Moving MXG off the Mainframe

Why move MXG ?

- If MXG is the only SAS usage on the mainframe the SAS license cost on a per user basis can be very high
  - One customer with a 2817-407 and 409 have annual costs of 160,000
  - The cost of upgrading the 407 to a 409 is 50,000 for the upgrade and an increase of 16,000 in the annual fee
- The cost of an ASCII license (Windows or LINUX) is much lower
- If you have a small CEC and can limit SAS to that CEC you may be able to mitigate that cost somewhat
- The BUILDPDB process often runs more quickly on a healthy PC than on zOS
  - zOS typically is managing many more tasks than the PC
  - SAS multi-threads better on ASCII

Why Move MXG?

- Reporting in ASCII SAS is more ‘robust’ than on zOS
  - Creating graphs on zOS can be done but can be complex
  - Can also be extremely memory intense
  - Is simplest with access to USS file structures
- The interfaces to the PC tools your customer uses (EXCEL, PDF, etc) are simpler
- Multiple ‘jobs’ can be running on ASCII against the same SAS data libraries
  - LOCKING on zOS is at the LIBNAME level – on ASCII it is at the SAS dataset level

An example - zOS

- Processing 5901MB of SMF data
- zOS 2828-S04 SAS 9.3
  - Elapsed time - 1:13:16
  - CPU Time - 0:41:24
  - Big Data step 0:54:02 elapsed time
  - Remaining steps 0:19:14 elapsed time

An Example Windows 8.1 Remote SMF

- Windows 8.1 Intel I7 4710 @ 2.5 Ghz SAS 9.4 24GB memory
- Processing 5901MB of SMF data
- FTP access method used from 2000 miles away
- Elapsed Time - 2:14:24
  - CPU Time - 0:26:08
  - Big Data Step 2:11:40
  - Remaining Steps 0:02:24

An Example – SMF on Local Disk

- Windows 8.1 Intel I7 4710 @ 2.5 Ghz SAS 9.4 24GB memory
- Processing 5901MB of SMF data
- Data downloaded via FTP – 2 hours
- Elapsed Time - 0:07:51
  - CPU Time - 0:06:51
  - Big Data Step 0:05:23
  - Remaining Steps 0:02:28

At Another Site Locally Connected

- 31GB of data to a Win 6.2 Server
- SMF data was read at 3768KB/second
- Elapsed Time - 3:15:51
  - Big Data Step - 2:24:30
  - Remaining Steps - 0:51:21

How Quickly It Will Run

- Depends on the speed
  - Of the connection to the data
  - Of the processor
  - Of the disk drives

An Example

- PDB size on zOS
  - Uncompressed bytes 5849M
  - Compressed bytes 2720M
- PDB size on Windows
  - Uncompressed bytes 6301M
  - Compressed bytes 2949M – 8% larger than on zOS - YMMV

What pieces of SAS do I need?

- You do NOT need a server license or enterprise guide though both will work – they are just not necessary but you do want a dedicated machine since SAS can easily 'takeover' most PCs
- BASE – absolutely required
- GRAPH – not required assuming you keep current and are at 9.3 or higher of SAS since ODS provides graphics capability
- ACCESS – may be convenient to at least have the interfaces to PC files (EXCEL WORD etc.)
- STAT – perhaps if you want to use some of those procedures
- Others – not required by MXG

How Much Space Do I Need?

- Use the space on zOS and add 10% as a start
- To find the size run this code:
  ```
  %VMXGSIZE;
  PROC SORT;BY LIBNAME MEMNAME;
  PROC PRINT;
  BY LIBNAME;
  SUMBY LIBNAME;
  VAR MEMNAME BYTES COMPBYTE;
  SUM BYTES COMPBYTE;
  ```
  - The SUM of COMPBYTE for the PDB is the size to use.
- As usual – ‘It depends’
  - How many days of data do you want to keep?
  - How much detail data?
  - How much to keep weekly/monthly?
  - Do you really need all of those variables
    - Do you have any 3420 3330 3350 3380 devices? Didn’t think so.

Dropping Variables – An Example

- For every dataset created by MXG there is an _Kxxxxxx exit that can be used to add or drop variables

```
%LET MACKEEP=%QUOTE(
   MACRO _KTY30U4
    DROP=EXCP3330 IOTM3330…
   %
);
%INCLUDE SOURCLIB(TYPE30);
```

What about managing the data?

- On zOS smart users are using GDGs and letting the operating system manage retention of data
- GDGs are a ‘foreign’ concept on ASCII systems
- MXG provides macros that allow you to manage directories on disk drives using dates in the directory names
  - You can specify the number of days/weeks/months of data to keep
  - You can specify the date format to use in building the directories
  - Detail CICS and DB2ACCT can be directed to different drives if desired
  - More on this later

Getting Started

- Identify the platform to be used and install SAS
- Download and install MXG
  - A single ZIP file contains the entirety of MXG
  - Does not have to be on the C drive – can be on any drive including a network drive
  - You will need (at least) 3 directories
    - SOURCLIB – MXG source
    - USERID – your USER mods
    - FORMATS – FORMAT library
- Download your USERID.SOURCLIB
  - Step 1 – JCLZERO
  - Step 2 – Download results of JCLZERO
  - Step 3 - IEBUPDTE

My Personal Preference

- I add a CHANGES directory to the SOURCLIB concatenation
- Changes that are needed between production releases go into this directory and are then cleared out when the new production release is installed
- Simplifies maintenance

JCLZERO

```
//STEP1 EXEC SAS
//USERID DD DSN=USERID.SOURCLIB,DISP=SHR
//OUTFILE DD DSN=USERID.SOURCLIB,DISP=(,CATLG),
//  SPACE=(CYL,(10,10),RLSE)
//SYSIN DD *
  PROC SOURCE INDD=USERID OUTDD=OUTFILE;
```

IEBUPDTE

Inside a SAS session:

```
%LET IN_DIR=D\;  ← Directory where you downloaded output of JCLZERO
%LET IEBFILNM=IEBUPDTE.MXG;  <- NAME of downloaded file
%LET OUT_DIR=c:\userid  <- directory for your USERID.SOURCLIB
%INCLUDE ‘C:\MXG\SOURCLIB\IEBUPDTE.SAS’;
```

This runs a SAS program that emulates IEBUPDTE and unloads your downloaded USERID.SOURCLIB into a directory with member names of xxxxxxxx.sas where xxxxxxxxxx is the member name in your USERID.SOURCLIB.

Editing Members

- Any text editor will do
- My preference is SPFPC
- But Notepad/Wordpad or any other will do – just be careful of line numbers

Build your AUTOEXEC.SAS

Choose the AUTOEXEC member in the SOURCLIB appropriate to your OS –

AUTOEXEC – Windows
AUTOEXEU – LINUX

These lines need to be changed to point at your MXG directories

```
/* REQUIRED FILENAME/LIBNAME FOR MXG SOURCE, FORMAT, LOCATIONS */
FILENAME SOURCLIB ('C:\MXG\USERID' 'C:\MXG\SOURCLIB');
LIBNAME  LIBRARY   'C:\MXG\FORMATS';
```

Build Your AUTOEXEC

Delete these lines – they will be handled later

```
 /* 1. TO READ DOWNLOADED Z/OS SMF DATA FROM ASCII DISK FILE AND TO   */
/*    ALWAYS HAVE AN INFILE SMF POINTING TO AN SMF FILE FOR TESTING: */

FILENAME SMF      'C:\MXG\SMFDATA\SMFSMALL.U'
                 RECFM=S370VBS LRECL=32760 BLKSIZE=32760;

/* 2. TO READ CONCATENATED DOWNLOADED SMF FILES, USE THIS SYNTAX

FILENAME SMF  (
            'C:\MXG\SMFDATA\SMFSMALL.U'
            'C:\MXG\SMFDATA\SMFSMAL2.U'
            )    RECFM=S370VBS LRECL=32760 BLKSIZE=32760;    */

/* 3. TO USE FTP ACCESS METHOD TO READ SMF DATA WITHOUT DOWNLOAD:

FILENAME SMF FTP ("'SYS1.SMF'" "'SYS2.SMF'" ... )
                USER='XXXXXX'  HOST='YYYYYYY' DEBUG
                S370VS RCMD='SITE RDW' LRECL=32760
                PASS='XXXXXXXX';
     NOTE: IF YOUR SMF DATA IS ON TAPE, YOU SHOULD USE:
                RCMD='SITE RDW READTAPEFORMAT=S'           */

LIBNAME  PDB      'C:\MXG\PDB';
LIBNAME  CICSTRAN 'C:\MXG\CICSTRAN';
LIBNAME  SPIN     'C:\MXG\SPIN';
LIBNAME  DB2ACCT  'C:\MXG\DB2ACCT ';
```

Build Your AUTOEXEC

- Save in your USERID SOURCLIB
- Create an MXG shortcut
  - Copy the SAS shortcut on your desktop
  - Add to the properties:
    - -autoexec ‘your userid sourclib\autoexec.sas'
  - Rename to MXG
  - Change the ICON if you wish

Now See If It Works

Click on your MXG shortcut – SAS should start and you should see something like this in the SASLOG – if you don’t there is a problem with your AUTOEXEC:

```
        WELCOME TO MXG SOFTWARE, FROM MERRILL CONSULTANTS, DALLAS,
TEXAS
        TECH SUPPORT:   214 351 1966    SUPPORT@MXG.COM    WWW.MXG.COM
        MXG 33.12 DATED NOV 25, 2015 HAS BEEN INITIALIZED.
        NOTE: Fileref= SOURCLIB
           Physical Name= D:\CHANGES
           Physical Name= D:\V3312
```

Setting up daily/weekly/monthly Jobs

- You could use your existing BUILDPDB job but…
- We recommend using a combination of UTILBLDP and BLDSMPDB
  - UTILBLDP allows you to add data to the BUILDPDB process
  - UTILBLDP allows you to REMOVE data from the BUILDPDB Process
  - BLDSMPDB can run daily/weekly/monthly all in one job
  - BLDSMPDB allows you to ‘trim’ datasets from weekly/montly jobs
  - BLDSMPDB invokes VMXGALOC to automatically create and manage PDB libraries

SAS Macros

- MXG uses many SAS macros to drive this process. Very few use positional parameters but instead use symbolics
- The rules for coding macro calls are simple
- %macroname(parameter1=x,parameter2=y);
- %macroname causes SAS to look in the AUTOCALLs file and if it finds a member of that name that defines a macro of the same name compiles it
- Parameters are separated by commas except for the final one specified which concludes the call with a);

SAS Macros

- There can also be macros with positional parameters separated by commas but they are rarely used in MXG and then only for macros that are ‘hidden’ in the source that you are not expected to use.

A Note about large datasets

- Many sites have routed CICSTRAN/DB2ACCT and other large datasets to tape on zOS.
- That is not an option on ASCII and is not really needed
- All data can be directed to PDB or kept separately if desired
- Locking on ASCII is at the SAS dataset level rather than the LIBNAME level as it is on zOS so multiple datasets can be pointed at the same LIBNAME without an issue

Using UTILBLDP

- UTILBLDP is a SAS MACRO driven by parameters that replaces the need to create the old EXPDB*** members in your USERID.SOURCLIB
- You can add data
- You can remove data
- You can set datasets to 0 OBSERVATIONS
- You can specify things to be included after BUILDPDB
- You can suppress things automatically included

UTILBLDP Important Parameters

- OUTFILE=INSTREAM – where to send the generated SAS code – though INSTREAM is not the default it is what is recommended
- BUILDPDB=YES to invoke BUILDPDB – A value of NO suppresses BUILDPDB and can be used to read multiple SMF record types in a single pass of the SMF dataset
- USERADD= records to be added to the PDB for IBM records with a value LE 128 all that is needed is the record number. For others the record type followed by the record number as HSM/251 to include TYPEHSM with a record ID of 251

UTILBLDP Important Parameters

- SUPPRESS – a list of records to be suppressed that will not be placed in the PDB – for examples SUPPRESS=74 78 would suppress the type 74 and 78 RMF data
- ZEROOBS – a list of types of data to be places in the PDB with 0 observations (rarely used)

UTILBLDP – Documentation/Example

- Many other parameters are documented in the MXG SOURCLIB member UTILBLDP
- Add the TYPE6156 data to the PDB along with HSM and suppress the TYPE74 data

```
 %UTILBLDP(
      BUILDPDB=YES,
      USERADD=6156 HSM/251,
      SUPPRESS=74,
        OUTFILE=INSTREAM
 );
```

UTILBLDP – Generated SAS Code

```
SOURCE CODE CREATED BY UTILBLDP
 /**********************************************************/
 /* COPYRIGHT 1999,2015 MERRILL CONSULTANTS DALLAS TX USA  */
 /* THIS SYSIN WAS CREATED BY UTILBLDP AT CHANGE 33.269.   */
 %LET MACKEEP=%QUOTE(
   /*MXG STRONGLY RECOMMENDS PUTTING THE FOLLOWING*/
   /*MACRO DEFINITIONS FOR THE ID MACROS IN YOUR  */
   /*USERID.SOURCLIB(IMACKEEP) MEMBER, RATHER THAN*/
   /*IN THE SYSIN INPUT, SO THEY ARE ALWAYS DEFINED*/
   MACRO _IDHSMDS 251 %
   MACRO _CDE74 IF 99 = -99 THEN RETURN; %
   /* MXGWARN: ONE OR MORE OF THE RMF RECORDS NEEDED */
   /* MXGWARN: BY RMFINTRV HAS BEEN SUPPRESSED. SOME */
   /* MXGWARN: FIELDS MAY BE EMPTY IN RMFINTRV.      */
   /* MXGWARN: SUPPRESSED RECORDS ARE: 74      */
   MACRO _S74      %
```

UTILBLDP – Generated SAS Code

```
MACRO _VARUSER /* USER SMF _VAR DEFINITIONS */
  _VAR6156
  _VARHSM
  _VAR113
 %
 MACRO _CDEUSER /* USER SMF _CDE DEFINITIONS */
  _CDE6156
  _CDEHSM
  _CDE113
 %
);
%LET EPDBOUT=%QUOTE(
 _S6156
 _SHSM
 _S113
);
```

UTILBLDP – Generated SAS Code

```
%LET EPDBINC=%QUOTE(
   VMAC6156
   VMACHSM
   VMAC113
 );
   %LET PTY74=&MXGWORK;      /* NO OUTPUT */
   %LET PTY74CA=&MXGWORK;    /* NO OUTPUT */
   %LET PTY74CF=&MXGWORK;    /* NO OUTPUT */
   %LET PTY74CO=&MXGWORK;    /* NO OUTPUT */
   %LET PTY74LK=&MXGWORK;    /* NO OUTPUT */
   %LET PTY74ME=&MXGWORK;    /* NO OUTPUT */
   %LET PTY74OM=&MXGWORK;    /* NO OUTPUT */
   %LET PTY74PA=&MXGWORK;    /* NO OUTPUT */
   %LET PTY74ST=&MXGWORK;    /* NO OUTPUT */
   %LET PTY74SY=&MXGWORK;    /* NO OUTPUT */
   %LET PTY74TD=&MXGWORK;    /* NO OUTPUT */
   %LET PTY746B=&MXGWORK;    /* NO OUTPUT */
   %LET PTY746F=&MXGWORK;    /* NO OUTPUT */
   %LET PTY746G=&MXGWORK;    /* NO OUTPUT */
```

UTILBLDP – Generated SAS Code

```
 /* NOW RUN BUILDPDB */
 %INCLUDE SOURCLIB(BUILDPDB);
 /* ADDITIONAL CODE INCLUSIONS */
 %INCLUDE SOURCLIB(ASUM113);
 %INCLUDE SOURCLIB(ASUMUOW);  /* RECOMMENDED */
 %INCLUDE SOURCLIB(ASUMCICX); /* RECOMMENDED */
 %INCLUDE SOURCLIB(ASUM70PR); /* RECOMMENDED */
 %INCLUDE SOURCLIB(ASUMTAPE); /* RECOMMENDED */
 %INCLUDE SOURCLIB(ASUMTMNT); /* RECOMMENDED */
 %INCLUDE SOURCLIB(ASUMTALO); /* RECOMMENDED */
 %INCLUDE SOURCLIB(ASUMDBAA); /* RECOMMENDED */
RUN;
```

UTILBLDP – Generated SAS Code

```
MACRO _VARUSER _VARUSER %
MACRO _CDEUSER _CDEUSER %
%LET MACKEEP=;
%LET EPDBOUT=;
%LET EPDBINC=;
%LET EPDBCDE=;
%LET EPDBVAR=;
%LET MACDB2H=;
%LET MAC110H=;
%LET MACFILE=;
OPTIONS OBS=9223372036854775807;
%LET DONEANALID= ;
   MACRO _CDE74 _CDE74 %
   MACRO _S74 _S74 %
 /* USER SMF RECORD CLEAR  _CDE _VAR MACROS */

MACRO _VAR6156 _VAR6156 %
MACRO _CDE6156 _CDE6156 %
MACRO _S6156 _S6156 %
MACRO _VARHSM _VARHSM %
MACRO _CDEHSM _CDEHSM %
MACRO _SHSM _SHSM %
MACRO _VAR113 _VAR113 %
MACRO _CDE113 _CDE113 %
MACRO _S113 _S113 %
```

AD-Hoc Read of Multiple SMF Types

- Assume you need the 1415 42 and 74 records for an analysis of IO

```
FILENAME SMF FTP ("'your daily smf tape dsn'")
    USER='userid' HOST='IP address of zOS' DEBUG
   S370VS LRECL=32760 PASS='password'
    RCMD='SITE RDW READTAPEFORMAT=S';
%UTILBLDP(BUILDPDB=NO,
          USERADD=1415 42 74,
          OUTFILE=INSTREAM
);
%INCLUDE INSTREAM;
```

BLDSMPDB – AUTOALOC Parameters

- AUTOALOC=YES – invokes VMXGALOC to automatically create and manage SAS data libraries – almost like a GDG on zOS
- BASEDIR=C:\MXG – but can be any drive/directory you choose – this where the directories will be created
- BASECICS=C:\MXG – as above can be anywhere you choose – this is the destination for CICSTRAN
- BASEDB2=C:\MXG – as above the destionation for DB2ACCT
- IF BASECICS/BASEDB2 are left blank they will default to the same destination as BASEDIR

BLDSMPDB – AUTOALOC Parameters

- DATEFMT – yymmdd probably makes the most sense

```
 datefmt=date
the format of the date value that is imbedded in the directory
names. valid values are:
             mmddyy - length can be 6 8 or 10
             mmddyy6 produces 070511
             mmddyy8 produces 07-05-11
             mmddyyn8 produces 07052011
             ddmmyy - length can be 6 8 or 10
             julian - length can be 5 or 7
             yymmdd - length can be 6 8 or 10
             date   - default value of date7.
```

BLDSMPDB – Directory Names

- Assume you used BASEDIR=C:\MXG and DATEFMT=YYMMDD6
  - Daily directories Dyymmdd – where date is the date of the data so if you ran today it would be today-1
  - Weekly directories Wyymmdd – where date is the beginning of the week
  - Monthly directories Myymmdd – where date is the first of the month
  - Trend directories – Tyymmdd – where date is the date the TREND was run
  - SPIN directories – Syymmdd – where the date matches the DAILY

BLDSMPDB – AUTOALOC Parameters

- WEEKSTRT=MON – start weeks on this day of the week – MON has always been the MXG default but it can be any day of the week
- How many generations to keep?
  - DAY2KEEP=14 or as many as you like
  - CICSKEEP=14
  - DB2KEEP=14
  - WEK2KEP=12
  - MTH2KEP=100

BLDSMPDB – Parameters

- RUNDAY=YES – run the daily process
  - NO – do not run DAILY (used for recovery of WEEK/MONTH)
  - PDB – write only to PDB with no DAILY/WEEKLY processing
- RUNWEEK=YES – run the weekly process
  - NO – do not run weekly
  - WTD – run week-to-date
- RUNMNTH=YES – run the monthly process
  - NO – do not run monthly
  - MTD – run month-to-date
- RUNTRND=WEEKLY – run TRENDing when WEEK is run
  - NO – do not run TRENDing
  - DAILY – run with DAILY job

BLDSMPDB Parameters

- WEEKKEEP=_ALL_ - KEEP all datasets in daily PDB at WEEK level
  - Or a list of datasets to keep
- WEEKDROP=SPIN: SPUN: - DROP these datasets at WEEK level
  - Or a list of datasets to drop
- MNTHKEEP=_ALL_ - KEEP all datasets in WEEK1 at MONTH level
  - Or a list of datasets to keep
- MNTHDROP=SPIN: SPUN: - DROP these datasets at MONTH level
  - Or a list of datasets to drop

BLDSMPDB Parameters

- There are many other parameters available to control the way BLDSMPDB executes
- There is documentation of all in the BLDSMPDB member of the MXG SOURCLIB
- UTILBLDP/BLDSMPDB can also be used on zOS – AUTOALOC parameters will be ignored

BLDSMPDB Parameters

- BUILD=BUILDPDB – your BUILDPDB code
  - If you used OUTFILE=INSTREAM in UTILBLDP then make it BUILDPDB=INSTREAM
  - If you want to use the code you had as SYSIN for your job on zOS and it is a member of your USERID.SOURCLIB then BUILDPDB=YOURCODE

UTILCPLG – Copy the LOG/LIST dataset

- Copies the LOG and LIST of the SAS session
- %UTILCPLG(OUTDIR=directory);
- Embeds the date and time in the dataset name so that you can see the LOG/LIST output of your jobs

BLDSMPDB Example

- Assume you have mapped a network drive on your SAN to M with a directory named MXG as the destination for the data
- Assumes you used UTILBLDP and default values for other things
- This is the ONLY job you need

```
FILENAME SMF FTP ("‘your daily smf tape dsn'")
   USER=‘userid' HOST=‘IP address of zOS' DEBUG
   S370VS LRECL=32760 PASS=‘password‘
   RCMD='SITE RDW READTAPEFORMAT=S‘;
%UTILBLDP(OUTFILE=INSTREAM,BUILDPDB=YES,USERADD=HSM/251);
%BLDSMPDB(BUILDPDB=INSTREAM,WEEKSTRT=MON,
         RUNDAY=YES,RUNWEEK=YES,RUNMNTH=YES,RUNTRND=WEEKLY,
     BASEDIR=M:\MXG,BASECICS=,BASEDB2=,AUTOALOC=YES);
     %UTILCPLG(OUTDIR=M:\mxg);
```

Setting up the job

- Copy your MXG shortcut and add:

```
      -SYSIN 'c:\mxg\userid\yourjob.sas'
  -LOG 'c:\mxg\yourjob.log.txt'
      -PRINT 'C:\mxg\yourjob.list.txt'
```

- Test the job by clicking on the new shortcut – it will run in the background
- Add to a schedule – there are several options
  - Use the Windows scheduler based on time of day
  - If you have scheduling software that interfaces between zOS and ASCII then build a schedule based on the creation of the daily SMF data

Setting up the job

- Could also be setup as a BAT file
- Just copy the shortcut properties into a BAT file and make the same additions as shown above
- I have always found shortcuts simpler

What about reruns?

- FORCEDAY= forces the run date and allocates the correct LIBNAMES for that day. This is the day of the DATA NOT the day of the run. So if there was a problem with the job that ran on Jan 27 to rerun you would specify Jan 26.
  - Can be 26JAN16 (SAS date format)
  - Or TODAY()-x where x is the number of days to go back

Allocating LIBNAMEs for Ad-Hoc Reporting

- If all you need is the current set of LIBNAMEs re-execute VMXGALOC with READONLY=YES if you do not want changes made
- You may want to put VMXGALOC in your IMACINIT so that the current LIBNAMEs are always allocated
- VGETALOC will allocate a range of dates for reporting.
- LIBNAMES PDB1-PDBx will be allocated allowing you to exploit VMXGSET to get all of the data
  - DATA STEPS;
  - %VMXGSET(DATASET=STEPS);

VGETALOC Usage

- GETDATERANGE=
  - A range of SAS date values separated by spaces
- DATEFORMAT=
  - The date format you used in VMXGALOC
- TYPEOFDATA=
  - Daily Weekly Monthly CICS DB2
- BASEDIR=
  - The basedir you used in VMXGALOC

Sample Program Using VGETALOC/VMXGSET

```
%VGETALOC(BASEDIR=E:\,
     DATEFORMAT=YYMMDD,
     DATERANGE=01jan16 31jan16,
     TYPEOFDATA=WEEKLY
);
DATA STEPS;
%VMXGSET(DATASET=STEPS);
```

VMXGSET

- Also works on zOS but uses VGETDDS to find the DDNAMEs
- %VGETDDS(DDNAMES=PDB:) will find all of the DDNAMEs allocated with a DDNAME of PDBx where x is a number from 1-99
- Adding DEFER=YES to the VMXGSET call will allow UNIT=AFF in the JCL to limit tape drive usage

The more things change…

- The more they change
- MXG is constantly being modified/enhanced/repaired
- Stay current with the CHANGES published at MXG.COM
- Putting in a new version is not particularly onerous (depending on the amount of change control paperwork you may have to perform.)