

# Sybase® IQ Quick Start

Document ID: DC01014-01-1530-02

Last revised: May 2011

<b>Topic</b>	<b>Page</b>
Introduction	2
Demo database	2
Table names	2
Creating and using an IQ demo database	3
Converting the IQ demo database to multiplex	9
Starting the IQ Agent	9
Converting to multiplex	11
Adding a multiplex server (manual method)	24
Creating the multiplex server definition	24
Synchronizing and starting the multiplex server (manual method)	26
Adding a temporary store file (manual method)	28
Loading data	29
Distributed query processing	31
Adding shared temporary storage	32
Adding shared temporary storage (manual method)	35
Increasing file size in iq_main	36
Running a query that gets distributed	37
Configuring an ODBC Datasource on Windows	40
ODBC C Programming Example	44

## Introduction

This document describes how to create and query an IQ demo database. You can create copies of the database in any directory that you choose, and re-create it at any time.

You also can use the demo database to create and use an IQ multiplex, which is multiple IQ servers that access the same IQ store by way of a shared disk cluster. For demonstration purposes, you can use a file system file that is shared by multiple IQ servers to simulate an IQ multiplex environment.

To query or update the database, you can use either Interactive SQL (dbisql) or Sybase Central™, both of which are provided with the IQ server installation.

Do not cut and paste the commands directly from this document; extra characters have been added for formatting. Either manually type the command, or cut and paste it to Notepad or vi, edit the command to remove any formatting characters, then copy and paste it to the command line.

## Demo database

The demo data supports an imaginary sporting goods company. The database, called iqdemo by default, is updated for Sybase® IQ 15.3.

## Table names

If you are upgrading from an earlier version, see the following table for a list of current table names in the iqdemo database. Use this table to update scripts or applications.

**Table 1: Demo database table names**

12.7 name	15.3 name	15.3 table owner
contact	Contacts	GROUPO
customers	Customers	GROUPO
department	Departments	GROUPO
empl	empl	DBA
employee	Employees	GROUPO
fin_code	FinancialCodes	GROUPO
fin_data	FinancialData	GROUPO
product	Products	GROUPO
sale	sale	DBA
sales_order_items	SalesOrderItems	GROUPO
sales_order	SalesOrders	GROUPO

## Creating and using an IQ demo database

- 1 Install Sybase IQ Server. See the *Sybase IQ Installation and Configuration Guide*.
- 2 Create the demo database.

---

**Note** IMPORTANT: To convert your demo database to a multiplex, create the database using an absolute path (-absolute).

---

- **On UNIX –**

- 1 Change to the IQ install directory:
  - `cd <IQ install directory>/IQ-15_3`
- 2 Source the IQ environment.
  - For bash or Korn (ksh) shell, enter: `..IQ-15_3.sh`
  - For tcsh or C (csh) shell, enter: `source IQ-15_3.csh`
- 3 If it has not been created already, create the directory to hold the new database. Following the convention of this demo, the directory is */myiqdemo*.
- 4 Change to the directory where you want to create the database:
  - `cd /myiqdemo`

5 Run the following command:

- `$IQDIR15/demo/mkiqdemo.sh`

• **On Windows –**

1 Open a command window:

- **Start | Programs | MS-DOS Prompt** or **Start | Run** and enter `cmd`.

2 If it has not been created already, create the directory to hold the new database. Following the convention of this demo, the directory is `C:\myiqdemo`.

3 Change to the directory where you want to create the database, using either of the following:

- `c:`
- `cd C:\myiqdemo`

4 Run:

```
"%ALLUSERSPROFILE%\SybaseIQ\demo\mkiqdemo"
```

5 To create the demo database in a predetermined location, choose:

**Start | Programs | Sybase | Start Sybase IQ Demo Database.**

• To list all the options to customize database creation, execute:

• **On UNIX –**

```
$IQDIR15/demo/mkiqdemo.sh -help
```

• **On Windows –**

```
"%ALLUSERSPROFILE%\SybaseIQ\demo\mkiqdemo" -help
```

An IQ database consists of multiple stores. The paths to an IQ store can be relative or absolute.

You can create the demo database using either relative or absolute paths. If you want to copy or move your demo database around, it is easier to use a relative path. This is the default option for *mkiqdemo*. To convert your demo database to a multiplex, however, you must use an absolute path.

- **On UNIX –**

- 1 Change to the directory where you want to create the database:

```
cd /myiqdemo
```

- 2 Use the `-absolute` switch:

```
$IQDIR15/demo/mkiqdemo.sh -absolute
```

- **On Windows –**

- 1 Open a command window:

Select **Start | Programs | MS\_DOS Prompt** or **Start | Run** and enter `cmd`.

- 2 Change to the directory where you want to create the database

```
c:
cd \myiqdemo
```

- 3 Run:

```
"%ALLUSERSPROFILE%\SybaseIQ\demo\mkiqdemo" -
absolute
```

- 3 If SQL Anywhere® or an earlier version of Sybase IQ resides on the same machine or subnet as Sybase IQ 15.3, edit the *iqdemo.cfg* configuration file. By default, both products use the default port 2638; therefore, you must change the port for your IQ Server.

In addition, to avoid confusion on shared systems, add your user name to the server name to make it unique. For example, if your user name is *jsmith*, you might use *jsmith\_iqdemo* as the server name for the demo database.

The *iqdemo.cfg* file is created in the same directory as the demo database. You can edit this file and change the startup parameters. You can also copy this file and change the startup parameters to make configuration files for any of your databases.

- 4 Start the IQ server. Change to a directory where the configuration and database files reside and issue a command in this format:

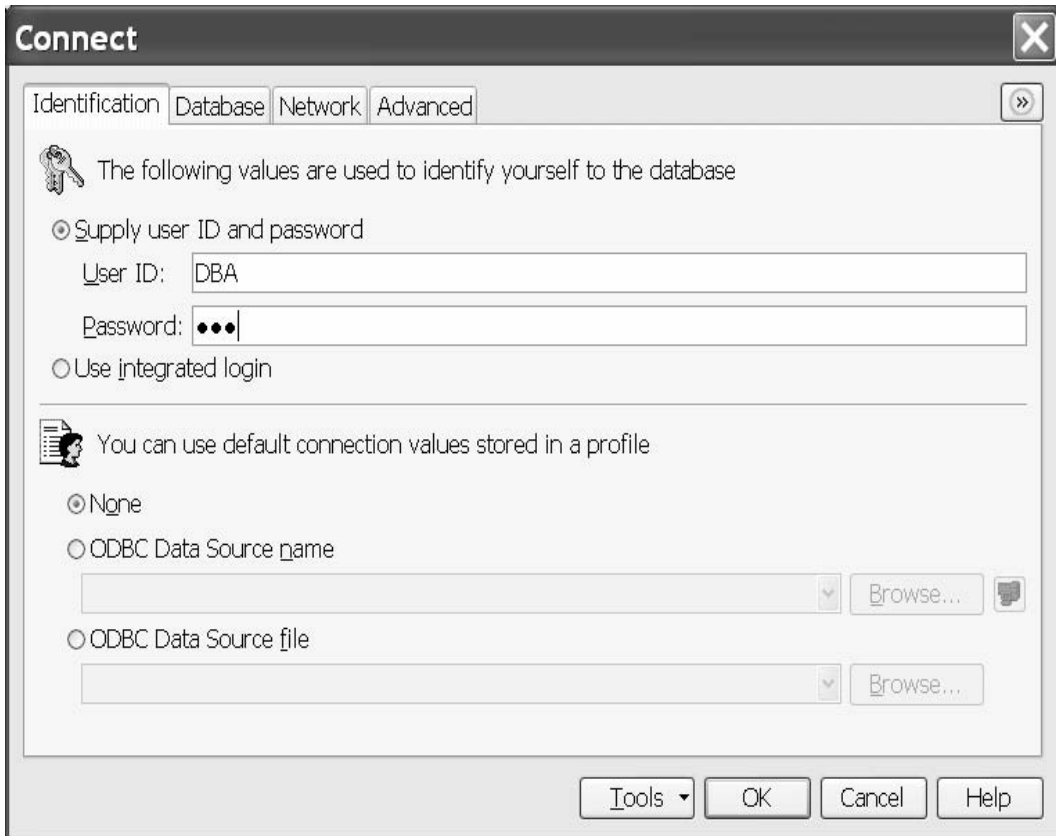
```
start_iq @configuration_file.cfg dbname.db
```

For example, to start the demo database, enter:

```
start_iq @iqdemo.cfg iqdemo.db
```

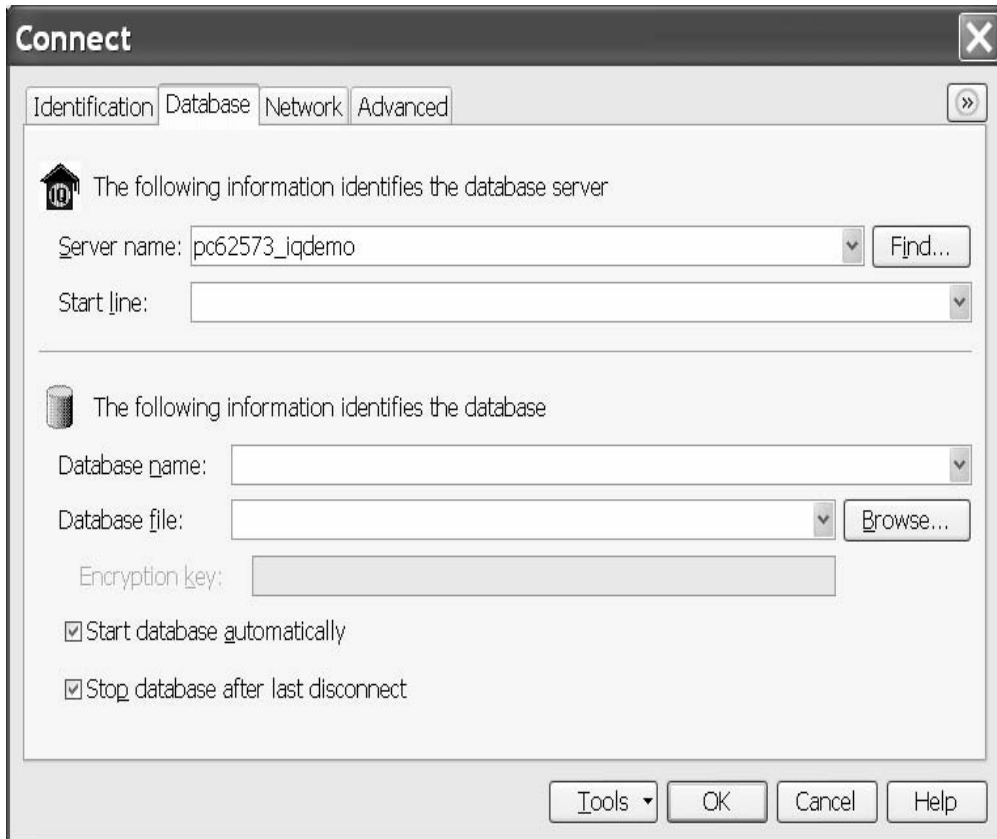
- 5 Start Interactive SQL, the query tool.
  - On UNIX, type dbisql at the system prompt.
  - On Windows, click **Start | Programs | Sybase | Sybase IQ 15.3 | Interactive SQL**, enter dbisql in a command shell, or select **Start | Run** and enter dbisql.
- 6 Connect to the demo database:
  - In the Connect dialog, enter **DBA** for the User ID and **sql** for the Password. (This example shows Windows output.)

**Figure 1: Connect dialog**



- On the Database tab, click **Find** and select the server name. If the server name is not listed, type the server name, and then click **OK**.

**Figure 2: Finding the server**

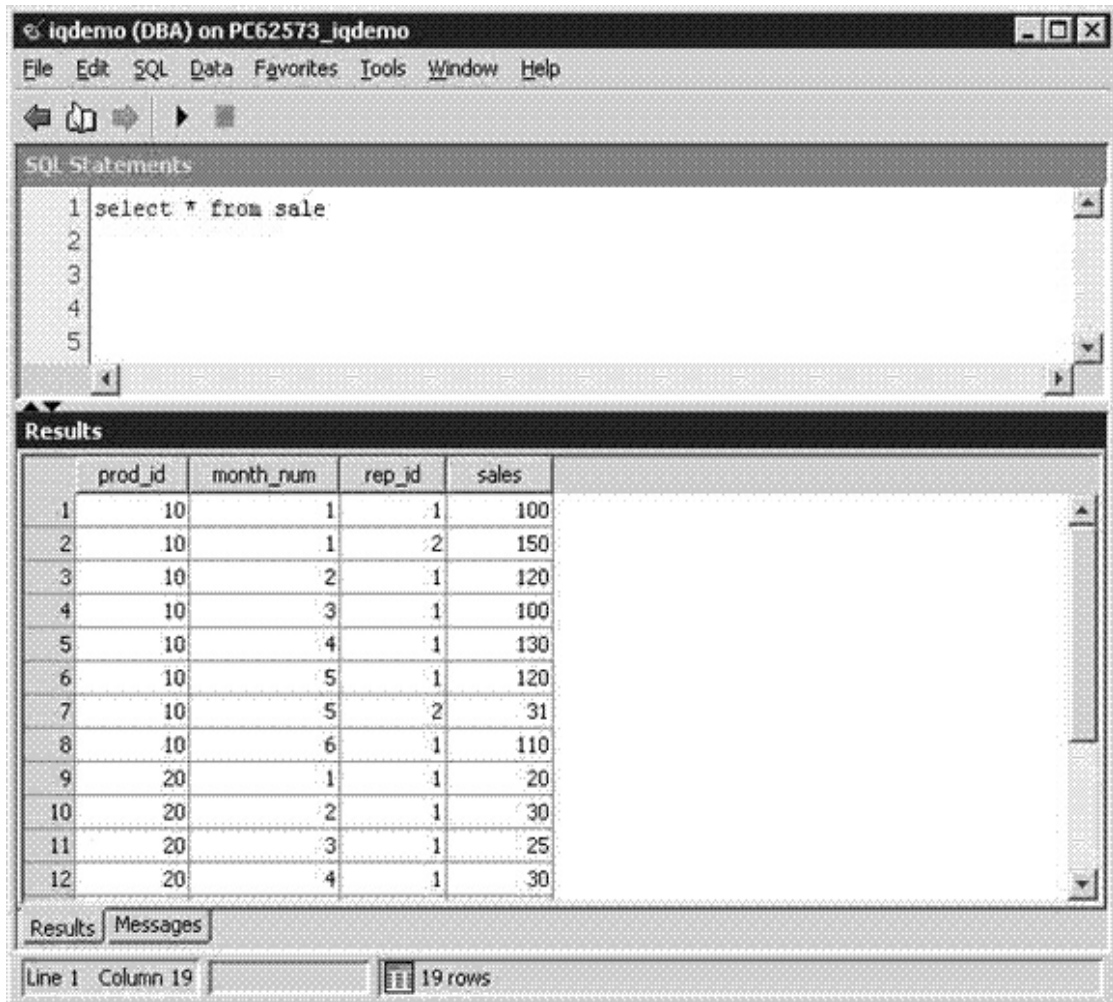


- 7 Run a test query to view columns in the sale table. In the top window (SQL Statements), type:

```
SELECT * FROM sale
```

Click **Execute** (right-facing triangle in the button bar).

Figure 3: Running a test query



8 To exit Interactive SQL, select **File | Exit**.



## Converting the IQ demo database to multiplex

**Note** IMPORTANT: To convert your demo database to a multiplex, you must use absolute paths.

Verify that the demo database was created using absolute paths (-absolute option). If the demo database uses relative paths, the multiplex creation fails.

A single server (IQ demo database) is converted to a multiplex server by adding a secondary server. You can convert the server and manage the resulting multiplex using Sybase Central. All servers in the multiplex appear in a single Multiplex folder.

### Starting the IQ Agent

Before you start Sybase Central, start the IQ Agent on the machines where the IQ servers will run.

#### On UNIX –

- 1 Start the agent using the *S99SybaseIQAgent15* script.
  - Command *S99SybaseIQAgent15* starts the agent at default port 1099.
  - Command *S99SybaseIQAgent15 -port 3871* starts the agent at port 3871.
- 2 To verify that the agent is running, execute:

```
stop_iq -agent
```

For example, the following agent is owned by user 'smith.' Do not stop the server.

##owner	PID	Started	CPU Time	Additional Information
-----	-----	-----	-----	-----
1: smith	15549	Feb. 18	10:38	PORT:2008
java -Diq.agent=/sun625742/users/smith/sybase/IQ-15_3/java/IQAgent1530.jar -D				

- 3 Verify that the agent log exists.

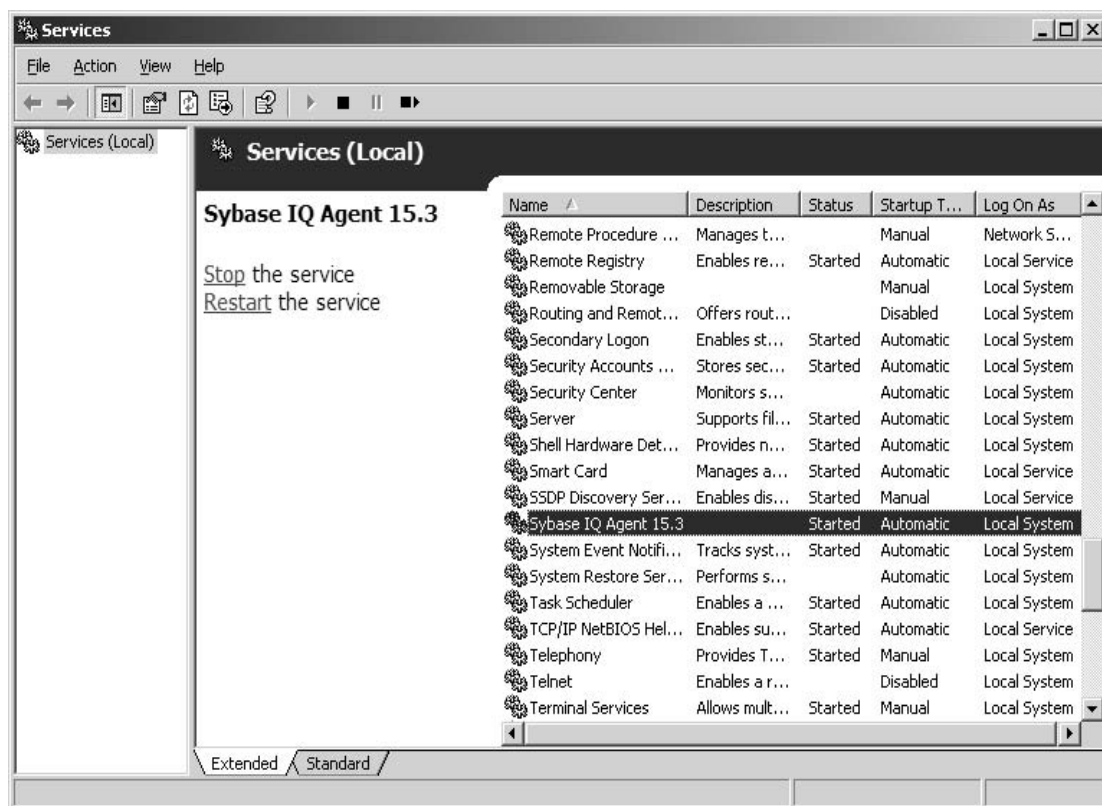
The agent log file is *\$IQDIR15/logfiles/SybaseIQAgentNNNN.MMM.log* where *NNNN* is the port number and *MMM* is a sequential number. If the environment variable *IQLOGDIR15* is set, the agent log files are in the *\$IQLOGDIR15* directory.

### On Windows –

The agent is configured to run as a service.

- 1 Start the Services Manager application to verify that the agent is running.

**Figure 4: Services Manager**



- 2 Verify that the agent log exists.

- If you rebooted after installing Sybase IQ, the agent log file is *%ALLUSERSPROFILE%\SybaseIQ\logfiles\SybaseIQAgent.NNN.log* where *NNN* is a sequential number.

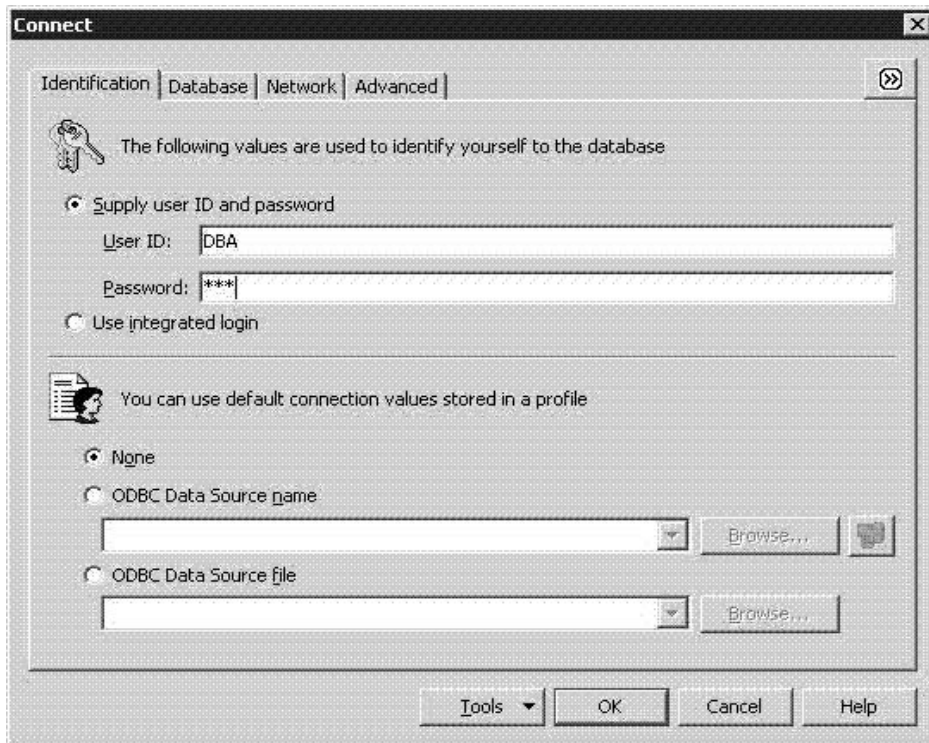
- If you did not, the agent log file is in %SYBASE%\IQ-15\_3\logfiles.
- If the environment variable IQLOGDIR15 is set, the agent log files are in the %IQLOGDIR15% directory.

## Converting to multiplex

### ❖ Converting the IQ demo database to multiplex

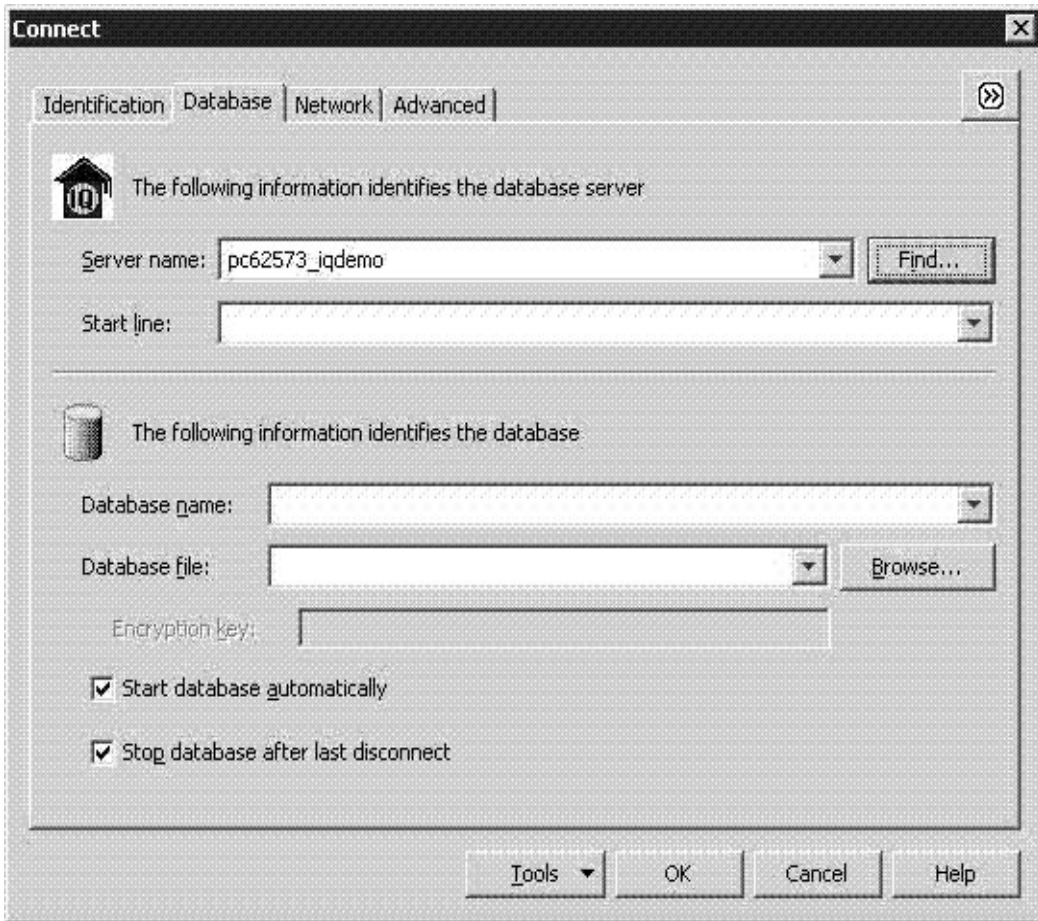
- 1 Launch Sybase Central.
  - On UNIX, type **scjview** at the system prompt.
  - On Windows, choose **Start | Programs | Sybase | Sybase IQ 15.3 | Sybase Central**.
- 2 Choose **Connections | Connect with Sybase IQ 15**. In the Connect dialog, type **DBA** for the User ID and **sql** for the Password.

**Figure 5: Connecting, ID tab**



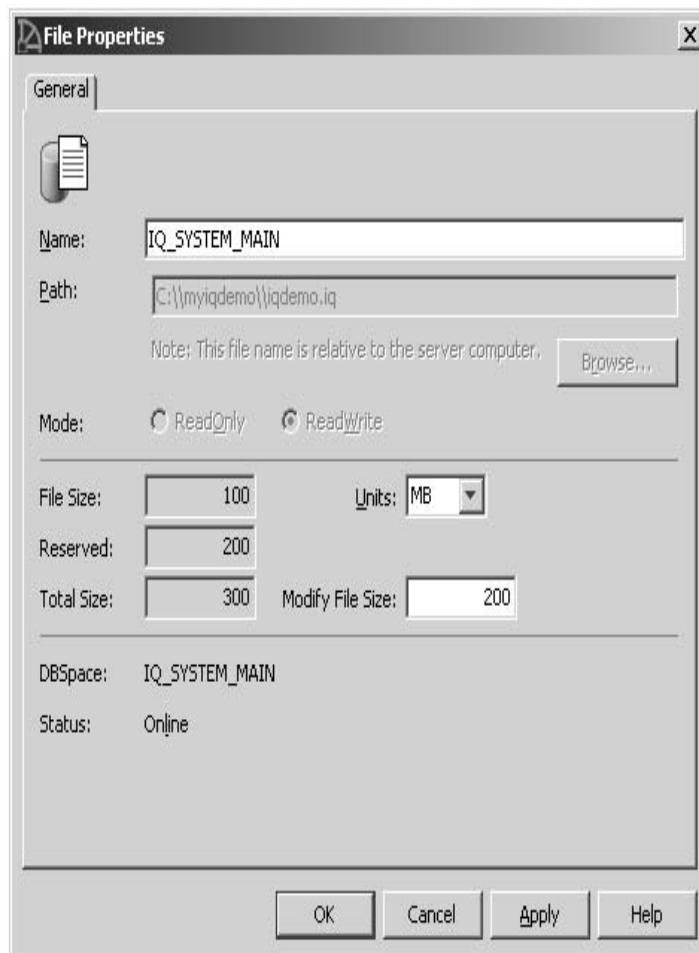
On the Database tab, click **Find** and select the server name. If the server name is not listed, type the server name, then click **OK**.

**Figure 6: Connecting, database tab**



- 3 Increase the file size in IQ\_SYSTEM\_MAIN:  
 In the folders view, double-click the DBSpaces folder. Then select the IQ\_SYSTEM\_MAIN folder. On the Files tab, right-click the file IQ\_SYSTEM\_MAIN and select **Properties**. In the Properties dialog, enter 200 in **Modify File Size**. This increases the IQ\_SYSTEM\_MAIN space to 200MB.

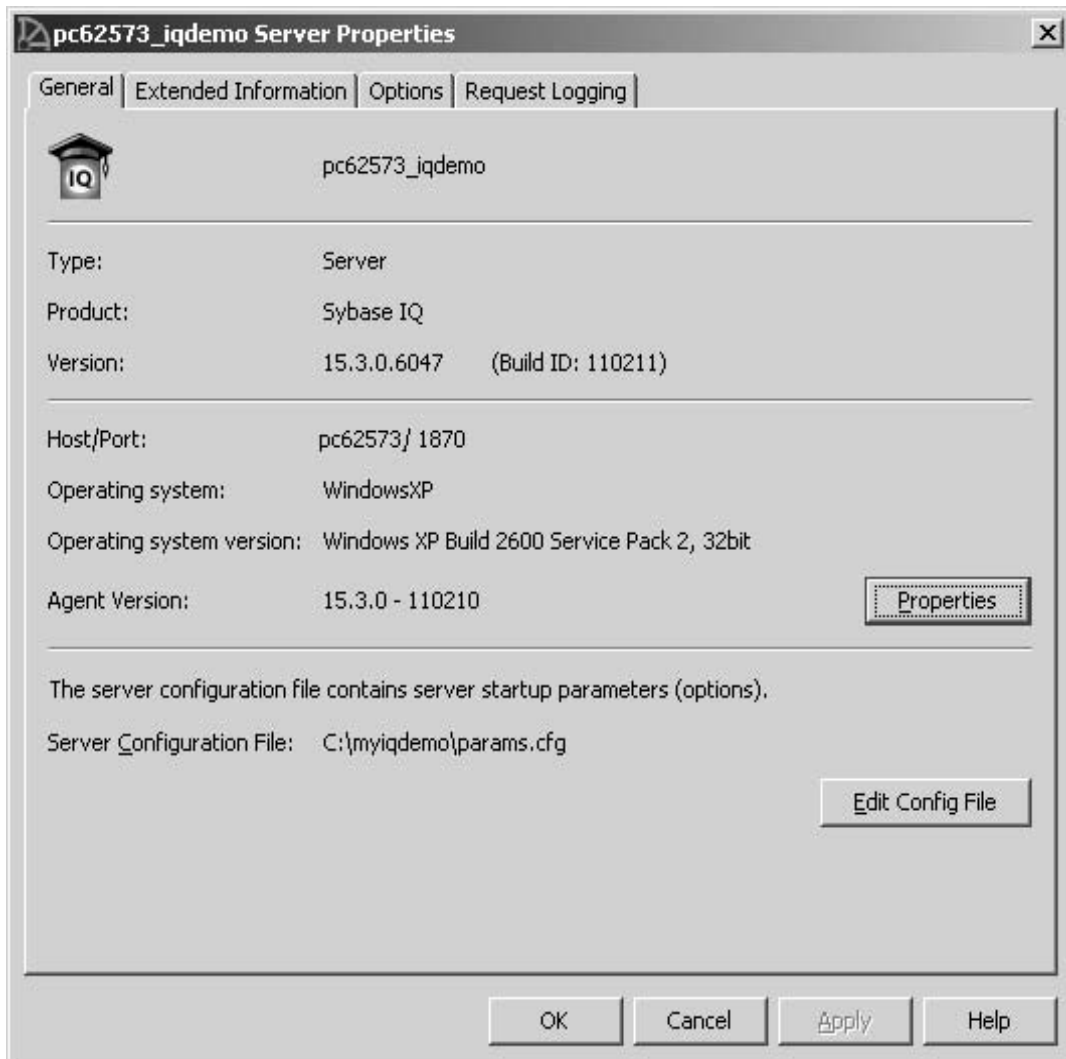
**Figure 7: File properties**



- 4 Click **Apply**, then **OK**.

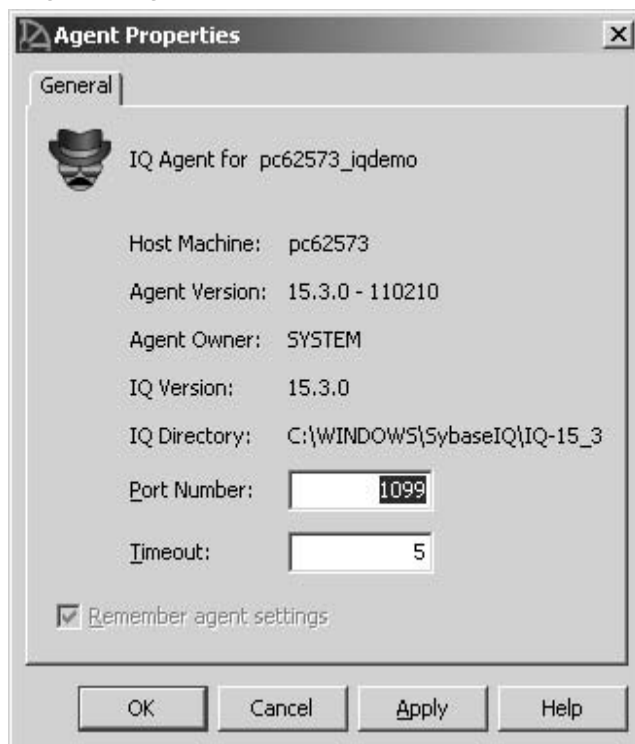
- 5 To confirm the agent port in the folders view, under the Servers folder, right-click the **iqdemo** server icon and click **Properties**.

**Figure 8: iqdemo server properties**



Click **Properties** next to Agent Version.

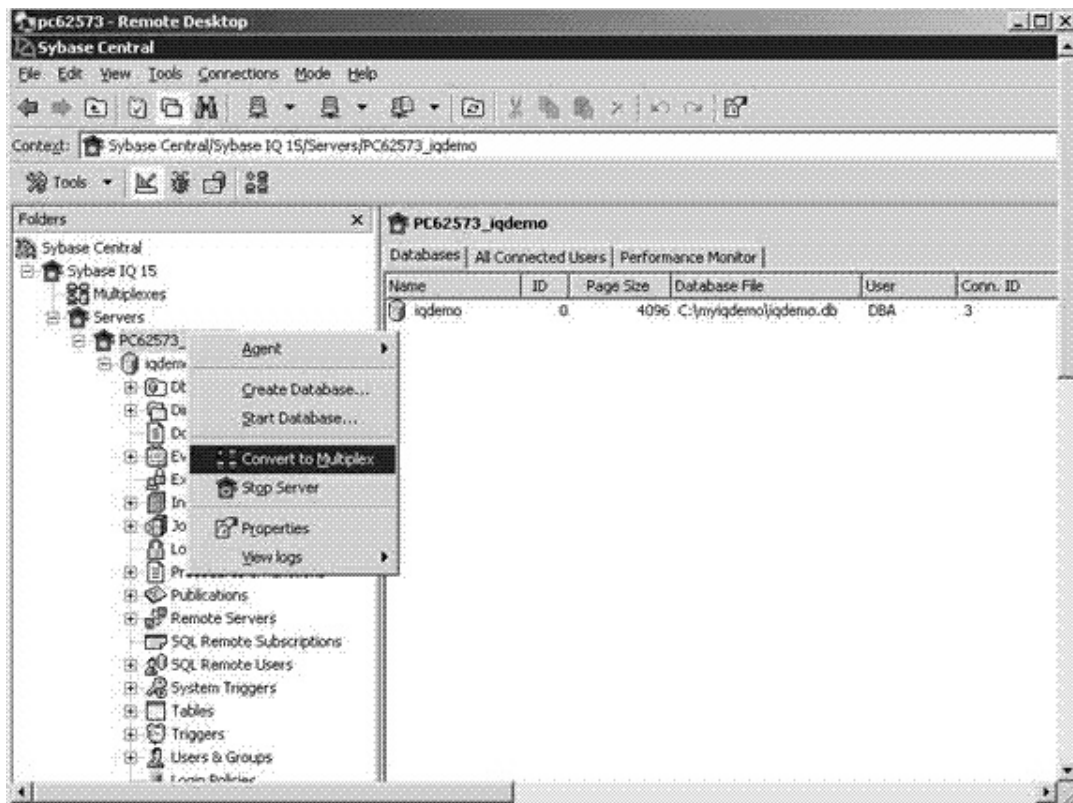
**Figure 9: Agent properties**



Verify that the agent port is the one on which you started the agent. Click **Cancel** to return to the Server Properties dialog and **OK** to return to the Folders view.

- 6 In the folders view, under the Servers folder, right-click the **iqdemo** server icon and choose **Convert to Multiplex**.

**Figure 10: Folders view**





7 This starts the Create Server wizard.

**Figure 11: Create Server wizard**

**Create Server Wizard**

**Node Configuration**

Specify configuration information for the new multiplex node. On a host with multiple network cards, unique names and ports may be specified for each card.

Specify the name for this new multiplex:

Specify the new secondary server's information:

Connection Profile:

Server Name:

Host Information:

Hostname	Port Number
PC62573	2639

Agent Port:

Mode:  Reader  **Writer**

Register new node as the designated fail-over node.

Create administrative shell scripts.

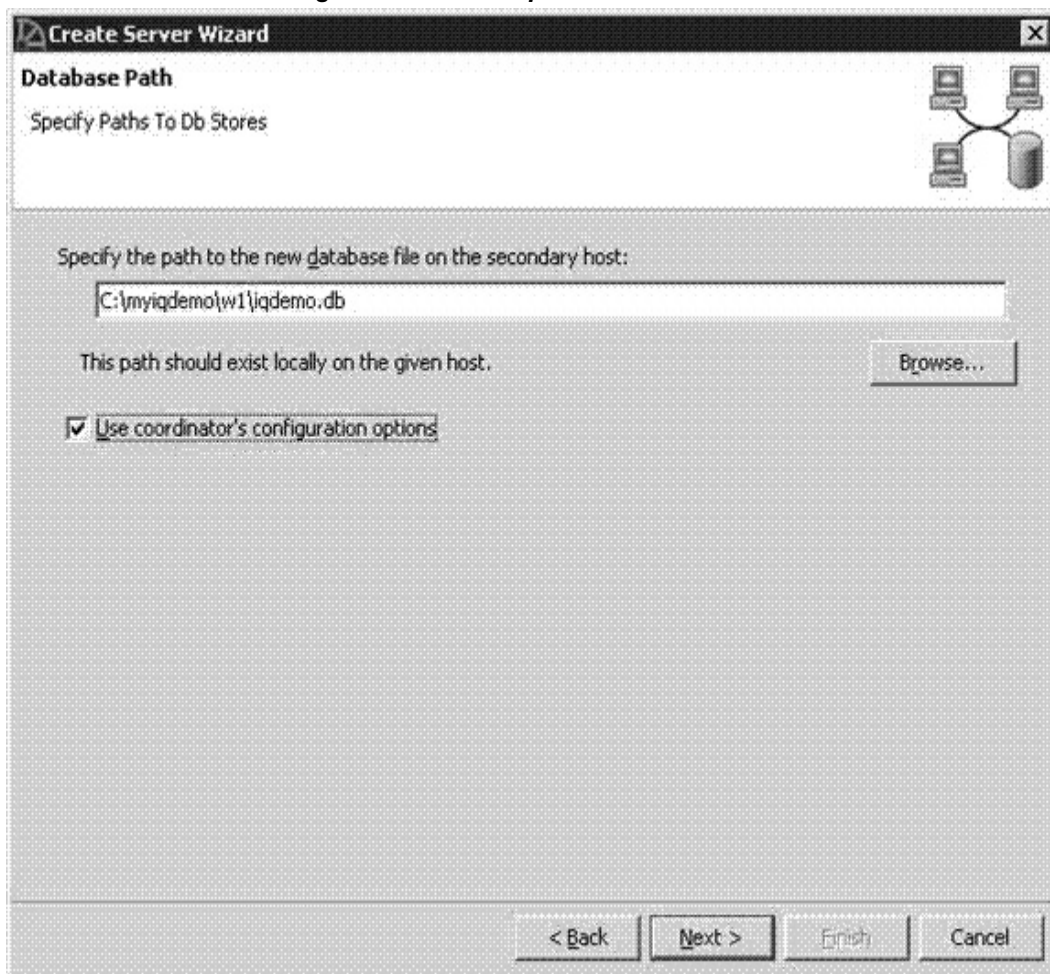
Fill all fields as shown.

- Multiplex Name:
  - Defaults to the original single server name, but may be changed (for example, as shown).
  - This is a logical name of the multiplex used to uniquely identify the multiplex within Sybase Central.

- Server Name:
    - Name of the first secondary server.
    - Must be different from the original server, which becomes the coordinator server for the multiplex.
  - Host Information:
    - Host name – For the demonstration, use the same host as the original server. This sets up the multiplex with multiple servers on the same host. In a production setup, use different hosts for each multiplex server.
    - Port Number – Specify a unique port number for the secondary server. This Port Number defaults to the original server port number + 1.
    - You can add multiple host and port pairs here, to configure a single server with multiple network addresses. In most cases, a multiplex server only has a single network address.
  - Private Host Information:

If you wish to use distributed query processing, specify host and port addresses for a private high-speed interconnection. The private network must support the TCP/IP protocol.
  - Agent Port – enter the port where the agent is started. The default port is 1099.
  - Mode – select **Writer** (the default is Reader). Later, you will load data from this node.
  - Select **Create administrative shell scripts**.
- 8 Click **Next**. Choose the path for the catalog database. This should be a different directory than where you created the IQ demo database. You can create a subfolder under the directory where the demo database was created.

Figure 12: Database path

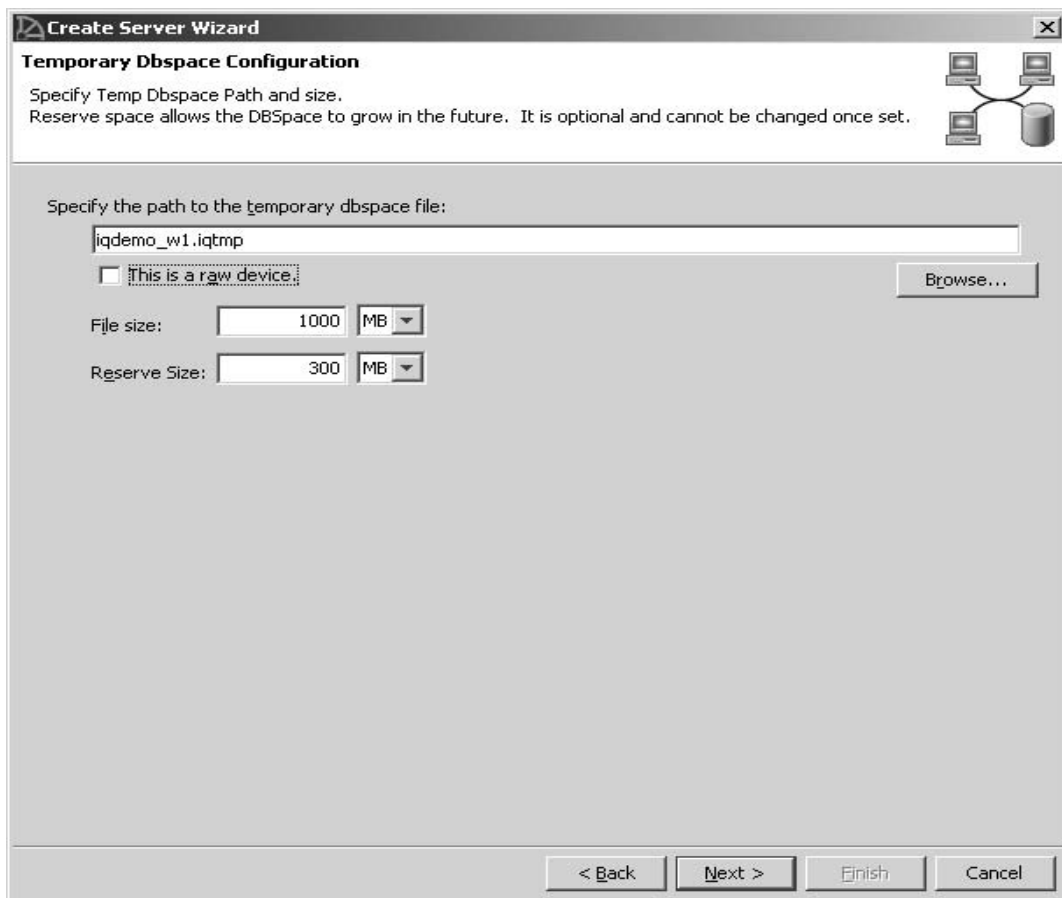


Complete the information on the Database Path page:

- Database path
  - The database path is an absolute path to the *db* file for the new multiplex secondary server.
  - This path must be different from the existing database's *db* file path. Sybase suggests that you use *iqdemo.db* as the file name; typically you can use any name you want.

- Select **Use coordinator's configuration options**. This option copies the `params.cfg` options used for the existing database to the new multiplex server's `db` directory.
- 9 Click Next. If the directory does not exist, click **Yes** when you are asked if you want to create it. The Temporary Dbspace Configuration page appears.

**Figure 13: Temporary dbspace configuration**



The screenshot shows a dialog box titled "Create Server Wizard" with a sub-tab "Temporary Dbspace Configuration". The dialog contains the following elements:

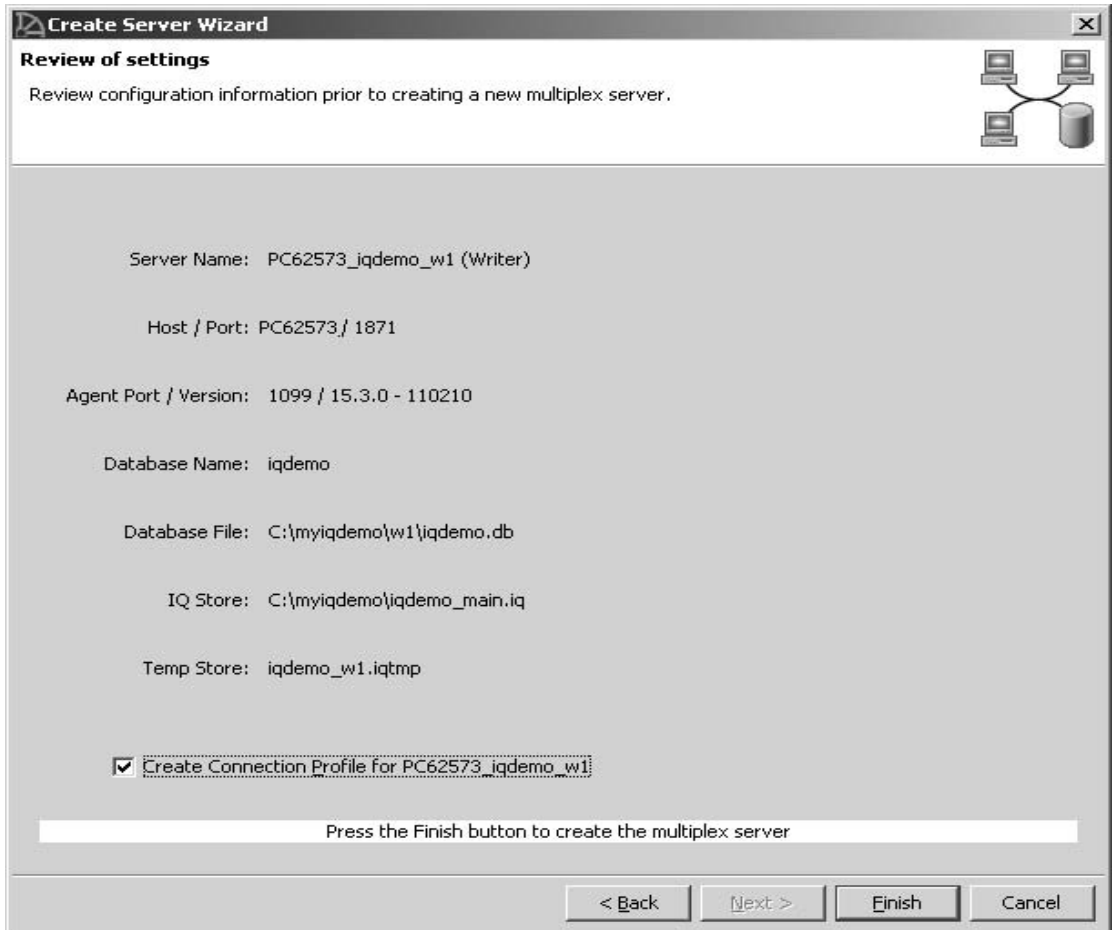
- Instructions: "Specify Temp Dbspace Path and size. Reserve space allows the DBSpace to grow in the future. It is optional and cannot be changed once set."
- A text input field containing the path "iqdemo\_w1.iqtmp".
- A checkbox labeled "This is a raw device." which is currently unchecked.
- A "Browse..." button to the right of the path field.
- Two rows of controls for file size and reserve size, each consisting of a text input field, a numeric value, and a unit dropdown menu:
  - File size: 1000 MB
  - Reserve Size: 300 MB
- Navigation buttons at the bottom: "< Back", "Next >", "Finish", and "Cancel".

Complete the information on the Temporary Dbspace Configuration page.

- Temporary dbspace file path: `iqdemo_w1.iqtmp`. The temporary *dbspace* file path may be absolute or relative to the database `.db` file path. Choose a unique file name.

- This is a raw device – leave unselected.
  - The file size is 1000MB.
  - The reserve size is 300.
- 10 Click **Next**. Review your settings.

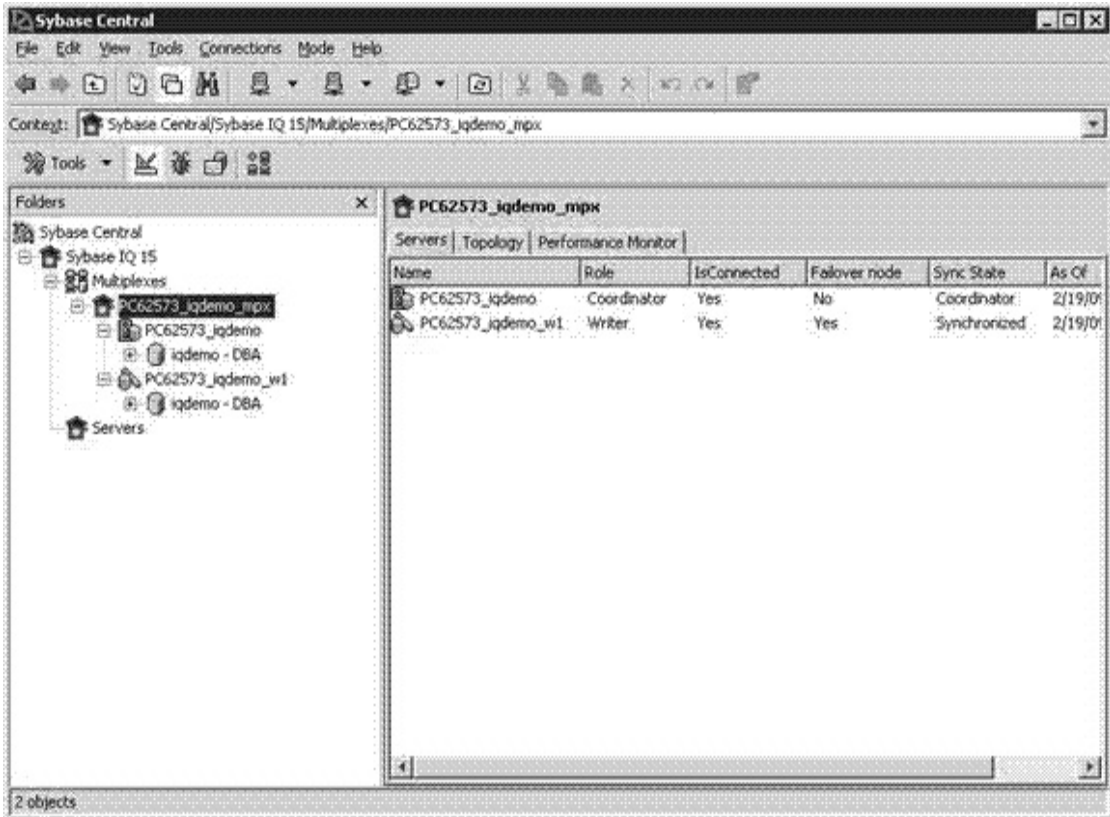
**Figure 14: Review of settings**



- 11 Click **Finish**. This creates the new multiplex secondary server files and configures the database for multiplex. This operation may take several minutes.

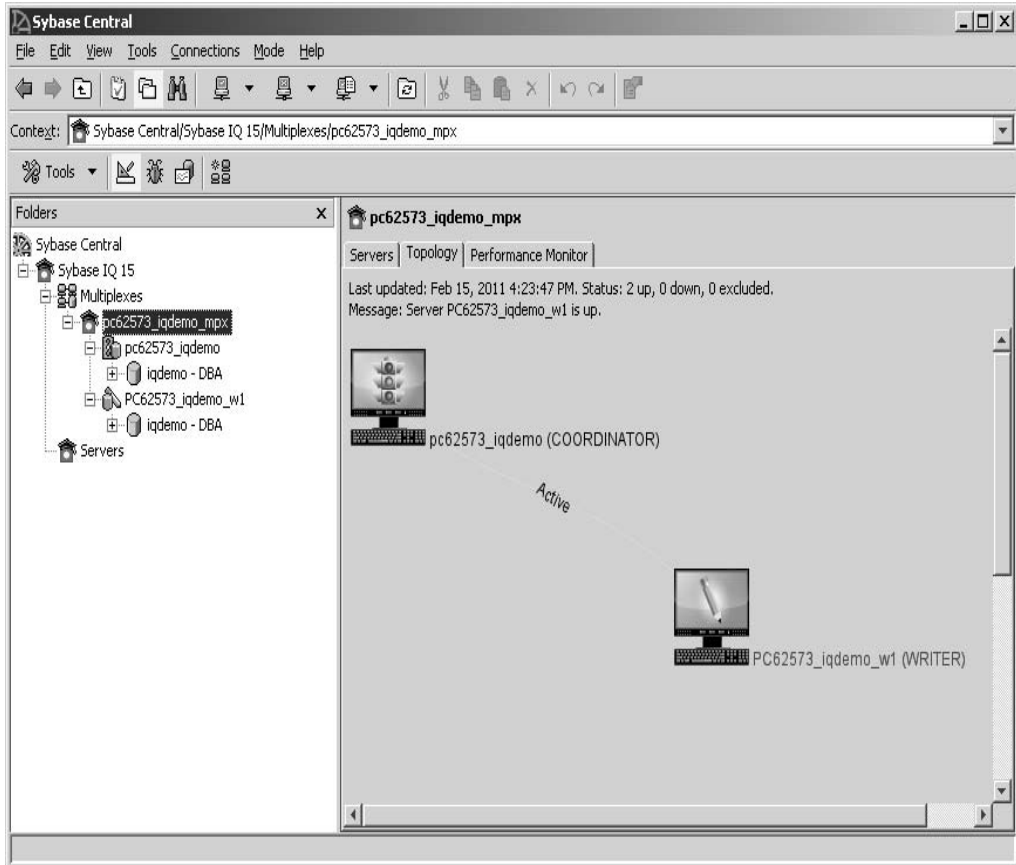
In the Sybase Central Folders view, the multiplex name is inserted into the Multiplexes folder. Expand the multiplex name to view the servers in the multiplex.

**Figure 15: Sybase Central multiplex servers**



- 12 In the Sybase Central main window, click the **Topology** tab to display the multiplex topology view. This is a graphical view of the multiplex configuration, which displays the status of each server and the communication links between servers. You can execute most multiplex operations either from this view or from the Servers view.

**Figure 16: Topology tab**



## Adding a multiplex server (manual method)

The current multiplex configuration contains two servers: the coordinator server and one multiplex writer server. Use the instructions in this section to manually add a second server in the reader role. The new reader server is called PC62573\_iqdemo\_r2.

---

**Note** IMPORTANT: To convert your demo database to a multiplex, you must use absolute paths.

---

Open Interactive SQL. Alternatively, you can launch the Create Server wizard in Sybase Central by choosing **File | Add Server**.

The following steps are required when you are manually creating a multiplex server; if you are using the Create Server wizard, these steps are automatically performed by Sybase Central.

### Creating the multiplex server definition

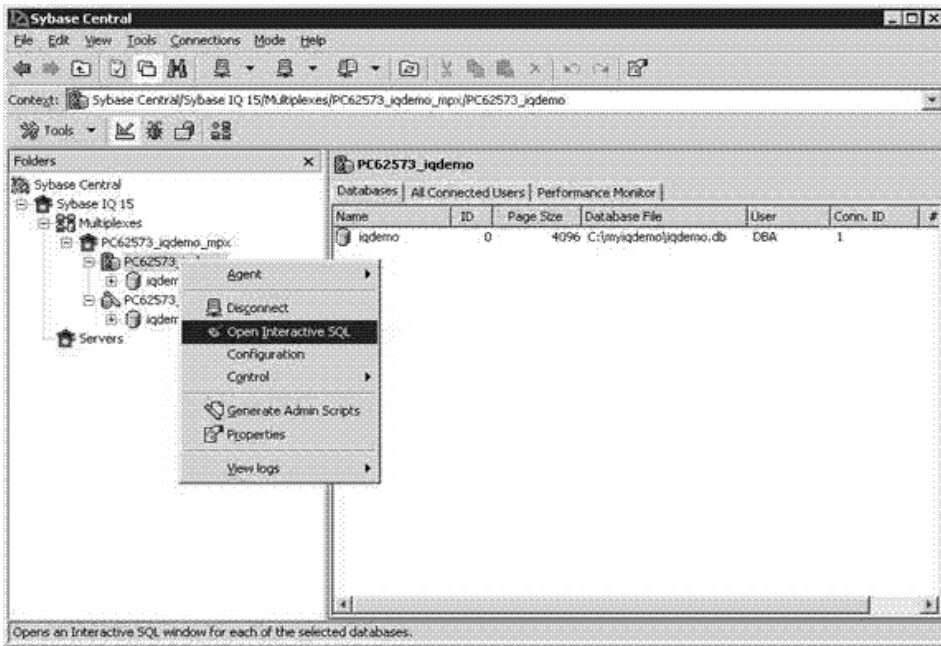
Define the multiplex server in the database system table SYSIQMPXSERVER.

❖ **Defining the multiplex server manually**

- 1 From the Sybase Central Folders view, launch Interactive SQL.



Figure 17: Launching Interactive SQL



2 In Interactive SQL, type the following command, providing the following:

- DATABASE
  - A full *db* file path in a local directory to contain the PC62573\_iqdemo\_r2 server's files.
  - A correct path. An incorrect path is not detected until it is time to synchronize the new multiplex server.
- HOST – the name of the local host
- PORT – a unique port number on the local host. Following the conventions of this demo, use port 2640.

**On UNIX –**

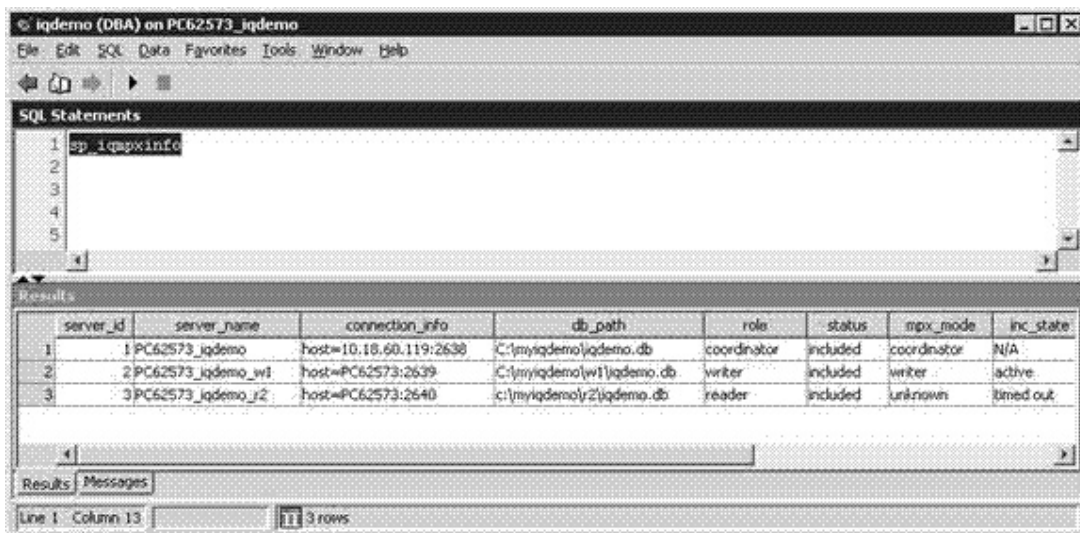
```
CREATE MULTIPLEX SERVER SUN62574_iqdemo_r2
DATABASE '/myiqdemo/r2/iqdemo.db' ROLE READER
HOST 'SUN62574' PORT 2640
```

**On Windows –**

```
CREATE MULTIPLEX SERVER PC62573_iqdemo_r2
DATABASE 'C:\myiqdemo\r2\iqdemo.db' ROLE READER
HOST 'PC62573' PORT 2640
```

- 3 Press F5 to execute the command.
- 4 To view the new multiplex configuration for all servers, execute the `sp_iqmpxinfo` stored procedure, which produces the following output:

**Figure 18: iqdemo (DBA)**



server_id	server_name	connection_info	db_path	role	status	mpx_mode	inc_state
1	PC62573_iqdemo	host=10.18.60.119:2638	C:\myiqdemo\iqdemo.db	coordinator	included	coordinator	N/A
2	PC62573_iqdemo_w1	host=PC62573:2639	C:\myiqdemo\w1\iqdemo.db	writer	included	writer	active
3	PC62573_iqdemo_r2	host=PC62573:2640	c:\myiqdemo\r2\iqdemo.db	reader	included	unl:rown	timed out

## Synchronizing and starting the multiplex server (manual method)

Generate a *db* file from the coordinator's catalog store and copy it to the newly created reader's directory, then start the new server.

Alternatively, you can do this in Sybase Central by right-clicking the multiplex server icon and choosing **Control | Synchronize**.

### ❖ Manually starting the multiplex server

- 1 If it has not been created already, create the multiplex server directory. Following the convention of this demo, the multiplex directory is:
  - **On UNIX** – `/myiqdemo/r2`
  - **On Windows** – `C:\myiqdemo\r2`

- 2 Change to the newly created multiplex server directory:
  - **On UNIX** – `cd /myiqdemo/r2`
  - **On Windows** – `cd C:\myiqdemo/r2`
- 3 If it does not exist, copy the *params.cfg* from the coordinator's directory to the reader's directory.
- 4 If a previous transaction log file exists in the directory (for example, *mpx.log*), remove it. Leaving old transaction logs in a multiplex server directory after synchronizing may stop the server from starting.
- 5 From the command line, run the following commands, substituting the appropriate path name, host name, and server name for the new multiplex server. Type each command below on a single line.

- **On UNIX** –

```
dbbackup -y -d -c
'uid=DBA;pwd=sql;eng=SUN62574_iqdemo;links="tcp
ip{host=SUN62574;port=2638}"' .
```

```
rm -rf "iqdemo.log"
```

```
dblog -r -t "iqdemo.log" "iqdemo.db"
```

```
start_iq @params.cfg
-n SUN62574_iqdemo_r2
-x "tcpip{host=SUN62574;port=2640}" iqdemo.db
```

- **On Windows** –

```
dbbackup -y -d -c
"uid=DBA;pwd=sql;eng=PC62573_iqdemo;links=tcpi
p{host=PC62573;port=2638}" .
```

```
erase /F /Q "iqdemo.log"
```

```
dblog -r -t "iqdemo.log" "iqdemo.db"
```

```
start_iq @params.cfg
-n PC62573_iqdemo_r2
-x "tcpip{host=PC62573;port=2640}" iqdemo.db
```

Generally, you use ODBC DSN for dbbackup, and edit the configuration file, add the links information, and the server name in the configuration file to start the server.

---

**Note** A problem occurs starting the server when there is a delay of more than 60 minutes between Step 3 and Step 5. This is because the default value of MPX\_AUTOEXCLUDE\_TIMEOUT is 60 minutes.

To correct the problem, connect to the coordinator and include the new server using `ALTER MULTIPLEX SERVER <new-server> STATUS INCLUDED`. After that, synchronize again (run dbbackup and dblog) and restart the server.

---

- 6 Confirm that the server is running by connecting to it using Interactive SQL.

## Adding a temporary store file (manual method)

You cannot query IQ data until you add a temporary store file for the secondary node. Repeat the following step as needed to add additional temporary store files if you need more temporary storage.

### ❖ Adding a temporary store file

The coordinator node and the secondary node must be running to perform this procedure.

- 1 Use Interactive SQL to connect to the reader.
- 2 Run the following command, substituting the appropriate file path. The temporary store file path must be a unique path for each server or host. The path may be absolute or relative, as long as the path resolves to a separate physical file for each multiplex server.

- **On UNIX –**

```
ALTER DBSPACE IQ_SYSTEM_TEMP ADD FILE
iqdemo_r2_temp 'iqdemo_r2_temp.iqtmp'
SIZE 300 RESERVE 300
```

- **On Windows –**

```
ALTER DBSPACE IQ_SYSTEM_TEMP ADD FILE
iqdemo_r2_temp 'iqdemo_r2_temp.iqtmp'
SIZE 300 RESERVE 300
```

Execute `sp_iqdbspace 'IQ_SYSTEM_TEMP'` to confirm that reader now has free temporary space.

## Loading data

In this section, you load data into the multiplex database using the coordinator and a writer. This is the most efficient use of multiplex resources because it distributes the workload to multiple servers. In this demonstration, different servers are used in serial to load data. In a production environment, you can perform loads on separate tables in parallel from multiple servers.

This demonstration uses the `Contacts` and `SalesOrderItems` tables using the load files that ship with IQ 15.3.

On UNIX, these files are in `$IQDIR15/demo/adata`.

On Windows, these files are in  
`%ALLUSERSPROFILE%\SybaseIQ\demo\adata`.

Substitute the appropriate path to these input files in the following LOAD statements.

### ❖ Loading data from the coordinator

- 1 Use Interactive SQL to connect to the coordinator server PC62573\_iqdemo.
- 2 Truncate the `Contacts` table because it gets loaded with the same data during the creation of the demo database.
- 3 Load the `Contacts` table. (Type the file path on a single line, not as shown.)

#### On UNIX –

```
TRUNCATE TABLE Contacts;

LOAD TABLE Contacts (ID, Surname, GivenName,
Title, Street, City, State, Country, PostalCode,
Phone, Fax, CustomerID)
USING FILE '/sun62574/users/userid/
/syb_install_dir/IQ-15_3/demo/adata/contact.dat'
ROW DELIMITED BY '|'
ESCAPES OFF;

COMMIT;
```

### On Windows –

```
TRUNCATE TABLE Contacts;

LOAD TABLE Contacts (ID, Surname, GivenName,
Title, Street, City, State, Country,
PostalCode, Phone, Fax, CustomerID)
USING FILE
'C:\Documents and Settings\All Users
\SybaseIQ\demo\adata\contact.dat'
ROW DELIMITED BY '|'
ESCAPES OFF;

COMMIT;
```

### ❖ Loading data from the writer

- 1 Use Interactive SQL to connect to the writer server PC62573\_iqdemo\_w1 server.
- 2 Truncate the SalesOrderItems table, as it gets loaded with the same data during the creation of the demo database.
- 3 Load the SalesOrderItems table. (Type the file path on a single line, not as shown.)

### On UNIX –

```
TRUNCATE TABLE SalesOrderItems;

LOAD TABLE SalesOrderItems (ID, LineID,
ProductID, Quantity, ShipDate)
USING FILE '/sun62574/users/userid/
/syb_install_dir/IQ-15_3/demo
/adata/sales_oi.dat'
ROW DELIMITED BY '|'
ESCAPES OFF;

COMMIT;
```

### On Windows –

```
TRUNCATE TABLE SalesOrderItems;

LOAD TABLE SalesOrderItems (ID, LineID,
ProductID, Quantity, ShipDate)
USING FILE 'C:\Documents and Settings\All Users\
SybaseIQ\demo\adata\sales_oi.dat'
ROW DELIMITED BY '|'
ESCAPES OFF;
```

```
ESCAPES OFF;
```

```
COMMIT;
```

When the data is loaded into the database, you can execute queries against the data from any server in the multiplex. You can use this base multiplex for further experimentation. You can also explore how to perform additional operations on this data by adapting the iqdemo scripts.

---

**Note** If you have questions about installing or running the software, contact Sybase Technical Support or the Sybase subsidiary in your area.

---

## Distributed query processing

Distributed query processing (DQP) is dividing a query into multiple independent pieces of work, distributing that work to other nodes in the multiplex, and collecting and organizing the intermediate results to generate the final result set for the query. The servers in a Sybase IQ multiplex connect to a central store for permanent shared data, such as a shared disk array. Multiplex has a hybrid cluster architecture that uses shared storage for permanent IQ data, and independent node storage for catalog metadata, private temporary data, and transaction logs.

A simple query runs fastest on a single machine. However, when large and complex queries exceed a single machine's CPU capacity, the Sybase IQ query optimizer breaks the query into parallel “fragments” to concurrently run the distributed query on other servers in the multiplex.

---

**Note** Distribution requires network and storage overhead to assign work, and store and transmit intermediate results. If a query does not fully use the CPU resources on a single machine, you see better results using a single machine. For example, if the optimizer is going to parallelize a query 7 ways (thereby keeping 7 threads at a time busy) on an 8-core box, Sybase IQ does not distribute the query.

---

Before using DQP, you must:

- Configure a multiplex. (See “Converting the IQ demo database to multiplex” on page 9.)

- Set up temporary storage shared between multiplex servers. (See “Adding shared temporary storage” on page 32 or “Adding shared temporary storage (manual method)” on page 35.)

“Running a query that gets distributed” on page 37 tells how to run a query that is distributed to the two servers in the demo multiplex.

## Adding shared temporary storage

Distributed query processing (DQP) requires temporary space for use by participating multiplex nodes. This method uses the Sybase Central GUI to add temporary storage.

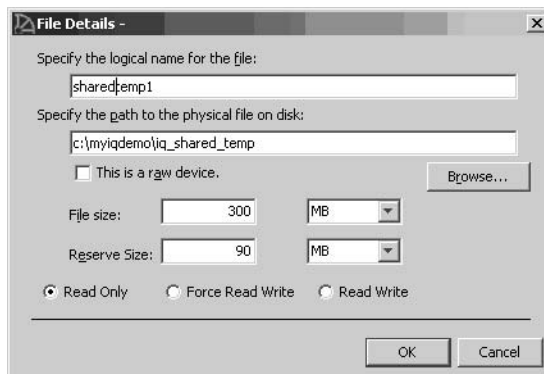
### ❖ Adding shared temporary storage

New and freshly migrated databases contain an empty IQ\_SHARED\_TEMP dbspace. To enable DQP, add files to this dbspace, then make the files read-write accessible from all multiplex nodes.

All multiplex servers must be running.

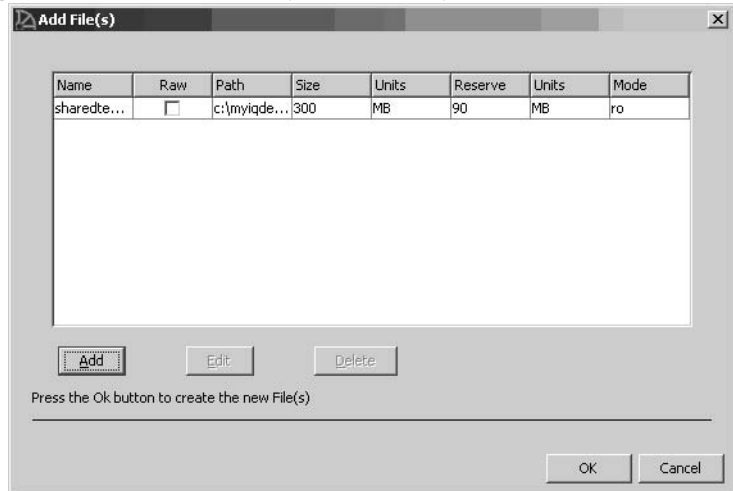
- 1 From Sybase Central, connect to the coordinator.
- 2 In the folders view, double-click the DBSpaces folder. In the Dbspaces folder, double-click IQ\_SHARED\_TEMP. Right-click IQ\_SHARED\_TEMP and choose New | File.
- 3 Click Add.

**Figure 19: Shared temporary file details**



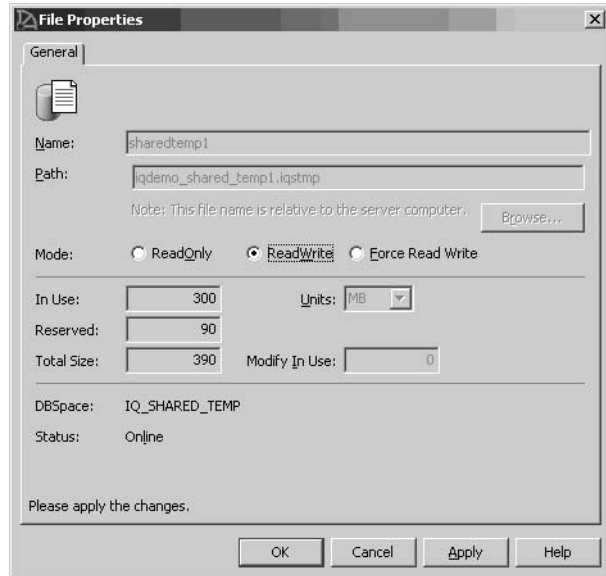
- 4 Type a logical name, absolute path, file size, and reserve size for the new file, and click OK.



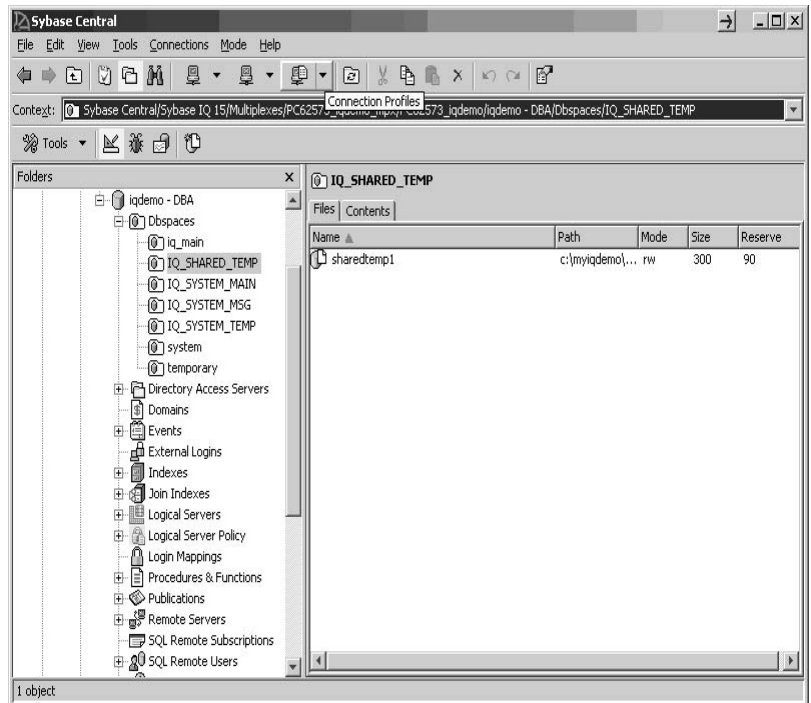
**Figure 20: Shared temporary file summary**

- 5 Verify your selections and click OK.
- 6 Wait for the new file information to propagate to all multiplex nodes. This may take a few minutes.
- 7 Right-click IQ\_SHARED\_TEMP and choose Properties.

**Figure 21: Shared temporary file properties**



- 8 Choose ReadWrite mode and click OK.

**Figure 22: Shared temporary file mode**

On the Files tab, verify that the mode (third column) displays rw as shown.

## Adding shared temporary storage (manual method)

Distributed query processing (DQP) requires temporary space for use by participating multiplex nodes. This method uses Interactive SQL to add temporary space.

### ❖ Adding shared temporary storage (manual method)

New or freshly migrated databases create an empty `IQ_SHARED_TEMP` dbspace. To enable distributed query processing, add files to this dbspace and make them read-write accessible from all nodes in the multiplex.

- 1 Start Interactive SQL and connect to the coordinator.

- 2 Run the ALTER DBSPACE ADD FILE command.

**On UNIX –**

```
ALTER DBSPACE IQ_SHARED_TEMP ADD FILE  
iqsharedtemp2 '/myiqdemo/iq_shared_temp2.iqstamp'
```

**On Windows –**

```
ALTER DBSPACE IQ_SHARED_TEMP ADD FILE iqsharedtemp2  
'c:\myiqdemo\iq_shared_temp2.iqstamp'
```

- 3 Wait for the new file information to propagate to all multiplex nodes. This may take a few minutes.
- 4 Change the new file to read-write mode:

```
ALTER DBSPACE IQ_SHARED_TEMP ALTER FILE  
iqsharedtemp2 READWRITE
```

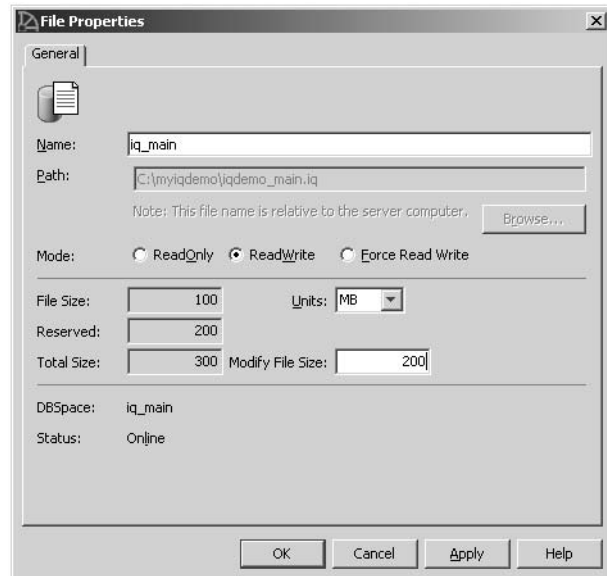
IQ\_SHARED\_TEMP is configured and your multiplex is ready for distributed queries.

## Increasing file size in iq\_main

Before running distributed queries, increase the file size in user dbspace iq\_main.

❖ **Increasing file size in iq\_main**

- 1 In the folders view, double-click the DBSpaces folder. Then select the iq\_main folder.
- 2 On the Files tab, right-click the file *iq\_main* and select Properties.
- 3 In the Properties dialog, enter 200 in Modify File Size and click OK. This increases the iq\_main space to 200MB.

**Figure 23: File properties for user dbspace iq\_main**

## Running a query that gets distributed

DQP introduces leader and worker nodes. A leader node is where a query originates. A worker node can be any node in the multiplex that is capable of accepting distributed query processing work.

Any multiplex node type (reader, writer, or coordinator) can be a leader or a worker node. When a client submits a query to the server, the query optimizer uses a cost analysis to determine whether to execute a parallel or distributed query. A parallel query is broken into query fragments of predicates and data flow subtrees. A query fragment can be distributed only if the Sybase IQ engine supports parallel and distributed execution of the query operators contained in the fragment.

When a query is distributed, the leader node assigns query fragments to worker nodes. Worker nodes do not make decisions about query distribution. They simply execute the work assigned to them and return intermediate results from the worker servers. A query is likely to get distributed when it uses a large number of rows. For example, queries on star schemas may benefit greatly from distributed query processing.

If the query optimizer determines that a distributed query is unlikely to scale appropriately, or might even degrade performance, the query is not distributed and is executed on a single node in the multiplex.

### ❖ Running a distributed query

Before running distributed queries, increase the file size in user dbspace `iq_main`.

- 1 In Interactive SQL, open and execute:

**On UNIX** – `-$SYBASE/IQ-15_3/demo/dqpdata.sql`.

**On Windows** – `%ALLUSERSPROFILE%\SybaseIQ\demo\dqpdata.sql`.

This creates the Dimension table and the Fact table (10 million rows).

- 2 In Interactive SQL, open and execute:

**On UNIX** – `-$SYBASE/IQ-15_3/demo/dqpquery.sql`.

**On Windows** – `%ALLUSERSPROFILE%\SybaseIQ\demo\dqpquery.sql`.

This query fetches three rows and generates query plans as HTML files in your demo directory and in the write server's directory. The query also writes the paths into the `.iqmsg` file.

---

**Note** Issue a `ROLLBACK` command from Interactive SQL after the three rows return to force Sybase IQ to flush the query plans.

---

- 3 Check the multiplex node directories for query plan HTML files. The leader node's directory has the comprehensive query plan, and each work unit distributed has an HTML file generated by the node that received it. Double-click the leader node's query plan to open it in a Web browser.

**Figure 24: Query tree for distributed query**

The three vertical bars that connect the query operators in the graphic indicate distribution. For example, between the group by node and the join, the three bars indicate that this operation occurs over parallel threads and is distributed over multiple servers. The operation between the join and leaf #01 is also distributed.

## Configuring an ODBC Datasource on Windows

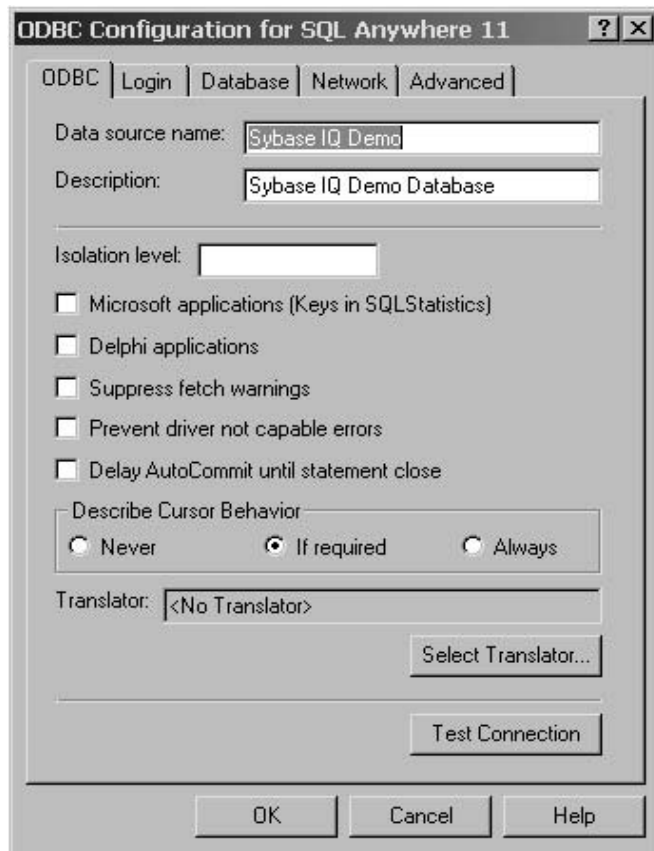
ODBC lets you use clients with the demo database.

### ❖ **Configuring an ODBC Datasource on Windows**

The following examples shows you how to set up an ODBC datasource name (DNS).

- 1 On the ODBC tab, specify the name of the ODBC datasource that the ODBC administrator can access.

**Figure 25: ODBC tab**





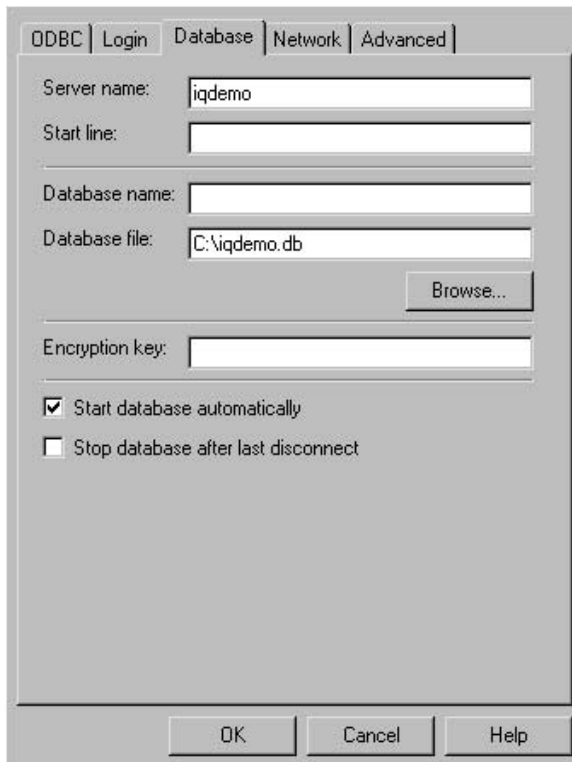
- 2 On the Login tab, specify the User ID and password associated with this connection to IQ. If you are using the IQ demo database, specify User ID **DBA** and Password **sql**.

**Figure 26: Login tab**



- 3 On the Database tab, specify the name of the server and database file. The server name is assigned when IQ starts with the -n parameter. The database file is the fully qualified location of the database file name associated with this connection to IQ.

**Figure 27: Database tab**



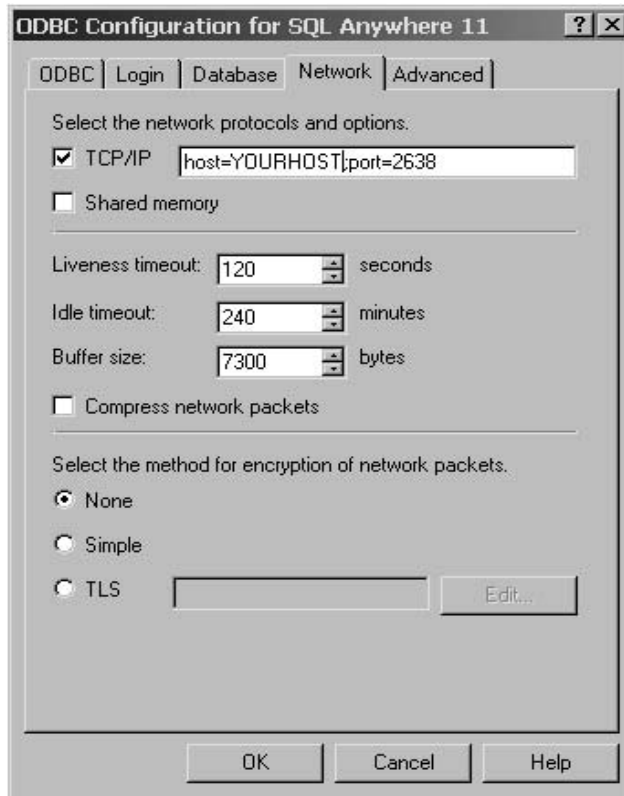
The image shows a screenshot of the ODBC configuration dialog box, specifically the Database tab. The dialog has five tabs: ODBC, Login, Database, Network, and Advanced. The Database tab is selected. The fields are as follows:

- Server name: iqdemo
- Start line: (empty)
- Database name: (empty)
- Database file: C:\iqdemo.db
- Encryption key: (empty)
- Start database automatically:
- Stop database after last disconnect:

At the bottom of the dialog are three buttons: OK, Cancel, and Help. A 'Browse...' button is located next to the Database file field.

- On the Network tab, specify the network protocols and options. The TCP/IP options you need are host (<yourhost>) and port (2638). The IQ server assigns the port option when it starts, and the host is the physical host on which the IQ server is running.

**Figure 28: Network tab**



## ODBC C Programming Example

This example shows a C implementation of ODBC that connects a client to a Sybase IQ database. You can find this example in the samples directory of the IQ installation.

```
//
*****

// Copyright (c) 2001-2009 iAnywhere Solutions, Inc.

// Portions copyright (c) 2001-2009 Sybase, Inc.

// All rights reserved. All unpublished rights reserved.

//
*****

/*
*****

This sample code is provided AS IS, without warranty or
liability of any kind. You may use, reproduce, modify
and distribute this sample code without limitation, on
the condition that you retain the foregoing copyright
notice and disclaimer as to the original iAnywhere code.

***** */

/* odbconnect.cpp : Defines the entry point for the
console application. */

    retcode = SQLAllocHandle( SQL_HANDLE_ENV,
SQL_NULL_HANDLE, &env );

    if (retcode == SQL_SUCCESS || retcode ==
SQL_SUCCESS_WITH_INFO) {
```

```

printf( "env allocated\n" );

/* Set the ODBC version environment attribute */

retcode = SQLSetEnvAttr( env,
SQL_ATTR_ODBC_VERSION, (void*)SQL_OV_ODBC3, 0);

retcode = SQLAllocHandle( SQL_HANDLE_DBC, env, &dbc
);

if (retcode == SQL_SUCCESS || retcode ==
SQL_SUCCESS_WITH_INFO) {

printf( "dbc allocated\n" );

retcode = SQLConnect( dbc,

(SQLCHAR*) "Sybase IQ Demo", SQL_NTS,

(SQLCHAR* ) "DBA", SQL_NTS,

(SQLCHAR*) "sql", SQL_NTS );

if (retcode == SQL_SUCCESS || retcode ==
SQL_SUCCESS_WITH_INFO) {

printf( "Connection successful\n" );

}
SQLDisconnect( dbc );
}
SQLFreeHandle( SQL_HANDLE_DBC, dbc );

}

SQLFreeHandle( SQL_HANDLE_ENV, env );

```

```
        return 0;  
    }
```