



## Using IBConsole to Execute MultiLoggerDB Scripts Database Application Note #2

### Overview

The Interbase database server utilized by MultiLoggerDB is a powerful ANSI-92 SQL database system. There are a number of stored procedures and triggers, essentially database programming, stored in the database which supports the various functionality provided by the MultiLoggerDB system. This database programming is routinely updated, revised and enhanced to support new functionality or to extend existing functionality.

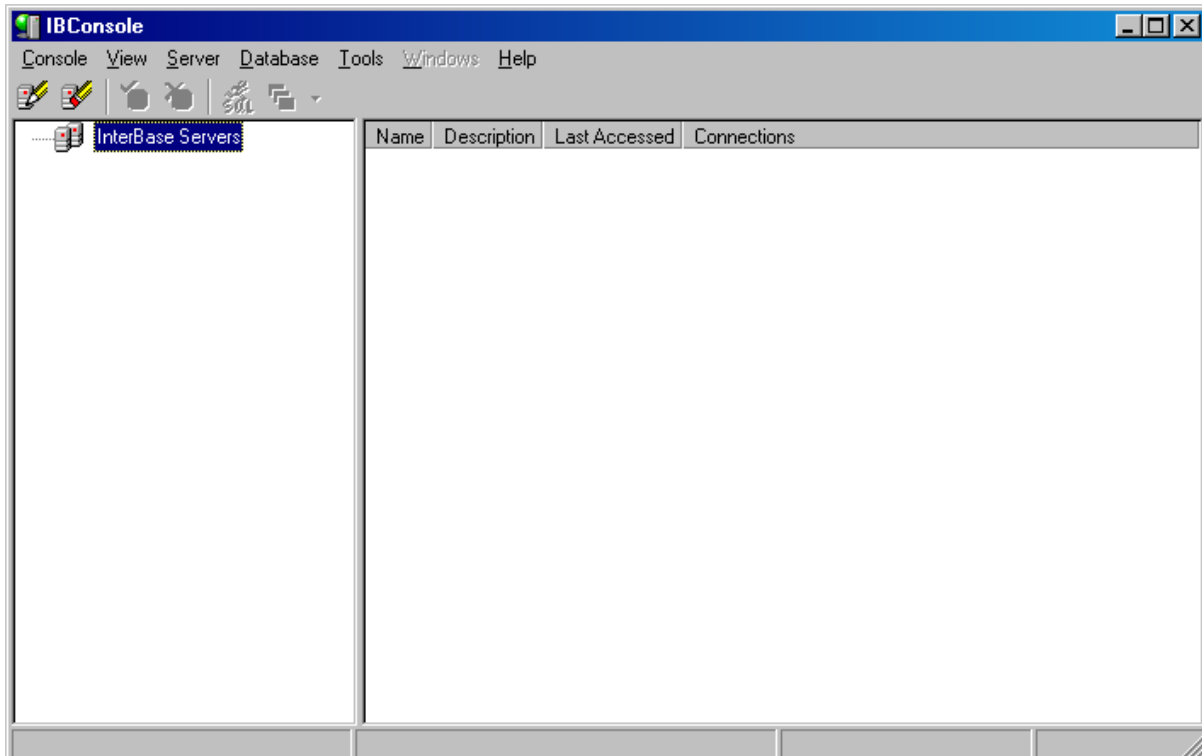
This Database Application Note will provide details regarding the configuration of IBConsole, a utility program included with the MultiLoggerDB system, and its use to execute scripts provided by Canary Systems to update the database programming.

IBConsole is installed using the Interbase Server install option on the MultiLoggerDB CD. Consult the MultiLoggerDB User's Guide for more details on the installation of Interbase Server.

### Configuration

The IBConsole.exe shortcut will normally be found in the Interbase 6.0 Open Edition group in your Programs menu. Double-click to launch IBConsole.

When IBConsole is first started an empty configuration will show.



The first step is to configure the server connections, local or remote connections to the server are supported. Double-click on the **Interbase Servers** item in the tree shown on the left side of the form. This will display the **Register Server and Connect** form.

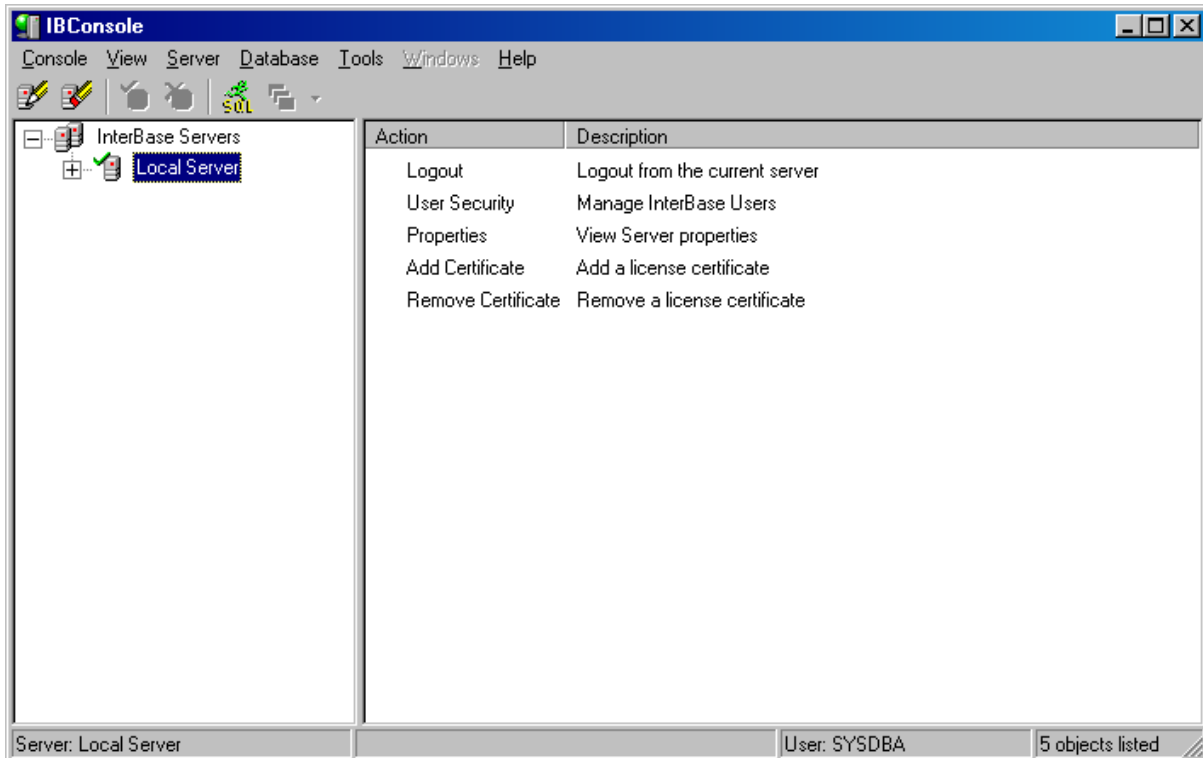
Usually **IBConsole** is operated on the machine hosting the Interbase database so the default selection of **Local Server** is typically used.

If the database is hosted on a remote machine then select **Remote Server**, enter the **Server Name** in DNS or IP format and select the **Network Protocol** (usually TCP/IP).

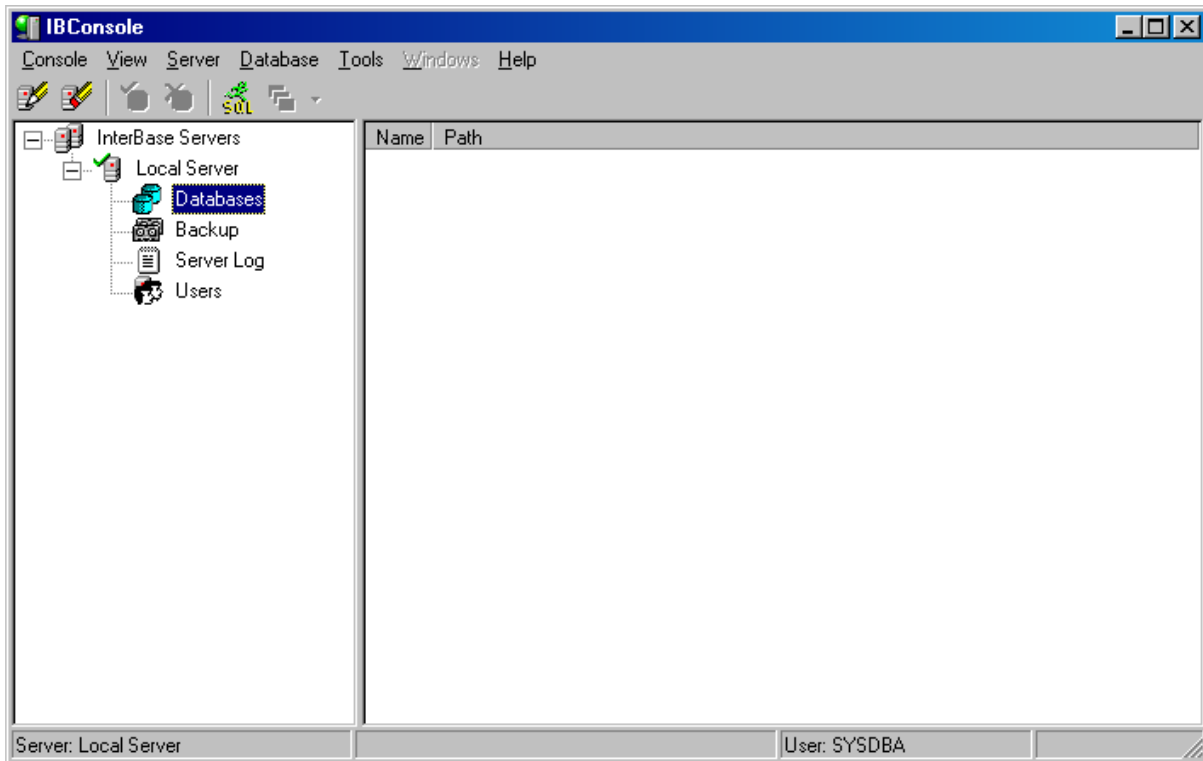
Note: One exception is where numerous sessions either with IBConsole or the MultiLoggerDB components, i.e. the Database Console and/or Insite will be hosted simultaneously on the machine which also hosts the database. In this case it is preferable to use **Remote Server**, a **Server Name** of **localhost**, an appropriate **Network Protocol** (usually TCP/IP) and an **Alias Name** to identify the server in the configuration.

Enter the **User Name** (default administrator User Name is shown, SYSDBA) and password. Press **OK** to connect to the server using the configuration options shown.

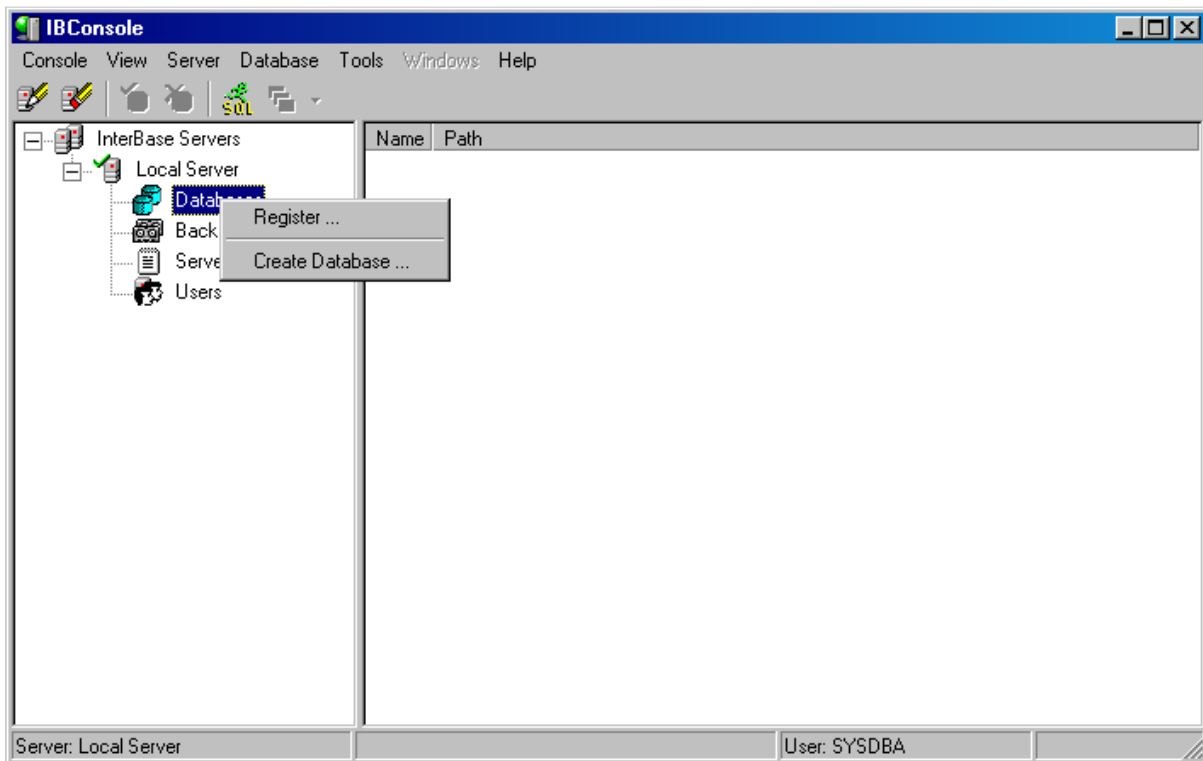
If the connection was successful the configuration in IBConsole will update.



Double-click on **Local Server** to expand the connection options, shown below.



Right-click on **Databases** to display the database registration menu.



Select **Register** to display the **Register Database and Connect** form.

The actual database file must be specified using this form. Press the file open button to the right of the Database File edit to browse and select the GDB file. When MultiLoggerDB is installed an empty GDB file named MULTILOGGER.GDB is copied to your **\Program Files\MultiLogger** directory, normally this file is copied or renamed to match your project, it is NOT recommended to use the MULTILOGGER.GDB as is.

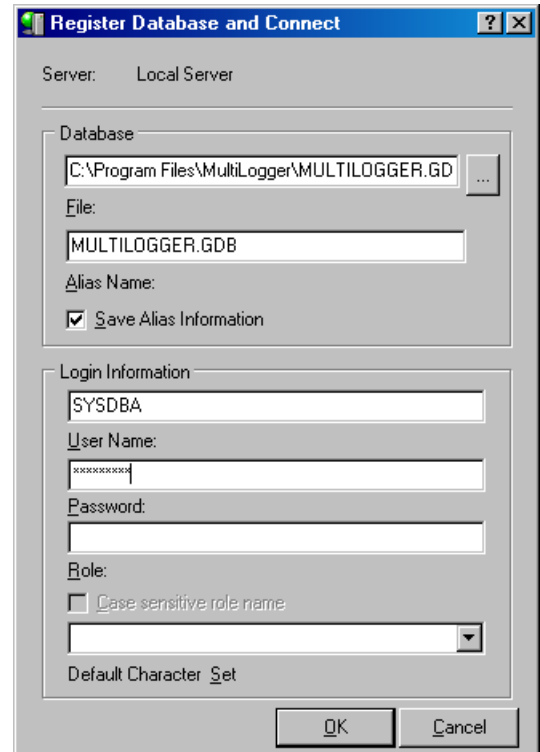
NOTE: When using Remote server connections the file browse button will be disabled, you must manually type in the path and name of the MultiLoggerDB GDB file.

Make sure **Save Alias Information** is checked.

Enter the **User Name** for the database server and the **Password** in the fields on the form.

Other options can be left at their defaults.

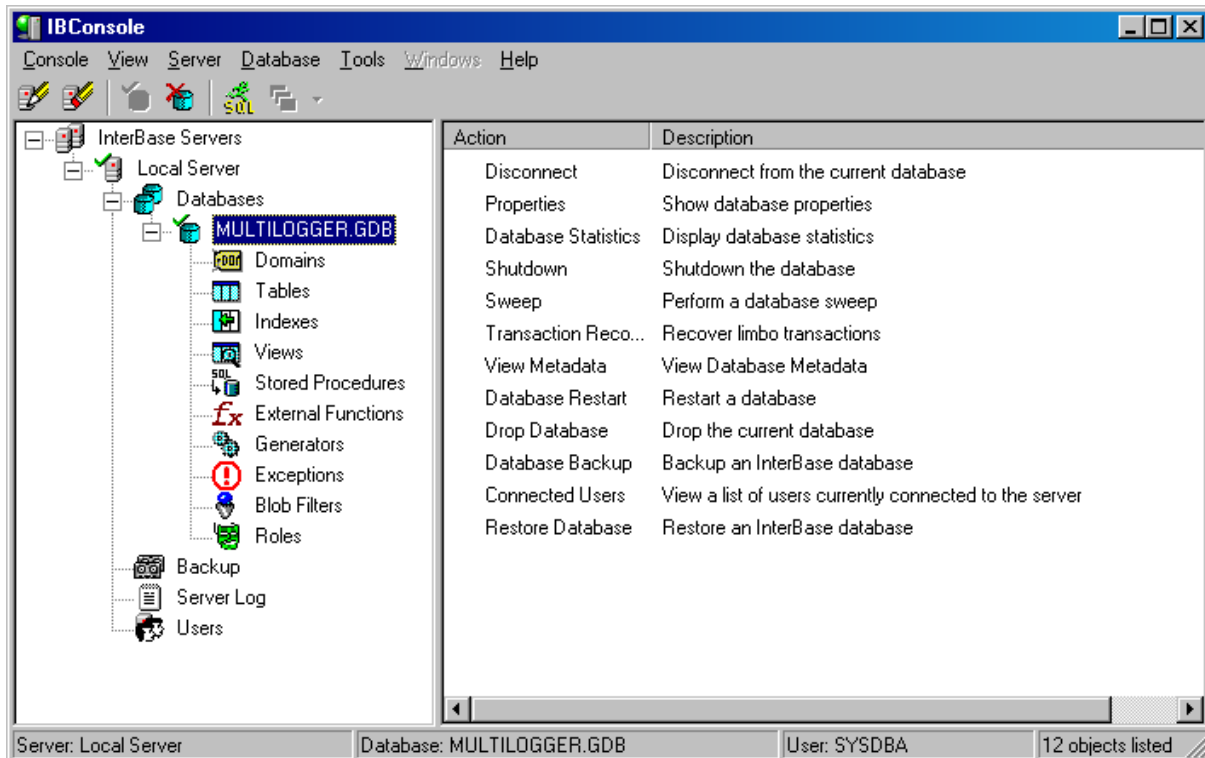
Click OK to connect to the specified database.



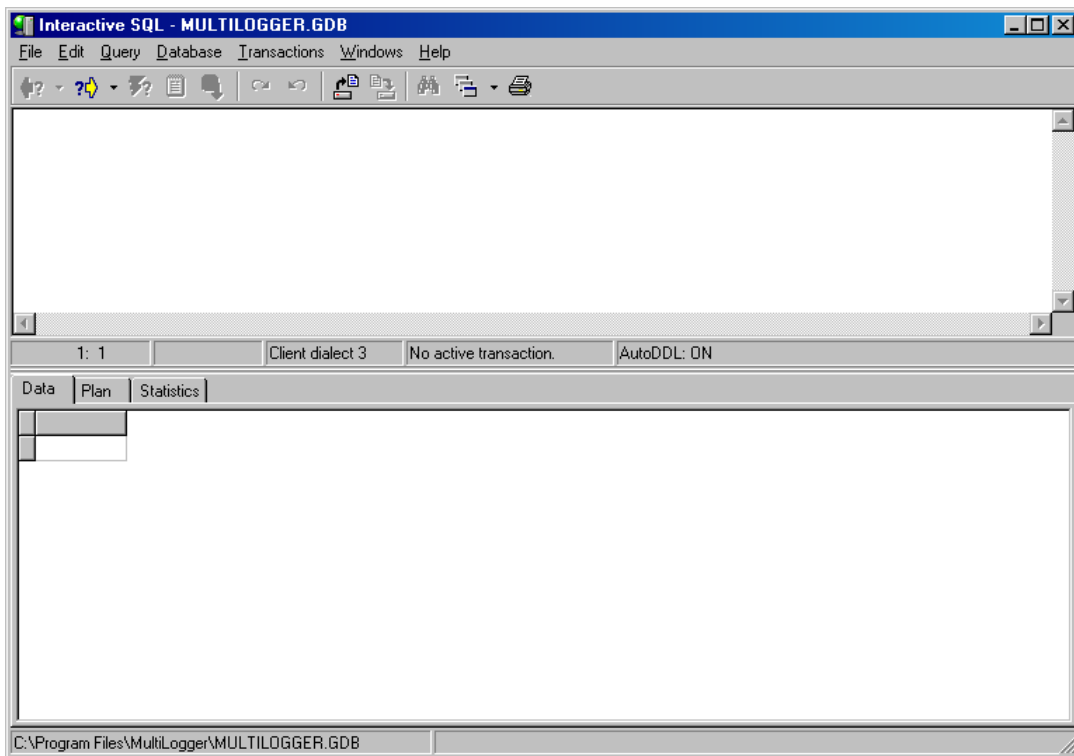
The dialog box titled "Register Database and Connect" has a blue title bar with a help icon and a close button. It contains the following fields and controls:

- Server: Local Server
- Database: C:\Program Files\MultiLogger\MULTILOGGER.GD (with a browse button)
- File: MULTILOGGER.GDB
- Alias Name: (empty)
- Save Alias Information
- Login Information:
  - SYSDBA
  - User Name: (password masked with asterisks)
  - Password: (empty)
  - Role: (empty)
  - Case sensitive role name
  - Default Character Set: (dropdown menu)
- Buttons: OK, Cancel

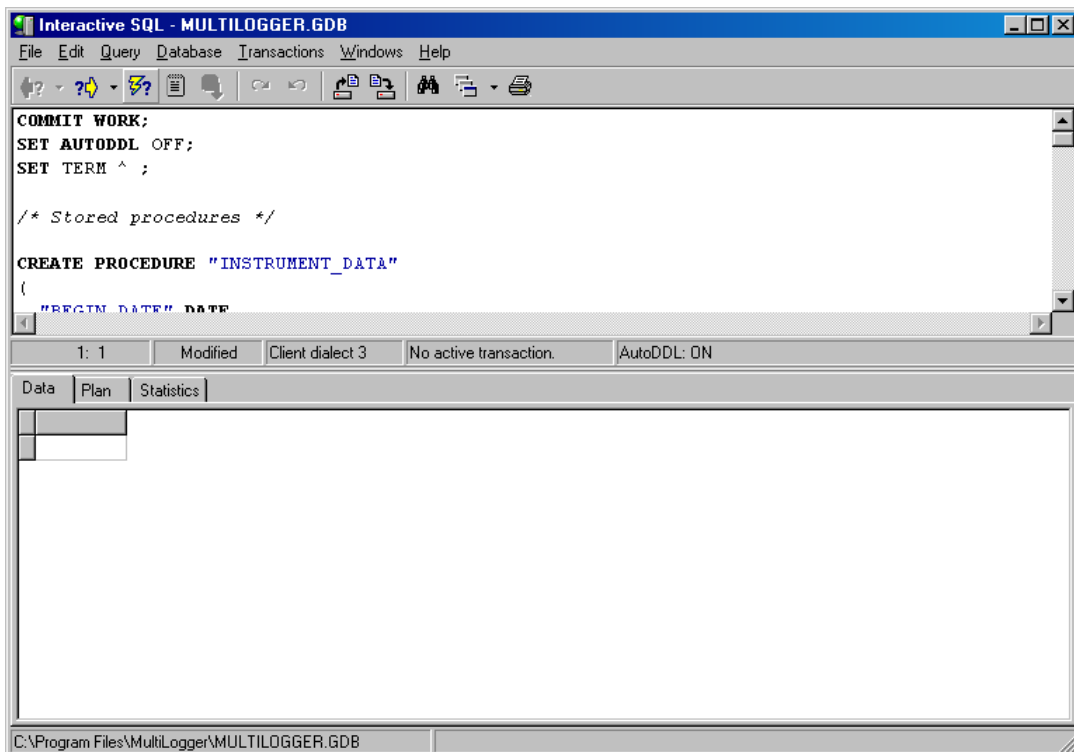
If the connection was successful then the configuration in IBConsole will update.



Click the SQL button on the toolbar or select the menu item **Tools | Interactive SQL** to display the Interactive SQL window, shown below. The top portion is the SQL statement window, SQL programming can be directly entered in this window, the lower window shows any results generated by the script.

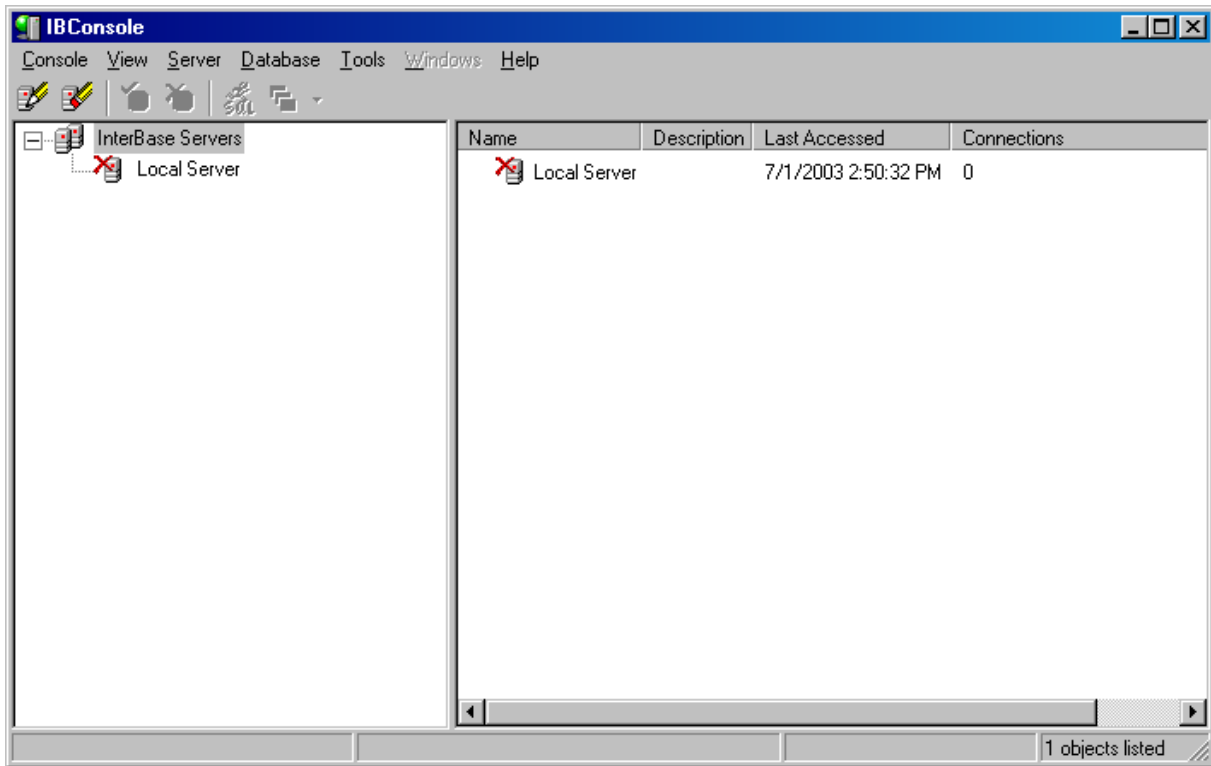


Select the menu item **Query | Load Script** to display a file dialog to browse and select the script. The script will then load in the upper window.

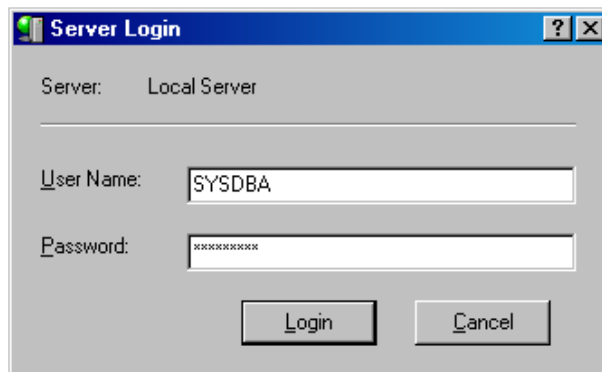


Click on the SQL execute button or select the menu item **Query | Execute** to execute the SQL programming. If the script executes correctly it will be erased from the window. Any errors will cause the script to halt. **Note any errors shown after executing the script.** Close the Interactive SQL window by clicking on the upper right corner or selecting the menu item **File | Close**, you will return to the IBConsole configuration. Script execution is complete!

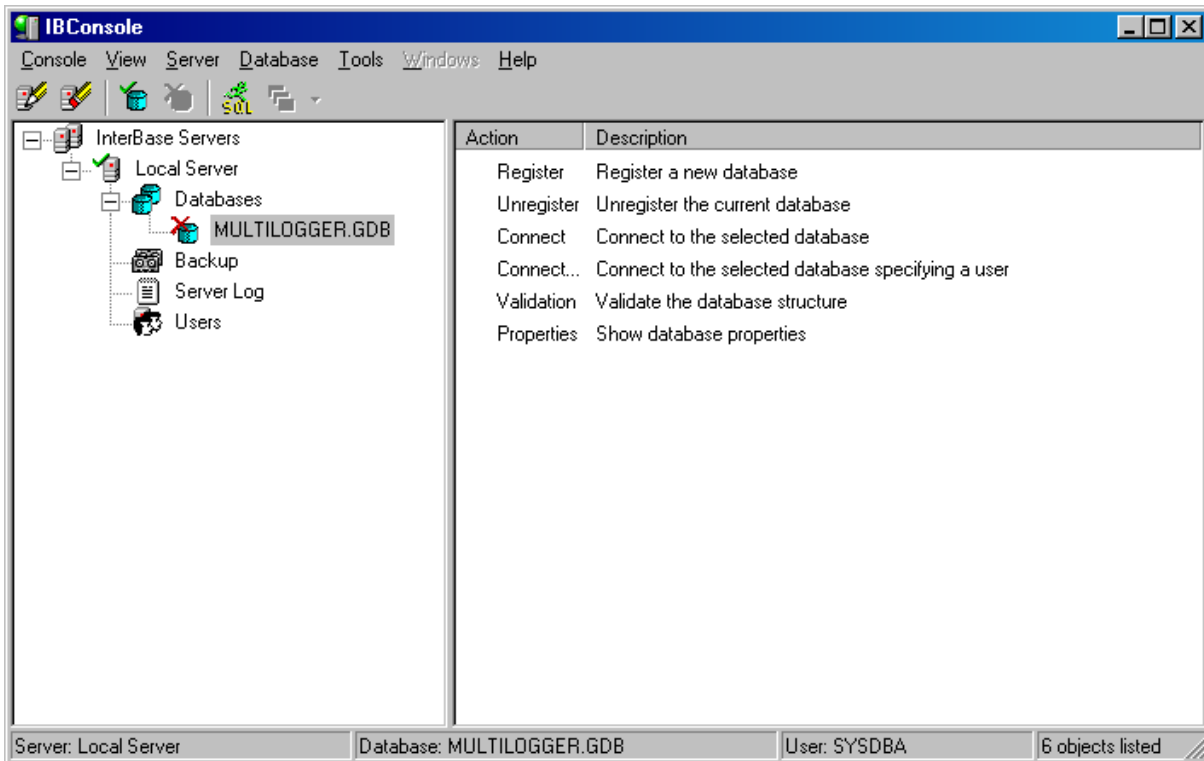
To exit IBConsole simply click on the upper right corner of the form or select the menu item **Console | Exit**. Note: At the next startup of IBConsole you will be logged out of the server and disconnected from the database, as shown below.



To login to the server simply double-click **Local Server** (or the Alias for Remote Servers), a password prompt will display. Enter your password to log in to the server.



Double-click the database you wish to connect to.



The configuration will update with the database options.

