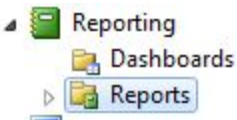


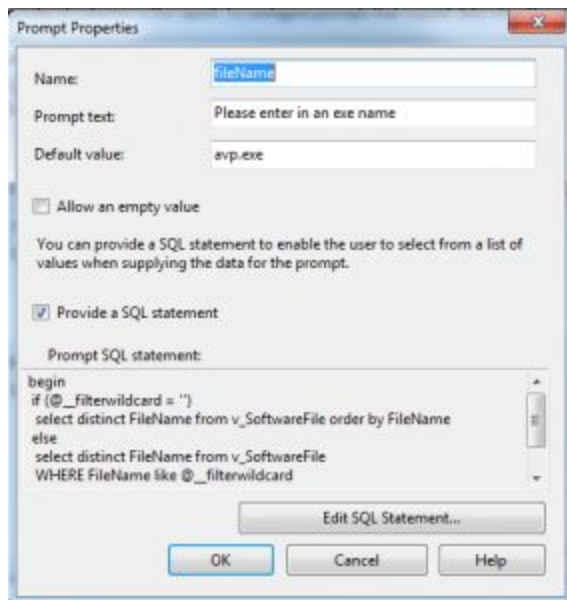
Adding Prompts to a SCCM Report

This is something simple to do and adds a lot more functionality to an otherwise static report. I will walk you through the basics of creating a prompt for a SCCM report.

- 1) Go to reporting and create a new report



- 2) Give your report a Name, Category and then click Edit SQL Statement.
- 3) Paste in your code for your SQL Statement and then click the Prompts button
- 4) A new window will appear. Clicking the Add button will bring up the Prompt Properties page. Complete the Name (used as @Name in your report SQL code), Prompt Text (the hint the user will see for the prompt), Default Value and if needed be click the Provide SQL Statement checkbox. Your window should look something like this.



- 5) Click OK to close the window and repeat step 4 as many times as you need.

You have now successfully added prompts (variables) to your SCCM report.

Now that you know the basics about adding prompts to a report, let's have a look at the proper usage of these little gems. Generally a prompt is used to allow the user to select a value that will affect the outcome of the report. This can be useful if you want to provide the user with more accurate / specific data.

SCCM allows the use of SQL Statements in your prompts.

Take a look at my prompt for the EXE name for metered software.

```
Begin
if (@__filterwildcard = '')
select distinct FileName from v_SoftwareFile order by FileName
else
select distinct FileName from v_SoftwareFile
WHERE FileName like @__filterwildcard
order by FileName
end
```

By making use of the @__filterwildcard provided with all prompts I am able to use the input the user might have entered into the prompt field to narrow down a search, or do a full search if it was left blank.

Note: the @__filterwildcard automatically wraps itself in '(s)' so you won't be able to wrap it like this '@__filterwildcard' much to my surprise the first time playing with it.

Like I mentioned before you can access all prompts in your SQL code by making use of the following naming convention. @[PromptName]. So the filename prompt in the example above will be accessible in our SQL code as @fileName, which allows us to test our code in SSMS with a simple declaration of a variable.

That's basically all you need to know about adding prompts to SQL in SCCM (or at least all I can think of sharing).