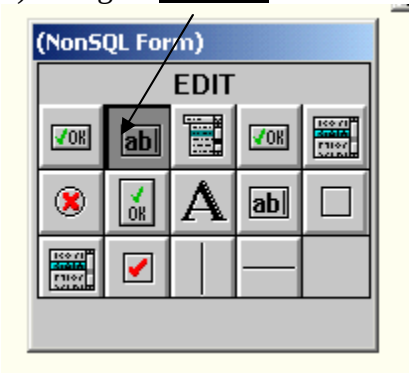
Adding Report Exclusions the Easy Way

It can be very frustrating when running CMS reports if you need to exclude more than one item when using GRS. This month, we're going to share a quick and easy way to add exclusion filters to the front end of your reports.

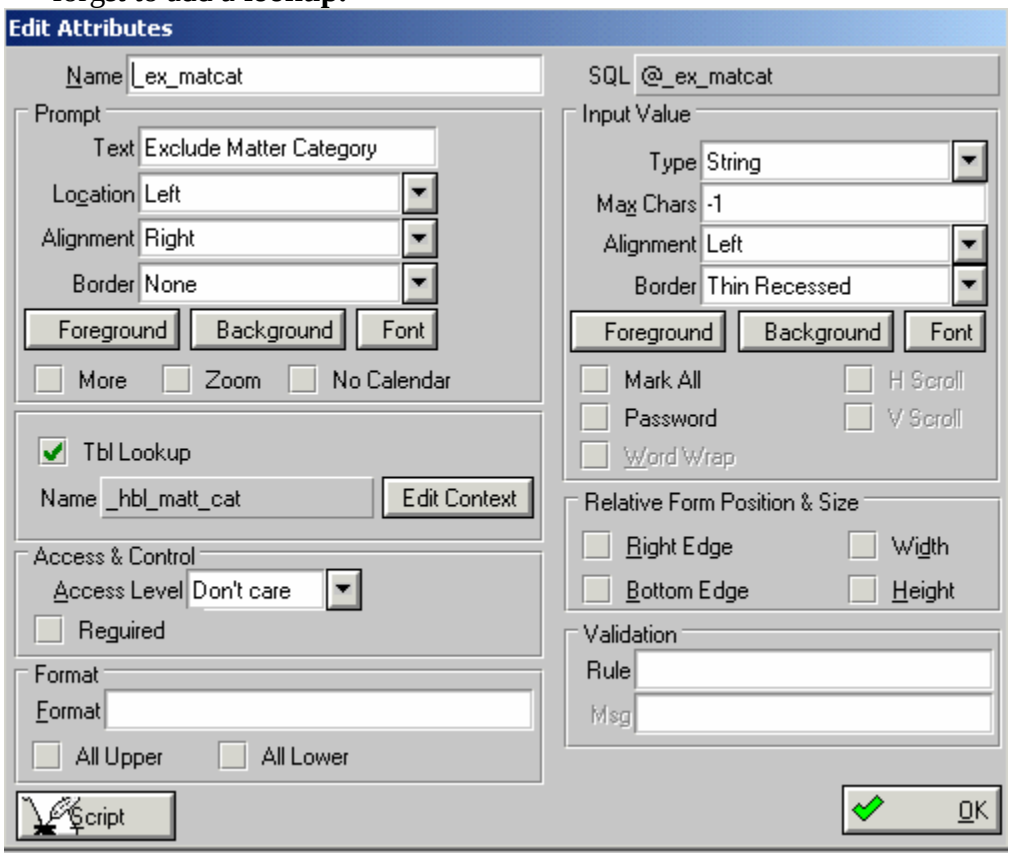
The process involves making changes and adding code to all three parts of the report (front end, view and report). To add these exclusion filters, you'll need to be comfortable making minor report changes. Also, please remember to back up your report and make pop scripts of the view and front end before you do anything. This way, you can always put things back the way they were should something go wrong.

Step 1: Adding your exclusions to the report front end:

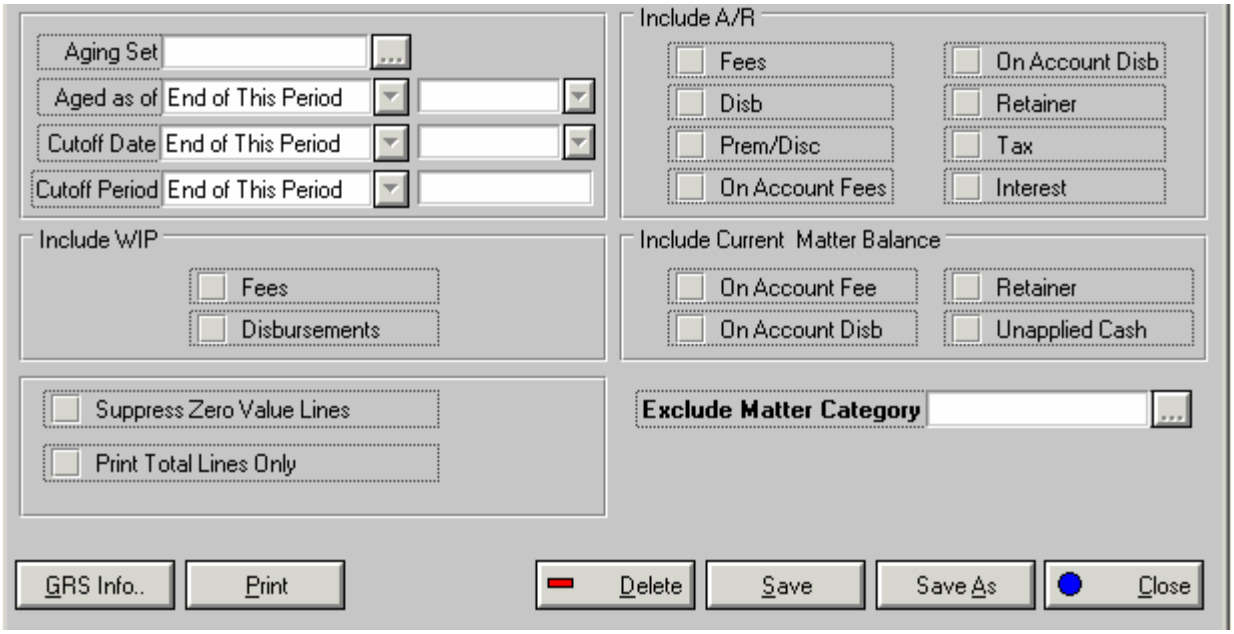
1) Drag an **edit box** onto some blank space on your report front end form.



2) Double click your new field to edit the field attributes. The important thing to note is the name you give the field. I tend to choose _ex_xxxx names to easily identify the field as an exclusion field. Don't forget to add a **lookup.**



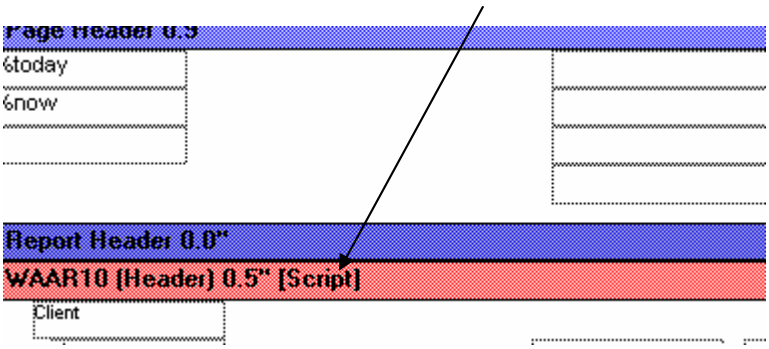
3) The finished field should look something like this. You can adjust the width of the field to meet your needs. That's all we need to do on the report front end. Just repeat this process for all the fields that you want to exclude.



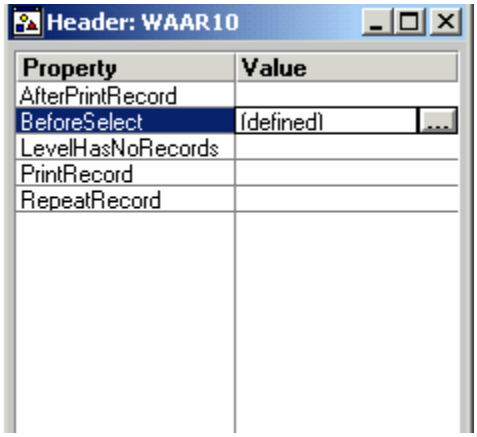
Step 2: Adjusting the Report

In the report, we need to grab the values in our new exclusion filters and pass them to the view. There are several ways to do this, but the easiest and quickest way is public variables.

Now we need to dig in and write some code. We'll be writing all our code for the report in the BeforeSelect Script entry point. You can find that **here**. It's the topmost light red band.



To enter the script, open your property sheet and click the script tab. Then click the ellipsis next to the BeforeSelect script entry point.



1) **Define Public Variables**

The first step is to define a public variable for each exclusion. You can see in the screenshot below that I've defined one for our matter category exclusion. I named it exmatcat$. The syntax is as follows:

public exmatcat$

```
public h%,LastSort$,Options(1 to 16) as integer, NbrRequestedOptions as integer, ZSupp$, tot$
public exmatcat$

function BeforeSelect as integer
    dim sql as new sqlaccess
    dim lf as levelfilters
    dim CutOffDate as new CMSEBDate
    dim AgeDate as new CMSEBDate

    tot$ = thisreport.userparameter("_tot")
'Exclusions
    exmatcat$ = thisreport.userparameter("_ex_matcat")
    if exmatcat$ <> "" then
      exmatcat$ = addquote(exmatcat$)
    end if
```

2) **Add New Function**

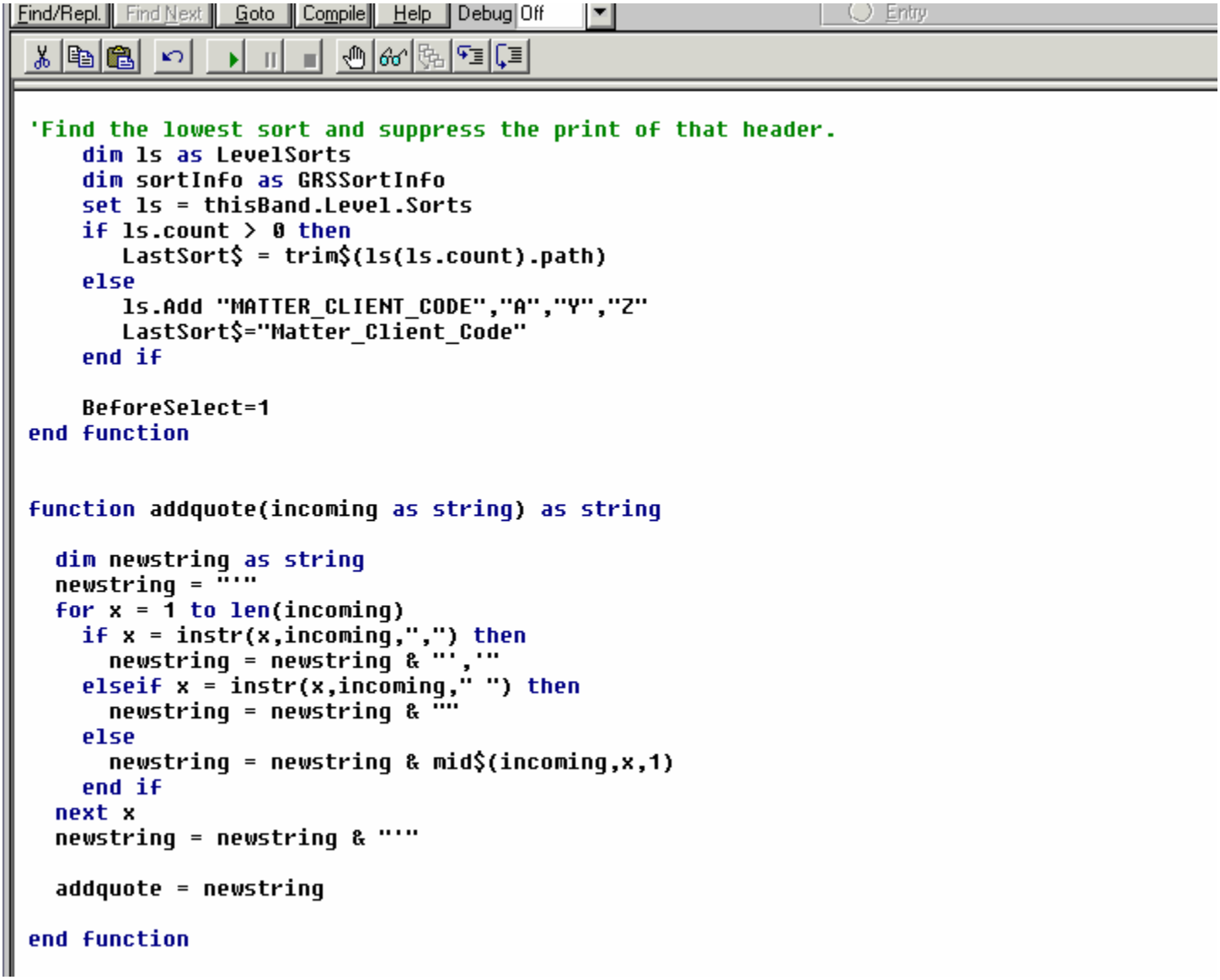
The next step is to copy in this new function which we will be using. You can copy and paste it below the Function BeforeSelect Script. The easiest way to copy this code in is to scroll down to the bottom of the script that is already there and copy it in the white space below that. See screenshot below for placement.

The function code:

```
function addquote(incoming as string) as string

  dim newstring as string
  newstring = """
  for x = 1 to len(incoming)
    if x = instr(x,incoming,",") then
      newstring = newstring & "',"'
    elseif x = instr(x,incoming," ") then
```

```
    newstring = newstring & ""
  else
    newstring = newstring & mid$(incoming,x,1)
  end if
next x
newstring = newstring & """

addquote = newstring

end function
```

The screenshot:




```
'Find the lowest sort and suppress the print of that header.
    dim ls as LevelSorts
    dim sortInfo as GRSSortInfo
    set ls = thisBand.Level.Sorts
    if ls.count > 0 then
       LastSort$ = trim$(ls(ls.count).path)
    else
       ls.Add "MATTER_CLIENT_CODE","A","Y","Z"
       LastSort$="Matter_Client_Code"
    end if

    BeforeSelect=1
end function


function addquote(incoming as string) as string

  dim newstring as string
  newstring = "'"
  for x = 1 to len(incoming)
    if x = instr(x,incoming,",") then
      newstring = newstring & "','"
    elseif x = instr(x,incoming," ") then
      newstring = newstring & ""
    else
      newstring = newstring & mid$(incoming,x,1)
    end if
  next x
  newstring = newstring & "'"

  addquote = newstring

end function
```

3) **Get the Value From the Front End**

The last thing we need to do in the report is get the value the user entered on the front end and pass that to the view. We do that with a very easy piece of code. You can just copy this into your script and change the variable names to match yours. You can paste this after the variable declaration part of the Beforeselect Code. See screenshot below for where I put my code.

The code: (Bold area are the "names" you need to change to match the names you used)

```
'Exclusions
   exmatcat$ = thisreport.userparameter("_exmatcat")
   if exmatcat$ <> "" then
     exmatcat$ = addquote(exmatcat$)
   end if
```

```
public h%,LastSort$,Options(1 to 16) as integer, NbrRequestedOptions as integer, ZSupp$, tot$
public exmatcat$

function BeforeSelect as integer
    dim sql as new sqlaccess
    dim lf as levelfilters
    dim CutOffDate as new CMSEBDate
    dim AgeDate as new CMSEBDate

    tot$ = thisreport.userparameter("_tot")
'Exclusions
    exmatcat$ = thisreport.userparameter("_ex_matcat")
    if exmatcat$ <> "" then
      exmatcat$ = addquote(exmatcat$)
    end if
```

And with that, we're done with the report modification.

Step 2: Adjusting the View

The view changes are the trickiest of all the changes we need to make, but they aren't too terrible The most important thing to remember is you have to make these changes in every component in the view (with the exception of the NonSql component if it exists).

Note: In most CMS views, there will be existing code you need to work around. Some components in views have no code, so you have to create these sections. If you are unsure about what exactly you need to add/change, please email for guidance before making changes to your view.

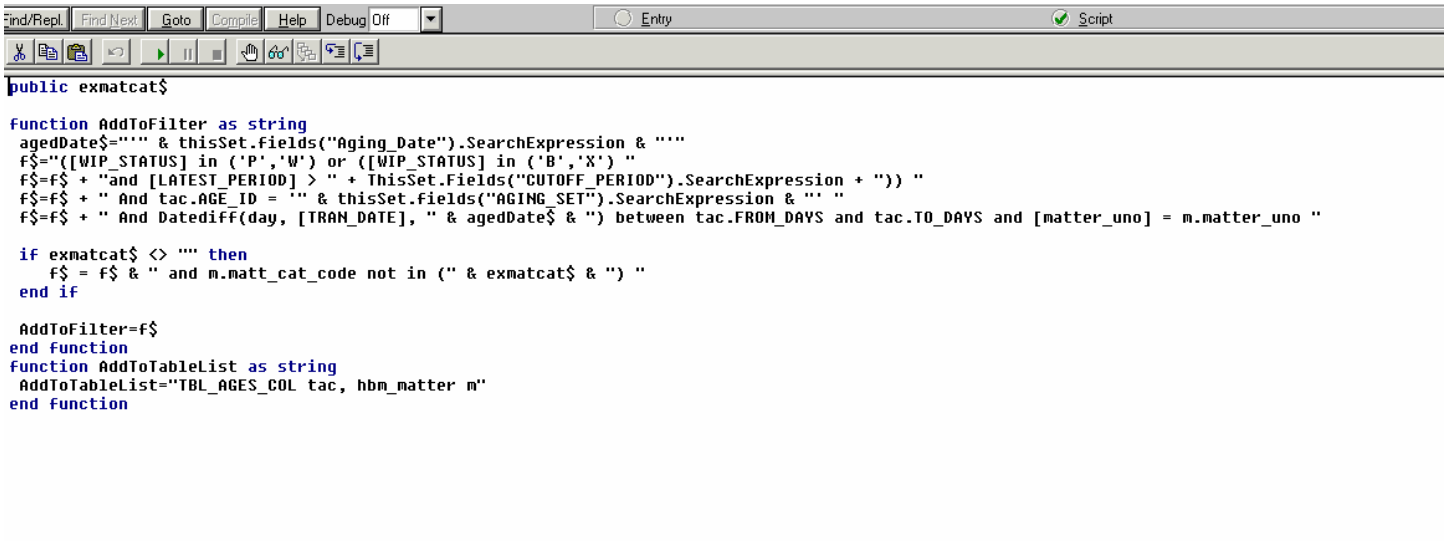
Technical note on CMS Views: The report issues a request for data to the view, and it only asks for fields that are on the report and in the sorts you have requested. If you are running a report at the client level but want to exclude a matter category, then we have to add HBM_MATTER to the table list so matter category is available to be excluded.

< 7.0 users start here

1) **Add the public variable you created**

   public exmatcat$

   See the screenshot below for a 7.0 installation, including all the code you need to add.



2) Adjust the AddtoTableList

The first step in the view is to add the necessary tables to get us to the field we are trying to exclude. In the case of matter category, we're know that field is in HBM_MATTER, so we add that table to the list.

Existing table already there:

function AddToTableList as string
AddToTableList="TBL_AGES_COL tac**, hbm_matter m**"
end function

Function does not exist:

function AddToTableList as string
AddToTableList="**hbm_matter m"**
end function

3) Adjust the AddToFilter

We need to do 2 things in the AddtoFilter. First, add the join for the table we added to the AddtoTableList and then add our exclusion code. AddtoFilter code is universally different, so you might not be able to copy and paste this code in. In the example below, a variable called f$ is used to build the AddtoFilter, but it might be Filter$ or something entirely different.

Sample AddtoFilter (existing code):

function AddToFilter as string

agedDate$ = "'" & thisSet.fields("Aging_Date").SearchExpression & "'"
f$="([WIP_STATUS] in ('P','W') or ([WIP_STATUS] in ('B','X') "
f$=f$ + "and [LATEST_PERIOD] > " + ThisSet.Fields("CUTOFF_PERIOD").SearchExpression + ")) "
f$=f$ + " And tac.AGE_ID = '" & thisSet.fields("AGING_SET").SearchExpression & "' "
f$=f$ + " And Datediff(day, [TRAN_DATE], " & agedDate$ & ") between tac.FROM_DAYS and tac.TO_DAYS and **[matter_uno] = m.matter_uno** "

if exmatcat$ <> "" then
f$ = f$ & " and m.matt_cat_code not in (" & exmatcat$ & ") "
end if

AddToFilter=f$
end function

Sample AddtoFilter (no code):

function AddToFilter as string

f$ = "[matter_uno] = m.matter_uno "

if exmatcat$ <> "" then
f$ = f$ & " and m.matt_cat_code not in (" & exmatcat$ & ") "
end if

AddToFilter=f$
end function

> 7.0 users start here

4) Add the public variable you created

public exmatcat$

5) Adjust the AddtoFromClause

The first step in the view is to add the necessary tables to get us to the field we are trying to exclude. In the case of matter category, we know that field is in HBM_MATTER, so we add that table to the list.

Existing table already there:

function AddToFromClause as string
AddToFromClause="join TBL_AGES_COL tac**, join hbm_matter m on [matter_uno] = m.matter_uno"**
end function

Function does not exist:

function AddToFromClause as string
AddToFromClause="**join hbm_matter m on [matter_uno] = m.matter_uno"**
end function

6) Adjust the AddToFilter

We only need to do one thing in the AddtoFilter for >7.0 installations and that is to add our exclusion code. AddtoFilter code is universally different, so you might not be able to copy and paste this code in. In the example below, a variable called f$ is used to build the AddtoFilter, but it might be Filter$ or something entirely different.

Sample AddtoFilter (existing code):

```
function AddToFilter as string

 agedDate$ = "'" & thisSet.fields("Aging_Date").SearchExpression & "'"
 f$="([WIP_STATUS] in ('P','W') or ([WIP_STATUS] in ('B','X') "
 f$=f$ + "and [LATEST_PERIOD] > " + ThisSet.Fields("CUTOFF_PERIOD").SearchExpression + ")) "
 f$=f$ + " And tac.AGE_ID = '" & thisSet.fields("AGING_SET").SearchExpression & "' "
 f$=f$ + " And Datediff(day, [TRAN_DATE], " & agedDate$ & ") between tac.FROM_DAYS and
tac.TO_DAYS"

 if exmatcat$ <> "" then
    f$ = f$ & " and m.matt_cat_code not in (" & exmatcat$ & ") "
 end if

 AddToFilter=f$
end function
```

Sample AddtoFilter (no code):

```
function AddToFilter as string

if exmatcat$ <> "" then
  f$ = f$ & " and m.matt_cat_code not in (" & exmatcat$ & ") "
end if

 AddToFilter=f$
end function
```

It looks more difficult than it is. Once you get a little experience adding these exclusions, you will be able to add them in less than 10 minutes and save your users lots of time.

Remember to make the views changes to every component in the view! If you have any questions, please feel free to email me at measterling@aac-us.com for clarification of anything in this tip.