



NetIQ Technical Support

How to Create Custom Reports in the DRA Reporting Tool.

by Sean Jackson

This document was last revised on September, 2002

Introduction

This document details how to create a custom report and make it available in the Directory and Resource Administrator reporting tool. To describe this process, an example will be used to illustrate the steps necessary to create and make available a custom report displaying user accounts that will expire in the next seven days. The report will be called **User Accounts that expire in the next seven days** in the reporting tool and will be available Users Accounts node of the reporting tool after User Account details are successfully imported..

Overview:

The initial steps required to create any custom report with the reporting tool are as follows:

- A. **Add a property to be imported.** If necessary, edit the **UserFields**, **ContactsFields**, or **GroupFields** tables to include the property that you wish to report on.
- B. **Create a custom query** to include the fields that you wish to display in your custom report.
- C. **Create a custom report** using the custom query as the source of the data.
- D. **Add the custom report to the reporting tool.** Edit the **Reports** table to include the newly created report.
- E. **Check the availability of custom report.** Edit the VB code that verifies availability of the new report.
- F. **Make the report available for viewing and printing.** Edit the VB code to add the custom report so it can be viewed and printed from the reporting interface and from the CLI.

A. Add a property to be imported.

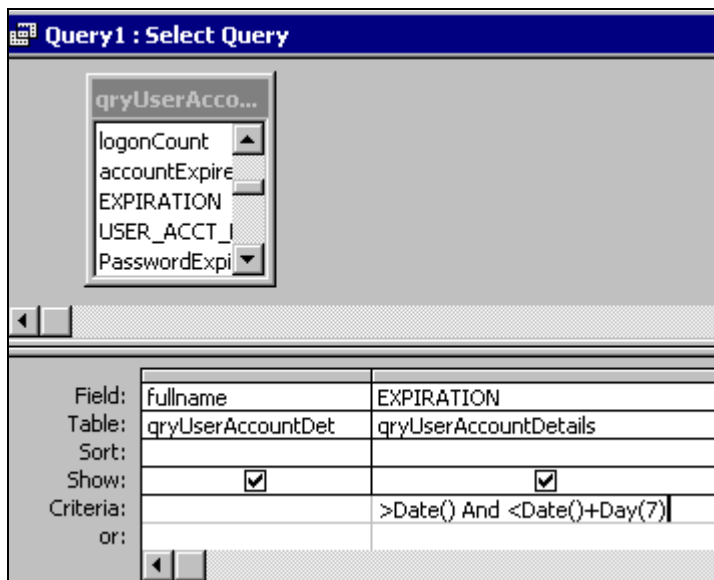
The **UserFields**, **ContactsFields**, and **GroupFields** tables list the properties that are imported into the database and the Users table holds the information itself. If you need a report that lists the **HomePostalAddress** of all the User Accounts in the domain, you would edit the **UserFields** table to include the **HomePostalAddress** attribute in the import process because this is not one of the default imported properties. For the example report **User Accounts that expire in the next seven days**, no modification of the **UserFields** table is necessary as all needed properties are already included in the import process.

UserFields : Table				
	Field	Import	AlwaysImport	UserTable
	homeDirectory	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1
	homeDrive	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1
	homePhone	<input type="checkbox"/>	<input type="checkbox"/>	1
	homePostalAddress	<input type="checkbox"/>	<input type="checkbox"/>	1
	info	<input type="checkbox"/>	<input type="checkbox"/>	1

B. Create a custom query.

A custom query must be created to filter and display the information you wish to include in the new report. Please perform the following steps to create a custom query for the report **User Accounts that expire in the next seven days**:

1. Launch the Directory and Resource Reporting interface.
2. Select **Enter Design Mode** from the **File** drop down menu.
3. Select **Queries** from the **Objects** list.
4. Select **Create query in Design view** and click **Open**.
5. Select the **Queries** tab in the **Show Table** dialog box.
6. Select **qryuseraccountdetails** and click **Add**.
7. Click the **Close** button.
8. Double-click the fields **fullname** and **EXPIRATION** to add them to the displayed table.
9. Enter the expression **>Date() And <Date()+Day(7)** in the criteria field under the **EXPIRATION** column.
10. Save the query as **Expires_in_a_week** and close the **Query 1: Select Query** screen.



C. Create a custom report.

Creating a custom report allows you to display the query in a easy-to-read format. To create the custom report **User Accounts that expire in the next seven days**, please perform the following steps:

1. Select **Reports** from the **Objects** list..
2. Select **Create report by using wizard** and click **Design**.
3. Select **Query: Expires_in_a_week** from the **Table\Queries** drop down list.
4. Click the **>** button twice to move the **Fullname** and **EXPIRATION** fields to the **Selected Fields** list and click **Next**.
5. Click **Next** on the screen and continue with the wizard, customizing the report format as needed.
6. Save the report as **Expires in a Week** when prompted to do so on the last screen of the wizard.

D. Add the custom report to the reporting interface.

The **Reports** table holds a list of all reports available in the reporting interface and is used to build the list of reports that are displayed. It also contains additional information regarding the types of reports and order in which the reports are listed.

1. Select **Tables** from the **Objects** list.
2. Select the **Reports** table and click **Open**.
3. Select **New Record** from the **Insert** drop down menu and enter the values below in the specified columns.

Report Name	Expires in a Week
REPORT_TYPE	1
Order	8
Set?	Selected
USER_DETAIL	Selected
CRITERIA_SELECTION	Criteria None
DESCRIPTION	Accounts that expire in a week

4. Click the **Save** button on the button bar and then select **Close** from the **File** drop down menu.

E. Check the availability of custom report.

While examining a list of reports in the Reporting tool, notice that reports are shown either as available or unavailable, depending on the selected import options. To properly display the availability of a custom report, VB code must be added to the Reporting interface. The following steps describe this process.

1. Select **Modules** from the **Objects** list.
2. Select the **Report** module from the list and click **Design**.
3. Select **UpdateReportsAvailable** from the (**Declarations**) drop down list.
4. Scroll down to the **User Reports** section.

The line **'Last Logon by Account, User Account Details, WTS Account Details, User Account Disk Quota** is followed by two lines of code for each report in the **User Accounts** category. This code checks the availability of all reports in this category. Add the two lines of code below to the **UpdateReportsAvailable** script to verify the availability of the **Accounts that expire in a week** report:

```
sReportName = "Expires in a Week"  
If Not UpdateReportAvailableFlag(bAvail, sReportName) Then GoTo UpdateReportsAvailable_Fail
```

As a result, the **User Reports** section of the script will appear as follows:

```
'Last Logon by Account, User Account Details, WTS Account Details, User Account Disk Quota  
bAvail = IIf(!TotalUserDetails = 0, False, True)  
sReportName = "Last Logon by Account"  
If Not UpdateReportAvailableFlag(bAvail, sReportName) Then GoTo UpdateReportsAvailable_Fail  
sReportName = "User Account Details"  
If Not UpdateReportAvailableFlag(bAvail, sReportName) Then GoTo UpdateReportsAvailable_Fail  
sReportName = "User Account Details (WTS properties)"  
If Not UpdateReportAvailableFlag(bAvail, sReportName) Then GoTo UpdateReportsAvailable_Fail  
sReportName = "Home Volume Disk Quotas"  
If Not UpdateReportAvailableFlag(bAvail, sReportName) Then GoTo UpdateReportsAvailable_Fail  
sReportName = "Expires in a Week"  
If Not UpdateReportAvailableFlag(bAvail, sReportName) Then GoTo UpdateReportsAvailable_Fail
```

F. Make the report available for viewing and printing.

To be able to print and make reports available via the CLI, a case must be added to the appropriate subroutine. For **User Accounts** reports, the subroutine is called **PrintUsers**, **Resources** reports, **PrintResources** and so on. To add a case for the custom report **Expires in a Week**, please perform the following steps:

1. Select **PrintUsers** from the **(Declarations)** drop down list.
2. Add a case for the custom report **Expires in a Week** using the following code:

```
Case "Expires in a Week"  
  If bgHTMLOutput Then  
    Rc = PublishReport(gsReport)  
  Else  
    DoCmd.OpenReport gsReport, gnPrintMode  
  End If
```

The resulting script should appear as follows:

```
Select Case gsReport  
Case "Last Logon Counts by Domain Controller"  
  If gbHTMLOutput Then  
    rc = PublishReport(gsReport)  
  Else  
    DoCmd.OpenReport gsReport, gnPrintMode  
  End If  
  
Case "Expires in a Week"  
  If bgHTMLOutput Then  
    rc = PublishReport(gsReport)  
  Else  
    DoCmd.OpenReport gsReport, gnPrintMode  
  End If  
  
Case "Last Logon by Account"  
  If gbHTMLOutput Then
```

3. Select **Save OaRept** from the **File** drop down menu.
4. Select **Close and Return to Microsoft Access** from the **File** drop down menu.
5. Select **Exit** from the **File** drop down menu to close the DRA Reporting tool.

G. Test the custom report.

1. Launch the Directory and Resource Reporting interface.
2. Select **Import** from the **File** drop down menu.
3. Select **User account details** under the **Account & Definitions** tab.
4. Click the **Import** button.
5. When the import is complete, close the **Last Import Statistics** and the **Import** windows.
6. Expand the **User Accounts** node and select the report **Expires in a Week**.
7. Click the **View** button in the lower right to view the report.

