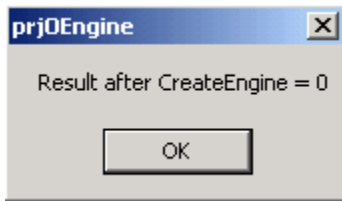


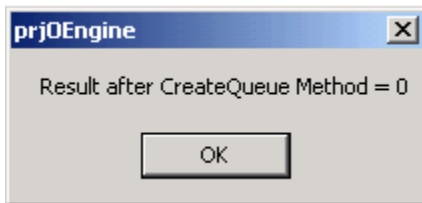
# Running the Visual Basic Program

Time to run the program. Save the code, then press **F5** (or **Run-Start** from the menu). Click on the button.

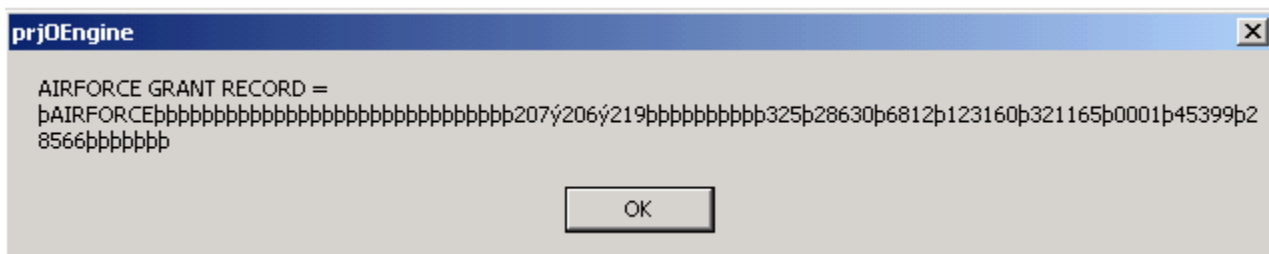
Four messages should display. [CreateEngine\(\)](#) needs to use Port 777. If the [CreateEngine\(\)](#) function succeeds, the return code will be zero, and the following message will display.



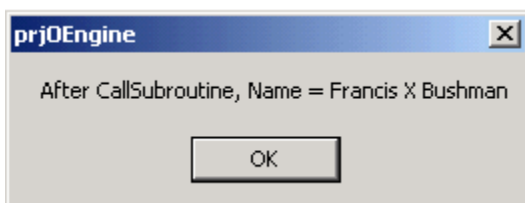
The next task is to create a queue. The queue will reference the database called **LAB** with the username of **LAB**. If successful, the following message will display:



The **VB\_Read\_Row()** function then is called using the [CallFunction\(\)](#) method of the Queue object. In this situation, a row called **AIRFORCE** from the table called **GRANTS** is read. Substitute your tablename and row name. Note that the returned value is passed in **ReturnValue**, the first argument to [CallFunction\(\)](#).



The **OI\_Change\_Name** subroutine is then called using the [CallSubroutine](#) method of the Queue object. Note that the variable myName is passed as the argument to [CallSubroutine](#). Passed as **Ira Krakow**, the OpenInsight subroutine changes it to **Francis X Bushman**.



The balance of the program is housekeeping, destroying the objects created to communicate with OpenInsight. This is needed to conserve memory. Unused objects should be destroyed when not needed.

So voila! Client-server programming with Visual Basic as the client and OpenInsight as the server!