

## **ASNA DataGate Studio**

### **Working with Database Names**

Working with Database Names .....	1
Creating Database Names .....	2
Modifying Database Names .....	5
Public and Private Database Names .....	8
Renaming Database Names .....	9
Advanced Connection Properties.....	9

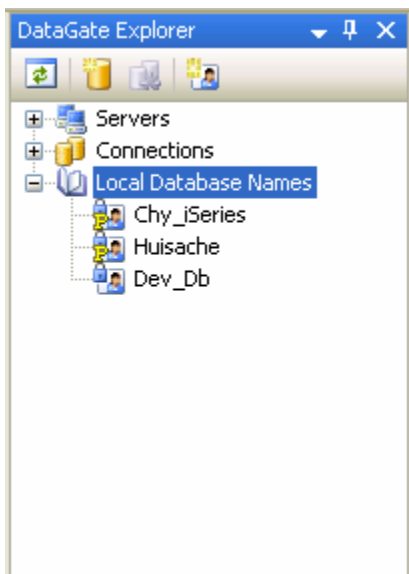
## Working with Database Names

The DataGate Explorer tool allows you to add, change, or delete a database name. A database name is the **identifier** you use in your applications to refer to a particular database. This name encapsulates not only the server and label of the database, but also the access method, and the account to use on the server.

A database must be "**Named**" in order to be accessed and used in applications. Naming a database creates a logical name on your local computer that points to a database on a local or remote server. You can add multiple logical database names that point to the same database.

Database names can also be used in DataGate Explorer to create database connections. Database connections allow you to perform database management tasks and to design database and print files in DataGate projects.

DataGate Explorer displays all database names registered on the local computer under the Local Database Names node, as shown below.



You have two ways of creating database names with DataGate Studio.

- **Add a New Database Name** Command. This DataGate Explorer command displays a dialog for entering database name parameters.
- **Database Wizard** Command. This DataGate menu command starts a wizard-style applet for creating database names, and optionally, a new server database and label.

Once a database name has been created, you can easily modify its identifier and properties in DataGate Explorer.

- Use the Edit Properties command on an existing database name to modify the connection properties of the database name.
- Use the Rename command on an existing database name to change its identifier.

---

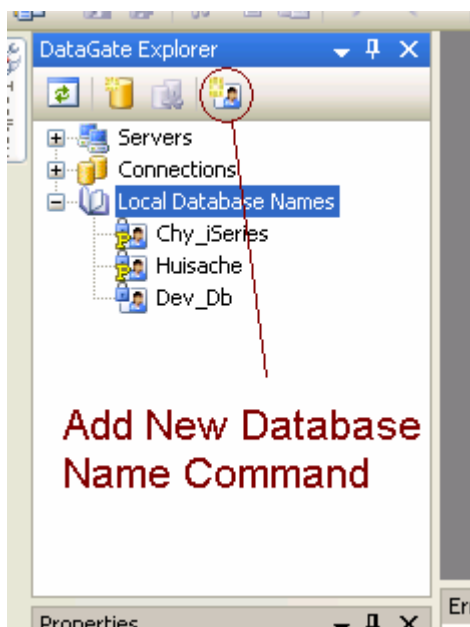
## Creating Database Names

Before you can access a DataGate database in an AVR or Monarch application, you must create a database name to refer to the database programmatically. You can create database names in DataGate Studio with the Add New Database Name command, as explained below.

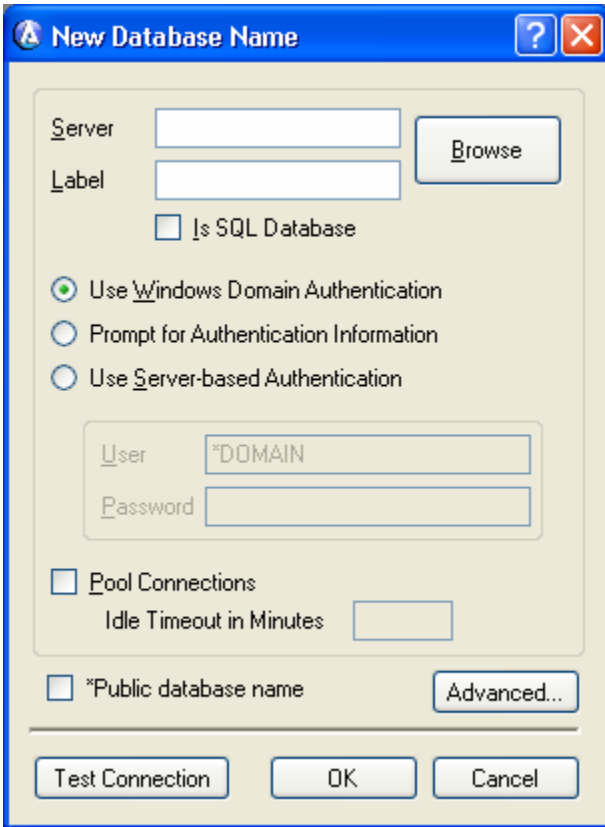
You can also easily create database names by selecting **Database Wizard** from the DataGate menu.

### To Create a New Database Name

1. Select **DataGate Explorer** from the DataGate menu. The DataGate Explorer tool window will display.



2. Click the Add New Database Name button on the toolbar at the top of DataGate Explorer window (or alternately, right-click the Local Database Names node and select Add New Database Name on the context menu). The New Database Name dialog is displayed, as shown below.



3. Select primary parameters for the database name, as explained below.

**Server:**

Enter the server that contains the actual database in which the database name entered above points to. Select the Browse button to the right to take you to a dialog box allowing you to Select a Database server and database label. Otherwise, enter any valid TCP/IP hostname or IP address that refers to the networked database server host.

**Label:**

Enter the label of the actual database in which the database name points to. Select the Browse button to the right to take you to a dialog box allowing you to Select a Database server and database label.

**Authentication:**

Select one of the three radio buttons to choose the authentication method for connecting to the DataGate server.

**Use Windows Domain Authentication.** Use the current Windows Domain user credentials to authenticate access to the server. Note that this method only works for Windows-based database servers, and only when both the client and server machines are members of the Windows Domain.

**Prompt for Authentication Information.** Upon each connection attempt, the DataGate client will raise a user/password dialog for the user to enter credentials for access to the server.

**Use Server-based Authentication.** In the provided text boxes, enter the user and password which should be used for every connection attempt to authenticate access to the server. Note in this case that DataGate saves encrypted user and password information on the local machine.

**User:**

Enter the name of a server user authorized to access the database. User name is enabled for editing only if Use Server-based Authentication is selected. Note that DataGate saves encrypted user and password information on the local machine when Use Server-based Authentication is selected.

**Password:**

Enter the password for the user name given above. Password is enabled for editing only if Use Server-based Authentication is selected. Note that DataGate saves encrypted user and password information on the local machine when Use Server-based Authentication is selected.

**Pool Connections:**

Select this option to enable database connection pooling. When database connection pooling is enabled, the communication to the server and to the server resources are kept in a special 'pool'. The next time the application requires database services, one of the available connections in the 'pool' is assigned to it. Thus, enabling Pool Connections results in faster database connections.

If connection pooling is not enabled, when a database and all its associated files are closed, the connection and server resources are ended (i.e., in the case of DataGate, the DataGate Job ends). When the same application needs the database services again, a new connection and corresponding job has to be established. This overhead wastes resources and slows down the application.

**Idle Timeout in Minutes:**

The **Idle Timeout in Minutes** field will become enabled when **Pool Connections** option is selected. You must specify an amount of time (in minutes) in which a connection will remain idle in the pool until it is closed and removed from the pool. The default value is 1 minute. Enter a time from 1 – 255 minutes.

You want to select a value that is long enough that permits the application to quickly open again, but not so long that many jobs are remaining idle.

**\*Public database name:**

Select this option to make this database name accessible to all users of the local machine. See "Public and Private Database Names" for more information.

**Advanced Button:**

Press this button to display the Connection Properties dialog window (Advanced Connection Properties). This dialog allows you direct access to all connection properties.

**Test Connection Button:**

Press this to create a test connection to the database server after you have entered all connection details associated with the database name. At the end of the test, a message box will display the results.

**OK Button:**

Select **OK** to Add a database name.

**Cancel Button:**

Select **Cancel** to abort the New Database Name dialog without adding the database name.

2. After clicking OK, the dialog will close and the identifier of the new database name will be selected in edit-mode, under Local Database Names in DataGate Explorer. Change the default, "New Database Name", to the identifier you wish to refer to the database name with.

Enter a "**Name**" to call the database on your computer. This is the **identifier** you will use to refer to the database in your applications.

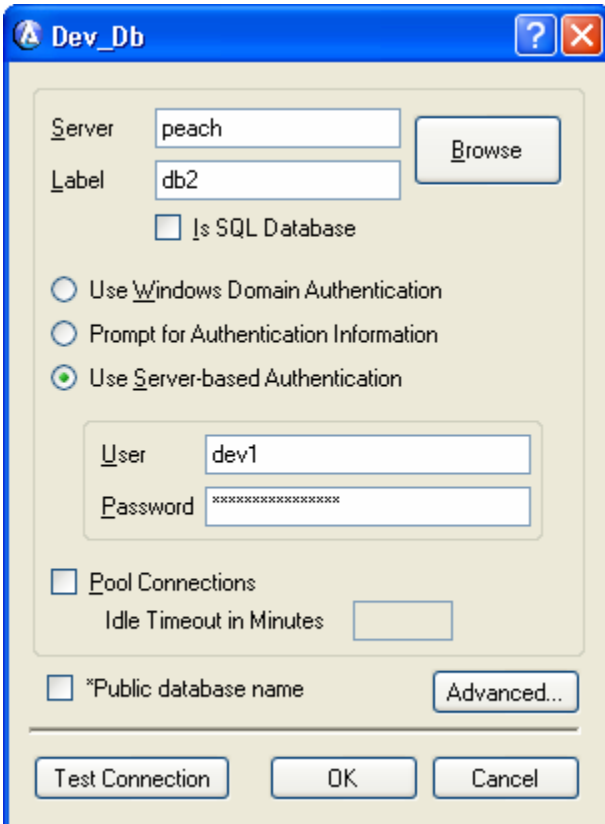
---

## Modifying Database Names

After creating a database name in DataGate Studio with the Add New Database Name command, you may want to modify its properties. This is accomplished with the Edit Properties command on the database name node's context menu. Note: to change the identifier for the database name, use the Rename command on the database name node in DataGate Explorer.

### To Modify an Existing Database Name

- 1 Select **DataGate Explorer** from the DataGate menu. The DataGate Explorer tool window will display.
- 2 Select the database name node under Local Database Names. Right-click the node to display the context menu, and select Edit Properties. The Database Name editing dialog will display, as shown below.



- 3 Modify the primary parameters for the database name, as explained below.

**Server:**

Enter the server that contains the actual database in which the database name entered above points to. Select the Browse button to the right to take you to a dialog box allowing you to **Select a Database** server and database label. Otherwise, enter any valid TCP/IP hostname or IP address that refers to the networked database server host.

**Label:**

Enter the label of the actual database in which the database name points to. Select the Browse button to the right to take you to a dialog box allowing you to Select a Database server and database label.

**Authentication:**

Select one of the three radio buttons to choose the authentication method for connecting to the DataGate server.

- **Use Windows Domain Authentication.** Use the current Windows Domain user credentials to authenticate access to the server. Note that this method only works for Windows-based database servers, and only when both the client and server machines are members of the Windows Domain.

- **Prompt for Authentication Information.** Upon each connection attempt, the DataGate client will raise a user/password dialog for the user to enter credentials for access to the server.
- **Use Server-based Authentication.** In the provided text boxes, enter the user and password which should be used for every connection attempt to authenticate access to the server. Note in this case that DataGate saves encrypted user and password information on the local machine.

**User:**

Enter the name of a server user authorized to access the database. User name is enabled for editing only if Use Server-based Authentication is selected. Note that DataGate saves encrypted user and password information on the local machine when Use Server-based Authentication is selected.

**Password:**

Enter the password for the user name given above. Password is enabled for editing only if Use Server-based Authentication is selected. Note that DataGate saves encrypted user and password information on the local machine when Use Server-based Authentication is selected.

**Pool Connections:**

Select this option to enable database connection pooling. When database connection pooling is enabled, the communication to the server and to the server resources are kept in a special 'pool'. The next time the application requires database services, one of the available connections in the 'pool' is assigned to it. Thus, enabling Pool Connections results in faster database connections.

If connection pooling is not enabled, when a database and all its associated files are closed, the connection and server resources are ended (i.e., in the case of DataGate, the DataGate Job ends). When the same application needs the database services again, a new connection and corresponding job has to be established. This overhead wastes resources and slows down the application.

**Idle Timeout in Minutes:**

The **Idle Timeout in Minutes** field will become enabled when **Pool Connections** option is selected. You must specify an amount of time (in minutes) in which a connection will remain idle in the pool until it is closed and removed from the pool. The default value is 1 minute. Enter a time from 1 – 255 minutes.

You want to select a value that is long enough that permits the application to quickly open again, but not so long that many jobs are remaining idle.

**\*Public database name:**

Select this option to make this database name accessible to all users of the local machine. See "Public and Private Database Names" for more information.

**Advanced Button:**

Press this button to display the Connection Properties dialog window (Advanced Connection Properties). This dialog allows you direct access to all connection properties.

**Test Connection Button:**

Press this to create a test connection to the database server after you have entered all connection details associated with the database name. At the end of the test, a message box will display the results.

**OK Button:**

Select **OK** to save your modifications to the database name.

**Cancel Button:**

Select **Cancel** to abort the Database Name dialog without modifying the database name.

3. Click OK to save the modifications, or Cancel to close the dialog without saving changes.

---

## Public and Private Database Names

The currently-registered database names of the local computer are shown under Local Database Names in DataGate Explorer.

The database name information is maintained in the Local Settings for the Current\_User. If another user logs on to the same Windows computer, the Database Name would be unknown to the new user (it is "**private**" to the first user). This can be circumvented by defining the database name as a "**public**" database.

You make a database name public by selecting the "\*Public database name" checkbox in the database name dialog. A public database name can be referred to explicitly by prepending the string "**\*Public/**", as in "\*public/my database name". Note however that a public database name is known implicitly to your applications as "my database name" without regard to the current user.

Since a public database name has no connection to the current user, a public database name is maintained in the Local Settings of the "**All Users**" area of the local computer.

Public database names are often used for web applications, or in other middle-tier applications where the user may be anonymous, have restricted capabilities, or is simply unknown to the application.

The database names that have been defined as public will display in DataGate Explorer as nodes with a special icon as shown below.

- Public database name icon
- Private database name icon

---

## Renaming Database Names

To change the identifier of an existing database name, use the DataGate Studio Rename Object command on the DataGate menu, or the Rename command on the context menu of the database name node in DataGate Explorer. You can also modify the connection properties of a database name, with the Edit Properties command.

### To Rename a Database Name Identifier

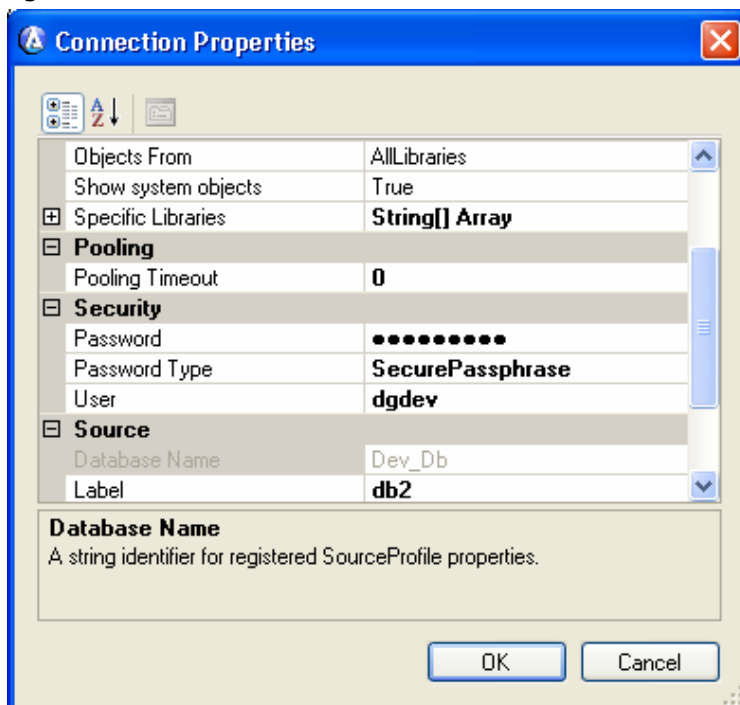
- 1 Select **DataGate Explorer** from the DataGate menu. The DataGate Explorer tool window will display.
- 2 Expand Local Database Names and select the database name node under it to be renamed.
- 3 Select the Rename Object command on the DataGate menu. Alternatively, right-click the database name node to display its context menu, and select Rename.
- 4 The text of the database name node (the identifier) will become editable. Change the text to the desired database name identifier.

Note that a third method of renaming the database name is to select the node, pause, and then click on the name. The text of the database name node will enter edit mode.

---

## Advanced Connection Properties

Other DataGate connection properties can be associated with a connection or database name by clicking the Advanced button on the New Database Name, Database Connection, and Edit Database Name dialogs. This button displays the Connection Properties dialog window, shown below. This dialog shows ALL properties associated with the database name, including those which can be set on the primary properties dialog.



The dialog is a Windows standard property grid, displaying the property name in the left column, and its value in the right column. Selecting a property name or value cell will display a description of the property in the bottom pane of the window. Buttons above the grid allow you to sort the properties in various ways. The property names and their values are as follows.

**Database Name:**

The string **identifier** for a database name is used to reference the database name in a program. This identifier is shown in the DataGate Explorer tree under Local Database Names. In the Advanced Connection Properties grid, the value of this property is read-only. To change the database name identifier, use the Rename command on the database name node after the database name dialog has been closed. If no database name is associated with the connection properties, this value **will be an empty string.**

**File Types:**

This property allows you to filter which objects are shown as library contents displayed by DataGate Explorer. You may filter these types:

- All – Show all types of objects not otherwise filtered (this is the default value).
- All Logical – Show only logical files (including simple logical, multiformat, join, and SQL logical), unless otherwise filtered.
- Join – Show only join logical files, unless otherwise filtered.
- Physical – Show only physical files, unless otherwise filtered.
- Print – Show only print files, unless otherwise filtered.
- Simple Logical – Show only simple logical files, unless otherwise filtered.
- Data Areas – Show only Data Areas (DataGate Studio 9.1 and later only), unless otherwise filtered.

Note that this property only affects the DataGate Explorer display, not general access to the database. Applications using the connection can access any object type that the user has authority to.

**Initial Library List:**

With this property, you may optionally specify a list of library names that should be used to set an initial library list on the server database, upon each successful connection. Some DataGate server platforms associate a library list with a user profile. Note that any library list specified here will override the user profile's library list. The default value of this property is an empty list.

**Label:**

The value of this property, the server database label, is identical to the value entered in the Label field of the primary connection dialog. Enter the label of the actual database in which the database name points to. The primary connection dialog features a Browse button which displays a dialog allowing you to Select a Database server and label. There is no default value for Label.

**Objects From:**

This property allows you to filter which libraries are displayed by DataGate Explorer. This is useful when a server database consisting of many hundreds of libraries is slow to respond to a request to enumerate all libraries. You may filter libraries using these settings:

- All Libraries – Display all libraries not otherwise filtered (this is the default value).
- Library List – Display only libraries from the library list associated with the server job, unless otherwise filtered.
- User Library List – Display only libraries from the user portion of the library list associated with the server job, ignoring the system portion of the library list, unless otherwise filtered.
- Specific Libraries – Display only those libraries listed in the Specific Libraries property of the Advanced Connection Properties (explained below), unless otherwise filtered.

Note that this property only affects the DataGate Explorer display, not general access to the database. Applications using the connection can access any library that the user has authority to.

**Password:**

The value of this property, the server user's password, is identical to the value entered in the Password field of the primary connection dialog. The displayed value of the property is obfuscated for security. The password value entered is only used if the value of the User property is not "\*DOMAIN" or "\*PROMPT". The default value is an empty string.

**Password Type:**

This property supports legacy DataGate servers using passwords of length 31 characters or less. In particular, when accessing DataGate/400 servers where the iSeries host uses the pre-V5R1 10-character password authentication scheme, this property may need to be set to the "Legacy" value. The default value is "**SecurePassphrase**".

- Legacy – The password sent to the server must be 31 characters or less in length. If the iSeries server requires pre-V5R1 password authentication, you may need to specify this value to access a DataGate/400 server.
- SecurePassphrase – The password sent to the server is not limited to 31 characters. Use this value to take advantage of the improved password security available in V5R1 and later iSeries servers.

**Platform Attribute:**

This property is used internally by DataGate client to initialize the connection to the database. Normally, its value should not be changed. When the "**Is SQL Database**" checkbox is selected on the primary connection dialog, the value of this property will be set to the keyword "\*SQLOLEDB". In all other cases, the value of this property is either "\*DATALINK" or an empty string. The default value is "\*DATALINK".

**Pooling Timeout:**

The value of this property, the database connection pooling timeout, is identical to the value entered in the Idle Timeout in Minutes field of the primary connection dialog. Setting the value of this property to zero **disables** the database connection pooling feature. Note that if the Pool Connections checkbox is cleared on the primary connection dialog, the value of this property is zero. The default value is zero.

**Server Host:**

The value of this property, the server database host, is identical to the value entered in the Server field of the primary connection dialog. You may enter the server name that hosts the database. The primary connection dialog features a Browse button which displays a dialog allowing you to Select a Database server and label. The value of this property may be any valid TCP/IP hostname or IP address that refers to the networked database server host. There is no default value.

**Show System Objects:**

This property allows you to filter objects displayed by DataGate Explorer. This is particularly useful on iSeries systems, where system-related objects may be interspersed with user or application objects. You may filter objects using these settings:

- True – Display all objects not otherwise filtered (this is the default value).
- False – Display only objects not otherwise filtered that do not have the "system" attribute.

**Specific Libraries:**

The value of this property defines the libraries displayed by DataGate Explorer, but **only** if the value of Objects From is 'Specific Libraries'. You may enter all the libraries you wish to be shown, subject to other filters, here. The default value is an empty list. Note that this property only affects the DataGate Explorer display, not general access to the database. Applications using the connection can access any library that the user has authority to.

**TCP Port:**

This property defines the TCP communications port number to be used by DataGate in establishing a connection to the server. The default value of this property is 5042, the registered port number for DataGate servers. In certain cases, DataGate servers may be configured to listen for incoming connections on an alternate port. You may change the default to the configured port using this property.

**Text:**

This property may be used to enter any comments to be associated with the connection. The default value is an empty string.

**User:**

The value of this property, the server database user, is identical to the value entered in the User field of the primary connection dialog. There is no default value. You may enter the name of a server user authorized to access the database. Otherwise, the property value may be one of the following keywords.

- \*DOMAIN – Use the current Windows Domain user credentials to authenticate access to the server. Note that this method only works for Windows-based database servers, and only when both the client and server machines are members of the Windows Domain.
- \*PROMPT - Upon each connection attempt, the DataGate client will raise a user/password dialog for the user to enter credentials for access to the server.

Note that DataGate saves encrypted user and password information on the local machine when a non-keyword value is entered for the User property.