



Reports 05.07.01 User Manual

Contents

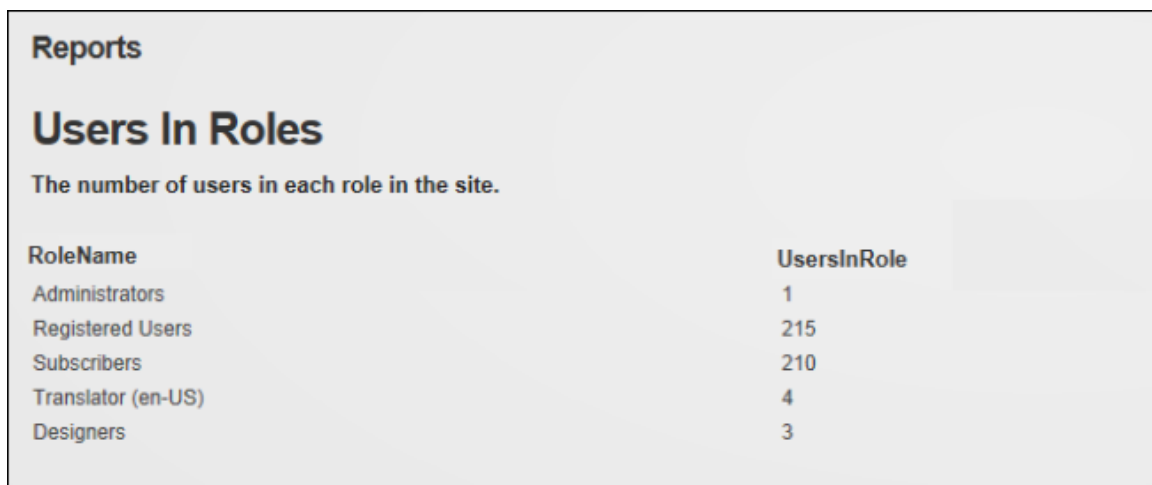
Reports	2
Settings	2
Configuring Display and Caching Settings	2
Configuring Grid Visualizer Settings	3
Configuring the HTML Template Visualizer Settings	5
Configuring the XSL Transformation Visualizer Settings.....	6
Super Users.....	7
Overview of Data Sources	7
Setting the Generic ADO.Net Data Source.....	9
Setting the DNN Data Source	10
Setting the Microsoft SQL Server Data Source	11
Setting Converters	12
Sample Report: Counting Users In Each Role	12
Styling the Grid Visualizer	13
Troubleshooting: Upgrading.....	14

Reports

The Reports module displays the results of an SQL query to the site's database. The resulting report is displayed as a series of records. Token replacement is enabled for calculated columns which are normally read only fields. Page Editors can modify the way reports are displayed however only Super Users can create reports. Tip: Use the XML module to view data from hierarchical sources such as XML.

Reports must be deployed and installed by a Super User. Install and source packages can be downloaded from the [releases page](#) and feature requests, bug links and general questions can be submitted via the [issues page](#).

Module Version: 05.07.01 / **Minimum DNN Version:** 07.01.02



The screenshot shows a report titled "Users In Roles" with the subtitle "The number of users in each role in the site." Below the subtitle is a table with two columns: "RoleName" and "UsersInRole". The table lists five roles and their corresponding user counts.

RoleName	UsersInRole
Administrators	1
Registered Users	215
Subscribers	210
Translator (en-US)	4
Designers	3

Reports

Settings

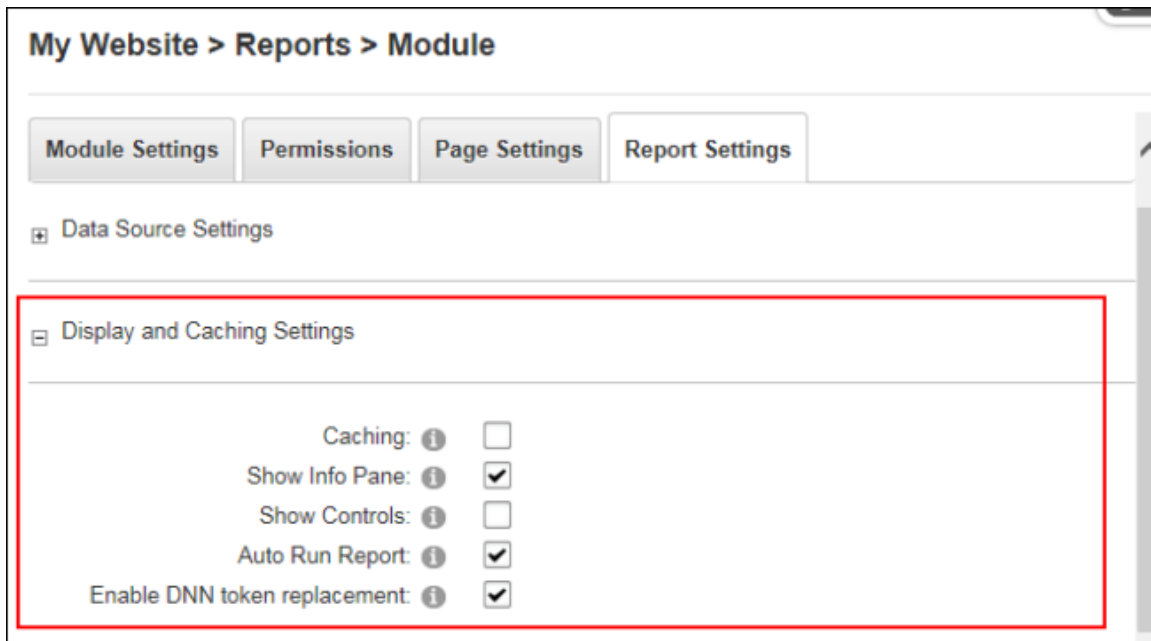
Configuring Display and Caching Settings

Page Editors can configure the Display and Caching Settings for the Reports module. The display settings allow an info pane that shows the report title and description can be displayed and the "Run Report" and "Hide Report Result" links to optionally be displayed. The report can be set to run automatically when a user views the page or the user can choose to manually run the report. This last option, which is useful for time-consuming and resource-intensive reports, requires the "Run Report" link to be displayed. Token replacement can also be enabled including performing calculations for calculated columns.

Data caching can be enabled which will retrieve a single copy of the data which is stored in-memory so that future requests for the same report can use the cached data rather than re-executing the Data Source. However, since this cached copy is always used to satisfy requests, any parameters used in the queries will be disregarded. Therefore parameterized queries are not allowed when caching is in use. If parameters are being used in query and caching are enabled, you will see an error message when the report attempts to run.

Prerequisite. Data Source Settings must first be configured by a Super User before any data is displayed.

1. Go to a Reports module.
2. Select **Manage** > **Settings** from the module actions menu.
3. Go to the **Display and Caching Settings** section of the **Report Settings** tab.
4. **Optional.** At **Caching**, mark the check box to enable data caching and then enter the number of minutes date will be cached for.
5. **Optional.** At **Show Info Pane**, mark the check box to display the report title and description.
6. **Optional.** At **Show Controls**, mark the check box to display the "Run Report" and "Hide Report Result" links.
7. At **Auto Run Report**, mark the check box to automatically run the report when a user views the page - OR - Unmark the check box to enable users to manually run the report. This option is useful for time-consuming and resource-intensive reports. Note: Ensure the "Show Controls" setting above to be selected so users can access the "Run Report" link.
8. At **Enable DNN Token Replacement**, mark the check box to replace tokens including calculated columns in displayed data.
9. At **Active Visualizer**, select a visualizer and complete the additional settings: [See "Configuring Grid Visualizer Settings"](#), [See "Configuring the HTML Template Visualizer Settings"](#), or [See "Configuring the XSL Transformation Visualizer Settings"](#).



10. Click the **Update** button.

Configuring Grid Visualizer Settings

Page Editors can configure the Active Visualizer field located in the Display and Cache section of the Reports Settings page to use a Grid Visualizer to define the way tabular data is displayed. Here the paging and sorting of data can be enabled and column headers

and grid lines can be shown. Finer grained control of the grid appearance can be achieved by attaching CSS Classes and CSS Attributes to the entire grid.

Here's how to set the Grid Visualizer settings on the Active Visualizer:

1. Go to a Reports module.
2. Select **Manage** > **Settings** from the module actions menu.
3. Go to the **Display and Caching Settings** section of the **Report Settings** tab.
4. At **Active Visualizer**, select **Grid Visualizer**.
5. At **Enable Paging?**, mark the check box to enable paging on the grid and then number of rows to be displayed on each page of data into the **Page Size** text box.
- OR - unmark the check box to disable paging and display all data on one page.
6. **Optional.** At **Enable Sorting**, mark the check box to enable users to sort data by clicking on a column header to sort by that column. Note: The "Shown Header" field must be enabled to access sorting.
7. **Optional.** At **Show Header**, mark the check box to display of column headers on the grid.
8. At **Grid Lines**, select **Both** to display both horizontal and vertical lines; select **Horizontal** to display horizontal lines only; select **None** to hide all lines; or select **Vertical** to display vertical lines only.
9. In the **Additional CSS** text box, enter any additional CSS attributes to be placed in the 'style' attribute of the 'table' element produced by this visualizer.
10. In the **CSS Class** text box, enter a list (separated by single spaces) of CSS classes to apply to the 'table' element produced by this visualizer. For more details see CSS Formatting below.

My Website > Reports > Module

Module Settings | Permissions | Page Settings | **Report Settings**

[-] Data Source Settings

[-] Display and Caching Settings

Caching:

Show Info Pane:

Show Controls:

Auto Run Report:

Enable DNN token replacement:

Active Visualizer:

Enable Paging?:

Page Size:

Enable Sorting?:

Show Header?:

Grid Lines:

Additional CSS:

CSS Class:

11. Click the **Update** button.

Configuring the HTML Template Visualizer Settings

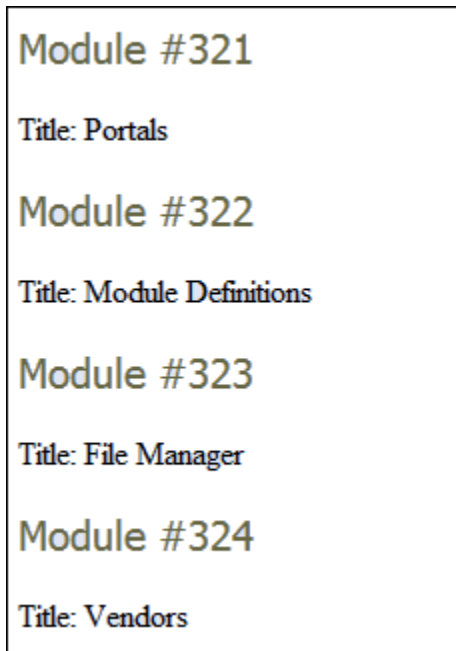
Page Editors can configure the Active Visualizer field located in the Display and Cache section of the Reports Settings page to use the HTML Template Visualizer which uses a HTML Template file to display the data retrieved in the report. The template file is processed once for each row and then rendered as a list. Each Template is processed by locating tokens of the form [Column Name] in the Template and replacing them with the value of Column Name in the returned data-set. The HTML Template Visualizer automatically processes every row retrieved from the data source. Users can display one row only, or a smaller set of rows, by changing the Data Source to only return the desired rows.

Here's how to configure the HTML Template Visualizer settings on the Active Visualizer:

1. Go to a Reports module.
2. Select **Manage** > **Settings** from the module actions menu.
3. Go to the **Display and Caching Settings** section of the **Report Settings** tab.
4. At **Active Visualizer**, select **HTML Template Visualizer**. This displays the Template File fields.
5. At **Template File**, set the link to the HTML template file or upload the new file.
6. Click the **Update** button.

E.g. the following HTML Template was used to produce the output seen below which lists the ID number and title of each module on the site.

```
<h1>Module #[ModuleID]</h1>
<p>Title: [ModuleTitle]</p>
```



Example of HTML Template

Configuring the XSL Transformation Visualizer Settings

Page Editors can configure the Active Visualizer field located in the Display and Cache section of the Reports Settings page to use a XSL Transformation Visualizer which uses an XSL Transformation file and XML produced by the Data Source to produce HTML that is used to display the report.

The XML produced by the Data Source has the following format:



```
<DocumentElement>
<QueryResults>
<Column1Name>Column1Value;/Column1Name><
Column2Name>Column2Value;/Column2Name>
...</QueryResults><QueryResults>...
Same as above, for Row 2 ...
</QueryResults>
```

Use the Show Xml Source command to check the exact XML returned by the Data Source configuration. For example, to reproduce the results displayed in the "Configuring the XSL Transformation Visualizer Settings on the Active Visualizer" tutorial, the following XSL would be used:

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="xml" indent="yes" doctype-
system="http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd" doctype-public="-//W3C//DTD XHTML 1.0 Transitional//EN"/>
<xsl:template match="/DocumentElement/QueryResults">
<h1>Module =<xsl:value-of select="ModuleID"/></h1>
<p>Title: <xsl:value-of select="ModuleTitle"/></p>
</xsl:template>
</xsl:stylesheet>
```

Similar to the HTML Template Visualizer, all of the data retrieved from the Data Source will be displayed in this Visualizer. Users must modify the Data Source configuration to restrict the amount of data displayed.

Here's how to configure the XSL Transformation Visualizer settings on the Active Visualizer of the Reports module:

1. Go to a Reports module.
2. Select  **Manage** >  **Settings** from the module actions menu.
3. Go to the **Display and Caching Settings** section of the **Report Settings** tab.
4. At **Active Visualizer**, select **XSL Transformation Visualizer**.
5. At **Transform File**, select the XSLT file to use to generate the display. Failure to set this option to a file may result in an error.
6. At **Link Type**, select the link to the transform file.
7. Click the **Update** button.

Super Users

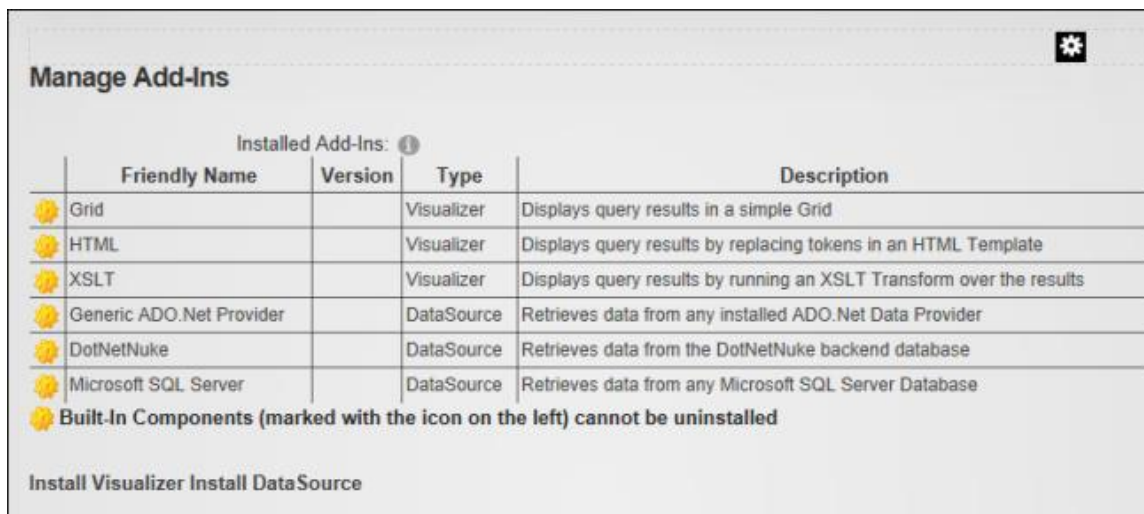
Overview of Data Sources

The Reports module retrieves data using a data source which is a component that is capable of retrieving tabular data. The following data sources can be set and configured by a Super User on the module settings page:

- **Generic ADO.Net Provider Data Source:** Retrieves data from any ADO.Net-compatible data source, such as Oracle, MySQL or Access.
- **DotNetNuke Data Source:** Retrieves data from the database configured for this DNN installation.
- **Microsoft SQL Server Data Source:** Retrieves data from any accessible Microsoft SQL Server instance.

Managing Custom Add-In Data Sources

The Reports module can be extended in two ways by installing custom add-in data sources that allow users to build reports to extract data from any kind of tabular data source. Custom visualizers allow users to display report data in any way they need. To manage the add-ins installed in the system, select **Edit** > **Manage Add-Ins** from the module actions menu. This displays a grid of the installed add-ins.



The screenshot shows a window titled "Manage Add-Ins" with a gear icon in the top right corner. Below the title is a sub-header "Installed Add-Ins: 1" with an information icon. The main content is a table with four columns: "Friendly Name", "Version", "Type", and "Description". The table lists several add-ins, each with a yellow gear icon in the left margin. At the bottom of the table, there is a note: "Built-In Components (marked with the icon on the left) cannot be uninstalled". Below the table, there are two links: "Install Visualizer" and "Install DataSource".

	Friendly Name	Version	Type	Description
	Grid		Visualizer	Displays query results in a simple Grid
	HTML		Visualizer	Displays query results by replacing tokens in an HTML Template
	XSLT		Visualizer	Displays query results by running an XSLT Transform over the results
	Generic ADO.Net Provider		DataSource	Retrieves data from any installed ADO.Net Data Provider
	DotNetNuke		DataSource	Retrieves data from the DotNetNuke backend database
	Microsoft SQL Server		DataSource	Retrieves data from any Microsoft SQL Server Database
	Built-In Components (marked with the icon on the left) cannot be uninstalled			

Install Visualizer Install DataSource

Built-in components list

New add-ins can be installed using the Install Visualizer and Install Data Source commands at the bottom of the page. After clicking one of these links, select a ZIP file containing a Reports Add-In package and install it into the system. Custom Add-Ins installed using these commands can be uninstalled by clicking the **Delete** button in the far left column of the grid. A number of additional Add-Ins are included with the Reports module package, but are not installed by default due to external dependencies. This section details how to install the required components and use these Add-Ins.

Parameterized Queries

All data sources support the use of parameters in their queries. The following parameters are supported by default:

- **@UserID**: Replaced with the ID of the User who is currently viewing the report.
- **@TabID**: Replaced with the ID of the Tab (Page) on which this module is situated. Note: If the same module is copied to a different page, the @ModuleID parameter will be the same but the @TabID parameter will be different.
- **@ModuleID**: Replaced with the ID of the Reports module on which the report is defined.
- **@PortalID**: Replaced with the ID of the site on which the report is running.

To allow users to extend the Reports module queries a simple URL parameter system has been added which allows users to provide a list of QueryString parameters which will be added to the query. The parameter names are in the following format: @url_[QueryString Parameter Name]. Only the parameters specified in the "Allowed URL Parameters" section will be added. In the Allowed URL Parameters field add a comma separated list of

URL QueryString parameters which can be used in the query. URL Parameter names will be prefixed with "url_", plus any data source specific prefix (such as "@" for SQL Server).

Setting the Generic ADO.Net Data Source

The Generic ADO.Net Data Source executes queries against any data source supported by ADO.Net. ADO.Net providers are provided for Oracle, OLE DB, ODBC and SQL Server as part of the ASP.Net environment. Additional providers can be downloaded and installed for many other databases, including [MySQL](http://dev.mysql.com/downloads/connector/net/5.0.html) (see <http://dev.mysql.com/downloads/connector/net/5.0.html>) , [IBM DB2](http://www-306.ibm.com/software/data/db2/windows/dotnet.html) (see <http://www-306.ibm.com/software/data/db2/windows/dotnet.html>) and others. However, unlike the Microsoft SQL Server Data Source, a Connection String must be manually entered because each database provider has its own format for these strings. There are many resources on the web, such as <http://www.connectionstrings.com>, that help build these connection strings. Note: This Data Source does not support the "{oQ}" and "{dO}" tokens since it is assumed the queries are for databases that are not related to DNN.

The Generic ADO.Net Data Source can be configured in the "Data Source Settings" section under Report settings.

Here a report title can be entered. Note: "Title" and "Description" are both displayed on the module when "Show Info Pane" is selected and they are also included in exported content. The URL Querystring parameters to be used in this query must be entered into the "Allowed Url Parameters" text box.

Generic ADO.Net Data Source must be selected as the "Active Data Source" and these setting configured:

- At **ADO.Net Data Provider**, select a data provider from a list of installed ADO.Net data providers.
- In the **Connection String** text box, enter a complete ADO.Net Connection String for connecting to the data source.
- In the **Parameter Prefix** text box, enter a parameter prefix - OR - Leave this field blank as required. Review the section on Parameters ([See "Overview of Data Sources"](#)) and see notes below for more details.

The SQL query can be set by uploading an SQL file or entering an SQL query into the Query text box.

Optional. [See "Setting Converters"](#).

Note: The Generic ADO.Net Data Source supports the four default parameters listed ([See "Overview of Data Sources"](#)). However, since different database providers support parameters in different ways, there are a few caveats:

- They cannot be used with Data Sources such as Microsoft Access which do not support named parameters.
- If the data source expects parameter names to start with a specific prefix (for example, SQL Server requires they be prefixed with the "@" character), that prefix must be entered in the Parameter Prefix setting.

- They cannot be used with Data Sources that expect parameter names to end with a specific suffix.



Setting the DNN Data Source

How to the set the Reports module to use the DNN Data Source which retrieves data by executing a SQL query against the database configured for this DNN application. For example, the following query retrieves a table of all the pages in the site:

```
SELECT * FROM {oQ}Tabs
```

Aside from the standard SQL syntax, this data source accepts two special 'tokens' which will be replaced when the query is run. The "{oQ}" token is replaced with the object qualifier configured for the database. The "{dO}" token is replaced with the database owner configured for the database. Users can also use the longer "{objectQualifier}" and "{databaseOwner}" tokens. This allows the queries to be moved between database configurations without changing the query.

Here's how to set the DNN Data Source:

1. Go to a Reports module.
2. Select  **Manage** >  **Settings** from the module actions menu.
3. Go to the **Data Source Settings** section of the **Report Settings** tab.
4. In the **Title** text box, enter a title for the report. Note: "Title" and "Description" are both displayed on the module when "Show Info Pane" is selected. The Title and Description are also included in export files when Export Content is selected.
5. **Optional.** In the **Description** text box, enter a description of the report.
6. In the **Allowed Url Parameters** text box, enter the URL QueryString parameters to be used in this query.
7. At **Active Data Source**, select **DotNetNuke Data Source**.
8. To set the SQL query, select the SQL query by clicking the **Upload SQL File** and selecting the file, which will then be displayed in the Query text box, from your computer - OR - Enter the SQL query into the **Query** text box.
9. **Optional.** Click the [Test Data Source](#) link to test the data source. If the data runs successfully, the number of records returned will be listed. A message will be displayed detailing any specific errors.
10. **Optional.** Click the [Show Xml Source](#) link view the Xml of the data source. If there is no errors, then the Xml Source used in XSLT transformations of the data is displayed in the Xml Source text box. A message will be displayed detailing any specific errors.
11. **Optional.** [See "Setting Converters"](#).
12. Click the **Update** button.

Parameters: The DNN Data Source supports parameterized queries. E.g. To retrieve the record in the Users table corresponding to the current user, use the query: SELECT * FROM {oQ}Users WHERE UserID = @UserID. [See "Sample Report: Counting Users In Each Role"](#).

Setting the Microsoft SQL Server Data Source

How to set the Microsoft SQL Server Data view in Source option on the Active Date Source field on the Reports module. Note: This Data Source does not support the "{oQ}" and "{dO}" tokens since it is assumed the queries are for databases that are not related to DNN.

1. Go to a Reports module.
2. Select **Manage** > **Settings** from the module actions menu.
3. Go to the **Data Source Settings** section of the **Report Settings** tab.
4. In the **Title** text box, enter a title for the report.
5. **Optional.** In the **Description** text box, enter a description of the report.
6. At **Active Data Source**, select **Microsoft SQL Server Data Source**. This displays the following settings:
7. At **Configuration Mode** select one of the following two options:
 - Select **Generate a Connection String** to generate a connection string based on options entered.
 1. In the **Server Name** text box, enter the name of the server to connect to.
 2. In the **Database**, text box, enter the name of the database to connect to.
 3. Choose from the following authentication options:
 - At **Use Integrated Security**, mark the check box to use the user on the web server that is running DNN (usually the *NT AUTHORITY\NETWORK SERVICE* user account) to determine the exact name of the account that will be used to authenticate with the SQL Server.
 - Enter the **User Name** and **Password** of the user that will be used to authenticate with the SQL Server.
 - Select **Use the Connection String I Enter** and manually enter a complete ADO.Net Connection String for connecting to a Microsoft SQL Server instance into the **Connection String** text box.
8. To set the SQL query, select the SQL query by clicking the **Upload SQL File** and selecting the file, which will then be displayed in the Query text box, from your computer - OR - Enter the SQL query into the **Query** text box.
9. **Optional.** Click the [Test Data Source](#) link to test the data source. If the data runs successfully, the number of records returned will be listed. A message will be displayed detailing any specific errors.
10. **Optional.** Click the [Show Xml Source](#) link view the Xml of the data source. If there is no errors, then the Xml Source used in XSLT transformations of the data is displayed in the Xml Source text box. If there is an error then details of the specific issue will be displayed as an error message.
11. **Optional.** [See "Setting Converters"](#).
12. Click the **Update** button.

Tip: Both the Title and the Description are displayed on the module when Show Info Pane is checked. The title and description are exported with the report when Export Content is selected from the module actions menu.

Parameters: The Microsoft SQL Server Data Source supports four default parameters and a set of custom user defined parameters. [See "Overview of Data Sources"](#).

Setting Converters

Super Users can set the converters that allow users to apply special conversions to the data retrieved by a Data Source before that data reaches the Reports module Visualizer via the Data Source Settings - Converters section of the Report Settings page. The appropriate converter will be applied to each column specified in the list.

1. Go to a Reports module.
2. Select **Manage** > **Settings** from the module actions menu.
3. Go to the **Data Source Settings** section of the **Report Settings** tab and **Maximize** the **Converters** section.
4. In the **HTML Decode** text box, enter a comma-separated list of columns to be HTML Decoded. Note: This field decodes HTML Encoded Data, converting text such as `"Hello"` into **"Hello"**. This is useful for decoding data which was HTML Encoded when it was entered into the database so that the tags will be interpreted by the Web browser when the report is displayed.
5. In the **HTML Encode** text box, enter a comma-separated list of columns to be HTML Encoded. Note: HTML Encodes Data, converting text such as **"Hello"** into `"Hello"`. This is useful for displaying raw HTML tags specified in the data.



The screenshot shows a web interface for configuring converters. At the top, there is a tab labeled 'Converters'. Below it, there are two rows of settings. The first row is for 'Html Decode', with an information icon and a text input field containing the code 'Hello'. The second row is for 'Html Encode', with an information icon and a text input field containing the code '"Hello"'.

Click the **Update** button.

Sample Report: Counting Users In Each Role

This sample report lists the number of users in each role of the site. This tutorial is based on using a new instance of Reports and accepting all default for other fields.

1. Go to a Reports module.
2. Select **Manage** > **Settings** from the module actions menu.
3. Go to the **Data Source Settings** section of the **Report Settings** tab.
4. In the **Title** text box, enter `Users In Roles`.
5. In the **Description** text box, enter `The number of users in each role in the site.`
6. At **Active Data Source**, select **DotNetNuke Data Source**.
7. In the **Query** text box, enter the below query:

```
SELECT R.RoleName, COUNT(UR.UserID) AS 'UsersInRole'  
FROM {oQ}UserRoles AS UR
```

```
INNER JOIN {oQ}Roles AS R ON UR.RoleID = R.RoleID
WHERE PortalID = @PortalID
GROUP BY R.RoleName
```

Awesome Cycles > Reports > Module

Module Settings | Permissions | Page Settings | Report Settings

Data Source Settings

Warning! Anyone who can view this module will be able to view the results of the report. Queries entered here can pull data from any table in the database, be aware of privacy and security issues when writing them.

Title:

Description:

Allowed Url Parameters:

Active Data Source:

Upload SQL File: [Upload](#)

Query:

```
SELECT R.RoleName, COUNT(UR.UserID) AS 'UsersInRole'
FROM {oQ}UserRoles AS UR
INNER JOIN {oQ}Roles AS R ON UR.RoleID = R.RoleID
WHERE PortalID = @PortalID
GROUP BY R.RoleName
```

[Test Data Source](#) [Show Xml Source](#)

8. Go to the **Display and Caching Settings** section and mark the **Show Info Pane** and **Auto Run Report** check boxes.
9. Click the **Update** button.

Reports

Users In Roles

The number of users in each role in the site.

RoleName	UsersInRole
Administrators	1
Registered Users	215
Subscribers	210
Translator (en-US)	4
Designers	3

Sample Report: Counting Users In Each Role

Styling the Grid Visualizer

For even more control over the styling of the grid, the Grid Visualizer defines the following CSS Classes that can be implemented in a site's CSS file:

DNN_Reports_Grid_Header Applied to the header of the Grid: DNN_Reports_Grid_Row Applied to every row of the Grid: DNN_Reports_Grid_AlternatingRow Applied to alternating rows of the Grid: DNN_Reports_Grid_Pager Applied to the pager at the bottom of the Grid.

These classes can be used either to change the style of all Grids in a site by defining them in a Portal CSS file, or by using the CSS Class setting to apply a unique CSS Class to the entire grid. The cascading nature of CSS can also be used to apply styles to particular instances of the Grid Visualizer. For example, if the CSS Class setting for a particular instance of the Grid Visualizer has been set to CustomersReport, then the following CSS will change the background color of alternating rows in the Grid to "white":

```
.CustomersReport .Reports_AlternatingRow
{
background-color: white;
}
```

Troubleshooting: Upgrading

The following warning message may be displayed after upgrading the Reports module: "It looks like you upgraded from a previous version of the Reports module. Due to a minor limitation, there are a few manual steps that must be performed to properly clean up old files. Please read the DesktopModules/Reports/AppCodeCleanup.text file for instructions. This message will disappear when the clean-up is complete".

This warning indicates that a Code Sub-directory could not be found. The below information and steps (as documented in the DesktopModules/Reports/AppCodeCleanup.text file) may correct the issue:

In this version of the Reports Module, the module has been converted from an "App_Code-based" module to a pre-compiled module. If upgraded from a previous version, the App_Code/Reports directory must be removed. The upgrade process should have emptied this folder, but due to some minor limitations, the directory itself cannot simply be removed. Follow the steps below to remove the directory. Removing this directory is not mandatory, as the site should operate correctly if it is not removed. However, it is recommended to ensure no issues arise.

Open the web.config file and remove the entire line:

```
'<add directoryName="Reports" />'
```

from the area between

```
'<codeSubDirectories>'
```

and

```
'</codeSubDirectories>'
```

E.g. If the web.config contains the following:

```
<codeSubDirectories>
```

```
<add directoryName="HTML" />  
<add directoryName="Reports" />  
</codeSubDirectories>
```

After this process, this section should contain the following:

```
<codeSubDirectories>
```

```
<add directoryName="HTML" />  
</codeSubDirectories>
```

Finally, delete the App_Code/Reports folder.