



Welcome to the Virtual IMS user group newsletter. The Virtual IMS user group at [www.fundi.com/virtualims](http://www.fundi.com/virtualims) is an independently-operated vendor-neutral site run by and for the IMS user community.

## Virtual IMS user group presentation

The latest webinar from the Virtual IMS user group was entitled, "Memoirs of a HALDBA", and was presented by Neil Price, a Senior DBA and Systems Programmer with TNT Express ICS.

Neil first encountered IBM mainframes as a student in 1971 and started working with them full-time in 1975. He has been involved with IMS almost continuously since 1977. He has been with TNT Express ICS at their worldwide data centre in Warwickshire since 1994. As Senior DBA and Systems Programmer he now supports IMS DBCTL and DB2 on the mainframe. Neil has been Chairman of the GSE UK IMS Working Group since 2002, having successfully avoided it as a committee member for several years previously!

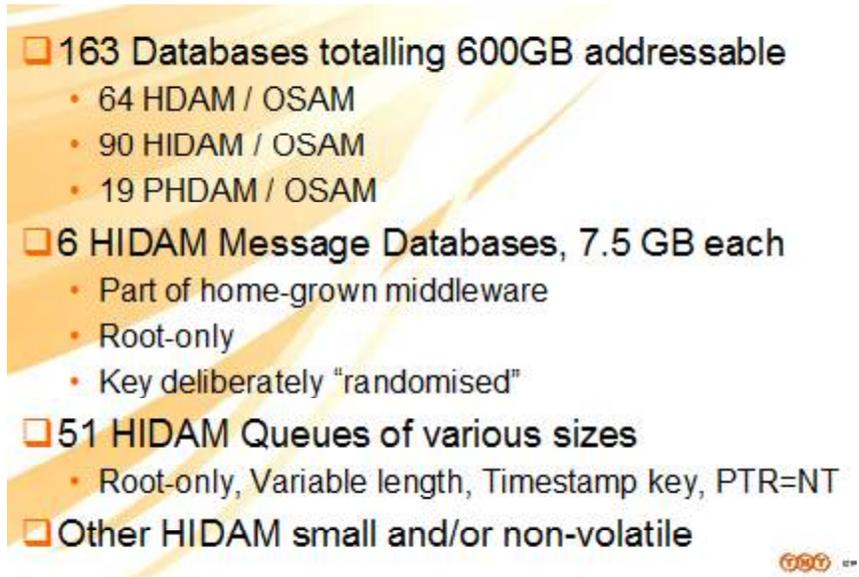


Figure 1: TNT's IMS databases

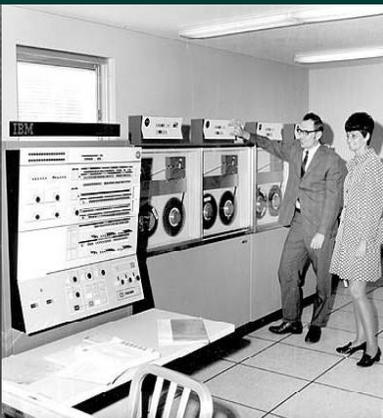
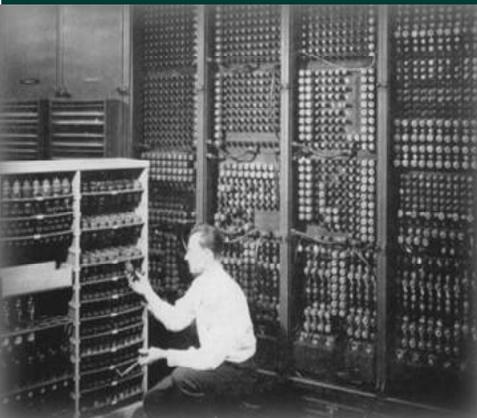
Neil started by giving the user group a clear idea of what hardware and software TNT Express use. This is relevant because it affects the decisions that Neil and his organization had to make.

They have two machines – one hosting the main production LPAR; and the other running the network, testing, and one or two other things. This has a total disk

capacity of about 18TB, of which just over 4TB are mirrored to their Disaster

### Contents:

Virtual IMS user group presentation	1
Meeting dates	6
Recent IMS articles	6
IMS news	6
About the Virtual IMS user group	6



Recovery site, about 100 miles away in London. TNT's IMS databases are illustrated in Figure 1.

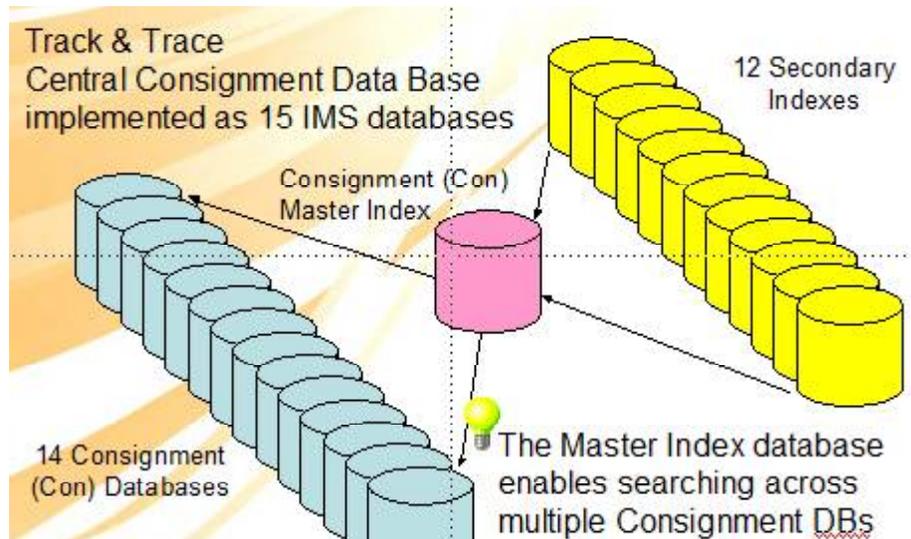
Neil informed us that they always used OSAM because they believe it's more efficient. More recently, of course, it gave them the ability to have datasets up to 8GB in size.

Very few of their databases ever get disorganized enough to need a reorg unless they're running out of space. The use of HDAM for all of the most volatile databases is a major factor in this. A few are reorganised weekly, 3 of them in order to remove logically-deleted data with a User Exit, and a few quarterly.

For the indexes that don't get rebuilt by reorgs, they use a non-IMS ISV utility to back up and restore the KSDS, some every week but mostly just once a quarter. All their indexes are defined as non-recoverable.

The 6 HIDAM Message databases aren't anywhere near filling up, but in any case they have the option to use up to 10. They use a key, which avoids a single insert hotspot whilst still keeping all the data for a single message together.

51 of the 90 HIDAM databases are root-only databases used as queues and have a constantly ascending timestamp-based key. These were changed to have no pointers very early on



**Figure 2: Application partitioning - CCDB**

because of frequent deadlocks involving the high-values root. Since then, virtually all of their HIDAM databases have been changed to have NOTWIN on the root.

And the other 33 HIDAM databases are small, or get very few updates, or both.

Track and Trace enquiries are their most heavily-used transactions, and the Central Consignment Data Base supports these. Consignment data is split across 14 identical databases (see Figure 2), originally because of the 4GB size limit, and these have no indexes of their own. Instead a subset of the data from all 14 databases is duplicated into corresponding segments of the Con Master Index database, and indexes for the various search criteria implemented in the application are built on that.

The Master Index root has the same key as Consignment and holds a "PCB number" from zero to 13 to show which database holds the Consignment record, or a value of '99' to indicate a logically deleted Con. This is done to eliminate physical deletions, so the corresponding index entries are not "sparsed out" by an Index Maintenance Routine but have to be recognized in the application code. The '99' entries are removed each weekend by a User Exit routine in an offline ISV reorg.

One Consignment database takes all the new records for a week. The Archiving suite takes the "oldest" database, before it's reinitialized and reused, and copies any records not ready for archive into a newer database. The PCB number for every record in the database gets changed to either '99' or the new

database number, so the oldest database becomes logically empty – although nothing actually gets deleted.

There's a parallel set of 15 databases holding billing information for Consignments. They are processed in a similar way, but they only have 6 Indexes and don't share the key of Consignment.

In terms of HALDB, Neil told us that since they started investigating and testing HALDB in 2004, they've converted 18 HDAM databases, which were getting close to the 8GB OSAM dataset size limit, including the whole of the CCDB and one with an index that was threatening to hit 4GB between weekly reorgs.

Just 2 of their HALDBs are partitioned by key range. Most of the HALDBs got up to 16 partitions on model-3 logical volumes over time, but went down to 6 partitions when they moved them to model-9s.

Apart from Consignment, each HALDB has at least one index. 5 have 28 indexes between them, and 13 of the indexes are also partitioned.

Neil was pleased to have avoided certain challenges at his site. For example, the DBRC question – they decided to have DBRC forced registration in every system from the very beginning, and, because there's virtually no DL/I batch, there's just one

copy of each database and it's permanently online.

It didn't take much thinking to decide that their 14 cloned databases would stay separate when they became HALDBs. Anything else would have involved major application code changes.

The databases using a partitioning exit all use the Consignment key, so they've effectively had to write only one simple exit routine. They still maintain separate modules, though, just in case they need to vary the logic at any time.

To simplify everything they treat the HALDBs just the same as before except in very exceptional circumstances and haven't changed their names. In particular, not taking individual partitions offline means that the applications don't need to handle partitions being unavailable.

In terms of challenges they faced, Neil said that the first database they converted was one using the Consignment key, and it was fairly obvious that they wouldn't get a predictable spread of data using key ranges. In the end they adapted the example in the IBM Redbook "The Complete IMS HALDB Guide" (SG24-6945) to select a partition based on a portion of the key, in their case the trailing characters of the 9-

digit character numeric Consignment number.

With equally-spaced "limits" this has resulted in a very even spread of data across the partitions. They started with 2 digits, but once they went from 10 to 12 and then 16 partitions it seemed sensible to start using 3 digits for a slightly more even spread. Of course the overall physical sequence of records isn't the same as in HDAM.

For databases with key-range partitioning, they used the HALDB Migration Aid utility DFSMAID0 to choose the limits.

When it comes to monitoring challenges, Neil told the group that for non-HALDBs they get a list by e-mail of any thresholds that have been exceeded, for example the number of database records or the number of HDAM roots not stored in their "home" block. This is sent by a simple SAS program that processes a daily report of all metrics and threshold exceptions from their Pointer Checker statistics repository. The statistics are gathered as part of the daily online Image Copy backups. They find the e-mail preferable to having to navigate ISPF panels or a GUI to find out what exceptions have occurred, if any.

HALDB statistics, on the other hand, are written to a completely different

repository, which can be viewed through a GUI but didn't come with facilities to generate reports or e-mails. So they segregated the HALDBs into their own backup jobs, which direct the Pointer Checker report to a file. An additional SAS step then processes the file, writes the results to a SAS database and a summary report, and generates an e-mail if the percentage of usable free space in any partition falls below a hard-coded threshold.

Key-range partitions can be so different from each other that they need to be treated like independent databases. For example if the volume of data in one partition is growing much faster than in the others, it might need a lot more free space so that you don't have to resize or split it too soon, and you might want to set the free space warning threshold lower, so that you have time to react.

Another example is where the average database record length in one partition is much lower than in the others. If you want to be alerted when the record lengths increase, the threshold for this partition would need to be correspondingly lower.

When it comes to tuning – what Neil called “Performance Tweaks” – they have various processes where values are obtained in physical sequential order from the root

keys of one database and used to access one or more related databases with the same root key.

If the databases involved are all HDAM and use the DFSHDC40 randomizing routine, this works well because the randomizer tends to keep keys in more or less the same sequence regardless of the total number of Root Anchor Points, as long as this is much greater than the number of roots. This means that all the databases are accessed in physical sequential order, which more or less guarantees that each database block is read only once and so minimizes I/O, as well as generating a sequential access pattern that maximizes read hits in disk cache and more recently has enabled them to take advantage of OSAM Sequential Buffering.

But once you go to PHDAM, the keys are only kept in this sequence within each partition. Optimal performance is only restored when all the databases involved are HALDBs with identical partitioning arrangements.

Neil Price identified something that he said confuses a lot of people, including some in IBM support, not least because the manuals are unclear.

Normally when a KSDS data CI fills up, it's split roughly in half. That's not good for ascending keys such as the timestamps used by most of their HIDAM databases, because the old or “lower” CI will stay half-empty forever. In the same way a CA split will leave the old CA with half its CIs empty and the rest usually half-empty, in other words only about a quarter full.

If the insert rates are high, the KSDS can end up with a very high proportion of unusable free space and might approach the 4GB limit.

However if Sequential Mode is used for the inserts, CI and CA splits are done at the point of the insert – which for ascending keys means that it starts a new, empty CI or CA and leaves the old one as it is. This is good for performance – CA splits have been known to take seconds to complete – as well as greatly slowing down the dataset growth.

It's possible to specify sequential-mode inserts in the IMS buffer pool specification member by coding `INSERT=SEQ` on the `OPTIONS` statement, but that would apply to every KSDS. Instead they include a `DBD` statement for each index they want treated this way and specify `FREESPACE=YES`, which is the dataset-level equivalent.

Sequential-mode inserts also honour the free space specifications for the KSDS. This means that if free space is specified and all inserts are done this way, the free space will never get used and is a complete waste of space. All their inserts are done through the online system, so FREESPACE=YES implies that there should be no Free Space specified for the cluster!

FREESPACE=YES would work almost as well with constantly descending keys. It might also be beneficial for some indexes whose keys are generally, rather than strictly, ascending or descending, but until they've got IMS 12 and can change buffer pool parameters dynamically it's going to be difficult to determine.

Currently they specify FREESPACE=YES for 6 of their PSINDEXes, of which only one has a strictly ascending key and the rest start with a date. There used to be more, but after analysing the VSAM statistics, Neil reverted some to normal inserts and non-zero freespace. This reduced the rate of CA splits, ie growth, and also the overall I/O.

Neil reminded us that PSINDEX records can be relatively large, which makes the total index much larger than its non-HALDB version. It also means that fewer records

are retrieved in one I/O, which is bad for sequential processing. For Neil's site, most of their indexes are accessed with a partial key, which in some cases will match hundreds of entries.

Neil attempted to minimize the effects by making the VSAM buffer pools bigger, although he knew he wouldn't be able to get the I/O rates back down to pre-HALDB levels. He reasoned that the buffers ought to hold the same number of VSAM data records as before, so he aimed to increase the number of buffers in line with the increase in the record length.

When he calculated the new subpool sizes he discovered that some came out greater than the limit of 32,767 buffers, and in any case there wasn't enough virtual storage available for such large increases, so he had to scale them all back anyway.

But the I/O rates for most of the OSAM subpools are higher than for any of the VSAM ones, so this is maybe not the area with the most potential for I/O reduction.

Also, he discovered early on that FREESPACE=YES had no effect on PSINDEX datasets, so they halted their first production HALDB conversion until IBM fixed it.

Neil said his site currently specify FREESPACE=YES

for 6 PSINDEXes, of which only one has a strictly ascending key and the rest start with a date. There used to be more, but he changed some back to normal skip-sequential insert and found that the overall I/O went down. The other 5 might be better that way, but until he can change buffer pool parameters dynamically (in IMS 12) there's no easy way to try it out.

To sum up, Neil informed us that HALDBs can change your life if you're a DBA!

Once you've converted your databases, that's when the fun really begins, he said, adding that he hadn't mentioned some of the software bugs and other quirks he'd had to deal with, especially with disabled partitions.

Indexes will probably have higher I/O rates than before, and in general things can get a lot more complicated straight away. Imagine what it might be like when the databases have dozens or even hundreds of partitions!

But in some aspects the separate partitions can be easier to manage than a single large database, and there can be advantages in being able to tune them separately. Keeping the partitions in shape according to whatever criteria and priorities you have can be interesting. On the other hand

some of the possible tweaks might have very little effect. HALDBs in fact perform just the same as “normal” databases most of the time. And there are so many potential challenges that your experiences could be completely different to ours!

A copy of Neil’s presentation is available for download from the Virtual IMS user group Web site at [www.fundi.com/virtualims/presentations/HALDBAFeb12.pdf](http://www.fundi.com/virtualims/presentations/HALDBAFeb12.pdf).

You can see and hear the whole user group meeting by downloading the WMV file from [www.fundi.com/virtualims/presentations/2012-02-07meeting.wmv](http://www.fundi.com/virtualims/presentations/2012-02-07meeting.wmv).

## Meeting dates

- 3 April 2012 – Fundi Software’s James Martin, will be talking about “Using IMS Performance Solution Pack for z/OS to analyze IMS performance problems”.

## Recent IMS articles

*CICS Transaction Server for z/OS V4.2 and Threadsafe Database Control for IMS* by Russ Evans in *zJournal* (December 2011/January 2012). You can find the article at [www.maintenancezone.com/it-management/cutting-costs-on-z-os-with-websphere-mq-and-ims](http://www.maintenancezone.com/it-management/cutting-costs-on-z-os-with-websphere-mq-and-ims).

*Cutting Costs on z/OS With WebSphere MQ and IMS* by Ben Johnson, Pete Siddall, and Jack Yuan in *zJournal* (March/April 2012). You can find the article at [www.maintenancezone.com/it-management/cics-transaction-server-for-z-os-v4.2-and-threadsafe-database-control-for-ims](http://www.maintenancezone.com/it-management/cics-transaction-server-for-z-os-v4.2-and-threadsafe-database-control-for-ims).

*Piloting IMS Through the 21st Century* by John McKenny in *zJournal* (March/April 2012). You can find the article at [www.maintenancezone.com/it-management/piloting-ims-through-the-21st-century](http://www.maintenancezone.com/it-management/piloting-ims-through-the-21st-century).

*Sidebar: How to Extend the Size of Your IMS Database* by John McKenny in *zJournal* (March/April 2012). You can find the article at [www.maintenancezone.com/it-management/sidebar-for-the-article-piloting-ims-through-the-21st-century](http://www.maintenancezone.com/it-management/sidebar-for-the-article-piloting-ims-through-the-21st-century).

## IMS news

Compuware has announced Abend-AID Version 12.2. This fault resolution tool includes new features such as a COBOL PERFORM trace, consolidated displays of IMS and DB2 data to speed access to crucial information, and easier access to Abend-AID reports. More information can be found at [www.compuware.com/about/release/643555/compuware-latest-innovations-enable-enterprises-to-successfully-navigate-the-new-normal-of-maintenance](http://www.compuware.com/about/release/643555/compuware-latest-innovations-enable-enterprises-to-successfully-navigate-the-new-normal-of-maintenance).

Select Business Solutions has announced NOMAD 7.54 for z/OS, its business intelligence reporting tool and database. The IMSSCHEM utility, which generates NOMAD schemas from IMS PSBs, has been enhanced. There is a new method of generating item, master, and segment names for IMS fields with similar names or with names containing invalid characters for NOMAD. More information can be found at [www.selectbs.com/press-releases/press-release-120203](http://www.selectbs.com/press-releases/press-release-120203).

## About the Virtual IMS user group

The Virtual IMS user group was established as a way for individuals using IBM’s IMS hierarchical database and transaction processing systems to exchange information, learn new techniques, and advance their skills with the product.

The Web site at [www.fundi.com/virtualims](http://www.fundi.com/virtualims) provides a central point for coordinating periodic meetings (which contain technically-oriented topics presented in a webinar format), and provides articles, discussions, links, and other resources of interest to IBM IMS practitioners. Anyone with an interest in IMS is welcome to join the Virtual IMS user group and share in the knowledge exchange.

To share ideas, and for further information, contact [trevor@itech-ed.com](mailto:trevor@itech-ed.com).

The Virtual IMS user group is free to its members.