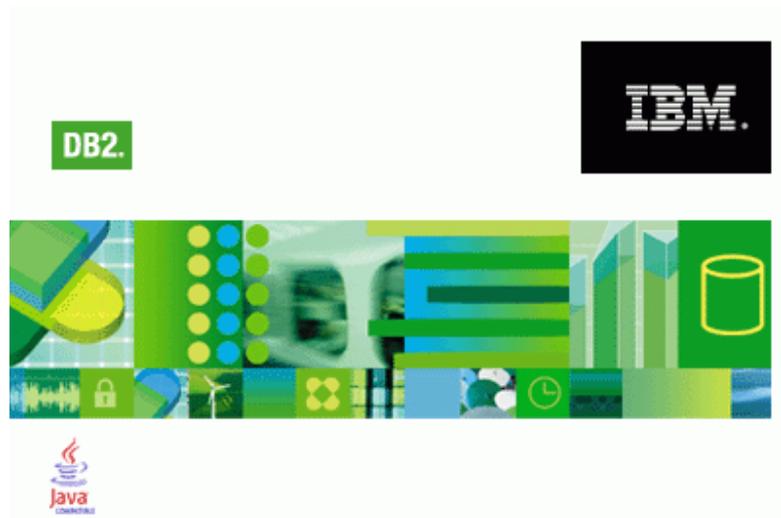


# DB2® Database pureScale Simulation Program Version 9.8 Installation and Reference Guide



George Baklarz, PhD  
DB2 Worldwide Sales Support  
DB2 Worldwide Technical Sales Support  
IBM Toronto Laboratory

## **DB2 pureScale Simulation Program Version 9.8 Usage Guide**

This document contains proprietary information of IBM. It is provided under a license agreement and is protected by copyright law. The information contained in this publication does not include any product warranties, and any statements provided in this manual should not be interpreted as such.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 2010. All rights reserved. US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

### **Credits**

Material in this document was prepared by the DB2 Worldwide Sale Support group.

## Legal Notices

This page may contain other proprietary notices and copyright information, the terms of which must be observed and followed.

INFORMATION IN THIS DEMONSTRATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THE ABOVE EXCLUSION MAY NOT APPLY TO YOU.

Information in this demonstration may contain technical inaccuracies or typographical errors. Information may be changed or updated without notice. IBM may also make improvements and/or changes in the products and/or the programs described in this information at any time without notice.

Information IBM publishes in this demonstration may contain references or cross references to IBM products, programs and services that are not announced or available in your country. Such references do not imply that IBM intends to announce such products, programs or services in your country. Consult your local IBM business contact for information regarding the products, programs and services which may be available to you.

IN NO EVENT WILL IBM BE LIABLE TO ANY PARTY OR ANY DIRECT, INDIRECT, SPECIAL OR OTHER CONSEQUENTIAL DAMAGES FOR ANY USE OF THIS DEMONSTRATION, INCLUDING, WITHOUT LIMITATION, ANY LOST PROFITS, BUSINESS INTERRUPTION, LOSS OF PROGRAMS OR OTHER DATA ON YOUR INFORMATION HANDLING SYSTEM OR OTHERWISE, EVEN IF WE ARE EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## Copyright and Trademark information

### Disclaimer and Trademarks

This presentation may contain other proprietary notices and copyright information, the terms of which must be observed and followed. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements, or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only. This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious, and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

### Trademarks

The following terms are trademarks or registered trademarks of other companies and have been used in at least one of the pages of the presentation:

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both: AIX, AS/400, DataJoiner, DataPropagator, DB2, DB2 Connect, DB2 Extenders, DB2 OLAP Server, DB2 Universal Database, Distributed Relational Database Architecture, DRDA, eServer, IBM, IMS, iSeries, MVS, Net.Data, OS/390, OS/400, PowerPC, pSeries, RS/6000, SQL/400, SQL/DS, Tivoli, VisualAge, VM/ESA, VSE/ESA, WebSphere, z/OS, zSeries

Microsoft, Windows, Windows NT, Windows XP, Windows Vista, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

© Copyright IBM Corporation 2010. All rights reserved.

## Table of Contents

Installation Requirements .....	1
Installing the pureScale Simulator Program.....	1
Installation File .....	1
Setup and Installation of the Program .....	1
DB2 pureScale Overview .....	7
What is DB2 pureScale? .....	7
Program Overview .....	9
Client, Member, GPFS and CF Object Creation.....	10
Client Configuration and Running a Simulation .....	15
Select Button .....	16
Update Button .....	17
Commit and Rollback .....	18
Locking Mode .....	18
Connection Type .....	21
Members .....	23
GPFS.....	24
pureScale Cluster Caching Facility .....	25
Playback and Recording.....	27
Loading Scripts and Playing Them .....	27
Recording and Saving Scripts .....	29
Pre-Loading buffers.....	30
Annotations .....	31
Options .....	33
Program Credits.....	37
Support .....	37
Legal Disclaimers .....	38

## Installation Requirements

This program is meant to be used for demonstrating the functionality of the DB2 pureScale 9.8 product. In order to use this program, you need the following configuration:

- Windows Operating System (Windows XP/Vista/7 )
- Suitable space (20M) for installation files

## Installing the pureScale Simulator Program

### Installation File

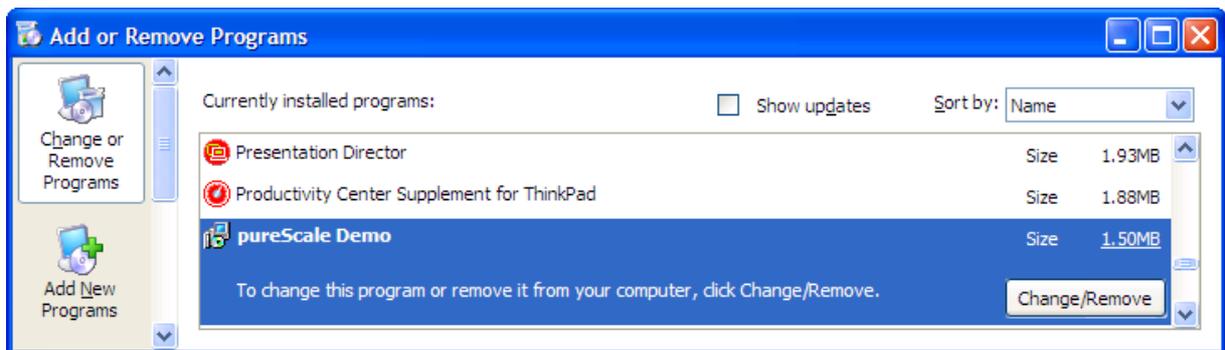
The pureScaleDemo installation program consists of one exe file that contains the executable along with all of the supporting files required to run the program. The file name is the same whether you downloaded this program from the internet or found it on a CD-ROM:

- pureScaleDemo.exe - Installation files for the pureScale program

Before beginning the installation, make sure that you have about 20M of disk space available.

### Setup and Installation of the Program

To install the simulator, just double-click on the pureScaleDemo.exe to start it. To uninstall the program, just run the program again, or go into the Add/Remove Program dialog in Windows and remove it from there.

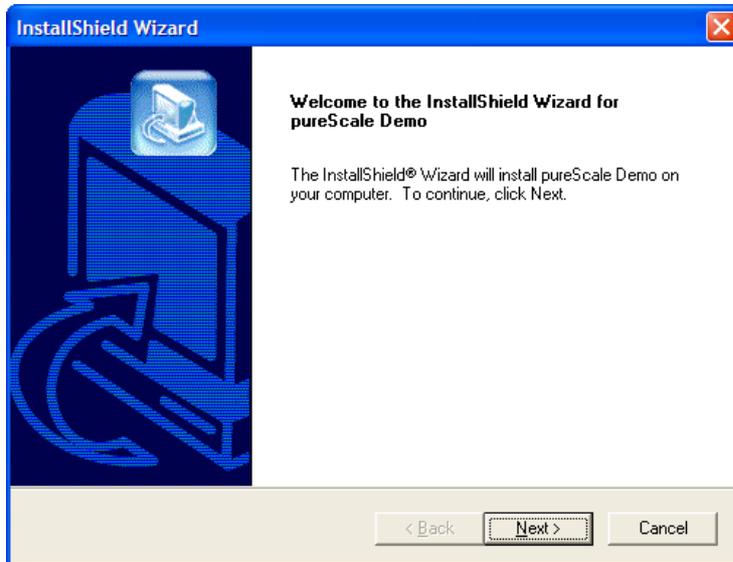


The next few pages explain the installation steps.

The first screen that is shown during installation is the DB2 pureScale graphic. During this step the program is unpacking the files that will be needed during installation.



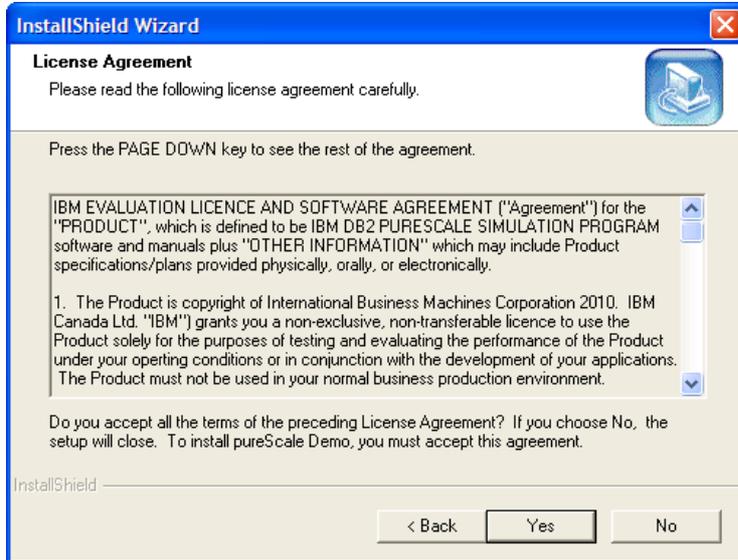
The first dialog that is displayed confirms that you are installing the pureScale Simulation demonstration.



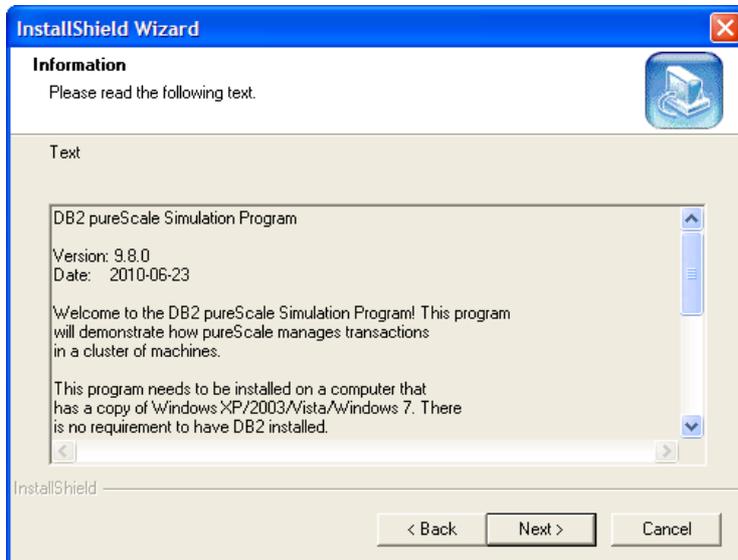
Click NEXT to continue.

## DB2 pureScale Simulation Program Version 9.8

The installation program will display the license agreement. You must accept the license agreement in order to install the program. A copy of this agreement can also be found in the installation directory.

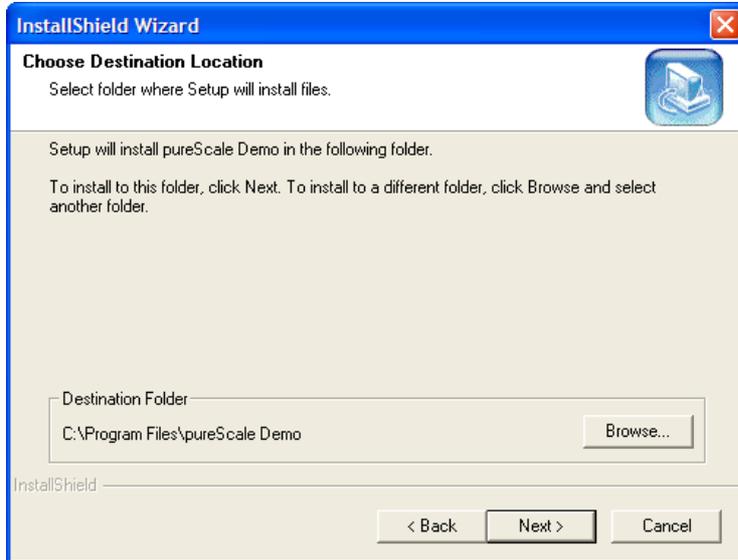


After clicking YES, the installation program will show what the current version of the pureScale simulation program is.

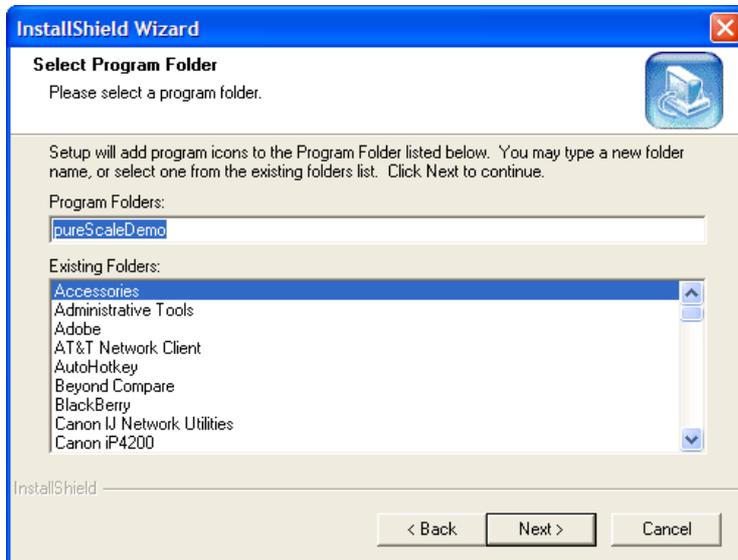


Clicking NEXT will continue with the installation.

The program will be installed in the default Windows installation directory within a directory called pureScale Demo. You can override this location by using the BROWSE button to select a new location. Pressing NEXT will use the default value.

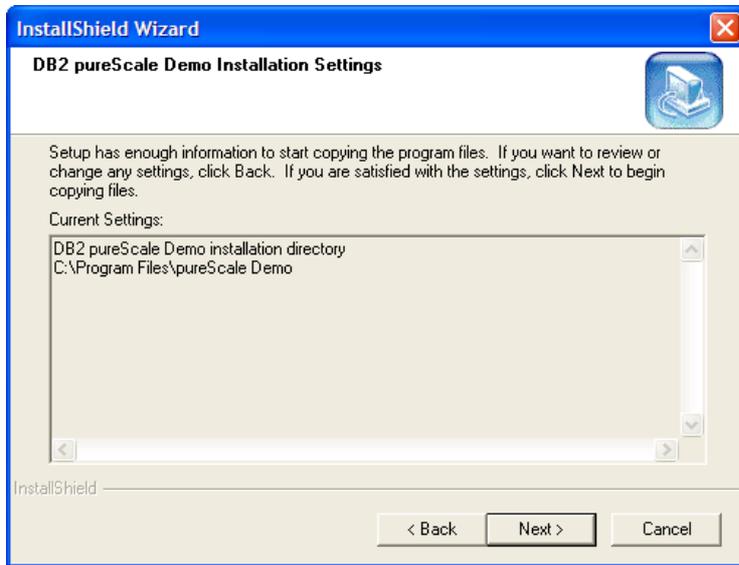


A program folder called pureScaleDemo will be created, but you can install it in another location.

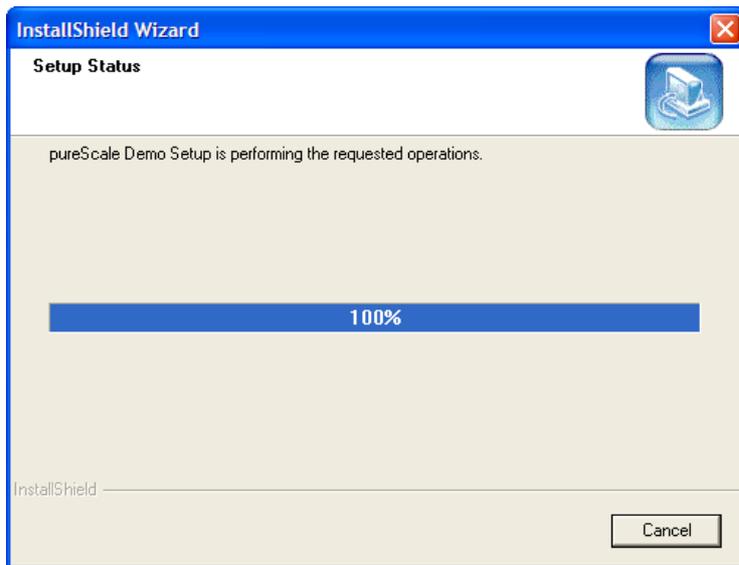


Press NEXT to continue on to the review step.

This final panel will let you review your choices of installation directory and folder name.

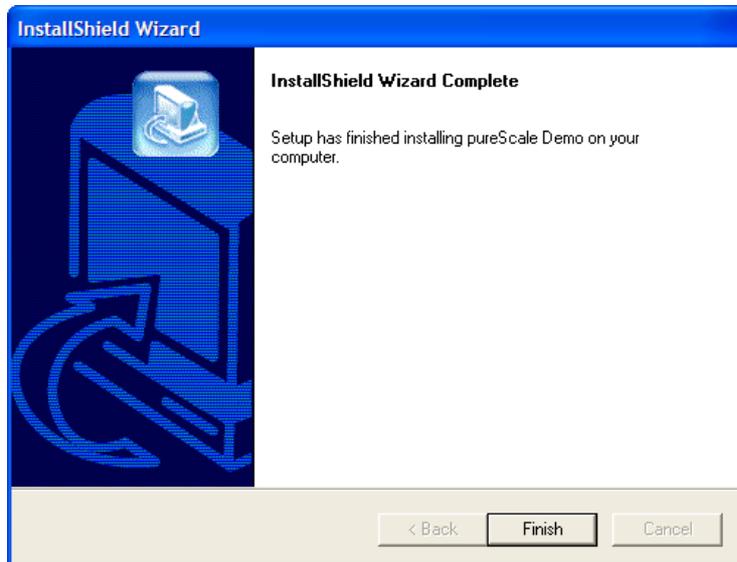


If you are satisfied with the installation settings, press NEXT to begin product installation.



A progress bar will be displayed while the demonstration is installed.

Once the installation is complete you will see this final panel. Click FINISH to close the dialog.



You can now start the program by clicking on the pureScale Simulator icon on the desktop.



## DB2 pureScale Overview

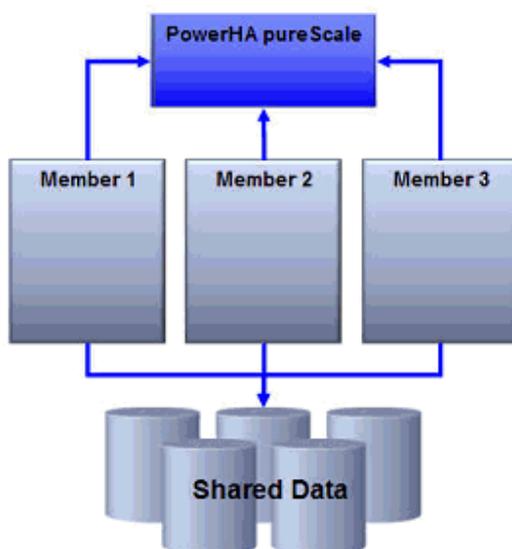
The DB2 pureScale simulator allows a user to explore some of the concepts that are associated with a pureScale system. The user can define simple configurations and try selects and updates against members using a variety of connection and locking strategies. Member and clients can be turned off to show the effects of the outage on the configuration and the resulting recovery that pureScale will automate.

While the simulator has been designed to show the concepts of running a DB2 pureScale system, it does not show all of the possible system actions and communications between the clients, members, and cluster adapters. A production pureScale system will have many additional optimizations that will minimize much of the communication that you may see as part of the simulation. However, the high-level messaging is essentially the same so you will get a sense of how the system works.

## What is DB2 pureScale?

DB2 pureScale is a new optional DB2 feature that allows you to scale out your database on a set of servers in an “active-active” configuration delivering high levels of both availability and scalability. In this configuration the copy of DB2 running on each host (or server) can simultaneously access the same data for both read and write. A collection of one or more DB2 servers that share DB2 data is called a data sharing group. A DB2 server that belongs to a data sharing group is a member of that group. All members of a data sharing group share the same database. Currently, the maximum number of members in a data sharing group is 128. In addition to the DB2 members, there is also the pureScale™ Cluster Caching Facility component which provides centralized lock management as well as a centralized global cache for data pages (known as the group buffer pool).

Each member in the data sharing group can interact directly with the pureScale Cluster Caching Facility component, as shown below, through a very efficient InfiniBand™ network which means each member has point-to-point connectivity to the centralized locking and caching facility.



In order to reduce communication between nodes in the cluster for lock management and global caching services, DB2 pureScale uses the pureScale Cluster Caching Facility (here-in referred to simply as the CF) along with RDMA technology to deliver transparent application scalability.

RDMA allows each member in the cluster to directly access memory in the CF and for the CF to directly access the memory of each member. For example, assume that a member in a cluster (Member 1) wants to read a data page that is not in its local buffer pool. DB2 assigns an agent (or thread) to perform this transaction; the agent then uses RDMA to directly write into the memory of the CF to indicate that it has interest in a given page. If the page that Member 1 wants to read is already in the CF's global centralized buffer pool, the CF will push that page directly into the memory of Member 1 instead of having the agent on that member perform the I/O operation to read it from disk. The use of RDMA allows Member 1's agent to simply make a memcopy (memory copy) call to a remote server without the need for costly process-to-process communication calls, processor interrupts, IP stack calls, and so on. Quite simply, pureScale allows a member's agent to perform what appears to be a local memory copy operation when in fact the target is the memory address of a remote machine.

These lightweight remote memory calls, along with a centralized buffer pool and lock management facilities, means that an application does not have to connect to the member where the data already resides. It is just as efficient for any member in the cluster to receive a data page from the global buffer pool regardless of the size of the cluster. Most RDMA calls are so fast that the DB2 process making the call does not need to yield the CPU while waiting for the response from the CF and does not have to be rescheduled to complete the task. For example, to notify the CF that a row is about to be updated (and therefore an X lock is required) a member's agent performs a Set Lock State (SLS) request by writing the lock information directly into memory on the CF. The CF confirms that there are no other members in the cluster that already have this row X locked and will directly write into the requesting member's memory to grant the lock. The entire round trip for this SLS can take as little as 15 microseconds and therefore the agent does not need to yield the CPU.

The agent can continue to be productive rather than waiting on an IP interrupt (avoiding unnecessary context switches) as is the case with other scale-out architectures. If for a specific operation, such as long running batch transactions, it would make more sense for the DB2 agent to yield the CPU, DB2 will make an autonomic decision to dynamically yield the CPU.

Another important DB2 scalability feature that goes hand-in-hand with transparent application scaling includes the DB2 pureScale built-in load balancing across the members of a cluster. Applications do not need to be cluster aware to take advantage of the load balancing. The same client-side drivers that DB2 for z/OS data sharing customers use today work with DB2 pureScale for cluster load balancing.

DB2 pureScale takes availability to a new level when compared to other offerings available on distributed platforms. DB2 pureScale provides full access to every page of data that does not need recovery and is aware at all times of which specific pages need recovery without having to perform a single I/O operation. This is yet another important innovation made possible through the unique capabilities of a centralized CF.

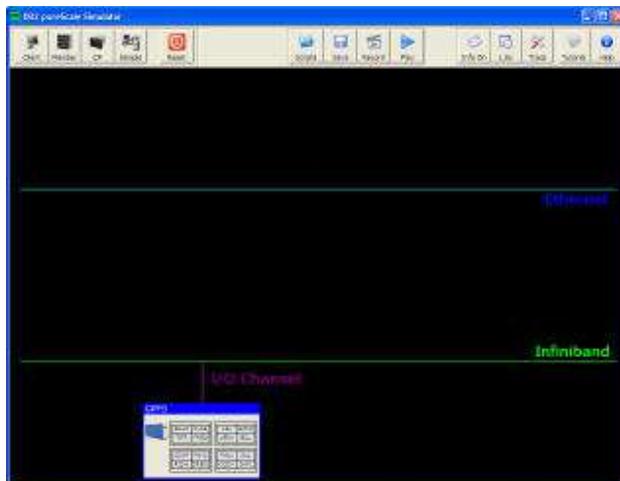
## Program Overview

The pureScale simulator is distributed as an exe file and can be installed in any directory on a Windows-based machine. There is no requirement to have a DB2 system installed in order to run this simulator.

When the program first starts it will display a legal disclaimer which can be read and then dismissed by closing the window.



Once you've closed the legal disclaimer you will be presented with the following screen:



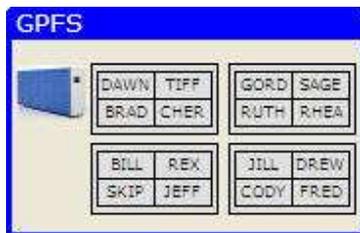
The top row of the pureScale simulator contains a number of buttons that control the behavior of the system. The buttons can be divided into three groups:

- pureScale objects for building a system to simulate
- record and playback buttons for running pre-configured systems
- options for setting the behavior of the simulator

Below the screen you will see the network (Ethernet) for connecting clients to members, an infiniband channel for connecting the DB2 members to the pureScale Cluster Caching Facility, and an I/O channel to the file system. Note that the file system does not run over infiniband. The I/O system was placed off the infiniband channel to reduce the number of connections required in the simulator.

By default, an empty system is shown whenever the simulator starts. A file system is already created by the simulator and will contain some random names in 16 slots or rows in the table. The file system used by DB2 is GPFS, or the IBM General Parallel File System™ (GPFS™). GPFS is a high-performance enterprise file management that provides online storage management, scalable access, and integrated information lifecycle management tools capable of managing petabytes of data and billions of files.

The GPFS file system is shown the bottom left-hand corner of the display connect via an I/O channel to the infiniband network. Note that infiniband is not used for the actual I/O but is merely used for simulating the I/O channel.



You can customize the values that are shown in the file system by editing the `_gpfs.cf` file found in the system directory. For more details on how this is done, please refer to the section on the GPFS.

## Client, Member, GPFS and CF Object Creation

A pureScale system cannot run unless you have at least one client, one member, the GPFS file system, and one pureScale Cluster Caching Facility (referred to as the CF from now on).

To create these objects you will need to use the Objects toolbar found at the top left of the simulator program:



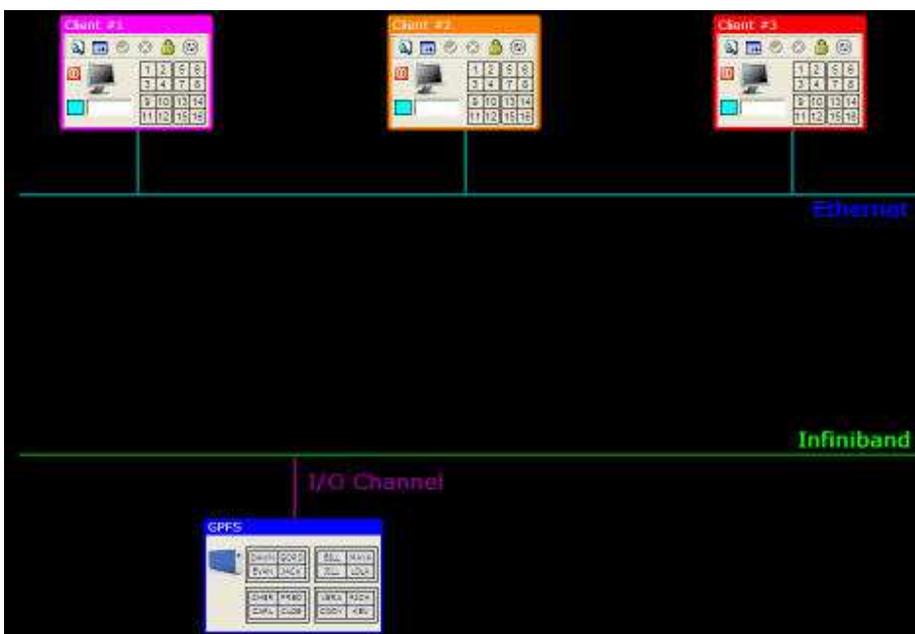
From left to right, these buttons represent the following objects. The first button is used for creating a client on the screen.



When the client button is pressed, one client will be placed onto the simulator screen. The client will be connected to the network (Ethernet).



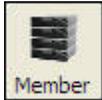
Up to three clients can be connected to the Ethernet and each one will have a unique number and color. The following image shows three clients on the screen. Client #1 is purple, client #2 is orange, and client #3 is red.



The client color will be used in the simulation to mark which client is holding an update lock on a particular row. The CF will be described later, but here is an example of the CF with locks being held by various clients:



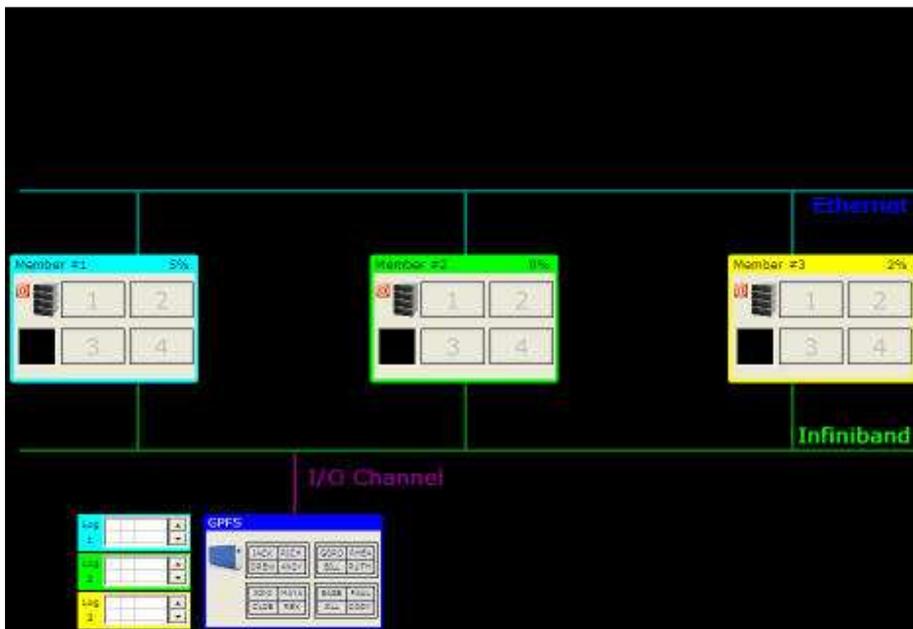
A pureScale system must have at least one member, so the member button is used to create a DB2 server (member) in the simulator.



Pressing the member button will place a member on the screen, connected to both the Ethernet, for client communication, as well as the infiniband network, for communicating with the CF.



In similar fashion to the clients, each member has a unique number and color as shown in the following image.

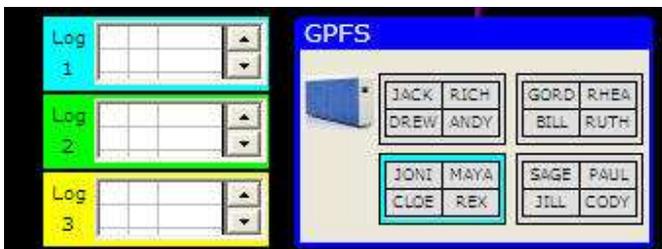


Member #1 is light blue, member #2 is light green, and member #3 is yellow. These colors are used in the CF to highlight which member has read a page into its local buffer pool. In the CF, read "interest" in a page is marked by small squares below the lock icon. The color of the square indicates which member has read a particular page.



In addition to creating a member on the screen, the member button will also create a log file associated with the member. This log file is placed to the left of the GPFS file system to indicate that these files are found in the GPFS, but are separate for each member.

The following image shows three logs files created for three members. A log file will only be created if the member exists.



The final object that can be created is the pureScale Cluster Caching Facility, or CF.



DB2 pureScale uses the pureScale Cluster Caching Facility along with RDMA technology to deliver transparent application scalability. The CF provides centralized buffer pool and lock management facilities, which means that an application does not have to connect to the member where the data already resides. It is just as efficient for any member in the cluster to receive a data page from the global buffer pool regardless of the size of the cluster.

The CF is placed on the bottom right-hand side of the screen.



Only one CF exists in the simulator, but a pureScale system can have both a primary and a secondary CF installed. For true high-availability operation, it is recommended that two CF's exist in the event that one of them fails.

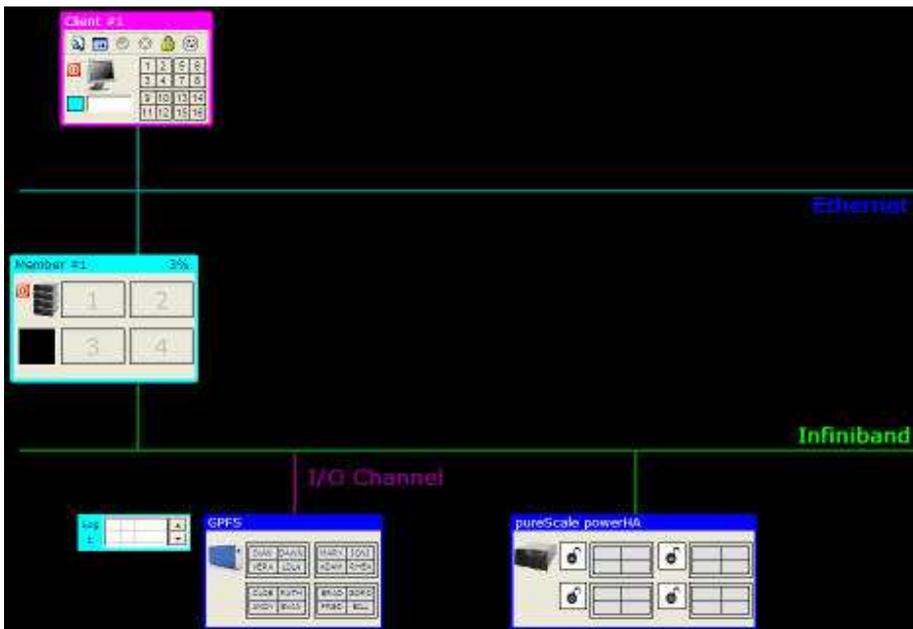
A system can be built by adding clients, members, and CF. In fact; you can start with a simple configuration and add members or clients to increase the complexity of the system. Once you hit the maximum number of clients or members, the simulator will not allow you add any more. Note: Clients can be killed with the power off button that is found on them. You can then re-add the client later if you want. However, killing a DB2 member will cause an automatic recovery if there are any outstanding transactions on that member. If this is the case, the member will be restarted automatically. If there are no outstanding transactions, the member will be turned off.

Stopping clients or members can only take place when any activity in the system (like a select or update) has completed.

To quickly build a one client, one member system, click on the Sample button.



This button will generate the following configuration.



If you want to delete your configuration, or start from the beginning, click on the Reset button.

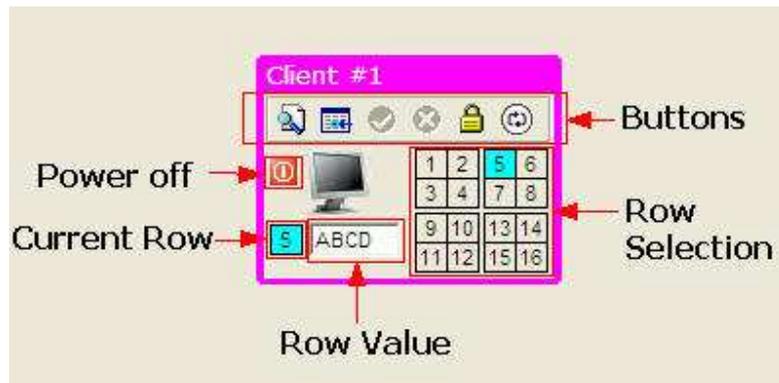


The reset button can be pressed at any time when using the pureScale simulator. If you are running your own script, a pre-built one, or in the middle of a recording, hitting the Reset

button will stop all activity and create a clean configuration. Note that hitting reset will delete any recording that you may have done and destroy the configuration you had.

## Client Configuration and Running a Simulation

Before discussing some of the recording, playback, and options available, it's best to look at how the simulator works. The main interface to running a simulation is through the client. Examining the client you will see that there are a number of buttons and values in it.



There are five sections in the client:

- Buttons - these control the execution of the simulation and client
- Row Selected - used for selecting one of sixteen rows to process
- Power Off - used to kill a client to test error recovery
- Current Row - the row that was selected or updated in a transaction
- Row Value - the value of a row returned during a select, or the value that the row was updated to

The buttons are the key component of the client since they control how the simulation runs.



Some of these buttons may be grayed out depending on what the client is currently doing. However, the two buttons that are used to control the actual SQL that the client will run are always available. The buttons from left to right are:

- Select - used to select a row from the table
- Update - used to update the value of a particular row
- Commit - only available when an update has been done
- Rollback - only available when an update has been done

- Locking mode - changes the locking behavior for a select only
- Connect mode - determines how the client will connect

Each one of these buttons is described in detail below.

### Select Button

The Select button is used to select a row from the table. There are sixteen rows in the table and each one contains a first name. These names are randomly loaded into the DB2 table that resides on the GPFS file system.

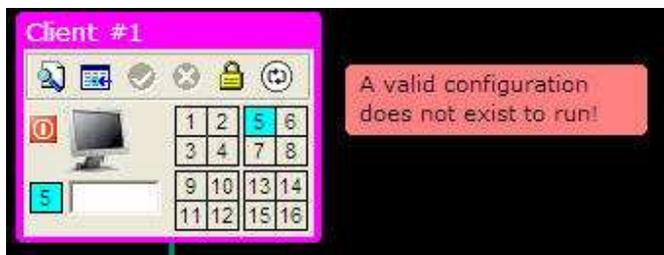
The client can only select one row at a time (no SELECT \* ... available here!). To indicate which row you want to select, click on one of the sixteen row numbers found below the buttons.

1	2	5	6
3	4	7	8
9	10	13	14
11	12	15	16

When you select a particular row, the square with the row number will turn blue, and the blue box below the monitor will be updated to contain the number. For instance, selecting row 5 with this client will result in the following display.



Once a row has been selected, the SELECT button can be pressed and the simulation will start. If you do not have a valid configuration, an error message will be displayed beside the client.



Once the select transaction has completed, the row value in the client will be updated to contain the value of the row from the table.



### Update Button

The update button is found beside the select button and works in a similar manner. The client can only update one row during a "click" but many rows can be updated before a commit or rollback is done. In other words, the client could potentially update all 16 rows before requiring an update or a commit to occur.

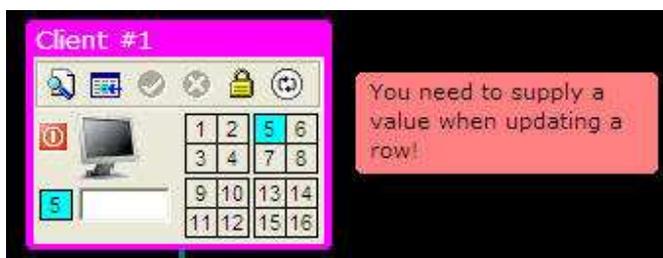
Selecting a row for update is identical to retrieving a row. You first need to select the row number and the box containing the row number will turn blue. Before issuing an update, you must place a value beside the row number box below the monitor.

The following image shows the client replacing the value in row 5 with the value ABCD.



You can also click on the row number box beside the row value to generate a random name. This is more convenient if you want to use the mouse to run the simulation rather than using the keyboard to enter a value.

Once you've entered a value for the row you can press the update button. If you do not supply a value for the row, an error message will be generated.



Once an update has successfully completed, the row box in the grid will turn yellow to indicate that this row has been updated.



The colors of the boxes will reset to gray when all updates are either committed or rolled back.

### Commit and Rollback

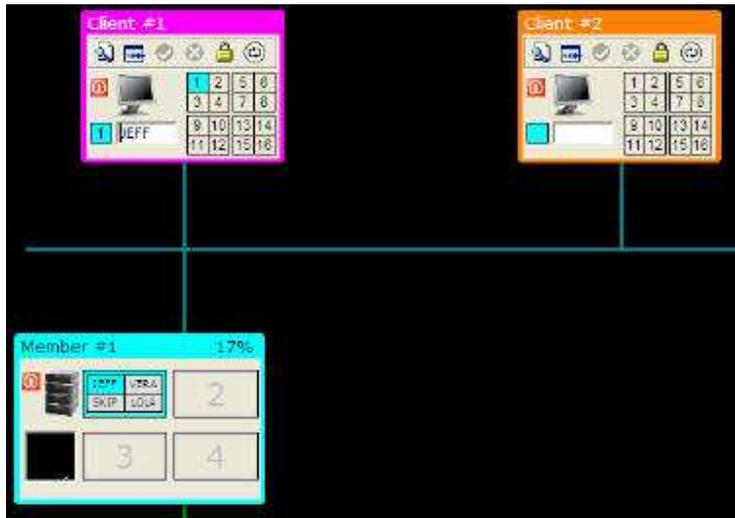
The Commit and Rollback buttons are only available when you have successfully updated a row. Prior to this, the commit and rollback buttons are grayed out. Once we update a row we will implicitly create a transaction. The implications of this are that any subsequent SELECT or UPDATE that is run from this client can only connect to the member that the original update went to. In other words, all transactions within this unit of work will have affinity (or be restricted to) to the member that the transaction started on.

Selecting the green arrow (Commit) will cause the updates to be committed on the CF and all locks released. Selecting the red "X" (Rollback) will cause the updates to be rolled back and no changes will have been placed into the table.

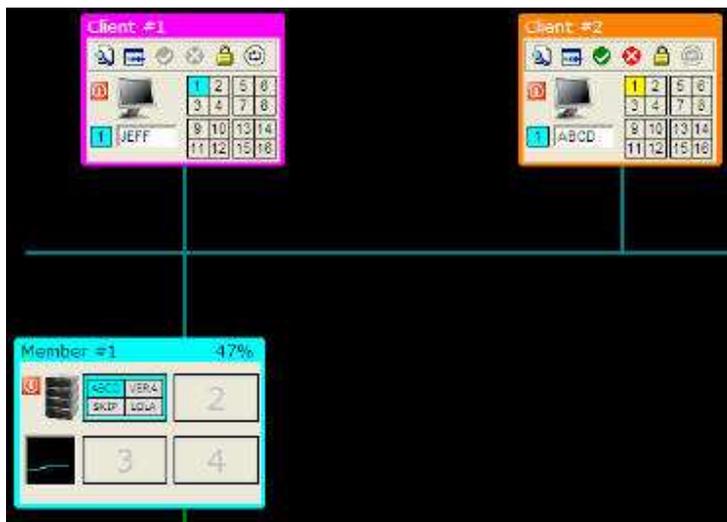
### Locking Mode

The locking mode button lets you change how SELECTs (not UPDATES!) are handled by the member. By default, the client will request that currently committed semantics be used by the client. This will (in most cases) give the client the last properly committed value for a row. For instance, with the normal locking mode (closed lock) any select statement to a member will only read committed information. Normally if there are no locks on a row you would receive the current data that is in the local buffer pool or what is on disk.

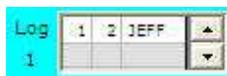
The image below shows that client #1 has retrieved the contents of row #1 (JEFF).



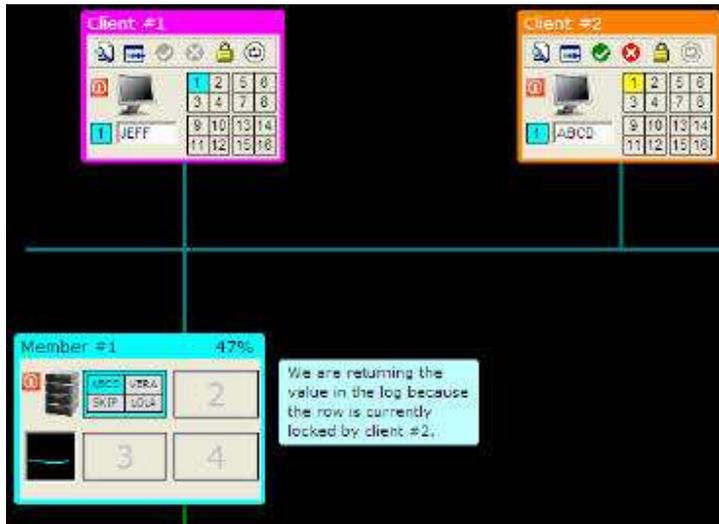
The next image shows client #2 updating the value of row #1 to the value ABCD.



The value in the buffer has changed to ABCD for row #1. Note also that the log file will now contain the original value of the row before the update.



When client #1 requests the row again, it will return the value of JEFF from the log file rather than the value from the buffer. The reason is that with currently committed semantics, we get the last known committed value of the row from the log, rather than the current "dirty" data in the buffer. Note that the row is not committed to disk or the CF until the client does a COMMIT.



If we want to get the value of the row currently in the buffer, we would change the locking mode to UR (Uncommitted Read). Clicking on the lock icon will change it to an open lock.

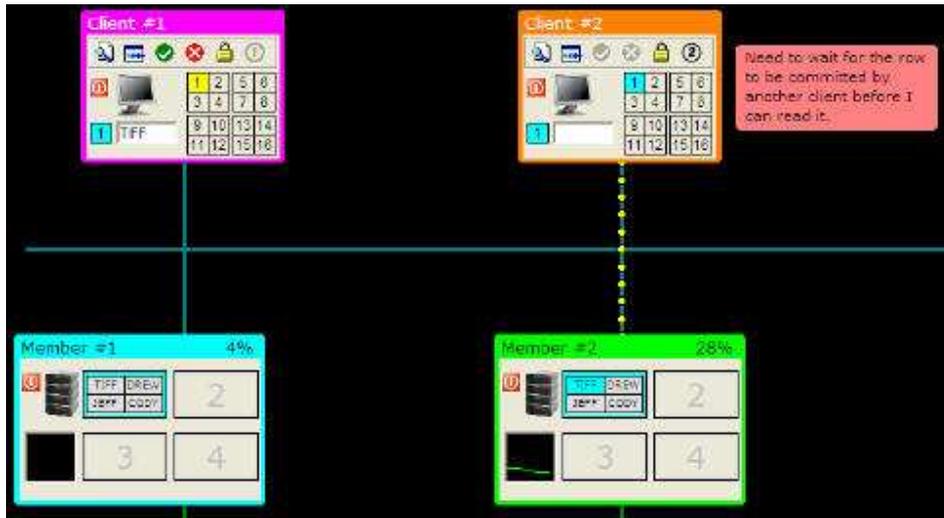


Now issuing a SELECT statement will result in the current data on the page being returned to the client.



Note that the client that initiated an update against a row will always see the current contents of the row, regardless of the locking model. Since the client did the update, they will always see the change that they made, rather than the previous, committed version. Clients that attempt to access a row that is currently locked on a different member will always wait for that update to complete (when using currently committed locking). The reason for this is that a member cannot access the log of another member (go "off" member) to get the previous value of the row.

For instance, the following image shows an updated row in member #1 (row 1 = TIFF) and member #2 requesting the same row via member #2. The client (#2) will have to wait for the update to complete before viewing the contents of the row.



Client #2 could also change its locking model to UR and it would then return the current value in its local buffer.

### Connection Type

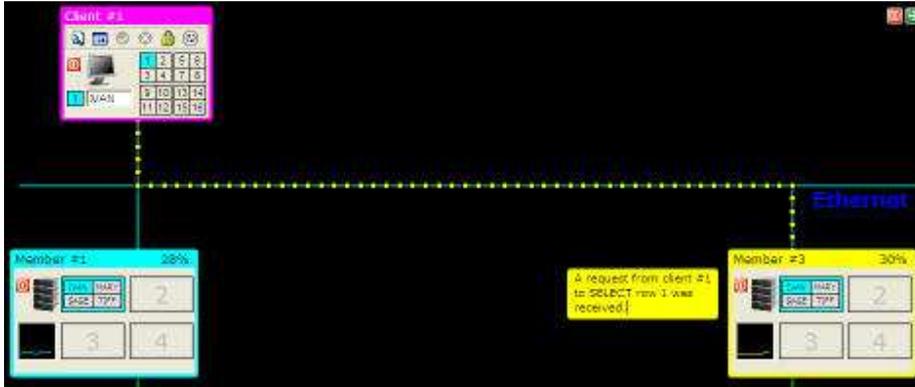
In order for a select or update to take place, a connection must first be made to a DB2 member. In a pureScale system, a connection to a member is done when the client first connects; however, the connection can move to another member during subsequent CONNECT requests or at a transaction boundary.

In the pureScale simulator, a new connection occurs whenever a SELECT takes place. The same is true when an UPDATE occurs, except that the client will NOT connect to a different member (for any subsequent SELECTs or UPDATEs) until a COMMIT or ROLLBACK occurs. The connection type box will turn gray when an UPDATE occurs, so all subsequent SELECTs and UPDATEs will continue to go to the same member.

Whenever a transaction is started from a client, it will connect to one of the DB2 members that is available. Which member it connects to is based on the Connection Type icon found at the far right of the buttons at the top of the client.



By default the connection is set to round-robin. This means that at the start of any transaction the client will connect to the next member. For instance, the first connection will always be to member 1. The second connection will be to member 2, and so on and eventually returning back to member 1. If a member is not available, the connection will go to the next available member. The following image shows how a transaction will go to member 3 since member 2 is missing.



Clicking on the connection button will make it change to a small graph (think CPU performance). This mode is called workload balanced.



When running with workload balancing, the connection will go to the least-used member. At the top of each member is the current load of the machine.



The load of the member over time is shown as a small graph underneath the cpu image.



When the client attempts a connection, it will look to see which is the least-busy member is and connect there. The loads on the systems are random and change as the simulation runs.

The next three connection options are based on "affinity" or connection to a specific server. Clicking on the connection button will now change it from cpu (or workload balanced) to connecting directly to member 1, 2, or 3 as shown in the following image.

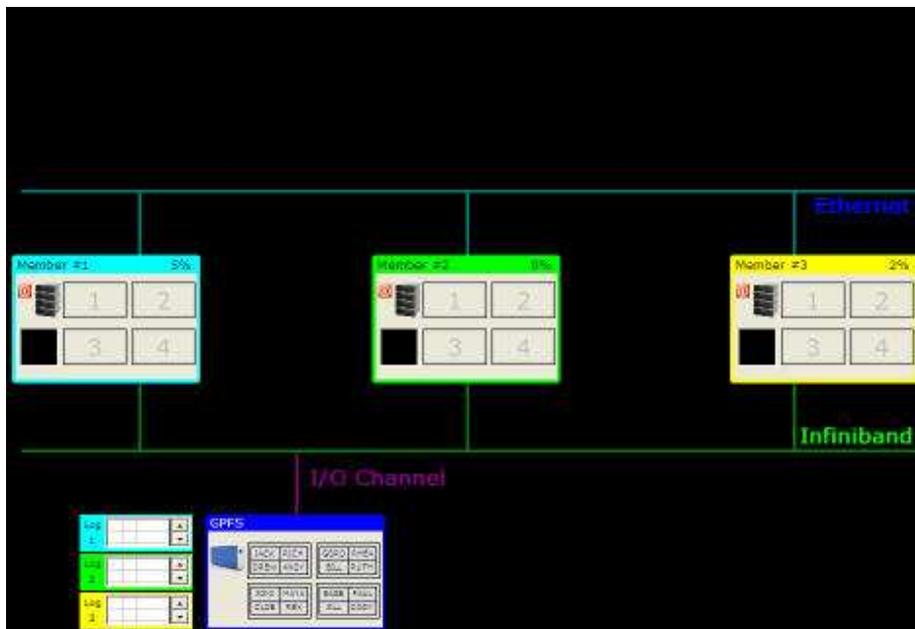


When using affinity-based connections, the client will not be able to run transactions if their affinity server is not available.

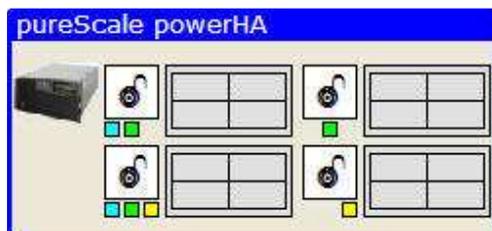


## Members

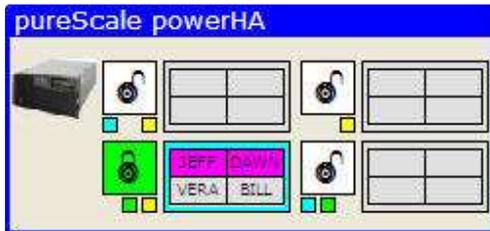
DB2 pureScale consists of members (DB2 Instances) that run on a physical machine or an LPAR. The simulator allows up to three members to be defined in the system. Each member has a unique number and color as shown in the following image.



Member #1 is light blue, member #2 is light green, and member #3 is yellow. These colors are used in the CF to highlight which member has read a page into its local buffer pool. In the CF, read "interest" in a page is marked by small squares below the lock icon. The color of the square indicates which member has read the particular page.



If a member has a page latch during an update statement, the lock symbol in the CF will change to the color of the member. For instance, if member #2 has a page latch on page 3, the CF will look similar to the following.



The member contains a shutdown button that can be used to cause a hardware or software outage. If a member is currently involved in an UPDATE transaction, pressing the power off button will cause a recovery to occur. Any in-flight transactions will be rolled back and the server will be restarted.

If the member does not have any outstanding transactions, pressing the power off button will shut it down, similar to placing it in maintenance mode. So if you want to remove a member from a configuration, make sure that there are no update transactions pending through that member.

## GPFS

The GPFS or General Parallel File System, is used by DB2 to store the database, logs, and control files.

The GPFS is created for you whenever you start up the simulator, or if you reset the system. The GPFS is immediately filled with 4-character names - a unique value for all sixteen rows found in the table.

Every time that you reset or start the system, the names will change. If you run a simulation multiple times you will find that the data in the rows will change.

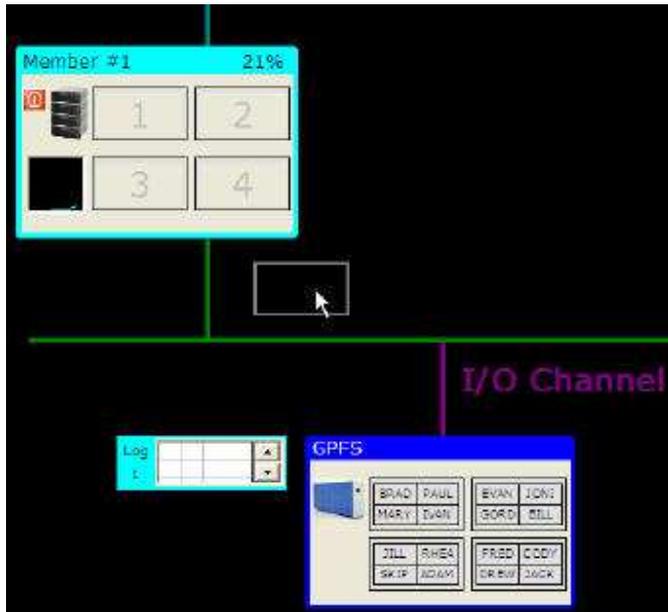
If you want to change the names that are used by the simulator, you need to edit the file called `_gpfs.psc` found in the system directory. This file has a line similar to the following.

```
names = {GORD,BILL,JILL,CODY,...}
```

Replace the names with up to 32 different values, and the simulator will randomly select 16 names to fill up the table with.

Before you start running a simulation, you can drag pages from the GPFS to a member or the CF to pre-populate them with data. You can only move a page from the GPFS to the same page slot in a member (or CF). For instance, page 1 from the GPFS must move to page 1 of a member.

To move a page, click on a page in the CF and then while holding the LEFT mouse button down, drag the page to a member or CF.

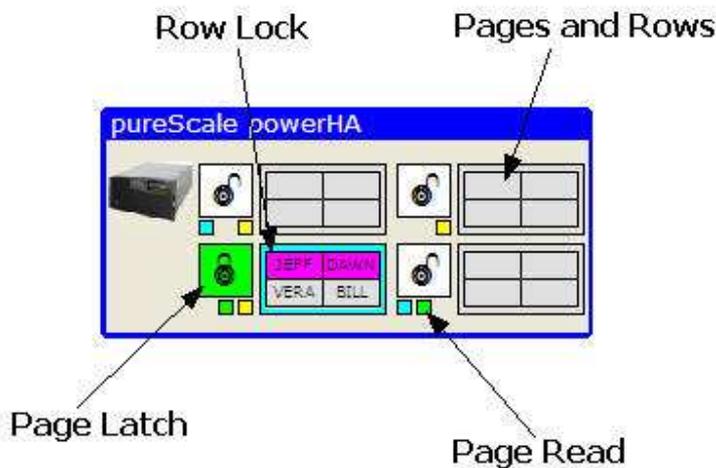


When you release the mouse over a page in a member or CF, the contents of that page will be updated with the data from the GPFS.



### *pureScale Cluster Caching Facility*

The pureScale Cluster Caching Facility, or CF, is used to control access to pages and locking across all of the members. There can be up to two CF's in a pureScale system, but the simulator only uses one. You must configure a CF before the simulator will run.



Aside from the pages and rows of data, the CF contains status information about the way the pureScale system is running.

- Row locks
- Pages currently in members buffers
- Pages that are being updated

### Row Locks

Whenever a client requests a lock on a row for update, the CF will highlight the row with the color of the client. In this example two rows are currently locked by client #1. When a commit or rollback occurs, the lock will be released and the background color of the row will turn gray.

### Page Read

When a page is read into the buffer of a member, the CF will turn on a small block under the page with the color of the member. For instance, the first page (top left) has two blocks on - blue and yellow. This means that both member 1 and member 3 have read the page into their buffers. This information is important to the CF since it needs to track who has a page in memory in the event that the page needs to be invalidated because of an update.

### Page Latch

A page latch is used to control updates to a page. A latch is not the same as a lock since it is not kept for the duration of an update. The latch is only in place to guarantee that the page gets updated properly before moving back to the CF and subsequently to another member. In the example, page 3 is currently in the process of being updated by member #2 (green). If another member needs to update this page, the CF will request this page back from member #2 (and makes sure that member #2 has finished updating any data on the page) before giving access to the other member.

## Playback and Recording

The pureScale simulator comes with a number of "pre-built" simulations as well as the ability for you to create your own scenarios. The middle section of buttons at the top of the pureScale screen contains the playback and recording buttons.



From left to right the buttons are:

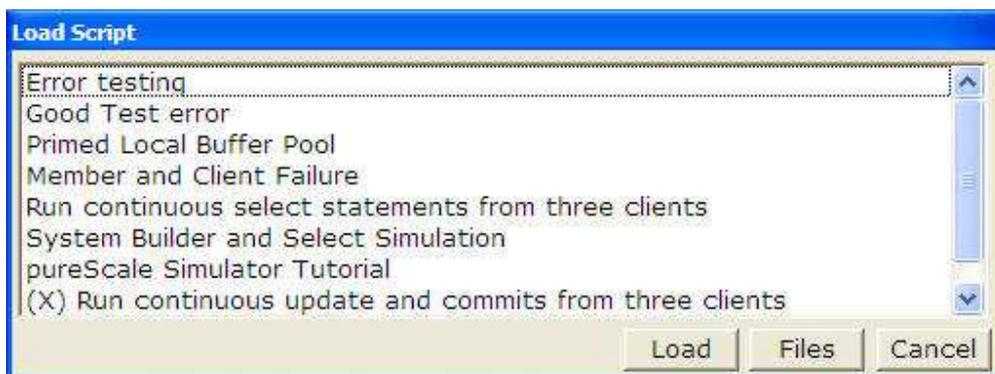
- Scripts - Pressing this button will give you the list of pre-defined scripts and any that you may have created.
- Save - If you have done a recording session, pressing the Save button will let you store it for future use.
- Record - Pressing Record will place the simulator into record mode where it will track every button that you click.
- Play - When you load or record a script, pressing the Play button will play it back.

### Loading Scripts and Playing Them

The Scripts button is used to load a pre-defined script into the simulator.

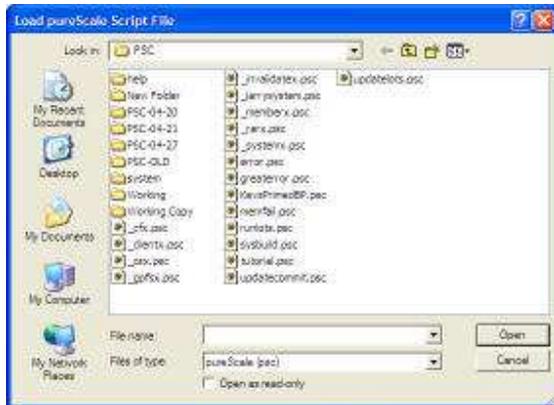


When this button is pressed, the simulator will display a list of pre-defined and saved simulations.

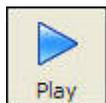


To load a file into the simulator, either double-click on the description of the simulation or select the description by clicking on it once and then hitting the load button.

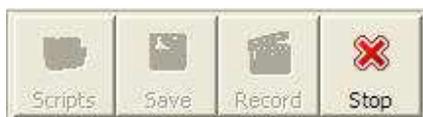
If you click on the FILES button, it will give you a list of the files in the directory that the program is running in.



Select a file from the list and then click on the open button. Whichever technique you use, the file will be loaded into the simulator, but it will not begin running until you hit the play button.



The play button will start the simulation that you loaded and it will disable all of the recording buttons and replace the play button with the stop button.



You can press the STOP button at any time to terminate the simulation. Note that while the simulation is running, you will see the following two buttons on the far right side of the screen.



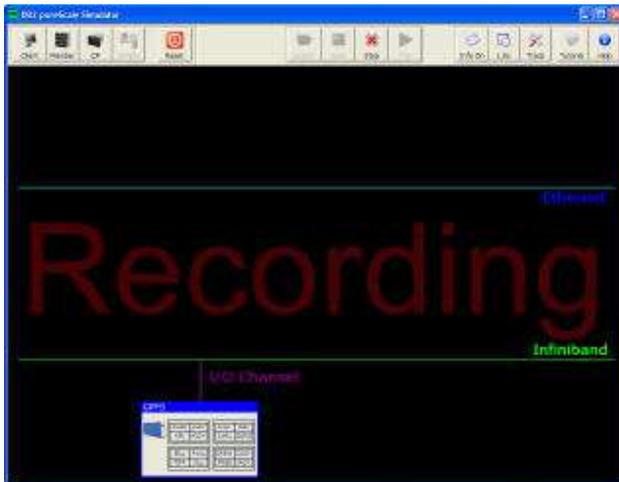
The section on running a simulation mentioned how clicking anywhere on the black network background would advance the simulation to the next step or message. The green button in this sidebar is used to make the simulation close the current message and move to the next step.

Similarly, the red stop button is used to complete the current step - not stop the entire simulation. The stop button that is shown on the top menu bar is used to completely stop a simulation. A subsequent press of the PLAY button would start the simulation from the beginning. When you press this stop button, all it does is complete the current transaction (without animation or messages). So for instance, if you were half way through an UPDATE statement, the STOP button would complete the UPDATE (with all locking and buffer changes, etc...) so that the database is consistent. Then the next SELECT or UPDATE can take place.

This is useful for skipping a number of steps that you don't want to see. However, remember that this is not the same as stopping the entire simulation.

### Recording and Saving Scripts

If you want to record a particular scenario to demonstrate to others, you can do it with the record button. When the record button is pressed, any current configuration found on the screen is deleted and you are left with a blank configuration (except for the GPFS). The background will have the word RECORDING set in red.



Once recording has been started, the buttons in the recording section will turn off, except for the STOP recording button.



The following keystrokes will be recorded in this mode:

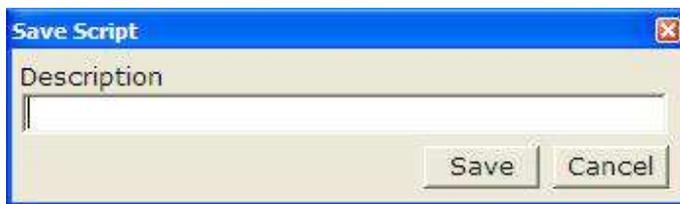
- Create Client
- Create Member
- Create CF
- Drop / Kill Client
- Drop / Kill Member
- Select Row
- Update Row

- Change locking mode, connection type
- Pre-loading buffers

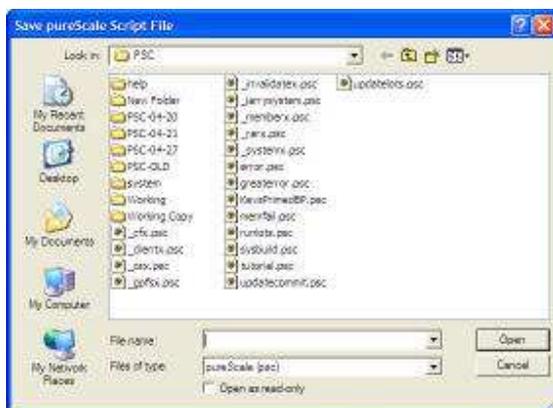
The selects and updates that you run as part of the simulation will be captured (i.e. the actual workload that you run). Once you are done with the simulation you can press the STOP button and then either playback the simulation (with the PLAY button), or press the SAVE button to file your work.



When you press the SAVE button, the simulator will first ask you to give a description of the simulation you just created. This is the same description that will be placed in the FILE menu when you request all the pre-defined and saved scripts on disk.



After pressing SAVE, a file menu will be displayed where you will enter the actual filename that you want to assign to the simulation.



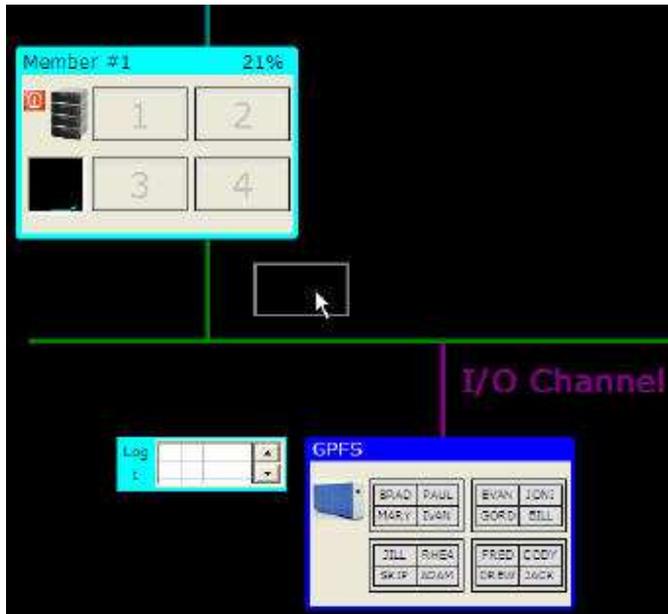
By default all simulator files have an extension of ".psc". There are some special files found in the /system directory that are used by the simulator to run the examples. These files all start with the underscore "\_" and are not meant to be changed. Simulations that are saved are normally placed in the root directory where the simulator program is found.

### Pre-Loading buffers

A handy shortcut is available that will let you pre-load the buffers of the members or CF before running any simulations. Before you start running your simulation, you can drag pages from the GPFS to a member or the CF. You can only move a page from the GPFS to the same

page slot in a member (or CF). For instance, page 1 from the GPFS must move to page 1 of a member. You do not have to populate all of the members or the CF with data so it is possible to load up the CF with data without filling any buffer pages in the members.

To move a page, click on a page in the CF and then while holding the LEFT mouse button down, drag the page to a member or CF.

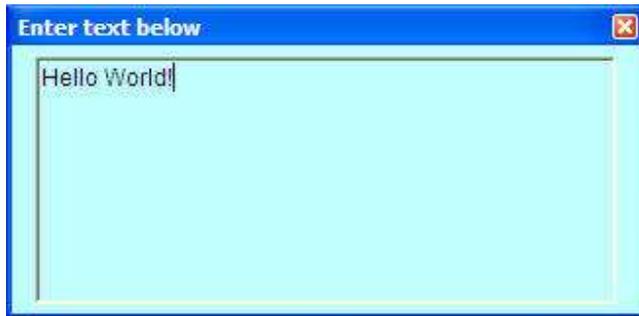


When you release the mouse over a page in a member or CF, the contents of that page will be updated with the data from the GPFS.



### Annotations

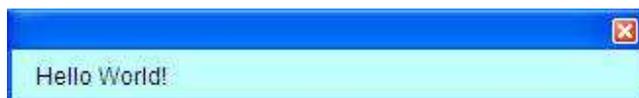
You can add annotations to your recorded script as you go along. To add an annotation, place the mouse at the location on the screen where you want a message placed and then right click. This will popup a comment box that you can then enter your text.



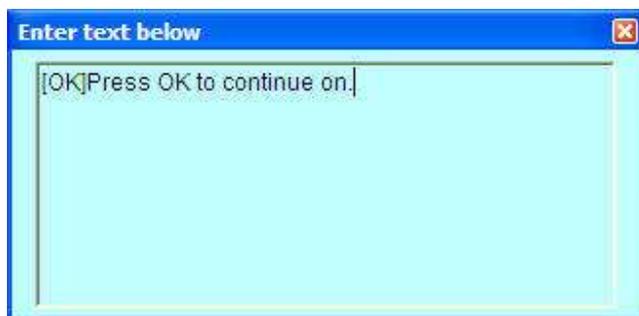
You can move the comment box anywhere on the screen by left clicking on the title bar and holding down the mouse while you drag it to the location you want.

Within the text box you can add any text that you want. When you are finished typing, hitting ENTER will save the comment. Then when your script is run, the comment will display for 4 seconds before continuing onto the next step.

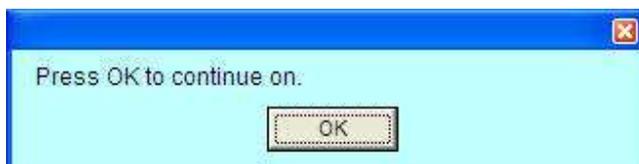
For instance, the text in the previous screen would look like the following image.



By default, the message will display for a few seconds before going to the next step in your simulation. You can add a button to the message to make it wait for the person to press before continuing on. The following script adds a button to the annotation.



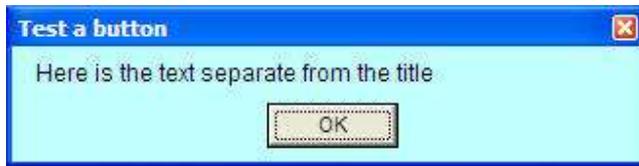
When the script runs, the user will see the following.



One final thing that you can add to an annotation is a title. A title will follow any button definitions and is separated from the message with a colon (:).

```
[OK]Test a button:Here is the text separate from the title
```

When this is run, the user will see the following.



## Options

There are a variety of options that are available to the user to control the speed of animation and the length that messages are displayed. The buttons on the far right side of the simulator are:

- Message (Info) display
- Animation time
- Tracing (Advanced)
- Tutorial
- Help



### Tutorial and Help

The two buttons on the far right side of the toolbar are used to guide you through the use of the simulator.



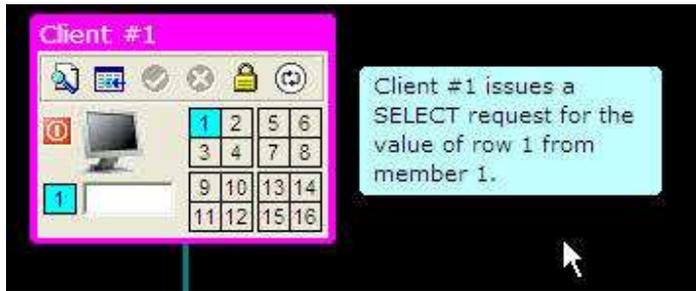
The TUTORIAL button will run a script taking you through some of the features that are found in the simulator, while the HELP button displays the full help information (this file).

### Message Display

The Message (INFO) button is used to control the display of informational messages on the screen.



The default value for informational messages is on. This means that the simulator will display messages as it runs, and wait 1.5 seconds before continuing on.



These messages are usually placed beside one of the objects on the screen, such as clients, members, CF, or GPFS. If you want to skip the message you can click anywhere on the black background or the green sidebar that displays on the far right-hand side of the display while the simulation is running.



If you click on the message itself, the background color of the message will turn yellow.



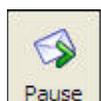
When the message is yellow, the system will pause until you click on the message again, the black background, or the green arrow on the sidebar.

Clicking on the INFO button will change its status to OFF.



When messages are off, no informational messages will be shown. So your simulations will run really fast but you won't know what is going on! However, there are situations where you may be debugging or testing a configuration and you want to skip the messages that are being shown.

If you press the INFO button one more time it will change to a status of PAUSE.

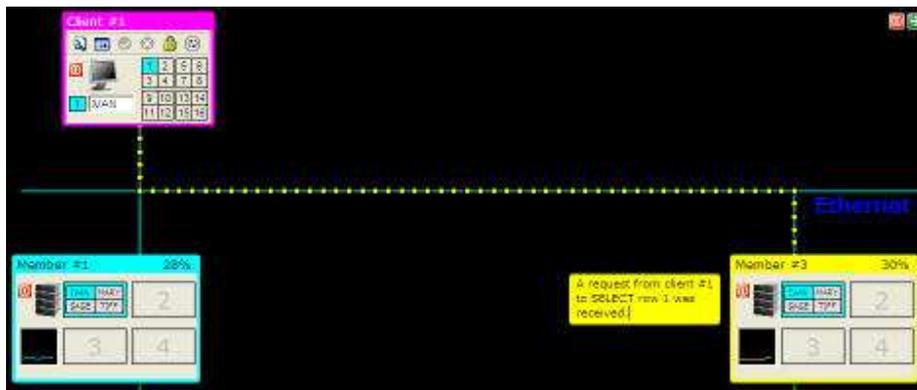


When the messages are in PAUSE status, every message will pause waiting for you to click on the message again, the black background or the arrow on the sidebar.

If you click the INFO button one more time it will cycle back to messages being ON.

### Animation Speed

When the simulation runs, the messages that are sent between clients, members, GPFS, and the CF are animated using a series of dots. For example, the following image shows a message being sent between a client and a member.



On average, these dots take between 1 to 2 seconds to reach their destination depending on distance (Note! In a real pureScale system, one of these communication requests takes about 15 microseconds!). You have the option of speeding up the animation and the length of time a message is displayed by clicking on this button. The values of the animation speed can be:

- 1.0x - The default speed of 1-2 seconds depending on distance
- 1.5x - A 50% increase in speed
- 2.0x - Twice as fast
- 4.0x - Four times as fast
- Off

If the animation speed is set to a numeric value (1.0, 1.5, 2.0, 4.0), the messages will continue to be displayed, but the pause time (for INFO ON) will be reduced. So instead of the default 1.5 seconds for a message to display, a 2.0x setting will make the message stay on the screen for only .75 seconds. You have the option of clicking on a message to pause it for further reading, or use the PAUSE setting for messages.

The final value (OFF) will draw all of the dots immediately from the source to the destination without individually turning them on and off. This mode speeds up the drawing of the communication lines, but it does not speed up the messages. The messages will continue to pause for 1.5 seconds before continuing on. This mode is a compromise if you want to speed up the simulation but still have time to read the messages.

You can change the default values that the program uses for the message delay, message format, and the animation speed by modifying the `_options.psc` file found in the `/system` directory. To change the amount of time that a message pauses before continuing on, you

would change the following line to include the delay time. In this example, the pause time is set to 3.5 seconds.

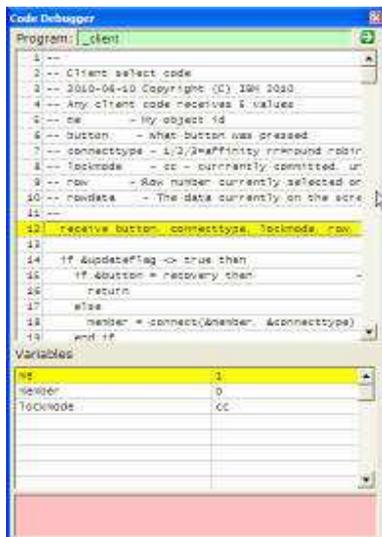
```
option message,3.5
```

### Tracing (Advanced)

The pureScale simulator is built upon a number of scripts that run in the background. If an error occurs, the trace window will appear and describe what went wrong. If you do encounter this, please let us know so we can figure out what the problem is. If you are interested in what the scripts are doing, click on the TRACE button.



The following image is a sample trace window that would get generated when you start a SELECT statement from a client.



The top of the trace window will show what statement is currently being executed while the bottom half will show the status of variables within the procedure. The trace window will update as the simulation runs. If your INFO setting is set to PAUSE, you will end up single-stepping through the code. You will have to click on the black background or the green arrow at the top corner of the trace window to continue to the next step.

If the INFO setting is ON, the code will run at the rate of statement every 1.5 seconds. Setting the animation button to 2x or 4x may make it more bearable to watch the code running! This trace window is usually reserved for debugging scripts, so you shouldn't have to use this for running your simulations unless you are curious about what is happening "under the covers".

Pressing the TRACE button again will remove the trace window from the screen.

## Program Credits

This program and documentation was compiled and created by:

**George Baklarz, PhD**  
**DB2 Worldwide Sales and Technical Support**  
**IBM Toronto Laboratory**  
**Markham, ON, CANADA**

I hope you enjoy using this demonstration package, and if you have any questions or comments, please send them to me at:

**[baklarz@ca.ibm.com](mailto:baklarz@ca.ibm.com)**

Special thanks to Kevin Taylor for his help in debugging, testing, and suggestions during the building of the simulator.

Aamer Sachedina from DB2 pureScale development was invaluable for his insight, patience, and knowledge during the prototype phases of development.

Thanks to Rex Wiederanders for his encouragement as we travelled around the world trying to figure out how to bring this concept to reality.

Finally, a special thanks to Katrina for her suggestions on the interface design and putting up with the late hours of debugging the code!

Thank-you for your time in trying out the DB2 pureScale Simulator!

## Support

If you have any problems with the pureScale simulator, please contact George Baklarz at:

[baklarz@ca.ibm.com](mailto:baklarz@ca.ibm.com)

If the problem is associated with a simulation run, please try and record the session and send the PSC file as part of the message so that we can try reproducing the problem.

As always, comments, suggestions, and opinions are welcome.

## Legal Disclaimers

IBM EVALUATION LICENSE AND SOFTWARE AGREEMENT ("Agreement") for the "PRODUCT", which is defined to be IBM pureScale Simulator Demonstration software and manuals plus "OTHER INFORMATION" which may include Product specifications/plans provided physically, orally, or electronically.

The Product is copyright of International Business Machines Corporation 2010. IBM Canada Ltd. ("IBM") grants you a non-exclusive, non-transferable license to use the Product solely for the purposes of testing and evaluating the performance of the Product under your operating conditions or in conjunction with the development of your applications. The Product must not be used in your normal business production environment.

The program simulates some of the features of the DB2 pureScale product (at time of introduction) but does not necessarily show all optimizations or steps involved in running an actual system. IBM reserves the right to modify the operation of pureScale to differ from that shown by the simulator.

You may not distribute, sublicense or otherwise transfer the Product, create derivative works, decompile, disassemble or otherwise attempt to translate or seek to gain access to the Product's source code. You also undertake that you will not transfer the Product, by any means, out of the country in which it is provided to you for use. In order to enable you to accomplish the objectives of this evaluation, you may make a reasonable number of copies of the Product, EXCEPT THAT YOU MAY NOT COPY THE OTHER INFORMATION WITHOUT IBM'S PRIOR WRITTEN PERMISSION. All copies that you make must bear the same legends and notices that appear on the original Product.

The Product may also contain code that will enable you to view and/or print the associated documentation. You may only use this code for that purpose. You may not copy, transfer, modify, merge, reverse compile, sublicense, rent, lease or otherwise assign this code. You understand that the product and any limited technical support are being provided to you "AS IS" without any warranties (express or implied) whatsoever, including but not limited to any implied warranties or conditions of merchantability, quality, performance or fitness for any particular purpose. Furthermore, IBM accepts no liability for damages you may suffer as a result of the technical support provided or not provided, or your use of the Product, including any damages claimed by you, based on any third party claims. In no event will IBM be liable for any indirect, special or consequential damages, even if IBM has been advised of the possibility of their occurrence.

Some jurisdictions do not allow the exclusion or limitation of warranties or consequential or incidental damages, so the above may not apply to you.

Although IBM will try to answer technical support questions you may have regarding your use of the Product, this assistance does not obligate IBM to provide support or maintenance services for the Product at this time.

IBM does not warrant or guarantee that the operability of any of your applications running with the Product will be maintained with any generally available versions of the Product or that any version of the Product will ever be made available or marketed.

You agree that IBM may use all suggestions, inventions, improvements and written materials you furnish to IBM in connection with this Agreement and that IBM may include them in any IBM product, and with respect to any such written materials, may claim copyright thereon as part of IBM's overall rights in any IBM product, without accounting to you.

The term of this Agreement will begin upon your receipt of this product.

IBM may terminate your license at any time, by giving you ten (10) days' prior written notice. In the event of your breach of any terms, IBM may terminate the license immediately. You must destroy the Product within ten (10) days of the expiration or termination of your license.

This Agreement shall be governed by and interpreted in accordance with the laws of the Province of Ontario.

All terms and conditions, which by their nature should survive the termination or expiration of this Agreement, shall survive.

This Agreement is the only understanding and agreement we have for your use of the Product. It supersedes all other communications, understandings or agreements (oral or written) that we may have had prior to this agreement.