

HOW TO CREATE TOAD® REPORT MANAGER REPORTS

LESSON TWO

**A step by step guide to calling a stored
procedure from a TOAD® Report**

**By
Gary Piper**

August 2009

TABLE OF CONTENTS

1	Prerequisite	3
2	Lesson overview	3
3	Copy the existing Page Requests report	4
4	Create a reports procedure	4
5	Modify the SQL for the inclusion into the reports procedure.....	5
6	Change the Report Manager SQL to a PLSQL call	6
7	Some final report editing.....	7
8	Added benefits of using a procedure call.....	8
9	Disclaimer.....	8
10	Example PLSQL Code.....	10

1 Prerequisite

It is recommended that prior to completing this lesson that you:

- ❖ have completed and understood lesson one - How to Create Toad® Report Manager Reports
- ❖ possess a basic knowledge of PLSQL
- ❖ know how to copy a TOAD Report Manager report

2 Lesson overview

This lesson will take you through the steps required to create a report that uses the output from a stored procedure to create the report rather than straight SQL as is normally required.

This process can be used for more complex reports and for reports where you want to wrap the SQL to limit the availability of the raw SQL.

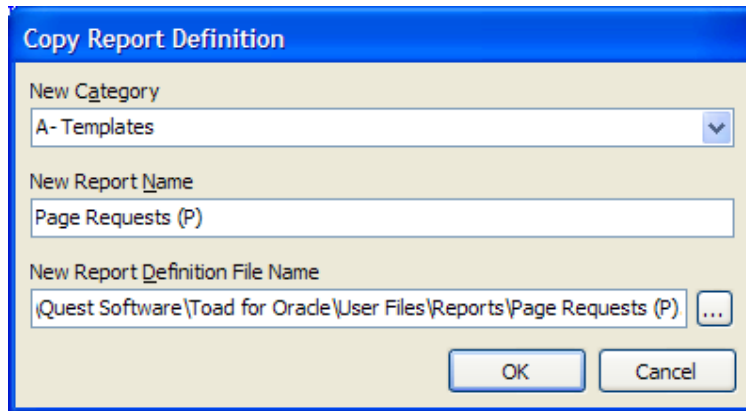
In this lesson we will:

- ❖ copy the page request report created in lesson one
- ❖ replace the Oracle SID in the report header with the application site name as this is more meaningful to the end user
- ❖ add your name to the report

All templates and reports created in this lesson will be provided in a TRD format.

3 Copy the existing Page Requests report

The first step is to create a copy the “Page Requests” report from lesson one.



To differentiate the two reports I have named the new report “Page Requests (P)” where the (P) symbolises (PLSQL) format.

Once the report entry is created, exit the TOAD® Reports Manager to save the entry.

4 Create a reports procedure

The next step is to create the procedure that will be called and provide the report content.

The example procedure is provided at the end of this document as part of this lesson and will be provided on the piper-rx.com web site:

[*reports_tutorial_two.pkb*](#)

Create the package and package body by launching TOAD® and opening the above file.

5 Modify the SQL for the inclusion into the reports procedure

Whilst the SQL is provided in the procedure [reports_tutorial_two.pkb](#) the following detail steps you through the changes made (marked in red in the revised SQL version) between the SQL from the original report and the SQL in the procedure.

Original SQL

```
SELECT to_char(sysdate, 'DD-Mon-YY fm(Day)') page_heading_date,
       :sample_date v_sample_date,
       database_name.sid_name page_heading_sid_name,
       fu.user_name,
       sum(isess.counter) page_requests
FROM   icx.icx_sessions isess,
       applsys.fnd_user fu,
       (SELECT name sid_name
        FROM gv$database
        WHERE inst_id = 1) database_name
WHERE  isess.user_id = fu.user_id
       and trunc(isess.first_connect) = to_date(:SAMPLE_DATE, 'DD-Mon-YY')
GROUP by fu.user_name,
         database_name.sid_name
ORDER by 4 desc
```

Revised SQL (changes shown in red)

```
SELECT to_char(sysdate, 'DD-Mon-YY fm(Day)') page_heading_date,
       v_sample_date sample_date,
       v_site_name page_heading_site_name,
       v_my_name my_name,
       fu.user_name,
       sum(isess.counter) page_requests
FROM   icx.icx_sessions isess,
       applsys.fnd_user fu
WHERE  isess.user_id = fu.user_id
       and trunc(isess.first_connect) = to_date(v_sample_date, 'DD-Mon-YY')
GROUP by fu.user_name
ORDER by 5 desc;
```

The changes are:

- the variable **v_site_name** has been added, the value of which is provided in the PLSQL
- the sub query for database name has been removed as we are now substituting the SID with the application site name **v_site_name**
- the variable **v_my_name** has been added to show how additional values can be added. The procedure runs the following SQL to get the value:

```
▪ SELECT 'Gary Piper'
▪ INTO v_my_name
▪ FROM dua
```

- removed the sub query that gets the SID name
- changed the **:SAMPLE_DATE** variable to **v_sample_date** which is the input variable that will be passed to the package
- changed the sort order from 4 to 5

6 Change the Report Manager SQL to a PLSQL call

Once you have compiled the procedure, you can replace the existing SQL in the Page Requests (P) report with the following statement:

```
DECLARE

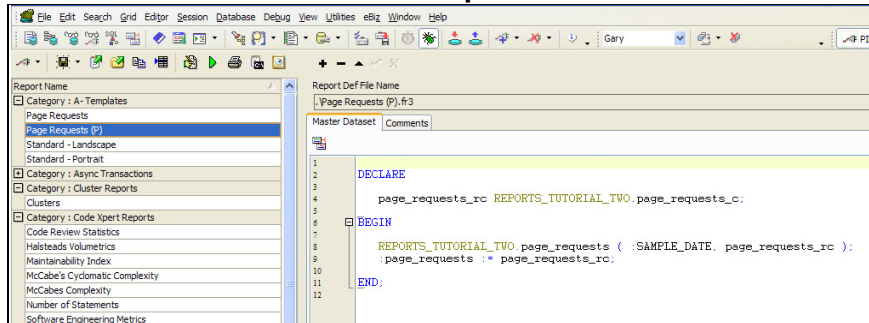
page_requests_rc REPORTS_TUTORIAL_TWO.page_requests_c;

BEGIN

REPORTS_TUTORIAL_TWO.page_requests ( :SAMPLE_DATE, page_requests_rc );
:page_requests := page_requests_rc;

END;
```

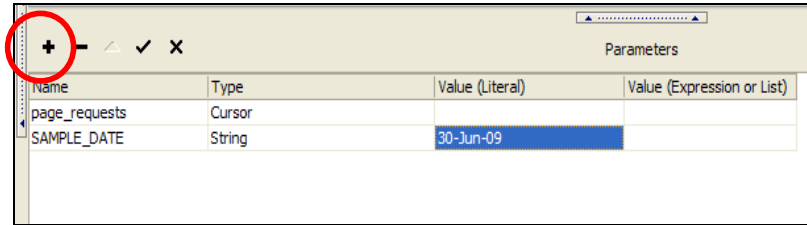
Example



Note: The variable: SAMPLE_DATE is still used on the TOAD report manager page

You now need to add the cursor parameter to the report parameter list.

A new parameter is added by selecting the + sign.



The cursor name is the name of the procedure being called.

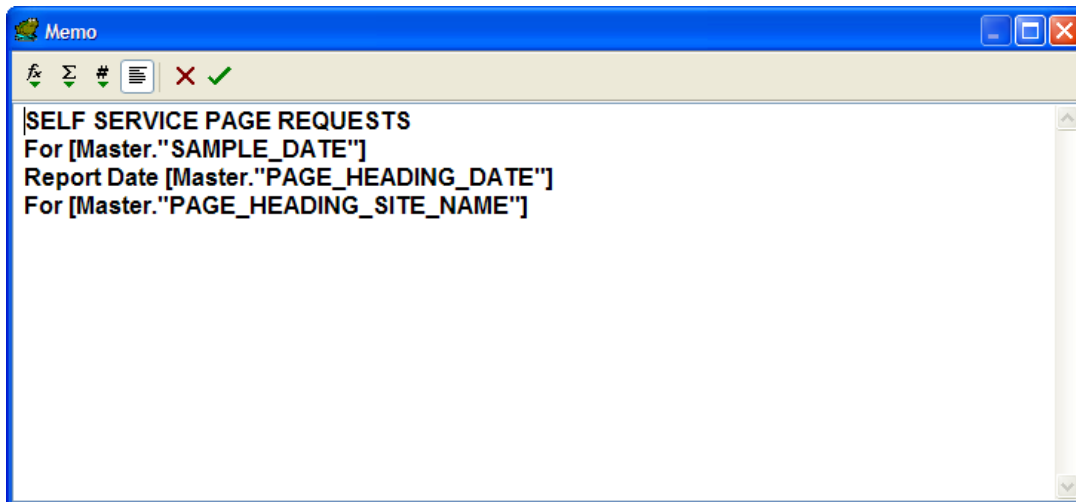
Note: There is no value (literal) for a cursor

Tip: Don't forget to save the changes by exiting the Reports Manager...

7 Some final report editing

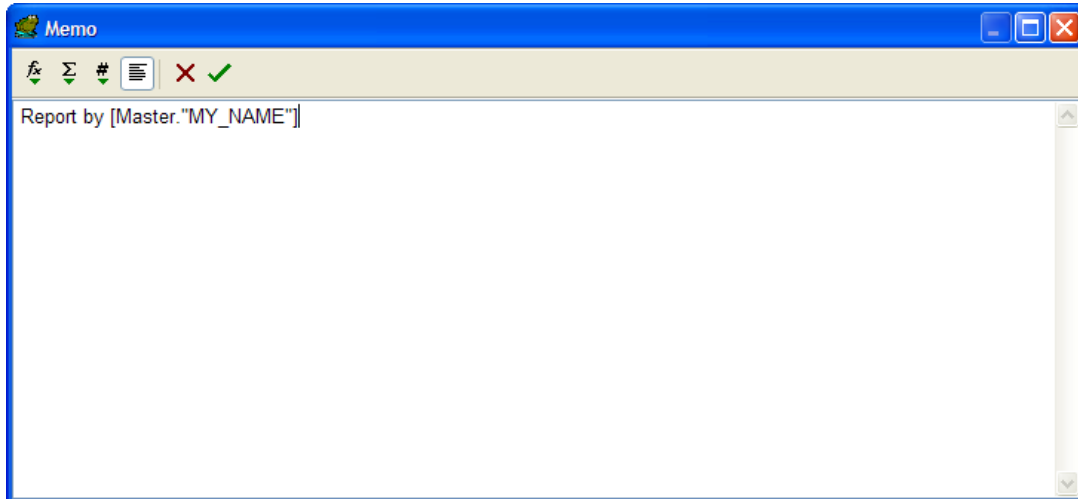
As we have changed some of the variable names and added additional columns to the report, some final report editing is required.

- Header changes:
 - ❖ V_SAMPLE_DATE to SAMPLE_DATE
 - ❖ PAGE_HEADING_SID_NAME to PAGE_HEADING_SITE_NAME



- Footer Changes (optional)

Change the footer to include your name passed from the procedure



Your report should now look something like this:

SELF SERVICE PAGE REQUESTS	
For 30-Jun-09	
Report Date 02-Jul-09 (Thursday)	
For APPS 12i	
User Name	Page Requests
SYSADMIN	42
GUEST	3

8 Added benefits of using a procedure call

There are many added benefits when using a procedure call, these include:

- ❖ you can add any number of reports to the one procedure
- ❖ you can perform complex report queries above and beyond a single SQL statement
- ❖ you can wrap the procedure to hide your source SQL
- ❖ in a procedure can create a temporary table, report on it and then drop the table
- ❖ etc. etc.....

9 Disclaimer

The material contained in this document is provided by the author "as is" and any express or implied warranties, including, but not limited to, any implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event

shall the author be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of any content or information, even if advised of the possibility of such damage. It is always recommended that you seek independent, professional advice before implementing any ideas or changes to ensure that they are appropriate.

TOAD® is a registered trademark of Quest Software
© 2009 G Piper All Rights Reserved

10 Example PLSQL Code

```

CREATE OR REPLACE PACKAGE REPORTS_TUTORIAL_TWO IS

/*****
***
PIPER_RX PAM REPORTS - TUTORIAL TWO
By Gary Piper

(C) Copyright G.Piper 2009
All rights reserved

File Name:  REPORTS_TUTORIAL_TWO:

-----
---
+Version 1.0 G.Piper Jul-09  Initial Creation

****
****

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU Affero General Public License as
published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU Affero General Public License for more details.

a copy of the GNU Affero General Public Licenses provided is part of
the
down load Terms and Conditions available on our site www.piper-rx.com
or
you can refer to www.gnu.org/licenses/gpl.html

****/

TYPE page_requests_c      IS REF CURSOR;

PROCEDURE get_site_name  ( v_site_name          OUT  VARCHAR2);

PROCEDURE page_requests  ( v_sample_date varchar2,
                           page_requests_rc OUT page_requests_c );

END REPORTS_TUTORIAL_TWO;
/

CREATE OR REPLACE PACKAGE BODY REPORTS_TUTORIAL_TWO AS

v_site_name      varchar2(65);

-- ****
-- ****

PROCEDURE get_site_name (v_site_name OUT VARCHAR2 ) AS

BEGIN
  SELECT nvl(fpov.profile_option_value, 'SITE NAME NOT SET' )
         INTO v_site_name
         FROM applsys.fnd_profile_option_values fpov

```

```
WHERE fpov.level_id(+) = 10001
      and fpov.application_id = 0
      and fpov.profile_option_id = 125;

END get_site_name;

-- *****
-- *****

PROCEDURE page_requests ( v_sample_date varchar2,
                          page_requests_rc OUT page_requests_c ) AS
v_my_name  varchar2(30);

BEGIN

    REPORTS_TUTORIAL_TWO.get_site_name (v_site_name);

    SELECT 'Gary Piper'
           INTO v_my_name
           FROM dual;

    OPEN page_requests_rc FOR
    SELECT to_char(sysdate, 'DD-Mon-YY fm(Day)') page_heading_date,
           v_sample_date sample_date,
           v_site_name page_heading_site_name,
           v_my_name my_name,
           fu.user_name,
           sum(isess.counter) page_requests
    FROM   icx.icx_sessions isess,
           applsys.fnd_user fu
    WHERE  isess.user_id = fu.user_id
           and trunc(isess.first_connect) =
                to_date(v_sample_date, 'DD-Mon-YY')
    GROUP by fu.user_name
    ORDER by 5 desc;

END page_requests;

END REPORTS_TUTORIAL_TWO;
```

Note: This code is a simple example only and should be modified to conform to your site's coding standards.