

Paper 079-2011

Detail Forms w/ SAS Stored Procedures

David Wells, Sinclair Community College, Dayton, OH, USA

ABSTRACT

Many times, the viewer of a list of data wants detailed information for a few items in the data, but not all the data. In SAS® Web Report Studio, it is easy to give drill-in capability to reports, but some reports cannot be done with SAS Web Report Studio. However, reports that are created from SAS® Stored Processes can be as detailed or as brief as you want, and can be forced to look like anything you can think up.

I allow drilling into data from a stored procedure by linking to a second stored procedure, using parameters in the link statement from the first stored procedure's listing of the data.

INTRODUCTION

An example of the type of report that we commonly use to drill into the data is one used by administrators showing error logs. In our student web portal, we have a database table that contains a listing of all the errors seen by the application. Sometimes, this list can be very long, so we summarize the errors by type of error. We have a link on this summary that will expand (or give detail to) the error type.

We start off by filtering the number of days to look back as a parameter to our first stored process (see Figure 1):

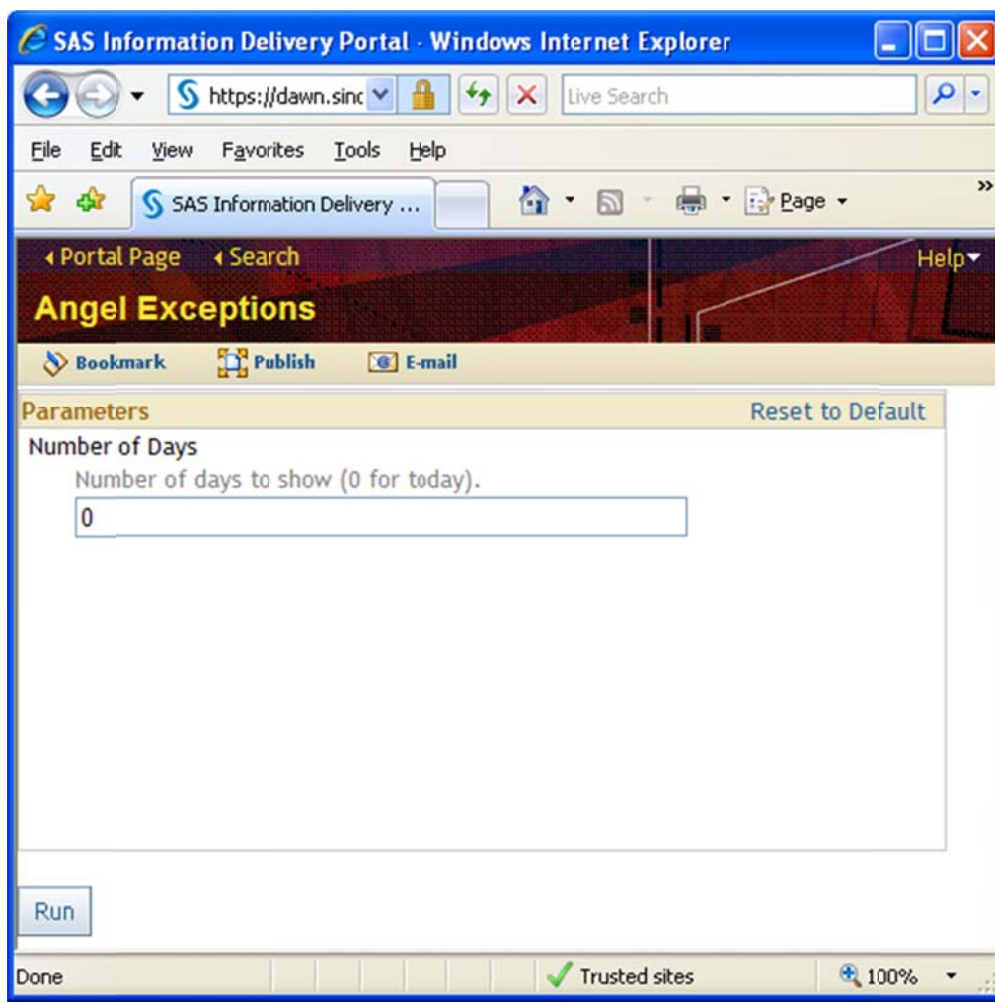


Figure 1. Parameter Selection for Initial Stored Process

Next we present the user with a list of Exception Types which contains the type, count and an “expand” link (see Figure 2):

Angel Exception Types for the last 0 days

Obs	ExceptionType	Count	link
1	CLL.S3.Framework.Exceptions.FrameworkException	1072	expand
2	System.Data.SqlClient.SqlException	618	expand
3	CLL.S3.Framework.Database.DatabaseException	607	expand
4	System.Reflection.TargetInvocationException	420	expand
5	System.Runtime.InteropServices.COMException	327	expand
6	ASP Error - Active Server Pages	77	expand
7	System.NullReferenceException	61	expand
8	System.Threading.ThreadAbortException	45	expand
9	ASP Error - Microsoft OLE DB Provider for SQL Server	21	expand
10	System.ArgumentNullException	19	expand
11	ASP Error - ADODB.Recordset	16	expand

Figure 2. List of Error Types

Finally, once the user has expanded the error, they see the detailed error message with a count of the number of times it has occurred within the number of days specified (see Figure 3):

Angel Exception Messages for the last 0 days
For Exception Type: System.InvalidOperationException

Obs	Message	Count
1	This SqlTransaction has completed; it is no longer usable.	2

Figure 3. Detail of Error Type

CALLING A STORED PROCESS WITH PARAMETERS

The trick to creating the “link” column is that the report is going to be displayed in the SAS Information Portal and thus is an HTML display. So quite simply, we create an HTML link as a data item that will call the “detail” stored process with all the needed parameters.

Recall, in an HTML FORM, when using the GET method, it passes all the parameters from the form on the address line separated by ampersand signs (i.e., “&”). The SAS Stored Process Viewer uses the same convention for its parameters. These parameters, which can be captured by the detail stored process by declaring a global variable by the same name as the parameter, provide the means to pass information from one stored process to another.

```

*ProcessBody;
%GLOBAL days;
%STPBEGIN;

LIBNAME ANGEL /* put your connection information here */;

/* Get the list of Exception Types */
PROC SQL;

    CREATE TABLE ExceptionCount AS
    SELECT ExceptionType , count(*) AS Count
    FROM ANGEL.exceptionlog
    WHERE "&sysdate."d - &days. <= DATEPART(timestamp)
    GROUP BY ExceptionType
    ORDER BY Count DESC;

QUIT;

/*
https://yoursasserver/SASStoredProcess/do?_program=%2FSinclair%2FStored+Processes%2
FAngel+Exceptions+Detail+Levell&_action=update%2Cnewwindow&_promptkey=681062127
*/

DATA ExceptionCountWithLink;
    LENGTH link $ 2000;
    SET ExceptionCount;

/* Build the link to the detail stored procedure */
    link='<A
HREF="/SASStoredProcess/do?_program=%2FSinclair%2FStored+Processes%2FAngel+Exceptio
ns+Detail+Levell&days=' || trim("&days.") || '&exceptionType=' || ExceptionType ||
' ">expand</a>';
RUN;

/* Present the Data */
TITLE1 "Angel Exception Types for the last &days. days";
FOOTNOTE1 "test";

PROC PRINT data=ExceptionCountWithLink;
    VAR ExceptionType Count Link;
RUN;

%STPEND;

```

Figure 4. Listing for Calling Stored Process

In the code (see Figure 4 above), you will notice that I have a commented out a URL. That is just a crutch to help in the building of the Link column. This URL is created by running the Detail Stored Process from the portal (manually) and copying the address line from the browser. As you can see, I removed some of the “generated” arguments (i.e., `_action` and `_promptkey`). Also removed are the HTTPS tag and the server and port information, so that the reference is done as a relative instead of a full URL. This provides the ability to move the stored process from a development environment easily to the production environment.

THE DETAIL STORED PROCESS

The other file that needs to be created is that of the Detail Stored Process. As shown above, this presents the user with the detail (or drilled into) data. As a point of order, I have placed this second in the paper, but I generally build this process first. This is used in the Calling Stored Process to help build the link.

```
*ProcessBody;
%GLOBAL days;
%GLOBAL exceptionType;
%STPBEGIN;

LIBNAME ANGEL /* put your connection information here */;

PROC SQL;
  /* Build the Detail information */
  CREATE TABLE ExceptionCount AS
    SELECT Message , count(*) AS Count
    FROM ANGEL.exceptionlog
    WHERE "&sysdate."d - &days. <= DATEPART(timestamp)
      AND ExceptionType = "&exceptionType."
    GROUP BY Message
    ORDER BY Count desc;
QUIT;

/* Present the Detail Information */
TITLE1 "Angel Exception Messages for the last &days. days";
TITLE2 "For Exception Type: &exceptionType.";

PROC PRINT DATA=ExceptionCount;
RUN;
%STPEND;
```

Figure 5. Listing for Detailed Stored Process

As you can see (see Figure 5. above), the parameters passed from the first stored process are global variables in the detail process. These parameters provide filters for the data to get detailed messages about our errors.

CONCLUSION

Admittedly, the example I have walked through could be done in Web Report Studio, but it should give an idea of how you can provide detailed information on a list of data with very little coding needed. Other, more formatted output, in the detail stored process can be presented in ways differing from a simple PROC PRINT. For example, at Sinclair Community College, we have another report that shows a list of applicants at the college. It includes a "detail" page that has the look and feel of the paper application form. Our registration office uses it to print the application and "put it on file". This formatted output was done by creating an HTML page using put statements containing the HTML tags in a DATA step.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

David Wells
 Sinclair Community College
 444 W. Third St
 Dayton, OH 45402
 (937) 512-2454
 Email: David.Wells@sinclair.edu
 Web: www.sinclair.edu

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.