Mixed Page Group Implementation

We have a large database PED.FPEN003D-AREA which spreads across 12 files with two files sitting on each 3390-9. It has a number of cross-pointer areas (FPEN004D, 5D, 6D, 7X, 8X, 9X, 13X, 18X)
They are all part of segment PED with page group 5

Over the years we have expanded FPEN003D-AREA, changing its page size several times (keeping the ORIGINAL PAGE SIZE 13680 CHARACTERS). We have also increased its page range a number of times. Now we have reached the capacity limits in the number of database pages and records that applications can access.

The solution is to use Mixed Page Group support.

Mixed Page Group Support was introduced in IDMS release 14.1.
It allows a single run unit to access data residing in areas defined to different page groups.
The main restriction is that there can be no chain sets that cross page group boundaries.
However, a new feature, Mixed Page Group Index (MPGI) was introduced in IDMS release 18.5 which allows the owner and member records within an index set relationship to reside in different page groups.

To take advantage of the MGPI feature, which supports a database with a user-owned index set or system-owned index in which the owner and member record types are in separate areas associated with different segments that are to be assigned to different page groups, the following may be implemented in parts or in total:

- Convert a chain set to an index set.
- Move the member or owner record to a separate area.
- Move the index structure of a system-owned index into an area different from that of the indexed records.
- Assign an area to a different segment.
- Assign a segment to a different page group.
- Alter a DBNAME to allow mixed page groups.
- Specify the new auto-READY extension for an area.

See more details here:
https://support.ca.com/cadocs/7/CA%20IDMS%2018%205-ENU/Bookshelf_Files/HTML/IDMS_v18.5_ReleaseNotes_ENU/1975369.html

IDMS, by default, does not support the ability to access data in areas with different page groups or maximum records per page from a single run unit. IDMS has a limit on the number of pages that can be defined within a page group due to the structure of the DBKEY that is used to identify a record. Page groups were introduced to increase the number of available pages. Using the default DBKEY format, a single page group can have 16,777,214 pages, and you can define up to 32,767 page groups.

To access a database which exceeds the size limits of a single page group or which uses different record maximums from a single run unit, we need to specify the MIXED PAGE GROUP BINDS ALLOWED option on the DBNAME statement that defines our PED database.

Since page group information is not kept in database record, using mixed page group does not require any change to existing subschema or user application. That is the case with us.
However, if your application program retrieves a record by dbkey, the following consideration is observed:

- Specify on the DML command the name of the record that it is trying to retrieve
- Specify on the DML command the page group and maximum records per page of the record that it is trying to retrieve

For dictionary and catalog the following rules are observed:

- The DDLDML and DDLDCLOD areas must be in the same page group and have the same maximum records per page. The DDLDCMSG area (if included in the DBNAME) must also have the same page group and record maximum.
- The DDLCAT, DDLCATX, and DDLCATLOD areas must be in the same page group and have the same maximum records per page.

Changes we made in SEGMENT, DBTABLE and DMCL

```
     ALTER
     SEGMENT PED               (PED has PAGE GROUP 5 and MAXIMUM RECORDS PER PAGE 255)
         ;
     DROP
     PHYSICAL AREA PED.FPEN003D-AREA
         :                (repeat for all related cross-pointer areas in page group 5)

     DROP
     FILE PED.FPEN103D               (repeat for related files in page group 5)
         ;


     CREATE
     SEGMENT PED10
         FOR NONSQL
         PAGE GROUP 10
         MAXIMUM RECORDS PER PAGE 255
       ;
     CREATE
     PHYSICAL AREA PED10.FPEN003D-AREA (repeat for all related areas in page group 10)
         PRIMARY SPACE 2603988 PAGES  FROM PAGE 13396010
         MAXIMUM SPACE 2603988 PAGES
         PAGE SIZE 13680 CHARACTERS
         WITHIN FILE FPEN103D
             FROM 1 FOR 216999 BLOCKS
*+           MAP TO PAGES 13396010 THRU 13613008
         WITHIN FILE FPEN203D
             FROM 1 FOR 216999 BLOCKS
*+           MAP TO PAGES 13613009 THRU 13830007
         WITHIN FILE FPEN303D
             FROM 1 FOR 216999 BLOCKS
*+           MAP TO PAGES 13830008 THRU 14047006
         WITHIN FILE FPEN403D
     CREATE
     FILE PED10.FPEN103D               (repeat for all related files in page group 10)
         ASSIGN TO PPEN103D
         DSNAME 'IDMSPED.DB.FPEN103D'
         DISP SHR
         NONVSAM
*+       FILE BLOCK SIZE 13680 CHARACTERS
*+       BLOCKS 1 THRU 216999 MAP TO AREA FPEN003D-AREA PAGES 13396010 THRU
*+             13613008
*+       OVERRIDES FOR DMCL IDMSDMCL
*+           BUFFER G13680-BUFFER
*+           SHARED CACHE DEFAULT
*+           MEMORY CACHE NO
         ;


     ALTER
     DBTABLE IDMSDBTB
         ;
     ALTER
     DBNAME IDMSDBTB.PEDDICT
*+       ALTERD 2009-09-24-14.09.52.379978    BY GZWMS0
*+       LAST UPDATED 2009-10-06-11.19.44.601620   BY GZCDZK
         MATCH ON SUBSCHEMA OPTIONAL
         MIXED PAGE GROUP BINDS NOT ALLOWED
         FOR GENERAL USE
         INCLUDE SEGMENT PEDCAT
         INCLUDE SEGMENT PEDDICT
         INCLUDE SEGMENT PEDMSG
         ;
     ALTER
     DBNAME IDMSDBTB.PED
         MATCH ON SUBSCHEMA OPTIONAL
         MIXED PAGE GROUP BINDS ALLOWED
         FOR GENERAL USE
         INCLUDE SEGMENT PED                     <- Page Group 5
         INCLUDE SEGMENT PED10                   <- Page Group 10
         INCLUDE SEGMENT PEDDICT                 <- Page Group 5
         INCLUDE SEGMENT PEDMSG                  <- Page Group 5
         INCLUDE SUBSCHEMA SS0050 USES DBTABLE MAPPING
         INCLUDE SUBSCHEMA SS?????? MAPS TO SS??????
         INCLUDE SUBSCHEMA ???????? USES DBTABLE MAPPING
         ;
```

```
,  D SEG PEDDICT                                                            ,
---------- Area -----------   Lock    Lo-Page    Hi-Page #Ret  #Upd #Tret #Ntfy
PEDDICT.DDLDML                 Upd     300001     320000    0     0     0     0
 Stamp: 2010-11-22-10.37.12.058465  Pg grp: 5      NoShare  NoICVI  NoPerm
PEDDICT.DDLDCLOD               Upd     350001     356000    0     0     0     0
 Stamp: 2009-09-24-14.04.32.466101  Pg grp: 5      NoShare  NoICVI  NoPerm
V5  ENTER NEXT TASK CODE:      CA IDMS release 18.0 tape GJI00B node SYST0005
```

```
,  D SEG PEDCAT
---------- Area -----------   Lock    Lo-Page    Hi-Page #Ret  #Upd #Tret #Ntfy
PEDCAT.DDLCAT                  Upd     200001     220000    0     0     0     0
 Stamp: 2009-09-24-14.02.10.257828  Pg grp: 5      NoShare  NoICVI  NoPerm
PEDCAT.DDLCATLOD               Upd     280001     280500    0     0     0     0
 Stamp: 2009-09-24-14.02.10.315603  Pg grp: 5      NoShare  NoICVI  NoPerm
PEDCAT.DDLCATX                 Upd     290001     290500    0     0     0     0
 Stamp: 2009-09-24-14.02.10.329328  Pg grp: 5      NoShare  NoICVI  NoPerm
```

```
V5  ENTER NEXT TASK CODE:      CA IDMS release 18.0 tape GJI00B node SYST0005
```

```
,  D SEG PEDMSG
---------- Area -----------   Lock    Lo-Page    Hi-Page #Ret  #Upd #Tret #Ntfy
PEDMSG.DDLDCMSG                Ret      10001      13000    0     0     0     0
 Stamp: 2009-10-06-11.13.24.949226  Pg grp: 5      NoShare  NoICVI  NoPerm
V5  ENTER NEXT TASK CODE:      CA IDMS release 18.0 tape GJI00B node SYST0005
```

```
     ALTER
     DMCL IDMSDMCL
         DBTABLE IDMSDBTB
         DATA SHARING NO
         MEMORY CACHE LOCATION ANYWHERE STORAGE LIMIT OPSYS
         INCLUDE SEGMENT PED10
             ON STARTUP SET STATUS TO UPDATE
             ON WARMSTART MAINTAIN CURRENT STATUS
             DATA SHARING NO
             DEFAULT SHARED CACHE NULL
         INCLUDE FILE PED10.FPEN103D   (repeat for all related files in page group 10)
             BUFFER G13680-BUFFER
             SHARED CACHE DEFAULT
             MEMORY CACHE NO
```

For testing, we use OLQ MENU with existing subschema, SS0004 (no change to subschema) to retrieve records from page group 5 database areas, and records from page group 10 database areas

Once the database area, FPEN003D and its related areas, FPEN004D, 5D, 6D, etc. have been moved to page group 10

Unload/reload of these area will reference to segment, PED10, e.g.

```
//SYSIPT     DD *
    UNLOAD SEGMENT PED10
            USING SS0004
    AREA FPEN003D-AREA, FPEN004D-AREA, FPEN005D-AREA, FPEN006D-AREA
            RELOAD INTO PED10
            USING SS0004
            DMCL IDMSDMCN
```