

IBM Information

>>> On Demand

2007



HDR, BAR, and CLR within IDS 11.10

*Ronald C. Privett,
Advanced Technical Support Engineer, IBM,
rprivett@us.ibm.com*

TIX-1318C & Data Servers - IDS



Act.Right.Now.

IBM INFORMATION ON DEMAND 2007

October 14 - 19, 2007

Mandalay Bay

Las Vegas, Nevada

Introduction

- High-Availability Data Replication (HDR) and Backup And Restore (BAR) are long-established components of IDS, both are robust and reliable.
- This presentation aims to cover both current functionality and then provide details on changes made in 11.10, with a focus on Continuous Log Restore (CLR) for Disaster Recovery.
- Goals...
 - for Novice IDS users:
 - Gain insight into what HDR and BAR can bring to their systems.
 - for Experienced IDS users:
 - Become informed of changes and determine which can be implemented.



IBM Data Servers

- **Reduce cost of deployment and management of data**
 - *Innovation to reduce the cost of infrastructure*
 - *Innovation to manage the lifecycle of data - from modeling and design through change management and sunsetting*
- **Enable rapid use of data throughout the enterprise**
 - *Innovation that accelerates SOA and XML initiatives*
 - *Innovation that leverages Web 2.0 and situational applications*

2



Agenda



1. Current HDR/BAR functionality
2. New HDR features in 11.10
 1. Encryption
 2. Index Page Logging
 3. Global Temp Table Logging
3. New BAR features in 11.10
 1. Parallelized Whole System Backup
 2. Optimal Ordering of Dbspaces
 3. BAR Filters
 4. ONTAP Backup To Directories
 5. Continuous Log Restore (CLR) of Logical Logs
4. Example of Using a CLR Server
5. Review / Questions





Current HDR & BAR functionality

Act.Right.Now.



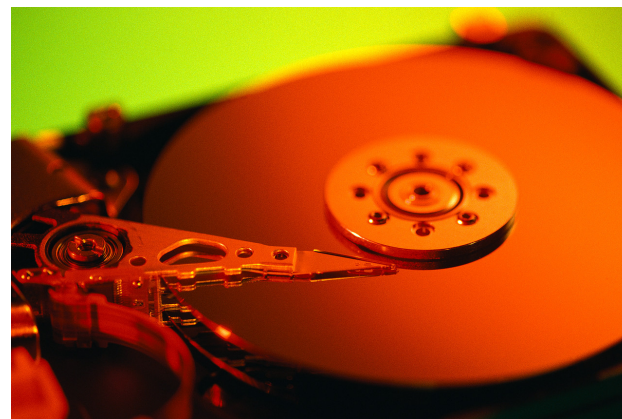
Current HDR functionality

- Homogeneous (RDBMS/Hardware)
- Asynchronous/Synchronous
- Replicates Entire Instance structure
- Only Logged Databases are replicated
- Data in dbspaces and sbspaces supported
- All built-in and extended data types supported
- Primary (Read/Write)/ Secondary (Read only)
- Ontape or OnBar for Recovery
- BlobSpace Data not supported
- Automatic or manual failover
- 1st available in 7.10.UC1



Current BAR functionality

- 2 different tools: OnBar and Ontape
- Archiving Modes: Online, Quiescent, External, Incremental
- Archives all data, or selected dbspaces (onbar)
- Continuous log backup
- Restores all data, selected dbspaces, or tables
- Restore options: Cold, Warm, Mixed, Imported, Point-in-time, Redirected, Restartable
- Verification of archive (archecker)
- Ability to use a Storage Manager (onbar)
- Parallel archive and restore (onbar)
- Separate physical/logical restore (onbar)
- Archive to Standard Output (onbar)
- Used to Initialize HDR systems





New HDR features in 11.10

Act.Right.Now.



New HDR features in 11.10

- HDR Encryption
- Global Temp Table Logging
- Index Page Logging



HDR Encryption

- Encryption already exists for Enterprise Replication (ER), making this an easy feature to add into HDR. This feature was actually added in 10.00.xC5.
- HDR will now use the same onconfig parameters for Encryption that ER uses, with the addition of:
 - ENCRYPT_HDR
 - Default value = 0 (off), set to 1 (on) to enable.
 - Bounce required for setting to be recognized.
 - If clients connect with encryption (i.e. Communication Support Module - CSM), a separate network port is required for HDR without CSM.
 - With added security, comes added cost. Extra CPU cycles are used to encrypt and decrypt HDR data.



Details on ER Encryption

- Added to 9.40, based on OpenSSL, encrypts entire conversation
- Uses Diffie-Hellman Key Exchange Protocol
- Optional MAC (Message Authentication Code) generation/verification
 - MAC key files
- Works with NIF compression
- Dynamic (re)negotiation using 30+ algorithms
 - Key and Algorithm
- Distinct send/receive rules



HDR Encryption (cont.)

- ENCRYPT_CIPHERS - specifies the ciphers and modes to use for encryption
- ENCRYPT_MAC - controls the level of message authentication code (MAC) generation
- ENCRYPT_MACFILE - specifies a list of the full path names of MAC key files
- ENCRYPT_SWITCH - specifies the number of minutes between automatic renegotiations of ciphers and keys



HDR Encryption - How does it work?

- HDR encryption:
 - *will encrypt* Logical logs and Index pages
 - *will not encrypt* Control messages
- Additional memory buffers are used for encryption.
- Checks are done to ensure no HDR traffic occurs over a network port that is uses a CSM.
- The HDR pair negotiates encryption settings prior to starting HDR encryption and it will be switched off if negotiation fails.



Global Temp Table Logging



- Default logging behavior of temp tables matches that of the database.
 - In logged databases, temp tables are also logged.
 - In non-logged database, temp tables are not logged.
- May be desirable to have non-logged temp tables in a logged database.
- New onconfig parameter TEMPTAB_NOLOG allows global control of temp tables without an application change (1=temp tables are not logged, 0=old behavior).
- Bounce required for setting to be recognized.



Index Page Logging (IPL)

- Part of the feature set to enable MACH-11
- Default behavior of 'Index Shipping'
 - Lock the table partition when index is created.
 - A logical log record marks the index creation.
 - New index is shipped page by page to the secondary.
 - After acknowledgement, the primary unlocks the partition.
- With IPL, the contents of the index are placed in the log stream. This ensures consistency between the log and index at any time.
 - Use new parameter LOG_INDEX_BUILDS to enable.



Index Page Logging Overview

- Normal HDR environments (Primary and Secondary) can take advantage of this new feature. If you do not enable Index Page Logging, you will get the old behavior (e.g. You can choose to use either the new “index page logging” or the old “index shipping”).
- If you want to use Remote Standalone Servers (RSS), then enabling LOG_INDEX_BUILDS is required.
- A bounce of the engine is required for this setting to be recognized.





New BAR features in 11.10

Act.Right.Now.



New BAR features in 11.10

- Parallelized Whole System Backup
- Optimal Ordering of Dbspaces
- BAR Filters
- ONTAPE Backup To Directories
- Continuous Log Restore (CLR) of Logical Logs



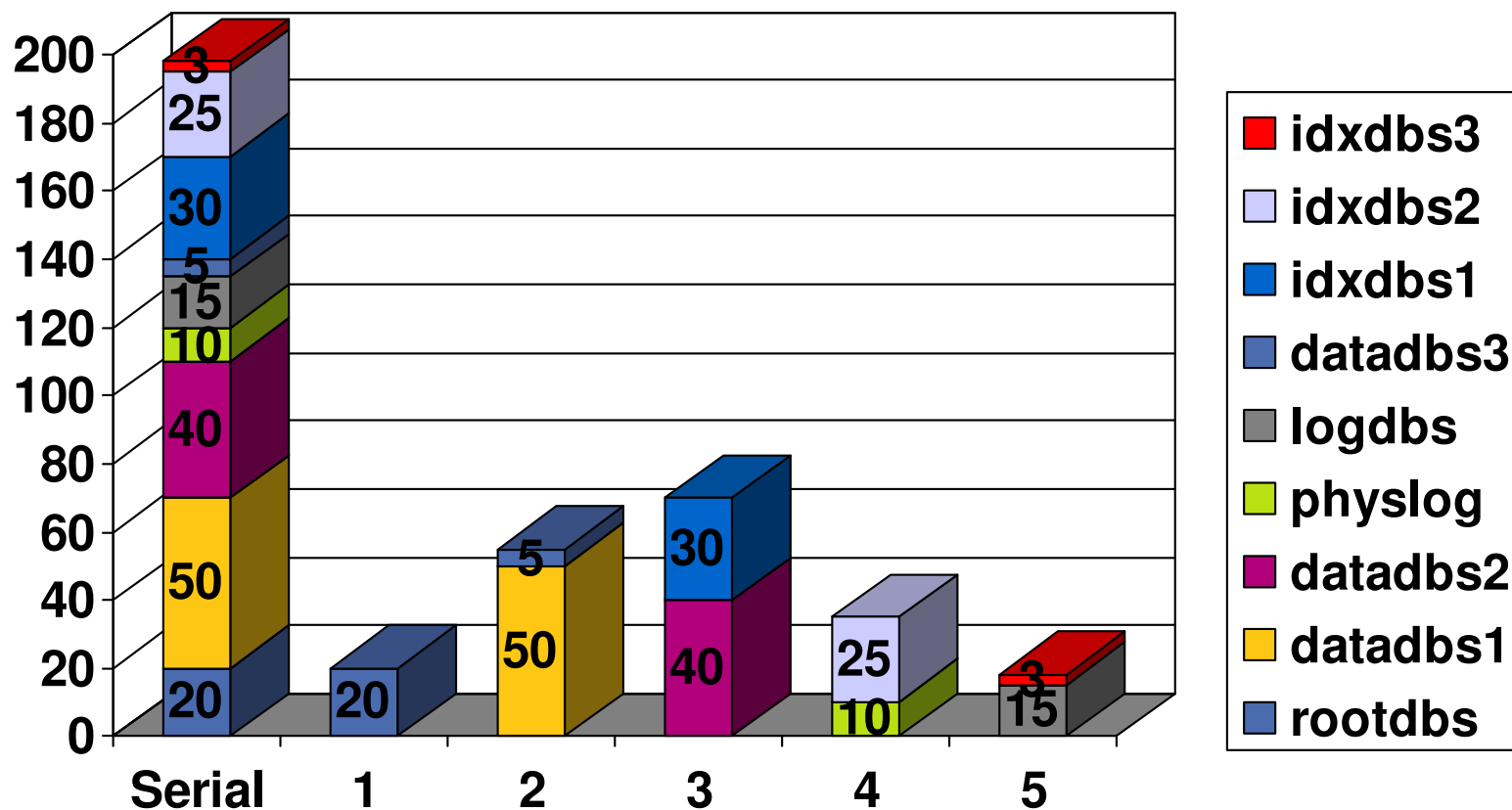
Parallelized Whole System Backup

- Why do Serial Backups at all?
 - Technically there is no reason why a whole system backup and restore needs to be serial -- the serial behavior was inherited from ontape.
- Parallelization of Whole System Backups
 - rootdbs is still backed up first and by itself.
 - The rest of the dbspaces are backed up in parallel, based on BAR_MAX_BACKUP.
 - New default value for BAR_MAX_BACKUP = 4.



Parallelized Whole System Backup

- Example of old Serial method, and new Parallel method with `BAR_MAX_BACKUP=4`



Optimal Ordering of Dbspaces

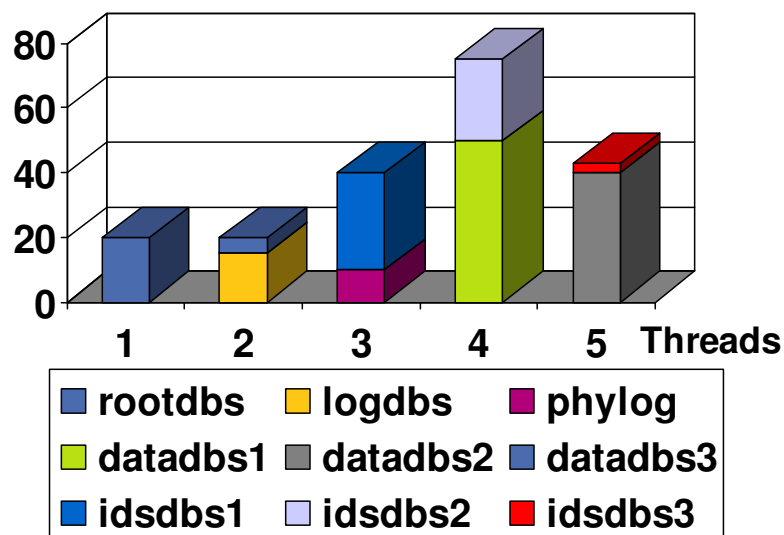
- Optimize performance by sorting dbspaces by size:
 - Sort dbspaces by decreasing size, so the largest dbspaces will be backed up first, and smaller spaces last.
 - The size of a dspace is determined by the used-pages count at start time of backup.
 - Dspace with most pages used will be backed up first.
 - This ensures better parallelism, no matter how BAR_MAX_BACKUP is set or how many pages are to be backed up in different dbspaces.
 - New column 'ins_backup_order' column added to sysutils:bar_instance table to track backup order. Also, a new column was added to the ixbar file to track this information.
 - Restore uses the same order as the backup to ease SM's access to stored objects.
 - This feature is applied to all archives (ontape and onbar).



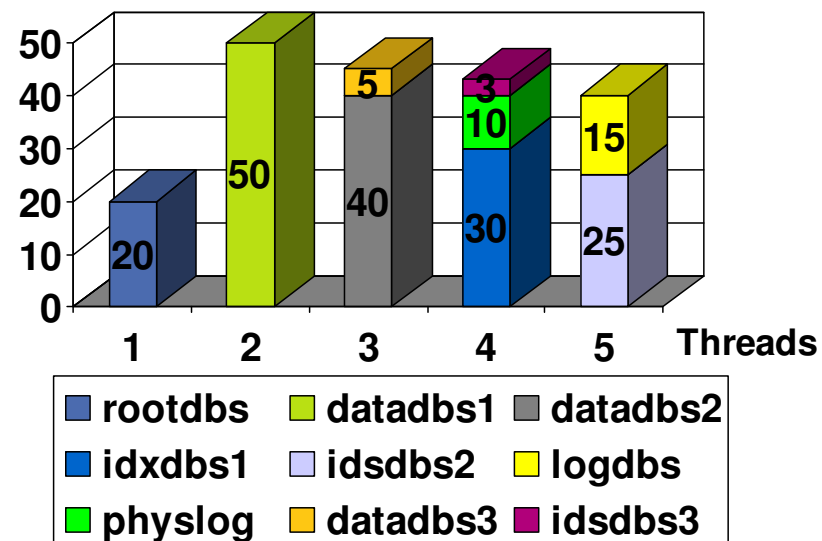
Optimal Ordering of Dbspaces

- The old method used the creation time of the dbspace to determine backup order
- The new method uses the number of used pages within the dbspace

Old Database Ordering Method

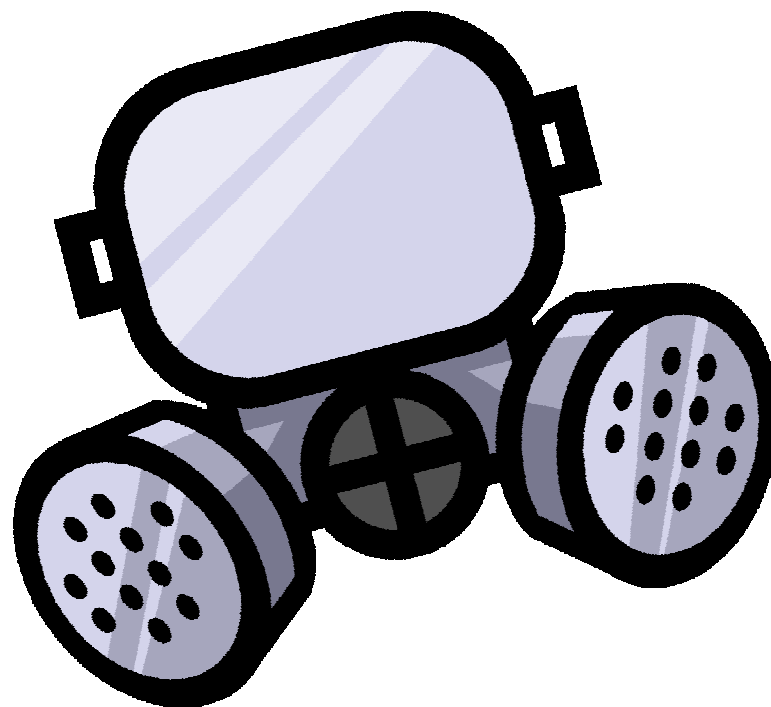


New Database Ordering Method

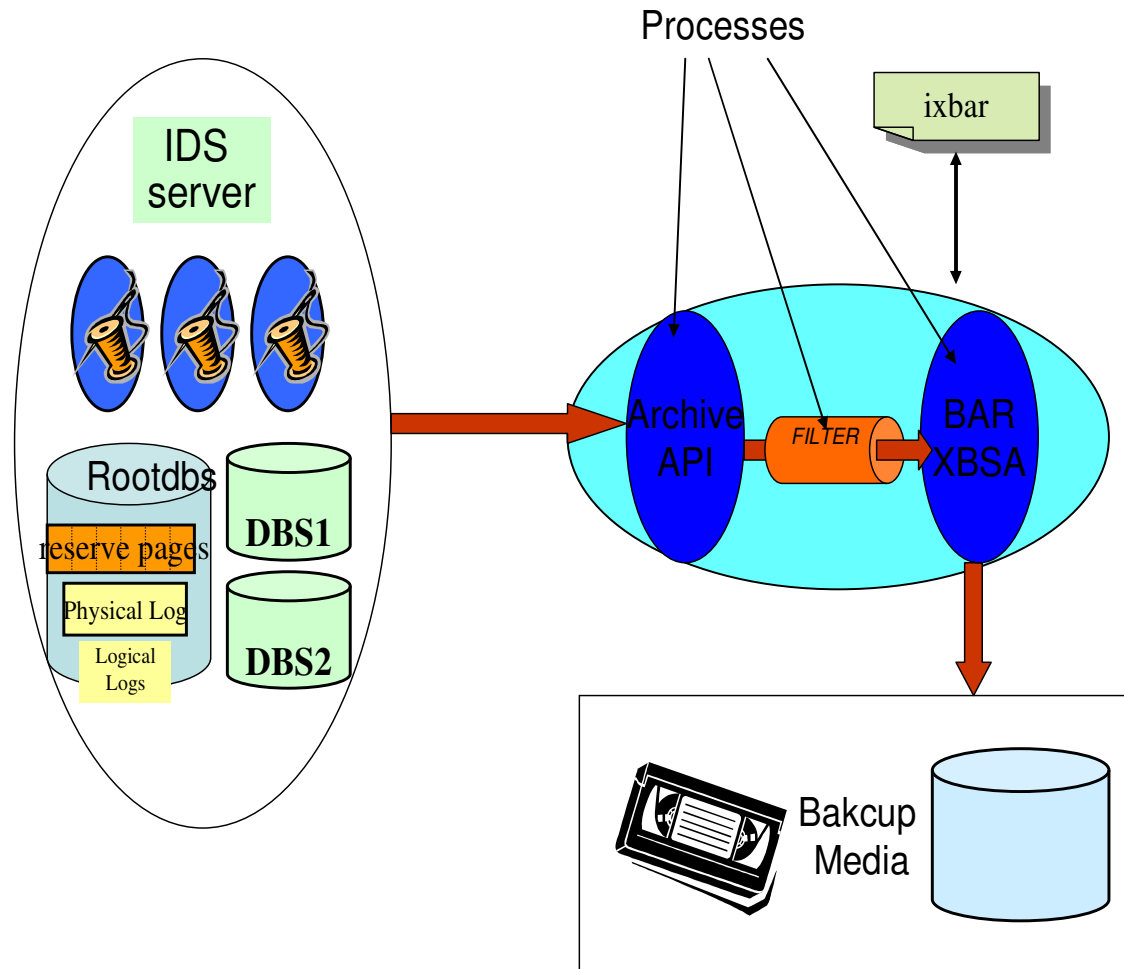


BAR Filters

- Allows customers and 3rd parties the ability to develop modules to manipulate data during backups and restores via a pluggable module with ontape and onbar.
- Possible uses:
 - Encryption
 - Compression
 - Other uses?



BAR Filters (cont.)



BAR Filters (cont.)

- How to activate
 - The functionality applies to ontape and onbar.
 - Two new onconfig parameters.
 - BACKUP_FILTER (data out - blank by default and so IDS uses old style).
 - RESTORE_FILTER (data in -blank by default and so IDS uses old style).
 - These parameters will point to the filters (programs) that perform the encryption, compression or the customers required functionality.
 - The filters can reside in \$INFORMIXDIR/bin or be absolute path.
 - Filters can be passed options and/or filenames by enclosing the onconfig parameters in single quotes.
 - Servername, servernumber, etc will be available for the filter.



BAR Filters - Caveats

- Will not work with external backups.
- Performance may drop, so try to use parallelism where possible.
- There are discussions on whether there will be some form of certification provided for the third party products.
- Customers developing their own filters should :
 - Full tests must be performed to ensure the filters work!
 - Suggest they run oncheck during and after testing?



ONTAPE Backup To Directory

- Finally!!! Support for backup to directories with ontape.
 - Simple to setup and administer.
 - Allows for automated archives.
 - Automatically renames old archives when new ones are created.



Backup to Directory - Example

- 1st Backup taken with TAPEDEV set to c:\IDS

```
C:\IDS>ontape -s -L 0
```

```
10 percent done.
```

```
20 percent done.
```

```
30 percent done.
```

```
40 percent done.
```

```
100 percent done.
```

```
File created: c:\IDS\IBM-65G8X7HJPJY_0_L0
```

Please label this tape as number 1 in the arc tape sequence.

This tape contains the following logical logs:

```
38
```

Program over.



Backup to Directory - Example (cont.)

- 2nd Backup taken with TAPEDEV set to c:\IDS

```
C:\IDS>ontape -s -L 0
```

```
10 percent done.
```

```
20 percent done.
```

```
30 percent done.
```

```
40 percent done.
```

```
100 percent done.
```

```
File created: c:\IDS\IBM-65G8X7HJPJY_0_L0
```

Please label this tape as number 1 in the arc tape sequence.

This tape contains the following logical logs:

38

Program over.



Backup to Directory - File Naming

- Naming of the backups

Directory of C:\IDS

```
04/06/2007 03:22 PM <DIR>      .
04/06/2007 03:22 PM <DIR>      ..
04/06/2007 02:49 PM      16,941,056 IBM-65G8X7HJPJY_0_20070406_144941_L0
04/06/2007 03:22 PM      16,941,056 IBM-65G8X7HJPJY_0_L0
                2 File(s)  33,882,112 bytes
                2 Dir(s)  50,500,202,496 bytes free
```

- First L0 backup was renamed when 2nd backup taken.

The backup file <hostname>_<servernum>_L0 was renamed to
<hostname>_<servernum>_<YYYY-MM-DD_HHMMSS>_L0



Backup to Directory - Log Example

- Log backup taken with LTAPEDEV set to c:\IDS

```
C:\IDS>ontape -a -d
```

```
Your evaluation license will expire on 2007-10-01 00:00:00
```

```
Performing automatic backup of logical logs.
```

```
File created: c:\IDS\IBM-65G8X7HJPJY_0_Log0000000041
```

```
Do you want to back up the current logical log? (y/n) n
```

```
Program over.
```

- Current log **NOT** backed up



Continuous Log Restore (CLR)

- An enhancement to both ontape and OnBar
 - When logs are full and backed-up, the log files can be copied to other supporting instances in CLR mode
- Logs are applied to target instances...
 - Manually - as part of daily or weekly activities
 - Automatically - as set by a TASK, or other processes
 - `ontape -l -C`
 - `onbar -r -l -C`
- Note: A general assumption here is that the source server will archive logs to disk files, rather than tapes.

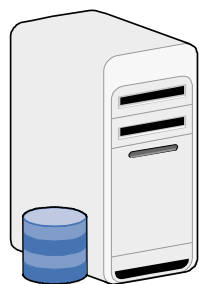


Example of Continuous Log Restore

Customer has immediate failover in local office but also wants an offsite copy. Telco/network outages are common, and throughput is too erratic for RSS.

OR

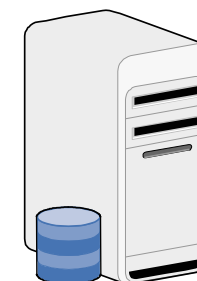
Customer needs to be able to recover quickly but doesn't want / need immediate failover



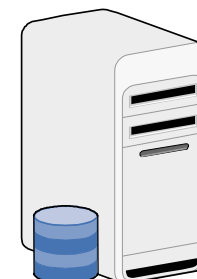
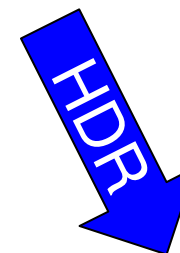
Continuous Log Restore Server

Continuous Log Restore (CLR)

Customer uses O/S utilities to transfer logs and backups as needed. CLR instance can be up or not. Backups and logs can be applied to instance as business rules dictate.



HDR Primary



HDR Secondary



Advantages of using a CLR server

- Another server in case of Disaster
 - Can you ever have enough DR servers?
- Faster Recovery Time
 - If you need to restore your entire instance, having a CLR server means you could be up and running faster.
- Verification of Logical Restore
 - Archecker only verifies the physical restore, a CLR server can verify the logical restore.
- Manual or Automatic application of Logical Logs
 - Ever drop a production table, and it is sent to the HDR secondary, or another RSS node before you can break the connection? With a CLR server, you decide when to apply the logical log files.



Steps to setup a CLR node (ontape)

1. Create a level-0 archive of source server:

```
ontape -s -L 0
```

2. Copy archive to target server, and restore:

```
ontape -p
```

3. As logical logs fill and are backed up on the source server, copy the files to the target server and rename the files as needed (or use environment parameter `IFX_ONTAPE_FILE_PREFIX` to set the filename):

```
cp {source file} {target directory}
```

```
mv {hostname_servernum_L0} {newhost_servernum_L0}
```

4. The CLR node stays in Recovery Mode, any copied logs are applied as needed, manually - or automatically (via a script):

```
ontape -l -C
```

5. When needed, stop roll forward and put server in quiescent mode:

```
ontape -l -X
```



Steps to setup a CLR node (onbar)

1. Create a level-0 archive of source server:

```
onbar -b -L 0
```

2. Copy all archive objects to target server, and perform imported physical restore:

```
onbar -p
```

3. As logical logs fill and are backed up on the source server, copy the required files to the target server to allow for imported restore of logical logs.

4. The CLR node stays in Recovery Mode, any copied logs are applied as needed, manually or automatically (via a script):

```
onbar -r -l -C
```

5. When needed, stop roll forward and put server in quiescent mode:

```
onbar -r -l -X
```





Example of Using a CLR Server

Act.Right.Now.



Example of using a CLR server

- In this example there are 3 servers
 - HDR Primary
 - HDR Secondary
 - CLR Server
- The example will show...
 - The process of setting up the CLR server (w/ ontape)
 - Log transfer and application
 - Stopping / Restarting the CLR server
 - Completing the CLR process



Setting up the CLR server

- Create a level-0 archive of source server:

```
ontape -s -L 0
```

- Copy archive to target server, and restore:

```
ontape -p
```

- In this case, both the CLR and HDR Primary servers are on the same host, using relative paths. Before running the restore, you need to change the SERVERNUM portion of the archive name to match the CLR servernum.
 - HDR Primary SERVERNUM = 101
 - CLR servers SERVERNUM = 103
- ```
cp {hostname}_101_L0 {hostname}_103_L0
```
- Or use the environment parameter IFX\_ONTAPE\_FILE\_PREFIX to set the filename.



# Setting up the CLR server (cont)

- Notice the mode of the engine after the physical restore, and the status of the dbspaces:

```
> onstat -d
```

```
IBM Informix Dynamic Server Version 11.10.UC1 -- Fast Recovery -- Up 00:00:37 --
 28820 Kbytes
```

## Dbspaces

| address  | number | flags   | fchunk | nchunks | pgsize | flags | owner    | name    |
|----------|--------|---------|--------|---------|--------|-------|----------|---------|
| 44cd37f0 | 1      | 0x40401 | 1      | 1       | 2048   | NP B  | informix | rootdbs |
| 44cd3e60 | 2      | 0x40401 | 2      | 1       | 2048   | NP B  | informix | dbs1    |
| 44d87430 | 3      | 0x40401 | 3      | 1       | 2048   | NP B  | informix | dbs2    |
| 44d87590 | 4      | 0x40401 | 4      | 1       | 2048   | NP B  | informix | dbs3    |
| 44d876f0 | 5      | 0x48401 | 5      | 1       | 2048   | NPSB  | informix | sbs1    |
| 44d87850 | 6      | 0x48401 | 6      | 1       | 2048   | NPSB  | informix | sbs2    |
| 44d879b0 | 7      | 0x40401 | 7      | 1       | 2048   | NP B  | informix | tempdbs |

7 active, 2047 maximum

Physically Restored

## Chunks

| address  | chunk/dbs | offset | size | free  | bpages | flags | pathname  |
|----------|-----------|--------|------|-------|--------|-------|-----------|
| 44cd3950 | 1         | 1      | 0    | 50000 | 28649  | PI-B  | ./rootdbs |
| 44d87b10 | 2         | 2      | 0    | 20000 | 19947  | PI-B  | ./dbs1    |
| 44d87ce0 | 3         | 3      | 0    | 20000 | 19947  | PI-B  | ./dbs2    |
| 455aa018 | 4         | 4      | 0    | 20000 | 19947  | PI-B  | ./dbs3    |
| 455aa1e8 | 5         | 5      | 0    | 10000 | -1     | PISB  | ./sbs1    |
| 455aa3b8 | 6         | 6      | 0    | 10000 | -1     | PISB  | ./sbs2    |
| 455aa588 | 7         | 7      | 0    | 5000  | 4897   | PI-B  | ./tempdbs |

7 active, 32766 maximum

Inconsistent



## Log transfer

- As logical logs fill and are backed up on the source server, copy the files to the target server and rename the files as needed.
  - On the HDR Primary, rename the files:

```
> cp eleven_101_Log0000000011 eleven_103_Log0000000011
> cp eleven_101_Log0000000012 eleven_103_Log0000000012
> cp eleven_101_Log0000000013 eleven_103_Log0000000013
```
  - Or use the environment parameter `IFX_ONTAPE_FILE_PREFIX` to set the filename.
- In this example, the CLR node is in Recovery Mode, and copied logs are applied manually.

```
> ontape -l -C
```

```
Roll forward should start with log number 11
```

```
Rollforward log file /home/informix/arc/eleven_103_Log0000000011 ...
```

```
Rollforward log file /home/informix/arc/eleven_103_Log0000000012 ...
```

```
Rollforward log file /home/informix/arc/eleven_103_Log0000000013 ...
```



## Stopping / Restarting the CLR server

- You can stop the engine while in Recovery mode with the following commands

```
onmode -ky
```

- You can restart the engine in Recovery Mode with the following:

```
oninit -r
```

- NOTE: The CLR server will be broken if you change into Quiescent, or Online Mode. You would need to restore the CLR server from scratch to enable CLR again.



# Completing the CLR process

- To complete the process of CLR, and create a usable engine, run the following:

```
> ontape -l -X
Program over.
```

- Check the status of the engine and review the online log for details about recovery.

```
> onstat -m
```

```
IBM Informix Dynamic Server Version 11.10.UC1 -- Quiescent -- Up 00:59:56 -- 45204 Kbytes
```

```
Message Log File: /home/informix/clr/online.log.informix
```

```
12:49:56 Suspending Logical Restore
```

```
13:35:51 Resuming Logical Restore
```

```
13:35:57 Logical Recovery has reached the transaction cleanup phase.
```

```
13:35:58 Checkpoint Completed: duration was 1 seconds.
```

```
13:35:58 Fri Sep 14 - loguniq 17, logpos 0x18, timestamp: 0x2cb83 Interval: 24
```

```
13:35:58 Maximum server connections 0
```

```
13:35:58 Checkpoint Statistics - Avg. Txn Block Time 0.000, # Txns blocked 0, Plog used 297,
Llog used 1
```

```
13:35:58 Logical Recovery Complete.
```

```
879 Committed, 0 Rolled Back, 0 Open, 0 Bad Locks
```

```
13:35:58 Logical Recovery Complete.
```

```
13:35:58 Quiescent Mode
```

Changed from Fast Recovery

Stats for entire CLR process



## CLR Issues to be aware of...

- CLR is designed to assist Disaster Recovery, not as a High Availability solution. However, you can 'promote' a CLR server into an HDR secondary server (details on the next slide).
- There is no Point-In-Time restore with CLR, it's the whole log or nothing (similar to a Point-In-Log restore).
- To aid the process of log movement between source and target servers - consider the following:
  - Using the environment variable `IFX_ONTAPE_FILE_PREFIX` to set the filename of ontape 'backup to directory' files.
  - Using a shared disk to store archive files
  - Script the process of removing old 'applied' log files to back them up to tape or other archive media.



# CLR Promotion into an HDR Secondary

- If needed, you could promote a CLR server into an HDR secondary server. Here are the steps:

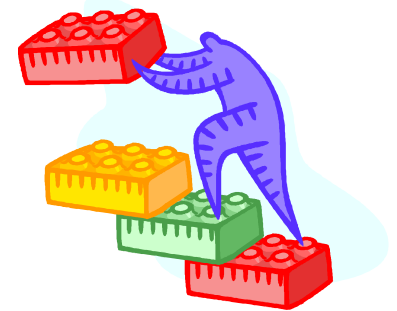
- Break HDR with the current secondary server

- On Primary, use the CLR servername:

```
onmode -d primary {secname}
```

- On the CLR server:

```
onmode -d secondary {priname}
```



- Primary server sees the secondary needs recovery, and will start sending logs:

```
20:32:56 DR: Secondary server needs failure recovery
```

- Secondary sever recovers the logs and becomes available

```
20:33:02 DR: HDR secondary server operational
```





# Review

*Act.Right.Now.*



# Review

- Current HDR/BAR functionality
- New HDR features in 11.10
  - Encryption
  - Index Page Logging
  - Global Temp Table Logging
- New BAR features in 11.10
  - Parallelized Whole System Backup
  - Optimal Ordering of Dbspaces
  - BAR Filters
  - ONTAPE Backup To Directories
  - Continuous Log Restore (CLR) of Logical Logs
- Example of using a CLR server





Questions?

*Act. Right. Now.*





Thank You!

*Act.Right.Now.*

