

SPCaller - A Component That Calls Stored Procedures

Posted At : July 4, 2008 12:50 PM | Posted By : Bob Silverberg

Related Categories: SPCaller, ColdFusion

SPCaller is a component that automatically calls stored procedures (SPs) for you. I wrote it several years ago, and have been using in most of my projects since then. Well, that's not exactly true. I used it in most of my projects until about a year ago, at which point I started drinking the ORM kool-aid. Now I hardly use any SPs any more.

Still, if you're using SPs it is a very useful tool. It's probably saved me hundreds of hours over its life span, so I thought it worth releasing as my first ever Open Source project.

Here's a bit of background on it. When I first started using ColdFusion, almost 10 years ago, I also started using MS SQL Server (version 7 at the time). I quickly adopted the process of doing almost all of my database work via SPs. I found that there was very little that I couldn't do inside an SP, and at the time it seemed like a good idea for a number of reasons. I also developed a method of introspecting the database to generate the bulk of my SP code, so I was able to develop sites very quickly. The one thing that bothered me was the fact that I had to define all of my SP's parameters in the T-SQL code, and I also had to specify all of them in ColdFusion when calling my SP, either via cfquery or cfstoredproc. That seemed like duplication to me, so I was determined to do something about it, and that was how SPCaller was born.

It has gone through many iterations and changes over the years, from custom tag, to UDF, and finally to CFC. I won't bore you with the details of its original, abominable implementation, but I will describe what it does and how it works.

The object has one main method, called, imaginatively enough, callSP(). There are a few helper methods, but we don't need to go into those here. callSP() accepts the following arguments:

- **SPName** - The name of your stored procedure.
- **DataStruct** - An optional argument which is a structure of data that should be passed into the SP's parameters. This is optional as often an SP will not have any parameters.
- **DSN** - The datasource to be used when calling the SP. This is also optional. When initializing the component you can pass it a datasource name, which will be used as the default datasource for all SP calls. This means that you can use it to call SPs from more than one datasource.

When you call an SP, the first thing it does is check to see if it already has the parameters in its cache. If it does not, it calls a helper method that introspects the database and processes the information it retrieves about your SP into a format to be used when calling the SP. Then it builds a cfstoredproc call for you, including a cfprocparam tag for each of the SP's parameters. It attempts to find a suitable value for each parameter in the DataStruct that you passed in. It works by convention, assuming that the name of the parameter in the SP will be the same as the name of the key in the struct. It also includes some logic to deal with un-checked checkboxes and to deal with NULLs, although that is limited at the moment. After calling the SP with the cfstoredproc tag which is built on the fly, it returns a single result set.

I have a customized version of it that has some special case processing for parameters that are common to many of my SPs, but I haven't included that in the OS version as they are specific to my business requirements. I will be looking to build extensibility into the product in the future to allow others to do this with ease.

I built this for use with **Fusebox** (although it can be used with any framework, or no framework at all), so I always passed in the attributes scope, which contains all Form and URL variables. This allowed the whole process to be pretty much automatic.

The component has a few limitations, many of which may be addressed in future releases:

- It will only work with SPs that return either one or zero result sets.
- It currently only works with SQL Server.
- It supports all SQL Server datatypes, but I have been unable to successfully test it with the uniqueidentifier datatype.
- Its NULL support is somewhat limited.
- It does not support default values (defined in T-SQL) for SP parameters. This is due to the fact that ColdFusion does not allow you to pass parameter names via cfprocparam, therefore it must generate one cfprocparam tag for each and every SP parameter (even if you are not supplying a value for that param). This was not an issue with a previous version of the component, that used cfquery instead of cfstoredproc, but I ran into a couple of issues with that version. If there's any interest I could resurrect that version and include it in the component as an option.

And that's about it. If anybody decides to try it out I'd be very happy to hear what you think of it. Also, if there's any interest I could put together another blog post with more details about how to implement it in a project with Coldspring (although it's incredibly straightforward). I also have a few ideas about trying to get it working with Transfer, but at this point they are all hair-brained.

Update

The code is now available for download on **RIAForge**.