Migration Guide for z/OS

SAS to WPS

"To guide you through the tasks for migrating your existing z/OS SAS data and applications for use with WPS."

world programming

Version: 3.1.8

www.worldprogramming.com

Contents

Introduction

About This Guide

Overview

This document will help guide you through installing the World Programming System (WPS) on the z/OS platform. It also has sections on how to use WPS and what to do if you have any existing programs written in the language of SAS and any data associated with them.

Notation

Whenever you need to type in code or the guide is showing a screenshot of some code, it will be indicated like this:

```
Rem : Here is some code
```

For the most part, this guide will indicate filenames, paths, folders and single commands or phrases with a different font, `like this`.

Suggested values or user defined values will be shown between < >.

Overview

When migrating an environment to WPS software on z/OS, the following aspects should be taken into consideration:

- Migrating JCL
- Migrating Data
- Migrating programs written in the language of SAS
- Performance Considerations
- MXG Migration Considerations
- Migrating Stored Macro Libraries

Legal Notices

General Acknowledgements

- World Programming Limited is not associated in any way with the SAS Institute
- WPS is not the SAS System
- The phrase "language of SAS" used in this document is used in a generic way to describe the computer language often called "the SAS language" or simply "SAS"

Trademarks

- WPS and World Programming are registered trademarks or trademarks of World Programming Limited in the European Union and other countries. (r) or ® indicate European Community registration.
- SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.
- MXG is a trademark of Merrill Consultants
- All other trademarks are the property of their respective owner

Third-Party Notices

WPS includes software developed by third parties. More information can be found in the THANKS or acknowledgments.txt file included in the WPS installation.

Migrating JCL

Overview of JCL Migration

There are a small number of changes that will need to be made to JCL when migrating to WPS software.

WPSHOST

The actual executable program name in the WPS load library is `WPSHOST`. In general this is called from `WPSPROC`, but if you need to call WPS directly use a statement similar to `EXEC PGM=WPSHOST`.

WPSPROC

In the WPS `CNTL` library provided with the installation, there is a member called `WPSPROC`. This contains a JCL procedure called WPSPROC that sets up any required DDNAMES for a default configuration. If your site uses a common equivalent `SASPROC` JCL procedure, modifying jobs to use WPS software is as simple as changing the `PROCLIB` and changing `EXEC SASPROC` to `EXEC WPSPROC`.

DDNAMES required

If your site does not use a common JCL procedure, the changes that need to be made will be more widespread as each JCL job will need to be modified to a greater extent. DDNAMES will need to be modified/added/removed to suit WPS. These changes will vary from site to site depending on usage.

The following files are essential for initialising WPS:

- `//STEPLIB` - Location of the WPSHOST executable - unless the library is specified in the LNKLST
- `//SETINIT` - Location of licensing information once WPS has been installed
- `//CONFIG` - Location of a list of default OPTION settings
- `//WORK` - Location of a WPS-formatted file (usually temporary) for intermediate files
- `//SASHELP` - Location of translation tables for language encoding, among other support items

The following file is opened very early in the WPS initialisation sequence IF it is specified in the CONFIG file:

- `//NEWS` - Default 'News' file for display at the start of the SASLOG

The following files provide for client-site overrides

- `//CEEOPTS` - Language Environment (LE) options
- `//DFSPARMS` - Basic host sort parameters

The following files are only accessed if required

- `//ODSCSS` - Default CSS stylesheet for ODS HTML
- `//WPSAOINI` - WPS specific DB2 configuration
- `//DSNAOINI` - Default DB2 connection configuration

- `//SASAUTOS` - Macro auto-call library
- `//WPSFONTS` - Location of TrueType font definitions

Various SYSOUT files, may be omitted, re-directed to disk files, or DUMMY'd, as required

- `//SASLOG` - Program source listing, progress notes
- `//SASLIST` - Output from various PROCs used in the program
- `//SORTMSGS` - Output generated by calls to the host SORT, depending on settings in `//DFSPARMS`
- `//WPSTRACE` - Output target for WPS service trace output
- `//CEEDUMP` - Output target for crash dump files
- `//CEERPT` - Output target for CEE report information
- `//SYSPRINT` - For LE output, equivalent to 'stdout'
- `//SYSOUT` - For LE output, equivalent to 'stderr'
- `//DSNTRACE` - For DB2 trace output. Use with care. The amount output involved could be very large

Migrating Programs

Overview of Program Migration

There is some syntax of the language of SAS that is not supported by WPS. To determine those programs that may contain unsupported language syntax, three methods can be used: Manual inspection, Automated static analysis, and Automatic dynamic analysis.

Manual Inspection

The *WPS Reference for Language Elements* document supplied with the WPS software contains an up-to-date list of all language syntax elements currently supported by WPS. This document is updated each time a new version of WPS software is released.

Automated Static Analysis

A tool that analyses programs written in the language of SAS is supplied with the WPS Workbench software. This is available on all non-z/OS platforms with a powerful Graphical User Interface (GUI). It can, however, be used to analyse programs from the z/OS environment. This product contains a language analyser that will produce two reports:

- Usage report showing all syntax usage of the language of SAS in the analysed programs
- Compatibility report showing only those language of SAS syntax elements that are not supported by WPS

The analyser can process a single file or a complete directory of files including subdirectories.

To analyse programs in the language of SAS from a z/OS environment, these programs must first be downloaded to the PC environment in ASCII text format. This can be done in a number of ways including:

- Convert the mainframe files to XMIT format and transfer to the PC in binary mode using for example ftp or TN3270 file transfer utility. Then unpack using a PC-based XMIT tool such as the commonly used XMIT manager. XMIT manager cannot handle PDSE files, these should be copied to a PDS before being XMITed.
- Download directly using ftp in text mode - i.e. convert to ASCII from EBCDIC as the transfer is made.

Contact World Programming for details of obtaining a PC version of WPS suitable for performing static code analysis, and for advice on file transfer.

Dynamic Analysis

It is possible to analyse a live production z/OS system for usage of language of SAS syntax. This requires specialist tools and services from World Programming and can provide invaluable insight for larger sites before and during any migration project to WPS.

Migrating Data

Overview of Data Migration

There are various methods that can be used to migrate the contents of existing Data Libraries produced using SAS® software on the z/OS platform, including from both DASD and TAPE media, into WPS format.

WPS can read z/OS-based SAS data set libraries (SAS V6 and later) directly using the SASDASD library engine. At the time of writing the WPS SASDASD engine has the following restriction:

- No support for reading CATALOGS on DASD. CATALOGS will be skipped when importing a SAS DASD LIBRARY into WPS. Format catalogs will need to be recreated from source using WPS, or exported from SAS using `PROC FORMAT CNTLOUT` and re-imported to WPS using `PROC FORMAT CNTLIN`.

This means that you do not need to convert historical SAS DASD data set libraries to WPS format if they are purely MTYPE=DATA. However, WPS cannot write to the SASDASD data library format. Therefore, persistent SAS data libraries that get updated during production processing will need to be migrated to WPS data libraries. This can be achieved using the COPY procedure.

WPS include full support for standard physical sequential multi-volume data libraries.

WPS can read standard SAS TAPE data libraries created with the V7TAPE, V8TAPE and V9TAPE data library engines of SAS software (these are very similar). WPS uses the SASSEQ engine to read these V7TAPE, V8TAPE and V9TAPE TAPE-based data libraries. At the time of writing the WPS SASSEQ engine has the following restriction:

- No support for reading CATALOGS on TAPE; support for this is not yet planned. CATALOGS are skipped when importing a V7TAPE, V8TAPE and V9TAPE DATA LIBRARY into WPS.

WPS can also write data to the SASTAPE data library format.

Migrating Data from SAS® software DASD Data Libraries

The process of migrating the contents of a SAS software DASD data library to WPS is a one step process using the WPS SASDASD engine to copy data into WPS DASD data library format.

Example Migration Job

Example batch job JCL and macros in the language of SAS are provided in the @SAS2WPS member of the supplied WPS CNTL LIBRARY. The contents of this member is shown shown in Appendix A.

Follow the instructions at the top of the example JCL to edit the required parameters of the JOB, then submit.

Migrating Stored Macro Libraries

Migration of Stored Macro Libraries

Stored Macro libraries generated by SAS must be migrated to WPS before use.

If the original source program that created the stored macro(s) is available, simply run the program under WPS, after making suitable changes to the requisite DD statements. As always, WPS will not write to a SAS Library, so new WPS-based Libraries must be allocated and used.

If the originating source program is not available, it may be that the stored macro(s) were generated with the `/ STORE SOURCE` options specified. If this is so, then the macro source can be retrieved using the `%copy <macroname>/source;` statement. This results in the original source code being replicated in the SASLOG. Given this source code it would be a relatively simple task to build a WPS-generated stored macro library.

If the originating source code is completely unavailable, please contact World Programming for further assistance.

MXG Migration Considerations

MXG Migration Overview

The examples in this guide make the assumptions that:

1. This is a basic MXG+SAS software installation with a single LPAR producing SMF data and a single reporting repository situated on the same LPAR
2. Your DAILY PDB is deleted each day.

Your MXG installation may differ from that described above to a small or large extent. The MXG migration section of this guide is structured as a worked example. There is a section giving general advice on larger MXG installations. If it is not clear how the migration process can be applied to your particular MXG installation, please contact World Programming for assistance and advice.

The migration process outline is as follows:

- Verify Pre-requisites
- Prepare MXG+WPS software
- Migrate MXG Data
- Validate PDB
- Other MXG Issues

Verify Pre-requisites

MXG version 25.11 or later supports running using WPS software. This means that no modifications should be required to your MXG code to be able to run basic MXG processing using WPS.

It is recommended that the latest available version of WPS is installed as this will contain the latest fixes and enhancements to provide the best platform for running MXG.

Install Latest Version of WPS

Install the latest version of WPS onto your z/OS mainframe system. There are step-by-step instructions for doing this in the WPS z/OS installation & user guide. Ensure that you have applied a license key using the documented setinit procedure and have run the Installation Verification job (@VERIFY) to check that the WPS installation has been successful.

Upgrade Production MXG+SAS software to Latest MXG Version

It is advised that you have upgraded your production MXG+SAS software installation to the latest version of MXG before you move to the next phase in the SAS software to WPS software migration process. It is much easier to compare the PDB processing output of MXG+WPS software to that of MXG+SAS software if the two environments are running the same version of MXG.

MXG version 25.11 or later supports running under WPS product and should require little or no customisation in order to be able to run under WPS in a standard configuration.

If you are unable to upgrade to the latest version of MXG, please contact World Programming for advice.

Modifcations for MXG

The best way to apply any required modifications to MXG is to create a `TAILRORED SOURCLIB` (normally referred to as the 'USERID.SOURCELIB') and place any modified members in it. This will then be included before the main MXG SOURCLIB in JCL and modified members will override the standard members.

Prepare MXG+WPS

Now that the latest MXG is installed and the latest WPS is installed, a parallel MXG+WPS software application can be set up based on the configuration of the MXG+SAS software application. This involves the following steps:

WPS Config

An example config file for use of MXG with WPS software is supplied with MXG as the member named configw2.

WPS JCL Procedure

An example JCL Procedure file for use of MXG with WPS software is supplied with MXG as the member named MXGWPSV2.

Generate the FORMATS library

WPS cannot currently read catalogs generated by SAS Software. This includes the catalog used to store the MXG FORMATS library. To generate the MXG+WPS software FORMATS library, edit the `@MXGFMTS` as per the instructions at the top of the member and submit. See Appendix A for a listing of the contents of the `@MXGFMTS` member.

Allocate MXG+WPS DAILY PDB and SPIN Datasets

These data sets can be allocated manually using, for example, ISPF option 3.2, or they can be allocated as part of the example BUILDPDB job `@MXGPDB` (contents listed in Appendix A.

It is assumed that the data sets will be allocated in the job in which case the rest of this section can be skipped. If you choose to allocate the data sets manually, then you must comment out the allocation step at the start of the BUILDPDB job.

Prepare the MXG+WPS Job (JCL)

Take a copy of the MXG+SAS software Job used to run the DAILY BUILDPDB, e.g. copy `<mxgsaspfx>.MXG.SAS.CNTL(BUILDPDB)` to `<mxgwpspfx>.MXG.WPS.CNTL(BUILDPDB)`. Now edit `<mxgwpspfx>.MXG.WPS.CNTL(BUILDPDB)`, change:

- `JOBCARD` as appropriate to distinguish MXG+SAS software output from MXG+WPS output
- `PROCLIB` to point to the `<wpspfx>(WPSPROC)`

- `EXEC` statement to call `WPSPROC`
- `LIBRARY` DD to point the newly created FORMATS library
- `PDB` DD to point to newly allocated DAILY PDB
- `CICSTRAN` DD to point to newly allocated CICSTRAN library
- `DB2ACCT` DD to point to newly allocated DB2ACCT library
- `SPIN` DD to point to newly allocated DAILY SPIN file
- `SMF` DD to point to dummy (null)

See Appendix A for a simple example of what a BUILDPDB JCL may look like.

Run the Empty MXG+WPS Job

Now run the MXG+WPS software job created in the previous step. This will initialise MXG and process the buildpdb macros and DATA steps without actually processing any SMF data. The return code should be 0 (zero). If it is not, use normal diagnostic processes to troubleshoot until a return code of 0 is attained. See the WPS Installation and User Guide for help.

Check the contents of SASLIST in the output. This will contain the results of the PROC CONTENTS and should show all the datasets of the DAILY PDB (with zero records in all datasets).

Provide some SMF Data

Now change the WPS+MXG JCL changing the dummied-out SMFINPUT DD to point to a file containing SMF data e.g. change:

```
//SMF DD DISP=SHR,DSN=DUMMY
```

to

```
//SMF DD DISP=SHR,DSN=<SMF data file name>
```

Submit the job

Check SASLIST in the output. The PROC CONTENTS output should now show all datasets of the DAILY PDB with some data in some of the datasets, depending on what the SMF data contained.

At this point, the basic MXG+WPS software installation is complete.

Migrate MXG Data

WPS cannot currently write to SAS DASD data libraries directly, therefore, if there is a need to update these libraries, a migration process must be carried out to convert the MXG+SAS DASD data libraries to MXG+WPS Data libraries.

Typically an MXG+SAS software environment has a DAILY PDB and SPIN file as SAS software DASD data libraries. The SPIN file is carried over each day and used as input to the next day's DAILY PDB processing. The MXG+SAS software SPIN file therefore needs to be migrated to the MXG+WPS software environment. Note there needs to be a copy of the MXG+SAS software SPIN file **prior** to running the

MXG+SAS software DAILY BUILDPDB process as this will generate a new SPIN file. For simplicity, this will generally mean "yesterday's" SPIN file, created by the previous run of BUILDPDB.

The DAILY PDB only needs to be migrated for comparison purposes, i.e. once MXG+WPS software DAILY BUILDPDB has been run successfully we need to check that the resultant PDB is the same as that produced by MXG+SAS software.

Finally, the daily SMF data must be available for input to the MXG+WPS software process.

The standard data migration procedure as described elsewhere in this guide can be used to transfer the data from the MXG+SAS software environment to MXG+WPS software.

Assuming that the previous step has been completed so that the MXG+WPS software FORMATS library has been generated, the empty MXG+WPS software DAILY PDB and SPIN file have been allocated, and that MXG+WPS software has been tested with no SPIN file, we can now migrate the MXG+SAS MXG data environment to MXG+WPS software for testing and validation.

Run the SPIN File Data Migration Job

Use the example batch job `@SAS2WPS` to migrate the SPIN file. Check the SASLOG for successful completion and SASLIST for the output from PROC DATASETS of the complete migrated data library.

Run MXG+SAS® Software

For comparison, we need to be able to compare the PDB produced by MXG+WPS software against the PDB produced by MXG+SAS software. Now is a good time to run the MXG+SAS software using the SPIN file and SMF data that will be used in the MXG+WPS software test.

Validate PDB Processing

Now that there is a working MXG+WPS software installation, the SPIN FILE has been migrated, and the MXG+SAS DAILY BUILD PDB has been run ready for comparison, the MXG+WPS DAILY BUILDPDB process can be run, and the output compared to that generated by MXG+SAS.

1. Run MXG+WPS BUILDPDB Process

Using the migrated SPIN FILE and the real SMF data, run the MXG+WPS software DAILY BUILDPDB process. Check for a return code of zero and sanity check the output in SASLIST.

2. Compare Resulting MXG+WPS PDB

Now there are two DAILY PDBs, one generated by the MXG+WPS software environment, and one generated by the MXG+SAS software environment. The content of the two DAILY PDB libraries must now be compared for consistency/equality.

See appendix A for a listing of the `@COMPARE` JCL and the `XMIGRATE` MACRO member.

Take a copy of the `@COMPARE` member of the WPS CNTL library. Edit the following:

- Change `<wpspfx>` to the WPS installation dataset prefix

- Change <sas-dasd-data-library> to the SAS DASD PDB data library
- Change <wps-dasd-data-library> to the WPS DASD PDB data library
- Change <sas-formats-library> to the MXG FORMATS LIBRARY if necessary
- Change <migration-macro-library> to the MIGRATION macro library

Run the job and check for a return code of zero, then check the SASLIST output for the results of the compare. Note that if you are using the compare macro for MXG PDB comparison, you should set the MXG macro variable to YES. This will cause the compare to ignore the ZDATE and ZTIME variables which will always contain different values since these are the timestamps of the run time of the BUILDPDB process.

Larger MXG Installations

MXG installations vary in size and complexity. The previous steps have demonstrated how a simple single LPAR site may process SMF data into a DAILY PDB that is recreated each day. Discussed below are some other topics relevant to an MXG installation and migration.

Larger Installations

Larger MXG installations will often process SMF data from more than one LPAR. Sometimes the SMF will be processed on the LPAR and the resulting PDB merged into a central repository. Alternatively the raw SMF data from LPARS may be fed to a single CENTRAL system for processing and cross-LPAR reporting, or on very large sites, fed to a number of HUBS for processing of the raw SMF data, the HUBS then supplying PDB data for merging onto a CENTRAL system as required.

MXG Reporting

Reporting from MXG tends to vary from site to site. Some notes are provided below which may be useful in the area.

Standard MXG Reporting

Most reports from MXG should run with no additional migration work required.

Custom MXG Reporting

It is not possible to provide exact instructions for migrating and validating user-written reporting processes within a generic guide such as this.

This is best dealt with on an individual basis. Please contact World Programming for guidance and assistance.

In certain circumstances the printed report output layout may differ slightly from SAS to WPS. In some cases this may require minor adjustments to downstream processes that consume such output.

Handling compressed CICS and DB2 SMF data

Since CICS TS V3.2 and DB2 version 10, generated SMF data can be in compressed form. The actual compression technique (known as *Run Length Encoding*) is the same for both types of data. Consult your CICS and/or DB2 system administrators to understand the situation at your site.MXG code can automatically detect and decompress such data if it is presented to a data input program such as `buildpdb`, but this has an extremely high CPU cost. It is far better to activate the `CICSIFUE` exit supplied in the MXG `SOURCELIB` as member `EXITCICS`.
Review the instructions in this member to inform your decisions about how you will implement the exit at your site. Note that it is necessary to change the parameter string passed to the `LKED` step due to current limitations within WPS. The required change is from:

```
//LKED EXEC PGM=IEWBLINK,
//            PARM='XREF,LIST',
//            COND=(0,NE,ASM)
```

to:

```
//LKED EXEC PGM=IEWBLINK,
//            PARM='XREF,LIST,LET,RMODE=ANY,AMODE=31,RENT,REUS,REFR',
//            COND=(0,NE,ASM)
```

If you omit this change, any WPS program that uses the exit will abort with a `U4038` abend code, even if there is no compressed data in the input file.

The intended end-point is:

1. To have a load library with a program named `CICSIFUE` in it
2. To have the catalogued procedure that it used to invoke WPS with MXG modified so that the load library features in the `STEPLIB` concatenation
3. To have the affected program(s) feature the statement `%LET SMFEXIT=CICS;` as practically the first statement in the program source

For any specific SMF data input program, there are four possible combinations relating to compressed data and the `CICSIFUE` exit. These combinations and the potential results are tabulated below.

The exit is present and active	The input file contains compressed CICS and/ or DB2 data	Result
No	No	Processing continues as normal. There is no compressed data that needs to be handled.
No	Yes	MXG will identify the fact that compressed data is to be processed and will do so using the so-called *dynamic* decompression exit. An accompanying `MXGNOTE` in the `SASLOG` will confirm the situation. The actual text of this note (as of MXG V32.09) is: `ERROR:`

The exit is present and active	The input file contains compressed CICS and/ or DB2 data	Result
		`YOUR <CICS/DB2> RECORDS ARE COMPRESSED, BUT YOU ARE NOT USING THE CICSIFUE "INFILE EXIT" TO DECOMPRESS YOUR <CICS/DB2> SMF RECORDS ON Z/OS. INSTEAD, YOU USED MXG DEFAULT INTERNAL CODE THAT CORRECTLY, BUT EXPENSIVELY, DECOMP-S; IT TAKES 20-30 TIMES MORE CPU TIME THAN THE DESIGNED-FOR-Z/OS CICSIFUE INFILE EXIT. READ THE INSTRUCTIONS IN MEMBER EXITCICS TO CREATE THE CICSIFUE LOAD MODULE AND ENABLE MXG TO USE THE ASM-CODE EXIT, TO DECOMPRESS EITHER DB2 AND/OR CICS COMPRESSED SMF DATA. SEE CHANGE 27.260.`
Yes	No	Processing continues as normal. There is no compressed data that needs to be handled. The exit is not triggered. However, the presence of the exit is noted in the SYSLOG: `+CICSDB2: CICSIFUE EXIT VERSION 3 FOR CICS/DB2 DATA`
Yes	Yes	Processing continues as normal. The exit automatically decompresses the data as it is read. The presence of the exit is noted in a message in the SYSLOG: `+CICSDB2: CICSIFUE EXIT VERSION 3 FOR CICS/DB2 DATA`

Appendix A - Examples

Example WPS Migration JCL

(provided in the <wpspfx.>CNTL library as member @SAS2WPS)

```
// <add a jobcard here>
//PROCLIB  JCLLIB ORDER=(<wpspfx>.CNTL)
//*
//* SYMBOL SASDL IS THE DSNAME OF THE SAS DATA-LIBRARY ON DASD
//        SET SASDL=<sas-dasd-data-library>
//*
//* SYMBOL WPSDL IS THE DSNAME OF THE WPS DATA-LIBRARY ON DASD
//        SET WPSDL=<wps-dasd-data-library>
//*
//* NOTE : USE REGION=0M TO OBTAIN MAXIMUM AVAILABLE MEMORY
//*
//*--------------------------------------------------------------------*/
//* SAMPLE JOB TO MIGRATE A SAS DASD DATA LIBRARY TO WPS DASD          */
//*--------------------------------------------------------------------*/
//*
//* (1)  ADD A SUITABLE JOBCARD
//* (2)  CHANGE <wpspfx> TO THE WPS INSTALLATION DATASET PREFIX
//* (3)  CHANGE <sas-dasd-data-library> TO THE SOURCE SAS DATASET
//* (4)  CHANGE <wps-dasd-data-library> TO THE WPS TARGET DATASET
//* (5)  CHANGE <wps-procedure-name> TO THE WPS PROCEDURE NAME
//* (6)  SUBMIT THIS JOB AND THEN CHECK THE OUTPUT
//* (7)  CHECK FOR A JOB RETURN CODE OF ZERO
//*
//*--------------------------------------------------------------------*/
//* TO MIGRATE A SAS TAPE DATA LIBRARY TO WPS DASD,                    */
//*   REPLACE 'LIBNAME SASDL SASDASD;' WITH 'LIBNAME SASDL SASSEQ;'    */
//*--------------------------------------------------------------------*/
//*
//WPS      EXEC <wps-procedure-name>
//SOURCLIB DD DISP=SHR,DSN=<wpspfx>.CNTL
//SASDL    DD DISP=SHR,DSN=&SASDL,
//            DCB=BUFNO=32
//WPSDL    DD DISP=(NEW,CATLG),DSN=&WPSDL,
//            UNIT=SYSDA,SPACE=(TRK,(9000,900),RLSE)
//*
//SYSIN    DD DATA,DLM='++'

*OPTIONS SOURCE SOURCE2 MPRINT MACROGEN MLOGIC;

 LIBNAME SASDL SASDASD;
*LIBNAME SASDL SASSEQ;

PROC DATASETS LIB=WPSDL KILL; RUN;
PROC COPY IN=SASDL OUT=WPSDL MEMTYPE=DATA; RUN;
```

```
RUN;

++ END OF //SYSIN
```

Example WPS Migration Macros

(provided in the <wpspfx.>CNTL library as member XMIGRATE)

```
  %*********************************************************************;
  %* SAS SOURCE LIBRARY CONTAINING MACROS TO SUPPORT THE SAS TO WPS    *;
  %* MIGRATION PROCESS. SEE SAS2WPS MEMBER OF CNTL LIBRARY FOR AN      *;
  %* EXAMPLE JCL JOB THAT USES THESE MACROS.                           *;
  %*********************************************************************;


%MACRO WPSORSAS();
  %*********************************************************************;
  %* WPSORSAS                                                          *;
  %* DESCRIPTION: MACRO TO CHECK WHETHER WPS OR SAS IS RUNNING         *;
  %* ARGUMENTS:   NONE                                                 *;
  %* RETURNS:     WPS OR SAS                                           *;
  %*********************************************************************;
  %IF %SYSPROD(WPS) = 1 %THEN
  %DO;
    WPS
  %END;
  %ELSE
  %DO;
    SAS
  %END;
%MEND WPSORSAS;


%MACRO SD2ST(SASDL=SASDL, SASTL=SASTL);
  %*********************************************************************;
  %* SD2ST                                                             *;
  %* DESCRIPTION: MACRO TO EXPORT A SAS DASD LIBRARY TO SAS TAPE       *;
  %*  ARGUMENTS:                                                       *;
  %*        SASDL - THE SOURCE SAS DASD DATA LIBRARY                   *;
  %*        SASTL - THE TARGET SAS TAPE DATA LIBRARY                   *;
  %* NOTE1: THIS MUST BE RUN FROM SAS NOT WPS                          *;
  %*********************************************************************;
  OPTIONS COMPRESS=NO;
  LIBNAME &SASTL TAPE;
  PROC COPY IN=&SASDL OUT=&SASTL NOCLONE MEMTYPE=DATA; RUN;
%MEND SD2ST;


%MACRO ST2WD(SASTL=SASTL, WPSDL=WPSDL);
  %*********************************************************************;
  %* ST2WD                                                             *;
```

```
  %* DESCRIPTION: MACRO TO IMPORT A SAS TAPE LIBRARY TO WPS DASD      *;
  %*  ARGUMENTS:                                                      *;
  %*        SASTL - THE SOURCE SAS TAPE DATA LIBRARY                  *;
  %*        WPSDL - THE TARGET WPS DASD DATA LIBRARY                  *;
  %*                                                                  *;
  %* NOTE1: THIS MUST BE RUN FROM WPS NOT SAS                         *;
  %*******************************************************************;
  LIBNAME &SASTL SASSEQ;
  PROC DATASETS LIB=&WPSDL KILL; RUN;
  PROC COPY IN=&SASTL OUT=&WPSDL; RUN;
  %IF "&VERBOSE" EQ "YES" %THEN
  %DO;
    TITLE2 "CONTENTS OF &WPSDL WPS DASD DATA LIBRARY";
    PROC DATASETS LIB=&WPSDL;
    RUN;
  %END;
%MEND ST2WD;


%MACRO S2WCOMP(SASDL=SASDL, WPSDL=WPSDL, METHOD=, CRITERION=);
  %*******************************************************************;
  %* S2WCOMP                                                         *;
  %* DESCRIPTION: COMPARE SAS TAPE LIBRARY WITH WPS DASD LIBRARY      *;
  %*  ARGUMENTS:                                                      *;
  %*        SASDL     - A SAS TAPE DATA LIBRARY                     *;
  %*        WPSDL     - A WPS DASD DATA LIBRARY                     *;
  %*        METHOD    - METHOD FOR PROC COMPARE                     *;
  %*        CRITERION - CRITERION FOR PROC COMPARE                  *;
  %* NOTE1: THIS MUST BE RUN FROM WPS NOT SAS                        *;
  %*******************************************************************;
  TITLE1 "S2WCOMP - SAS TO WPS DATA MIGRATION LIBRARY COMPARISON";
  TITLE2 "CONTENTS OF &SASDL SAS TAPE DATA LIBRARY";
  PROC DATASETS LIB=&SASDL;
  RUN;
  %IF "&VERBOSE" EQ "YES" %THEN
  %DO;
    TITLE2 "CONTENTS OF &WPSDL WPS DASD DATA LIBRARY";
    PROC DATASETS LIB=WPSDL; RUN;
  %END;

  PROC SQL;
  CREATE TABLE MEMNAMES AS
  SELECT MEMNAME FROM DICTIONARY.MEMBERS
  WHERE LIBNAME LIKE "&SASDL" AND MEMTYPE = 'DATA';

  DATA _NULL_;
  SET MEMNAMES END=LAST;
  LENGTH SASCODE $80;
  SASCODE =
      '%COMPRMBR('
    || "SASLIB=&SASDL"
    ||",WPSLIB=&WPSDL"
    ||",MEMNAME="||TRIM(MEMNAME)
    ||",METHOD=&METHOD"
```

```
    ||",CRITERION=&CRITERION"
    ||")" ;
  CALL EXECUTE(SASCODE);
%MEND S2WCOMP;

%MACRO COMPRMBR(SASLIB=SASDL, WPSLIB=WPSDL, MEMNAME=MEMNAME,
                METHOD=, CRITERION=);
  %*****************************************************************;
  %* COMPRMBR                                                      *;
  %* DESCRIPTION: COMPARE INDIVIDUAL SAS MEMBER WITH WPS MEMBER    *;
  %*  ARGUMENTS:                                                   *;
  %*        SASDL     - THE SAS DATA LIBRARY                       *;
  %*        WPSDL     - THE WPS DATA LIBRARY                       *;
  %*        MEMNAME   - THE MEMBER NAME                            *;
  %*        METHOD    - METHOD FOR PROC COMPARE                    *;
  %*        CRITERION - CRITERION FOR PROC COMPARE                 *;
  %* NOTE1: IF &MXG MACRO VARIABLE IS SET TO YES THEN ZDATE ZTIME  *;
  %*        VARIABLES WILL BE EXCLUDED FROM THE COMPARISON         *;
  %*        THIS IS HELPFUL WHEN COMPARING MXG PDB LIBRARIES       *;
  %*****************************************************************;
  TITLE1 "COMPRMBR - SAS TO WPS DATA MIGRATION MEMBER COMPARISON";
  TITLE2 "MEMBER NAME: &MEMNAME";

  %IF "&VERBOSE" EQ "YES" %THEN
  %DO;
    TITLE3 "SAS DATA LIBRARY: &SASLIB";
    PROC CONTENTS DATA=&SASLIB..&MEMNAME;
    PROC PRINT DATA=&SASLIB..&MEMNAME(OBS=8) LABELANDNAME;
    TITLE3 "WPS DATA LIBRARY : &WPSLIB";
    PROC CONTENTS DATA=&WPSLIB..&MEMNAME;
    PROC PRINT DATA=&SASLIB..&MEMNAME(OBS=8) LABELANDNAME;
  %END;

  TITLE2 "COMPARE RESULTS";
  PROC COMPARE BASE=&WPSLIB..&MEMNAME
    COMP=&SASLIB..&MEMNAME
    OUT=WORK.SWMCOMP_&MEMNAME
    OUTNOEQUAL
 %IF "&METHOD" NE "" %THEN %DO;
 METHOD=&METHOD
 %END;
 %IF "&CRITERION" NE "" %THEN %DO;
 CRITERION=&CRITERION
 %END;
    NOTE;
    %IF "&MXG" EQ "YES" %THEN
    %DO;
    EXCLUDEVAR ZDATE ZTIME;
    %END;
    RUN;

  TITLE "RESULTS OF COMPARE OF MEMBER: &MEMNAME";
  PROC PRINT DATA=WORK.SWMCOMP_&MEMNAME; RUN;
%MEND COMPRMBR;
```

Example MXG FORMATS JCL

(provided in the <wpspfx.>CNTL library as member @MXGFMTS)

```
// <add a jobcard here>
//PROCLIB  JCLLIB ORDER=(<wpspfx>.CNTL)
//*
//*-------------------------------------------------------------------*/
//* SAMPLE JOB TO RUN MXG FORMATS LIBRARY BUILDING JOB                */
//*-------------------------------------------------------------------*/
//*
//* (1) ADD A SUITABLE JOBCARD
//* (2) CHANGE <wpspfx> TO THE WPS INSTALLATION DATASET PREFIX
//* (3) CHANGE <mxgpfx> TO THE MXG SOURCLIB DATASET PREFIX
//* (4) CHANGE <mxgwpspfx> TO THE MXG+WPS DATASET PREFIX
//* (5) SUBMIT THIS JOB AND THEN CHECK THE OUTPUT
//* (6) CHECK FOR A JOB RETURN CODE OF ZERO
//*
//*-------------------------------------------------------------------*/
//*
//*====================================================================
//*
//ALLOC    EXEC PGM=IEFBR14
//LIBRARY  DD DISP=(NEW,CATLG),
//            DSN=<mxgwpspfx>.LIBRARY.WPSDATA,
//            SPACE=(TRK,(70,20)),
//            DCB=(RECFM=FS,LRECL=27648,BLKSIZE=27648),
//            DSORG=PS
//*
//*====================================================================
//*
//@MXGFMTS EXEC WPSPROC,CONFIG='<mxgpfx>.SOURCLIB(CONFIGW2)'
//SOURCLIB DD DISP=SHR,DSN=<mxgpfx>.USERID.SOURCLIB
//         DD DISP=SHR,DSN=<mxgpfx>.SOURCLIB
//LIBRARY  DD DISP=OLD,DSN=<mxgwpspfx>.LIBRARY.WPSDATA
//SYSIN    DD DATA,DLM='++'

*OPTIONS SOURCE SOURCE2 MPRINT MLOGIC MACROGEN;

OPTIONS FMTSEARCH=(LIBRARY);
%INCLUDE SOURCLIB(FORMATS);
RUN;

++ END OF //SYSIN
```

Example MXG BUILDPDB JCL

(provided in the <wpspfx.>CNTL library as member @MXGPDB)

```
// <add a jobcard here>
```

```
//PROCLIB  JCLLIB ORDER=(<wpspfx>.CNTL)
//*
//*-------------------------------------------------------------------*/
//* SAMPLE JOB TO RUN EMPTY MXG BUILDPDB JOB                          */
//*-------------------------------------------------------------------*/
//*
//* (1) ADD A SUITABLE JOBCARD
//* (2) CHANGE <wpspfx> TO THE WPS INSTALLATION DATASET PREFIX
//* (3) CHANGE <mxgpfx> TO THE MXG SOURLIB DATASET PREFIX
//* (3) CHANGE <mxgwpspfx> TO THE MXG+WPS DATASET PREFIX
//* (4) SUBMIT THIS JOB AND THEN CHECK THE OUTPUT
//* (5) CHECK FOR A JOB RETURN CODE OF ZERO
//*
//*-------------------------------------------------------------------*/
//*
//*===================================================================
//*
//ALLOC    EXEC PGM=IEFBR14
//CICSTRAN DD DISP=(NEW,CATLG),
//            DSN=<mxgwpspfx>.CICSTRAN.WPSDATA,
//            SPACE=(TRK,(450,450))
//DB2ACCT  DD DISP=(NEW,CATLG),
//            DSN=<mxgwpspfx>.DB2ACCT.WPSDATA,
//            SPACE=(TRK,(450,450))
//PDB      DD DISP=(NEW,CATLG),
//            DSN=<mxgwpspfx>.PDB.WPSDATA,
//            SPACE=(CYL,(600,150))
//SPIN     DD DISP=(NEW,CATLG),
//            DSN=<mxgwpspfx>.SPIN.WPSDATA,
//            SPACE=(CYL,(10,100))
//*
//*===================================================================
//*
//@MXGPDB  EXEC WPSPROC,CONFIG='<mxgpfx>.SOURCLIB(CONFIGW2)'
//SOURCLIB DD DISP=SHR,DSN=<mxgpfx>.USERID.SOURCLIB
//         DD DISP=SHR,DSN=<mxgpfx>.SOURCLIB
//LIBRARY  DD DISP=SHR,DSN=<mxgwpspfx>.LIBRARY.WPSDATA
//CICSTRAN DD DISP=OLD,DSN=*.ALLOC.CICSTRAN
//DB2ACCT  DD DISP=OLD,DSN=*.ALLOC.DB2ACCT
//PDB      DD DISP=OLD,DSN=*.ALLOC.PDB
//SPIN     DD DISP=OLD,DSN=*.ALLOC.SPIN
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(150,150))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(150,150))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(150,150))
//SMF      DD DUMMY
//SYSIN    DD DATA,DLM='++'

*OPTIONS SOURCE SOURCE2 MPRINT MLOGIC MACROGEN;

OPTIONS FMTSEARCH=(LIBRARY);
%INCLUDE SOURCLIB(BUILDPDB);
RUN;

++ END OF //SYSIN
```