



WPS Installation and User Guide for z/OS

*To guide you through installing and using
WPS on the z/OS platform*



Version: 4.0.3

Copyright © 2002-2018 World Programming Limited

www.worldprogramming.com



Contents

Introduction.....	5
About This Guide.....	5
Overview of the Installation and Configuration Process.....	5
Prerequisites.....	7
Planning for the Installation.....	7
z/OS Version.....	7
Language Environment (LE) Settings.....	7
Security Considerations.....	7
The WPS Software.....	9
Required Files.....	9
Distribution Package.....	9
Obtaining the Distribution Package.....	9
About the z/OS Installation File.....	9
Licence Key File.....	10
Obtaining a Licence Key.....	10
Installing the Software.....	11
Allocate z/OS Files for Upload (PC to z/OS).....	11
Upload WPS DLIB. TER from the PC to z/OS.....	12
Process the Installation File.....	12
Unpacking the Terse File.....	12
Submit the @INSTALL Job.....	14
Configuration.....	22
The WPSPROC Batch JCL Procedure.....	22
Applying the WPS Licence Key.....	23
Installation Verification.....	24
Edit the Sample Verification Job.....	25
Submit the Sample Job.....	25
Placement of Data Libraries.....	35
Sequential z/OS Dataset-Based WPS Data Libraries.....	35
VSAM Linear z/OS Dataset-Based WPS Data Libraries.....	36
USS Directory-Based WPS Data Libraries.....	36
WPS Data Files.....	37
Data Sharing between Multiple WPS Users.....	37
The WORK Library.....	38
WPS Configuration Options.....	38



- CONFIG File.....38
- Overriding the Default CONFIG File..... 40
- SAS Language OPTIONS Statement..... 40
- Listing Current SAS Language OPTIONS..... 40
- Configuration Options for sorting data..... 41
- Fine-tuning WPS to Optimise Runtime Performance..... 41

- Running WPS..... 42**
 - Executing Programs written in the language of SAS.....42
 - Files Created During Execution..... 42

- Using WPS.....43**
 - Examples of Programs and the JCL Required to Execute them..... 43
 - Using SAS Data Libraries..... 43
 - Using SAS XPORT Files.....43
 - Using SAS7BDAT Files.....44
 - Using SD2 Files..... 44
 - Using SASTAPE Files.....44
 - Using RDBMS (DB2, SQLServer).....44
 - Using CPORT Files.....44
 - Running Existing Programs written in the language of SAS..... 44
 - Migrating Existing WPS Data.....45
 - Migrating Existing WPS Programs.....45
 - Using MXG.....45
 - Using DB2.....45
 - Using sequential engines on tape devices.....46
 - Using Fonts.....47

- Further Reading..... 48**
 - Available Reference Material..... 48
 - WPS Migration Guide for z/OS.....48

- Appendix A - Running WPS from UNIX Systems Services.....49**
 - Introduction..... 49
 - Install the WPS USS Components..... 49
 - Permission Settings.....49
 - Edit the @INSTUSS JCL..... 50
 - Submit the @INSTUSS Job.....50
 - About the USS z/OS Dataset.....51
 - Launching WPS.....51
 - Making it Easier..... 51
 - Installation Verification.....51

- Appendix B - Running WPS from TSO.....53**



Introduction.....	53
Arguments and default settings for the TSOWPS CLIST.....	53
Before Using the CLIST.....	56
Launching WPS.....	57
Installation Verification.....	58
Appendix C - Sending files to World Programming.....	59
Appendix D - WPS Load Modules and LPA Usage.....	62
Legal Notices.....	63

Introduction

About This Guide

Overview

This document is intended to help guide you through installing WPS (World Programming System) on the z/OS platform. It also has sections on how to use WPS and what to do if you have 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 `< >`.

When referring to z/OS or MVS datasets or files, we generally refer to them as a "z/OS datasets". The word "dataset" on its own is reserved for use with WPS datasets held within a "data library", referring to the proprietary binary dataset storage format used by WPS.

Overview of the Installation and Configuration Process

The installation and Configuration of WPS on z/OS consists of the following sequence of steps:

- Ensure that all pre-requisites are met
- Upload the WPS software to z/OS
- Run the installation jobs
- Apply the WPS licence key
- Configure WPS and the environment
- Migrate existing data and code libraries
- Use WPS to execute programs written in the language of SAS.



A detailed description of these steps is outlined in the following chapters of this guide.

Prerequisites

Planning for the Installation

WPS uses z/OS system services to implement some functionality. Some of these services are a part of the z/OS Unix System Services (USS) and, as such, require the user to have a valid OMVS segment set up. This will avoid errors that may otherwise occur during WPS usage.

The process of installing WPS on the z/OS platform requires the creation of a number of datasets. It may also require the installer to specify z/OS dataset names and (possibly) USS directory and path names for some existing System Software components.

The following information should be gathered and noted before commencing the installation process:

Information Required	Referred to as...
z/OS TSO USERID of the person installing WPS	<userid>
z/OS HLQ for WPS software libraries/datasets	<wpspfx>
WPSHOME USS HFS directory for the WPS software	<wpshome>
Target z/OS dataset name for the WPS Distribution Library	<wpsdlib>

z/OS Version

WPS is supported on z/OS versions 1.9 and above.

Language Environment (LE) Settings

WPS makes use of the z/OS Language Environment (LE). It is important to be aware of your system's LE configuration. This information is only relevant when it comes to running WPS. It does not affect the WPS installation process.

Security Considerations

As part of the installation process as documented in this guide, you may need to alter directory and file settings. Refer to *Permission Settings* [↗](#) (page 49).



You should also consider protection of your resources using IBM's RACF® (Resource Access Control Facility) [↗](#).

The WPS Software

Required Files

To install the WPS software on the z/OS platform, the following components are required:

- WPS Distribution Package (zip file)
- Licence Key File specific to your site

Distribution Package

The WPS software is distributed as a single zip file, which contains:

- The WPS z/OS installation file in the IBM TERSE format, which, when processed, will install both the WPS software and the associated documentation in plain text format. This documentation includes the installation and upgrade notes, and the End User Licence Agreement.

Note:

Once installed, the end user licence agreement is located in the partitioned dataset `<WSPFX>.LICENSE`, with a suffix that is composed of the language shortcut (for example EN, FR and JA) and associated EBCDIC (Extended Binary Coded Decimal Interchange Code) code page name - for example : `<WSPFX>. LICENSE(EN1047)`, `<WSPFX>. LICENSE(FR297)` and `<WSPFX>. LICENSE(JA930)`.

- The associated migration and user documentation under *Available Reference Material* [↗](#) (page 48), in PDF format.

Obtaining the Distribution Package

For more information on how to obtain the WPS distribution package, please contact World Programming via sales@teamwpc.co.uk [↗](#).

About the z/OS Installation File

The installation file containing all components of WPS for the z/OS platform is called `WPSDLIB.<wpsrel>.TER`. This file is in the standard IBM TERSE compressed format.

Licence Key File

As part of the installation process you will need to apply a licence key to the WPS software. Without a valid licence key, you will not be able to run WPS.

On purchase of WPS, a licence key is provided in a file ending in **.wpskey** that is separate from the distribution package. It is supplied in plain text and contains information specific to your site, together with an encrypted password that allows the WPS software to be executed by you. It is applied using the `SETINIT` procedure described in the section on *Applying the WPS Licence Key* [↗](#) (page 23).

Obtaining a Licence Key

If you do not already have a valid licence key file, please contact World Programming via sales@teamwpc.co.uk [↗](#).

Installing the Software

Allocate z/OS Files for Upload (PC to z/OS)

The installation of WPS may require you to pre-allocate a sequential dataset into which the WPS installation file `<wpsrel>.DLIB.TER` can be uploaded.

If necessary, prepare for uploading the WPS installation file by allocating the following sequential dataset using ISPF 3.2 or equivalent:

```
<wpspfx>.DLIB.TER
```

The following attributes should be allocated to this dataset:

```
DCB=(LRECL=1024, BLKSIZE=27648, RECFM=FB, DSORG=PS),
SPACE=(CYL, (500, 10))
```

Once allocated, the attributes for this dataset should be similar to those shown below:

Dataset Information	
Dataset Name : <wpspfx>.DLIB.TER	
General Data	Current Allocation
Management class . . : **None**	Allocated cylinders : 500
Storage class : SC270	Allocated extents . . : 1
Volume serial : SMSP12	
Device type : 3390	Current Utilization
Data class : **None**	Used cylinders : 0
Organization : PS	Used extents : 0
Record format : FB	
Record length : 1024	
Block size : 27648	
1st extent cylinders: 500	
Secondary cylinders : 10	
Dataset name type :	SMS Compressible . . : NO
Creation date : 2015/09/01	Referenced date . . . : ***None***
Expiration date . . . : ***None***	

Upload WPS DLIB.TER from the PC to z/OS

Using suitable file transfer software (such as IND\$FILE, FTP or the 'transfer' file utility option of your TN3270 application), upload the installation file from your PC to the z/OS dataset created in the previous step.

You may need to:

- Set the name of the input file on the PC so that it is similar to `wps-<wpsrel>-zos-s390.dlib.ter`.
- Set the name of the receiving file on the host system so that it is as defined in *Allocate z/OS Files for Upload (PC to z/OS)* [↗](#) (page 11).
- Set the host operating system to `MVS/TSO`.
- Set any Record Format option to `DEFAULT`, i.e. use the existing DCB information from the pre-allocated dataset.
- Turn off any translations (with regard, for example, to CR/LF, Record Truncation, ASCII/EBCDIC), so that a binary file upload is performed.

Process the Installation File

Unpacking the Terse File

Having successfully uploaded WPS DLIB.TER to your z/OS Host System, you will need to run a job to unpack the WPS DLIB from the TERSE file. The resulting PDSE file will require approximately 500 cylinders of 3390 disk space. Example JCL to perform the unpacking is shown below:

```
// <add a jobcard here>
//*
/*-----*/
/* Unpack the the WPS DLIB Terse file
/*-----*/
/*
/* (1) ADD A SUITABLE JOB CARD
/* (2) CHANGE <wpsdlib> TO THE WPS DISTRIBUTION LIBRARY NAME
/* (3) SUBMIT THIS JOB AND THEN CHECK THE OUTPUT
/*
//STEP      EXEC PGM=TRSMAIN,PARM=UNPACK
//SYSPRINT DD SYSOUT=*
//INFILE    DD DISP=SHR,DSN=<wpsdlib>.TER
//OUTFILE   DD DSN=<wpsdlib>,DISP=(NEW,CATLG),
//          SPACE=(CYL,(500,10),RLSE),UNIT=SYSDA,
```



```
//          DSNTYPE=LIBRARY,
//          DCB=(DSORG=PO,LRECL=80,BLKSIZE=27920,RECFM=FB)
```

If the job runs to successful completion, you should see output similar to the following:

```
** AMA572I STARTING TERSE DECODE  UNPACK          08:35:10  09/22/2009
** AMA527I INPUT  - DDNAME : INFILE  DSNAME: <wpspfx>.DLIB.TER
** AMA528I OUTPUT - DDNAME : OUTFILE DSNAME: <wpspfx>.DLIB
** AMA555I THE VALUES ARE:  BLKSIZE= 27920  LRECL=80          PACKTYPE=PACK
REFM=FIXED
** AMA583I INPUT DATASET SIZE IN BYTES: 145809408 OUTPUT DATASET SIZE IN BYTES:
325206964 COMPRESSION RATIO: 44%
** AMA573I TERSE COMPLETE DECODE  UNPACK          08:36:00  09/22/2015
** AMA504I RETURN CODE: 0
```

The resulting <wpspfx>.DLIB is a PDSE which contains members that are associated with installing and executing WPS on the z/OS platform. Those members are described in the \$README member, which is reproduced below:

```
=====
WORLD PROGRAMMING SYSTEM (WPS) DLIB README
=====

NOTES      : INSTALLATION INSTRUCTIONS ARE PROVIDED IN THE WPS
            INSTALLATION AND USER GUIDE. MEMBERS PREFIXED X ARE XMIT
            FORMAT FILES. THESE FILES ARE EXPANDED USING THE MEMBERS
            PREFIXED @ AS DESCRIBED IN THE GUIDE. DO NOT USE THE
            REXX FILES DIRECTLY.

COMPONENT SUMMARY
=====

NOTE       : MEMBERS STARTING WITH $ CONTAIN DOCUMENTATION
            : MEMBERS STARTING WITH @ CONTAIN JCL
            : MEMBERS STARTING WITH R CONTAIN REXX FILES
            : MEMBERS STARTING WITH X CONTAIN XMIT FILES

$README    : THIS MEMBER
$VERSION   : THE VERSION NUMBER OF THIS WPS INSTALLATION
@INSTALL   : JCL FOR FIRST STAGE WPS INSTALL PROCESS
@INSTSDK   : JCL FOR OPTIONAL INSTALL OF WPSSDK COMPONENTS
@INSTUSS   : JCL FOR OPTIONAL INSTALL WPS HOME DIRECTORY TO USS
RINST      : REXX FOR FIRST STAGE WPS INSTALL PROCESS
RINSTSDK   : REXX FOR OPTIONAL INSTALL OF WPS SDK COMPONENTS
RINSTUSS   : REXX FOR OPTIONAL INSTALL OF WPS HOME DIRECTORY TO USS
XAUTOLIB   : SAS AUTOCALL LIBRARY IN XMIT FORMAT
XCNTL      : JCL AND SOURCE COMPONENTS IN XMIT FORMAT
XFONT      : TRUETYPE FONTS FOR USE WITH WPS GRAPHING IN XMIT FORMAT
XLICENSE   : END-USER LICENSE AGREEMENT (EULA) IN ENGLISH, FRENCH
            AND JAPANESE
XLOAD      : LOAD MODULE COMPONENTS IN XMIT FORMAT
XSASHELP   : WPS SASHELP VIEW COMPONENTS IN XMIT FORMAT
XSDKCNT    : C CNTL LIBRARY FOR WPS SDK IN XMIT FORMAT
XSDKCH     : C HEADER FILES FOR THE WPS SDK IN XMIT FORMAT
XSDCSRC    : SOURCE CODE FOR SAMPLE C WPS SDK IN XMIT FORMAT
XSDKOSLB   : OS LINK OBJECT LIBRARY FOR WPS SDK IN XMIT FORMAT
XSDKSCNT   : ASSEMBLER CNTL LIBRARY FOR WPS SDK IN XMIT FORMAT
XSDKSMAC   : ASSEMBLER MACROS AND COPY FILES FOR WPS SDK IN XMIT FORMAT
XSDKSSRC   : SOURCE CODE FOR SAMPLE ASSEMBLER WPS SDK IN XMIT FORMAT
XSDKXPLB   : XPLINK OBJECT LIBRARY FOR WPS SDK IN XMIT FORMAT
```



```

XUSS          : OPTIONAL USS COMPONENTS IN XMIT AND PAX FORMAT

@INSTALL JOB WILL CREATE THE FOLLOWING LIBRARIES:
<WSPSPFX>.AUTOLIB : SAS AUTOCALL LIBRARY
<WSPSPFX>.CLIST   : CLIST
<WSPSPFX>.CNTL   : JCL, PROCS AND WPS SOURCE CODE
<WSPSPFX>.FONTS  : ODS FONTS
<WSPSPFX>.LICENSE : END-USER LICENSE AGREEMENT(EULA) IN ENGLISH,
                   FRENCH AND JAPANESE
<WSPSPFX>.LOAD   : LOAD MODULES
<WSPSPFX>.SASHELP : VIEW MODULES
<WSPSPFX>.SETINIT : HOLDS COMPILED SETINIT
<WSPSPFX>.USS    : COMPONENTS FOR USS ENVIRONMENT

@INSTUSS JOB WILL CREATE/UPDATE:
<WPSHOME>      CONTAINS WPS USS COMPONENTS

@INSTSDK JOB WILL CREATE THE FOLLOWING LIBRARIES:

<WSPSPFX>.SDK.C.H           CONTAINS THE C HEADER FILES FOR THE WPS SDK
<WSPSPFX>.SDK.C.SRC        CONTAINS SOURCE CODE FOR SAMPLE C WPS SDK
                           MODULES
<WSPSPFX>.SDK.C.CNTL       CONTAINS JCL FOR COMPILING THE SAMPLE C WPS SDK
                           MODULES
<WSPSPFX>.SDK.XPLINK.LIB   OBJECT LIBRARY FOR WPS SDK MODULES COMPILED
                           USING XPLINK
<WSPSPFX>.SDK.CLINK.LIB   OBJECT LIBRARY FOR WPS SDK MODULES COMPILED
                           NON-XPLINK
<WSPSPFX>.SDK.ASM.MACLIB  CONTAINS ASSEMBLER MACROS AND COPY FILES FOR
                           WRITING WPS SDK MODULES IN ASSEMBLER.

```

The extraction of these datasets is performed by executing the job contained in the @INSTALL member of the PDSE <wpspfx>.DLIB.

Submit the @INSTALL Job

The @INSTALL JCL requires editing before submission. It can be edited using ISPF Option 2 or an equivalent editor. Follow the instructions at the top of the file, substituting appropriate values where stated. Before editing, the file should look similar to the following:

```

// <add a jobcard here>
// *
// *-----*/
// * @INSTALL : INSTALL THE WPS DISTRIBUTION LIBRARIES
// *-----*/
// *
// * (1) ADD A SUITABLE JOB CARD
// * (2) CHANGE <wpsdlib> TO THE WPS DISTRIBUTION LIBRARY NAME
// * (3) CHANGE <wpspfx> IN ALL PLACES BELOW TO THE D/S PREFIX
// *     FOR WPS INSTALL LIBRARIES
// *     OR ..
// * (3) CHANGE:
// *     *AUTOLIB, *CNTL, *CLIST, *FONTS, *LOAD, *SASHELP AND *USS
// *     TO:
// *     @AUTOLIB, @CNTL, *CLIST, @FONTS, @LOAD, @SASHELP AND @USS

```



```

//*      IF THESE HAVE BEEN PREALLOCATED
//* (4) SUBMIT THIS JOB AND THEN CHECK THE OUTPUT
//*
//*
//STEP01 EXEC PGM=IEFBR14
//SETINIT DD DISP=(NEW,CATLG),DSN=<wpspfx>.SETINIT,
//        DSORG=PS,RECFM=FS,LRECL=27998,BLKSIZE=27998,
//        SPACE=(TRK,1),UNIT=SYSDA
//*
//STEP02 EXEC PGM=IKJEFT1B,DYNAMNBR=999,COND=(0,NE)
//SYSEXEC DD DISP=SHR,DSN=<wpsdlib>
//*AUTOLIB DD DISP=SHR,DSN=<autolib>
//*CNTL DD DISP=SHR,DSN=<cntl>
//*CLIST DD DISP=SHR,DSN=<clist>
//*FONTS DD DISP=SHR,DSN=<fonts>
//*LOAD DD DISP=SHR,DSN=<load>
//*SASHELP DD DISP=SHR,DSN=<sashelp>
//*USS DD DISP=SHR,DSN=<usspax>
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//SYSTSIN DD DATA,DLM='++'
PROF NOPREFIX
RINST 'PFX(<wpspfx>)'
++ END OF //SYSTSIN

```

On submission, the @INSTALL job will produce six outputs with the same job ID. Check that the return code is zero for all six outputs.

This job will create the following datasets:

- <wpspfx>.AUTOLIB - A PDSE containing some AUTOCALL macros that may prove useful. Also a potential location for similar site-specific macros.
- <wpspfx>.CLIST - A PDSE containing the TSOWPS CLIST, for using WPS under TSO
- <wpspfx>.CNTL - A PDSE containing example JCL, procedures and source members
- <wpspfx>.FONTS - A PDSE providing a location for TrueType fonts
- <wpspfx>.LOAD - A PDSE containing all the WPS program load modules
- <wpspfx>.SASHELP - A WPS-format data library containing various required support items
- <wpspfx>.SETINIT - A sequential dataset, 'flat' file, that will contain encoded licence key information.
- <wpspfx>.USS - A sequential dataset that contains optional WPS components to be copied to the USS environment if it is required to run WPS from USS.

You should see results similar to the following:

Output #1

```

                                     IEBCOPY MESSAGES AND CONTROL STATEMENTS
                                     PAGE      1
IEB1135I IEBCOPY  FMID HDZ1D10  SERVICE LEVEL UA61306  DATED 20110713 DFSMS 01.13.00
z/OS    01.13.00 HBB7780  CPU 1090
IEB1035I <userid>  STEP02   14:27:20 TUE 25 AUG 2015 PARM=''
COPY INDD=( (SYS00007,R) ),OUTDD=SYS00005

```



```
IEB1013I COPYING FROM PDSU INDD=SYS00007 VOL=WPTMP2
DSN=SYS13302.T142720.RA000.<userid>.R0140300
IEB1014I          TO PDSE OUTDD=SYS00005 VOL=ZDSYS1 DSN=WPS.V310.B29291.AUTOLIB
IGW01552I MEMBER DS2CSV HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER SYSRC HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER VERIFY HAS BEEN LOADED AND REPLACED
IGW01550I 3 OF 3 MEMBERS WERE LOADED
IEB147I END OF JOB - 0 WAS HIGHEST SEVERITY CODE
```

Output #2

```

                                     IEBCOPY MESSAGES AND CONTROL STATEMENTS
                                     PAGE          1
IEB1135I IEBCOPY  FMID HDZ1D10  SERVICE LEVEL UA61306  DATED 20110713 DFSMS 01.13.00
z/OS      01.13.00 HBB7780  CPU 1090
IEB1035I <userid>  STEP02    14:27:21 TUE 25 AUG 2015 PARM=''
COPY INDD=((SYS00015,R)),OUTDD=SYS00013
IEB1013I COPYING FROM PDSU INDD=SYS00015 VOL=WPTMP2
DSN=SYS13302.T142720.RA000.<userid>.R0140304
IEB1014I          TO PDSE OUTDD=SYS00013 VOL=ZDSYS1 DSN=WPS.V310.B29291.CNTL
IGW01552I MEMBER $README HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER @COMPARE HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER @LIVE HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER @LOADMGR HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER @MXGFMTS HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER @MXGPDB HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER @SAS2WPS HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER @SAS2WPX HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER @SETINIT HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER @SPAWNER HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER @VERIFY HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER @XSEQFB HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER @XSEQVB HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER CEEOPTS HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER CONFIG HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER DFSPARM HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER NEWS HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER SETINIT HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSAOINI HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSCLIST HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSPROC HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER XAPPSVR HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER XMIGRATE HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER XSEQFB HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER XSEQVB HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER XVERIFY HAS BEEN LOADED AND REPLACED
IGW01550I 27 OF 27 MEMBERS WERE LOADED
IEB147I END OF JOB - 0 WAS HIGHEST SEVERITY CODE
```

Output #3

```

                                     IEBCOPY MESSAGES AND CONTROL STATEMENTS
                                     PAGE          1
IEB1135I IEBCOPY  FMID HDZ1D10  SERVICE LEVEL UA61306  DATED 20110713 DFSMS 01.13.00
z/OS      01.13.00 HBB7780  CPU 1090
IEB1035I <userid>  STEP02    14:27:22 TUE 25 AUG 2015 PARM=''
COPY INDD=((SYS00023,R)),OUTDD=SYS00021
IEB1013I COPYING FROM PDSU INDD=SYS00023 VOL=WPTMP1
DSN=SYS13302.T142722.RA000.<userid>.R0140308
IEB1014I          TO PDSE OUTDD=SYS00021 VOL=ZDSYS1 DSN=WPS.V310.B29291.CLIST
```




```
IGW01552I MEMBER TSOWPS HAS BEEN LOADED AND REPLACED
IGW01550I 1 OF 1 MEMBERS WERE LOADED
IEB147I END OF JOB - 0 WAS HIGHEST SEVERITY CODE
```

Output #4

```

                                IEBCOPY MESSAGES AND CONTROL STATEMENTS
                                PAGE          1
IEB1135I IEBCOPY  FMID HDZ1D10  SERVICE LEVEL UA61306  DATED 20110713 DFSMS 01.13.00
z/OS      01.13.00 HBB7780  CPU 1090
IEB1035I <userid>  STEP02   14:27:23 TUE 25 AUG 2015 PARM=''
COPY INDD=((SYS00031,R)),OUTDD=SYS00029
IEB1013I COPYING FROM PDSU  INDD=SYS00031 VOL=WPTMP1
DSN=SYS13302.T142722.RA000.<userid>.R0140312
IEB1014I          TO PDSE OUTDD=SYS00029 VOL=ZDSYS1 DSN=WPS.V310.B29291.FONTS
IGW01552I MEMBER VERA HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER VERABD HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER VERABI HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER VERAIT HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER VERAMOBH HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER VERAMOBK HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER VERAMOBM HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER VERAMOBP HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER VERAMOBQ HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER VERAMOBV HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER VERAMOBW HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER VERAMOBX HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER VERAMOBZ HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER VERAMONH HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER VERAMONK HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER VERAMONM HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER VERAMONP HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER VERAMONQ HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER VERAMONV HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER VERAMONW HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER VERAMONX HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER VERAMONZ HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER VERASE HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER VERASEBD HAS BEEN LOADED AND REPLACED
IGW01550I 10 OF 10 MEMBERS WERE LOADED
IEB147I END OF JOB - 0 WAS HIGHEST SEVERITY CODE
```

Output #5

```

                                IEBCOPY MESSAGES AND CONTROL STATEMENTS
                                PAGE          1
IEB1135I IEBCOPY  FMID HDZ1D10  SERVICE LEVEL UA61306  DATED 20110713 DFSMS 01.13.00
z/OS      01.13.00 HBB7780  CPU 1090
IEB1035I <userid>  STEP02   14:27:47 TUE 25 AUG 2015 PARM=''
COPY INDD=((SYS00039,R)),OUTDD=SYS00037
IEB1013I COPYING FROM PDSU  INDD=SYS00039 VOL=WPTMP1
DSN=SYS13302.T142723.RA000.<userid>.R0140316
IEB1014I          TO PDSE OUTDD=SYS00037 VOL=ZDSYS1 DSN=WPS.V310.B29291.LOAD
IGW01552I MEMBER DLIIFUE HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER IMSSUBT HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER IXMI51DA HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER IXMI51IN HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER IXMI51UC HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSACATA HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSACCEP HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSADISK HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSADUMM HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSAEMAI HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSAFTP HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSAHTTP HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSANOVA HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSAPIPE HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSAPPSR HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSARIMA HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSASOCK HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSATEMP HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSBASEP HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSCATAL HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSCHART HAS BEEN LOADED AND REPLACED
```



```
IGW01552I MEMBER WPSCIMPO HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSCLUST HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSCOMPA HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSCORR HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSCPORT HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSDATAS HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSDBF HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSDB2EX HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSDISCR HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSDISTA HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSDOWNL HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSEDB2 HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSESAS HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSESPSS HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSETD HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSEWPS HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSEXPAN HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSEXPOR HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSEXPRT HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSFACTO HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSFASTC HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSFOREC HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSFORMA HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSFORMS HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSFREQ HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSGCHAR HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSGDEVI HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSGFONT HAS BEEN LOADED AND REPLACED
```

IEBCOPY MESSAGES AND CONTROL STATEMENTS

PAGE 2

```
IGW01552I MEMBER WPSGLM HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSGOPTI HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSGPLOT HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSGREPL HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSHOST HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSHTTP HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSIMPOR HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSJAVAI HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSKDE HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSLOGIS HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSMEANS HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSODSTE HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSOHTML HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSOLIST HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSOMARK HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSOOUTP HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSOPTIO HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSOPTLO HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSOPTSA HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSPDS HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSPDSCO HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSPLOT HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSPRINC HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSPRINT HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSPRTTO HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSPWENC HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSRANK HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSREG HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSRELEA HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSREPOR HAS BEEN LOADED AND REPLACED
```



```

IGW01552I MEMBER WPSSCORE HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSSETIN HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSSGREN HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSSOAP HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSSORT HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSSOURC HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSSQLAC HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSRSLT HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSSTAND HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSSTDIZ HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSSTEPD HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSTABUL HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSTEMPL HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSTRANS HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSTRANT HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSTREE HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSTTEST HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSUBITS HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSUCHAR HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSUCOMB HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSUDBCS HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSUDSIO HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSUDTMM HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSUEXTF HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSUEXTR HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSUFNCL HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSUINET HAS BEEN LOADED AND REPLACED

```

IEBCOPY MESSAGES AND CONTROL STATEMENTS

PAGE 3

```

IGW01552I MEMBER WPSUISPF HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSUMACR HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSUMATH HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSUNIVA HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSUNLS HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSUPLOA HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSUPROB HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSUPRX HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSURAND HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSUSPEC HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSUSTAT HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSUTRNC HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSUVINF HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSUWEB HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSUZIP HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSVARCL HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSXCHRT HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSXCOMB HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSXCOMM HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSXCONB HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSXCORE HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSXDSTP HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSXEXEC HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSXGLBL HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSXGRPH HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSXICUC HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSXJAVA HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSXNET HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSXODS HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSXOGRA HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSXPARS HAS BEEN LOADED AND REPLACED

```



```

IGW01552I MEMBER WPSXPROC HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSXREGI HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSXSDK HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSXSDKD HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSXSORT HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSXSUMM HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSXUTIL HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSXVER HAS BEEN LOADED AND REPLACED
IGW01552I MEMBER WPSX12 HAS BEEN LOADED AND REPLACED
IGW01550I 146 OF 146 MEMBERS WERE LOADED
IEB147I END OF JOB - 0 WAS HIGHEST SEVERITY CODE

```

Output #6

```

READY
PROF NOPREFIX
READY
RINST 'PFX(<wpspfx>)'
/*=====*/
/* WPS Z/OS INSTALL PROCESS : STARTED */
/*=====*/

>>> USING '<wpspfx>.DLIB' AS RECEIVE SOURCE
>>> USING '<wpspfx>.AUTOLIB' AS RECEIVE TARGET FOR XAUTOLIB, PRE-ALLOC(N)
>>> USING '<wpspfx>.CNTL' AS RECEIVE TARGET FOR XCNTL, PRE-ALLOC(N)
>>> USING '<wpspfx>.CLIST' AS RECEIVE TARGET FOR XCLIST, PRE-ALLOC(N)
>>> USING '<wpspfx>.FONTS' AS RECEIVE TARGET FOR XFONTS, PRE-ALLOC(N)
>>> USING '<wpspfx>.LOAD' AS RECEIVE TARGET FOR XLOAD, PRE-ALLOC(N)
>>> USING '<wpspfx>.SASHELP' AS RECEIVE TARGET FOR XSASHELP, PRE-ALLOC(N)
>>> USING '<wpspfx>.USS' AS RECEIVE TARGET FOR XUSS, PRE-ALLOC(N)

/*=====*/
/* RECEIVING USS COMPONENTS */
/*=====*/

INMR901I Dataset WPCUK02.WPS.V32.USS.PAX from BUILD on N1
INMR906A Enter restore parameters or 'DELETE' or 'END' +
INMR908A The input file attributes are: DSORG=SEQUENTIAL, RECFM=U, BLKSIZE=27998,
LRECL=0, File size=4430K bytes +
INMR909A You may enter DSNAME, SPACE, UNIT, VOL, OLD/NEW, or RESTORE/COPY/DELETE/END
INMR001I Restore successful to dataset '<wpspfx>.USS'

/*=====*/
/* RECEIVING AUTOLIB COMPONENTS */
/*=====*/

INMR901I Dataset WPCUK02.WPS.V32.AUTOLIB from BUILD on N1
INMR154I The incoming data set is a 'DATA LIBRARY'.
INMR906A Enter restore parameters or 'DELETE' or 'END' +
INMR908A The input file attributes are: DSORG=PARTITIONED, RECFM=FB, BLKSIZE=3120,
LRECL=80, File size=58K bytes +
INMR909A You may enter DSNAME, SPACE, UNIT, VOL, OLD/NEW, or RESTORE/COPY/DELETE/END
INMR001I Restore successful to dataset '<wpspfx>.AUTOLIB'

/*=====*/
/* RECEIVING CNTL COMPONENTS */
/*=====*/

INMR901I Dataset WPCUK02.WPS.V32.CNTL from BUILD on N1
INMR154I The incoming data set is a 'DATA LIBRARY'.

```



```
INMR906A Enter restore parameters or 'DELETE' or 'END' +
INMR908A The input file attributes are: DSORG=PARTITIONED, RECFM=FB, BLKSIZE=3120,
  LRECL=80, File size=359K bytes +
INMR909A You may enter DSNAME, SPACE, UNIT, VOL, OLD/NEW, or RESTORE/COPY/DELETE/END
INMR001I Restore successful to dataset '<wpspf>.CNTL'

/*=====*/
/* RECEIVING CLIST COMPONENTS */
/*=====*/

INMR901I Dataset WPCUK02.WPS.V32.CLIST from BUILD on N1
INMR154I The incoming data set is a 'DATA LIBRARY'.
INMR906A Enter restore parameters or 'DELETE' or 'END' +
INMR908A The input file attributes are: DSORG=PARTITIONED, RECFM=FB, BLKSIZE=3120,
  LRECL=80, File size=120K bytes +
INMR909A You may enter DSNAME, SPACE, UNIT, VOL, OLD/NEW, or RESTORE/COPY/DELETE/END
INMR001I Restore successful to dataset '<wpspf>.CLIST'

/*=====*/
/* RECEIVING FONTS COMPONENTS */
/*=====*/

INMR901I Dataset WPCUK02.WPS.V32.FONTS from BUILD on N1
INMR154I The incoming data set is a 'DATA LIBRARY'.
INMR906A Enter restore parameters or 'DELETE' or 'END' +
INMR908A The input file attributes are: DSORG=PARTITIONED, RECFM=U, BLKSIZE=27998,
  LRECL=0, File size=791K bytes +
INMR909A You may enter DSNAME, SPACE, UNIT, VOL, OLD/NEW, or RESTORE/COPY/DELETE/END
INMR001I Restore successful to dataset '<wpspf>.FONTS'

/*=====*/
/* RECEIVING LOAD COMPONENTS */
/*=====*/

INMR901I Dataset WPCUK02.WPS.V32.LOAD from BUILD on N1
INMR154I The incoming data set is a 'PROGRAM LIBRARY'.
INMR906A Enter restore parameters or 'DELETE' or 'END' +
INMR908A The input file attributes are: DSORG=PARTITIONED, RECFM=U, BLKSIZE=27998,
  LRECL=0, File size=X'0005133B'K bytes +
INMR909A You may enter DSNAME, SPACE, UNIT, VOL, OLD/NEW, or RESTORE/COPY/DELETE/END
INMR001I Restore successful to dataset '<wpspf>.LOAD'

/*=====*/
/* RECEIVING SASHELP COMPONENTS */
/*=====*/

INMR901I Dataset WPCUK02.WPS.V32.SASHELP from BUILD on N1
INMR906A Enter restore parameters or 'DELETE' or 'END' +
INMR908A The input file attributes are: DSORG=SEQUENTIAL, RECFM=F S, BLKSIZE=6144,
  LRECL=6144, File size=37393K bytes +
INMR909A You may enter DSNAME, SPACE, UNIT, VOL, OLD/NEW, or RESTORE/COPY/DELETE/END
INMR001I Restore successful to dataset '<wpspf>.SASHELP'

/*=====*/
/* WPS Z/OS INSTALL PROCESS : COMPLETED */
/*=====*/

READY
END
```

Configuration

The WPSPROC Batch JCL Procedure

An example JCL procedure for invoking WPS in a z/OS batch job is provided in the WPSPROC member of the PDS <wpspfx>.CNTL. The example needs to be modified if it is to be used. You can edit WPSPROC using ISPF Option 2 or an equivalent editor. Before editing, it should look similar to the following:

```

/*-----*/
/* WPSPROC : BATCH INTERFACE TO THE WORLD PROGRAMMING SYSTEM (WPS) */
/*-----*/
/*
/*
/* (1) CHANGE <wpspfx> BELOW TO THE WPS INSTALLATION DATASET PREFIX */
/*
/*-----*/
/*
/*-----*/
/* DEFINE WPSPROC AND DEFAULT ARGUMENTS
/*-----*/
//WPSPROC PROC WPSPFX='<wpspfx>',          /* WPS DATASET PREFIX */
// CONFIG=NULLFILE,                       /* USER CONFIG FILE */
// LOAD='*.NULLLOAD,VOL=REF=*.NULLLOAD',   /* DUMMY LOAD CONCAT */
// OPTIONS=' ',                            /* WPS OPTIONS */
// SASAUTO='*.NULLAUTO,VOL=REF=*.NULLAUTO', /* DUMMY SASAUTOS CONCAT*/
// SYSPARM=' ',                            /* PROGRAM PARAMETERS */
// WORKDSN='&&WPSWORK',                    /* WORK DATASET NAME */
// WORKUNI=TRK,WORKPRI=450,WORKSEC=450     /* DEFAULT WORK SPACE */
/*
/* EXECUTE WPSHOST
//WPS EXEC PGM=WPSHOST,REGION=0M,
// PARM=('SYSPARM='&SYSPARM'&OPTIONS')
/*
/* DEFINE NULL DDNAMES
//NULLLOAD DD DISP=(MOD,PASS),DSN=&&MTLOAD,UNIT=SYSDA,
// SPACE=(TRK,(1,1,1)),LIKE=&WPSPFX..LOAD
//NULLAUTO DD DISP=(MOD,PASS),DSN=&&MTAUTO,UNIT=SYSDA,
// SPACE=(TRK,(1,1,1)),LIKE=&WPSPFX..AUTOLIB
/*
/* DEFINE STEPLIB
//STEPLIB DD DISP=(SHR,PASS),DSN=&LOAD
// DD DISP=SHR,DSN=&WPSPFX..LOAD
/*
/* DEFINE WORK DDNAME
//WORK DD DISP=(NEW,DELETE),DSN=&WORKDSN,
// SPACE=(&WORKUNI,(&WORKPRI,&WORKSEC))
/*
/* DEFINE WPS-SPECIFIC DDNAMES
//CONFIG DD DISP=SHR,DSN=&WPSPFX..CNTL(CONFIG)
// DD DISP=SHR,DSN=&CONFIG
//NEWS DD DISP=SHR,DSN=&WPSPFX..CNTL(NEWS)
//SASAUTOS DD DISP=(SHR,PASS),DSN=&SASAUTO

```

```
//          DD DISP=SHR,DSN=&WPSPFX..AUTOLIB
//SASHELP  DD DISP=SHR,DSN=&WPSPFX..SASHELP
//SASLIST  DD SYSOUT=*
//SASLOG   DD SYSOUT=*,RECFM=VBA,LRECL=137,BLKSIZE=141
//SETINIT  DD DISP=SHR,DSN=&WPSPFX..SETINIT
//WPSFONTS DD DISP=SHR,DSN=&WPSPFX..FONTS
//WPSTRACE DD SYSOUT=*
// *
// * DEFINE LANGUAGE ENVIRONMENT (LE) DDNAMES
//CEEDUMP  DD SYSOUT=*
//CEEOPPTS DD DISP=SHR,DSN=&WPSPFX..CNTL(CEEOPPTS)
//CEERPT   DD SYSOUT=*
// *
// * DEFINE SORT DDNAMES
//DFSPARM  DD DISP=SHR,DSN=&WPSPFX..CNTL(DFSPARM)
//SORTMSG  DD SYSOUT=*
// *
// * DEFINE DB2 DDNAMES
//WPSAOINI DD DISP=SHR,DSN=&WPSPFX..CNTL(WPSAOINI)
//DSNAOINI DD DISP=(NEW,DELETE),DSN=&&DSNAOINI,
//          DSORG=PS,RECFM=FB,LRECL=80,
//          SPACE=(TRK,1),UNIT=SYSDA
// *
// * DEFINE SYSPRINT AND SYSOUT
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//          PEND
```

The `WPSPFX` parameter is the only value that must be changed. Further changes may be required to suit site-specific requirements.

Applying the WPS Licence Key

A valid licence key must be applied to activate your installed copy of WPS. This requires the special execution of `PROC SETINIT` using the values contained in your site-specific licence key file (refer to the *Licence Key File* [🔗](#) (page 10) section of this guide).

Note:

New licence keys may be issued from time to time and re-applied using the same process discussed in this section.

For the purpose of applying the licensing information, the following two members of the `<wpspfx>.CNTL` library are used:

- `XSETINIT` contains the actual licensing information;
- `@SETINIT` contains example JCL, referring to the `XSETINIT` member, which can be used to apply the licence.

Before modification, the `@SETINIT` looks similar to the following:

```
// <add a jobcard here>
//PROCLIB  JCLLIB ORDER=(<wpspfx>.CNTL)
```

```

/**
/**-----*/
/** WPS SETINIT JOB
/**-----*/
/**
/** (1) ADD A SUITABLE JOBCARD
/** (2) CHANGE <wpspfx> TO THE WPS INSTALLATION DATASET PREFIX
/** (3) CORRECTLY CONFIGURE <wpspfx>.CNTL(WPSPROC)
/** (4) PLACE THE SETINIT LICENSING CODE, OBTAINED FROM WORLD
/** PROGRAMMING, INTO THE 'XSETINIT' MEMBER OF THIS DATASET
/** (5) SUBMIT THIS JOB AND THEN CHECK THE OUTPUT
/** (6) CHECK FOR A JOB RETURN CODE OF ZERO
/**
/**-----*/
/**
/**@SETINIT EXEC WPSPROC,OPTIONS='SETINIT'
/**SYSIN DD DISP=SHR,DSN=<wpspfx>.CNTL(XSETINIT)

```

To apply the licence and activate your copy of WPS you need to follow these steps:

1. Substitute your licensing information in the XSETINIT member described above. Be sure to include the entire file, from the 'PROC SETINIT;' statement to the final 'RUN;' statement.

Note:

Site-specific security requirements may mandate that the contents of the XSETINIT member are moved to a more secure location. If so, the SYSIN DD statement in the @SETINIT member will need to be changed accordingly.

2. Add a job card.
3. Change the two occurrences of <wpspfx> to the required value.
4. Submit the JCL.

When the licence key has been successfully applied by PROC SETINIT, you will see a message in the SASLOG output, saying:

```
setinit applied successfully.
```

Note:

We recommend that you specify REGION=0M on your jobcard when submitting WPS job steps, in order to obtain the maximum available memory allocation.

Installation Verification

The @VERIFY member of PDS <wpspfx>.CNTL, supplied with WPS, contains a sample job that can be used to verify that the WPS software installation and licensing process has been successfully completed.

Edit the Sample Verification Job

You can edit the JCL in @VERIFY using ISPF Option 2. Before editing, it should look similar to the following:

```
// <add a jobcard here>
//PROCLIB JCLLIB ORDER=(<wpspfx>.CNTL)
//*
/*-----*/
/* SAMPLE JOB TO VERIFY WPS INSTALLATION */
/* BY RUNS THE INSTALLATION VERIFICATION PROGRAM (XVERIFY) */
/*-----*/
/*
/* (1) ADD A SUITABLE JOBCARD
/* (2) CHANGE <wpspfx> TO THE WPS INSTALLATION DATASET PREFIX
/* (3) SUBMIT THIS JOB AND THEN CHECK THE OUTPUT
/* (4) CHECK FOR A JOB RETURN CODE OF ZERO
/*
/*-----*/
/*
/*@VERIFY EXEC WPSPROC
//SOURCLIB DD DISP=SHR,DSN=<wpspfx>.CNTL
//SYSIN DD DATA,DLM='++'
OPTIONS SOURCE2;
ODS LISTING;
%INCLUDE WPSIN;
++ END OF //SYSIN
/*
//WPSIN DD DATA,DLM='++'
%INCLUDE SOURCLIB(XVERIFY);
++ END OF //WPSIN
```

Edit the value of <wpspfx> to be the PREFIX of the WPS installation libraries that were created by the @INSTALL; job, e.g. 'WPS.V2401'.

A suitable jobcard needs to be added.

We recommend that you specify REGION=0M on your jobcard when submitting WPS job steps, in order to obtain the maximum available memory allocation.

Submit the Sample Job

Submit the @VERIFY JCL and check that the job completes with a return code of zero. The output from running the job should be similar to the example shown below:

```
NOTE: This session is executing on the z/OS platform

*****
*
*           World Programming System (WPS) Version 3
*
* Edit the contents of the default NEWS file or specify a different *
```



```

* NEWS file using the NEWS system option to change this message.      *
*                                                                      *
*****
1      OPTIONS SOURCE2;
2      ODS LISTING;
3      %INCLUDE WPSIN;
Start of %INCLUDE(level 1) ADJC.ADJCVERF.JOB03362.D0000102.?
4      + %INCLUDE SOURCLIB(XVERIFY);
Start of %INCLUDE(level 2) SOURCLIB(xverify)
5      + /*****
6      + /*
7      + /* XVERIFY
8      + /* SAS SOURCE LIBRARY CONTAINING INSTALLATION VERIFICATION PROGRAM */
9      + /*
10     + /*****
11     +
12     + PROC OPTIONS; RUN;

```

Portable Options:

<code>_LAST_ =</code>	The last data set created by a DATA step or PROC
<code>NOAUTOSIGNON</code>	Remote submit will not attempt to automatically signon
<code>BASEENGINE=WPD</code>	The library engine to use when BASE is specified
<code>BUFNO=1</code>	Specifies the number of buffers used by a library engine for a data set (not honoured by all engines)
<code>BUFSIZE=0</code>	Specifies the size of a page for a WPS data set
<code>BYERR</code>	Generate an error when a null dataset is used as input to PROC SORT
<code>BYLINE</code>	Generate a BY line title for each BY group in the output
<code>NOCAPS</code>	Do not translate source source and data lines to uppercase
<code>CARDIMAGE</code>	Treat CARDS lines as 80 character width punch card records
<code>CENTER</code>	Align listing output to the center of the page
<code>NOCHARCODE</code>	Do not allow character combinations as a substitute for special characters not on the keyboard
<code>COMAMID=TCP</code>	The communication method to use for establishing remote connections
<code>COMPRESS=NO</code>	Specifies whether to compress observations in output SAS data sets
<code>CONFIG=</code>	List of config files processed during initialisation
<code>CONNECTPERSIST</code>	The remote connection will be persisted after a RSUBMIT block
<code>CONNECTREMOTE=</code>	Identifies the remote server that will be connected to
<code>CPORTVER=SAS92</code>	Controls which type of CPORT file is generated by default by PROC CPORT
<code>DATE</code>	Print the date and time at the top of each page of output
<code>DATESTYLE=LOCALE</code>	Controls how numerical dates are interpreted in the ANYDT informats
<code>NODETAILS</code>	Do not show additional details of data sets when listing data libraries
<code>DEVICE=</code>	Device to be used for graphical output
<code>DFLANG=ENGLISH</code>	Language for EURDF date/time formats and informats
<code>DKRICOND=ERROR</code>	Action for DROP/KEEP/RENAME error conditions on input data sets
<code>DKROCOND=WARN</code>	Action for DROP/KEEP/RENAME error conditions on output data sets
<code>NODMR</code>	Invokes a CONNECT local session
<code>DQUOTE</code>	Toleration support for obsolete DQUOTE option

2

09:34 Wednesday, October 23, 2013

The WPS System

DSNFERR Generate an error when a data set not found error occurs

```

NODTRESET          Does not update the date and time in the titles of the log and
listing file for each new page written
DYNALLOC           Host utility is assumed to support dynamic allocation of work
files
NOECHOAUTO         Do not echo sources lines from the AUTOEXEC file to the log
ENGINE=WPD         The default library engine for data set files
NOERRORABEND      Execution will not terminate when an error occurs, but continue
ERRORS=20         Maximum number of observations for which complete errors
messages are printed
FIRSTOBS=1        The number of the first observation to process in a dataset
FMTERR            Treat missing formats as an error
FMTSEARCH=(WORK.FORMATS)
                  Search path to use when locating user formats
FORMCHAR=|----|+|---+=|-/\<>*

FORMDLIM=         Character to delimit page breaks in listing output
NOFULLSTIMER      Do not write performance statistics to the log
IBUFNO=0          Specifies the number of index file buffers used by a library
engine for a data set (not honoured by all engines)
IBUFSIZE=0        Specifies the size of an index page for a WPS data set
NOIMPLMAC         Do not allow statement-style macro calls
INITSTMT=         Initial statements to execute before any submitted program
INVALIDDATA=.     Value to assign when invalid numeric data is encountered on
input
LABEL             Allow WPS to retrieve and use a label associated with a
variable
LINESIZE=132      Line length for log entries
LOG=SASLOG        Log file configuration parameters
LOGPARM=          Log file configuration parameters
MACRO             Allow use of the macro facility
NOMACROGEN        Do not trace the execution of old-style macros
NOMAUTOLOCDISPLAY Do not display the location from which the autocall macro
source code is compiled
MAUTOSOURCE       Allow the macro autocall feature
MCOMPILE          Allow new macros to be compiled
MCOMPILENOTE=NONE Issue a note to the log when a macro has been successfully
compiled
MEMSIZE=0         Limit on the total amount of memory used by the system
MERGENOBY=NOWARN Sets whether to issue a warning, an error, or no warning when a
MERGE statement is provided with no BY
statement.
MERROR            Generate an error when an undefined macro reference occurs
MINDELIMITER=     Identifies the character to use as the delimiter of the macro
IN operator
MISSING=.         Character to represent missing numeric value
NOMLOGIC          Do not trace the execution of macros
NOMLOGICNEST     Do not display macro nesting information in MLOGIC output
NOMPRINT          Do not display WPS statements generated by macro execution
NOMPRINTNEST     Do not display macro nesting information in MPRINT output
NOMRECALL         Do not search the autocall libraries for an undefined macro
name each time it is invoked
NOMSGCASE         Do not translate messages to uppercase
MSGLEVEL=N        The level of messages displayed
NOMSTORED         Do not use stored compiled macros
MSYMTABMAX=1048576 The maximum size of the macro variable symbol tables
MVARSIZE=8192     The maximum length for in-memory macro variables
NEWS=NEWS         Specifies a file containing messages that are written at the
top of the log
NOTES             Display NOTES in the log

```



```

NUMBER          Print the page number at the top of each ODS LISTING output
page
OBS=9223372036854775807
                The number of the last observation to process in a dataset

NOOPLIST
NOOVP
PAGEENO=1       Page number to use for the next page of printed output
PAGESIZE=60     Controls the number of lines that make up a page of output
PARM=           Parameter string to pass to external program
PARMCARDS=      Name of a fileref to use as the PARMCARDS file
QUOTELENMAX     Warns when quoted string literals exceed 262 characters
3
                The WPS System
                09:34 Wednesday, October 23, 2013

REPLACE         Allow replacement of permanent data sets
S=0             The length of source statements and data lines
SASAUTOS=(SASAUTOS)
                Search list for autocall macros
SASCMD=         Specifies the command to be used by CONNECT to start another
local WPS session
SASHELP=SASHELP Location of the SASHELP library
SASMSTORE=     Name of library containing stored compiled macros
SASSCRIPT=     Location of CONNECT signon scripts
SASTRACE=      Specify the level of debug tracing in the database engines
SASTRACELOC=   Specify where the debug tracing in the database engines is
written
SASUSER=       Location of the SASUSER library
SEQ=8          The number of digits in the numeric part of the sequence field
SERROR         Generate an error when an undefined macro variable reference
occurs
NOSORTCHECK    Do not Warn if the SORTDEV option contains the name of a device
group rather than a specific device such as 3390
SORTCONFIG=    Configuration parameters for internal sort
SORTDEVWARN    Warn if the SORTDEV option contains the name of a device group
rather than a specific device such as 3390
SORTDUP=PHYSICAL Whether the NODUP option is applied to physical or logical
records?
SORTEQUALS     Maintain the order of observations with the same BY value in
PROC SORT
NOSORTMMAP     The internal sort program should not use memory mapped files
SORTSEQ=       Default collation sequence for PROC SORT
SORTSIZE=10M   Hint about the amount of memory to use when performing a sort
NOSORTSTATS
NOSORTVALIDATE Do not validate the sort order on data sets with user specified
sort specifications
SORTWKNO=3     Specifies how many work files are to be allocated for sort
SORTWORK=      Location(s) to put SORT procedure work files
SOURCE         Show source statements in log
SOURCE2        Show source lines from included source files
STIMEFMT=N     Specifies the format to be used for displaying step timings
SUMSIZE=0      The default maximum storage to be used by PROC SUMMARY and PROC
MEANS
NOSYMBOLGEN    Do not write the results of resolving macro variable references
to the log
SYSPARM=       A character string that can be passed to a WPS program
S2=0           The length of secondary source statements, such as those in
included files
TABSIZ=0
TAPEENGINE=WPSTAPE The library engine to use when TAPE is specified
TBUFSIZE=32768 Buffer size for remote communication

```



```

TERMSTMT=          Final statements to execute after any submitted program
NOTHREADS          Disable multi-thread processing
TRANTAB=( , , EOL1_UCS, EOL1_LCS, EOL1_CCL, , , , )

USER=              Default location for all one-level data set names
VALIDVARIABLE=V7  Control the rules that govern what makes up a valid variable
name
VARLENCHK=NOWARN  Controls the behaviour in the DATA step when variables from
different input data sets have different lengths
VNFERR             Generate an error a missing variable condition is encountered
with a _NULL_ data set
WORK=WORK          The location of the WORK library
WORKINIT           Erase all files in the WORK directory on WPS initialisation
NOWORKTERM         Do not erase any files from the WORK directory on WPS
termination
NOWPDHUGE          Do not allow new WPD data sets to have more than 2G records
WPSDSCOMP          DATA steps will be compiled to native machine code
YEARCUTOFF=1920   Cutoff year used when interpreting or generating 2 digit years
in functions and formats

```

Host Options:

```

AUTOEXEC=          The location of a file automatically executed at WPS
initialisation
NOBLKALLOC         Allow a zero BLKSIZE when performing a dynamic allocation for a
data library
BLKSIZE=27998      Specifies the default block size for WPS data libraries
BLKSIZE(DISK)=0    Specifies the default block size for WPS data libraries on DISK
devices
BLKSIZE(OTHER)=6144
                   Specifies the default block size for WPS data libraries on DISK
devices

```

4

The WPS System

09:34 Wednesday, October 23, 2013

```

BLKSIZE(3375)=8192 Specifies the default block size for WPS data libraries on 3375
devices
BLKSIZE(3380)=23476
                   Specifies the default block size for WPS data libraries on 3380
devices
BLKSIZE(3390)=27998
                   Specifies the default block size for WPS data libraries on 3390
devices
BLKSIZE(9345)=6144 Specifies the default block size for WPS data libraries on 9345
devices
CPUCOUNT=1         Number of CPUs available to the application
DB2IN=             The default DB2 tablespace in which to create tables
DB2READBUFF=1     Specifies the number of rows to read from DB2 at a time on z/OS
DB2SSID=DB2       The default DB2 sub-system id
DLDSNTYPE=NONE    Default value to use for the DSNTYPE= option on a LIBNAME
DLEXPCPCOUNT      Report EXCP count for WPS data libraries
NODMS              Use the non-windowed environment
EMAILAUTHPROTOCOL=NONE
                   Sets whether authentication is used for SMTP email connections
EMAILHOST=localhost
                   SMTP server host for email access method
EMAILID=           User id for connecting to SMTP
EMAILPORT=25      Port number for SMTP server for email access method
EMAILPW=*****    Password for connecting to SMTP
EMAILSYS=SMTP     The system with which to send emails

```



```

ENCODING=OPEN_ED-1047
    Specifies the default character encoding
FILEBLKSIZE(DISK)= Specifies the default block size for external files on DISK
devices
FILEBLKSIZE(OTHER)=6400
    Specifies the default block size for external files on OTHER
devices
FILEBLKSIZE(SYSOUT)=264
    Specifies the default block size for external files on SYSOUT
devices
FILEBLKSIZE(TAPE)= Specifies the default block size for external files on TAPE
devices
FILEBLKSIZE(TERM)=264
    Specifies the default block size for external files on TERMINAL
devices
FILEBLKSIZE(3375)=17600
    Specifies the default block size for external files on 3375
devices
FILEBLKSIZE(3380)=23476
    Specifies the default block size for external files on 3380
devices
FILEBLKSIZE(3390)=27998
    Specifies the default block size for external files on 3390
devices
FILEBLKSIZE(3400)=32760
    Specifies the default block size for external files on 3400
devices
FILEBLKSIZE(3480)=32760
    Specifies the default block size for external files on 3480
devices
FILEBLKSIZE(3490E)=32760
    Specifies the default block size for external files on 3490E
devices
FILEBLKSIZE(3590)=32760
    Specifies the default block size for external files on 3400
devices
FILEBLKSIZE(9345)=22928
    Specifies the default block size for external files on 3400
devices
NOFILECC      Use SIZE instead of FILSZ in the host sort control string
FILEDEV=SYSDA Specifies the default device name for new physical files
FILEMSGS      Display messages in the log resulting from dynamic DDname
allocations
FILESPPRI=1   Specifies the default primary space allocation for new physical
files
FILESPPRI=1   Specifies the default secondary space allocation for new
physical files
NOFILESTAT    Do not maintain ISPF member statistics in partitioned data sets
FILESYSOUT=   Specifies the default SYSOUT class for a printer file
FILESYSTEM=MVS Default filesystem to be used when a filename is ambiguous
FILEUNIT=CYL  Specifies the default unit of allocation for new physical files
FILSZ        Use FILSZ in the host sort control string
FONTPATH=(WPSFONTS)

```

5

The WPS System

09:34 Wednesday, October 23, 2013

```

    Search list for TrueType fonts
FULLSTATS     Write expanded step statistics to the log
IMSDLDBR=Y    Controls whether IMS should set the DBRC parameter when it
invokes an IMS DLI region. Y for YES, N for NO, * for

```



```
subsystem default
  NOISPCAPS          Do not convert printable characters to upper case in parameters
to ISPLINK and ISPEXEC
  NOISPCHARF        Do not convert character variables using their associated
formats and informats when they are used as ISPF
variables
  ISPCSR=           Names a variable that will be set by the ISPF interface to the
name of a variable whose value is found to be
invalid
  ISPEXECV=         Names an ISPF variable that, when accessed, invokes an ISPF
service
  ISPMISS=          The value to be assigned to WPS character variables defined to
ISPF when the associated ISPF variable has length
zero
  ISPMSG=           The name of an ISPF variable that will be set to a message ID
when a variable is found to be invalid
  NOISPNOTES        ISPF error messages will not be written to the WPS log
  ISPNUMF           Convert numeric variables using their associated formats and
informats when they are used as ISPF variables
  NOISPZNZTRC       Non-zero ISPF service return codes will not be written to the
WPS log
  NOISPPT           ISPF parameter pointers and lengths will not be written to the
WPS log
  NOISPTRACE        ISPF parameter lists and service return codes will be written
to the WPS log
  NOISPVDEFA        Only variables passed automatically to the VDEFINE user exit
are defined to ISPF
  NOISPVDLT         VDELETE is not issued before a variable is defined with the
VDEFINE service
  NOISPVDTTC        Calls to VDEFINE are not traced to the WPS log
  ISPVMSG=          Specifies the ISPF message ID to be set by the VDEFINE user
exit when an informat for a variable returns a error
return code
  ISPVRMSG=         Specifies the ISPF message ID to be set by the VDEFINE user
exit when a variable has a null value
  ISPVTMSG=         Specifies the ISPF message ID to be set by the VDEFINE user
exit when the ISPVTRAP option is in effect
  ISPVTNAM=         Restricts the information displayed by the ISPVTRAP option to
the specified variable only
  ISPVTPNL=         Specifies the ISPF panel that is to be displayed by the VDEFINE
user exit when the ISPVTRAP option is in effect
  NOISPVTRAP        The VDEFINE user exit will not write debugging information to
the WPS log
  ISPVTVARS=        Specifies the prefix for the ISPF variables to be set by the
VDEFINE user exit when the ISPVTRAP option is in
effect
  LOCALE=ENGLISH_UNITEDSTATES
                    Specifies the current locale for the WPS session
  NOMEMRPT
  PATH=             Specifies the location of binaries
  PRINT=SASLIST     Specifies the location to which the listing output will be
written
  SEQENGINE=WPDSEQ  The default library engine for sequential data set files
  NOSORTBLKMODE     The sort program does not support a block mode interface
  SORTCUTP=4194304 Amount of storage above which the host sort utility is
appropriate
  SORTDEV=
  SORTEQOP          Host sort routine implements EQUAL option
  NOSORTLIST        The LIST option is not to be sent to the host sort utility
  SORTLOCALE        Host sort routine implements LOCALE option
```



```

SORTMAXKEY=4084    Specifies the maximum key length for the host sort routine
SORTMAXOFF=4092   Specifies the maximum key offset permitted for the host sort
routine
NOSORTMSG          The option MSG=CP to be passed to the host sort utility
SORTNAME=SORT     The name of the host sort utility
SORTOPTS          An OPTIONS statement should be generated for the host sort
utility
SORTPARM=         Additional options to be passed to the host sort utility
SORTPGM=BEST     The sort program to be used by PROC SORT
SORTSUMF          The host sort utility supports SUM FIELDS=NONE
NOSTDIO           Use the options SYSIN, LOG and PRINT for the input, log and
output
STIMER           Record performance statistics after each PROC or DATA step
SYSIN=SYSIN      Specifies the location from which source code will be read
SYSPREF=ADJC     Specifies a prefix for partially qualified physical file names
NOS99NOMIG       Do not restore migrated data sets
NOVERBOSE
NOVSAMLOAD        Empty VSAM datasets cannot be loaded
VSAMREAD         VSAM datasets can be read using INFILE statements

```

6

The WPS System

09:34 Wednesday, October 23, 2013

```

NOVSAMUPDATE      VSAM datasets opened using INFILE statements cannot be updated
WORKPERMS=700    The permissions to give the WORK library
WPSDSCOMPARCH=0  Specifies the maximum architecture level for which to generate
code when compiling DATA steps
NOWPSDSCOMPDEBUG Does not generate debug information about compiled DATA steps
XCMD             The "X" command is available to use.

```

NOTE: Procedure OPTIONS step took :

```

real time : 0.030
cpu time  : 0.026
EXCP count: 0

```

```

13      +
14      + /* CREATE SINGLE LARGE NATIVE DATA SET */
15      +
16      + %MACRO VERIFY;
17      +
18      + DATA _NULL_;
19      +   FORMAT NOW DATETIME21.2;
20      +   NOW=DATETIME();
21      +   PUT "START: " NOW DATETIME21.2;
22      + RUN;
23      +
24      + %DO _M=1 %TO 8;
25      +   DATA WORK.TEMP(DROP=_I _N);
26      +   ARRAY WPS(100) WPS1-WPS100;
27      +   DO _I=1 TO 1000;
28      +     DO _N=1 TO 100;
29      +       WPS(_N) = _I;
30      +     END;
31      +   OUTPUT;
32      +   END;
33      +   RUN;
34      +
35      + %END;
36      +
37      + OPTIONS NOCENTER;

```




```
38      +
39      + PROC CONTENTS DATA=WORK.TEMP;
40      + RUN;
41      +
42      + OPTIONS OBS=32;
43      +
44      + PROC PRINT DATA=WORK.TEMP;
45      + RUN;
46      +
47      + DATA _NULL_;
48      +   FORMAT NOW DATETIME19.2;
49      +   NOW=DATETIME();
50      +   PUT "   END: " NOW DATETIME21.2;
51      + RUN;
52      +
53      + %MEND;
54      +
55      + %VERIFY;
```

START: 23OCT2013:09:34:19.28

NOTE: The data step took :

7

The WPS System

09:34 Wednesday, October 23, 2013

real time : 0.030
cpu time : 0.012
EXCP count: 0

NOTE: Data set "WORK.TEMP" has 1000 observation(s) and 100 variable(s)

NOTE: The data step took :

real time : 0.127
cpu time : 0.111
EXCP count: 0

NOTE: Data set "WORK.TEMP" has 1000 observation(s) and 100 variable(s)

NOTE: The data step took :

real time : 0.125
cpu time : 0.118
EXCP count: 0

NOTE: Data set "WORK.TEMP" has 1000 observation(s) and 100 variable(s)

NOTE: The data step took :

real time : 0.131
cpu time : 0.127
EXCP count: 0

NOTE: Data set "WORK.TEMP" has 1000 observation(s) and 100 variable(s)

NOTE: The data step took :

real time : 0.110
cpu time : 0.107
EXCP count: 0



```
NOTE: Data set "WORK.TEMP" has 1000 observation(s) and 100 variable(s)
NOTE: The data step took :
      real time : 0.123
      cpu time  : 0.119
      EXCP count: 0
```

```
NOTE: Data set "WORK.TEMP" has 1000 observation(s) and 100 variable(s)
NOTE: The data step took :
      real time : 0.132
      cpu time  : 0.128
      EXCP count: 0
```

```
NOTE: Data set "WORK.TEMP" has 1000 observation(s) and 100 variable(s)
NOTE: The data step took :
      real time : 0.103
      cpu time  : 0.101
```

8

The WPS System

09:34 Wednesday, October 23, 2013

```
EXCP count: 0
```

```
NOTE: Data set "WORK.TEMP" has 1000 observation(s) and 100 variable(s)
NOTE: The data step took :
      real time : 0.102
      cpu time  : 0.099
      EXCP count: 0
```

```
NOTE: Procedure CONTENTS step took :
      real time : 0.139
      cpu time  : 0.078
      EXCP count: 0
```

```
NOTE: 32 observations were read from "WORK.TEMP"
NOTE: Procedure PRINT step took :
      real time : 0.215
      cpu time  : 0.206
      EXCP count: 0
```

```
END: 23OCT2013:09:34:20.61
```

```
NOTE: The data step took :
      real time : 0.002
      cpu time  : 0.002
      EXCP count: 0
```

```
End of %INCLUDE(level 1) ADJC.ADJCVERF.JOB03362.D0000102.?
56      + RUN;
End of %INCLUDE(level 2) SOURCLIB(xverify)
```

```
NOTE: Submitted statements took :  
      real time : 2.098  
      cpu time  : 1.837]]>
```

Placement of Data Libraries

A WPS data library can be held within a sequential z/OS dataset, a VSAM linear z/OS dataset, or a USS (UNIX System Services) HFS/zFS directory. Whilst using USS files offers certain advantages over sequential z/OS or VSAM linear z/OS datasets, particularly when copying, renaming, and transferring using UNIX utilities, such files are far from ideal, as HFS volumes cannot be allocated, used, and de-allocated within a JCL script in the same way that a z/OS file can be.

The default WPS dataset engine is called WPD. WPS will automatically detect whether the library holding a dataset is contained within a z/OS dataset, a VSAM linear z/OS dataset, or a USS directory, and operate on the library accordingly. WPS can use all three types of library within a single program.

For details of the syntax related to data libraries supported by WPS, please consult the separate document called *WPS Reference for Language Elements*.

The default WORK library is a z/OS sequential dataset-based library. The default SASHELP library is provided as a permanent z/OS sequential dataset-based library.

Sequential z/OS Dataset-Based WPS Data Libraries

A WPS data library within a sequential z/OS dataset is represented by a single binary structure. Each individual WPS dataset is held within this structure. The structure of the library and its member datasets is a format proprietary to WPS. Datasets can only be added, deleted and moved from within a native z/OS file data library using WPS.

For sequential z/OS dataset libraries, we recommend attributes of half-track blocking and the record format should be 'undefined'. As such, the record length is therefore effectively immaterial. For example, when initially creating the file on a 3390 disk, specify the following for the best results:

```
DCB=(DSORG=PS,RECFM=U,BLKSIZE=27998,LRECL=27998)
```

Storage of a WPS dataset using z/OS datasets is best configured through use of DD statements within the JCL used to launch WPS. The DDNAME on such a statement becomes an implicit WPS Library identifier within the program and can therefore be used directly as if a LIBNAME statement has been issued. For instance if a DDNAME "mylib" is declared, then a WPS dataset called "dataset1" in the z/OS dataset referred to by "mylib" can be referenced as "mylib.dataset1".

VSAM Linear z/OS Dataset-Based WPS Data Libraries

A WPS data library within a VSAM linear z/OS dataset is very similar to a z/OS sequential dataset in terms of internal structure. Each individual WPS dataset is held within the structure, which is in a format that is proprietary to WPS. Within the structure, datasets can only be added, deleted and moved by using WPS.

Initially, defining the VSAM linear z/OS dataset would normally be performed via an IDCAMS `DEFINE` similar to:

```
DEFINE CLUSTER(NAME(<LDS_name>) LINEAR CYLINDERS(<pri sec>) SHAREOPTIONS(1,3))
```

Subsequently, just as with sequential z/OS datasets, a `DD` statement would be used to form the association between a library name and the `LDS_name`.

USS Directory-Based WPS Data Libraries

A WPS data library can be represented by a USS directory, with each dataset within the library being represented by a single file with a file extension of `.wpd`: z/OS datasets can be added to, and removed from, the library by use of file manipulation tools within USS such as `cp` (copy), `mv` (move), `rm` (delete) etc.

When the contents of a USS file-based data library are listed, the list of members returned is the list of files in the associated directory with the following extensions:

- `wpd` = WPS Dataset
- `wpcat` = WPS Catalogue
- `wpcvw` = WPS View
- `wpidix` = WPS Index

Before WPS can use a USS directory-based data library, an HFS or zFS volume must be allocated and mounted into the USS file system. The WPS user must also have sufficient privileges to perform the operations they require on the library.

WPS libraries can be defined using, for example, the `LIBNAME` statement. Before issuing a `LIBNAME` statement, the USS directory to which the statement refers must exist.

USS Permissions

To create a dataset, the user must have `read`, `write` and `execute` permissions on the USS directory.

To read a dataset, the user must have `read` and `execute` permissions on the USS directory and `read` permissions on the `.wpd` file containing the WPS dataset.

WPS Data Files

WPS datasets are stored within either a z/OS dataset or a USS directory. The `WORK` library may be defined as either of these types of library. Consider the following example program :

```
LIBNAME mylib '/u/<userid>/wpsdata';
DATA mylib.data1;
A = 1;
run;
```

This will create a WPS data file called `/u/<userid>/wpsdata/mydata.wpd`.

Note:

Under the USS file system, path, directory and file names are case sensitive.

Data can be imported to and exported from native WPS files and DB2 tables if necessary. WPS and DB2 are discussed in the [Using DB2](#) (page 45) section of this guide.

Alternatively, assuming that a `DDNAME` of `mylib1` is defined in the JCL used to launch the program, and the `DDNAME` points to a z/OS dataset containing a WPS data library, then the following program could be used to access a dataset within the `mylib1` z/OS dataset:

```
DATA mylib1.data1;
A = 1;
run;
```

Data Sharing between Multiple WPS Users

The processing of data files within WPS data libraries in sequential z/OS datasets is controlled through the `DISP` parameter of the relevant `DD` or `libname` statement.

If the contents of a data library are to be changed in any way, then the `DISP=` value should be specified as either `NEW` (when creating a new data library) or `OLD` (when modifying the contents of an existing one). Specifying `DISP=SHR` automatically prevents modification of the data library. Data files within a library allocated with `DISP=SHR` may only be used as input to the program.

The concurrent processing of data files within a WPS data library held within a VSAM linear z/OS dataset is controlled through the `SHAREOPT` parameter values. Nevertheless, only one user should be allowed to update data files in the library at any point in time. For that reason, the first ('crossregion') parameter value should be set to '1', e.g. `SHR(1, n)`.

For USS-based data libraries, the access mode setting for the directory, and files contained within it, is key. This is because WPS is reliant on the permissions granted to the library files that the program being run is accessing. In addition to this, WPS coordinates the concurrent access to the files using system-level file locking, to prevent multiple programs from writing to the same file.

The WORK Library

The WORK data library for each user should be unique and not shared.

When using WPS under z/OS, the supplied JCL procedure will allocate a temporary WORK data library each time it is executed. This can be overridden within the JCL.

When using a USS directory-based WORK library, when a WPS job is submitted, a new unique temporary directory is created below the USS WORK directory specified. This temporary directory will be deleted on completion of the WPS job.

To change the WORK library to use a USS directory location instead of a z/OS dataset, edit the WPSPROC member of the CNTL library. Change the WORK DD name to point to a USS directory. The entry should be similar to the example below:

```
//WORK DD PATHDISP=(KEEP,KEEP),PATH='/u/wps/work'
```

WPS Configuration Options

WPS configuration options can be set via any of the following methods:

- The CONFIG file.
- The OPTIONS statement.
- A PARM clause on the EXEC JCL statement. (In the supplied JCL procedure, this is defined by the parameter immediately following OPTIONS=).

These methods are discussed in the sections below.

If option names are duplicated within these possibilities, the order of precedence is:

- The OPTIONS statement settings override settings made via:
- The PARM clause on the EXEC statement, which in turn override settings made via:
- The CONFIG file.

Note:

Some options may only be specified at the time WPS is invoked, that is, via the CONFIG file or the PARM clause.

CONFIG File

Default WPS configuration options are set in a <wpspfx>.CNTL member called CONFIG.



Note:

Unlike option values, option names are not case sensitive.

The contents of the default CONFIG file delivered with WPS is shown below:

```
* WPS SYSTEM CONFIG FILE CONTAINING SYSTEM OPTIONS
BLKSIZE=27998
BYLINE
CENTER
NOCHARCODE
DATE
DB2SSID=DB2
DKRICOND=ERROR
DKROCOND=WARN
DSNFERR
ENGINE=WPD
NOERRORABEND
FILEUNIT=CYL
FILESPPRI=1
FILESPPSEC=1
FIRSTOBS=1
FMTERR
FMTSEARCH= (WORK.FORMATS)
FONTPATH= (WPSFONTS)
LINE SIZE=132
LOG=SASLOG
MACRO
MAUTOSOURCE
MERROR
NOMLOGIC
NOMPRINT
MSGLEVEL=N
NEWS=NEWS
NOTES
OLDMAC
PAGE NO=1
PAGE SIZE=60
PRINT=SASLIST
REPLACE
S=0
S2=0
SASAUTOS= (SASAUTOS)
SASHELP=SASHELP
SEQ=8
SEQENGINE=WPDSEQ
SERROR
SORTCUTP=4194304
SORTPGM=BEST
SORTSIZE=10M
SOURCE
SOURCE2
SUMSIZE=0
SYSIN=SYSIN
STIMER
NOSYMBOLGEN
WORK=WORK
WORKINIT
NOWORKTERM
WPSTRACE=ERROR
```

```
YEARCUTOFF=1920
```

Overriding the Default CONFIG File

When WPS is launched, the CONFIG option may be coded via the EXEC statement.

An example of an EXEC statement without the CONFIG option can be found in the *Executing Programs written in the language of SAS* [↗](#) (page 42) section of this guide, but, by default, the EXEC statement looks like this :

```
//WPSEXEC EXEC WPSPROC
```

With the CONFIG option passed in, the JCL will look something like this :

```
//WPSEXEC EXEC WPSPROC, CONFIG=<userid>.WPSCNTL(CONFIG)
```

Note:

<userid>.WPSCNTL(CONFIG) is a z/OS dataset or PDS member containing a WPS CONFIG file. It can be used to override all WPS CONFIG options.

SAS Language OPTIONS Statement

Specific options can be set during WPS execution by using the SAS language OPTIONS statement. For example:

```
OPTIONS NOERRORABEND NOSOURCE2 S2=80;
```

Note:

Unlike option values, option names are not case sensitive.

Listing Current SAS Language OPTIONS

To list the current SAS language option settings during a WPS session, use the PROC OPTIONS statement as shown below:

```
PROC OPTIONS; run;
```


Configuration Options for sorting data

There are a number of `CONFIG` options that apply to sorting data. Three of them may deserve specific attention, depending on site-specific requirements:

- `SORTSIZE` - defines the amount of memory that may be used for sorting data. This option applies to both the `WPS` sort and whichever `SORT` utility is installed on the host system. The amount specified will be taken from the `REGION` size specified for the batch job.
- `SORTCUTP` - works in conjunction with the `SORTPGM` option value. The value defines the point at which the `WPS` internal sort is used in preference to whichever `SORT` utility is specified by the `SORTPGM` option.
- `SORTPGM` - may be set to `HOST`, `BEST` or `WPS`. If `WPS` is specified, the internal sort program provided with `WPS` is always used. If `HOST` is specified, then whichever `SORT` utility is installed on the host system is used. If `BEST` is specified, then the decision to use `HOST` or `WPS` sort is based on the `SORTCUTP` option value.

Fine-tuning WPS to Optimise Runtime Performance

On z/OS, `WPS` makes use of the LE (Language Environment) heap storage pool facility to improve the performance of memory management activities. In general, this is beneficial, but it does come at the cost of a slightly elevated baseline memory usage, as more memory is required to manage the heap storage pools.

Note:

If you wish to switch off the usage of heap storage pools, ensure that the content read from the `CEEOPTS` DD contains the line `HEAPPOOLS (OFF)`.

The heap storage pool facility can be dynamically tuned to improve the performance associated with particular application memory usage patterns and footprint. The initial default settings used by `WPS` have been selected to provide a reasonable balance of performance for the majority of `WPS` usage. However, it may be that, for a particular workload, the performance can be improved.

The heap storage pool facility can be configured to report the optimum settings associated with a particular run of the application. Once an alternative heap storage pool configuration has been obtained, the `CEEOPTS` can be modified so that future job runs always use this configuration. The `CEEOPTS` can either be tuned for a single job, so that those settings are always used for that job, or the required settings can be applied to the default `CEEOPTS` for all `WPS` runs using `WPSPROC`.

Note:

For more information about how to tune the heap storage pool facility configuration, consult the "tuning heap storage" section of the IBM z/OS LE programming guide.

Running WPS

Executing Programs written in the language of SAS

The @VERIFY member of PDS <wpspfx>.CNTL that was used for installation verification may be used as a starting point for running any other language of SAS programs.

Files Created During Execution

The WPS Batch Job interface creates and references a number of z/OS datasets, in addition to any that may be created by user-written programs.

The default files are:

Default Output File	Description
SASLOG	Default log location
SASLIST	Default listing location

Note:

Other JCL DDNAMEs are allocated within the WPSPROC JCL procedure and these may contain diagnostic output from WPS.

Using WPS

Examples of Programs and the JCL Required to Execute them

To demonstrate the capabilities of WPS, there are some example programs, together with the JCL required to execute them, in the `<wpspfx>.CNTL` library :

Member Name	Description
@SEQFB	Demonstration of reading and writing sequential datasets that have RECFM(FB)
@SEQVB	Demonstration of reading and writing sequential datasets that have RECFM(VB)

The use of these sample jobs is similar to the WPS sample job listed in *Edit the Sample Verification Job* [↗](#) (page 25).

Note:

You are advised to review the example program and JCL prior to submission.

Using SAS Data Libraries

WPS can read z/OS-based SAS dataset libraries directly using the `SASDASD` library engine. However, WPS is unable to write to the `SASDASD` format. Therefore, persistent SAS dataset libraries that get updated will need to be migrated to WPS dataset libraries, which can be achieved using a simple `PROC COPY`. For more information see the *WPS Migration Guide for z/OS*.

World Programming is able to provide consultancy on the migration of data. It is important to consider data migration before any existing environment that is to be run in parallel becomes unavailable. The following sections are a summary of the data formats that can be accessed using WPS.

Using SAS XPORT Files

WPS is able to read and write `XPORT` files using the `XPORT` data library engine.

Using SAS7BDAT Files

WPS is able to read and write `SAS7BDAT` (SAS v7/8) files using the `SASBDATA` data library engine.

Using SD2 Files

WPS is able to read `SD2` (SAS v6) files using the `SD2` data library engine.

Using SASTAPE Files

WPS is able to read `SASTYPE` (SAS v6) files using the `SASTAPE` data library engine.

Using RDBMS (DB2, SQLServer)

WPS is able to read and write data for a variety of relational database management systems (RDBMS) such as SQL Server (Microsoft) and DB2 (IBM). There is a section later in this chapter about *Using DB2* [↗](#) (page 45).

Using CPORT Files

WPS is able to read and write `CPORT` files using `PROC CPORT` and `PROC CIMPORT`.

Running Existing Programs written in the language of SAS

Many existing programs written in the language of SAS will run unaltered. However, other programs may require modification depending on the complexity and nature of the programs.

Migrating Existing WPS Data

Typically, data produced by earlier versions of WPS can be accessed transparently. There may, however, be performance benefits associated with moving the data to a new library created by the latest version of WPS. Check with the WPS release notes for information.

Migrating Existing WPS Programs

Existing programs written/run with previous versions of WPS are typically compatible with the latest version of WPS. However, you should first refer to the relevant release notes before running older programs.

Using MXG

MXG is an application written by Merrill Consultants in the language of SAS. If you are considering using WPS with MXG, you will find relevant information in the WPS Migration Guide for z/OS [↗](#) (page 48).

Using DB2

WPS can connect to, and make use, of DB2 data.

Note:

The lower-case `vm` characters in the dataset names below represents the version and modification level of the DB2 sub-system in use. Replace these characters with `91` for version 9.1, `A1` for version 10.1, and so on. You may need to ask your system administrator where these DB2 CLI datasets can be found.

WPS uses the DB2 Call Level Interface (CLI) provided by IBM. As WPS is a 31-bit XPLINK application, it specifically requires the XPLINK version of the CLI (DSNAOCLX). To make a connection to DB2 from WPS, the target DB2 installation must include support for the DB2 CLI.

Briefly, the CLI must be bound into the DB2 sub-system in use. Refer to the DSNTIJCL member in the DB2 sample library `DSNvm0.SDSNSAMP` for an example job that accomplishes this process.

For detailed information on installing and setting up the DB2 runtime environment to enable support for the DB2 CLI, please consult the manual entitled "ODBC Guide and Reference" available from the IBM website. Links to the different versions of the DB2 for z/OS manuals are given at: <http://www-01.ibm.com/support/docview.wss?uid=swg27039165> [↗](#).

Once you have installed the DB2 CLI on your mainframe, there is an additional configuration step required to get WPS to connect to DB2: three additional libraries must be included in your STEPLIB. (We recommend that you do this on an installation-wide basis by modifying the STEPLIB in the WPSPROC member of the <wpspfx>.CNTL library).

Before you edit the STEPLIB in WPSPROC, it will look like this:

```
//* DEFINE STEPLIB
//STEPLIB DD DISP=(SHR,PASS),DSN=&LOAD
// DD DISP=SHR,DSN=&WPSPFX..LOAD
```

You need to add the following three libraries to the STEPLIB:

```
//* DEFINE STEPLIB
//STEPLIB DD DISP=(SHR,PASS),DSN=&LOAD
// DD DISP=SHR,DSN=&WPSPFX..LOAD
// DD DISP=SHR,DSN=DSNvm0.SDSNLOAD2
// DD DISP=SHR,DSN=DSNvm0.SDSNEXIT
// DD DISP=SHR,DSN=DSNvm0.SDSNLOAD
```

Once you have completed this step, you should be able to connect to DB2 from WPS.

Important:

Please make sure you check for DB2/CLI connectivity on your z/OS system before you attempt to use WPS to access the database.

Once the DB2 CLI support has been enabled by following these steps, WPS can make use of DB2 via a LIBNAME statement specifying "DB2" as the engine name, or by using a PROC SQL CONNECT statement.

Using sequential engines on tape devices

WPS supports the writing of sequential libraries (WPSSEQ, SASSEQ, XPORT) to tape, although such a library can only be written to in one step, be it a DATA or a PROC step.

Important:

To work around this limitation, multiple PROC COPY or PROC DATASETS; COPY statements can be merged into a single PROC DATASETS; COPY statement that will copy datasets from multiple input libraries to one tape sequential library. For example:

```
PROC COPY IN=INLIB1 OUT=SEQLIB; RUN;
PROC COPY IN=INLIB2 OUT=SEQLIB; RUN;
```

could be rewritten as:

```
PROC DATASETS LIB=INLIB1 NOLIST NODETAILS;
COPY IN=INLIB1 OUT=SEQLIB;
COPY IN=INLIB2 OUT=SEQLIB;
RUN;
```



Each time a subsequent DATA or PROC step writes to the sequential library, a completely new version of the library is written, overwriting the previous version.

Using Fonts

WPS can support any TrueType font. Only a single font family - Vera - is supplied with WPS. If you require additional fonts, you can copy them into members in PDSE `<wpspfx>. FONTS` and use them from WPS.

Further Reading


Available Reference Material

The following separate guides are also available as reference material:

- WPS Migration Guide for z/OS
- WPS Reference for Language Elements
- WPS Communicate Guide
- 'What's New' document for the current version.

WPS Migration Guide for z/OS

This guide will help you with the process of moving to WPS on the z/OS platform. It has sections on how to migrate programs and data and includes information specifically to help with migrating an MXG environment.

The migration guide is supplied as part of the distribution package in PDF format; otherwise, it can be downloaded from the following section of our website: <http://www.worldprogramming.com/support/docs/wps> .

Appendix A - Running WPS from UNIX Systems Services

Introduction

WPS can be run from a UNIX Systems Services (USS) session. The following sections show you how to prepare for this and then how to go on and run WPS from USS.

Install the WPS USS Components

The installation of the USS components and the optional creation of the `WPSHOME` directory is performed by the submission of a single JCL job contained in the `@INSTUSS` member of `<wpspfx>.DLIB`.

Permission Settings

The user installing WPS via these instructions will become the 'owner' of the WPS directory and files created in the HFS directory structure. The installation process described will create the `WPSHOME` directory and create within it files necessary for the execution of WPS.

Permission settings for the directory and all files contained within it are 755. This means that:

- *The 'owner' of the file has:* read, write, and execute permissions
- *Members of the file group have:* read and execute permissions
- *All other users have:* read and execute permissions.

If these permission settings are inappropriate for your site standards, they can easily be changed by using the `CHMOD` command in `OMVS`.

For example:

```
chmod -R <permissions> "<wpshome>"
```

Edit the @INSTUSS JCL

The @INSTUSS member must be edited prior to submission using ISPF Option 2 or an equivalent editor. Follow the instructions at the top of the file, making the appropriate substitutions as required. Before editing, it should look similar to the following:

```
// <add a jobcard here>
//*
/*-----*/
/* @INSTUSS : INSTALL COMPONENTS INTO THE WPS USS HOME DIRECTORY*/
/*-----*/
/*
/* (1) ADD A SUITABLE JOB CARD
/* (2) CHANGE <wpsdlib> TO THE WPS Distribution library name
/* (3) CHANGE <wpspfx> TO THE D/S PREFIX FOR WPS INSTALL LIBRARIES
/* (4) CHANGE <wpshome> TO THE WPS USS HOME Directory name
/* (5) SUBMIT THIS JOB AND THEN CHECK THE OUTPUT
/*
//STEP01 EXEC PGM=IKJEFT1B,DYNAMNBR=999
//SYSEXEC DD DISP=SHR,DSN=<wpsdlib>
//@USS DD DISP=SHR,DSN=<wpspfx>.USS
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//SYSTSIN DD DATA,DLM='++'
PROF NOPREFIX
RINSTUSS WPSHOME(<wpshome>) WPSPFX(<wpspfx>)
++ END OF //SYSTSIN
```

Important:

<wpshome> is a USS directory. Since USS directories are case sensitive, you should enter `CAPS OFF` in the Command section of your TSO edit screen before you make any changes. This will ensure that TSO does not convert the entire text to upper case.

Submit the @INSTUSS Job

Submit the updated JCL in @INSTUSS and check that the job completes with a return code of zero.

Note:

If the wpshome directory has not been pre-allocated, then the @INSTUSS job will allocate it with a permission of 755.

After the @INSTUSS job has been executed, the <wpshome> directory should be populated with the WPS/USS entities.

About the USS z/OS Dataset

Once the @INSTUSS job has completed and the <wps>home> directory has been populated you may optionally delete the <wps>prefix>.USS z/OS dataset as it is no longer needed for the execution of WPS.

The USS PAX command that is used to populate the <wps>home> directory uses the -pp option that will restore the files with default access permissions of 755.

Launching WPS

USS Command

WPS is run from USS using the command:

```
<WPSHOME>/bin/wps <filename>
```

Output

WPS uses the standard output (STDOUT) stream for its logging and the standard error (STDERR) stream to show errors. By default, STDOUT and STDERR write to the screen. If a file is required for the log and error output then it can be redirected using a command similar to this:

```
<WPSHOME>/bin/wps <filename> >log.txt 2>&1
```

Making it Easier

It will make life easier for those users regularly executing WPS on USS if their login scripts contain the following lines:

```
export WPSHOME=<wps>home>
alias runwps=${WPSHOME}/bin/wps
```

This will enable WPS to be run by issuing the command:

```
runwps <filename>
```

Installation Verification

The @VERIFY member of PDS <wps>prefix>.CNTL, supplied with WPS, contains a sample job that can be used to verify that the installation of WPS has been successfully completed for z/OS.

The equivalent process should now be carried out to show that the USS installation of WPS has been successful. Enter the command:

```
runwps '// '<wpspfx>.CNTL(XVERIFY) ' " >log.txt 2>&1
```

to run the XVERIFY source program. The SASLOG output will appear as `log.txt` and the SASLIST output as `XVERIFY.lst`. These files should be studied to verify that the installation has been successfully completed.

Appendix B - Running WPS from TSO

Introduction

One function of the `<wpfpx>.DLIB(@INSTALL)` job that was used at the start of the installation procedure is to populate z/OS library `<wpfpx>.CLIST` with a member named `TSOWPS`. This member is a `CLIST` that allows use of WPS in real time under native TSO (Time Sharing Option) or under TSO/ISPF. The only difference between the two environments is that the final output is presented in slightly different ways.

The `CLIST` features a large number of arguments which may either be left set at their default values, or set to comply with site-specific standards, prior to making the facility available to end users. The list of defined arguments is shown below.

Note:

The main reason for the length of this list is the number of file allocations that need to be made to enable WPS to run.

Arguments and default settings for the TSOWPS CLIST

- **WSPFX:** This string must contain the z/OS dataset prefix for the WPS installation that is to be used. Either set this on the invocation, or ensure that the `CLIST` specifies the appropriate default for the installation. This prefix is used to locate all the z/OS datasets within the installation. This parameter must have a value, either explicit or an implicit default, in order for the `CLIST` to be able to initiate the WPS invocation.
- **USERPFX:** This string defaults to the user's high-level z/OS dataset name qualifier, and will be used when qualifying unquoted z/OS dataset arguments. This is similar to the way that ISPF (Interactive System Productivity Facility) employs the user prefix for unquoted z/OS dataset names. Pass in an alternative `HLQ` string to be used as the initial optional qualification name.
- **OUTDSNPFX:** An optional parameter that, if specified, will be used to form the default stem name of the names of all output z/OS datasets that are not otherwise specified. If the `OUTDSNPFX` name is not a quoted string, then it will be qualified with the `USERPFX` value. If this parameter is not specified, then the default stem name will be formed from the `USERPFX` and `DSQUAL` values.

- **DSQUAL:** An optional parameter that defaults to `.WPS` and is used when `OUTDSNPFEX` is not specified. It is combined with `USERPFEX` to form the default stem name for all output z/OS datasets that are not specified elsewhere.
- **OPTIONS:** An optional parameter that can be used to pass option values into the invocation of WPS.
- **SYSPARM:** An optional parameter that can be used to pass any `SYSPARM` bindings into the invocation of WPS.
- **CONFIG:** An optional parameter that can be used to specify a single configuration z/OS dataset name or z/OS dataset member that will be passed into the WPS invocation before the installation-based configuration member. If an unquoted name is used, then it will be prefix-qualified with the `USERPFEX` value.
- **SASAUTOS:** An optional parameter that can be used to specify a single z/OS dataset name or z/OS dataset member that will be passed into the WPS invocation as the source of `AUTOCALL` macros ahead of the installation-based source. If an unquoted name is used, then it will be prefix-qualified with the `USERPFEX` value.
- **SASHELP:** An optional parameter that can be used to specify a single z/OS dataset name or z/OS dataset member that will be passed into the WPS invocation as the source of the location of the `SASHELP` library. If an unquoted name is used, then it will be prefix-qualified with the `USERPFEX` value.
- **WPSFONTS:** An optional parameter that can be used to specify a single z/OS dataset name that will be passed into the WPS invocation as the source of font data ahead of the installation-based source. If an unquoted name is used, then it will be prefix-qualified with the `USERPFEX` value.
- **WPSLOAD:** An optional parameter that can be used to specify a single z/OS dataset name that will be passed into the WPS invocation as the load library ahead of the installation based load library. If an unquoted name is used, then it will be prefix-qualified with the `USERPFEX` value.
- **SASLIST:** An optional parameter that can be used to specify the z/OS dataset name to which the `SASLIST` output will be written. The special value "DUMMY" can be used to cause the output to be ignored by binding it to `DUMMY`. The special value "*" can be used to cause the output to be directed to the terminal session. If an unquoted z/OS dataset name is used, it will be prefix-qualified with the `USERPFEX` value. Any existing output z/OS dataset will be reused and overwritten. If the z/OS dataset name does not exist, it will be allocated using the allocation parameters from the `SASLISTSIZE` parameter.
- **SASLISTSIZE:** An optional parameter that specifies the default size for the `SASLIST` output z/OS dataset.
- **SASLOG:** An optional parameter that can be used to specify the z/OS dataset name to which the `SASLOG` output will be written. The special value "DUMMY" can be used to cause the output to be ignored by binding it to `DUMMY`. The special value "*" can be used to cause the output to be directed to the terminal session. If an unquoted z/OS dataset name is used, it will be prefix-qualified with the `USERPFEX` value. Any existing output z/OS dataset will be reused and overwritten. If the z/OS dataset name does not exist, it will be allocated using the allocation parameters from the `SASLOGSIZE` parameter.
- **SASLOGSIZE:** An optional parameter that specifies the default size for the `SASLOG` output z/OS dataset.

- **WPSTRACE:** An optional parameter that can be used to specify the z/OS dataset name to which the `WPSTRACE` output will be written. The special value "DUMMY" can be used to cause the output to be ignored by binding it to `DUMMY`. The special value "*" can be used to cause the output to be directed to the terminal session. If an unquoted z/OS dataset name is used, it will be prefix-qualified with the `USERPFX` value. Any existing output z/OS dataset will be reused and overwritten. If the z/OS dataset name does not exist it will be allocated using the allocation parameters from the `WPSTRACESIZE` parameter.
- **WPSTRACESIZE:** An optional parameter that specifies the default size for the `WPSTRACE` output z/OS dataset.
- **CEEDUMP:** An optional parameter that can be used to specify the z/OS dataset name to which the `CEEDUMP` output will be written. The special value "DUMMY" can be used to cause the output to be ignored by binding it to `DUMMY`. The special value "*" can be used to cause the output to be directed to the terminal session. If an unquoted z/OS dataset name is used, it will be prefix-qualified with the `USERPFX` value. Any existing output z/OS dataset will be reused and overwritten. If the z/OS dataset name does not exist it will be allocated using the allocation parameters from the `CEEDUMPSIZE` parameter.
- **CEEDUMPSIZE:** An optional parameter that specifies the default size for the `CEEDUMP` output z/OS dataset.
- **CEERPT:** An optional parameter that can be used to specify the z/OS dataset name to which the `CEERPT` output will be written. The special value "DUMMY" can be used to cause the output to be ignored by binding it to `DUMMY`. The special value "*" can be used to cause the output to be directed to the terminal session. If an unquoted z/OS dataset name is used, it will be prefix-qualified with the `USERPFX` value. Any existing output z/OS dataset will be reused and overwritten. If the z/OS dataset name does not exist it will be allocated using the allocation parameters from the `CEERPTSIZE` parameter.
- **CEERPTSIZE:** An optional parameter that specifies the default size for the `CEERPT` output z/OS dataset.
- **SORTMSG:** An optional parameter that can be used to specify the z/OS dataset name to which the `SORTMSG` output will be written. The special value "DUMMY" can be used to cause the output to be ignored by binding it to `DUMMY`. The special value "*" can be used to cause the output to be directed to the terminal session. If an unquoted z/OS dataset name is used, it will be prefix-qualified with the `USERPFX` value. Any existing output z/OS dataset will be reused and overwritten. If the z/OS dataset name does not exist it will be allocated using the allocation parameters from the `SORTMSGSSIZE` parameter.
- **SORTMSGSSIZE:** An optional parameter that specifies the default size for the `SORTMSG` output z/OS dataset.
- **SYSPRINT:** An optional parameter that can be used to specify the z/OS dataset name to which the `SYSPRINT` output will be written. The special value "DUMMY" can be used to cause the output to be ignored by binding it to `DUMMY`. The special value "*" can be used to cause the output to be directed to the terminal session. If an unquoted z/OS dataset name is used, it will be prefix-qualified with the `USERPFX` value. Any existing output z/OS dataset will be reused and overwritten. If the z/OS dataset name does not exist it will be allocated using the allocation parameters from the `SYSPRINTSIZE` parameter.

- **SYSPRINTSIZE:** An optional parameter that specifies the default size for the `SYSPRINT` output z/OS dataset.
- **SYSOUT:** An optional parameter that can be used to specify the z/OS dataset name to which the `SYSOUT` output will be written. The special value "DUMMY" can be used to cause the output to be ignored by binding it to DUMMY. The special value "*" can be used to cause the output to be directed to the terminal session. If an unquoted z/OS dataset name is used, it will be prefix-qualified with the `USERPFX` value. Any existing output z/OS dataset will be reused and overwritten. If the z/OS dataset name does not exist it will be allocated using the allocation parameters from the `SYSOUTTSIZE` parameter.
- **SYSOUTSIZE:** An optional parameter that specifies the default size for the `SYSOUT` output z/OS dataset.
- **WORK:** An optional parameter that can be used to specify the `WORK` z/OS dataset name. If an unquoted z/OS dataset name is used, it will be prefix-qualified with the `USERPFX` value. Any existing z/OS dataset will be reused. If the z/OS dataset name does not exist, it will be allocated using the allocation parameters from the `WORKSIZE` and `WORKAP` parameters.
- **WORKSIZE:** An optional parameter that specifies the default size for the `WORK` z/OS dataset.
- **WORKAP:** An optional parameter that specifies other allocation parameters that are passed to the allocation command when the `WORK` z/OS dataset is allocated.
- **SYSIN:** The name of the WPS programme source, this parameter will be qualified with the `USERPFX` if it is not a quoted z/OS dataset name. If the parameter is not specified on the command line, then the `CLIST` will prompt the user to supply a suitable name.
- **STAMP:** When this optional switch parameter is specified, the `CLIST` will qualify all of the output z/OS dataset names with a date and time of the form `YYYYY.Dnnn.Thhmmss`, where `YYYYY` is the four digit year, `nnn` is the three digit day, and `hhmmss` represent the current hours, minutes and seconds respectively. The default is to not qualify the output z/OS dataset names.
- **NOBROWSE:** When this optional switch parameter is specified, the `CLIST` will behave, when run in TSO/ISPF, as if it was running outside of ISPF, and will not use the ISPF browse and view services to display the `SASLOG` and other z/OS datasets output by WPS.
- **DDVERBOSE:** When this optional switch parameter is specified, the `CLIST` will display more detailed output prior to invoking WPS. This additional output shows the allocated `DDNAMEs` with which WPS will be invoked.

Before Using the CLIST

Edit the CLIST

It is advisable to modify the `CLIST` to provide a default `<wpspfx>` argument value. This change is aimed at facilitating use of the `CLIST`. Subsequently, end users will not need to know the value of the argument, and will not need to specify it, other than under special circumstances.

Before modification, the first few lines of the CLIST appears like this:

```
000001 PROC 0
000002 WSPFX() /* PREFIX FOR THE WPS INSTALLATION */+
000003 USERPFX(&SYSUID) /* USER PREFIX FOR USER DATASET QUALIFICATION*/+
...
```

After modification, the code should appear similar to the following:

```
000001 PROC 0
000002 WSPFX(''WPS.<CURRENT>.INST'')
000003 USERPFX(&SYSUID) /* USER PREFIX FOR USER DATASET QUALIFICATION*/+
...
```

Note:

The amount of quote symbols is forced by CLIST syntax requirements.

Test the CLIST

Following modification, the CLIST should be tested. Firstly, make it available by either:

- Moving the CLIST to a library on your own SYSEXEC concatenation; or
- Modifying your SYSEXEC concatenation to feature <wspfx>.CLIST.

The CLIST test command should be similar to:

```
tsowps sysin(''<wspfx>.cntl(xverify)'')
```

Following successful testing, the CLIST may be moved to a system-wide SYSEXEC library to make it generally available.

Launching WPS

Important:

The WPS program requires a large amount of memory to load and run. For this reason, prospective TSO users will probably need to have their default memory size parameter value changed. This value is normally on the initial TSO LOGON panel. It represents the KB of above-the-16MB-line storage requested when logging on. WPL recommends a minimum value of 150000.

Having made the previously recommended changes, you can run WPS from TSO or TSO/ISPF using the following command:

```
tsowps sysin(''<source-code-location>'')
```

The command may be entered from any of the following:

- the command line of any ISPF panel, by prefixing the command with 'tso', for example:

```
tso tsowps sysin(''wps.v310.b29602.cntl(xverify)'')
```

- the command line provided in the ISPF Command Shell (option 6 from the ISPF Primary Option Menu); in this case, the prefixing 'tso' is optional;
- native TSO, at the 'READY' prompt. In this case the command must NOT be prefixed with 'tso'.

Once program execution is complete, the resulting files are presented in different ways, depending on whether native TSO or TSO/ISPF was used for the task:

- If TSO/ISPF was used, the user is presented with contents of the resulting `SASLOG` file in 'VIEW' mode. From here, use of the 'END' (PF3) command will result in a full list of all the `SYSOUT`-type files that have been generated. The user is then free to handle these outputs as necessary.
- If native TSO was used, the user is simply presented with the list of generated files. There is no intermediate 'VIEW' of the `SASLOG`.

Installation Verification

The `@VERIFY` member of PDS `<wpspfx>.CNTL`, supplied with WPS, contains a sample job that can be used to verify that installation of WPS has been successfully completed for z/OS.

The equivalent process should now be carried out to show that TSO installation of WPS has been successful.

From the ISPF Command Shell (Option 6 from the ISPF Primary Option Menu), enter the following command:

```
tso tsowps sysin(''wps.v310.b29602.cntl(xverify)'')
```

to run the `XVERIFY` source program. On completion, `SASLOG` output will be displayed in 'VIEW' mode. It should be checked to ensure that the program did indeed run to successful completion. Exiting from the 'VIEW' will result in a display of all the `SYSOUT` files generated by the `CLIST`. Chief among these is the file with the suffix `SASLIST`. This file should also be studied to confirm successful installation.

Appendix C - Sending files to World Programming

You may occasionally need to package data files and send them to engineers at World Programming to aid in the diagnosis and resolution of your support issues. There are numerous ways to do this, depending on the nature of the data, and this section describes some of the more common methods.

Compress a file using **TERSE**

Data libraries can be fairly large entities and they can often be compressed with the IBM-supplied TRSMMAIN program before transmission. TRSMMAIN has been included in the z/OS distribution since z/OS version 1.9, and will normally be present in your system's SYS1.LINKLIB. You will, however, need to test to establish if your site-specific privileges allow you to use it.

Note:

TRSMMAIN is often given an alias of AMATERSE.

The size of the output file is directly affected by the nature of the data in the input file, so there is no possibility of accurately forecasting how much space should be specified for it. Experience shows that TRSMMAIN delivers compression ratios of approximately 2.5:1, offering an indicative 60% space saving as a starting point.

The suggested JCL is as follows:

```
// (job statement)
//stepname EXEC PGM=TRSMMAIN,PARM=PACK
//SYSPRINT DD SYSOUT=*
//INFILE DD DSN=<input filename>,DISP=SHR
//OUTFILE DD DSN=<output filename>,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(<sufficient space for the file>),
// DSORG=PS,RECFM=FB,BLKSIZE=27648,LRECL=1024
```

PDS and PDS/E files may be used for input, as well as standard sequential files.

Note:

Tape-based files are not supported.

The LRECL (Logical Record Length) for the output file is always 1024 bytes.

Once the output file is available, it need to be uploaded to your host system in **BINARY** format. Do not specify any kind of translation or CR/LF processing.

Naturally, the compressed file must not be edited in any way prior to decompression.

Retrieve SYSOUT data from the JES2 SPOOL

Examining the SYSOUT generated from a batch execution of a WPS program is a good way to begin exploring a problem. Although such SYSOUT comprises at least four sections on the spool, we request that you send all sections to us, rather than selected parts - we would rather have too much diagnostic information than too little.

To recall a spooled SYSOUT to a file, use the xdc command in ISPF:

1. Navigate to the SDSF display for the OUTPUT, HOLD or STATUS queue and locate the desired JOB output. Enter xdc in the NP column:

```

Display Filter View Print Options Search Help
-----
SDSF HELD OUTPUT DISPLAY ALL CLASSES LINES 1,953      LINE 1-1 (1)
COMMAND INPUT ==>                                     SCROLL ==> CSR
PREFIX=ADJC* DEST=(ALL) OWNER=** SORT=JobID/A SYSNAME=
NP  JOBNAME  JobID  Owner  Prty C ODisp Dest      Tot-Rec  Tot-
xdc_ ADJCDSMF JOB06523 ADJC      144 H HOLD  LOCAL      1,953

```

2. Press ENTER and the Open Print dataset screen is displayed. Enter the name of a file to hold the SYSOUT, together with any required site-specific allocation details:

```

SDSF Open Print Data Set
COMMAND INPUT ==> _                                     SCROLL ==> CSR

Data set name ==> 'ADJC.SLOG1'
Member to use ==>
Disposition   ==> MOD      (OLD, NEW, SHR, MOD)

Management class ==>      (Blank for default management class)
Storage class   ==>      (Blank for default storage class)
Volume serial   ==>      (Blank for authorized default volume)
Device type     ==>      (Generic unit or device address)
Data class      ==>      (Blank for default data class)
Space units     ==> BLKS  (BLKS, TRKS, CYLS, BY, KB, or MB)
Primary quantity ==> 1000 (In above units)
Secondary quantity ==> 1000 (In above units)
Directory blocks ==>      (Zero for sequential data set)
Record format   ==> VBA
Record length   ==> 27994
Block size      ==> 27998
Data set name type ==>      (LIBRARY, blank, ... See Help for more)
Extended attributes ==>      (NO, OPT, or blank)

```

Note:

To upload this file to your desktop system, the transfer must be specified as a 'text' file transfer, with CR/LF characters appended to the end of each line.

Alternatively, it may be that the resulting file is very large, in which case it might be appropriate to compress it prior to transmission as described earlier.

Selecting SMF data using IFASMFDP

IFASMFDP is a standard IBM-supplied utility program for handling SMF data. It is fully described in the IBM System Management Facilities manual SA22-7630 [↗](#), but the basic JCL is as follows:

```
// (job statement)
//DUMPIT   EXEC PGM=IFASMFDP
//DUMPIN   DD   DSN=(input SMF data file),DISP=SHR
//DUMPOUT  DD   DSN=(SMF data extract),
//          DISP=(NEW,CATLG,DELETE),SPACE=(as required),
//          DCB=DUMPIN
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
INDD(DUMPIN,OPTIONS(DUMP))
OUTDD(DUMPOUT,TYPE(70:79))
DATE(yyyyddd,yyyyddd)
START(hhmm)
END(hhmm)
```

The `INDD` parameter points to the source SMF data file via that file's `DDNAME`.

The `OUTDD` parameter points to the output data file via that file's `DDNAME`. It is also used to specify the required SMF record types.

It may well be that your own input data file covers a number of days of system activity, so the `DATE` parameter can be used to specify a range of date limits to apply to the selection of the data to be output.

The `START` and `END` parameters are used to specify a time period on each of the days in the `DATE` parameter.

Once the output file is created, it must be compressed using `TERSE` (see above) before being uploaded to your desktop.

If you need to send several files to World Programming, it is perfectly acceptable to gather them into a `.zip` file before uploading.

Appendix D - WPS Load Modules and LPA Usage

During the WPS installation, the `<wpspfx>.LOAD PDSE` library is created, which contains all the WPS program load modules.

A common naming convention is used for all of the modules within this load library:

- Some modules have names formed from the 4-letter `WPSX` prefix and a unique suffix.
- Other modules have the 3-letter `WPS` prefix and a unique suffix.

WPS is built from a number of common core modules that will generally always be loaded at job startup - these are the modules with the `WPSX` prefix.

All of the other modules contain code that is loaded on demand as part of the WPS plug-in architecture to provide implementation for the functionality required for PROCs, functions and CALL routines. These modules are loaded, as needed, when the WPS program being run first requires access to the functionality contained within them.

When installing WPS at some mainframe sites, the administrator may decide to place some of the WPS load modules into the LPA (Link Pack Area). As all of the modules with the `WPSX` prefix are required when WPS is used, these should be considered as initial candidates for placing into the LPA. Other WPS load modules should be placed into the LPA, based on evidence of their usage during the routine WPS-based system workload.

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.