

# Easy Access to Primary Keys in Transfer Objects

Posted At : August 22, 2008 7:43 AM | Posted By : Bob Silverberg

Related Categories: ColdFusion, Transfer

I just added a couple of new methods to my [AbstractTransferDecorator](#) to make some of my ColdFusion development tasks easier. I found that when trying to write abstract code I needed to know the name of the primary key column of my object, and also sometimes needed to know the value of that column for the current object. So I wrote two new methods to address those needs:

- `__getPKColumn()` - Returns the name of the id column for the current object.
- `__getPKValue()` - Returns the value of that column for the current object.

I chose to prefix the method names with a double underscore so I wouldn't end up overwriting the getters if any of my objects ever have those names as properties. Although I'm fairly certain that would never happen it seems like a good idea nonetheless. Here's the code:

```
<cffunction name="__getPKColumn" access="public" output="false" returntype="any" hint="Returns the name of the PK column">

    <cfreturn getTransfer()

        .getTransferMetaData(getClassName())

        .getPrimaryKey()

        .getColumn() />

</cffunction>
```

In this single line of code I'm asking Transfer for the Metadata for the current class, then grabbing the PrimaryKey object from that and finally getting the Column name from the PrimaryKey object. Not much code, but I'd still rather have it in one place than all over my app. I believe that some people address this need by naming their id column in transfer.xml as "id", but I prefer to keep my id/property names in synch between transfer.xml and my database.

```
<cffunction name="__getPKValue" access="public" output="false" returntype="any" hint="Returns the value of the PK column">

    <cfset var theMethod = this["get#__getPKColumn()#"] />

    <cfreturn theMethod() />

</cffunction>
```

Here I'm making use of the `__getPKColumn()` method to determine the name of the primary key column for this object, and I'm using a technique to invoke a dynamic method name on the object, building the method name by prepending "get" to my PKColumn name. This technique is an alternative to using `CFINVOKE`, which used to be my standard way of accomplishing this feat. I've seen this technique documented a few times, the most recent of which was in [a comment on Ben Nadel's blog](#) by [Elliott Sprehn](#). Thanks guys for the neat trick.