

# Command Object and IColumnsRowset

---

*Source:* <http://www.tech-archive.net/Archive/Data/microsoft.public.data.oledb/2007-04/msg00022.html>

---

- *From:* [mohan.gulati@xxxxxxxxxx](mailto:mohan.gulati@xxxxxxxxxx)
  - *Date:* 30 Apr 2007 06:36:55 -0700
- 

Hi,

I am trying to use ADO to retrieve information about a SQL Statement that I am executing. In my searches on MSDN I came across the Command Object and saw that it had dynamic properties to retrieve information like the "BASE Table name" or "BASE column name". This information would be very useful to me. But honestly I could not figure out how to get this to work at all!

Here is some code that I wrote in C++.

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
MYADO::PropertyPtr pProperty =  
pCommandPtr->GetProperties()-  
  
    GetItem( CComVariant(_T("IColumnsRowset")));  
  
VARIANT nVariant = pProperty->GetValue();  
if ( ( nVariant.vt == VT_BOOL ) && ( V_BOOL( &nVariant ) ==  
VARIANT_TRUE ) )  
{  
    CComQIPtr< IColumnsRowset > pMyColumnInfo =  
    pCommandPtr.GetInterfacePtr();  
    if ( pMyColumnInfo != NULL )  
    {  
        // Get the rowset information  
    }  
  
}
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

But pMyColumnInfo is always NULL?

I really need an example that would show me how to from the Command Object extract detailed information about each column in the

## Command Object and IColumnsRowset

statement.

For me this would be similar to the ODBC methods SQLGetColumnAttr.

Regards,  
Mohan

.