WPS Installation and User Guide
for z/OS

"To guide you through installing and using the World Programming System (WPS) on the z/OS platform"

world programming

Version: 3.1.8

www.worldprogramming.com

Contents

Introduction

About This Guide

Overview

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

Notation

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

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

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

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

Overview 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 license key
- Configure WPS and the environment
- Use WPS to execute programs written in the language of SAS
- Migrate existing data and code libraries

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

Legal Notices

General Acknowledgements

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

Trademarks

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

Third-Party Notices

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

Pre-Requisites

Planning for the Installation

WPS uses the USS root directory to store information during some processing. Therefore, if a user does not have an OMVS segment set up, it is highly recommended that they get a default OMVS segment defined. 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 documented in this guide, you may need to alter access rights to z/OS datasets that are used.

There are sections later in this guide concerning configuration and security issues.

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 or CD)
- Licence Key File specific to your site

Distribution Package

The WPS software is distributed either on a CD or as a single zip file. In both cases the distribution contains:

- WPS z/OS Installation File in the IBM TERSE format
- Documentation (PDF and plain text)

Obtaining the Distribution Package

For more information on how to obtain the WPS distribution package please contact World Programming via the website at www.teamwpc.co.uk or email 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.

About Package Documentation

The distribution package contains the following documentation:

Name	Type	Description
WPS Installation and User Guide for z/OS	PDF	(This document) Help to install, configure, license and execute WPS
WPS Reference for Language Elements	PDF	Option and syntax details of the supported elements in the language of SAS
SAS to WPS Migration Guide for z/OS	PDF	Help to migrate your existing programs written in the language of SAS and any data associated with them

A more detailed description about the ‘WPS Reference for Language Elements’ can be found in the *#unique_13* section of this guide.

License Key File

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

On purchase of WPS, the license key is provided in a file separate to the distribution package and will have a name similar to `YourCompanyName_v3_zos.wpskey`.

About the License Key File

The license key file is supplied in plain text. It 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 a section in this document called *Applying the WPS License Key*.

Obtaining a License Key

If you do not already have a valid license key file, please contact World Programming via the website at www.teamwpc.co.uk or email sales@teamwpc.co.uk for details on how to obtain one.

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 WPS installation file `<wpsrel>.DLIB.TER` is 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
```

This dataset should be allocated with the following attributes:

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

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

```
                          Data Set Information

Data Set Name . . . . : <wpspfx>.DLIB.TER

General Data                          Current Allocation
 Management class . . : **None**       Allocated cylinders : 500
 Storage class  . . . : SC270          Allocated extents . : 10
  Volume serial . . . : SMSP12
  Device type . . . . : 3390
 Data class . . . . . : **None**      Current Utilization
  Organization  . . . : PS             Used cylinders. . . : 0
  Record format . . . : FB             Used extents  . . . : 0
  Record length . . . : 1024
  Block size  . . . . : 27648
  1st extent cylinders: 500
  Secondary cylinders : 10
  Data set name type  :                SMS Compressible  . : NO

  Creation date . . . : 2009/09/01    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.

The following options may need to be set:

- The name of the input file on the PC will be similar to `wps-<wpsrel>-zos-s390.dlib.ter`
- The name of the receiving file on the host system will be as defined is the previous 'Allocate' section
- 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 (CR/LF, Record truncation, ASCII/EBCDIC).

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  09/22/2009
** AMA504I  RETURN CODE: 0
```

This 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

COMPONENT SUMMARY
=================

$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
INSTALL   : REXX FOR FIRST STAGE WPS INSTALL PROCESS
INSTSDK   : REXX FOR OPTIONAL INSTALL OF WPS SDK COMPONENTS
INSTUSS   : 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
XFONTS    : TRUETYPE FONTS FOR USE WITH WPS GRAPHING IN XMIT FORMAT
XLOAD     : LOAD MODULE COMPONENTS IN XMIT FORMAT
XSASHELP  : WPS SASHELP VIEW COMPONENTS IN XMIT FORMAT
XSDKCCNT  : C CNTL LIBRARY FOR WPS SDK IN XMIT FORMAT
XSDKCH    : C HEADER FILES FOR THE WPS SDK IN XMIT FORMAT
XSDKCSRC  : 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
XSETINIT  : SETINIT CODE IN XMIT FORMAT
XUSS      : OPTIONAL USS COMPONENTS IN XMIT AND PAX FORMAT

@INSTALL JOB WILL CREATE THE FOLLOWING LIBRARIES:
  <WPSPFX>.AUTOLIB : SAS AUTOCALL LIBRARY
  <WPSPFX>.CLIST   : CLIST
  <WPSPFX>.CNTL    : JCL, PROCS AND WPS SOURCE CODE
  <WPSPFX>.FONTS   : ODS FONTS
  <WPSPFX>.LOAD    : LOAD MODULES
  <WPSPFX>.SASHELP : VIEW MODULES
  <WPSPFX>.SETINIT : HOLDS COMPILED SETINIT
  <WPSPFX>.USS     : COMPONENTS FOR USS ENVIRONMENT

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

```
@INSTSDK JOB WILL CREATE THE FOLLOWING LIBRARIES:

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


==============================
(C) WORLD PROGRAMMING 2002-2019
==============================
```

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

Submit @INSTALL

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
INSTALL '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` - An empty 'flat' file that will contain encoded license 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 29 OCT 2013 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 29 OCT 2013 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 ODSCSS  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 29 OCT 2013 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 29 OCT 2013 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 VERAMOBD HAS BEEN LOADED   AND REPLACED
IGW01552I MEMBER VERAMOBI HAS BEEN LOADED   AND REPLACED
IGW01552I MEMBER VERAMOIT HAS BEEN LOADED   AND REPLACED
IGW01552I MEMBER VERAMONO 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 29 OCT 2013 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 WPSSRSLT 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 WPSUDTTM 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
INSTALL '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.V31.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.V31.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.V31.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 '<wpspfx>.CNTL'

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

INMR901I Dataset WPCUK02.WPS.V31.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 '<wpspfx>.CLIST'
```

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

INMR901I Dataset WPCUK02.WPS.V31.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 '<wpspfx>.FONTS'

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

INMR901I Dataset WPCUK02.WPS.V31.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 '<wpspfx>.LOAD'

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

INMR901I Dataset WPCUK02.WPS.V31.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 '<wpspfx>.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)
```

```
//ODSCSS   DD DISP=SHR,DSN=&WPSPFX..CNTL(ODSCSS)
//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=*
//CEEOPTS  DD DISP=SHR,DSN=&WPSPFX..CNTL(CEEOPTS)
//CEERPT   DD SYSOUT=*
//*
//* DEFINE SORT DDNAMES
//DFSPARM  DD DISP=SHR,DSN=&WPSPFX..CNTL(DFSPARM)
//SORTMSGS 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 License Key

A valid license 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 license key file. License keys are discussed in the *License Key File* section of this guide.

> **Note:** New license 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 license information, two members of the <wpspfx>.CNTL library are used.

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

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 'SETINIT' 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(SETINIT)
```

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

1. Substitute your license information in the SETINIT member described above. Be sure to include the entire file, from the 'PROC SETINIT;' statement to the final 'RUN;' statement.
2. **Note:** Site-specific security requirements may mandate that the contents of the SETINIT member are moved to a more secure location. If so, the SYSIN DD statement in the @SETINIT member will need to be changed accordingly.
3. Add a job card
4. Change the two occurrences of <wpspfx> to the required value
5. Submit the JCL

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

```
setinit applied successfully.
```

Installation Verification

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

Edit the Sample 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 the PREFIX of the WPS installation libraries that were created by the @INSTALL; job, e.g. 'WPS.V2401'.

A suitable JOB Card needs to be added.

We recommend that you specify REGION=0M on your JOB card when you submit WPSPROC 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. Extracts from the SASLOG data set or output from the example is 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:

  _LAST_=            The last data set created by a DATA step or PROC
  NOAUTOSIGNON       Remote submit will not attempt to automatically
 signon
  BASEENGINE=WPD     The library engine to use when BASE is specified
  BUFNO=1            Specifies the number of buffers used by a library
 engine for a data set (not honoured by all engines)
  BUFSIZE=0          Specifies the size of a page for a WPS data set
  BYERR              Generate an error when a null dataset is used as
 input to PROC SORT
  BYLINE             Generate a BY line title for each BY group in the
 output
  NOCAPS             Do not translate source source and data lines to
 uppercase
  CARDIMAGE          Treat CARDS lines as 80 character width punch card
 records
  CENTER             Align listing output to the center of the page
  NOCHARCODE         Do not allow character combinations as a substitute
 for special characters not on the keyboard
  COMAMID=TCP        The communication method to use for establishing
 remote connections
  COMPRESS=NO        Specifies whether to compress observations in output
 SAS data sets
  CONFIG=            List of config files processed during initialisation
  CONNECTPERSIST     The remote connection will be persisted after a
 RSUBMIT block
```

```
 CONNECTREMOTE=     Identifies the remote server that will be connected
to
 CPORTVER=SAS92     Controls which type of CPORT file is generated by
default by PROC CPORT
 DATE               Print the date and time at the top of each page of
output
 DATESTYLE=LOCALE   Controls how numerical dates are interpreted in the
ANYDT informats
 NODETAILS          Do not show additional details of data sets when
listing data libraries
 DEVICE=            Device to be used for graphical output
 DFLANG=ENGLISH     Language for EURDF date/time formats and informats
 DKRICOND=ERROR     Action for DROP/KEEP/RENAME error conditions on input
data sets
 DKROCOND=WARN      Action for DROP/KEEP/RENAME error conditions on
output data sets
 NODMR              Invokes a CONNECT local session
 DQUOTE             Toleration support for obsolete DQUOTE option
2                                                          The WPS System
                   09:34 Wednesday, October 23, 2013

 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
 PAGENO=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
  TABSIZE=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
  VALIDVARNAME=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
 DLEXCPCOUNT        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
  FILESPSEC=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
  NOISPNZTRC          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
 NOISPVDTRC          Calls to VDEFINE are not traced to the WPS log
 ISPVIMSG=           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

Please note that when referring to z/OS or MVS datasets or files in this guide, we will generally refer to them as a "z/OS dataset". The phrase "data set" will be reserved for use with WPS data sets held within a "data library", referring to the proprietary binary data set storage format used by WPS.

A WPS data library can be held within a z/OS sequential dataset, a VSAM linear dataset or a Unix System Services (HFS/ZFS) directory. Whilst using USS files offers certain advantages over z/OS sequential or VSAM linear datasets e.g. copying, renaming, transferring using Unix utilities, they are not 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 data set engine is called WPD. WPS will automatically detect whether the library in which a data set is held is held within a z/OS dataset, a VSAM LDS 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.

z/OS Sequential Dataset-Based WPS Data Libraries

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

For z/OS sequential 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 for best results.

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

Storage of a WPS data set 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 data set called "dataset1" in the z/OS dataset referred to by "mylib" can be referenced as "mylib.dataset1".

VSAM Linear Dataset-Based WPS Data Libraries

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

Initially defining the LDS 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 z-OS sequential 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 data set within the library being represented by a single file with a file extension of .wpd. Data sets can be added and removed from the library by use of file manipulation tools within USS such as cp (copy), mv (move), rm (delete) etc.

When listing the contents of a USS file-based data library, the list of members returned is the list of files within that directory with the following extensions:

- wpd = WPS Dataset
- wpccat = WPS Catalogue
- wpcvw = WPS View
- wpdidx = 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 LIBNAME 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 directory and read permissions on the .WPD file containing the WPS dataset.

WPS Data Files

WPS data sets 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* 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 file 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;
```

Multiple WPS Users and Data Sharing

Processing of data files within WPS data libraries in z/OS sequential 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 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 for input to the program.

Concurrent processing of data files within a WPS data library held within a VSAM LDS 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 within is the controlling influence.

The WORK Library

Each user of WPS should allocate their own WORK data library. These should be unique to each user and not shared between users.

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 us 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 similar to the example below:

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

WPS Configuration Options

WPS configuration options can be set via

- CONFIG files
- The OPTIONS statement
- A PARM clause on the EXEC JCL statement. (In the supplied JCL procedure, this is provided for by an OPTIONS parameter)

These methods are discussed in the sections below.

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

- 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 Files

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

> **Note:** Option names are not case sensitive, however option values may be.

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
FILESPSEC=1
FIRSTOBS=1
FMTERR
FMTSEARCH=(WORK.FORMATS)
FONTPATH=(WPSFONTS)
LINESIZE=132
LOG=SASLOG
MACRO
MAUTOSOURCE
MERROR
NOMLOGIC
NOMPRINT
MSGLEVEL=N
NEWS=NEWS
NOTES
OLDMAC
PAGENO=1
PAGESIZE=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 on the EXEC statement.

An example of an EXEC statement without a CONFIG can be found in the *Executing Programs (written in the language of SAS)* 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)
```

Where <userid>.WPSCNTL(CONFIG) is a z/OS dataset or PDS member containing a WPS CONFIG file. All WPS CONFIG options can be overridden using this file.

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:** Option names are not case sensitive, however option values may well be.

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;
```

An example of a PROC OPTIONS statement can be found in the @VERIFY sample job, see *Executing Programs (written in the language of SAS)* section of this guide.

Configuration Options for sorting data

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

- SORTSIZE - defines the amount of memory that may be used for sorting data. This option applies to both the WPS sort and whatever 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 whatever 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 whatever SORT utility is installed on the host

system is used. If BEST is specified, the decision to use HOST or WPS sort is based on the SORTCUTP option value.

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 data sets 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

Example Programs and Batch Jobs

To demonstrate the capabilities of WPS, there are some example programs together with the JCL required for executing them. These can be found in the `<wpspfx>.CNTL` library :

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

The use of these sample jobs is similar to the WPS sample job discussed in the *Executing Programs (written in the language of SAS)* section of this guide. You are advised to review the example program and JCL before submission.

Using SAS Data Libraries

WPS can read z/OS-based SAS data set libraries directly using the `SASDASD` library engine. However, WPS is unable to write to the `SASDASD` format. Therefore, persistent SAS data set libraries that get updated will need to be migrated to WPS data set libraries, which can be achieved using a simple PROC COPY.

More information about this topic can be found in a separate document called *SAS to WPS Migration Guide for z/OS*. How to obtain this and other documents is discussed in the *#unique_13* section of this guide.

World Programming is able to provide consultancy on migration of data. It is important to consider data migration before any existing environment 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*.

Using CPORT Files

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

Running Existing Programs written in the language of SAS

Many existing programs may run unaltered. Some 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 release notes for information.

Migrating Existing Programs used with WPS

Existing programs written/run with earlier versions of WPS are fully compatible with later versions.

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 information on how this can be done in the *SAS to WPS Migration Guide for z/OS*, which can be found in the *#unique_13* section of this guide

Using DB2

WPS can connect to and make use of DB2 data.

> **Note:** The lower-case 'vm' characters in 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, etc. 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 for manuals for all current z/OS versions of DB2 are given in: `http://www-01.ibm.com/support/docview.wss?uid=swg27019288`

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 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=&amp;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.SDSNLOD2
//         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 writing sequential libraries (`WPSSEQ`, `SASSEQ`, `XPORT`) to tape although limitations currently apply. 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 one "PROC DATASETS; COPY" statement that will copy datasets from multiple input libraries to be copied 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;
> ```

Using Fonts

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

Further Reading

Available Guides and Reference Material

The following seperate guides and reference material are also available:

- SAS to WPS Migration Guide for z/OS
- WPS Reference for Language Elements

SAS to WPS Migration Guide for z/OS

The *SAS to WPS Migration Guide for z/OS* will help guide you through 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, or can be freely downloaded from our web site at `www.teamwpc.co.uk/support/docs/wps`.

WPS Reference for Language Elements

The *WPS Reference for Language Elements* documents all the elements and options in the language of SAS that are supported by your installed copy of WPS. In other words, if you cannot find the language element, option or syntax in this guide, it is not yet supported.

The language reference guide is supplied as part of the distribution package in PDF format, or can be freely downloaded from our web site at `www.teamwpc.co.uk/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 how to prepare for this and how to run WPS from USS.

Install the WPS USS Software 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`.

Security Considerations

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

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

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

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>"
```

where `<wpshome>` is your WPS home directory.

Edit @INSTUSS

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
INSTUSS 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.

Execute 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 permission of 755.

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

About the USS Dataset

Once the @INSTUSS job has completed and the <wpshome> directory has been populated you may optionally delete the <wpspfx>.USS dataset as it is no longer needed for the execution of WPS.

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

Running WPS from USS

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

```
export WPSHOME=<wpshome>
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 <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 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 may be studied to prove that installation has been successfully completed.

Appendix B - Running WPS from TSO

Introduction

One function of the `<wpfpfx>.DLIB(@INSTALL)` job that was used at the start of the installation procedure populates a z/OS library name `<wpfpfx>.CLIST` with a member named TSOWPS.

This member is a CLIST that allows use of WPS in real time under native TSO 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 be left to 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. The main reason for the length of this list is the number of file allocations that need to made to enable WPS to run.

Arguments and default settings for the TSOWPS CLIST.

- WPSPFX: This string must contain the dataset prefix for the WPS installation that is to be used. Either set this on the invocation or ensure that the CLIST member specifies the appropriate default for the installation. This prefix is used to locate all of the installation located datasets. This parameter must have a value, either explicit or an implicit default, for the CLIST procedure to be able to initiate the WPS invocation.
- USERPFX: Defaults to the user's high level dataset name qualifier, this string will be used when qualifying unquoted dataset arguments. This is similar to the way that ISPF uses the user prefix for unquoted dataset names. Pass in an alternative HLQ string to be used as the initial optional qualification name.
- OUTDSNPFX: Optional parameter that if specified will be used to form the default stem name of all output datasets names 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 value and the DSQUAL value.
- DSQUAL: An optional parameter that defaults to ".WPS" that is used when OUTDSNPFX is not specified. It is combined with USERPFX to form the default stem name for all output datasets have not been 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 dataset name or 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 USERPFX value.
- ODSCSS: An optional parameter that can be used to specify a single dataset name or dataset member that will be passed into the WPS invocation as a source of output data service cascaded style sheets ahead of the installation based source. If an unquoted name is used then it will be prefix qualified with the USERPFX value.

- SASAUTOS: An optional parameter that can be used to specify a single dataset name or dataset member that will be passed into the WPS invocation as a source of data ahead of the installation based source. If an unquoted name is used then it will be prefix qualified with the USERPFX value.
- SASHELP: An optional parameter that can be used to specify a single dataset name or dataset member that will be passed into the WPS invocation as a source of data ahead of the installation based source. If an unquoted name is used then it will be prefix qualified with the USERPFX value.
- WPSFONTS: An optional parameter that can be used to specify a single dataset name that will be passed into the WPS invocation as a source of font data ahead of the installation based source. If an unquoted name is used then it will be prefix qualified with the USERPFX value.
- WPSLOAD: An optional parameter that can be used to specify a single 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 USERPFX value.
- SASLIST: An optional parameter that can be used to specify the dataset name that the SASLIST output will written to. 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 dataset name is used it will be prefix qualified with the USERPFX value. An existing output dataset will be reused and overwritten. If the 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 dataset.
- SASLOG: An optional parameter that can be used to specify the dataset name that the SASLOG output will written to. 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 dataset name is used it will be prefix qualified with the USERPFX value. An existing output dataset will be reused and overwritten. If the 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 dataset.
- WPSTRACE: An optional parameter that can be used to specify the dataset name that the WPSTRACE output will written to. 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 dataset name is used it will be prefix qualified with the USERPFX value. An existing output dataset will be reused and overwritten. If the 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 dataset.
- CEEDUMP: An optional parameter that can be used to specify the dataset name that the CEEDUMP output will written to. 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 dataset name is used it will be prefix qualified with the USERPFX value. An existing output dataset will be reused and overwritten. If the 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 dataset.
- CEERPT: An optional parameter that can be used to specify the dataset name that the CEERPT output will written to. 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 dataset name is used it will be prefix qualified with the USERPFX value. An existing output dataset will be reused and overwritten. If the 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 dataset.
- SORTMSGS: An optional parameter that can be used to specify the dataset name that the SORTMSGS output will written to. 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 dataset name is used it will be prefix qualified with the USERPFX value. An existing output dataset will be reused and overwritten. If the 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 SORTMSGS output dataset.
- SYSPRINT: An optional parameter that can be used to specify the dataset name that the SYSPRINT output will written to. 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 dataset name is used it will be prefix qualified with the USERPFX value. An existing output dataset will be reused and overwritten. If the 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 dataset.
- SYSOUT: An optional parameter that can be used to specify the dataset name that the SYSOUT output will written to. 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 dataset name is used it will be prefix qualified with the USERPFX value. An existing output dataset will be reused and overwritten. If the 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 dataset.
- WORK: An optional parameter that can be used to specify the WORK library name. If an unquoted dataset name is used it will be prefix qualified with the USERPFX value. An existing dataset will be reused. If the 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 dataset.
- WORKAP: An optional parameter that specifies other allocation parameters that are passed to the allocation command when the WORK library is allocated.
- SYSIN: The name of the WPS programme source, this parameter will be qualified with the USERPFX if it is not a quoted 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 data set names with a data and time based qualification of the form "Yyyyy.Dnnn.Thhmmss" where yyyy 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 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 output data sets generated by WPS.

- DDVERBOSE: When this optional switch parameter is specified the CLIST will display more output prior to invoking WPS. This additional output shows the allocated ddnames that WPS will be invoked with.

Before Using the CLIST

Edit CLIST

It is advisable to modify the CLIST to provide a default <wpspfx> argument value. This change is aimed at easing 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   WPSPFX()              /* 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   WPSPFX('''WPS.V301.GA''')                                    +
000003   USERPFX(&SYSUID)      /* USER PREFIX FOR USER DATASET
 QUALIFICATION*/+
...
```

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

Following such a modification, the CLIST may be tested. Make the CLIST available by either:

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

The CLIST should now be tested with a command similar to:

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

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

> **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 normally on the initial TSO LOGON panel. It represents KB of above-the-16MB-line storage requested when logging on. WPL recommend a minimum value 150000.

Running WPS from TSO

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

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

The command may be entered:

- from the command line of any ISPF panel, by prefixing the command with 'tso' e.g.:
  ```
  tso tsowps sysin('''wps.v310.b29602.cntl(xverify)''')
  ```
- from 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;
- from 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 if 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 command:

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

to run the XVERIFY source program. Upon completion, SASLOG output will be displayed in 'VIEW' mode. It should be checked to ensure that the program ran 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 a low-level-qualifier of 'SASLIST'. This file should also be studied the confirm successful installation.