

# Communicating Between Visual Basic and OpenInsight

A number of OpenInsight customers would like to run in a client-server mode, where a GUI development tool such as Visual Basic, Delphi, or PowerBuilder acts as the front end presentation server, and OpenInsight acts as the back end database server. This is a common setup with back ends such as Microsoft SQL Server or Oracle.

This article provides a Visual Basic 6.0 program that calls OpenInsight stored procedures (functions or subroutines), and receives results from OpenInsight. Although this example is specific to Visual Basic, it can be generally applied to other Windows-based GUI front ends.

This example was written using the Windows XP Professional operating system, but should work on any Windows computer (such as a Windows 2000 or Windows NT class machine) that supports TCP/IP.

The procedure consists of the following steps:

1. [Registering XREV.DLL](#). **XREV.DLL**, found in the **OI32bit** folder in the OpenInsight installation directory, is the Revelation Software supplied Dynamic Link Library containing the object definitions and methods required for communicating with OpenEngine.
2. [Running OpenInsight as a server](#). The server will run using the TCP/IP protocol on an unused port. For this example, we will use Port **777**.
3. [Writing the OpenInsight Stored Procedures](#). This example will call an OpenInsight function that reads a row from an OpenInsight table, and execute an OpenInsight procedure that returns a status code.
4. [Creating the Visual Basic project](#). The project must reference the **Revelation Software Type Library** included in XREV.DLL. Including the project reference allows a Visual Basic program to call OpenEngine.
5. [Writing the Visual Basic program](#). The program will create an instance of OpenEngine, create a Queue object that references an OpenEngine Queue, and then use the [CallFunction\(\)](#) and [CallSubroutine\(\)](#) functions to invoke the OpenInsight stored procedures.
6. [Running the Visual Basic Program](#). Using the Visual Basic MsgBox() function the program will examine the results of the OpenEngine calls.