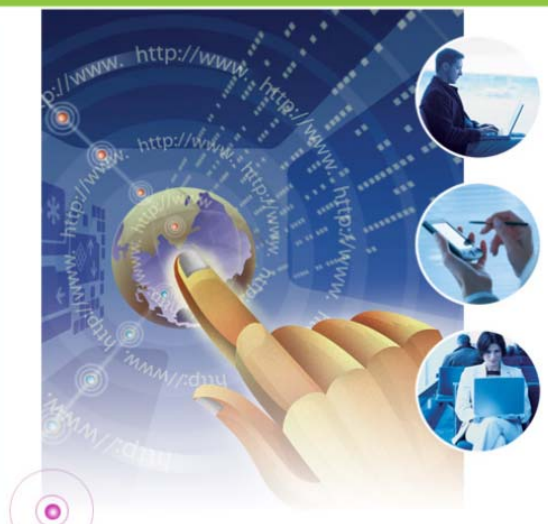
IDUG® 2004 – Europe



Enabling Your
On Demand DB2 World

Tuning Group Buffer Pools – One Command, One Spreadsheet

G05

Bryce D. Krohn
Krohn Enterprises, Inc.
www.krohnskorner.com

Tuesday, October 12, 2004
9:00 a.m. – 11:30 a.m.

DB2 for z/OS
Intermediate

INTERNATIONAL DB2 USERS GROUP

Group Buffer Pools are an important component of data sharing, first introduced in Version 4 DB2 data sharing. This presentation examines the mechanics of the group buffer pools, explains the performance considerations for those mechanics, and examines the output of DB2's DISPLAY GROUPBUFFERPOOL command. Key fields in the command output are explained in terms of their use in an Excel spreadsheet for use in tuning the group buffer pools.

This presentation is based on an excerpt from the three-day performance and tuning course CF340, "DB2 for z/OS and OS/390 Data Sharing Performance and Tuning", by Krohn Enterprises, Inc. Details can be found on the Internet at www.krohnskorner.com.

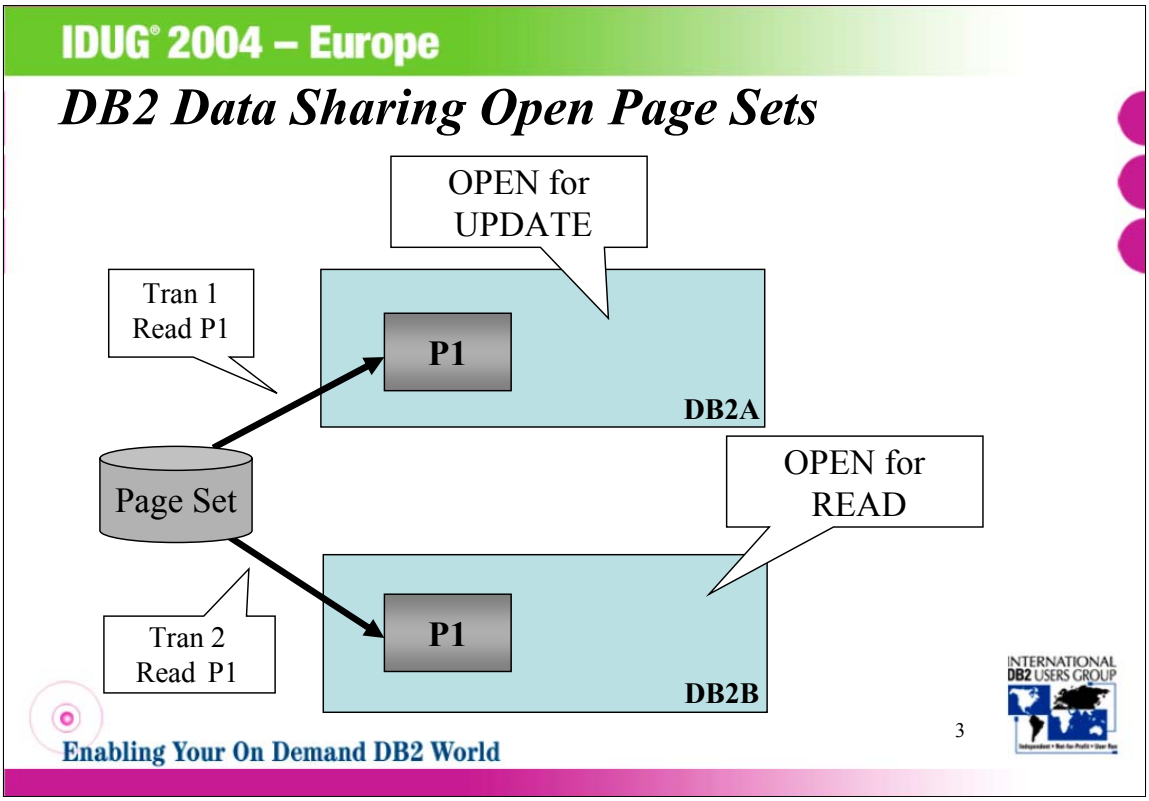
Tuning Group Buffer Pools – 1 Command, 1 Spreadsheet

- **GBP Mechanics Review**
- **DB2 Interactions with the GBP**
- **D GBPOOL GDETAIL**
- **D GBPOOL MDETAIL**
- **Spreadsheet Tracking Tool**

This presentation will first discuss the mechanics of how the group buffer pools work in DB2 data sharing mode. This discussion will be summarized with a look at the major interactions that DB2 has with the group buffer pools.

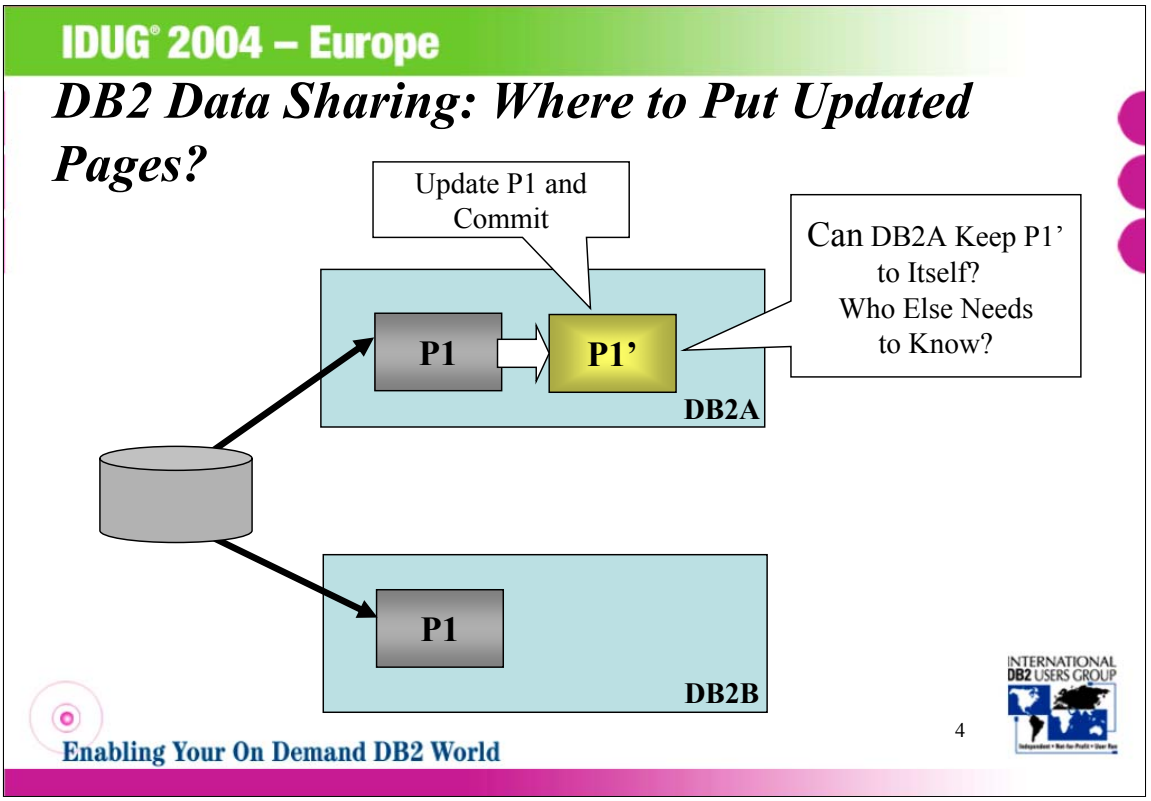
The major tuning tool for looking at the performance of the group buffer pools is the DISPLAY GROUP BUFFER POOL command. Both the GDETAIL, as well as the MDETAIL version of this command will be discussed in detail, pointing out the performance indicators in the display output.

Lastly, the presentation will explain the use of an Excel spreadsheet, a graphics tool that will assist in the analysis and tracking of group buffer pool performance.



Data sharing designers had to address the scenario of multiple DB2 members having a page set OPEN at the same time. In the example shown here, DB2A wishes to OPEN the page set for UPDATE, while the second member wishes to OPEN it for READ.

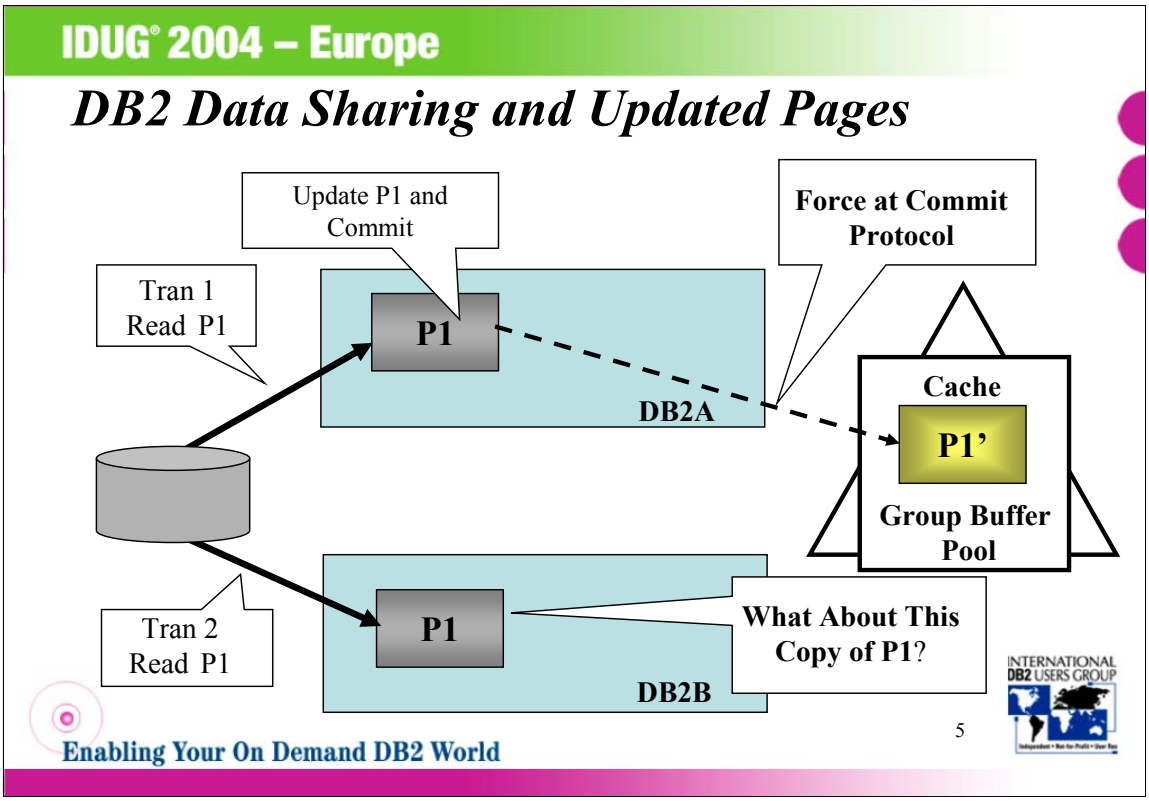
One of the challenges of the data sharing designers was devising a method of tracking multiple member's interest in OPEN page sets.



Once DB2 data sharing designers came up with a method of tracking interest in OPEN page sets, they also had to answer the question, "where to put updated pages", when the page belongs to a page set that is OPEN by multiple members.

Obviously, if multiple DB2 members have an interest in a page set, they must all know about updated pages. The updated page may contain data needed by a transaction running on a non-updating DB2 member.

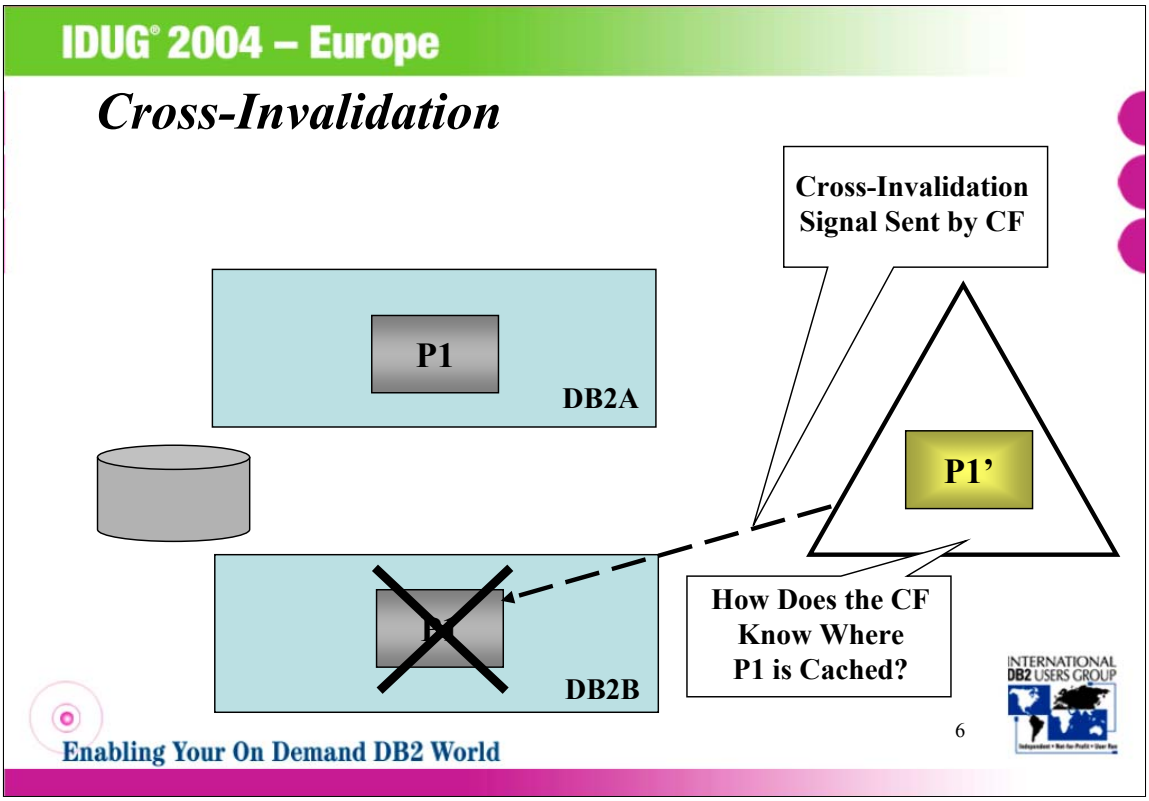
When DB2 updates a page in data sharing mode, there are additional considerations as to what DB2 should do with the updated page. Other DB2 members may have this same page set open and may have pages from this page set locally cached in their local buffer pools. Therefore, the DB2 member that updates a page must somehow communicate to the other DB2 members the fact that there is an updated page. It also must make the updated page available to the other DB2 members.



The answer to the question of "where to put an updated page" is provided by the group buffer pools – a structure in the coupling facility that DB2 uses to store data that needs to be accessible by multiple DB2 members. One of the 'rules' of data sharing states that an updating DB2 member must 'externalize' an update page at COMMIT time by writing it to the group buffer pool. This is called '**FORCE AT COMMIT'** protocol, and creates an updated page in the group buffer pool that other members can access at electronic speeds.

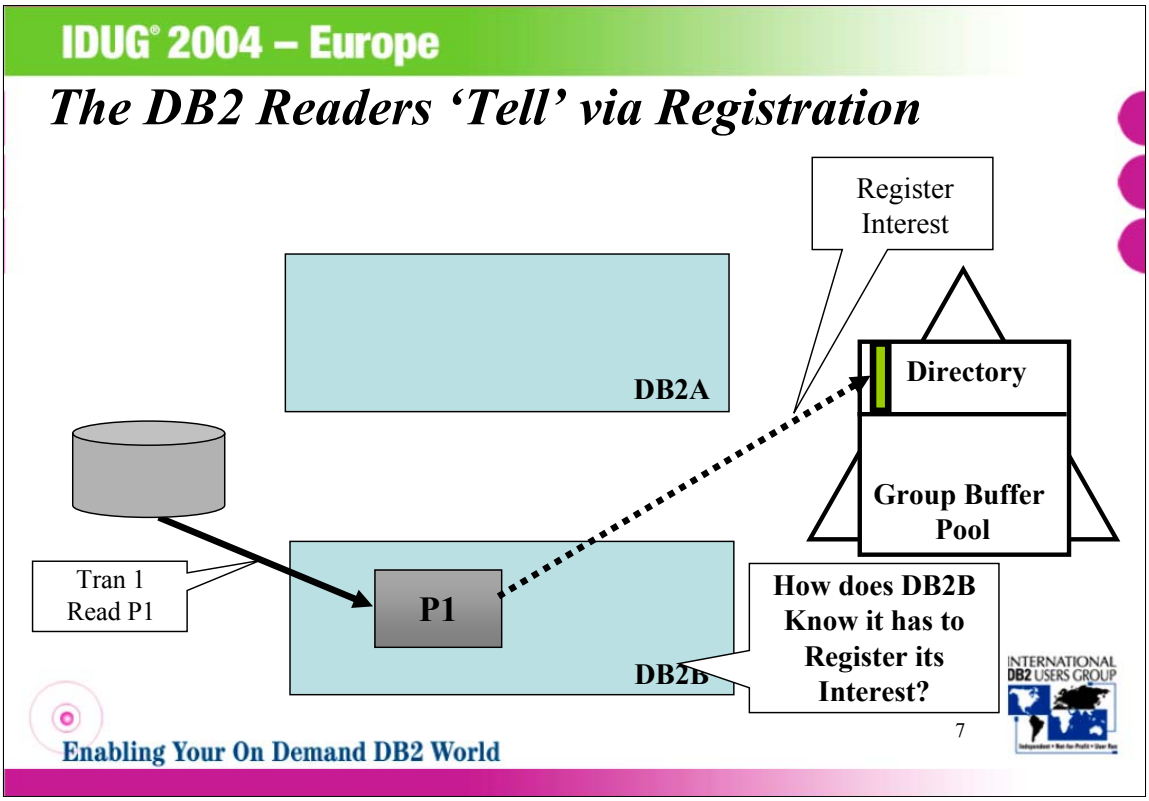
The additional consideration in data sharing is that another DB2 member also has the same page set open (e.g., DB2B in the above visual), and could have a copy of the same page (P1) in its local buffer pool that was just updated on another DB2 member (DB2A).

In this situation, what is the status of page P1 cached in DB2B?



The copy of page P1 locally cached in DB2B is now *'down-level'*, since it was updated by another DB2 member (DB2A in this example). DB2 data sharing must indicate to DB2B that its copy of page P1 is now invalid. DB2 does this by utilizing z/OS services provided by the micro-code running in the coupling facility. **CFCC** (Coupling Facility Control Code) sends an electronic signal to the z/OS image on which DB2B is running that is the electronic way of stating that page is invalid.

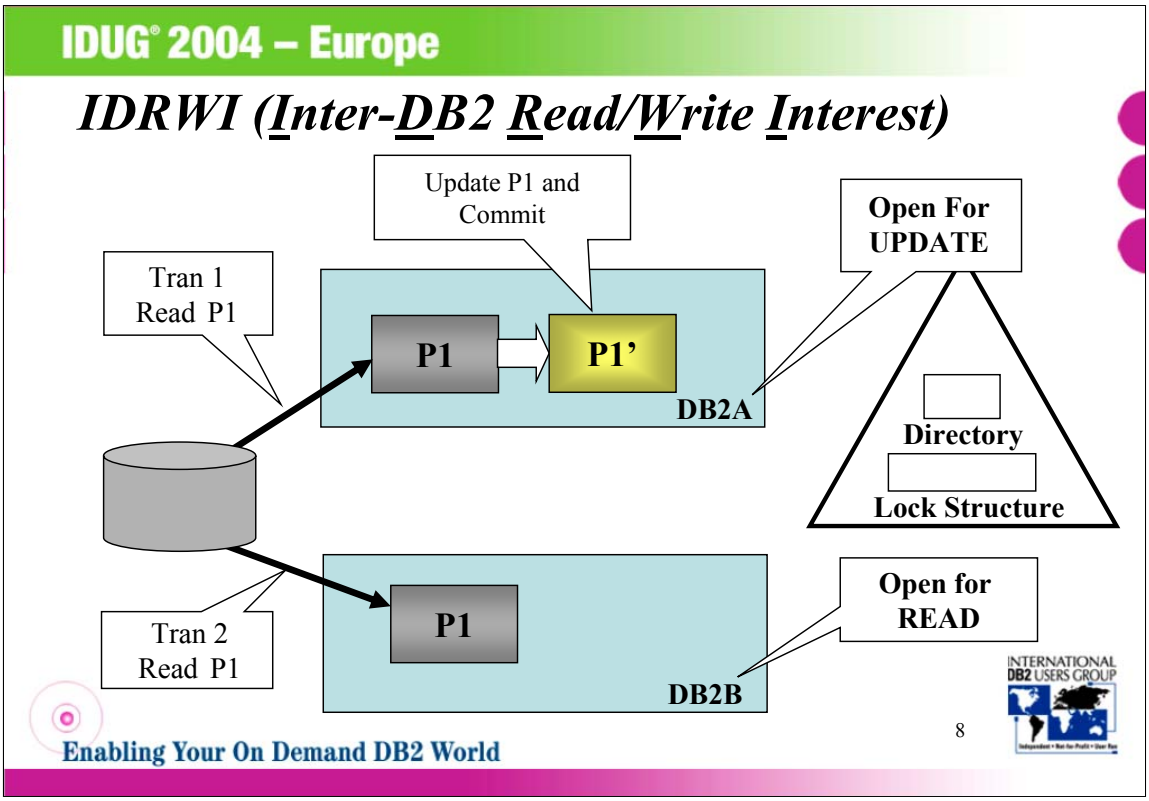
In the next few visuals, we will explore the details of that electronic signal as well as answer the question, "How does CFCC know where page P1 is locally cached?".



If a reading DB2 member knows that the page set it has open is being shared by other DB2 members, it knows it must tell the coupling facility about pages it has locally cached. It does this by 'registering' its interest in each page it reads into its local buffer pool. This interest is posted in a special part of the group buffer pool called the directory. The DB2 member posts that interest by sending a message to the coupling facility where a directory entry is created for each read page. This registration process is necessary for the coupling facility microcode to know the location of locally cached pages. The directory entry shows the coupling facility where to send cross-invalidation signals.

The next question to explore is this, "How does a DB2 reading subsystem know when it has to do this registering process?" The registering process utilizes XES services of z/OS to send the register instruction to the CF, but it does represent overhead to the DB2. How does DB2 know when to do this extra work?

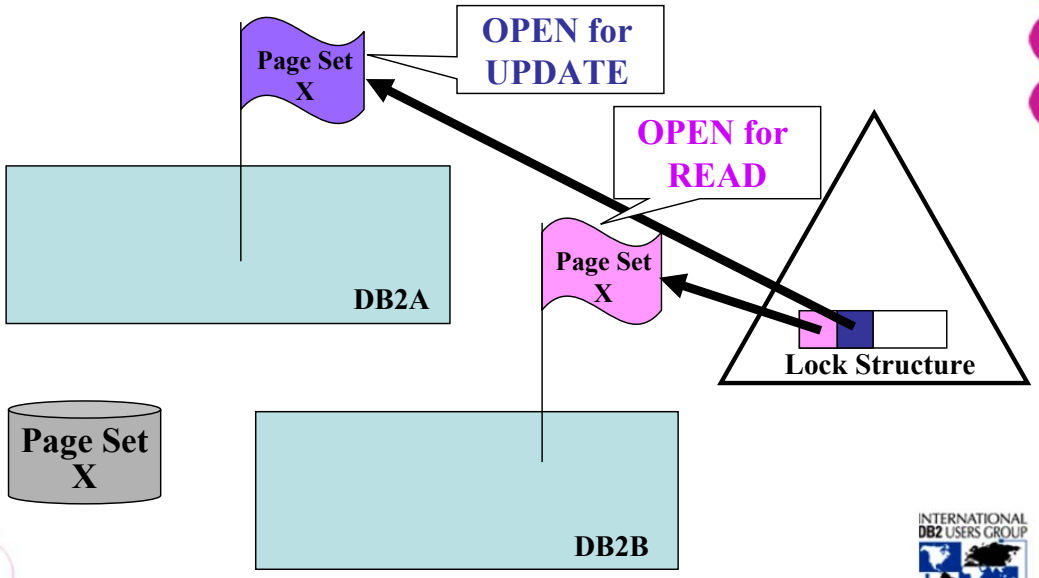
The answer lies in something called **IDRWI** – **I**nter-**D**B2 **R**ead/**W**rite **I**nterest. The next few visuals will explore this concept.



The cross-invalidation mechanism we just looked at is only necessary when multiple DB2s have the page set OPEN. This condition is known as **IDRWI** (Inter-DB2 Read/Write Interest). The official definition of IDRWI states that one DB2 member has the page set OPEN for UPDATE intent, and at least one other DB2 member has the page set OPEN for READ intent.

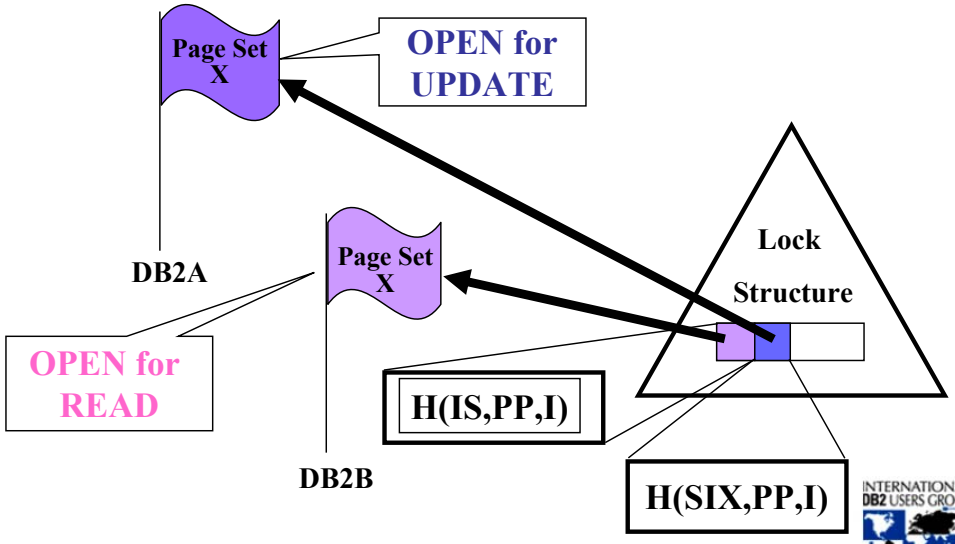
As we will see in later visuals, when IDRWI exists, a data sharing-enabled DB2 has additional overhead it must do.

IDRWI Indicators



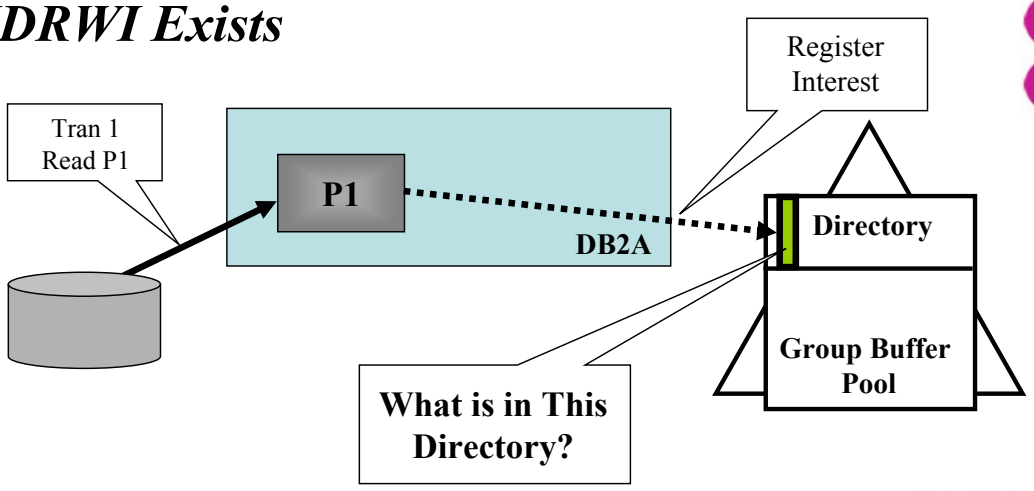
In order to know that IDRWI exists, each DB2 subsystem must know the level of interest of other subsystems for open page sets. This level of interest is posted for each DB2 subsystem at OPEN time, i.e., an indicator is set in the system that makes its level of interest in an open page set visible to other members. Logically, you can think of these indicators as “flags” posted by each subsystem such that other members can see them. We will look at the physical implementation of these logical indicators in the next visual.

IDRWI - Tracked by Page Set P-Locks



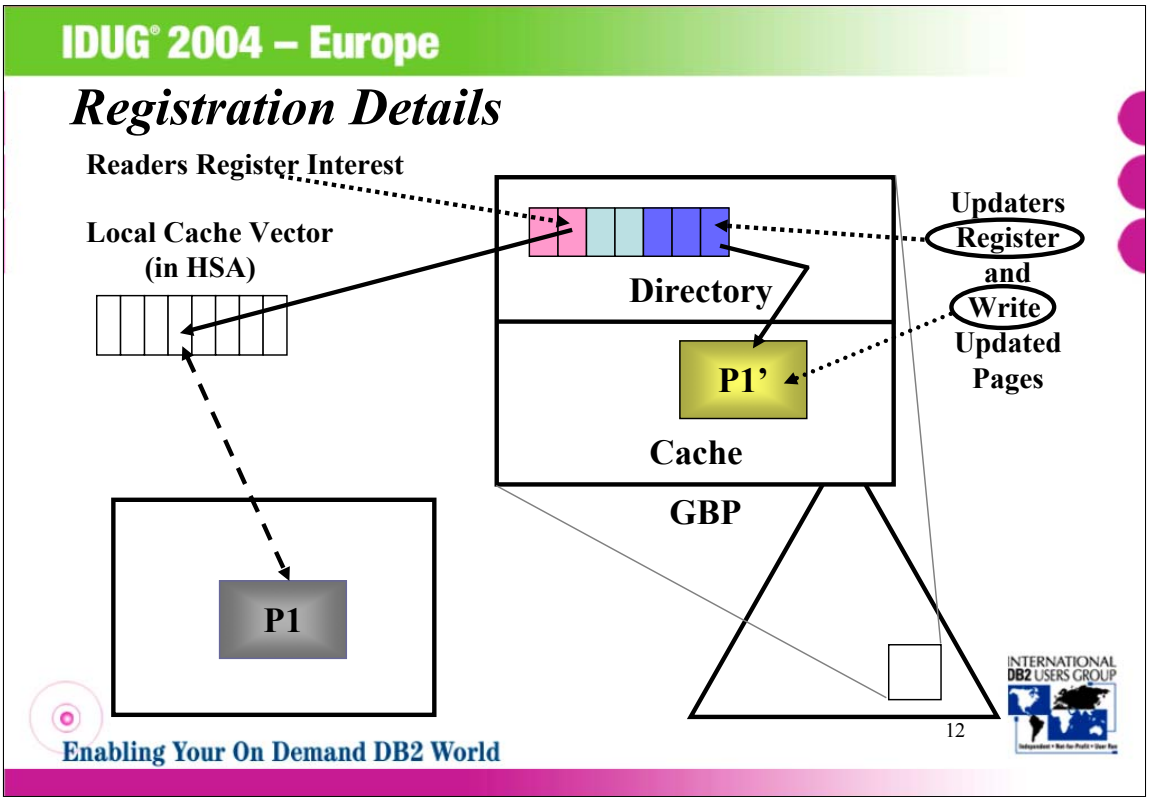
A new kind of lock, a page set/partition **P-lock** (Physical Lock), is the indicator used by a DB2 data sharing member to post its level of interest in an open page set. Since each DB2 member also posts its interest in an open page set at OPEN time, all the members know of the common interests in a particular page set. If at least one of those members has the page set open for UPDATE, (i.e., holds a page set P-lock with UPDATE intent), and at least one other member has the page set open for READ, (i.e., holds a page set P-lock with Read intent), then IDRWI exists.

Readers Must Register Interest When IDRWI Exists



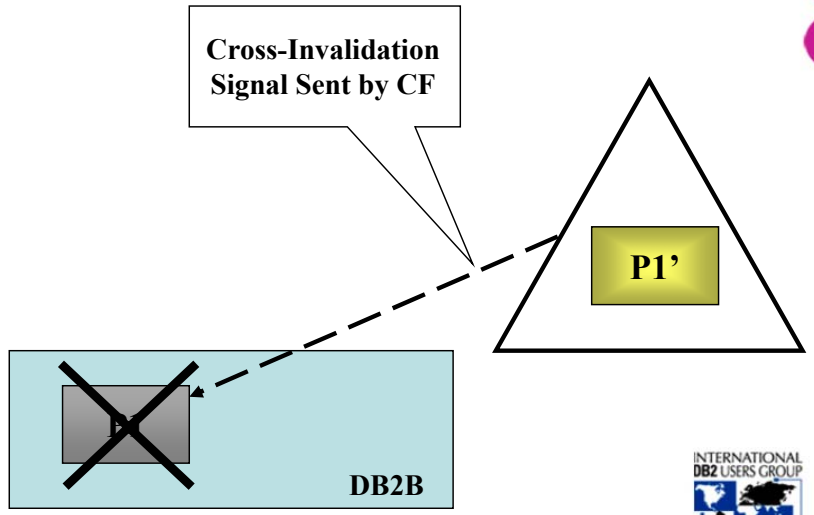
The reading DB2 member knows that it must register its interest if it knows that IDRWI exists. The existence of IDRWI is indicated by the page set P-locks held by each DB2 member on that open page set.

Next, we'll look at the details of what the directory entry contains.



Let's look at the details of the registration process. Each directory entry consists of a page id followed by an internal address pointer which is pointing at a bit in the **LCV** (Local Cache Vector) on the z/OS image where that page is cached in a DB2 local buffer pool. The LCV is a storage area set up by z/OS in one of its internal storage areas called the **HSA** (Hardware System Area). Each bit in the LCV represents a page frame in DB2's local buffer pools and hiperpools. As we will see in the next visual, this LCV bit will be the indicator used by the cross-invalidation mechanism.

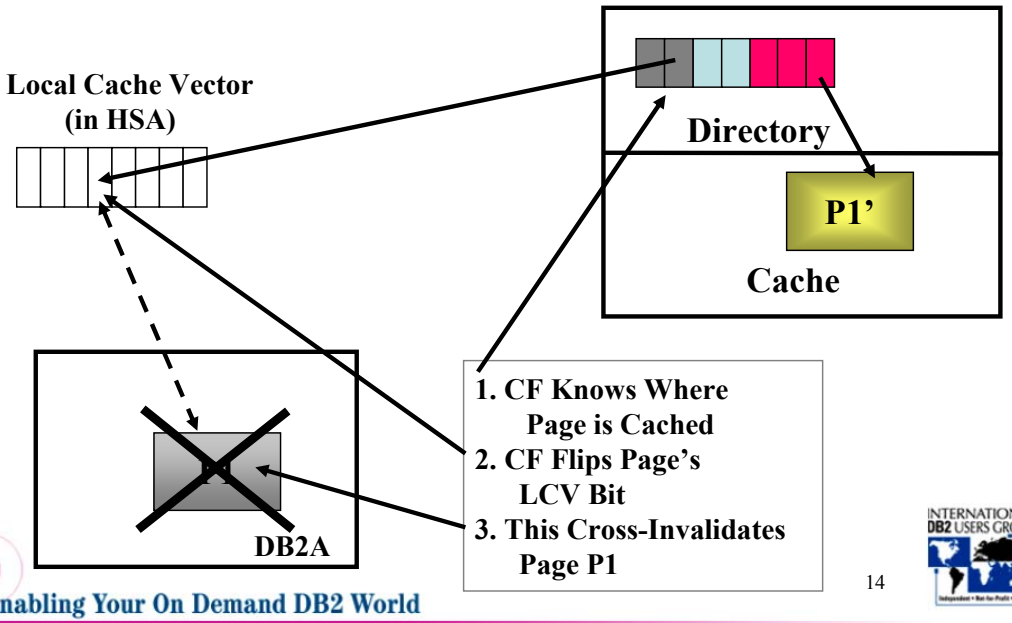
Cross-Invalidation Details



Since locally cached pages are in the directory in the group buffer pool (the reading DB2 subsystem registered its intent there), the coupling facility microcode knows where to send cross-invalidation signals.

We will look at the details of this mechanism in the next visual.

Cross-Invalidation Details . . .



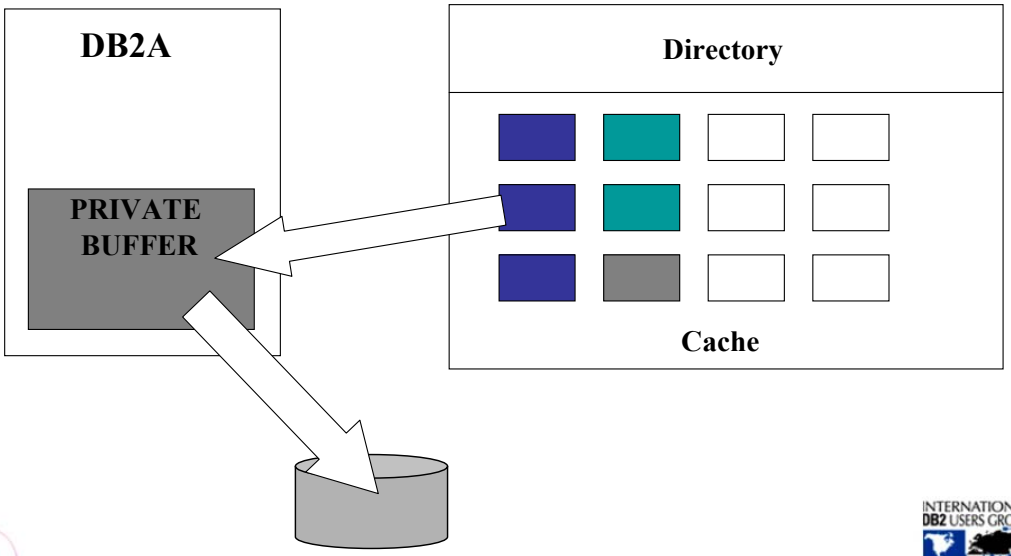
When IDRWI exists, the coupling facility control code at UPDATE time scans the directory for any page entries that match the page id just updated. If it finds one, it sends an electronic signal to the appropriate z/OS image to flip the bit that represents the page cached locally in that system. Later, at re-reference time, the local reading DB2 subsystem needs to check that bit to see if the page is still valid (validity checking).

DB2 Interactions with the Group Buffer Pool

- Writes to the GBP
  - Updated **Pages** of an IDRWI Page Set
  - **Register** Requests for a Read Page
  - **Register** Requests for an Updated Page
  - GBPCACHE ALL attribute
- Reads from the GBP
  - GETPAGE Requests

This visual lists the actions that make up the majority of data sharing overhead for global buffering. As indicated in the list on the visual, updating DB2 subsystems must write updated pages to the group buffer pool at COMMIT time, as well as register those updated pages. Reading DB2 subsystems must register their interest in read pages of an IDRWI page set. Reading DB2 subsystems may also have to write clean pages to the group buffer pool, depending on the GBPCACHE attribute, which we will explain in the next visual.

Group Buffer Pool Castout



Since there is no physical connection from the coupling facility to DASD, DB2 subsystems must do the work of writing updated pages to DASD. The DB2 who is the current castout owner reads updated pages into a private buffer in the DBM1 address space and does the physical I/O to write the updated pages to DASD. The castout ownership is assigned when a DB2 subsystem first opens a page set for update.

In the next visual, we will discuss the various thresholds that control this process of castout.

GBP Castout Thresholds

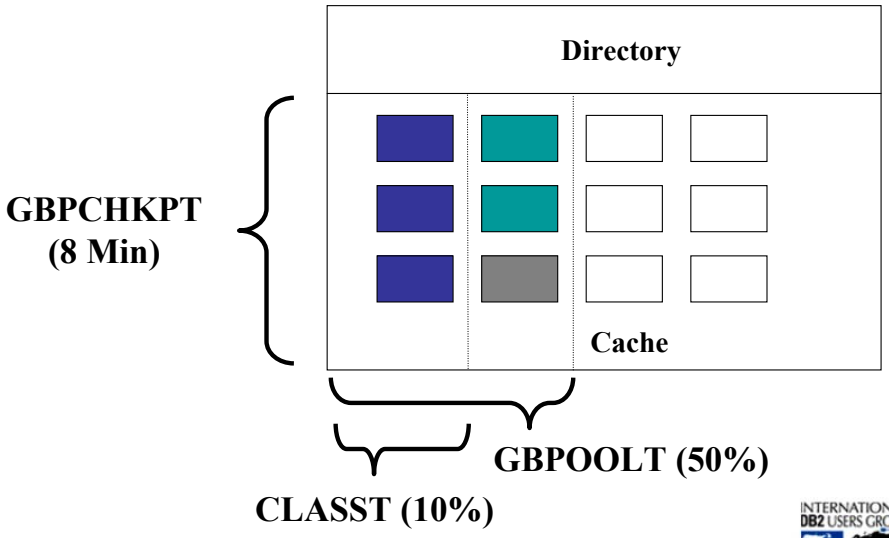
- Castout class (CLASST)
- GBP Threshold (GBPOOLT)
- GBP Checkpoint (GBPCHKPT)
- IDRWI going away
- GBP Rebuild (if back-up CF too small)

There are several triggers that will start the castout process. The first three, CLASST, GBPOOLT, and GBPCHKPT, are parameters of the **ALTER GROUPBUFFERPOOL** command.

When IDRWI goes away, i.e., the updating DB2 downgrades its interest in a page set to Read-Only, the castout process is started for the updated pages for that page set.

Another special case of castout can happen during a group buffer pool rebuild process which we will cover in a later visual. Let's look at the details of these thresholds.

GBP Castout Threshold Defaults

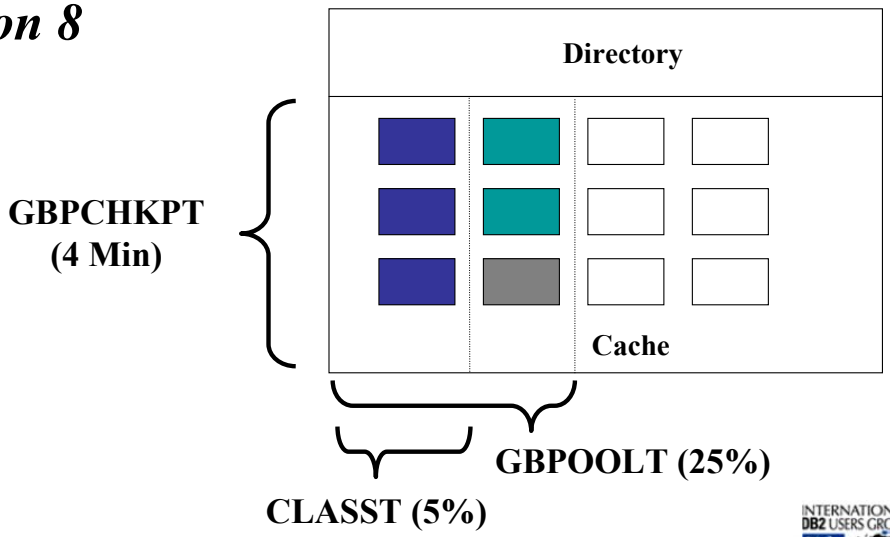


The first threshold we will discuss is the class threshold. z/OS and DB2 assign page sets to one of 1,024 classes in the group buffer pools. Each class is then tracked by the coupling facility. As soon as the updated pages in one class exceed the CLASST threshold, the castout process is started for that class.

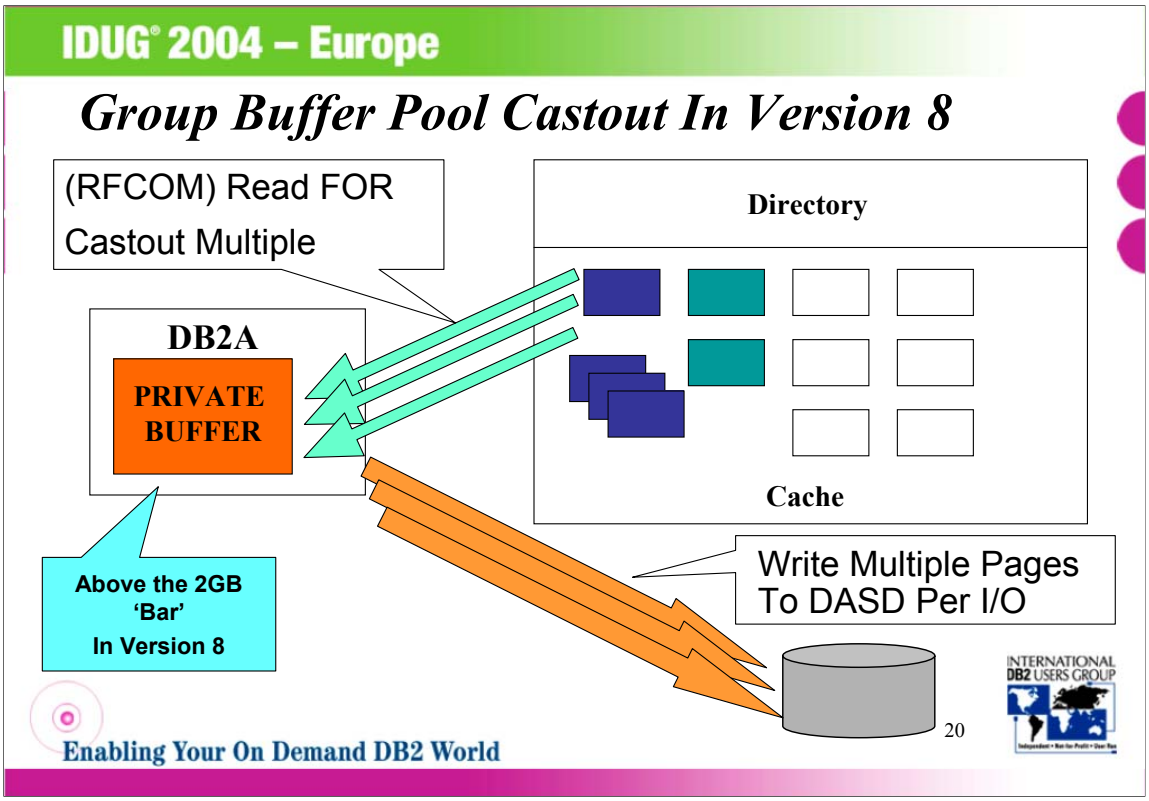
The second threshold is the group buffer pool threshold called GBPOOLT. This is the threshold that determines the percentage of CHANGED pages in the entry pool that will trigger castout. As soon as the number of CHANGED pages in all classes exceeds this threshold, castout begins.

The third threshold is the group buffer pool checkpoint interval called GBPCHKPT. This threshold sets the time interval at which DB2 will write all updated pages from the group buffer pool.

GBP Castout Threshold Defaults in Version 8



Note that the group buffer pool castout default values have been changed in Version 8 of DB2. They are half of what they used to be, mainly to take advantage of the faster z990 processors, as well as the newer instructions supported by z/OS to facilitate castout processing.



Version 8 includes a significant improvement to the GBP castout process. By taking advantage of new z/OS commands (z/OS 1.4 is required, along with CFCC level 12), the castout owner can read multiple pages for castout in one command, the **RFCOM** (Read For CastOut Multiple) command. These multiple pages are then written to DASD in multiples, saving many individual I/O requests.

This more efficient castout process, along with the more efficient register process utilizing the **WARM** (Write and Register Multiple) command, will lessen traffic to the CF, as well as reduce the overhead in DB2 code.

ONE Command – DISPLAY GBPOOL(gbp-name)

--DISPLAY GROUPBUFFERPOOL(GBP08) . . .

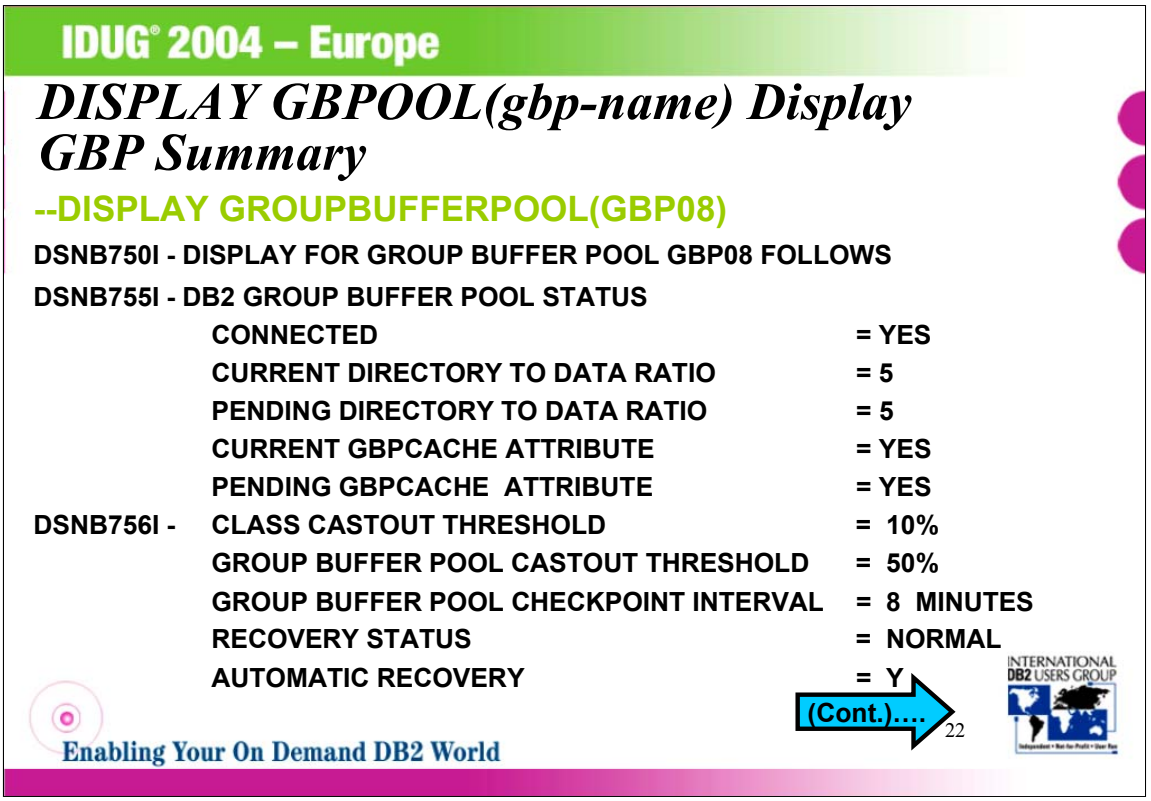
- **One DB2 Command**
- **Powerful Tuning Tool**
- **Three Modes:**
  - **Summary**
  - **GDETAIL – Group Detail**
  - **MDETAIL – Member Detail**

The DISPLAY GROUPBUFFERPOOL command displays a wealth of information for monitoring the usage of the GBPs by DB2. The summary format of the command, gives a snapshot view of the GBP attributes.

The GDETAIL option provides a look at the activity against the group buffer pool from the data sharing's group perspective. Statistics are provided to enable the user to check the current settings of the group buffer pool, and how well those tunable settings are handling the current workload.

The MDETAIL version of the command provides detailed information about the group buffer pool from the perspective of the DB2 member on which the command was issued. This information can also be used for detailed tuning work on the group buffer pool.

The next several visuals will explore this information, and explain how it can be used to tune the group buffer pools.

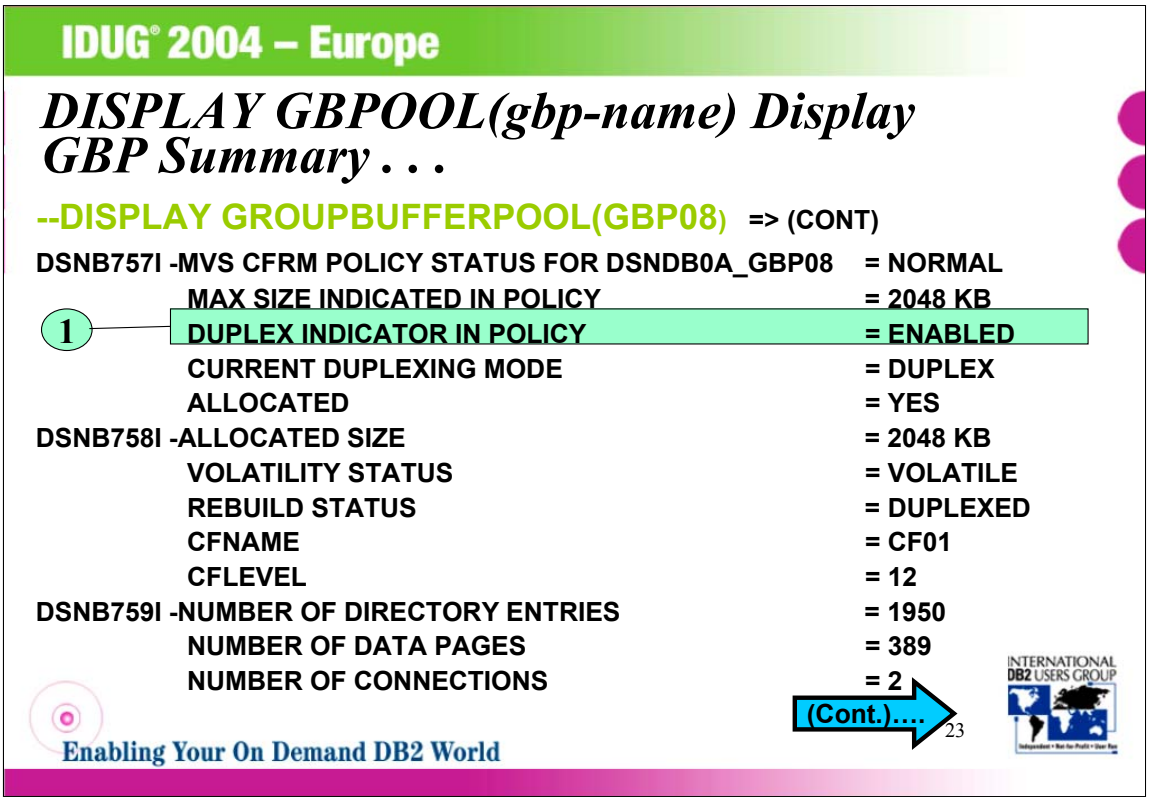


The DISPLAY GROUPBUFFERPOOL command displays a wealth of information for monitoring the usage of the GBPs by DB2. The summary format of the command, shown on this and subsequent visuals, gives a snapshot view of the GBP attributes.

The display shows whether or not the GBP is currently connected. In addition, the current as well as the pending values of the RATIO are displayed. This is the ratio of the number of directory entries to the number of data pages. If it is changed, the new value will show in the PENDING ratio field. It will take effect at the next allocation, or manual rebuild of the GBP.

The **DSNB756I** message reports the thresholds of the GBP, as well as the GBP checkpoint value.

The recovery status will be NORMAL, or **DAP** (Damage Assessment Pending). A status of DAP means DB2 is in the process of finding out which page sets were in the pool at the time of either a structure failure, or 100% loss of connectivity when in simplex mode. Those page sets will be marked as **GRECP** (Group Buffer Pool RECover Pending).



The SUMMARY report for the DISPLAY GROUPBUFFERPOOL command provides a detailed look at the attributes of an allocated GBP.

The allocated size is reported, as well as the indicated and current DUPLEXING mode. (**NOTE**: The CURRENT DUPLEXING MODE – which is also found in the GDETAIL and MDETAIL report - will be input to question #1 of the GBP spreadsheet.) The REBUILD status may be (depending on whether or not DUPLEXING is active, or in the process of happening):

- DUPLEXED
- QUIESCE
- CONNECT
- CLEANUP
- STOPPING

The CF holding the primary buffer pool (if duplexed), as well as the CFLEVEL of the microcode that DB2 is connecting to is shown. (Note that this CFCC level may differ from that shown by a D XCF command – which shows the actual level of code installed).

The number of directory entries, and space for data pages, as well as current number of connections to the structure are reported.

DISPLAY GBPOOL(gbp-name) Display GBP Summary. . .

```
-DISPLAY GROUPBUFFERPOOL(GBP08)  => (CONT)
DSNB798I -LAST GROUP BUFFER POOL CHECKPOINT
                                        17:03:41 OCT 02, 2001
GBP CHECKPOINT RECOVERY LRSN                = BF6BBAAB03CD
STRUCTURE OWNER                             = DB1A
DSNB799I -SECONDARY GBP ATTRIBUTES
          ALLOCATED SIZE                    = 2048 KB
          VOLATILITY STATUS                 = VOLATILE
          CFNAME                            = CF02
          CFLEVEL                           = 12
          NUMBER OF DIRECTORY ENTRIES       = 1950
          NUMBER OF DATA PAGES              = 389
DSNB790I -DISPLAY FOR GROUP BUFFER POOL GBP08 IS COMPLETE
DSN9022I -DSNB1CMD '-DISPLAY GBPOOL' NORMAL COMPLETION
```


The time and date of the last group buffer pool checkpoint, and the **LRSN** (Log Record Sequence Number) that would be the start of any subsequent recovery reads from the log, are reported.

The name of the current structure owner, the DB2 member responsible for the group buffer pool checkpoint, among other things, is also listed.

If DUPLEXING is currently active, corresponding attributes of the secondary group buffer pool is then listed.

NOTE: It is probably a good idea to make sure the actual CFLEVEL of the two CFs involved is the same. This is reported by a D XCF command.

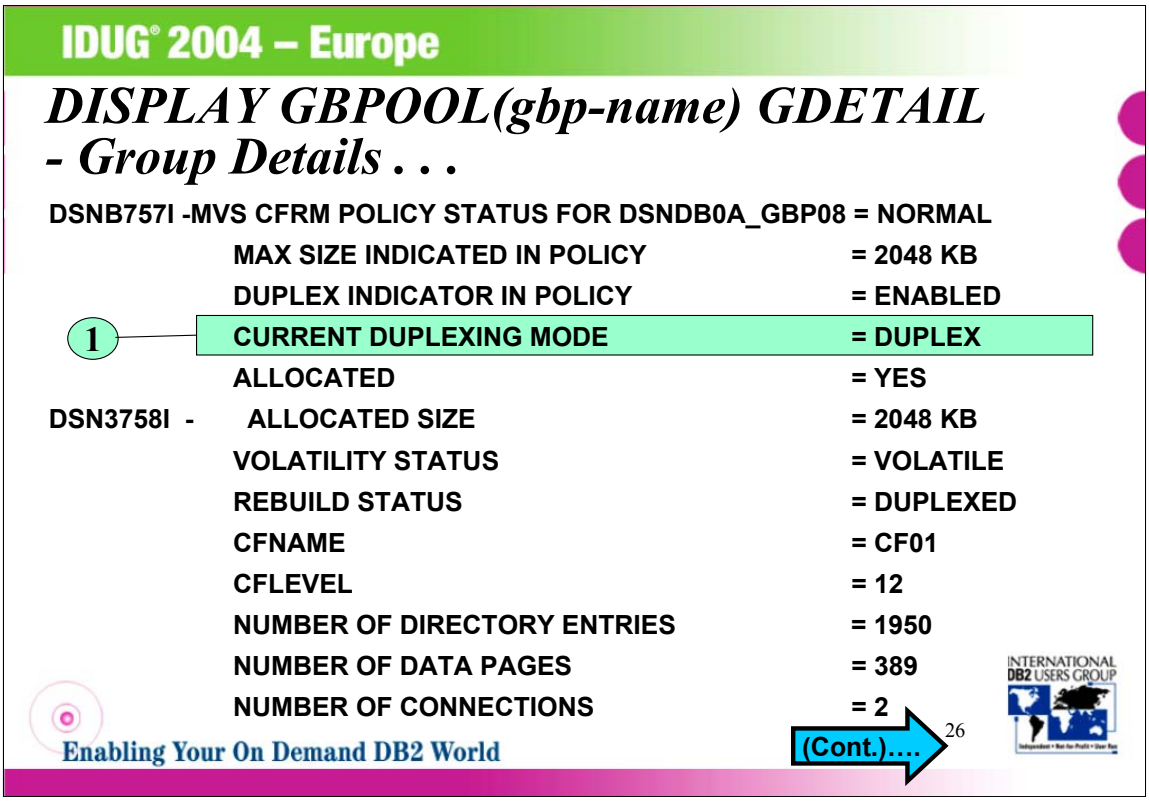
DISPLAY GBPOOL(gbp-name) GDETAIL - Group Details

```
-DISPLAY GROUPBUFFERPOOL (GBP08) GDETAIL
DSNB750I -DISPLAY FOR GROUP BUFFER POOL GBP08 FOLLOWS
DSNB755I -DB2 GROUP BUFFER POOL STATUS
                CONNECTED                                = YES
                CURRENT DIRECTORY TO DATA RATIO          = 5
                PENDING DIRECTORY TO DATA RATIO          = 5
                CURRENT GBPCACHE ATTRIBUTE               = YES
                PENDING GBPCACHE ATTRIBUTE               = YES
DSNB756I -CLASS CASTOUT THRESHOLD                        = 10%
                GROUP BUFFER POOL CASTOUT THRESHOLD      = 50%
                GROUP BUFFER POOL CHECKPOINT INTERVAL    = 8 MINUTES
                RECOVERY STATUS                          = NORMAL
                AUTOMATIC RECOVERY                       = Y
```

(Cont.)

The Group Detail version of the DISPLAY GROUPBUFFERPOOL command displays the GBP from the Group's perspective. It is a very valuable tool in monitoring and tuning the GBP.

The first part of the display is exactly the same as the summary format - displaying attributes of the currently allocated GBP - its size, location, and status.



This portion of the display is just like that of the summary format.

(**NOTE**: The 'CURRENT DUPLEXING MODE' value – DUPLEX OR SIMPLEX – will be used as input to question #1 of the GBP performance spreadsheet.)

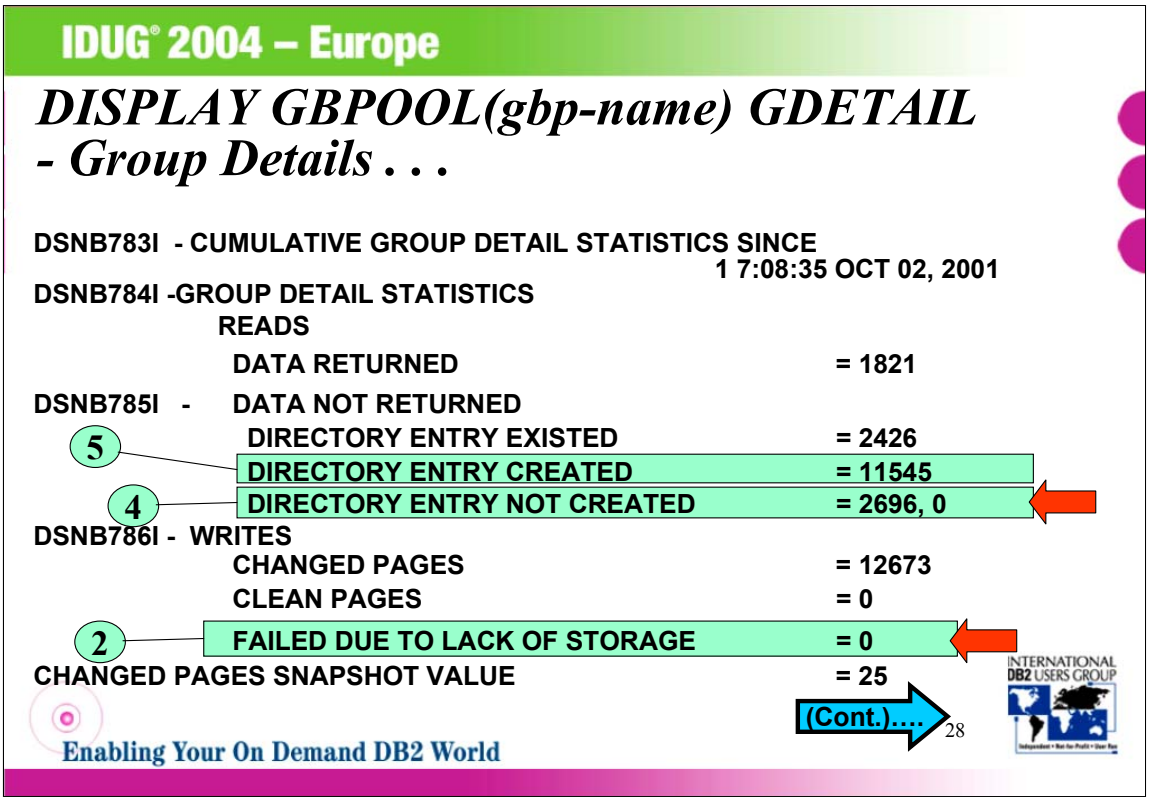
DISPLAY GBPOOL(gbp-name) GDETAIL - Group Details . . .

```
DSNB798I - LAST GROUP BUFFER POOL CHECKPOINT
                                        17:03:41 OCT 02, 2001
           GBP CHECKPOINT RECOVERY LRSN      = BF6BBAAB03CD
12         STRUCTURE OWNER                   = DB1A
DSNB799I -  SECONDARY GBP ATTRIBUTES
           ALLOCATED SIZE                    = 2048 KB
           VOLATILITY STATUS                 = VOLATILE
           CFNAME                            = CF02
           CFLEVEL                           = 12
           NUMBER OF DIRECTORY ENTRIES       = 1950
           NUMBER OF DATA PAGES              = 389
```

(Cont.)….

Again, the display is the same as the summary format, through the **DSNB799I** message, which gives the details of the secondary GBP, if DUPLEXing is active.

NOTE: The designation of the STRUCTURE OWNER. This will be the input to question #12 on the GBP spreadsheet later, and will be used to check the values of the castout thresholds that were met in this interval.



Message **DSNB783I** displays the start of the time interval being reported. It can be reset with the INTERVAL parameter.

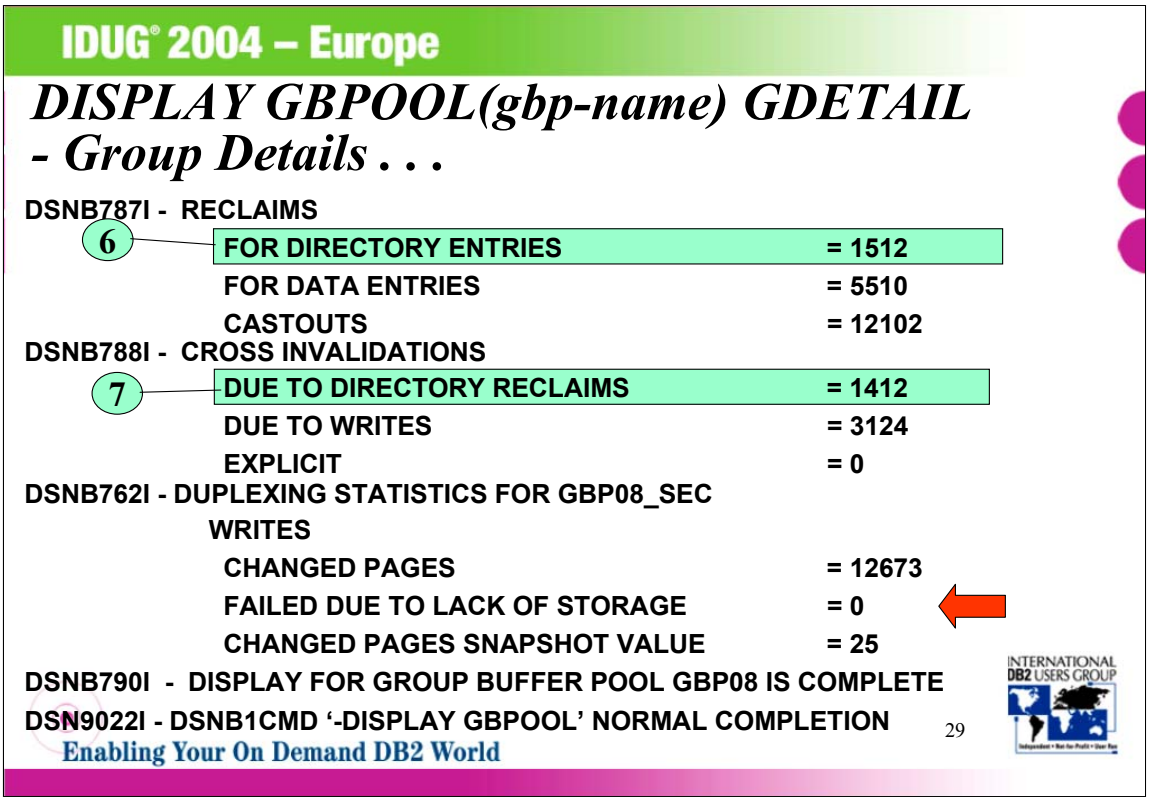
The **DSNB784I** READS counter shows the 'Read Hit' counter - the number of times DB2 requested a page from the GBP, and the read was successful. This is an important performance number, as reads from the GBP are much faster than reads from DASD.

The **DSNB785I** message breaks out unsuccessful reads, and whether or not directory entries were created at the time of the read request. The second counter in 'DIRECTORY ENTRY NOT CREATED' **SHOULD ALWAYS BE ZERO,** as it reports number of directories not created because of storage shortage. (**NOTE**: The number in the 'DIRECTORY ENTRY CREATED' counter will be input to question #5 in the GBP performance spreadsheet, and the second number in the DIRECTORY ENTRY NOT CREATED entry will be input to question #4 of the GBP performance spreadsheet.)

The **DSNB786I** counters report the number of writes to the GBP. CHANGED reports the updated pages written when IDRWI existed. A non-zero counter in 'CLEAN PAGES' indicates that GBPCACHE is specified either for the page set, or the GBP.

The 'FAILED DUE TO LACK OF STORAGE' counter **SHOULD ALWAYS BE ZERO**. A non-zero number here indicates DB2 had to put pages on the LPL, making them unavailable until recovered. (**NOTE**: This value will be input to question #2 on the GBP performance spreadsheet.)

The 'CHANGED PAGES SNAPSHOT' value is the number of changed pages in the pool at the time of the command.

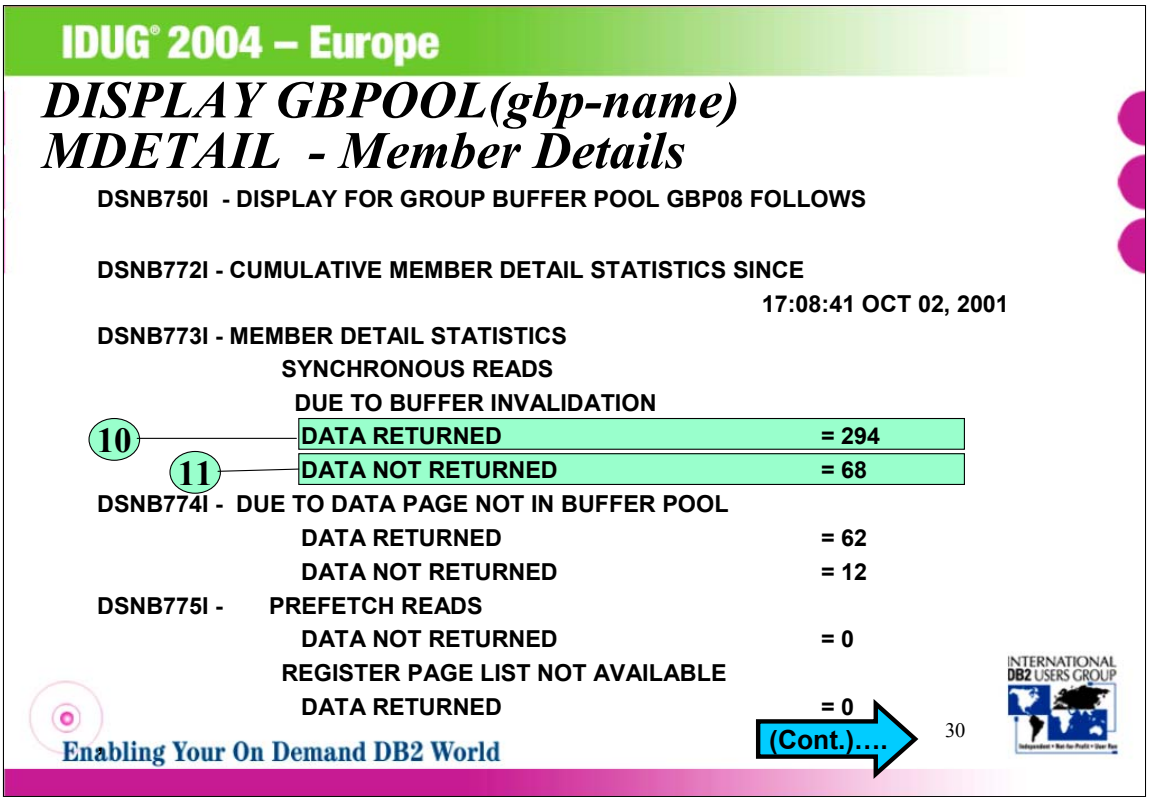


DSNB787I displays counters for *Reclaims*, the number of times DB2 needed to 'steal' entries for new requests, whether for a directory entry (**NOTE**: Used as input to question #6 on the GBP performance spreadsheet), or a data entry. *CASTOUTS* is the number of times changed pages were externalized to DASD, triggered by the thresholds.

DSNB788I reports the number of cross-invalidations performed by the CF. If the first counter, *DUE TO DIRECTORY RECLAIMS,* is too high, and there is a large amount of re-referencing of locally cached data pages, there is a high performance cost. The local DB2s have to refresh the page from DASD, and after suffering the overhead of a physical I/O, find the page just read is EXACTLY the same as the cross-invalidated one! (**NOTE**: This counter is input to question #7 on the GBP performance spreadsheet).

The *EXPLICIT* counter is the number of explicit XIs issued by the DB2s, because of a GBPCACHE NONE or SYSTEM specification on a page set.

The **DSNB790I** counters repeat the primary counters as they apply to the secondary GBP. In most cases, they should be equal - however, because of timing problems, they are not always exactly the same.



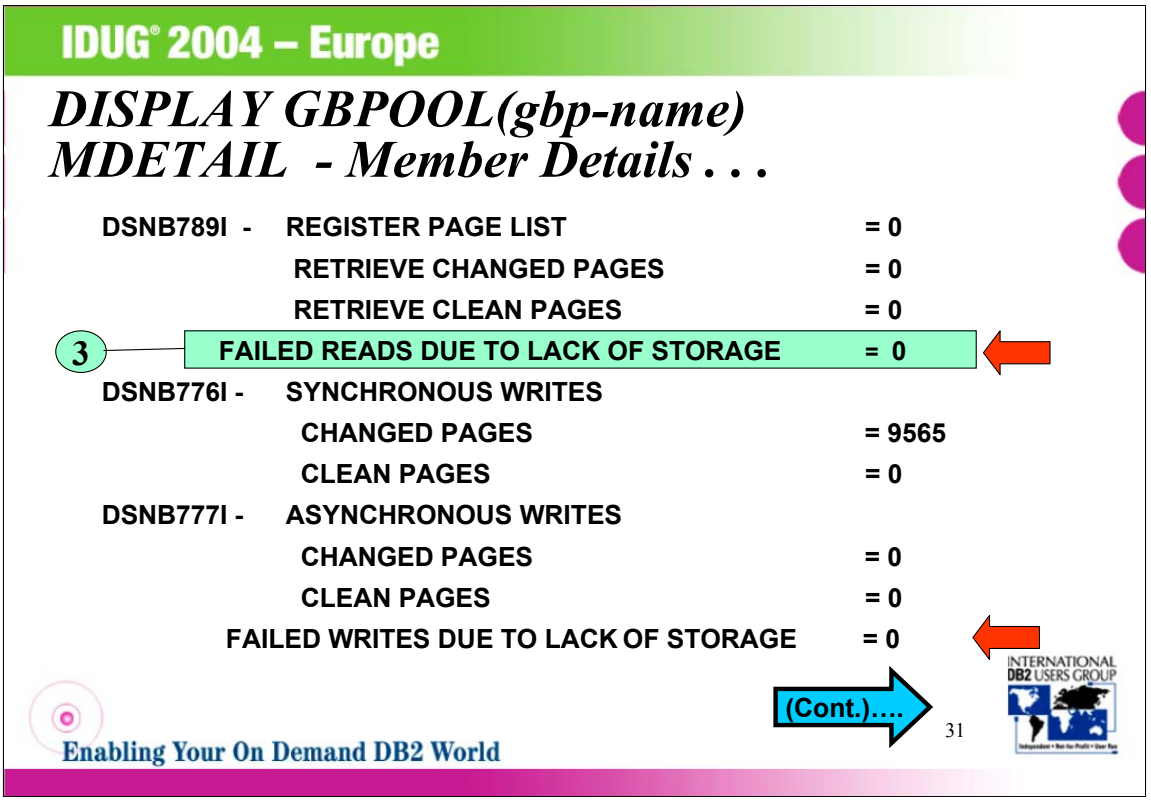
The **MDETAIL** version of the DISPLAY GBPOOL command reports detailed statistics on a specific GBP as seen by a member, not the entire group.

DSNB773I reports the synchronous reads issued by this DB2, and their results. 'Synchronous' in this context means that the reads were done under the user's execution unit, synchronous with processing of the user's transaction.

'DATA RETURNED' under 'DUE TO BUFFER INVALIDATION' reports the number of successful reads from the GBP when the member's locally cached pages were invalidated. This is a 'hit ratio' of the GBP, which should be as high as possible for performance reasons. (**NOTE**: This number will be input to question #10 on the GBP performance spreadsheet). 'DATA NOT RETURNED', on the other hand, is the number of unsuccessful attempts to refresh an invalidated page from the GBP. These requests will result in a physical I/O to DASD to get the page. (**NOTE**: This counter will be input to question #11 on the GBP performance spreadsheet, and will be used to calculate the hit ratio for pages that were cross-invalidated in the local buffer pools).

DATA RETURNED' under 'DUE TO DATA NOT IN POOL' reports the number of successful reads from the GBP when the requested page was not in the local pool. This is the second part of the 'hit ratio' of the GBP, which should be as high as possible for performance reasons.

The **DSNB775I** counters relate to the '*register page list'* function. See *Messages and Codes* for an explanation of these counters.

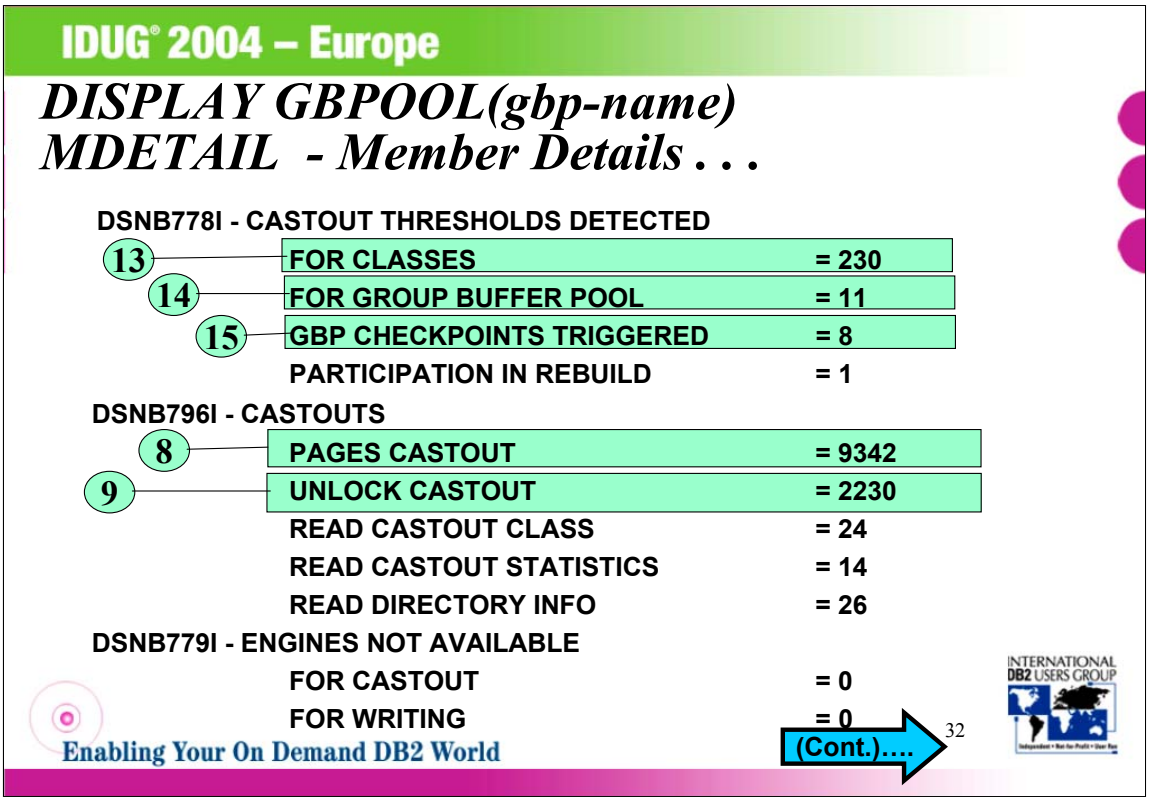


DSNB789I reports the CF interactions with the CF when '*register page list'* is active. This is true when DB2 is running on a service level of MVS 5.2 or higher, and in a CF with a CFLEVEL of 2 or higher.

- *REGISTER PAGE LIST* - Number of RPL requests to the CF to register a list of pages (when PREFETCH - sequential or list - is in effect.)
- *RETRIEVE CHANGED PAGES* - CF requests to retrieve pages marked as 'changed' in the CF as a result of feedback from registering that page.
- *FAILED READS DUE TO LACK OF STORAGE* - Number of CF reads that failed because of lack of storage for a directory entry. (**NOTE**: This value will be input to question #3 on the GBP performance spreadsheet.

DSNB776I and **DSNB777I** list the counters for writes to the GBP. Both clean and changed page counters are given.

The counter for '*FAILED DUE TO LACK OF STORAGE'* should, of course, always be zero. Failed writes will result in LPL entries, which means some data will not be available until recovered later.



DSNB778I reports on the thresholds that were detected by this member, as well as the number of GBP checkpoints triggered, and how many times, if any, this member participated in a rebuild of the GBP. (**NOTE**: The castout threshold values for classes, group buffer pool, and GBP checkpoints will be used as input to questions 13, 14, and 15 of the GBP performance spreadsheet).

DSNB796I gives statistics on castout activity: (**NOTE**: The first two counters, 'PAGES CASTOUT' and 'UNLOCK CASTOUT', will be used as input to questions 8 and 9 in the GBP performance spreadsheet).

- PAGES CASTOUT - Changed pages written out by this member.
- UNLOCK CASTOUT - Number of unlock requests to release locks on pages castout when the I/O completes.
- READ CASTOUT CLASS - Number of requests to the GBP to determine which pages of page sets are cached as changed pages, and therefore need to be cast out.
- READ CASTOUT STATISTICS - Number of times the GBP threshold was reached, and DB2 needs to find out which castout classes have changed pages.
- READ DIRECTORY INFO - Requests by the structure owner at GBP checkpoint time to find the oldest LRSN in the pool. May be issued several times per checkpoint.

Non-zero counters in **DSNB779I** indicate a shortage of system resources for writing and castout activity.

DISPLAY GBPOOL(gbp-name) MDETAIL - Member Details . . .

```
DSNB797I - OTHER INTERACTIONS
           REGISTER PAGE                        = 0
           UNREGISTER PAGE                      = 0
           DELETE NAME                          = 0
           READ STORAGE STATISTICS              = 0
           EXPLICIT CROSS INVALIDATIONS         = 0
DSNB764I - DUPLEXING STATISTICS FOR GBP08_SEC
           WRITES
             CHANGED PAGES                      = 9565
             FAILED DUE TO LACK OF STORAGE      = 0
DSNB793I - DELETE NAME LIST                     = 0
           READ CASTOUT STATISTICS              = 0
           DELETE NAME                          = 0
DSNB790I - DISPLAY FOR GROUP BUFFER POOL GBP08 IS COMPLETE
DSN9022I - DSNBICMD '-DISPLAY GBPOOL' NORMAL COMPLETION
```


DSNB797I reports several miscellaneous interactions with the CF:

- REGISTER PAGE - Number of times a directory entry was created because of IDRWI - i.e., the page set P-lock on the page set was downgraded from ‘S’ to ‘IS’, or ‘SIX’ to ‘IX’.
- UNREGISTER PAGE - Registered interest in locally cached pages was reversed, probably because DB2 locally re-used that page frame.
- DELETE NAME - DB2 deleted name and directory entries for a page set when it converted status from GBPD to non-GBPD, or was opened with a GBPCACHE ALL attribute.
- READ STORAGE STATS - DB2 requests statistics at GBP checkpoint time, as well as DIS GBPOOL GDETAIL execution.

DSNB764I reports writes to the secondary GBP.

DSNB793I: Several counters that are unique to the secondary GBP.

- READ CASTOUT STATISTICS records the number of times DB2 checked for ‘orphan’ pages in the secondary GBP.
- DELETE NAME - number of times DB2 deleted ‘orphaned’ data pages from the secondary GBP at ‘garbage collection’ time.

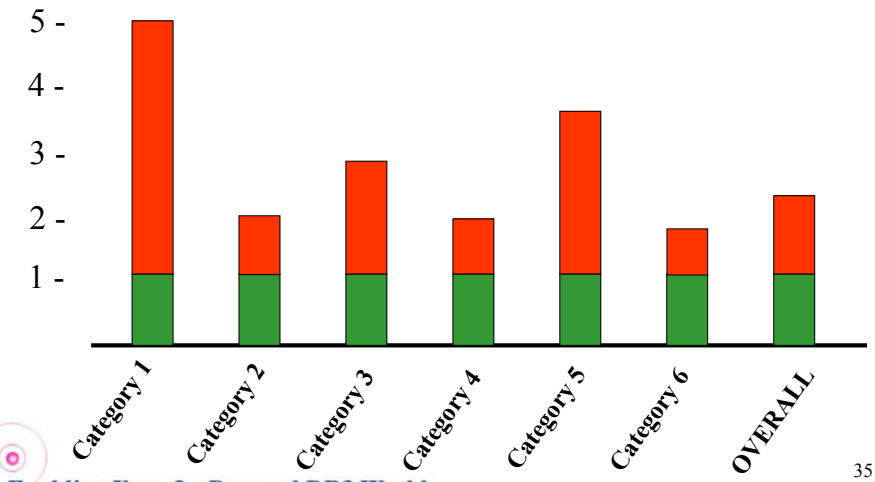
DISPLAY GROUPBUFFERPOOL Command

- **You don't need a lot of tools to start tuning GBPS!**
- **ONE COMMAND**
  - **A POWERFUL GBP TUNING TOOL!!!**
- **But What About all Those Numbers????**

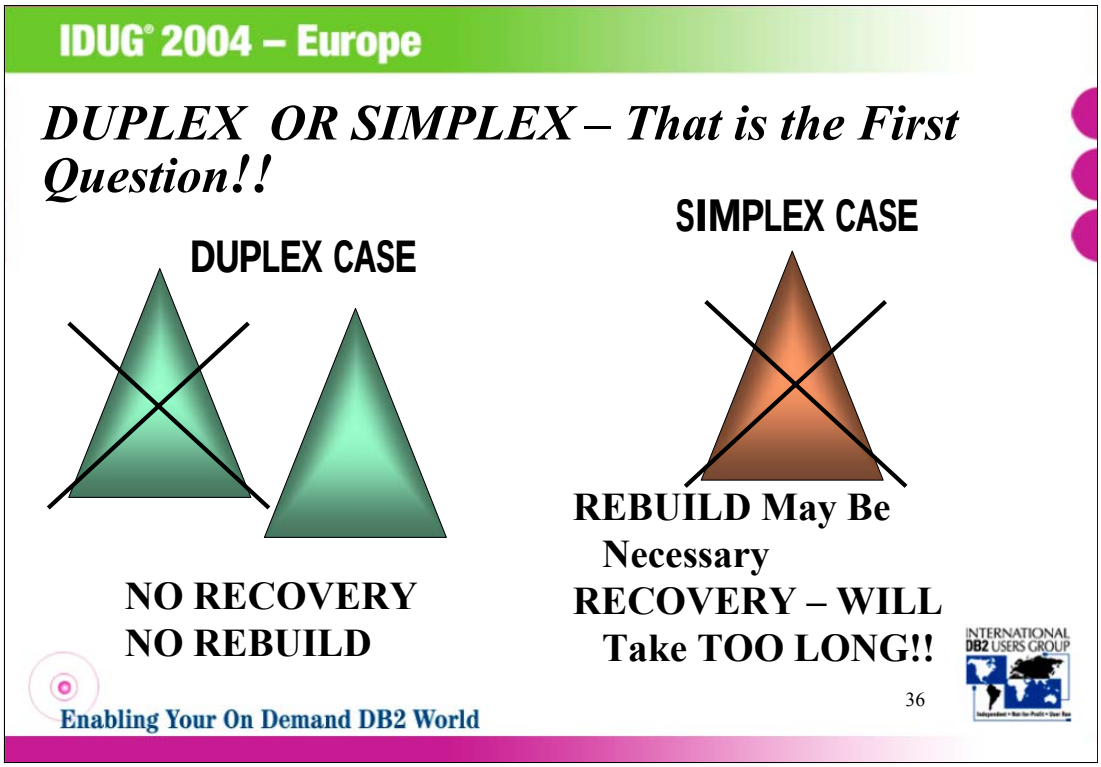
The DISPLAY GROUPBUFFERPOOL command provides a wealth of information useful for tuning purposes. Other monitoring and tuning products may assist in this effort, but the data provided by this one command is sufficient for a rather detailed tuning effort.

However, the numbers by themselves can be time-consuming to analyze. It would be helpful to have a graphical representation of the numbers we have discussed in this presentation, as well as the relationships between those numbers. One possible way to provide a graphical analysis of numbers is, of course, a spreadsheet. But before we look at the spreadsheet, let's review some of the key numbers provided by the DISPLAY GROUPBUFFERPOOL command, and the goals for those numbers we would like to tune for.

The Spreadsheet Graph – a Picture of Performance



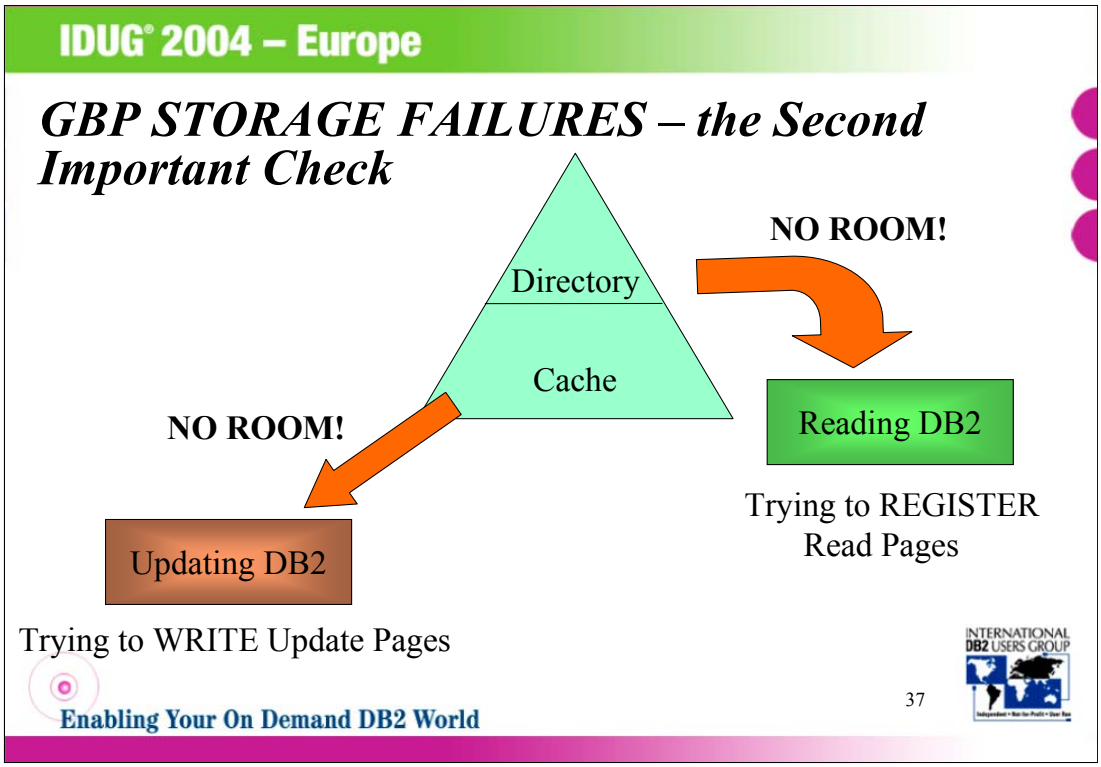
A way to represent performance numbers in the DISPLAY GROUP BUFFER POOL is shown on this visual. Each category is analyzed, and a number is calculated in the range of 1 to 5. Any result of 1 will graph as a green bar on the bar chart – the desired outcome. Any number higher than one will result in a bar graph showing red – the more red showing (5 being the worst), the more tuning effort must be extended. This is the approach that will be used in the GBP Performance spreadsheet discussed in this presentation. Of course, this methodology could be applied to any set of numbers being analyzed for being in proper ranges e.g., coupling facility performance figures, XCF response times, etc.



Even though it is primarily a recovery consideration, the matter of duplexing the GBPs is of highest importance. In the case of the loss of a simplexed GBP, data will be unavailable to the user community while DB2 attempts to rebuild the GBPs (assuming at least one DB2 is still connected to the old GBP, and its contents are valid), or, in the much worse case, where connectivity is completely lost, DB2 must first go through the process of **DAP** (Damage Assessment Processing). The function of this phase is to mark the page sets that had updated pages in the GBP that was lost. Then, DB2 will start the internal process of recovering those page sets by issuing START DB commands against the affected page sets. This process, though automatic, may take several hours.

Although the duplexing operation will cost some overhead (about 1 – 3% CPU), its advantage in a recovery scenario, if ever needed, make it an easy decision – YOU MUST DUPLEX YOUR GROUP BUFFER POOLS! Therefore, the first question in the GBP Performance spreadsheet asks if the GBP being analyzed is duplexed. If it is not, a red flag is immediately presented in the output graph.

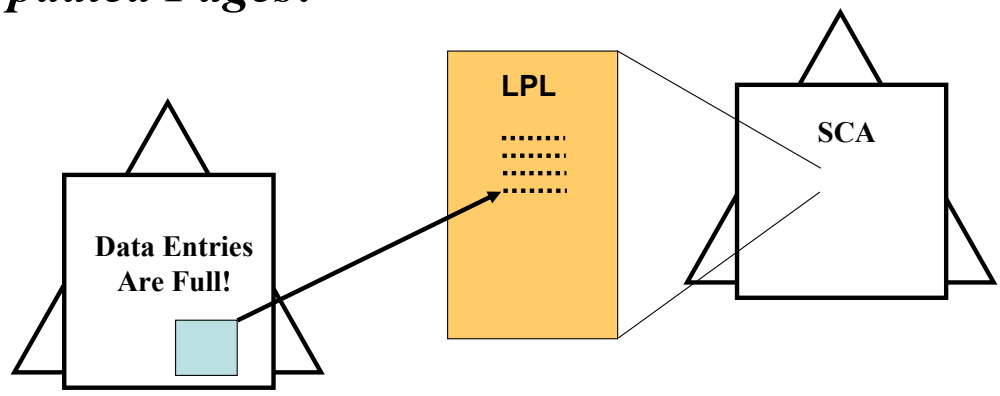
In our GBP spreadsheet, the matter of duplexing is the first category to be analyzed and graphed. As you might guess from these notes, any value other than DUPLEX will result in a red bar of maximum length! YOU MUST DUPLEX YOUR GROUP BUFFER POOLS!



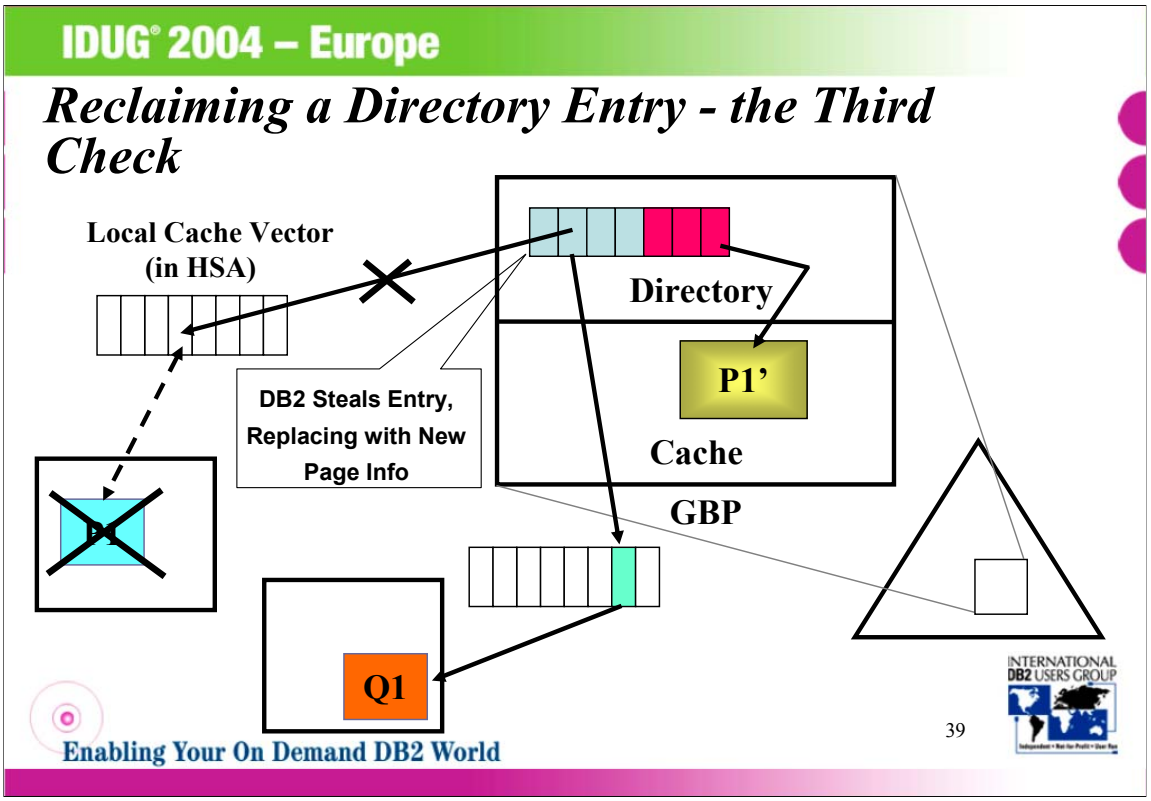
As we have seen in previous visuals, shortage of space in the group buffer pools can have serious negative impact on data sharing performance.

This check in the spreadsheet will chart the effect of running out of storage space in either/both of the parts of the GBP.

What if No Room in the GBP for Updated Pages?



When the FORCE AT COMMIT protocol is in effect, and either lack of space or a hardware error prevents the updated page being written, DB2 handles the error situation by placing the page on the **LPL** (Logical Page List), a list of pages that cannot be externalized. This list is stored in the **DBET** (Database Exception Table), which in data sharing mode is stored in the **SCA**. These pages must be recovered before re-use. They can be recovered with a START DATABASE command, using the SPACENAM parameter to recover the pages on the LPL. Obviously, we do not want to create a recovery situation by not having enough space in the data entry portion of the pool.



If a reading DB2 needs to register a page because of an IDRWI or GBPD situation, and the allocated part of the GBP is fully populated, DB2 needs to try to find a way to 'free up' a directory slot. It will scan those directory entries pointing to locally cached pages, and find the oldest one – the page that has not been referenced for 'awhile'. It will steal, or 'reclaim' that entry for the new page's information.

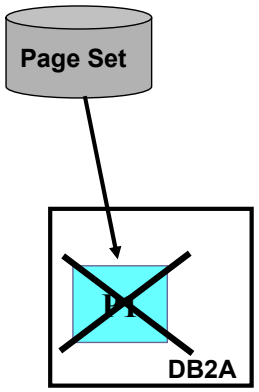
But what about the page the directory entry was pointing to (Page P1 in our example)? The CF now has no knowledge of where that page is located, and if a subsequent update would change that page, the CF could not cross-invalidate it, and we would have a data integrity exposure.

So, to provide the data integrity needed, the CF will cross-invalidate the page it was pointing to. This cross-invalidation is not due to an update, but is in fact due to a directory reclaim, or stealing. In our example, page 'P1' would be cross-invalidated.

The directory entry that was pointing to the LCV representing page P1, is now populated with the page id of the newly read page Q1, which could be in a different local buffer pool, in a different system.

But what happens if the first DB2 system needs to re-reference page P1? It hasn't changed, and it's still in that DB2's local buffer pool.

XIs (and I/Os!) – For the Wrong Reason!



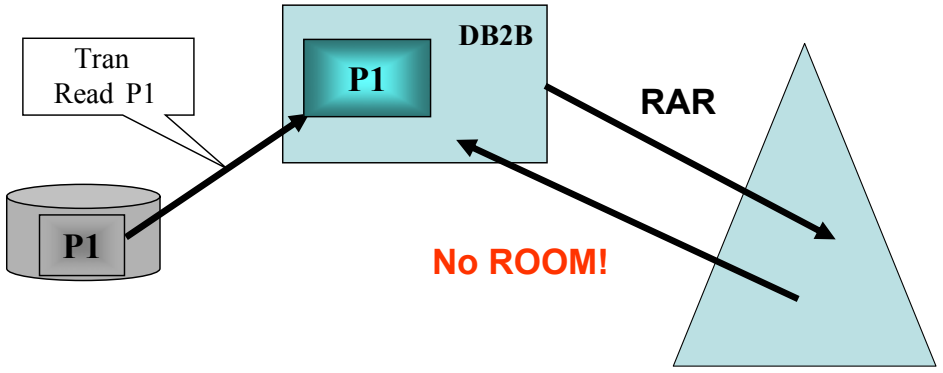
IF DB2A Needs to Re-reference Page P1:

- P1 is XI'ed
- Needs to be Refreshed
- Will Probably Require an I/O
- (A SYNCHRONOUS I/O!)
- And the Page Has NOT CHANGED!

When a local DB2 member re-references a locally cached page that belongs to a GBP page set, it must check its validity – i.e., has the LCV entry that points to this page been changed, indicating it has been updated by another member, (the good kind of cross-invalidation), or the LCV entry has been changed by the bad kind of cross-invalidation – that caused by a directory entry reclaim. Either of these reasons for cross-invalidation will cause DB2 to refresh the page. However, the cross-invalidation caused by a directory entry reclaim will probably cause a physical I/O – a SYNCHRONOUS I/O, to get the page. (The chances that the page is in the GBP in this case is very low – unlike the situation for a 'good' cross-invalidation, where the chance of the updated page that caused the cross-invalidation still being in the GBP is very high).

The worst part if this scenario is that when the page is read in from DASD, the contents will match the page already in the pool! DB2 just had to do a SYNCHRONOUS physical I/O for a page it already had cached! AND IT HASN"T CHANGED!

If a Directory Reclaim Not Possible, -904! For a Read!

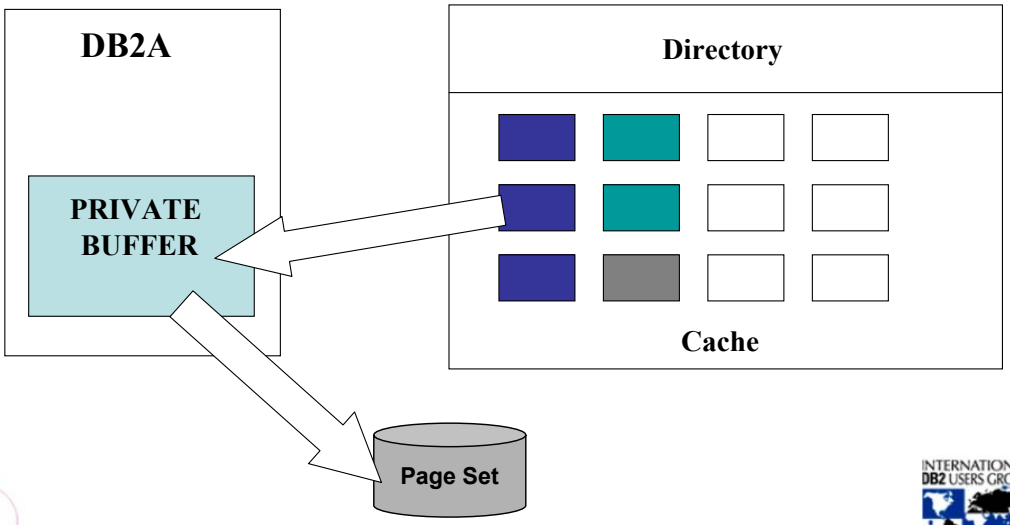


If a reading DB2 subsystem tries to register a page it is reading with a **RAR** (Read and Register) command, the coupling facility may return an error message stating that the directory portion of the GBP is full. If no directory entries are available for stealing (e.g., all populated directory entries point to updated pages in the GBP), DB2 needs to return a -904 error code to the reading transaction.

This will be visible in the DISPLAY GROUP BUFFER POOL command under the heading “FAILED READS DUE TO LACK OF STORAGE”.

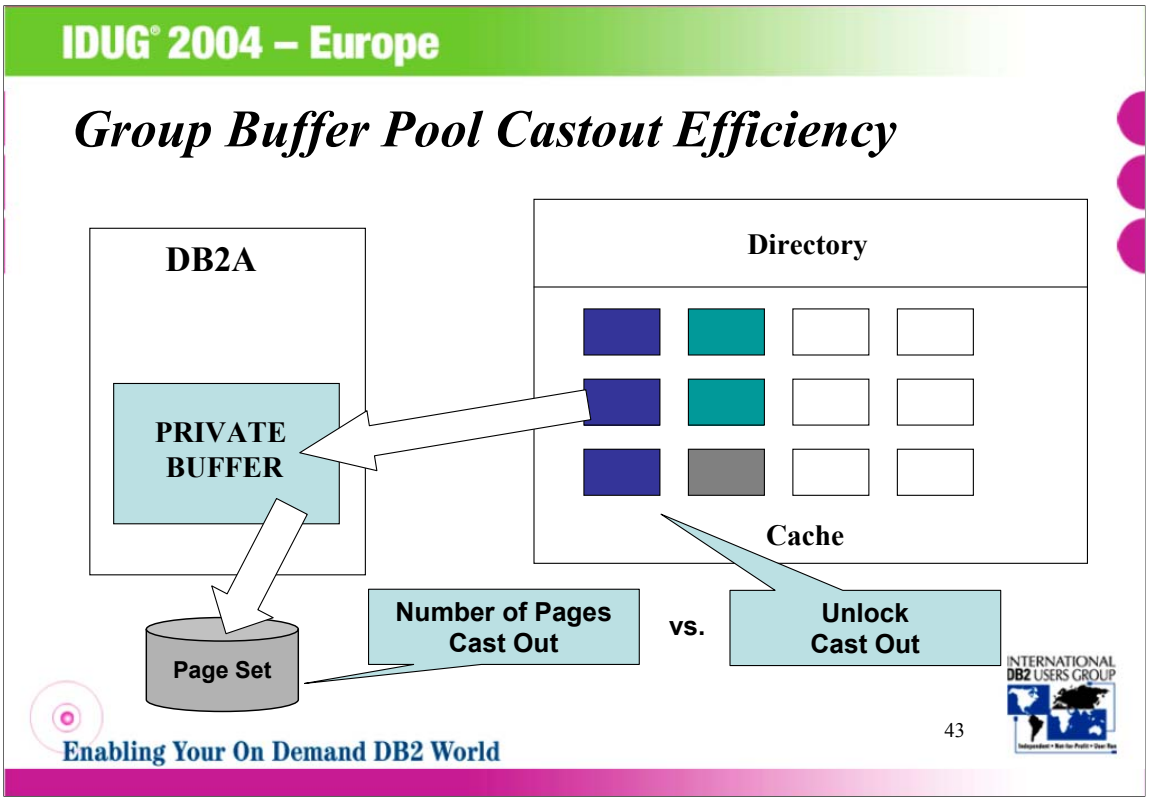
This is certainly not an acceptable situation – the transaction may just be trying to satisfy an on-line request for an account balance.

Group Buffer Pool Castout – the Fourth Check



Since there is no physical connection from the coupling facility to DASD, DB2 subsystems must do the work of writing updated pages to DASD. The DB2 who is the current castout owner reads updated pages into a private buffer in the DBM1 address space and does the physical I/O to write the updated pages to DASD. The castout ownership is assigned when a DB2 subsystem first opens a page set for update.

In the next visual, we will discuss the various thresholds that control this process of castout.

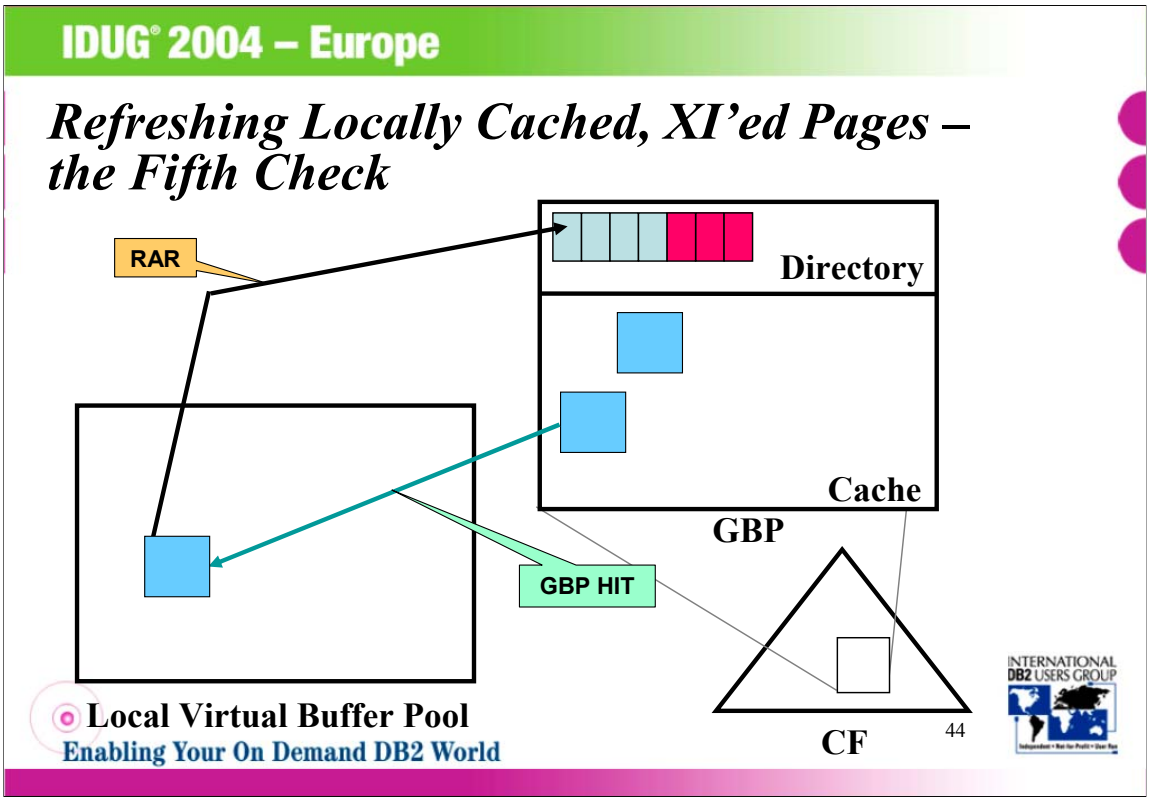


Since there is no physical connection from the coupling facility to DASD, DB2 subsystems must do the work of writing updated pages to DASD. The DB2 who is the current castout owner reads updated pages into a private buffer in the DBM1 address space and does the physical I/O to write the updated pages to DASD. The castout ownership is assigned when a DB2 subsystem first opens a page set for update.

The castout process is controlled by three castout thresholds, class threshold, GBP threshold, and GBP checkpoint processing.

The castout efficiency is measured by comparing the number of pages castout (reported on a D GBPOOL command MDETAIL report), to the UNLOCK CASTOUT number reported on the same command output. The greater the ratio between PAGES CASTOUT, and UNLOCK CASTOUT, the more efficiently the castout is working.

The spreadsheet assumes that a value of 70% or better is acceptable, i.e., UNLOCK CASTOUTs should be equal to or less than 70% of the PAGES castout.



Several of the major interactions of DB2 and the GBP address the directory portion of the GBP. The first is called an **RAR** (Read and Register) request. It is issued by a reading DB2 member, when reading from a GBPD page set. It is really asking for two things, a page that it needs, and the registering of that page in the directory (since the page set is GBPD, that is what a READER has to do). This visual illustrates a successful RAR (Read and Register) request – the DB2 found the page (a GBP HIT), and was successful in registering the page in the directory.

When this scenario happens in response to a locally cached page being cross-invalidated, and needing to be re-referenced, it is a good performance consideration. Reading an updated page from the GBP is much faster than having to do an I/O from DASD.

The fifth category of the worksheet maps the hit ratio for this scenario – the higher the hit ratio, the better performance this part of data sharing can provide. Obviously, a hit ratio of 100% would be optimum, but this is rarely possible in a production environment. The worksheet provides for this by allowing the user to enter the percentage of acceptable GBP misses, i.e., SYNCH READS, DATA NOT RETURNED.

Castout Thresholds – Across the Group – the Sixth Check

DSNB778I - CASTOUT THRESHOLDS DETECTED
FOR CLASSES = 230
FOR GROUP BUFFER POOL = 11
GBP CHECKPOINTS TRIGGERED = 8

Total CLASS Thresholds Triggered

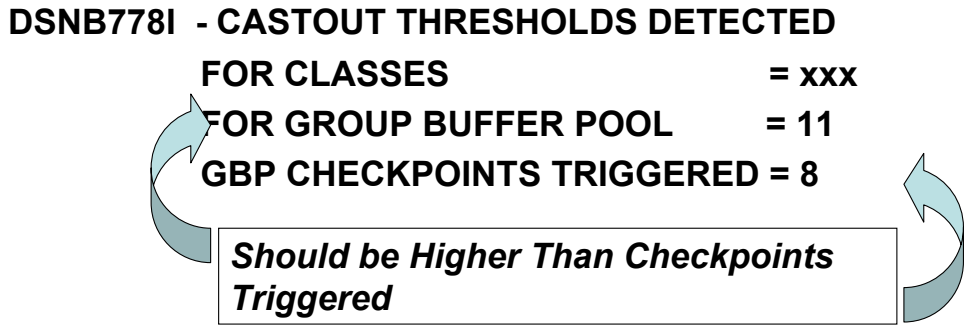
Structure Owner Responsibility

Across the data sharing group, for a heavy updating environment, The CLASS threshold should be hit more often than the GBP threshold, which should be triggered more often than the GBP Checkpoint. This will 'trickle' the castouts more evenly.

DSNB778I of the D GBPOOL MDETAIL command reports on the thresholds that were detected by this member, as well as the number of GBP checkpoints triggered, and how many times, if any, this member participated in a rebuild of the GBP.

When these reported thresholds are considered across the group, they should be triggered according to the sequence shown on the chart – class, GBP, and lastly GBP checkpoint. This sequence of castout thresholds will cause the I/O of castout to be 'trickled' out – not depending on a large burst of I/O at GBP checkpoint time.

Castout Thresholds – For This Member (as the Structure Owner)



For a member of the data sharing group that is a structure owner (i.e., the member that first connected to the GBP with UPDATE intent), the GBP threshold should be triggered at least as many times as the GBP checkpoint in an ideal situation (handling the castout for a very heavy updating data sharing group). This will even out the I/O load on this member, without causing large I/O bursts at the GBP checkpoint time.

The class thresholds may also be hit on this member – but it can't be expected to be very many. Class thresholds are set as first UPDATE transactions cause a page set to be OPENed and assigned to a new class. These thresholds will typically be spread across the group – which is by design a way to spread the castout I/O work across the group.

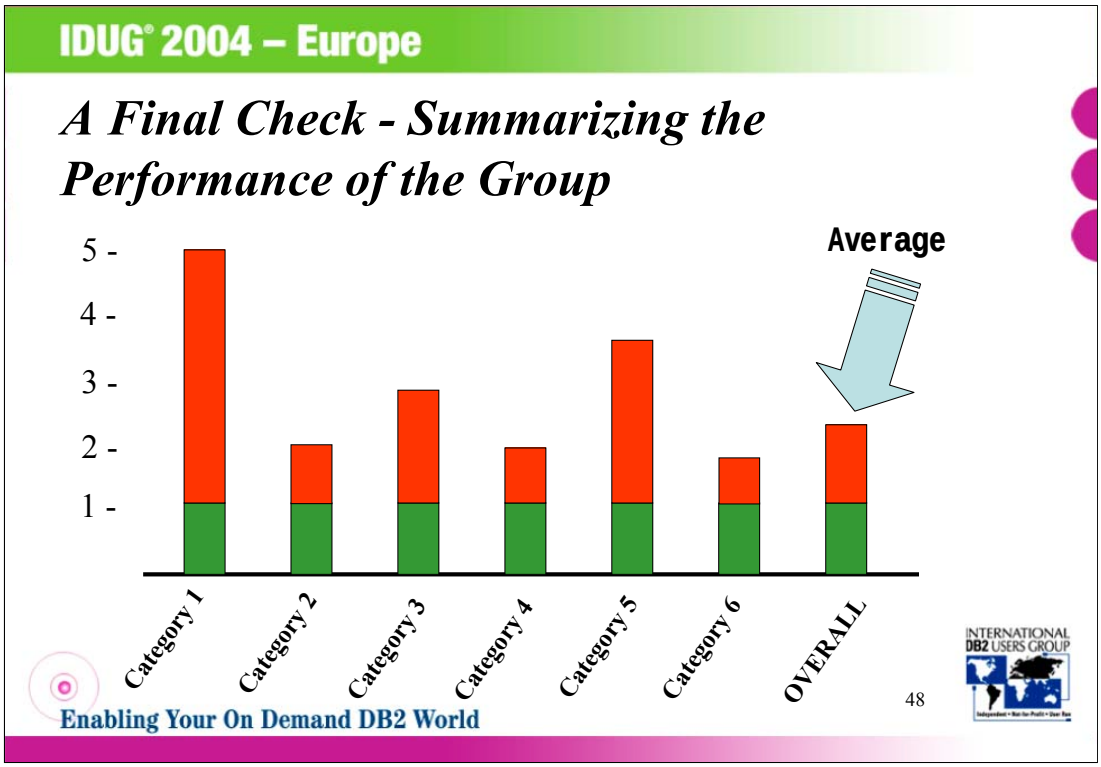
Castout Thresholds – For the Non-Structure Owner

DSNB778I - CASTOUT THRESHOLDS DETECTED
FOR CLASSES = xxx
FOR GROUP BUFFER POOL = 0
GBP CHECKPOINTS TRIGGERED = 0

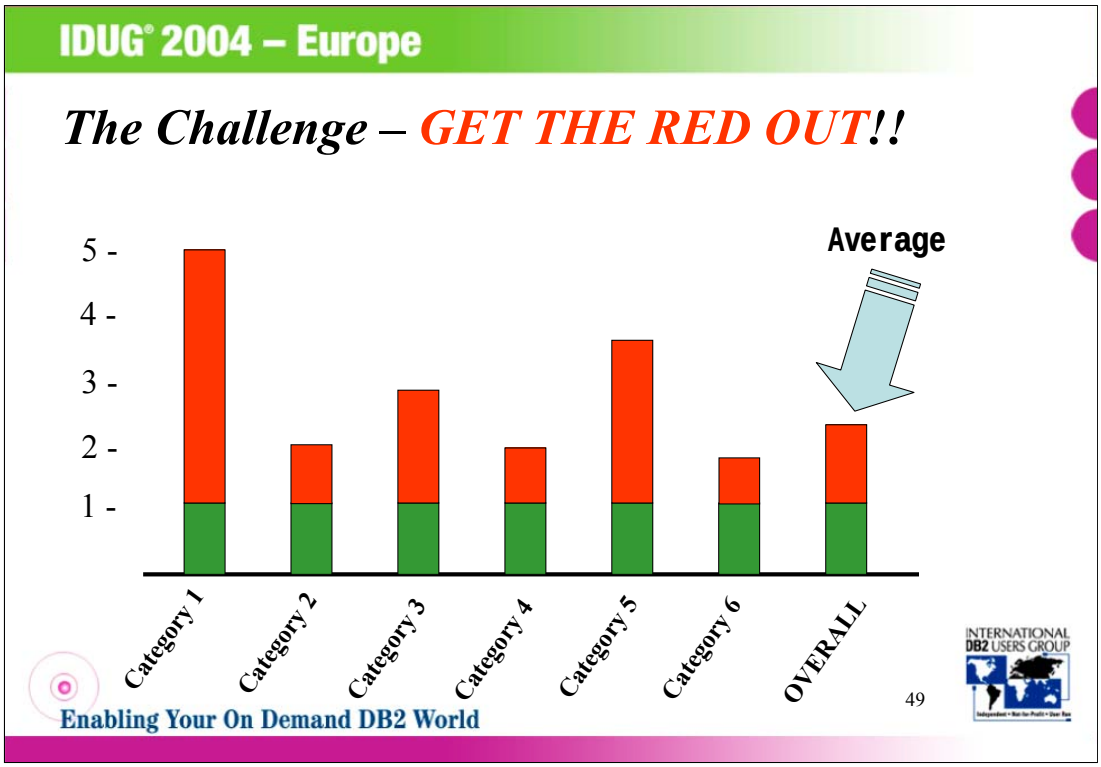
Each Non-Structure Owner Should Have Some Castouts

For a member of the data sharing group that is a non-structure owner (i.e., the member that first opened a page set with UPDATE intent for a page set assigned to this GBP), the class threshold should be triggered at least as many times as the GBP threshold in an ideal situation (handling the castout for a very heavy updating data sharing group). This will even out the I/O load on this member, without causing large I/O bursts at the GBP threshold at checkpoint time.

The class thresholds would ideally be hit on each member – but it can't be accurately predicted. Class thresholds are set as first UPDATE transactions cause a page set to be OPENed and assigned to a new class. These thresholds will typically be spread across the group – which is by design a way to spread the castout I/O work across the group.



The last bar on the GBP performance check spreadsheet is an overall look at the performance of the data sharing group. It is just an average of the six performance checks we have looked at. Each category has been analyzed, and a number calculated in the range of 1 to 5. Any result of 1 will graph as a green bar on the bar chart – the desired outcome. Any number higher than one will result in a bar graph showing red – the more red showing (5 being the worst), the more tuning effort must be extended. Of course, this methodology could be applied to any set of numbers being analyzed for being in proper ranges e.g., coupling facility performance figures, XCF response times, etc.



If this spreadsheet has been used to analyze a busy GBP, and lots of red is showing, there is performance tuning to be done! INITSIZEs may need to be increased, RATIOs may need to be adjusted up or down, thresholds may need to be lowered etc. (**NOTE:** Version 8 castout threshold defaults have been lowered significantly!).

This charting technique can be used as a daily or weekly check, a progress check over time, or simply as a reporting tool to show management that performance work should be done, or to show management that tuning work has been done!

Tuning Example 1 – GBP 12 GDETAIL

-DISPLAY GROUPBUFFERPOOL (GBP12) GDETAIL

```
DSNB750I -DISPLAY FOR GROUP BUFFER POOL GBP12 FOLLOWS
DSNB755I -DB2 GROUP BUFFER POOL STATUS
                CONNECTED                                = YES
                CURRENT DIRECTORY TO DATA RATIO          = 6
                PENDING DIRECTORY TO DATA RATIO          = 6
                CURRENT GBPCACHE ATTRIBUTE               = YES
                PENDING GBPCACHE ATTRIBUTE               = YES
DSNB756I -CLASS CASTOUT THRESHOLD                        = 10%
                GROUP BUFFER POOL CASTOUT THRESHOLD      = 50%
                GROUP BUFFER POOL CHECKPOINT INTERVAL    = 20 MINUTES
                RECOVERY STATUS                          = NORMAL
                AUTOMATIC RECOVERY                       = Y
```

(Cont.)

This is the first screen showing the DISPLAY GBP GDETAIL for our sample GBP tuning exercise. The production data sharing group this was drawn from was a SAP shop, running on TREX hardware, with the latest ESS DASD from IBM.

As noted on the chart, some of the defaults have been changed – which normally is a good thing. Someone has done some tuning in the past – the directory ratio is 6 (up from the default of 5), and the standard GBP CHKPT time has been changed to 20 minutes (up from 8).

Tuning Example 1 – GBP 12 GDETAIL . .

```
DSNB757I -MVS CFRM POLICY STATUS FOR DSNDB0A_GBP12 = NORMAL
          MAX SIZE INDICATED IN POLICY          = 281,344 KB
          DUPLEX INDICATOR IN POLICY            = ENABLED
(1)       CURRENT DUPLEXING MODE                = DUPLEX
          ALLOCATED                             = YES
DSNB758I  ALLOCATED SIZE                        = 281,344 KB
          VOLATILITY STATUS                     = VOLATILE
          REBUILD STATUS                        = DUPLEXED
          CFNAME                                = CF03
          CFLEVEL                               = 12
          NUMBER OF DIRECTORY ENTRIES           = 276,827
          NUMBER OF DATA PAGES                  = 46,137
          NUMBER OF CONNECTIONS                 = 2
```

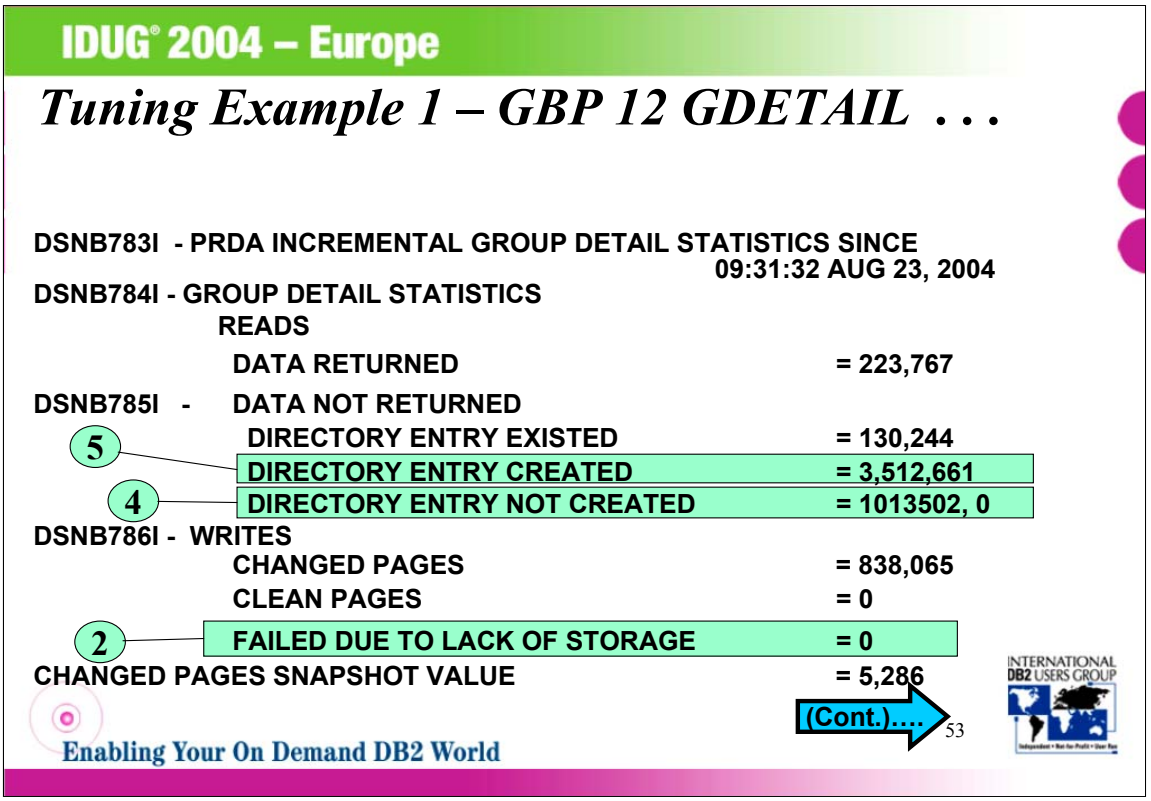
(Cont.)….

This GBP is a rather large one, and is in duplexed mode.

Tuning Example 1 – GBP 12 GDETAIL . .

```
DSNB798I - LAST GROUP BUFFER POOL CHECKPOINT
                                        10:29:42 AUG 23, 2004
          GBP CHECKPOINT RECOVERY LRSN      = BBB63EBCC782
12        STRUCTURE OWNER                   = PRD1
DSNB799I -  SECONDARY GBP ATTRIBUTES
          ALLOCATED SIZE                    = 281,600 KB
          VOLATILITY STATUS                 = VOLATILE
          CFNAME                            = CF04
          CFLEVEL                           = 12
          NUMBER OF DIRECTORY ENTRIES       = 276,827
          NUMBER OF DATA PAGES              = 46,137
                                            (Cont.)....
```


One of the important items on this display is the structure owner – this is needed to analyze the efficiency of the castout process.



Message **DSNB783I** displays the start of the time interval being reported. It can be reset with the INTERVAL parameter.

The **DSNB784I** READS counter shows the 'Read Hit' counter - the number of times DB2 requested a page from the GBP, and the read was successful. This is an important performance number, as reads from the GBP are much faster than reads from DASD.

The **DSNB785I** message breaks out unsuccessful reads, and whether or not directory entries were created at the time of the read request. The second counter in 'DIRECTORY ENTRY NOT CREATED' **SHOULD ALWAYS BE ZERO,** as it reports number of directories not created because of storage shortage. (**NOTE**: The number in the 'DIRECTORY ENTRY CREATED' counter will be input to question #5 in the GBP performance spreadsheet, and the second number in the 'DIRECTORY ENTRY NOT CREATED' entry will be input to question #4 of the GBP performance spreadsheet.)

The **DSNB786I** counters report the number of writes to the GBP. 'CHANGED' reports the updated pages written when IDRWI existed. A non-zero counter in 'CLEAN PAGES' indicates that GBPCACHE ALL is specified either for the page set, or the GBP.

The 'FAILED DUE TO LACK OF STORAGE' counter **SHOULD ALWAYS BE ZERO**. A non-zero number here indicates DB2 had to put pages on the LPL, making them unavailable until recovered. (**NOTE**: This value will be input to question #2 on the GBP performance spreadsheet.)

The 'CHANGED PAGES SNAPSHOT' value is the number of changed pages in the pool at the time of the command.

Tuning Example 1 – GBP 12 GDETAIL . .

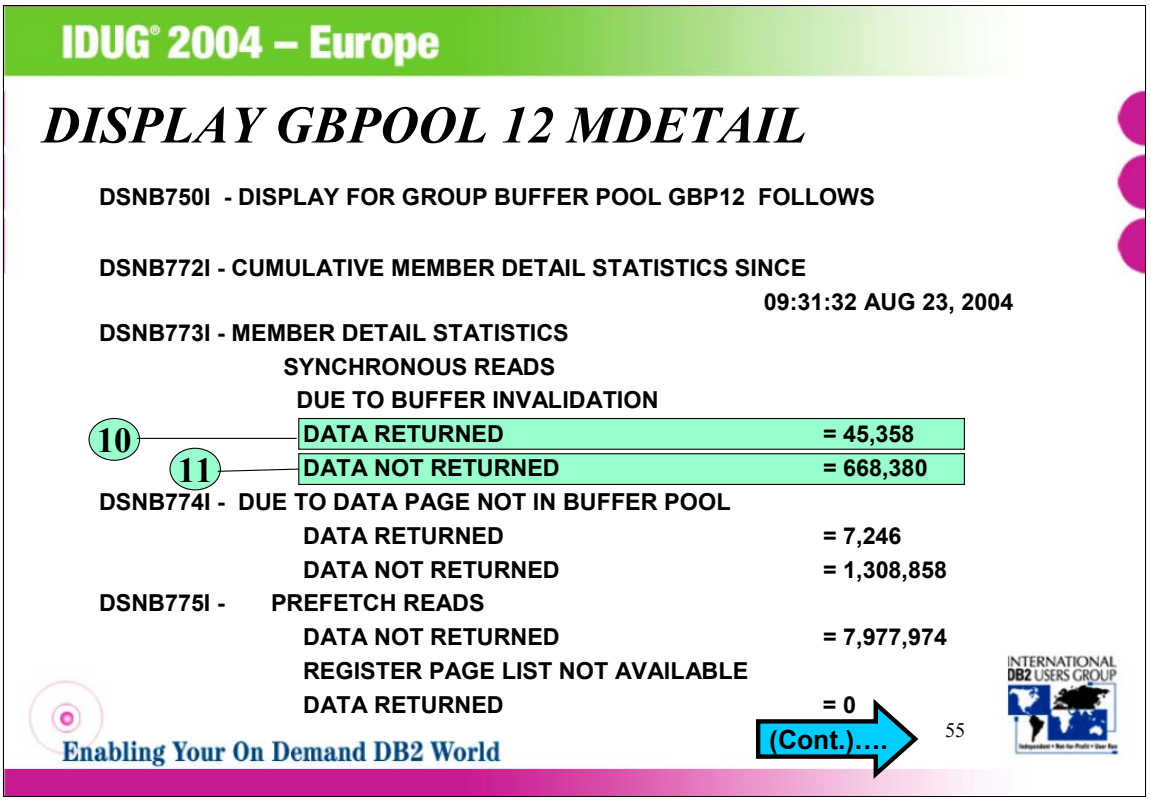
```
DSNB787I -  RECLAIMS
        6       FOR DIRECTORY ENTRIES          = 17,239,281
                FOR DATA ENTRIES               = 8,898
                CASTOUTS                       = 340,491
DSNB788I -  CROSS INVALIDATIONS
        7       DUE TO DIRECTORY RECLAIMS      = 17,642,455
                DUE TO WRITES                  = 108,264
                EXPLICIT                       = 0
DSNB762I - DUPLEXING STATISTICS FOR GBP08_SEC
                WRITES
                CHANGED PAGES                  = 838,065
                FAILED DUE TO LACK OF STORAGE  = 0
                CHANGED PAGES SNAPSHOT VALUE   = 5,286
DSNB790I  -  DISPLAY FOR GROUP BUFFER POOL GBP12 IS COMPLETE
DSN9022I - DSNB1CMD '-DISPLAY GBPOOL' NORMAL COMPLETION
```


DSNB787I displays counters for *Reclaims*, the number of times DB2 needed to 'steal' entries for new requests, whether for a directory entry (**NOTE**: Used as input to question #6 on the GBP performance spreadsheet), or a data entry. 'CASTOUTS' is the number of times changed pages were externalized to DASD, triggered by the thresholds.

DSNB788I reports the number of cross-invalidations performed by the CF. If the first counter, 'DUE TO DIRECTORY RECLAIMS', is too high, and there is a large amount of re-referencing of locally cached data pages, there is a high performance cost. The local DB2s have to refresh the page from DASD, and after suffering the overhead of a physical I/O, find the page just read is EXACTLY the same as the cross-invalidated one! (**NOTE**: This counter is input to question #7 on the GBP performance spreadsheet).

The 'EXPLICIT' counter is the number of explicit XIs issued by the DB2s, because of a GBPCACHE NONE or SYSTEM specification on a page set.

The **DSNB762I** counters repeat the primary counters as they apply to the secondary GBP. In most cases, they should be equal; however, because of timing problems, they are not always exactly the same.



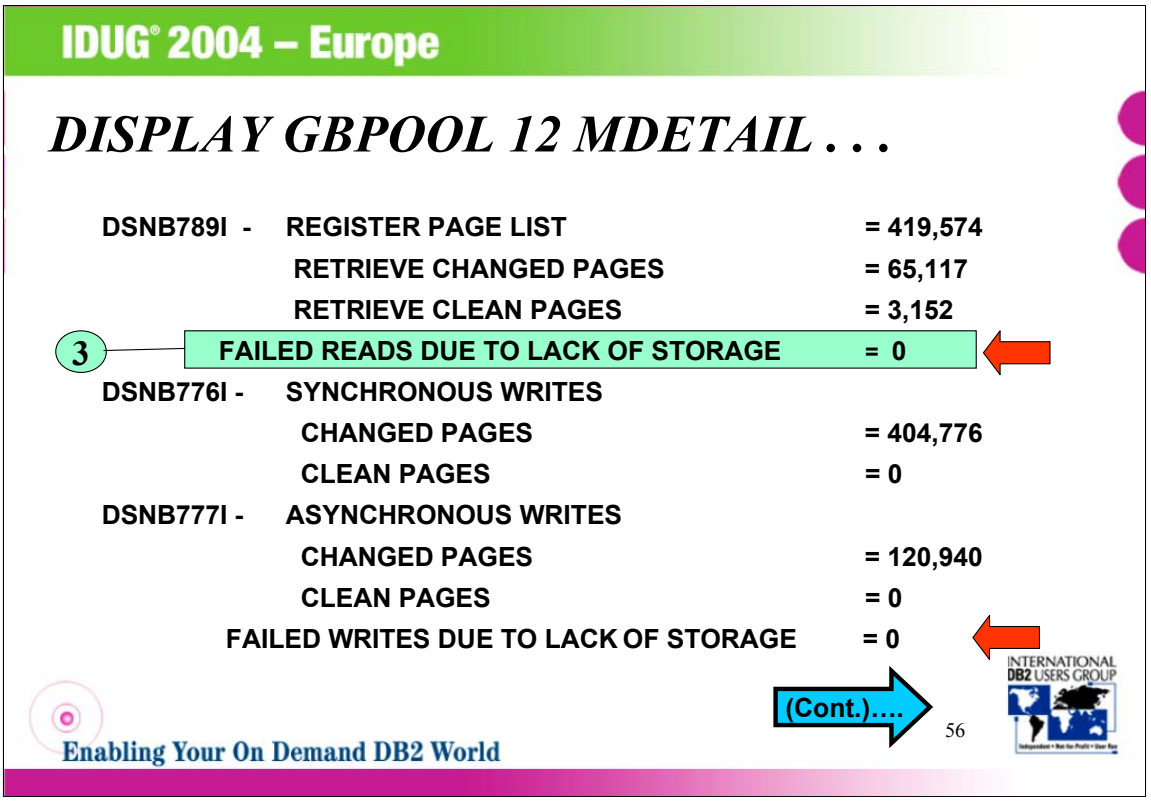
The **MDETAIL** version of the DISPLAY GBPOOL command reports detailed statistics on a specific GBP as seen by a member, not the entire group.

DSNB773I reports the synchronous reads issued by this DB2, and their results. 'Synchronous' in this context means that the reads were done under the user's execution unit, synchronous with processing of the user's transaction.

'DATA RETURNED' under 'DUE TO BUFFER INVALIDATION' reports the number of successful reads from the GBP when the member's locally cached pages were invalidated. This is a 'hit ratio' of the GBP, which should be as high as possible for performance reasons. (**NOTE**: This number will be input to question #10 on the GBP performance spreadsheet). 'DATA NOT RETURNED', on the other hand, is the number of unsuccessful attempts to refresh an invalidated page from the GBP. These requests will result in a physical I/O to DASD to get the page. (**NOTE**: This counter will be input to question #11 on the GBP performance spreadsheet, and will be used to calculate the hit ratio for pages that were cross-invalidated in the local buffer pools).

'DATA RETURNED' under 'DUE TO DATA PAGE NOT IN BUFFER POOL' reports the number of successful reads from the GBP when the requested page was not in the local pool. This is the second part of the 'hit ratio' of the GBP, which should be as high as possible for performance reasons.

The **DSNB775I** counters relate to the '*register page list'* function. See *Messages and Codes* for an explanation of these counters.

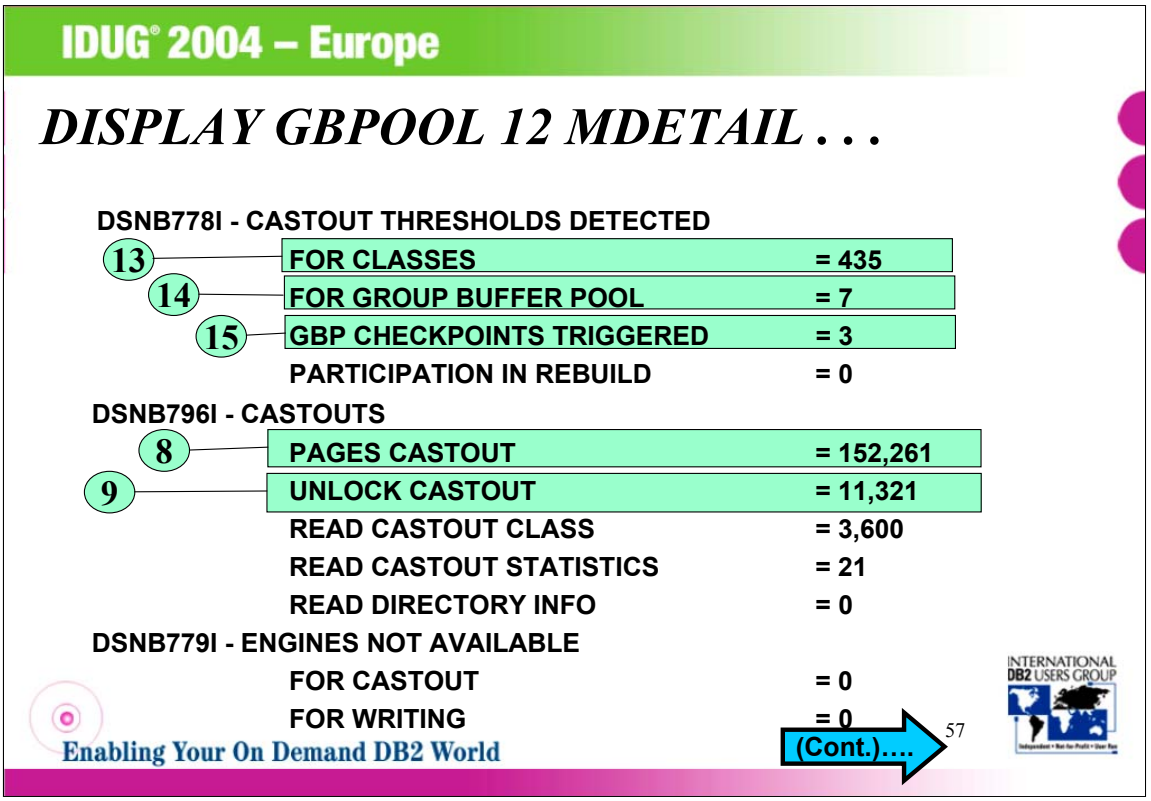


DSNB789I reports the CF interactions with the CF when 'register page list' is active. This is true when DB2 is running on a service level of MVS 5.2 or higher, and in a CF with a CFLEVEL of 2 or higher.

- **'**REGISTER PAGE LIST' - Number of RPL requests to the CF to register a list of pages (when PREFETCH - sequential or list - is in effect.)
- 'RETRIEVE CHANGED PAGES' - CF requests to retrieve pages marked as 'changed' in the CF as a result of feedback from registering that page.
- 'FAILED READS DUE TO LACK OF STORAGE' - Number of CF reads that failed because of lack of storage for a directory entry. (**NOTE**: This value will be input to question #3 on the GBP performance spreadsheet.

DSNB776I and **DSNB777I** list the counters for writes to the GBP. Both clean and changed page counters are given.

The counter for 'FAILED DUE TO LACK OF STORAGE' should, of course, always be zero. Failed writes may result in LPL entries, which means some data will not be available until recovered later.

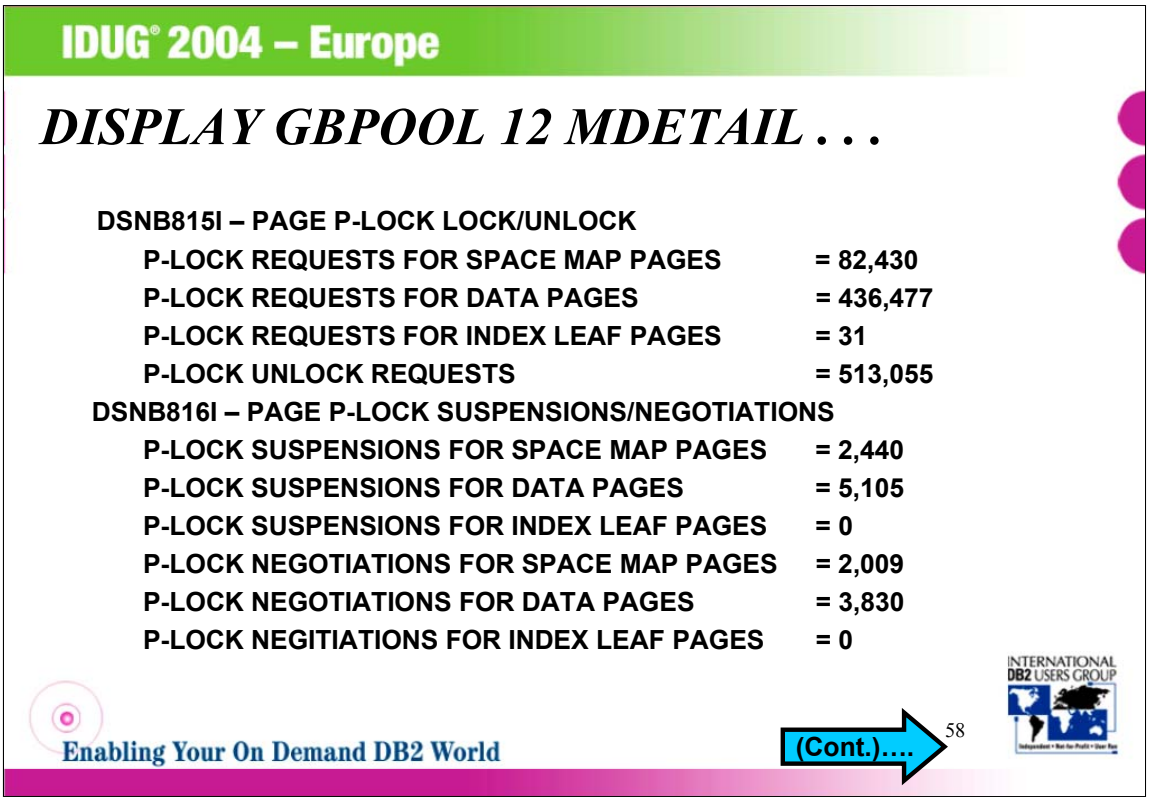


DSNB778I reports on the thresholds that were detected by this member, as well as the number of GBP checkpoints triggered, and how many times, if any, this member participated in a rebuild of the GBP. (**NOTE**: The castout threshold values for classes, group buffer pool, and GBP checkpoints will be used as input to questions 13, 14, and 15 of the GBP performance spreadsheet).

DSNB796I gives statistics on castout activity: (**NOTE**: The first two counters, 'PAGES CASTOUT' and 'UNLOCK CASTOUT', will be used as input to questions 8 and 9 in the GBP performance spreadsheet).

- PAGES CASTOUT - Changed pages written out by this member.
- UNLOCK CASTOUT - Number of unlock requests to release locks on pages cast out when the I/O completes.
- READ CASTOUT CLASS - Number of requests to the GBP to determine which pages of page sets are cached as changed pages, and therefore need to be cast out.
- READ CASTOUT STATISTICS - Number of times the GBP threshold was reached, and DB2 needs to find out which castout classes have changed pages.
- READ DIRECTORY INFO - Requests by the structure owner at GBP checkpoint time to find the oldest LRSN in the pool. May be issued several times per checkpoint.

Non-zero counters in **DSNB779I** indicate a shortage of system resources for writing and castout activity.



DSNB815I reports on the page P-lock requests to this GBP. These numbers reflect the kind of page sets that are assigned to the pool – table spaces, or index spaces.

DSNB816I gives statistics on negotiations and suspensions involving page P-locks.

DISPLAY GBPOOL 12 MDETAIL . . .

```
DSNB797I - OTHER INTERACTIONS
              REGISTER PAGE                          = 85,593
              UNREGISTER PAGE                        = 251,585
              DELETE NAME                            = 123
              READ STORAGE STATISTICS                = 844
              EXPLICIT CROSS INVALIDATIONS           = 0
DSNB764I - DUPLEXING STATISTICS FOR GBP08_SEC
              WRITES
                CHANGED PAGES                        = 525,711
                FAILED DUE TO LACK OF STORAGE        = 0
                ASYNCHRONOUS COMPLETION CHECKS= 207,401
DSNB793I - DELETE NAME LIST                          = 11,322
              READ CASTOUT STATISTICS                = 0
              DELETE NAME                            = 118
DSNB790I - DISPLAY FOR GROUP BUFFER POOL GBP12 IS COMPLETE
DSN9022I - DSNBICMD '-DISPLAY GBPOOL' NORMAL COMPLETION
```


DSNB797I reports several miscellaneous interactions with the CF:

- REGISTER PAGE - Number of times a directory entry was created because of IDRWI - i.e., the page set P-lock on the page set was downgraded from ‘S’ to ‘IS’, or ‘SIX’ to ‘IX’.
- UNREGISTER PAGE - Registered interest in locally cached pages was reversed, probably because DB2 locally re-used that page frame.
- DELETE NAME - DB2 deleted name and directory entries for a page set when it converted status from GBPD to non-GBPD, or was opened with a GBPCACHE ALL attribute.
- READ STORAGE STATS - DB2 requests statistics at GBP checkpoint time, as well as DIS GBPOOL GDETAIL execution.

DSNB764I reports writes to the secondary GBP.

DSNB793I: Several counters that are unique to the secondary GBP.

- READ CASTOUT STATISTICS records the number of times DB2 checked for ‘orphan’ pages in the secondary GBP.
- DELETE NAME - number of times DB2 deleted ‘orphaned’ data pages from the secondary GBP at ‘garbage collection’ time.

EXCEL Spreadsheet Input

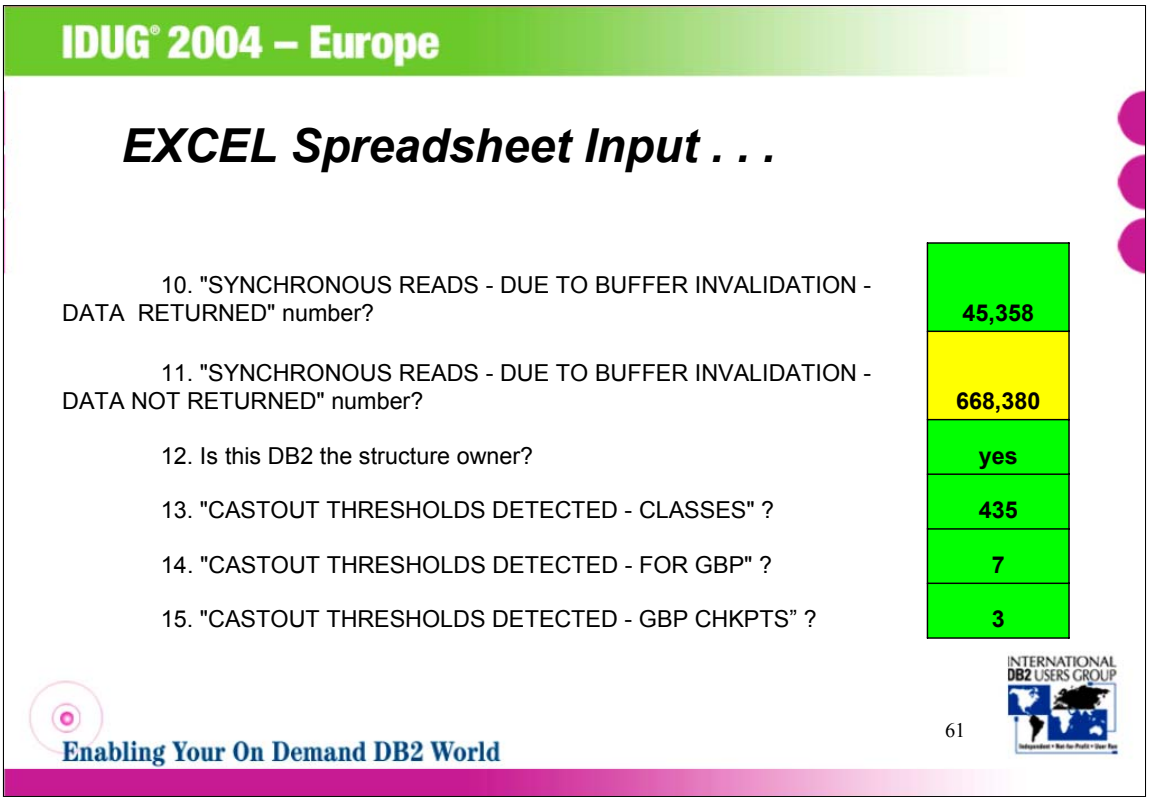
GBP 12 :	1. What is your "CURRENT DUPLEXING MODE"?	duplex
	2. "WRITES FAILED DUE TO LACK OF STORAGE" number?	0
	3. "FAILED READS DUE TO LACK OF STORAGE" number?	0
	4. "DIRECTORY ENTRY NOT CREATED yyyyyyy,X" X value?	0
	5. "READS, DATA RETURNED, DIRECTORY ENTRY CREATED" ?	3,512,661
	6. "RECLAIMS FOR DIRECTORY ENTRIES" ?	17,239,281
	7. "XIs DUE TO DIRECTORY RECLAIMS"?	17,642,455
	8. "PAGES CASTOUT" number?	152,261
	9. "UNLOCK CASTOUT" number?	11,321

This chart shows the values from the DISPLAY GBP commands for GBP 12 that we just looked at.

The read input cell for question 7 points to a potential problem for this pool – over 17 million XIs due to directory reclaims. (Remember, we are looking at only an hour's worth of data)!

The castout process seems to be working well – the ratio of pages castout to UNLOCK castout is very good.

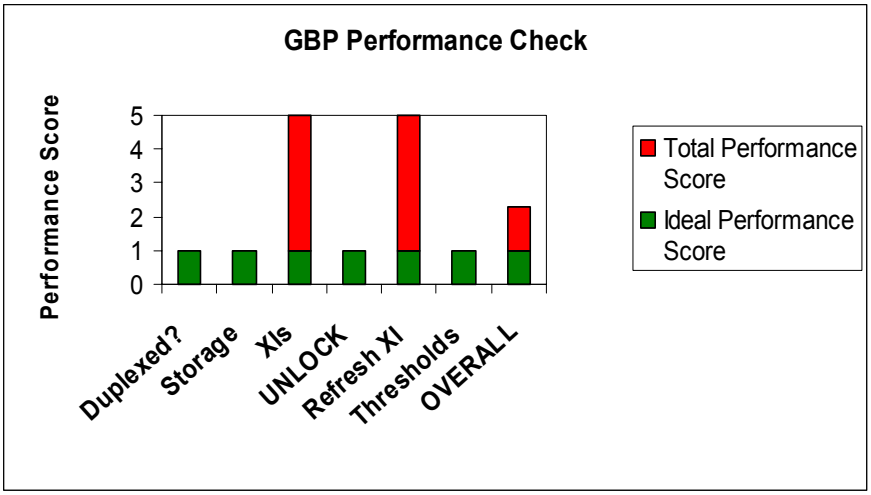
And, of course, READ and WRITE failures should ALWAYS be zero – this GBP is big enough to not have that problem!



Questions 10 and 11 point out another problem with this pool. The hit ratio for pages that have been invalidated is very low. Out of over 700,000 pages invalidated in this time interval, only 45,358 were found in the GBP.

The GBP castout counters in questions 13 through 15 are just what we want to see – more CLASST thresholds hit than POOLT thresholds, than GBP CHKPT thresholds. This is 'trickling' out the I/Os associated with the castout process – not waiting for GBP Checkpoint to do all the I/O work.

EXCEL Spreadsheet Input . . .



As could be expected from looking at the numbers in the DISPLAY GBP output, the XIs due to directory reclaims are the main tuning opportunities in this GBP. Looking back at the numbers, this GBP was creating over 17 million XIs in this hour time period. Assuming that the local buffer pool 12 in the members had a high hit ratio, that means that this data sharing group was doing something in the order of at least 15 million I/Os an hour – FOR NOTHING! WASTED I/Os! OVER 4,000 per second!

A related problem in this pool was the refresh rate for cross-invalidated pages. It's only about 6 %. That means that when a page was cross-invalidated in a local buffer pool, the local DB2 member had to do an I/O 94% of the time to refresh the page.

It looks like this pool is too small for the existing workload – both in the directory portion (directory reclaims too high), and the cache portion (XI refresh hit ratio far too low).

Fixing the Problem – GBP 12

- **August 28 – Increased INITSIZE from 281,344 to 375,808**
  - **Number Directory Entries Increased to 369,561**
  - **XIs Due to Dir Reclaims Stayed About the SAME!**
- **September 12 – Increased INITSIZE from 375,808 to 614,400; Raised the RATIO to 25**
  - **Number Directory Entries Increased to 1,230,621**
  - **XIs Due to Directory Reclaims Went to Zero**

The first obvious choice for the performance problem with GBP 12 was to make it bigger. Fortunately, this customer had available storage in the CF (ICFs in a TREX environment), so making the GBP bigger was not a problem. This first attempt at tuning was enlarging the pool by about 33 %.

The effect on directory reclaims was disappointing – in fact, on some days, they increased! (Workload is variable – so you probably will see this kind of reaction too!).

The second size change was made on September 12 – the pool was made larger, and the ratio was increased. This provided more than 4 times the number of directory entries – and that is the kind of increase that this particular workload needed. The XIs due to directory reclaims went to zero.

Fixing the Problem – GBP 12

- **September 18 – Decreased the RATIO to 20**
  - **Number Directory Entries Decreased to 1,137,863**
  - **XIs Due to Directory Reclaims Stayed at Zero**
- **Fourth Quarter – Large Increase in Workload Expected**
  - **Will Need to Monitor Effect on GBP 12 (and Others)**

After tracking the newly enlarged GBP for a couple of weeks, it was determined that not quite as many directory entries were needed. The RATIO was decreased to 20, without changing the overall pool size. Although this reduced the number of directory entries in the pool to 1,137,863, a decrease of over 7 %, it did not cause XIs due to directory reclaims to start increasing again.

This installation is gearing up for a major roll-out of additional workload in the fourth quarter. As that new workload is introduced, the same type of monitoring and analysis of the group buffer pools needs to be done.

Monitoring group buffer pools needs to be a constant and repetitive part of running a data sharing installation!

So, How Well Tuned are YOUR Group Buffer Pools ????

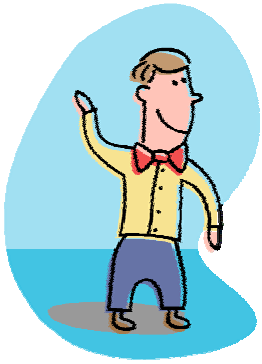
- They NEED to be Monitored!
- Waiting for the Phone to Ring is NOT the Recommended Way to Tune!
- EXCEL Worksheet Referenced in This Presentation Available at:

www.krohnskorner.com

How long has it been since you looked at YOUR group buffer pools?

The EXCEL worksheet referenced in this presentation is available at www.krohnskorner.com, along with a Word document explaining its use.

Thanks for Attending!



For any further questions about this topic, or other DB2 data sharing performance concerns, please contact us at the web site shown!

Bryce Krohn

www.krohnskorner.com

Please contact Bryce at bandpkrohn@aol.com if you have any further questions on this topic, or on the many other considerations for data sharing performance tuning.

This presentation has discussed the details of how DB2 data sharing group buffer pools work, and explained the use of the DB2 command DISPLAY GROUP BUFFER POOL in the analysis and tracking of the performance of the group buffer pools.

The use of an Excel spreadsheet in presenting several of the key numbers and ratios found in the display output was then explained. The spreadsheet, and an accompanying Word document explaining the use of the spreadsheet, can be downloaded from the web site, www.krohnskorner.com.