



# ***WPS Migration Guide*** ***for z/OS***

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



Version: 4.0.3

Copyright © 2002-2018 World Programming Limited

[www.worldprogramming.com](http://www.worldprogramming.com)



# Contents

<b>Introduction.....</b>	<b>3</b>
About this guide.....	3
Overview.....	3
<b>JCL migration.....</b>	<b>4</b>
<b>Migrating programs.....</b>	<b>6</b>
Overview of program migration.....	6
Manual inspection.....	6
Automated static analysis.....	6
Dynamic analysis.....	7
<b>Migrating data.....</b>	<b>8</b>
Overview of data migration.....	8
Migrating data from SAS® software DASD data libraries.....	8
<b>Migration of stored macro libraries.....</b>	<b>10</b>
<b>MXG migration considerations.....</b>	<b>11</b>
MXG migration overview.....	11
Verify prerequisites.....	11
Prepare MXG+WPS.....	12
Migrate MXG data.....	14
Validate PDB processing.....	15
Migrating larger MXG installations.....	16
MXG reporting.....	16
Handling compressed CICS and DB2 SMF data.....	16
<b>Appendix A - examples.....</b>	<b>19</b>
Example of JCL for simple migration to WPS.....	19
Example of JCL for migration to WPS via tape.....	20
Example of JCL for format migration to WPS.....	21
Example of WPS migration macros.....	22
Example of JCL for comparing migration output.....	26
Example of MXG FORMATS library build.....	27
Example of JCL for a sample MXG BUILDPDB.....	28
<b>Legal Notices.....</b>	<b>30</b>

# Introduction

## About this guide

### Overview

This guide will help guide you install the World Programming System (WPS) on the z/OS platform. It also contains sections that describe:

- How to use WPS
- What to do if you have any existing programs written in the language of SAS and any data associated with them

### Notation

Commands or code you need to enter is shown as follows:

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

Filenames, paths, folders and folder are shown like `this`.

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

## Overview

To migrate a SAS environment on z/OS to WPS, you need to consider how you will migrate:

- JCL
- Data
- Programs written in the language of SAS
- MXG
- Stored macro libraries

You should also take into account performance issues.

# JCL migration

You need to make a small number of changes to your JCL when you migrate 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, you should use a statement similar to `EXEC PGM=WPSHOST`.

## WPSPROC

The `WPS CNTL` library provided with the installation contains a member called `WPSPROC`. This member contains a JCL procedure called `WPSPROC` that sets up any required `DDNAME` parameters for a default configuration. If your site uses a common `SASPROC` JCL procedure, modify jobs to use WPS software by changing:

- The `PROCLIB`
- `EXEC SASPROC` to `EXEC WPSPROC`

## DDNAMEs required

If your site does not use a common JCL procedure, you need to make more wide-ranging changes, as each JCL job needs to be modified to a greater extent. The `DDNAME` parameters need to be modified, added or removed to suit WPS. These changes vary from site to site depending on usage.

The following files are essential for initialising WPS:

<code>//STEPLIB</code>	Location of the <code>WPSHOST</code> executable, unless the library is specified in the <code>LNKLST</code>
<code>//SETINIT</code>	Location of licensing information once WPS has been installed
<code>//CONFIG</code>	Location of a list of default <code>OPTION</code> settings
<code>//WORK</code>	Location of a WPS-formatted file (usually temporary) for intermediate files
<code>//SASHELP</code>	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:

<code>//NEWS</code>	Default 'News' file for display at the start of the <code>SASLOG</code>
---------------------	---

The following files provide for client-site overrides:

<code>//CEEOPTS</code>	Language Environment (LE) options
------------------------	-----------------------------------



//DFSPARMS            Basic host sort parameters

The following files are only accessed if required:

//WPSAOINI            WPS specific DB2 configuration  
//DSNAOINI            Default DB2 connection configuration  
//SASAUTOS            Macro autocall library  
//WPSFONTS            Location of TrueType font definitions

Various sysout data set files are required, which can be omitted, re-directed to disk files, or ignored using the DUMMY parameter, 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 utility, depending on the 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 of output involved could be very large.

# Migrating programs

## Overview of program migration

Some syntax in the language of SAS is not supported by WPS. You can find out which of your programs contain unsupported language syntax, in one of the following ways:

- *Manual inspection* [↗](#) (page 6)
- *Automated static analysis* [↗](#) (page 6)
- *Dynamic analysis* [↗](#) (page 7)

## Manual inspection

You can manually check the language syntax elements used in your programs against those listed in the **WPS Reference for Language Elements** supplied with WPS. Although this document is updated each time a new version of WPS software is released, we cannot guarantee that the list of language syntax elements is exhaustive.

## Automated static analysis

WPS provides a tool that can analyse programs and generate reports on the usage of the language of SAS by those programs. Two reports are produced:

- A usage report, showing all usage of syntax provided by the language of SAS
- A compatibility report, showing only those syntax elements of the language of SAS that are not supported by WPS

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

The tool runs on the WPS Workbench. To use it with files produced for the z/OS environment, you must first download the files to the PC environment. The files must be downloaded in ASCII text format. Preparing and downloading files can be done in number of ways. For example:

- Convert the mainframe files to XMIT format and transfer them to the PC in binary mode using, for example, FTP or the TN3270 file transfer utility. The XMIT files can then be unpacked using a PC-based tool such as XMIT Manager. Because the XMIT Manager cannot handle PDSE files, these should be copied to a PDS (Partitioned Dataset) before being transferred.

- Download the mainframe files using FTP in text mode - that is, convert the files to ASCII from EBCDIC during the download process.

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

World Programming can analyze a live, production z/OS system to discover how it uses syntax from the language of SAS. World Programming uses specialist tools and services to do this. The analysis can provide invaluable insight for larger sites before and during any migration project to WPS.

# Migrating data

## Overview of data migration

The contents of existing data libraries produced using SAS® software on the z/OS platform, and stored on various sources, including DASD and tape media, can be migrated to WPS. Various methods can be used to migrate the contents.

WPS can read z/OS-based SAS dataset libraries (SAS V6 and later) directly using the `SASDASD` library engine. However, WPS cannot read catalogs on DASD. Catalogs will be skipped when importing a `SASDASD` library into WPS. Catalogs containing formats will need to be recreated from source using WPS, or exported from SAS using `PROC FORMAT CNTLOUT` and then re-imported to WPS using `PROC FORMAT CNTLIN`.

You do not need, therefore, to convert historical `SASDASD` libraries to WPS format if they are specified as `MTYPE=DATA`. However, WPS cannot write to the `SASDASD` data library format. Therefore, persistent SAS data libraries that are updated during production processing will need to be migrated to WPS data libraries. You can do this using the `COPY` procedure.

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

The WPS `SASSEQ` engine can be used to read standard SAS TAPE-based data libraries created with the `SAS V7TAPE`, `V8TAPE` and `V9TAPE` data library engines. However, the `SASSEQ` engine cannot currently read catalogs on tape; any catalogs in a data library created with `V7TAPE`, `V8TAPE` and `V9TAPE` are therefore skipped.

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

## Migrating data from SAS® software DASD data libraries

The contents of a SAS software DASD data library can be migrated to WPS in one step, using the WPS `SASDASD` engine to copy data into WPS DASD data library format.

### Example migration jobs

Example batch job JCL and macros in the language of SAS are provided in three members in the supplied WPS `CNTL` library:



- The @SAS2WPS member shows a simple disk-to-disk migration method (refer to *Example of JCL for simple migration to WPS* [↗](#) (page 19) in Appendix A). The program code is held as part of the member.
- The @SAS2WPX member shows a more complex method, which uses an intermediate tape-based file to provide a backup copy of the original SAS data library (refer to *Example of JCL for migration to WPS via tape* [↗](#) (page 20) in Appendix A).
- The above member also features a comparison of the two sets of data after migration is complete. The program code is held as part of the member, but it uses macros that are defined in the XMIGRATE member (refer to *Example of WPS migration macros* [↗](#) (page 22)).

---

**Note:**

XMIGRATE is not a batch job to be submitted; it simply provides code that is used in the @SAS2WPX job.

---

If the source from your existing SAS® format catalogs is not available, you can use an alternative method to create WPS versions of the format catalogs. To do this, export the formats from the SAS system using `PROC FORMAT CNTLOUT` and re-import into WPS using `PROC FORMAT CNTLIN`. An example batch job is provided in member @FMT2WPS of the WPS CNTL library (refer to *Example of JCL for Format Migration to WPS* [↗](#) (page 21) in Appendix A).

Follow the instructions at the top of the example JCL to edit the required parameters of the job, before submitting it.

# 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 macros is available, run the program under WPS after making suitable changes to the appropriate DD statements. WPS will not write to a SAS Library, so new WPS-based libraries must be allocated and used.

If the original source program that created the stored macros is not available, the stored macros might have been generated with the `/ STORE SOURCE` options specified. If so, the macro source can be retrieved using the `%copy <macroname>/source;` statement. This results in the original source code being replicated in the `SASLOG`. You can then use this source code to build a WPS-generated stored macro library.

If the originating source code is not available, contact World Programming for further assistance.

# MXG migration considerations

## MXG migration overview

This chapter describes how to migrate data from MXG-based SAS programs. The combination of MXG and SAS software is labelled as *MXG+SAS software* in this manual.

This guide assumes that:

- You are migrating from a basic MXG+SAS software installation with a single Logical Partition (LPAR) producing System Management Facilities (SMF) data, and a single reporting repository situated on the same LPAR.
- The Performance Database (PDB) is deleted each day.

Your MXG installation might differ from that described above to a small or large extent.

The MXG migration section of this guide is structured as a worked example. The section *Migrating larger MXG installations* [↗](#) (page 16) gives general advice on larger MXG installations. If it is not clear how the migration process can be applied to your particular MXG installation, contact World Programming for assistance and advice.

In summary, you migrate data as follows:

- *Verify prerequisites* [↗](#) (page 11)
- *Prepare MXG+WPS* [↗](#) (page 12)
- *Migrate MXG data* [↗](#) (page 14)
- *Validate PDB processing* [↗](#) (page 15)

## Verify prerequisites

WPS software supports MXG version 25.11 or later. No modifications should be required to your MXG code to be able to carry out basic MXG processing using WPS.

We recommend 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 on your z/OS mainframe system. Step-by-step instructions for doing this are provided in the *WPS z/OS Installation and User Guide*. Ensure that you have applied a license key 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

We recommend that you upgrade your production MXG+SAS software installation to the latest version of MXG before you move to the next phase of 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 should require no customisation in order to be able to run under WPS in a standard configuration.

If you cannot upgrade to the latest version of MXG, contact World Programming for advice.

## Modifications for MXG

To apply required modifications to MXG, create a tailored `SOURCLIB` (normally referred to as `USERID.SOURCLIB`) 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

When the latest versions of MXG and WPS have been installed, you can set up a parallel MXG+WPS software application based on the configuration of the MXG+SAS software application. This involves the following steps:

### Configure WPS

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

### Create a JCL procedure for WPS

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

### Generate the format library

WPS cannot currently read a catalog generated by SAS Software. This includes the catalog used to store the MXG format library. To generate a format library using the MXG+WPS software, edit `@MXGFMTS` as described in the instructions at the top of the member, and submit.

See Appendix A for a listing of the contents of the `@MXGFMTS` member (refer to *Example of MXG FORMATS library build* [↗](#) (page 27)).

### Allocate MXG+WPS Daily PDB and Spin Datasets

The daily PDB and SPIN datasets can be allocated manually using, for example, ISPF option 3.2, or they can be allocated as part of the example `BUILDPDB` job `@MXGPDB` (the contents of this job are listed in the section *Example of JCL for a sample MXG BUILDPDB* [↗](#) (page 28) in *Appendix A - examples*).

It is assumed that the datasets will be allocated in the job, in which case the rest of this section can be skipped. If you choose to allocate the datasets 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 `<mxgsaspx>.MXG.SAS.CNTL(BUILDPDB)` to `<mxgwpsspfx>.MXG.WPS.CNTL(BUILDPDB)`. Now edit `<mxgwpsspfx>.MXG.WPS.CNTL(BUILDPDB)`, change:

- `JOB` as appropriate to distinguish MXG+SAS software output from MXG+WPS output
- `PROCLIB` to point to the `<wsspfx>` (`WSPROC`)
- `EXEC` statement to call `WSPROC`
- `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)

### 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 SMF Data

Edit the `WPS+MXG` JCL, so that the the dummied-out `SMFINPUT DD` points to a file containing SMF data. For example, 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 output from `PROC CONTENTS` should now show all datasets of the daily PDB with 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 directly to SAS `DASD` data libraries. 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 contains daily `PDB` and `SPIN` files 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 **before** you run the daily `BUILDPDB` process in MXG+WPS software, as this generates a new `SPIN` file. The previous `SPIN` file will therefore constitute *yesterday's* `SPIN` file, created by the previous run of `BUILDPDB`.

---

The daily `PDB` file only needs to be migrated for comparison purposes. That is, once the MXG+WPS software daily `BUILDPDB` has been run successfully, you need to check that the resulting `PDB` is the same as that produced by the MXG+SAS software.

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

The standard data migration procedure, as described in [Overview of data migration](#) (page 8), can be used to transfer the data from the MXG+SAS software environment to MXG+WPS software.

Assuming that the above 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, you can now migrate the MXG data environment used with MXG+SAS software 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

You need to be able to compare the `PDB` produced by MXG+WPS software against the `PDB` produced by MXG+SAS software. 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

The previous chapters provided details on how to install the the MXG+WPS software, migrate the `SPIN` file, and run the MXG+SAS daily `BUILDPDB` so that the `PDB` file can be compared with the `PDB` file created with the MXG+WPS daily `BUILDPDB`. You should now:

1. Run the MXG+WPS `BUILDPDB` process
2. Compare the resulting MXG+WPS `PDB` file with the MXG+SAS `PDB` file

These tasks are described in the following sections.

## 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 check that the output in `SASLIST` makes sense.

## Compare Resulting MXG+WPS and MXG+SAS PDBs

There are now two daily `PDB` files: one generated by the MXG+WPS software environment, and one generated by the MXG+SAS software environment. You should now compare the content of the two daily `PDB` libraries for consistency/equality.

See *Appendix A - examples* for example listings of the `XMIGRATE` macro (described in the section *Example of WPS migration macros* [↗](#) (page 22)) and the `@COMPARE` job (described in the section *Example of JCL for comparing migration output* [↗](#) (page 26)).

Copy the `@COMPARE` member of the `WPS CNTL` library, and then edit the copy to change:

- `<wpspfx>` to the WPS installation dataset prefix.
- `<sas-dasd-data-library>` to the SAS `DASD PDB` data library.
- `<wps-dasd-data-library>` to the WPS `DASD PDB` data library.
- `<sas-formats-library>` to the MXG `FORMATs` library if necessary.
- `<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 comparison.

---

**Note:**

If you are using the compare macro (`@COMPARE`) for MXG `PDB` comparison, you should set the `MXG` macro variable to `YES`. This causes the compare to ignore the `ZDATE` and `ZTIME` variables, which always contain different values as these are the timestamps of the run time of the `BUILDPDB` process.

---

# Migrating larger MXG installations

MXG installations vary in size and complexity. The previous steps have demonstrated how a site with a simple, single LPAR might 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 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 might 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 prove useful.

### 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. Contact World Programming for help.

In certain circumstances, the layout of the printed report output layout might differ slightly from SAS to WPS. In some cases this might require minor adjustments to 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 compressed. Both products use run-length encoding to compress 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 better to activate the `CICSIFUE` exit supplied in the MXG `SOURCLIB` as member `EXITCICS`. To do this:



1. You must have a load library that contains a program called CICSIFUE that forms part of the SOURCLIB concatenation.
2. Affected programs must have %LET SMFEXIT=CICS; as the first statement in the program source.

Review the instructions in this member to inform your decisions about how you will implement the exit at your site.

You must change the parameter string passed to the LKED step because of current WPS limitations. Change the code:

```
//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 do not make 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.

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 given in the following table:

Is the exit present and active?	Does the input file contain 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 identifies that compressed data is to be processed and will do so using the so-called <i>dynamic</i> decompression exit. An

Is the exit present and active?	Does the input file contain compressed CICS and/or DB2 data?	Result
		<p>accompanying MXGNOTE in the SASLOG will confirm the situation. The text of this note (as of MXG V32.09) is:</p> <pre> ERROR: YOUR &lt;CICS/DB2&gt; RECORDS ARE COMPRESSED, BUT YOU ARE NOT USING THE CICSIFUE "INFILE EXIT" TO DECOMPRESS YOUR &lt;CICS/DB2&gt; 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.           </pre>
Yes	No	<p>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 a message in the SYSLOG:</p> <pre> +CICSDB2: CICSIFUE EXIT VERSION 3 FOR CICS/DB2 DATA           </pre>
Yes	Yes	<p>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:</p> <pre> +CICSDB2: CICSIFUE EXIT VERSION 3 FOR CICS/DB2 DATA           </pre>

# Appendix A - examples

## Example of JCL for simple migration to WPS

This example code is 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 of JCL for migration to WPS via tape

This example code is provided in the <wpspfx.>CNTL library as member @SAS2WFX.

```
// <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 SASTL IS THE DSNAME OF THE SAS DATA-LIBRARY ON TAPE
//      SET SASTL=<sas-tape-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 <sas-tape-data-library> TO THE SAS TAPE DATASET
/* (5)  CHANGE <wps-dasd-data-library> TO THE WPS TARGET DATASET
/* (6)  CHANGE <sas-procedure-name> TO THE SAS PROCEDURE NAME
/* (7)  CHANGE <wps-procedure-name> TO THE WPS PROCEDURE NAME
/* (8)  SUBMIT THIS JOB AND THEN CHECK THE OUTPUT
/* (9)  CHECK FOR A JOB RETURN CODE OF ZERO
/*
/*-----*/
/*
/*=====
/* EXPORT A SAS DATA LIBRARY FROM DASD TO TAPE (NOT COMPRESSED)
/*=====
//SAS      EXEC <sas-procedure-name>
//SOURCLIB DD DISP=SHR,DSN=<wpspfx>.CNTL
//SASDL   DD DISP=SHR,DSN=&SASDL
//SASTL   DD DSN=&SASTL,DISP=(,CATLG,DELETE),
//        UNIT=(CART,,DEFER),VOL=(,,10),LABEL=EXPDT=98007
//
//SYSIN   DD DATA,DLM='++'

*OPTIONS SOURCE SOURCE2 MPRINT MACROGEN MLOGIC;
%INCLUDE SOURCLIB(XMIGRATE);
%LET VERBOSE=YES;
```

```
%SD2ST (SASDL=SASDL , SASTL=SASTL) ;

RUN;

++ END OF //SYSIN
/**
/**=====
/** IMPORT A SAS DATA LIBRARY FROM TAPE INTO A WPS DATA LIBRARY ON DASD
/**=====
/**
/**WPS      EXEC <wps-procedure-name>
/**SOURCLIB DD DISP=SHR,DSN=<wpspfx>.CNTL
/**SASTL    DD DISP=SHR,DSN=&SASTL,
/**          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;
%INCLUDE SOURCLIB(XMIGRATE);
%LET VERBOSE=YES;
%ST2WD(SASTL=SASTL,WPSDL=WPSDL);
RUN;
++ END OF //SYSIN
```

## Example of JCL for format migration to WPS

This example code is provided in the <wpspfx.>CNTL library as member @FMT2WPS.

```
// <add a jobcard here>
//PROCLIB JCLLIB ORDER=(<wpspfx>.CNTL)
/**
/** SYMBOL SASFL IS THE DSNAME OF THE SAS FORMAT LIBRARY
/**      SET SASFL=<sas-dasd-fmt-library>
/**
/** SYMBOL WPSFL IS THE DSNAME OF THE WPS FORMAT LIBRARY
/**      SET WPSFL=<wps-dasd-fmt-library>
/**
/** NOTE : USE REGION=0M TO OBTAIN MAXIMUM AVAILABLE MEMORY
/**
/**-----*/
/** SAMPLE JOB TO MIGRATE A SAS FORMAT LIBRARY TO WPS      */
/**-----*/
/**
/** (1)  ADD A SUITABLE JOBCARD
/** (2)  CHANGE <wpspfx> TO THE WPS INSTALLATION DATASET PREFIX
/** (3)  CHANGE <sas-dasd-fmt-library> TO THE SAS SOURCE FORMAT LIBRARY
/** (5)  CHANGE <wps-dasd-fmt-library> TO THE WPS TARGET FORMAT LIBRARY
/** (6)  CHANGE <sas-procedure-name> TO THE SAS PROCEDURE NAME
/** (7)  CHANGE <wps-procedure-name> TO THE WPS PROCEDURE NAME
/** (8)  SUBMIT THIS JOB AND THEN CHECK THE OUTPUT
/** (9)  CHECK FOR A JOB RETURN CODE OF ZERO
/**
```

```

/*-----*/
/*
/*=====
/* EXPORT A SAS FORMAT LIBRARY TO A TEMPORARY FILE
/*=====
//SAS      EXEC <sas-procedure-name>
//SOURCLIB DD DISP=SHR,DSN=<wpspfx>.CNTL
//SASFL   DD DISP=SHR,DSN=&SASFL
//SASDL   DD DSN=&&TEMPFILE,DISP=(NEW,PASS),
//          UNIT=SYSDA,SPACE=(TRK,(9000,900),RLSE)
//
//*
//SYSIN   DD DATA,DLM='++'

*OPTIONS SOURCE SOURCE2 MPRINT MACROGEN MLOGIC;
%INCLUDE SOURCLIB(XMIGRATE);

%LET VERBOSE=YES;
%SF2SD(SASFL=SASFL,SASDL=SASDL);

RUN;

++ END OF //SYSIN
/*
/*=====
/* CREATE A WPS FORMAT LIBRARY FROM THE TEMPORARY FILE
/*=====
/*
//WPS      EXEC <wps-procedure-name>
//SOURCLIB DD DISP=SHR,DSN=<wpspfx>.CNTL
//SASDL   DD DISP=SHR,DSN=&&TEMPFILE
//WPSFL   DD DISP=(NEW,CATLG),DSN=&WPSFL,
//          UNIT=SYSDA,SPACE=(TRK,(9000,900),RLSE)
//
//*
//SYSIN   DD DATA,DLM='++'

*OPTIONS SOURCE SOURCE2 MPRINT MACROGEN MLOGIC;
%INCLUDE SOURCLIB(XMIGRATE);

%LET VERBOSE=YES;

%SD2WF(SASDL=SASDL,WPSFL=WPSFL);

RUN;

++ END OF //SYSIN

```

## Example of WPS migration macros

This example code is 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.SWCOMP_&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.SWCOMP_&MEMNAME; RUN;
%MEND COMPMBR;

%MACRO SF2SD(SASFL=SASFL, SASDL=SASDL);
%*****;
%* SF2TM *;
%* DESCRIPTION: MACRO TO UNLOAD A SAS FORMAT LIBRARY TO A SAS DASD *;
%* LIBRARY MEMBER *;
%* ARGUMENTS: *;
%* SASFL - THE SOURCE SAS FORMAT LIBRARY *;
%* SASDL - THE TEMPORARY SAS DATA LIBRARY *;
%* NOTE1: THIS MUST BE RUN FROM SAS NOT WPS *;
%*****;
%IF "&VERBOSE" EQ "YES" %THEN
%DO;
PROC FORMAT LIBRARY=SASFL FMTLIB;RUN;
%END;
PROC FORMAT LIBRARY=SASFL CNTLOUT=SASDL.FORMATS;RUN;
%MEND SF2SD;

%MACRO SD2WF(SASDL=SASDL, WPSFL=WPSFL);
%*****;
%* SD2WF *;
%* DESCRIPTION: MACRO TO CREATE A WPS FORMAT LIBRARY FROM A SAS *;
%* DATA LIBRARY MEMBER *;
%* ARGUMENTS: *;
%* SASDL - THE SOURCE SAS DATA LIBRARY *;
%* WPSFL - THE TARGET WPS FORMAT LIBRARY *;
%* *;
%* NOTE1: THIS MUST BE RUN FROM WPS NOT SAS *;
%*****;
PROC FORMAT LIBRARY=WPSFL CNTLIN=SASDL.FORMATS;RUN;
%IF "&VERBOSE" EQ "YES" %THEN
%DO;
PROC FORMAT LIBRARY=WPSFL FMTLIB;RUN;
%END;
%MEND SD2WF;

```

# Example of JCL for comparing migration output

This example code is provided in the <wpspfx.>CNTL library as member @COMPARE.

```
// <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 RUN SAS TO WPS COMPARISON */
/**-----*/
/**
/** (1) ADD A SUITABLE JOBCARD
/** (2) CHANGE <wpspfx> TO THE WPS INSTALLATION DATASET PREFIX
/** (3) CHANGE <sas-dasd-data-library> TO THE SAS DASD DATA LIBRARY
/** (4) CHANGE <wps-dasd-data-library> TO THE WPS DASD DATA LIBRARY
/** (5) CHANGE <sas-formats-library> TO THE MXG FORMATS LIBRARY IF
/**      NECESSARY
/** (6) CHANGE <migration-macro-library> TO THE MIGRATION MACRO
/**      LIBRARY
/** (7) SUBMIT THIS JOB
/** (8) CHECK FOR A JOB RETURN CODE OF ZERO
/** (9) CHECK SASLIST FOR COMPARISON OUTPUT
/**
/**-----*/
/** TO COMPARE A SAS TAPE DATA LIBRARY TO WPS DASD, */
/** REPLACE 'LIBNAME SASDL SASDASD;' WITH 'LIBNAME SASDL SASSEQ;' */
/**-----*/
/**
/**
/**@COMPARE EXEC WPSPROC
/**SASDL DD DISP=SHR,DSN=&SASDL
/**WPSDL DD DISP=SHR,DSN=&WPSDL
/**LIBRARY DD DISP=SHR,DSN=<sas-formats-library> <= IF NECESSARY
/**MTSOURCE DD DISP=SHR,DSN=<migration-macro-library>
/**SYSIN DD DATA,DLM='++'

OPTIONS SOURCE SOURCE2 MPRINT MLOGIC MACROGEN;
OPTIONS FMTSEARCH=(LIBRARY.FORMATS WORK.FORMATS);
%INCLUDE MTSOURCE(XMIGRATE);
%LET VERBOSE = NO;
%LET MXG = YES;

LIBNAME SASDL SASDASD;
*LIBNAME SASDL SASSEQ;

%S2WCOMP (SASDL=SASDL,WPSDL=WPSDL);
```

```
RUN;
++ END OF //SYSIN
```

## Example of MXG FORMATS library build

This example code is 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 of JCL for a sample MXG BUILDPDB

This example code is 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 SOURCLIB 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
```

# Legal Notices

Copyright © 2002–2018 World Programming Limited.

All rights reserved. This information is confidential and subject to copyright. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system.

## Trademarks

WPS and World Programming are registered trademarks or trademarks of World Programming Limited in the European Union and other countries. (r) or ® indicates a Community trademark.

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.

All other trademarks are the property of their respective owner.

## General Notices

World Programming Limited is not associated in any way with the SAS Institute.

WPS is not the SAS System.

The phrases "SAS", "SAS language", and "language of SAS" used in this document are used to refer to the computer programming language often referred to in any of these ways.

The phrases "program", "SAS program", and "SAS language program" used in this document are used to refer to programs written in the SAS language. These may also be referred to as "scripts", "SAS scripts", or "SAS language scripts".

The phrases "IML", "IML language", "IML syntax", "Interactive Matrix Language", and "language of IML" used in this document are used to refer to the computer programming language often referred to in any of these ways.

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