properties of the role, right-click the browser and select Properties. You will see a description for the role, along with each of the tasks that the role can have. [Figure 11-12](https://cdn2.percipio.com/1619501603.1168ce432708e24d1fee24e0c8ce3c1ea8258f87/eod/books/49277/OEBPS/section-102.xhtml#ch11_fig12) shows you how this will look. From this window, you can edit the description or edit what tasks the role can perform. You do not want to change the Report Builder role tasks now, so have a look around to get a feel for what options you have with security changes, but click Cancel to discard any changes you might have made.

Figure 11-12: Viewing the Report Builder role task permissions

Larger View

From SSRS, you can also add new roles or edit any existing roles for the server. This is important if you need to create a new custom role for users who may need permission that is slightly elevated from a standard SSRS role. This interface is identical to the role-editing window and can be reached by right-clicking the Roles node and selecting New Role.

**Testing SSRS Role Assignments**

In this section, we will go through the process of adding folders and report objects that would be in line with what nurse jyoungblood would use. You will want to ensure that she will not have the ability to navigate to other folders and run other reports that may contain confidential information.

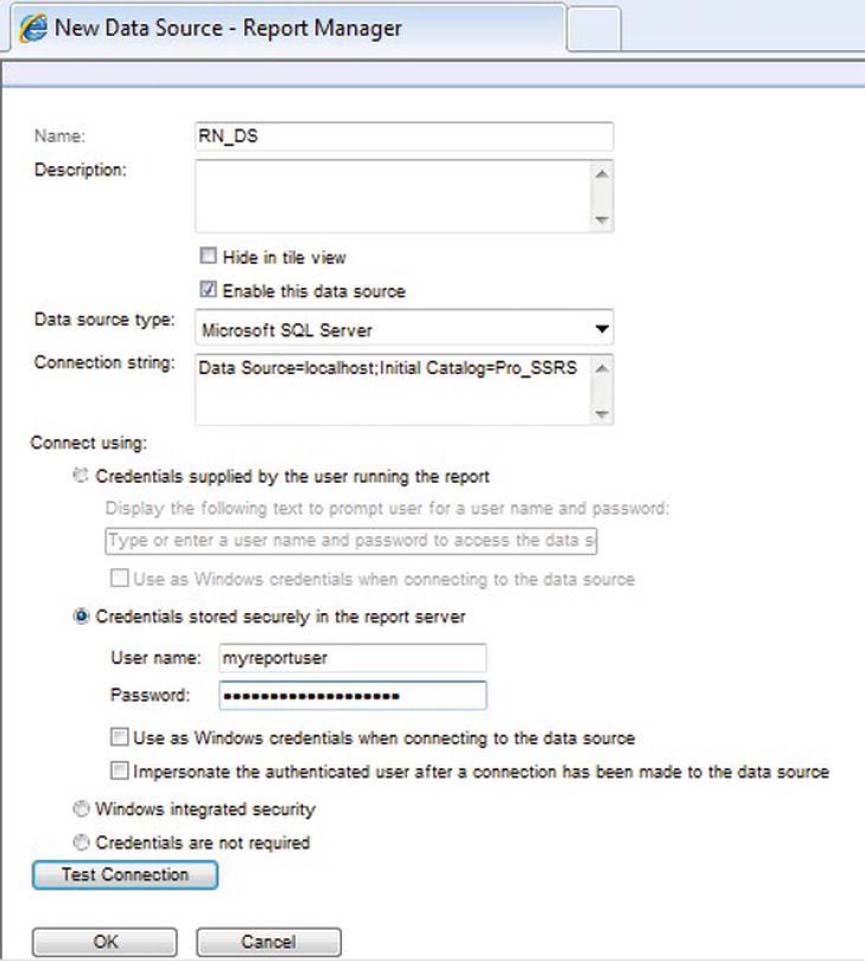
The first step in testing jyoungblood's access to the reports that have been defined for her security group, RN, is to publish the reports to a folder on the SSRS server that will contain reports for registered nurses. As the administrator for the test, open Report Manager, and create two new folders: one in the root folder called Clinical Reports and then one inside the Clinical Reports folder called Registered Nurse. To do this, simply click New Folder in Report Manager. Because both of these folders, by default, are inheriting permissions from the parent folder, which currently is configured for administrator access only, you will alter the permissions manually so that the new folder (and the reports and data source you will add to it) will maintain its own security settings.

To publish reports to the Registered Nurse folder, you could use any method already covered in the previous deployment Chapter, but for this test, simply upload a report you have already worked with: Daily Schedule. Then create a data source called RN\_DS for the purposes of testing security. Upload the report file Daily Schedule.rdl from Report Manager in the Pro\_SSRS project. In Report Manager, in the Registered Nurse folder, simply click Upload File, browse to Daily Activity.rdl, and click OK. To create the data source manually with Report Manager, simply click New Data Source in the Registered Nurse folder, name the data source RN\_DS, and set the connection string to the following (replacing localhost, if necessary, with your SQL Server server name where you have created the Pro\_SSRS database):

Data Source=localhost;Initial Catalog=Pro\_SSRS

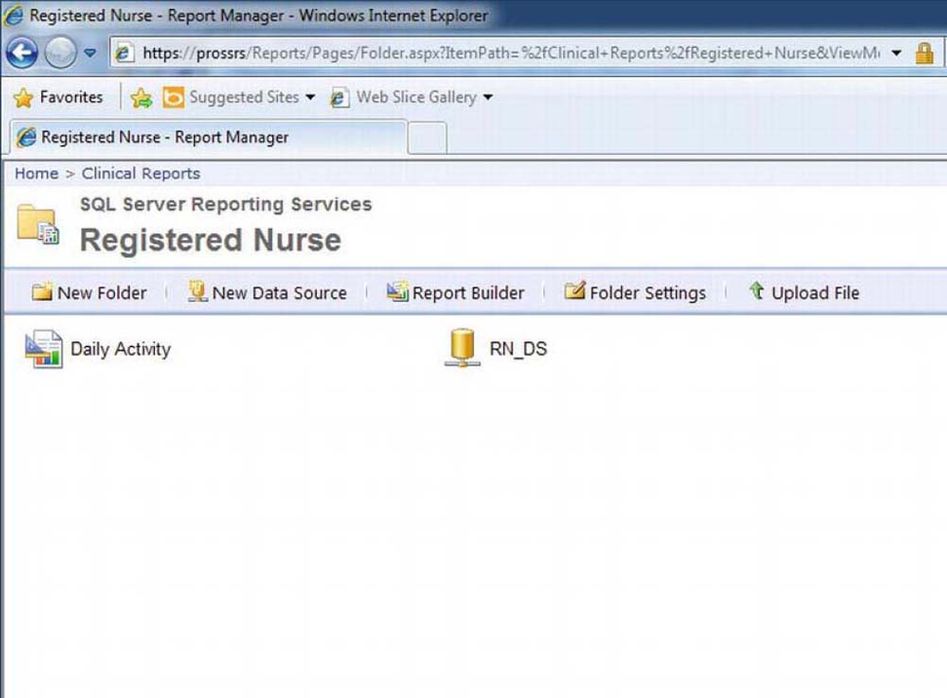
In this case, you will choose to set the data source authentication method to Credentials Stored Securely in the Report Server and supply a name and password that will be used to access the data in the Pro\_SSRS database. Assuming that jyoungblood's Windows account was granted access to the data source database, you could have selected the Windows Integrated Security option to pass through the Windows account to the SQL Server database. You know that you will configure the report to filter out data that are relevant only to the clinician jyoungblood, so you don't need to be overly concerned with the stored credentials. If you don't already have a SQL Server login with permissions to the Pro\_SSRS database, go ahead and create one called myreportuser and give it db\_datareader permissions to the database. You can see the completed screen for setting up this new data source in [Figure 11-13](https://cdn2.percipio.com/1619501603.1168ce432708e24d1fee24e0c8ce3c1ea8258f87/eod/books/49277/OEBPS/section-102.xhtml#ch11_fig13).

Figure 11-13: Setting up the RN\_DS data source



[Figure 11-14](https://cdn2.percipio.com/1619501603.1168ce432708e24d1fee24e0c8ce3c1ea8258f87/eod/books/49277/OEBPS/section-102.xhtml#ch11_fig14) shows the folder structure and setup of the report objects for the initial test you will perform. At this point, you have not granted SSRS role assignments to the Windows account, jyoungblood, or the security group, RN, of which she is a member. The Daily Activity report provides clinicians with a list of their daily activities. In a data-driven subscription, where the report can be mailed to the clinicians after processing, the parameter for the employee's ID was used to create reports with data unique to each individual. In this test now, however, you want to allow access to the same report to be run manually from Report Manager. This poses its own set of concerns, which we will cover as we step you through the process.

Figure 11-14: Report objects for registered nurse test

Larger View

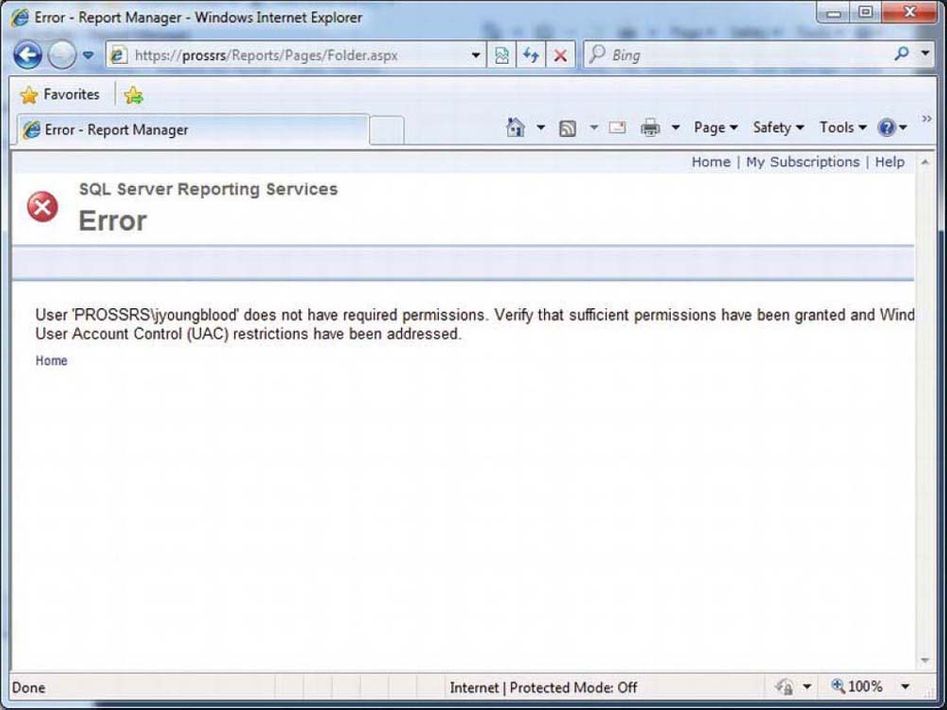
To begin the test, open a browser and log in with the user jyoungblood. Once the browser is open, paste the link in the address bar to the Registered Nurse folder you created previously, as shown here:

http://YourServerName/Reports/Pages/Folder.aspx?ItemPath=%2fClinical+Reports%2https://cdn2.percipio.com/1619501603.1168ce432708e24d1fee24e0c8ce3c1ea8258f87/eod/books/49277/images/ch011_p334_infig001.jpg

fRegistered+Nurse&ViewMode=Detail

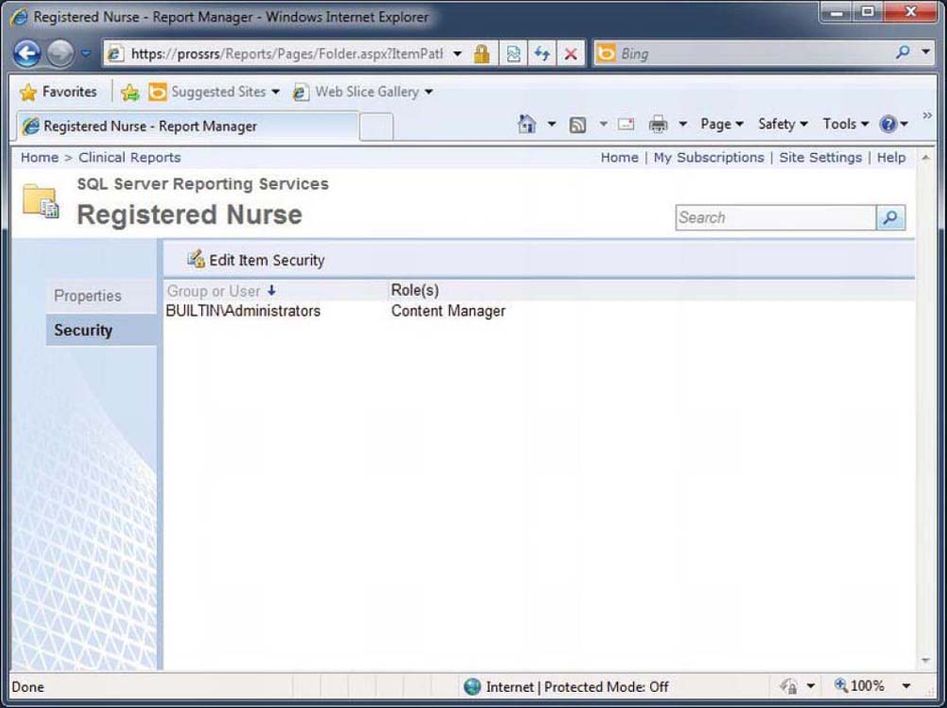
You have to paste the link into the browser because the permissions for jyoungblood currently do not allow navigation to the report directly through Report Manager. As you can see in [Figure 11-15](https://cdn2.percipio.com/1619501603.1168ce432708e24d1fee24e0c8ce3c1ea8258f87/eod/books/49277/OEBPS/section-102.xhtml#ch11_fig15), when you view this link, you receive an error message indicating that the user does not have permissions to view the resources in the folder.

Figure 11-15: Error message for insufficient permissions

Larger View

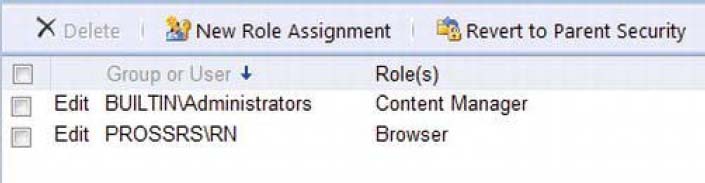
Running Report Manager as an administrator again, you are now going to set the permissions for jyoungblood. You can control security settings on the Property tab for each folder as well as on individual report items. In this case, you will set permissions at the folder level for the Registered Nurse folder. Navigate to the Registered Nurse folder while using a browser running from an SSRS Administrator account, click the Folder Settings button in the folder menu bar, and then select the Security option on the left. As you can see in [Figure 11-16](https://cdn2.percipio.com/1619501603.1168ce432708e24d1fee24e0c8ce3c1ea8258f87/eod/books/49277/OEBPS/section-102.xhtml#ch11_fig16), the default security group is BUILTIN\administrators, which is assigned to the Content Manager role.

Figure 11-16: Registered Nurse folder's default permissions

Larger View

To set permissions for the RN group, click Edit Item Security to break the security inheritance from the parent folder. Go ahead and click OK on the warning screen that you are breaking the security inheritance. Now you can click New Role Assignment and add the RN security group as the group name on the New Role Assignment form. For the role assignment, choose Browser, which will allow the users assigned to the RN group to view the Registered Nurse folder and all its child nodes, to view reports and resources, and to configure their own subscriptions. You can see this in the security list now in [Figure 11-17](https://cdn2.percipio.com/1619501603.1168ce432708e24d1fee24e0c8ce3c1ea8258f87/eod/books/49277/OEBPS/section-102.xhtml#ch11_fig17).

Figure 11-17: New security list for the Register Nurse folder

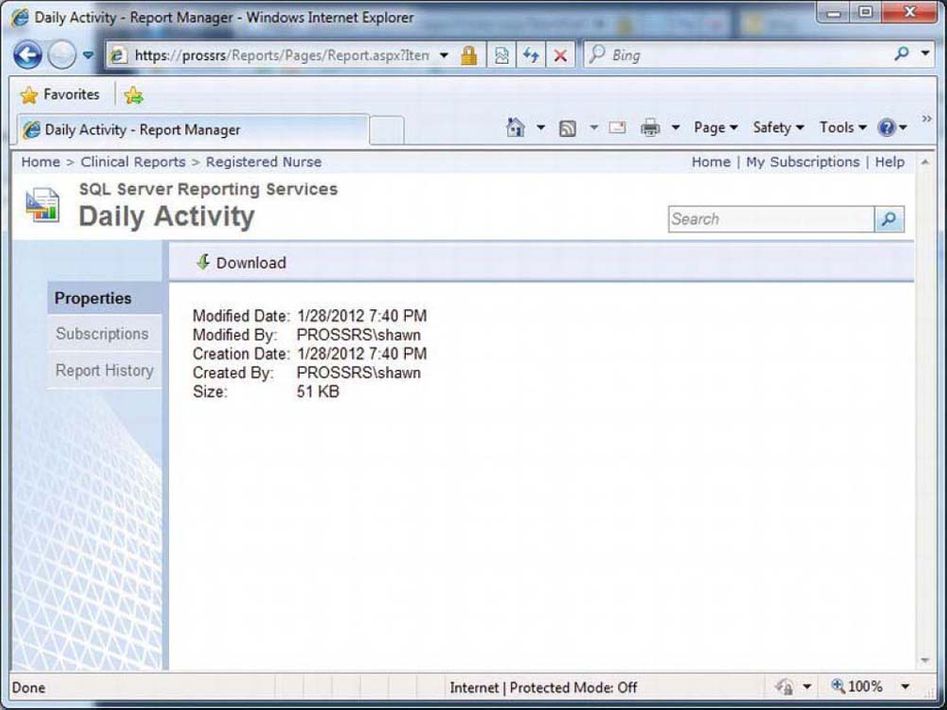


Now when you access the Registered Nurse folder logged in as jyoungblood, all you see are the reports that have been deployed to that folder, not the data source. In addition, all the properties for the objects you can view have limited accessibility and content. If you click the menu for that report item and select Manage, for example, you will see only the general properties information, such as the Creation Date and Modified Date of the Daily Schedule report. In contrast, an administrator viewing the same properties page would be able to see and modify other report property settings such as Parameters, Data Sources, Execution, History, and Security, as shown in [Figure 11-18](https://cdn2.percipio.com/1619501603.1168ce432708e24d1fee24e0c8ce3c1ea8258f87/eod/books/49277/OEBPS/section-102.xhtml#ch11_fig18).

There are two SSRS roles that you should be familiar with. We have just mentioned one, System Administrator, and the other available SSRS role is System User. These are site-wide roles that allow users to perform certain tasks within the SSRS installation. These are not different from the folder/item level security we have discussed previously, in that these roles grant privileges to the entire SSRS system. You can assign these roles to users under the Site Settings section of Report Manager. The following is a list of what each role allows the user/group to do when assigned the role:

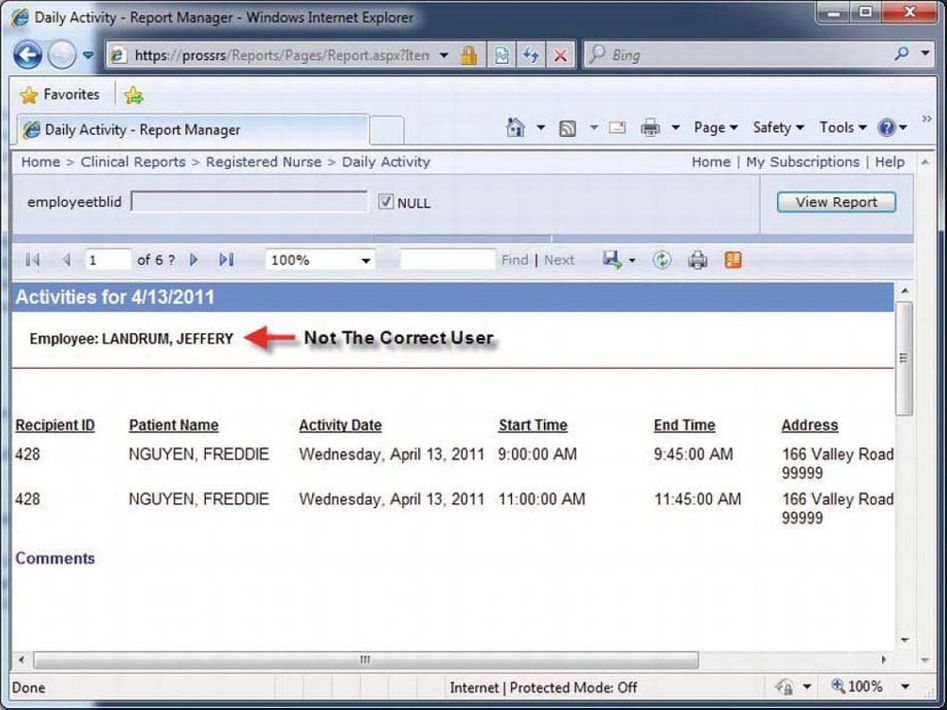
* System User
  + View system properties
  + View shared schedules
  + Allow use of Report Builder or other clients that execute RDL
* System Administrator
  + View/Modify system role assignments
  + View/Modify system role definitions
  + View/Modify system properties
  + View/Modify shared schedules
  + All other access granted to System User role

Figure 11-18: Report properties available for an administrator

Larger View

To complete the test, you will simply execute the Daily Schedule report as jyoungblood. You have granted permission for the RN Windows security group to inherit the SSRS Browser role, so you should not have a problem executing the report. The report executes successfully. However, [Figure 11-19](https://cdn2.percipio.com/1619501603.1168ce432708e24d1fee24e0c8ce3c1ea8258f87/eod/books/49277/OEBPS/section-102.xhtml#ch11_fig19) shows one glaring issue—even though jyoungblood has executed this report, she is seeing other employees' scheduled visits. Although she would be able to enter an EmployeeID parameter value that would limit the data on the report to only her data, she would still be able to see other employees' schedules by entering their IDs, assuming she knew what they were. Although this might be an acceptable practice for many companies, in the next section we will show how to go a step further to ensure that she will be able to view her schedule only.

Figure 11-19: Test account viewing other employee's data

Larger View

**Tip**

When building a testing environment, a number of available resources can simplify the process. Virtualization is common in most every environment today. Microsoft's Hyper-V or other desktop virtualization tools are advantageous tools for testing, as they allow you to run multiple operating systems simultaneously on a single machine.

**Filtering Report Content with User!UserID**

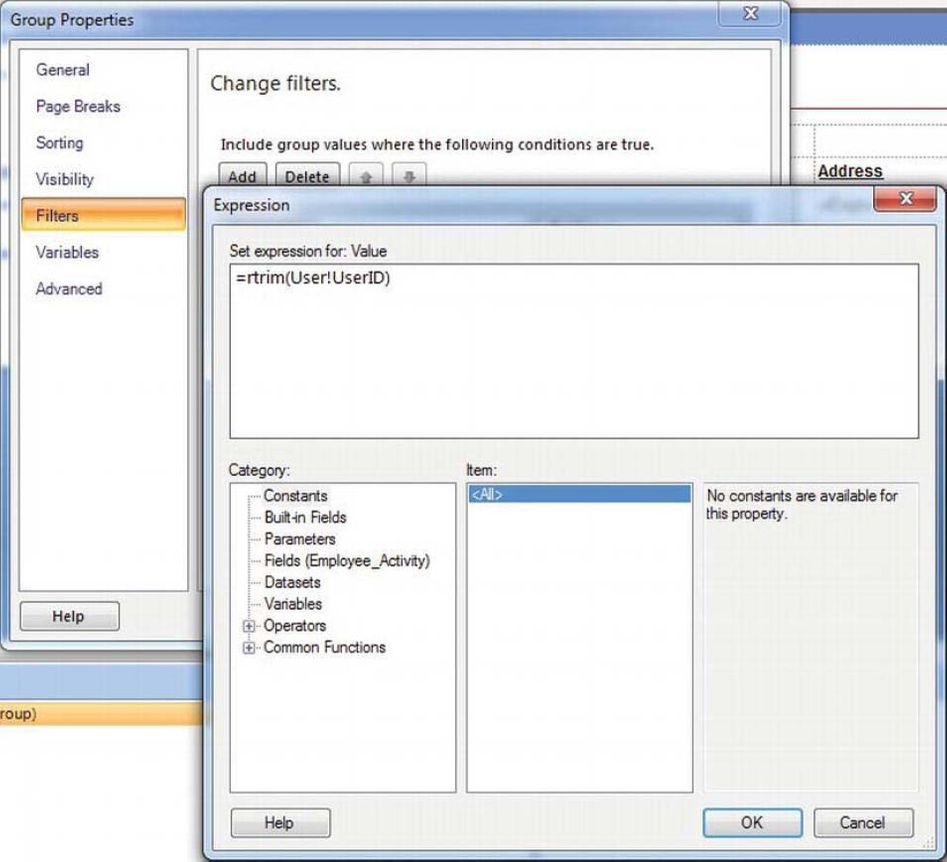
For the Daily Schedule report, say you have decided that you want the users to be able to view only their own schedules. SSRS allows you to accomplish this by creating a report filter that uses the value of the login account for the user executing the report. The login name value is returned from a global collection in SSRS. You have been using global collections all along— for example, when you use an expression such as =Fields!FieldName.Value, you are actually returning a value from the Fields global collection.

The global collection that you will use for the report filter is User, and the value you are interested in is UserID. The expression will therefore be =User!UserID.

To use User!UserID in the filter, you will need a field in the dataset that will equal the UserID value. In the dataset for the Daily Schedule report, you may recall that you have a field called HWUserLogin that you can use for this purpose. When compared by the filter, the two values will be identical—one value delivered with the dataset and the other at execution time of the report. After the filter is applied, the report will display only those records where the username of the employee executing the report matches the value of the HWUserLogin field returned with each record of the dataset.

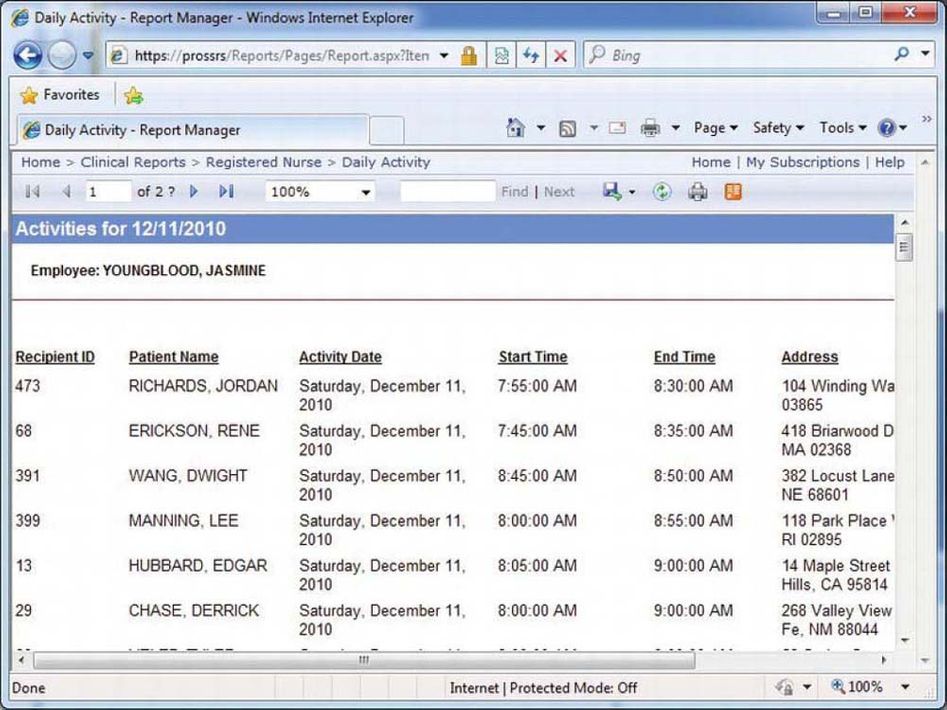
Unlike parameters, filters cannot be set through Report Manager. To set up a filter, you will need to modify the report itself, either in the RDL file directly, through Report Builder or in Visual Studio, as shown in [Figure 11-20](https://cdn2.percipio.com/1619501603.1168ce432708e24d1fee24e0c8ce3c1ea8258f87/eod/books/49277/OEBPS/section-102.xhtml#ch11_fig20). Notice that you can use the RtRIM function to strip off the trailing spaces; otherwise, the comparison may fail. Make sure that the employee Login matches the user that you have created so that the comparison will match. In the book example, this is PROSSRS\Jyoungblood, but it may differ for your examples. Add the filter to the group properties to filter out any other users except the viewing user.

Figure 11-20: UserID filter for daily schedule report

Larger View

Because this report may return several hundred records, even though it will filter automatically for each user, it is a good idea to cache the report for ten minutes. Caching will help alleviate the performance hit of requerying the data source every time a new user accesses the report. When the user jyoungblood executes the report again, you can see that the schedule now reflects only her schedule, as shown in [Figure 11-21](https://cdn2.percipio.com/1619501603.1168ce432708e24d1fee24e0c8ce3c1ea8258f87/eod/books/49277/OEBPS/section-102.xhtml#ch11_fig21).

Figure 11-21: Filtered report execution for a test user

Larger View

**Setting Data Source Security**

Once you have deployed the data source to the report server, you can specify its connection properties. This is an important step because the property settings determine how both the user and SSRS will connect to the data source. When executing unattended reports—for example, for a user subscription—SSRS will control passing authentication credentials to the data source and must have access to valid authentication credentials.

Four connection options for the data source are available in Report Manager:

* *The Credentials Supplied by the User Running the Report:* With this option, users are always prompted to log in to the data source when executing the report.
* *Credentials Stored Securely in the Report Server:* SSRS uses authentication credentials stored in the ReportServer database. The sensitive login information is encrypted.
* *Windows Integrated Security:* This option passes the login information for the current user to the data source. Don't choose this option if the data source will be used for unattended installs or if Kerberos is not configured for the Windows domain. You will also get issues when using this where multiple hops are made from SQL Servers. Unless Kerberos is installed and configured correctly across your entire domain, this may not work correctly in all situations
* *Credentials Are Not Required:* This is the least secure option and is used when the data source does not require authentication.

**Setting SQL Server Permissions**

In [Chapter 3](https://cdn2.percipio.com/1619501603.1168ce432708e24d1fee24e0c8ce3c1ea8258f87/eod/books/49277/OEBPS/chapter-3-20.xhtml#ch003), when you created the stored procedure called Emp\_Svc\_Cost, you set the permissions to allow public execution while designing the report. The environment you were working in was otherwise secure, as it was isolated from other networks and there was no fear of it being compromised.

Now that you are deploying the stored procedure in a production environment, you will need to lock down the stored procedure as well. You can do this through SSMS by right-clicking the stored procedure and selecting Properties (see [Figure 11-22](https://cdn2.percipio.com/1619501603.1168ce432708e24d1fee24e0c8ce3c1ea8258f87/eod/books/49277/OEBPS/section-102.xhtml#ch11_fig22)). Next, click the Permissions page. Search for the RN group using the Search… button to load its explicit permissions. Check the Execute permission under the Grant column and this will allow any user from the RN group to execute this stored procedure. You do not need to explicitly grant Execute rights to the test user jyoungblood, as she is a member of the RN security group.

Figure 11-22: Setting SQL Server stored procedure permissions

Larger View