A-9 93

CFM LOLA

BENEFITS NOTES

1. BTS

2. Databases

3. IMS - XPERT

4. PLI/DLI Techniques

5. MFS

1. BTS

1.1 Components

i) The main component is UAC.BTS.CLIST. This is maintained by Nigel Stratton Fiddel. The most important member on this library is BTS. DO NOT AMEND THIS MEMBER. Member BTS controls the invocation of BTS.

Other members in UAC.BTS.CLIST of significance are:

BTSALLOC	-	allocates datasets like PSB, DBD libraries etc.
BTSAL@F BTSAL@LA BTSAL@U2		allocates FA, LAY, U2 sets respectively
BTSIN		defines general LOLA transacting such as name matching, address matching.

ii) UHB/CCB/CTB testing transactions are defined in UAL.UHB.DATA(BTSTEST). Production trans are defined in UAL.UHB.DATA(BTSPROD).

iii) Program - on LOLA2.MPPLIB as Uxxx or on LOLA1.MPPLIB as xxx

Beware of member Uxxx being on LOLA1.MPPLIB as this will be picked up in preference to Uxxx off LOLA2.MPPLIB.

iv) PSB - on IMS1.LV1.PSBLIB or IMS2.LV1.PSBLIB

v) Vetting masks - will be copied from batch vetting masks to temporary database if requested.

vi) Paging database - as vetting masks

Generally little maintenance is required. UAC.BTS.CLIST(BTS) is maintained by Nigel. The database allocation members BTSAL@xx only required amendment if a database is added, or dataset names change but this is extremely rare. The transaction definition members in UAL.UHB.DATA require amendment if SPA size changes or a new transaction is written, again this is quite rare.

Boro transactions can be defined (where TEST BORO on database not used) by setting TC=bxxx where b = boro code eg T, see TU11 in BTSTEST as an example.

2. Benefits Utopia Databases

2.1 Copying from one set to another

2.1.1 UHB/CCB see Dataset UAL.CCB.CNTL

The following decks exist, these are for copying from the batch set to the 2 BTS testing sets:

DBCOPYF	copies UB to FA set
DBCOPYL	copies UB to LAY set

The following decks exist to copy from the batch set (UB) to the IMS set (UA) or vice versa, they should only be used in extremis when one set is unusable eg. corrupt.

DBCOPYA	copies UB to UA set
DBCOPYB	copies UA to UB set

Should both the UA and UB set become unusable it would be possible to create a deck to copy from the FA set and LAY set back to the UB set. This assumes the FA or LAY set are themselves usable.

2.1.2 CTB see dataset UAL.CTB.CNTL

The following members exist, these are for copying from the batch set to the 3 BTS testing sets:

DBCOPYF	copies UB set to FA set
DBCOPYL	copies UB set to LAY set
DBCOPY2	copies UB set to U2 set

The following members exist to copy to batch set (UB) and to the IMS set (UA), they should only be used in extremis when one set is unusable eg. corrupt.

DBCOPYA	copies UB set to UA set
DBCOPYB	copies UA set to UB set
DBCOPY2	copies LAY set to UB set

An additional member exists:

DBCOPY22 copies UA set to U2 set.

2.2 Database Dumps and Restores

From time to time, say every 3 months, its worth taking a dump of both the batch (UB) and IMS (UA) sets.

Decks exist for this as follows:

UHB/C 'B	UAL.C B.CNTL(DBDUMPUA)	dump of UA set
	UAL.C 'B.CNTL(DBDUMPUB)	dump of UB set
CTB	UAL.CTB.CNTL(DBDUMPUA)	dump of UA set
	UAL.CTB.CNTL(DBDUMPUB)	dump of UB set

Restoration decks exist as below but use only needed should the IMS(UA) or batch(UB) sets become unusable eg. corrupt.

These decks are as follows:

UHB/CCB	UAL.CCB.CNTL(DBRESTUA)	restore of UA set
	UAL.CCB.CNTL(DBRESTUB)	restore of UB set
CTB	UAL.CTB.CNTL(DBRESTUA)	restore of UA set
	UAL.CTB.CNTL(DBRESTUB)	restore of UB set

Note that dumps and restores use cartridges. A cycle of 5 cartridges is defined in the relative members.

2.3 Problems

Most problems can be resolved by using either the copy or restore jobs listed above.

However space problems are sometimes encountered. These can be solved in the following manner:

a) database full - increase space allocation in restore AND copy decks, then run restore/copy job. (in sequence 1, COPY 2. RESTORE)

b) pack full - consult OSG who will either clear down pack or recommend an alternative pack. (The latter will require amendment to both restore AND copy decks). Then run restore/copy job.

3. IMS-XPERT

3.1 Batch

3.1.1 Macro Amendments

These are controlled by the setting in "PL/I LAYOUT DATASET 1" on the TSO IMS-XPERT browse/edit screen.

If LOLA2.LBR.MACRO is specified then testing versions of macros will be picked up. If LOLA1.LBR.MACRO is specified then production versions of macros will be picked up.

3.1.2 New Segments

Introduction of a new segment requires amendment to the cross reference dataset UAL.IXP.XREF.

This is effected through option 7-XREF in IMS-XPERT.

3.2 IMS

Amendments to macro layouts and the introduction of new segments are implemented by running a batch job UAL.CCB.CNTL(IXDXREF) overnight. A production database run sheet must be submitted - see folder MEMO/EXPERT next to Phil for example.

3.3 Common Data

UBH(UU043A) uses the Multi Record Type Value (MULTI-RTV) means of defining common data segment layouts. In option 7 (XRREF) this involves specifying one line per common data record type. This method requires amendment to the XREF (Option 7) in IMS-XPERT.

CCB and CTB use a single macro - ERBCOMST/LCOMMONX - to define multiple layouts for the same segment. This method requires amendment to the general macro ERBCOMST/LCOMMONX.

3.4 UHB

HB Public and HB Private are on the same database. They have function 1 and 2 respectively in the root. To pick up the current macro layouts use DBD UU043A/UB043A respectively.

4. PL1/DLI Techniques

This section could run to 50 pages at least. I shall do no more than highlight 2 major areas of concern.

4.1 Database Calls

The benefits strategy is to use GNP calls. This has proved successful in the past and will continue to do so as long as database integrity remains intact.

If curious results are ever returned in production then beware that GNP in benefits terms relies on parentage set at the highest level ie. root level.

4.2 LOOPS

Use HBOUND and LBOUND. Hard coding numbers leads to program amendments and invites errors.

5. MFS (TP Screens)

CFM LOLA provides both mono and colour formats for their TP transactions. Benefits provide for this as below:

Formats are developed in mono. Testing is performed in mono in both BTS and IMS. Prior to implementation job TMFSCONV on UAL.CTB.CNTL is run on each format. This will generate source coding within the format for the colour elements of the screen. The format copy-up uses the colour source coding to generate colour formats on the production format library.

Note that colour/mono formats are picked up in IMS based on the DEFINITION of the terminal. If a colour terminal is DEFINED as mono it will pick up the mono formats. If a mono terminal is defined as colour it will not work in IMS. Jan's terminal V3F0018 is defined as being colour in IMSDEV, all others are mono. The way to tell how a terminal is defined is to find the terminal ID in IMS1.L13D.DEF(NETDEFD) - the NAME must end in C and TYPE = 3270-A2 must be present for the terminal to pick up colour formats.

Note that a terminal has two definitions in IMS, one in production (IMS1.L13P.DEF(NETDEFP)) one in development (IMS1.L13D.DEF(NETDEFD)). A terminal can be defined as mono in production and colour in development.

Note also that colour formats can be displayed in BTS provided you have a colour terminal and you reply Y to the question 'DO YOU WANT TO DISPLAY COLOUR SCREENS'. This is regardless of your terminals definition.