

# IBM System z

## Personal Development Tool Vol. 4

### Coupling and Parallel Sysplex

Basic use of Coupling Facility  
functions with zPDT

Using the Parallel Sysplex AD-CD  
starter system

Details for building the  
starter system



Bill Ogden

# Redbooks





International Technical Support Organization

**IBM System z Personal Development Tool Vol. 4  
Coupling and Parallel Sysplex**

May 2013

**Note:** Before using this information and the product it supports, read the information in “Notices” on page v.

## **Second Edition (May 2013)**

This edition applies to the IBM System z Personal Development Tool (zPDT) Version 1 Release 4 (with updates) and the AD-CD z/OS 1.13H release.

**© Copyright International Business Machines Corporation 2012, 2013 All rights reserved.**

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

# Contents

<b>Notices</b> .....	v
Trademarks .....	vi
<b>Preface</b> .....	vii
Authors .....	vii
Now you can become a published author, too! .....	vii
Comments welcome .....	viii
Stay connected to IBM Redbooks .....	viii
<b>Chapter 1. zPDT and the Coupling Facility</b> .....	1
1.1 Introduction .....	2
1.2 Hardware .....	3
1.3 Performance .....	4
1.4 z/VM usage .....	4
<b>Chapter 2. Starter system</b> .....	5
2.1 Design and goals .....	6
2.1.1 Names .....	8
2.2 Installation .....	10
2.3 Parallel Sysplex Operation .....	11
2.3.1 IPL parameters .....	11
2.3.2 Operational notes .....	14
2.3.3 Shutdown .....	17
2.3.4 Using the base AD-CD system .....	17
2.4 Rebuilding coupling data sets .....	18
2.5 Adding 3390 volumes .....	19
2.6 Adding more 3270 terminals .....	20
2.7 Scaling up .....	21
2.8 Useful commands .....	21
2.9 Multiple users and 3270 sessions .....	22
2.9.1 Methods to connect multiple 3270 sessions .....	22
2.9.2 Multiple user configurations .....	23
<b>Chapter 3. Building the starter system</b> .....	29
3.1 Volumes and coupling data sets .....	30
3.1.1 Clone BDCD specific data sets .....	30
3.1.2 SMS changes .....	34
3.1.3 HFS and zFS changes .....	36
3.1.4 CF structure definitions .....	36
3.2 PARMLIB changes .....	39
3.2.1 IEASYSxx members .....	39
3.2.2 Clock .....	41
3.2.3 COMMNDxx members .....	41
3.2.4 Consoles .....	42
3.2.5 Couple data sets .....	43
3.2.6 SMF parameters .....	44
3.2.7 Other PARMLIB changes .....	44
3.2.8 VTAMAPPL scripts .....	45
3.3 PROCLIB changes .....	47

3.3.1 VTAMAPPL started tasks . . . . .	47
3.3.2 TCPIP procedure . . . . .	47
3.4 SYS1.IPLPARM changes . . . . .	48
3.5 TCPIP profiles and data . . . . .	48
3.6 VTAMLST changes . . . . .	50
3.7 TCPIP RESOLVER . . . . .	51
3.8 Cross reference . . . . .	52
3.9 VM directory . . . . .	53
3.10 DEVMAPs . . . . .	57
<b>Chapter 4. Frequently asked questions . . . . .</b>	<b>59</b>
<b>Related publications . . . . .</b>	<b>63</b>
IBM Redbooks . . . . .	63
Other publications . . . . .	63
How to get Redbooks . . . . .	63
Help from IBM . . . . .	63
<b>Index . . . . .</b>	<b>65</b>

# Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.*

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.


## COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

# Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

CICS®	Parallel Sysplex®	System z®
DB2®	RACF®	VTAM®
IBM®	Redbooks®	WebSphere®
IMS™	Redpaper™	z/OS®
MVS™	Redbooks (logo)  ®	z/VM®

The following terms are trademarks of other companies:

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.



# Preface

This IBM® Redbooks® publication describes the usage of Coupling Facility (CF) functions with the IBM System z® Personal Development Tool (zPDT). It describes the System z Coupling Application Developer Controlled Distribution, which is a Parallel Sysplex® “starter system” based on the AD-CD package and lists the exact steps taken to turn the normal AD-CD z/OS® system into a Parallel Sysplex base.

This document assumes that the reader is familiar with basic zPDT usage and terminology, with z/OS, with the z/OS AD-CD system, with basic z/VM® usage, and with general Parallel Sysplex concepts. It is not intended as an introduction to any of these topics.

This version of the document is based on z/VM 6.2 (as available to authorized users in an AD-CD package) and z/OS 1.13 (as available to authorized users in the January 2013 update of the AD-CD package).

## Authors

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

**Bill Ogden** is a retired Senior Technical Staff Member at the International Technical Support Organization, Poughkeepsie Center. He enjoys working with entry-level z/OS systems and has written extensively about zPDT and previous very small z/OS development platforms. Ignoring minor details such as retirement, Bill has been with the ITSO (and the predecessor ITS and WTSC organizations) since 1978.

We acknowledge considerable help from **Rich Conway**, at the Poughkeepsie ITSO, in creating our starter Parallel Sysplex system. Rich is a very important “behind the scenes” helper for many ITSO projects.

We also acknowledge considerable help from **Frank Kyne**, at the Poughkeepsie ITSO, in extending the exploitation areas of an earlier version of this starter system. Frank provides extensive Parallel Sysplex skills for many ITSO residencies, publications, and workshops.

## Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

[ibm.com/redbooks/residencies.html](http://ibm.com/redbooks/residencies.html)

## Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

[ibm.com/redbooks](http://ibm.com/redbooks)

- Send your comments in an email to:

[redbooks@us.ibm.com](mailto:redbooks@us.ibm.com)

- Mail your comments to:

IBM Corporation, International Technical Support Organization  
Dept. HYTD Mail Station P099  
2455 South Road  
Poughkeepsie, NY 12601-5400

## Stay connected to IBM Redbooks

- Find us on Facebook:

<http://www.facebook.com/IBMRedbooks>

- Follow us on Twitter:

<http://twitter.com/ibmredbooks>

- Look for us on LinkedIn:

<http://www.linkedin.com/groups?home=&gid=2130806>

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>

- Stay current on recent Redbooks publications with RSS Feeds:

<http://www.redbooks.ibm.com/rss.html>



## zPDT and the Coupling Facility

Many zPDT users have requested that IBM provide Coupling Facility (CF) functions with the System z Personal Development Tool (zPDT). The primary motivation for these requests is the desire of developers to exploit Parallel Sysplex functions within their products. Doing so requires the developer to have a Parallel Sysplex system available while developing, testing, and supporting his products. Until now, owning a Parallel Sysplex configuration has been beyond the means of many developers (either money or skills or both). IBM has provided some developers remote access to Parallel Sysplex configurations, but this is not always suitable for a variety of reasons.

The recent availability of zPDT as a small System z development base has been well accepted. This document describes the addition of Coupling Facility functions to zPDT and then describes an optional “starter” Parallel Sysplex system that may help users begin exploitation of a Parallel Sysplex environment.

This document is based on z/VM 6.2 (as available to authorized users in an AD-CD package) and z/OS 1.13 (as available to authorized users in the January 2013 update of the AD-CD package).

## 1.1 Introduction

The zPDT product has two delivery streams, based on the token that is used:

- ▶ The 1090 tokens<sup>1</sup> are used by Independent Software Vendors (ISVs) that have obtained zPDT through the IBM Partner World for Developers organization. These tokens may be used with Coupling Facility functions (under z/VM) to enable Parallel Sysplex operation.
- ▶ The same 1090 tokens are used by many IBM employees, with the same capabilities as the ISV users.
- ▶ The 1091 tokens are used by RD&T or RDzUT clients. These tokens may have optional features enabled. The use of System z Coupling Facilities is an optional (priced) feature with 1091 tokens.

This document refers to 1090 tokens, but it also applies to 1091 tokens that have Coupling Facilities enabled. Note that the 1090 version of zPDT will operate only with 1090 tokens, and the 1091 version of zPDT will operate only with 1091 tokens.<sup>2</sup>

zPDT Coupling Facilities are available only under z/VM. It is possible to run multiple z/OS guests under a single z/VM instance, in a Parallel Sysplex configuration. It is *not* possible to link several zPDT instances together (in the same or separate base machines) to create a Parallel Sysplex configuration.

For 1090 users there are no zPDT device map (devmap) statements directly involved in using Coupling Facility functions.

For 1091 users an additional parameter is required in the devmap:

```
[system]
memory 10000m
3270port 3270
cpuopt zVM_CouplingFacility
processors 3
```

As shown, for 1091 users a new parameter is required in the cpuopt statement. The parameter may be shortened to zVM\_CouplingFac or zVM\_Coupling.

In both cases, at least one CF guest machine should be defined with a z/VM directory entry such as the following:

```
USER CFCC1 CFCC1 1024M 1024M G
XAUTOLOG CFCONSOL
OPTION CFVM TODENABLE
MACH ESA
CONSOLE 009 3125 T CFCONSOL
```

All the CF guest definitions can refer to a single console guest, defined as follows:

```
USER CFCONSOL CFCONSOL 4M 4M ABCDEFG
MACH ESA
CONSOLE 009 3215 T
```

This arrangement allows the use of one console (in one 3270 session) to control multiple CF guests. The memory size for the Coupling Facility (shown as 1024M here) depends on your usage. For the very basic Parallel Sysplex described in this document, 768M is more than ample.

<sup>1</sup> It is more correct to refer to the licenses that are acquired through a token, but for brevity we simply refer to a token.

<sup>2</sup> This separation of license control is new with zPDT updates released in 2Q13.

The z/VM directory entry for each z/OS guest using a Coupling Facility must contain OPTION and SPECIAL statements such as the following:

```
USER ADCD .....  
OPTION CFUSER TODENABLE  
.....  
SPECIAL 1400 MSGP CFCC1
```

This SPECIAL statement defines a Coupling Facility channel, which is emulated by z/VM. Also, the z/VM directory MDISK definition for volumes containing z/OS coupling data sets must contain a special parameter, as follows:

```
MDISK A9F 3390 DEVNO A9E MWV  
DASDOPT WRKALLEG          <==this parameter needed for coupling data set volumes
```

## 1.2 Hardware

No special base hardware (for the Intel®-compatible computer running Linux and zPDT) or Linux release is required. However, as a practical matter, a Parallel Sysplex configuration normally requires more memory than a simple z/OS configuration under z/VM. z/VM runs in System z memory. System z memory (in a zPDT environment) is simply virtual memory created by Linux. In principle, Linux virtual memory is only loosely related to the real memory of the PC.<sup>3</sup> In practice, the real memory of the PC should be large enough to avoid paging and swapping by Linux.

As a very general recommendation, our normal zPDT documentation states that the defined System z memory should be *at least* 1 GB smaller than the real PC memory for a dedicated zPDT system. In the case of a Parallel Sysplex configuration, we suggest that 8 GB of real PC memory should be the minimum size, and more would be better.<sup>4</sup> With 8 GB of real memory, defining a 6.0 or 6.5 GB System z would be reasonable. This could be used with two z/OS guests (at 2-3 GB each, for example), two Coupling Facilities (say, 758 MB each), z/VM itself, and leave memory for zPDT device managers, a reasonable disk cache, and other functions. Additional Linux workloads would require more memory, of course, as would the use of more or larger z/OS guests.

Ignoring hipervisor situations, zPDT does not actually partition PC memory in any way. The base Linux system has complete control of PC memory; zPDT runs in Linux virtual memory, just like any other Linux application. However, zPDT can be a major user of memory and runs faster if there is little paging at the Linux level. Discussions saying, for example, that an 8 GB laptop can have a 6 GB zPDT, running two 2 GB z/OS guests plus CFs *do not* imply that 6 GB is somehow partitioned solely for zPDT usage. Rather, it is a discussion about obtaining best performance by ensuring that there are sufficient memory resources.

In principle, a 1090-L01 zPDT system (one CP) could be used on a base PC with one processor (“core”) to run a z/VM system with several Parallel Sysplex z/OS guests. In practice, this is not very practical and might result in various z/OS timeouts. For any significant use of a Parallel Sysplex system under z/VM we suggest at least a 1090-L03 model and a 4-way processor (four “cores”). The individual z/OS guests (under z/VM) might be defined with one logical CP each.

<sup>3</sup> Instead of constantly using “Intel-compatible computer” we simply use “PC” throughout this document. We assume the reader understands that this “PC” has a more specific meaning than “Personal Computer.”

<sup>4</sup> These are minimum numbers; much larger systems could be used.

## 1.3 Performance

A zPDT Parallel Sysplex system is intended for basic development, self-education, minor proof-of-concept work, small demonstrations, and so forth. It is definitely not intended for any type of production or stressful work. It should *never* be used to gauge relative performance of any software. The primary performance limitations are in these areas:

- ▶ Disk access, especially on a laptop with a single, relatively slow disk. Physical disk access is reduced by the normal Linux disk cache functions; the effectiveness of the Linux disk cache is improved if ample PC memory is available.
- ▶ Memory management, both real and virtual. Paging (by Linux, by z/VM, and by z/OS guests) must be avoided as much as possible, due mostly to disk bottlenecks.
- ▶ System z memory (as defined for zPDT) and its relationship to real hardware memory.
- ▶ System z CP processing power. This is limited by the number of PC processors available, their speed, and the maximum number of CPs that can be defined under zPDT.
- ▶ z/VM overhead under zPDT. This overhead is somewhat greater than found on a larger System z machine.

## 1.4 z/VM usage

Some z/VM skills are needed to operate z/VM. This series of documents cannot attempt to describe all aspects of z/VM usage. A brief set of “reminders” is in the third document in this zPDT series (SG24-7723) in Chapter 4.



## Starter system

In this chapter we describe the System z Coupling Application Development Controlled Distribution system. We refer to it here as the *starter system*.

Users of zPDT with Coupling Facilities can create their own Parallel Sysplex environment, of course. This can be done by starting with the AD-CD z/OS base or with any other z/OS base.<sup>1</sup> However, creating a Parallel Sysplex environment is not a trivial task—especially the first time it is attempted. For this reason we have provided a zPDT “starter system” for Parallel Sysplex operation. As the name implies, this should be viewed as a rough starting point, not as a polished product.

The current AD-CD Parallel Sysplex starter system is built on the AD-CD z/OS 1.13 release.<sup>2</sup> The base Linux was a recent SUSE distribution but other supported Linux releases could be used for the base Linux. This starter distribution consists of the following:

- ▶ Eight volumes (3390-3) for z/VM 6.2, with the appropriate directory entries for the guest z/OS systems and the Coupling Facility guest machines.
- ▶ Eight volumes (3390-3) containing the basic z/OS 1.13 AD-CD IPLable system.
- ▶ Four paging volumes (two volumes for each of the two z/OS systems in our Parallel Sysplex).
- ▶ Four additional volumes:
  - A volume containing VSAM data sets for a second z/OS guest.
  - A volume containing coupling data sets and the jobs used to build the Parallel Sysplex.
  - A work volume, which is defined as a STORAGE volume in VATLST00.
  - The SARES1 volume, which is an optional stand-alone z/OS system that is useful for repairing the main z/OS system in special cases.

Only the volumes needed to IPL z/OS 1.13 and that use basic compilers and so forth are included. Other z/OS AD-CD volumes, for DB2, WebSphere, IMS, CICS, and so forth, are not included but could be added at a later time using the normal AD-CD z/OS 1.13 distributions.

<sup>1</sup> You must, of course, have appropriate licenses for whatever System z software you use.

<sup>2</sup> To be specific, the “winter” 2012/2013 z/OS 1.13 AD-CD is used; this particular AD-CD release is sometimes known as 1.13H because the 3390 volume serials all begin with the letter H.

This starter system was built on zPDT release 45.22, otherwise known as a fixpack for the “GA4” release. It was not tested with earlier versions of zPDT and we suggest using 45.22 or later releases. In particular, the first “GA4” release (45.18) cannot be used to run z/OS under z/VM 6.2.

In this chapter we describe the general design of the system, the installation of the system, and basic operation of the system. This starter system contains IBM-licensed software products and you must be properly licensed to obtain or use it.

Readers are assumed to be familiar with zPDT. IBM Redbooks publications SG24-7721, SG24-7722, and SG24-7723 discuss zPDT concepts and usage in considerable detail.

## 2.1 Design and goals

Our target is a z/VM system with five guests in Parallel Sysplex operation:

- ▶ A z/OS system named S0W1
- ▶ A z/OS system named S0W2
- ▶ A Coupling Facility named CFCC1
- ▶ A Coupling Facility named CFCC2
- ▶ A console function for both CFCCs named CFCONSOL

A view of the complete system is shown in Figure 2-1 on page 7, which illustrates an Intel-compatible computer (a laptop, desktop, or server machine) running under Linux. Within Linux there are many processes running. zPDT is one<sup>3</sup> of these and it creates a System z environment. z/VM runs in the System z environment and several guest systems run under z/VM. The guest machines under z/VM include two Coupling Facility images (running the CFCC program) and two z/OS guest systems named S0W1 and S0W2.

All the DASD volumes used by z/VM and z/OS are contained on the PC disks. These System z DASD volumes are emulated as 3390 devices. 3270 terminal sessions are provided within Linux by 3270 emulator programs, or externally (by 3270 emulator programs) connected through Ethernet.

---

<sup>3</sup> zPDT actually consists of many Linux processes and threads, but we ignore this detail here.



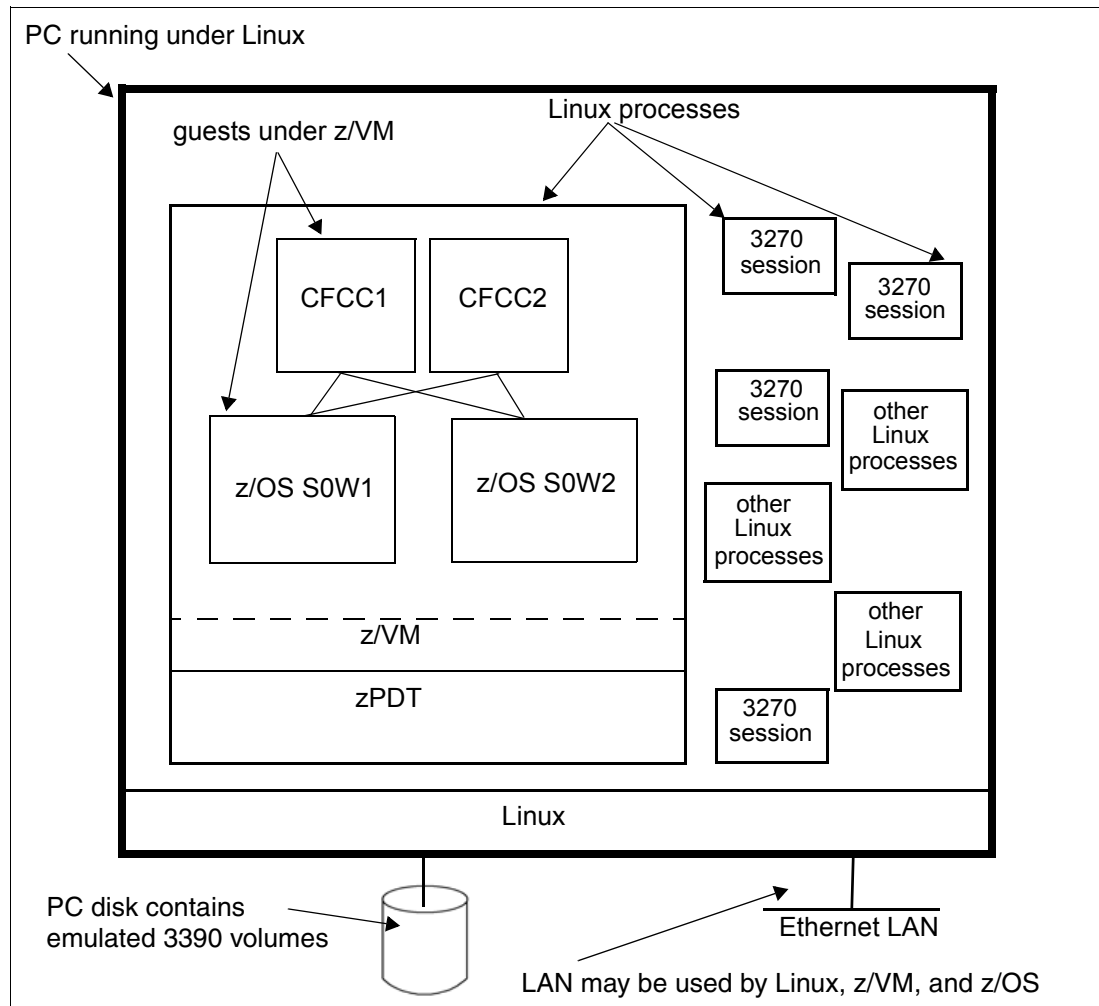


Figure 2-1 Conceptual view of a zPDT system containing a Parallel Sysplex environment

This starter system package has, in effect, three z/OS systems. One runs directly under zPDT (without using z/VM)<sup>4</sup> and the other two run only under z/VM. We have a single set of z/OS volumes, containing a modified version of the standard z/OS 1.13H AD-CD system. Our modifications allow us to IPL three different z/OS systems, all from the same volumes. This is done by selecting different IPL parameters to start each of the three systems. The three systems are:

- ▶ The standard AD-CD system, which operates as a monoplex and does not use the Coupling Facilities. It is named S0W1. This standard AD-CD is typically not used under z/VM. However, it may be used under z/VM with a guest named BASEAD. Do not attempt to run this standard AD-CD instance at the same time one of the other z/OS guests is running. It is not configured for parallel operation.
- ▶ Our first Parallel Sysplex z/OS system, which is also named S0W1. We retained the same name as the standard AD system simply to reduce the number of PARMLIB changes needed.
- ▶ Our second Parallel Sysplex z/OS system, which is named S0W2.

The z/VM guest images are named S0W1 and S0W2 (and BASEAD), the same as the z/OS names.

<sup>4</sup> It can also run under z/VM using the BASEAD guest definition.

We provide a fairly minimal Parallel Sysplex system. It contains the following sysplex-enabled functions:

- ▶ Shared MVS™ consoles (messages from all z/OS systems appear on all MVS consoles)
- ▶ MAS JES2 (“shared spool”).
- ▶ GRS (providing consistent enqueue functions across the sysplex).
- ▶ Shared USS file systems.
- ▶ CF log streams are used for OPERLOG, LOGREC, and SMF data. These functions may be redirected to their traditional data sets.
- ▶ Nonconflicting definitions are used for VTAM on the two z/OS systems.

We do not include any sysplex-related functions for TCP/IP, DB2®, CICS®, VSAM RLS, and so forth. These are left as exercises for you.

Our base AD-CD system (run without z/VM or under z/VM guest BASEAD) is the standard AD-CD z/OS 1.13H system with the following modifications:

- ▶ JES2 is in MAS (“shared spool”) mode. This should be transparent for most usage.
- ▶ The TCPIP parameter and data member names (in ADCD.Z113H.TCPPARMS) are special and should not be changed. The names are determined by symbolic names in the TCPIP JCL procedure. You may change the parameters, but do not change the member names.
- ▶ We removed the NFS definition from PARMLIB member BPXPRMCS because it was not being used. The removed definition was:

```
FILESYSTYPE TYPE(NFS) ENTRYPPOINT(GFSCINIT) ASNAME(NFSC)
```

You may restore it to the BPXPRMCS member if you want to use NFS.

- ▶ The IEBCOPY program distributed with z/OS 1.13 was renamed to IEBCOPYX and the distributed IEBCOPYO program was renamed to IEBCOPY. This was done to resolve a problem with the distributed IEBCOPY program. (Note that PTF UA67459, which is intended to correct the IEBCOPY problem, is not installed.)

A considerable number of emulated 3390 volumes are used, as noted in Figure 2-2 on page 9. All except CF0001, WORK01, and HDSYS2 are based on the normal z/OS AD-CD volumes or z/VM volumes.

## 2.1.1 Names

The z/OS 1.13H AD-CD used the system name S0W1.<sup>5</sup> To avoid changing more PARMLIB (and other) parameters, we retained this name for the base z/OS and the first sysplex z/OS. We then created name S0W2 for the second z/OS. These are not ideal names and are not compatible with earlier versions of this zPDT Parallel Sysplex starter system and with PARMLIB naming conventions we used earlier. Table 2-1 may help if there is confusion.

Table 2-1 PARMLIB naming conventions

z/OS name	PARMLIB suffix begins with	IPL parameter	OMVS data sets	Previous releases
S0W1	X	X1, X2	ADCD113H	ADCD
S0W2	Y	Y1, Y2	BDCD113H	BDCD

<sup>5</sup> We do not know how the S0W1 name was chosen or assigned.

As noted in Figure 2-2, there is a single IPL volume for z/OS. Different IPL parameters are used to initialize the three different z/OS systems we used (stand-alone S0W1, sysplex S0W1, and sysplex S0W2). The small numbers in Figure 2-2 are the addresses (device numbers) we used in the zPDT devmap and in this documentation.

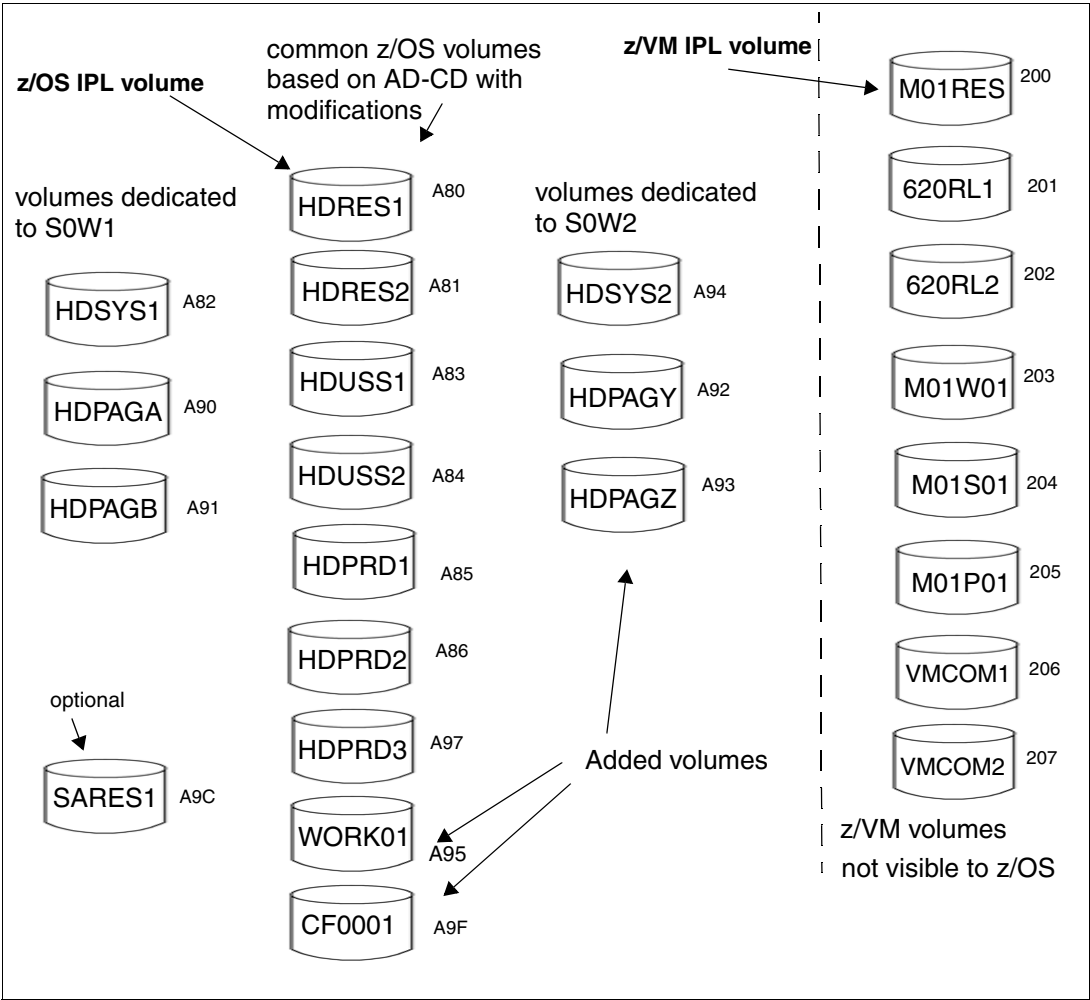


Figure 2-2 3390 volumes

### 2.1.2 LAN and TCP/IP configuration

Our distributed system defines the configuration shown in Table 2-2.

Table 2-2 IP addresses

	Ethernet	Tunnel
Base Linux	192.168.0.60	10.1.1.1
S0W1	192.168.0.61	10.1.1.2
S0W2	192.168.0.63	10.1.1.3

These are local, static IP addresses, including the Linux address. The LAN addresses are for a normal Ethernet adapter. The tunnel addresses are for communication between z/OS and Linux (and may be used for z/OS to z/OS communication, as well.) The S0W1 and S0W2 IP addresses are set by the relevant PROFILE definitions<sup>6</sup>. The Linux Ethernet address is set by

standard Linux definition functions and need not be the specific address shown in the table. The Linux tunnel address is set (or defaulted) in the devmap.

There is no provision for DHCP addresses. You could change the Linux IP address to DHCP, but the remaining addresses must be static. The standard z/OS TCP/IP does not operate as a DHCP client. Remember that 192.168.xxx.xxx and 10.x.x.x IP addresses cannot be used on external LANs. These addresses are only for local, private LANs. You may, of course, change these to IP addresses suitable for your LAN.

Our VM guest definitions (in conjunction with z/OS VTAMLST definitions) allow z/OS to directly use the two OSA adapters we define in the devmap. We elected not to use VSWITCH functions. We did not configure z/VM TCP/IP. Any of these configurations can be changed by you. However, we suggest you initially try our configuration for assurance that our TCP/IP definitions are operational.

Figure 2-3 illustrates the LAN configuration, as distributed.

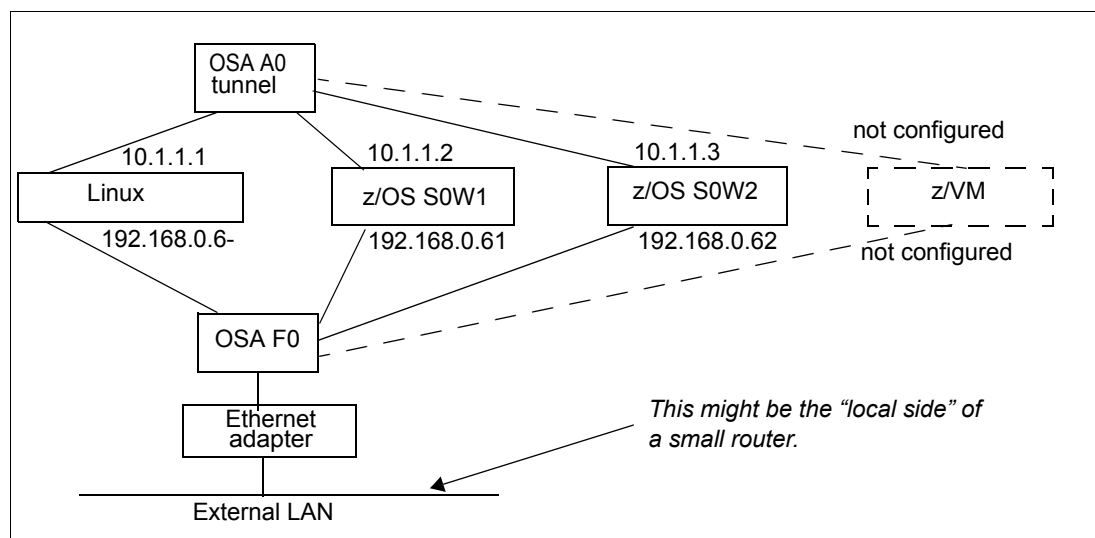


Figure 2-3 LAN configuration

## 2.2 Installation

Our starter system is packaged on two DVDs: PLEXDVD1 and PLEXDVD2. An alternative arrangement may be to download the files. In this case, there may be two .iso DVD images, or a considerable number of individual files (one for each volume in Figure 2-2) must be downloaded. Files are split across PLEXDVD1 and PLEXDVD2 so as to place approximately the same amount of data on each DVD.

The 3390 volumes, shown in Figure 2-2, are as follows:

- ▶ HDRES1, HDRES2, HDSYS1, HDUSS1, HDUSS2, HDPRD1, HDPRD2, HDPRD3, HDPAGA, HDPAGB are volumes distributed with the z/OS 1.13 AD-CD released in January 2013. Some modifications have been made to ADCD PARMLIB, PROCLIB, and other configuration files.
- ▶ HDSYS2, HDPAGY, HDPAGZ were created for the second z/OS system (S0W2).
- ▶ WORK01 is simply an empty work volume that is mounted as STORAGE in VATLST00.

<sup>6</sup> The z/OS TCP/IP profiles are members S0W1P and S0W2P in ADCD.Z113H.TCPPARMS. Do not change these member names!

- ▶ CF000 contains a library with the jobs used to construct the starter system. It also contains the coupling data sets used for Parallel Sysplex operation.
- ▶ The z/VM volumes are from the z/VM 6.2 AD-CD release, with modifications to the directory to define the various z/VM guests needed for the Parallel Sysplex.

Our examples assume the emulated 3390 volumes will be placed in the Linux /z directory. This is not required, but the distributed unzip scripts and the devmap make this assumption.

The unzip1 and unzip2 scripts may be executed to install the volumes or you may use individual Linux commands to install individual volumes. For example,<sup>7</sup>

```
$ cd /z
$ gunzip -c /run/media/ibmsys1/PLEXDVD1/M01RES.gz > /z/6M01RES
etc
$ cd /home/ibmsys1
$ cp /run/media/ibmsys1/PLEXDVD1/devmap1 devmap1
```

*You must examine the supplied devmap before using it, making any changes required by your configuration. You might want to move it to your home directory. If you did not restore the optional SARES1 volume, you could remove it from the devmap. Adjust the number of processors and System z memory, as needed. Change the LAN parameters as needed for your environment.*

### 2.2.1 Special notes

The distributed z/VM does not have any LAN or TCP/IP customization for z/VM itself. You may do this customization if you need LAN access from z/VM. (The z/OS guests under z/VM have LAN and TCP/IP definitions and access OSA interfaces directly.)

However, we strongly suggest you defer any extensions (such as TCP/IP for z/VM, or more processors for z/VM guests) until you have successfully run the system as it is distributed.

## 2.3 Parallel Sysplex Operation

Parallel Sysplex operation is more complex than the operation of a single z/OS system. There are many ways to organize the operation. The instructions we provide here are as basic as possible and are intended as a starting point. As usual, we suggest that you do it our way for your initial installation. After the installation is complete and operational, you can be more creative in your procedures.

### 2.3.1 IPL parameters

We IPL the AD-CD z/OS systems with the command `ipl a80 parm 0a82xx` (or `ipl a80 loadparm 0a82xx` when under z/VM). The following discussion about IPL parameters refers specifically to the two “xx” characters in these commands. We use the following IPL parameters with this system:

PARAMETER	Purpose
<b>CS</b>	Cold start non-sysplex SOW1 (CLPA, format JES2)
<b>00</b>	Warm start non-sysplex SOW1 (no CLPA, warm JES2)

(All the additional IPL parameters provided with the normal AD-CD system may also be used when not in the Parallel Sysplex configuration.)

<sup>7</sup> The examples assume you are logged on as ibmsys1. The “newer” Linux releases contain your Linux userid as part of the path name to access external devices.

<b>X1</b>	Cold start sysplex S0W1 (CLPA, cold JES2) (under z/VM)
<b>X2</b>	Warm start sysplex S0W1 (no CLPA, warm JES2) (under z/VM)
<b>Y1</b>	Cold start sysplex S0W2 (CLPA, warm JES2) (under z/VM)
<b>Y2</b>	Warm start sysplex S0W2 (no CLPA, warm JES2) (under z/VM)

Note that all six of these IPL options are for a z/OS system without DB2, CICS, and so forth. The non-sysplex S0W1 can also be IPLed with any of the normal AD system parameters (such as DC or DB) to include libraries for DB2 and other products.<sup>8</sup> The system user can create additional IPL sequences for sysplex operation to include these additional libraries; this is left as an exercise for you.

At least five 3270 sessions are needed for the most basic use of the Parallel Sysplex and it is quite easy to enter commands in the wrong 3270 session. We suggest two Linux desktop workspaces with the arrangement shown in Figure 2-4. This arrangement assumes that two gnome workspaces exist, wherein you can switch between them.<sup>9</sup> The 3270 windows related to the S0W2 system are in the second workspace, which helps prevent confusion when using the multiple 3270 windows. The addresses shown in Figure 2-4 correspond to the 3270 device addresses in the devmap on our DVD.

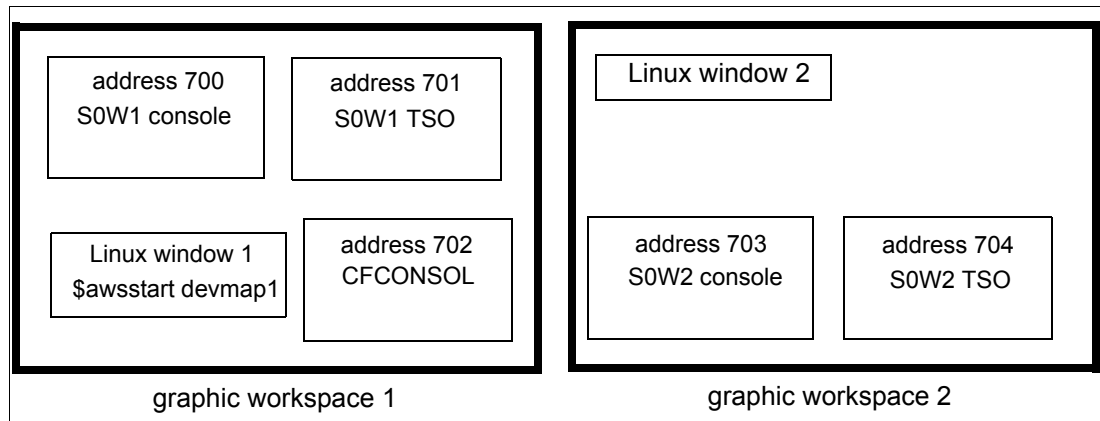


Figure 2-4 Typical display management

System startup for the Parallel Sysplex operation can be as follows (the parenthesis at the beginning of each of the following lines indicate which window is used for the command).

**Important:** The initial password for IBMUSER and ADCDMST (on both the S0W1 and S0W2 systems) is *syslab*. The z/VM passwords are the same as the z/VM userid. For example, the S0W1 login for z/VM uses the password S0W1.

```
(Linux 1) $ awsstart devmap1 (start zPDT)
(Linux 1) $ x3270 -port 3270 localhost & (starts 3270 as 700)
(Linux 1) $ x3270 -port 3270 localhost & (starts 3270 as 701)
(Linux 1) $ x3270 -port 3270 localhost & (starts 3270 as 702)
(Linux 2) $ x3270 -port 3270 localhost & (starts 3270 as 703)
(Linux 2) $ x3270 -port 3270 localhost & (starts 3270 as 704)
(Position the windows as shown in the figure)
(Check for zPDT "License Obtained" messages)
(Linux 1) $ ip1 0200 parm 0700 (IPL z/VM)
(See "Operational notes" below about the z/VM stand-alone loader.)
```

<sup>8</sup> To do this, you must install additional AD-CD volumes from the AD-CD z/OS 1.13 release to include the libraries for whatever additional subsystems that are implied by your IPL parameter.

<sup>9</sup> Another option is to use a separate PC for the 3270 sessions related to the S0W2 system.

(Once z/VM starts you may need to clear screen several times until normal z/VM startup is complete.)

```

(700)  DISC                                (disconnect VM operator)

(702)  logon as user CFCONSOL              (password CFCONSOL)
(702)  XAUTOLOG CFCC1                     (start first CF)
(702)  XAUTOLOG CFCC2                     (start second CF)
(702)  (Clear screen, as needed)

(700)  logon to z/VM as user SOW1          (password SOW1)
(700)  TERM CONMODE 3270
(700)  ip1 0a80 loadparm 0a82x1           (IPL SOW1 system)
                                           (Wait for z/OS messages. IPLing z/OS under z/VM is
                                           slower than when running natively under zPDT.
(700)  (Respond, as usual, to z/OS messages)
                                           (Wait until z/OS startup processing is complete)
(701)  Command==> DIAL SOW1               (connect for TSO session)
(701)  (The VTAM logon screen should appear)
(701)) logon ibmuser                      (password syslab) (logon to TSO)

(703)  logon to z/VM as user SOW2          (password SOW2)
(703)  TERM CONMODE 3270
(703)  ip1 0a80 loadparm 0a82y1           (IPL SOW2 system)
                                           (Wait for z/OS messages.
(703)  (Respond, as usual, to z/OS messages)
                                           (Wait until z/OS startup is complete)
(704)  Command==> DIAL SOW2               (connect for TSO session)
(704)  (The VTAM logon screen should appear)
(704)  logon adcdmst                      (password syslab) (logon to TSO)
                                           (Do not logon as IBMUSER !)
```

*Do not* log on to both z/OS systems (at the same time) as IBMUSER! With the default ISPF configuration, this will cause the ISPF profile data set to be uncataloged.

Always shut down the z/OS systems correctly. The next z/OS startup can be difficult if this is not done. The X1 and Y1 IPL parameters are for cold starts. Subsequent startups could use the X2 and Y2 parameters.

## Startup comments

Depending on your exact z/VM configuration, you may see the Stand Alone Program Loader when you IPL z/VM, as shown in Figure 2-5 on page 14.

```

STAND ALONE PROGRAM LOADER: z/VM VERSION x RELEASE 6.1

DEVICE NUMBER: 0200  MINIDISK OFFSET: 000000000  EXTENT: 1

MODULE NAME:  CPLOAD  LOAD ORIGIN: 10000

----- IPL PARAMETERS -----
fn=system ft=config pdnum=1 pdvol=0206 cons=0700
.
.
.
.
9= FILELIST 10= LOAD 11= TOGGLE EXTENT/OFFSET

```

Figure 2-5 Stand Alone Loader panel

If the pdvol address (for the VMCOM1 volume) and the cons address (for the operator 3270 session) are correct, simply press PF10. You may overwrite the IPL parameters, if necessary.

If you sometimes use the basic S0W1 system (not with the Xn or Yn IPL parameters), you may see unusual messages either during the basic startup or when you return to a sysplex startup.

```

ILR030A PAGE DATA SET MAY BE IN USE
ILR031A REPLY 'U' TO PREVENT ACCESS, 'CONTINUE' TO ALLOW USE OF
SYS1.....
r 00,continue

ILR003I WARM START FOR VIO FAILED
01 ILR003A REPLY 'U' TO INVOKE CVIO PROCESSING FOR WARM START
r 01,u

01 IXC405D REPLY I TO INITIALIZE THE SYSPLEX, J TO JOIN SYSPLEX ADCDPL,
OR R TO REINITIALIZE XCF
r 01,i      (usual reply; message occurs after some sysplex crashes)

```

Do not attempt to run the basic AD-CD system<sup>10</sup> under the S0W1 or S0W2 guest userids in z/VM.<sup>11</sup> These userids (S0W1, S0W2) assume that the Coupling Facilities are present and the basic AD-CD system parameters are not appropriate for this. Likewise, do not try to run the S0W1 or S0W2 systems (with IPL parameters of Xn or Yn) without z/VM and the Coupling Facilities; their parameters are not appropriate for this.

## 2.3.2 Operational notes

We repeat the warning: do not log on to both z/OS systems (at the same time) with the same userid, such as IBMUSER.

After a z/OS system is started (and the routine startups associated with IPL are completed) we suggest you always issue a D XCF,STR command:

```

D XCF,STR
STRNAME          ALLOCATION TIME      STATUS          TYPE
IFASMF_GENERAL   05/21/3012 10:11:16  ALLOCATED        LIST
ISGLOCK          05/21/3012 10:11:16  ALLOCATED        LOCK

```

<sup>10</sup> That is, with any IPL parameter other than X1, X2, Y1, or Y2.

<sup>11</sup> You may run the basic z/OS system under the z/VM guest named BASEAD.



IXC_DEFAULT_1	05/21/3012 10:11:16	ALLOCATED	LIST
IXC_DEFAULT_2	05/21/3012 10:11:16	ALLOCATED	LIST
IXC_DEFAULT_3	05/21/3012 10:11:16	ALLOCATED	LIST
RRS_DELAYEDUR_1	05/21/3012 10:11:16	ALLOCATED	LIST
RRS_MAINUR_1	05/21/3012 10:11:16	ALLOCATED	LIST
RRS_RESTART_1	05/21/3012 10:11:16	ALLOCATED	LIST
RRS_RMDATA_1	05/21/3012 10:11:16	ALLOCATED	LIST
SYSTEM_LOGREC	05/21/3012 10:11:16	ALLOCATED	LIST
SYSTEM_OPERLOG	05/21/3012 10:11:16	ALLOCATED	LIST
SYSZWLM_WORKUNIT	05/21/3012 10:11:16	ALLOCATED	CACHE

If all these structures are not ALLOCATED then you have a problem that should be investigated.

Parallel Sysplex operation provides a unified MVS console, in which messages from both z/OS systems appear on both MVS consoles. You can operate from a single console (such as the SOW1 MVS console) by using the ROUTE command (RO) when you want to send commands to the other system. For example, assuming you are using the SOW1 console:

P TCPIP	(stop TCPIP on this system, SOW1)
RO SOW2,P TCPIP	(stop TCPIP on SOW2)
S TCPIP	(start TCPIP on this system, SOW1)
RO SOW2,S TCPIP	(start TCPIP on SOW2)

As distributed, both the SYSLOG and the OPERLOG are active. When in SDSF you may use the commands LOG S to see the SYSLOG and LOG O to see the OPERLOG. A SYSLOG records console activity only for that z/OS system. An OPERLOG contains console activity for all the z/OS systems in the sysplex and may contain more detail than SYSLOG. OPERLOG data is recorded in LOGGER streams.

As distributed, SMF recording is active and goes to LOGGER streams. If you want to use the SMF data, remember that a different program—the new SMF dump program for logger is IFASMF DL—is used to read SMF data from LOGGER streams.

As distributed, LOGREC data goes to a LOGGER stream. If you need to read it, a different version of the EREP program is required.

These LOGGER streams are defined with retention values (expressed in the number of days to retain the data). When the retention period expires, the data is discarded. The retention period is in the definitions in job LATER6 in data set OGDEN.SYSPLEX.CNTL on the CF0001 volume. The retention period is set to seven days for SMF, LOGREC, and OPERLOG.

**Attention:** The default SMF parameters for the z/OS AD-CD system produce considerable output. Logstreams in the Coupling Facility are offloaded to DASD, using data set names such as IXGLOGR.IFASMF.GENERAL.A0000000. With our starter parameters these data sets are retained for seven days. If your system has a number of users running a variety of functions, the number of SMF logstream offload data sets can build up rather quickly throughout a week. If you have no use for SMF data, we suggest you change the SMF control parameter to NOACTIVE for both z/OS systems. We default to active SMF collection in a data stream in order to illustrate how it is done.

An alternative is to process the SMF data and delete it (in the logstream) after the processing.

## SMF logstreams

The SMF logstream, as defined, uses a staging data set. This helps ensure that continuity is not lost across system shutdowns and startups. However, while experimenting with Parallel Sysplex it is possible to cause data loss in the SMF logstreams. For example, this might happen if you switch to SMF data set recording for a while, or stop all SMF recording for a while. If the LOGGER determines that there is missing data in the SMF logstream (typically, this is detected during IPL processing) SMF will refuse to use the logstream and store current data in buffers. This is not good because the buffers eventually fill and more data is lost.

In a production system, lost SMF data is often considered a major problem. In our small starter system we usually are not concerned about the lost data. We simply want to start using the SMF logstream again. Doing this is a bit more complex than it might first appear.

You can use the D XCF,STR command to determine whether the IFASMF\_GENERAL structure is allocated; if so, this implies that logstream operation is normal. You can use the D SMF command to ask how SMF is logging. Our goal is to have it going to logstream IFASMF.GENERAL.

If SMF data is going to buffers (implying that something is wrong with the logstream) you can issue the commands SET SMF=X3 (on the S0W1 system) and SET SMF=Y3 (on the S0W2 system). These PARMLIB members turn off SMF recording.<sup>12</sup> (Again, turning off SMF recording is not something you would normally do in a production system, but we assume that losing some SMF data is not a serious problem in our little starter system.) After doing this, you might inspect and run job SMFLOGD1 in data set OGDEN.SYSPLEX.CNTL on CF0001. This job deletes SMF logstream IFASMF.GENERAL and creates a new (clean) one with the same name. Once this is done, the commands SET SMF=X1 (on the S0W1 system) and SET SMF=Y1 (on the S0W2 system) will start SMF recording in the logstreams again.

```
//SMFDLOG1 JOB OGDEN,MSGCLASS=X
// EXEC PGM=IXCMIAPU
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DATA TYPE(LOGR)

DELETE LOGSTREAM NAME(IFASMF.GENERAL)

DEFINE LOGSTREAM NAME(IFA.GENERAL)
STRUCTNAME(IFASMF_GENERAL)
LS_DATACLAS(LOGR4K)
HLQ(IXGLOGR)
LS_SIZE(1024)
LOWOFFLOAD(0)
HIGHOFFLOAD(80)
STG_DUPLEX(YES)
RETPD(7)
AUTODELETE(YES)
/*
```

All of the offloaded (to DASD) logstreams have high-level qualifier IXGLOGR. They are placed on STORAGE volumes (HDSYS1, HDSYS2, and WORK01) and are automatically deleted when the retention period (seven days) expires for records in the various offloaded logstreams. Under normal circumstances you should not manually delete these data sets.

<sup>12</sup> The various PARMLIB members discussed in this document exist only in our small starter sysplex system. They are not “standard” z/OS PARMLIB members.

### 2.3.3 Shutdown

Proper shutdown is important to prevent difficulties when restarting a Parallel Sysplex system. One way to shut down these systems is:

```
logoff of all TSO sessions
(703) S SHUTY (start shutdown script for SOW2)
      (wait for the ALL AVAILABLE FUNCTIONS COMPLETE message)
(703) $PJES2
      (wait for the JES2 ENDED message)
(700) S SHUTX (start shutdown script for SOW1)
      (wait for the ALL AVAILABLE FUNCTIONS COMPLETE message)
(700) $PJES2
      (wait for the JES2 ENDED message)
(700) V XCF,SOW2,OFFLINE
nn IXC371D CONFIRM REQUEST TO VARY SYSTEM SOW2 OFFLINE, REPLY
SYSNAME=SOW2 TO REMOVE BDCD OR C TO CANCEL
(700) nn,SYSNAME=SOW2
nn IXC102A XCF IS WAITING FOR SYSTEM BDCD DEACTIVATION. REPLY DOWN
WHEN MVS ON BDCD HAS BEEN SYSTEM RESET
      (If you simply wait for a few seconds z/VM will end BDCD and there is
      no need to QUIESCE and DOWN.)
(703) QUIESCE (This kills SOW2)13
(700) nn,DOWN (not needed if you wait for VM to stop SOW2)
      (wait for console activity to stop; usually after a CONSOLE
      PARTITION CLEANUP COMPLETE message.)
(700) V XCF,SOW1,OFFLINE (this allows PDSE, etc, to stop cleanly)
(700) nn,SYSNAME=SOW1
(700) QUIESCE
(CFCONSOL) FORCE CFCC1
(CFCONSOL) FORCE CFCC2
(CFCONSOL) SHUTDOWN (shutdown z/VM)
(Linux 1) $ awsstop
```

The shutdown process can be a bit messy. The V XCF OFFLINE function and the QUIESCE functions typically result in a message from z/VM detailing a disabled wait state for the z/OS guest. This is normal.

The V XCF,SOW2,OFFLINE command may produce the following message:

```
BPXI070E USE SETOMVS ON ANOTHER SYSTEM TO MOVE NEEDED FILE SYSTEMS, THEN REPLY
WITH ANY KEY TO CONTINUE SHUTDOWN.
```

We simply reply Y to this message.

### 2.3.4 Using the base AD-CD system

The base z/OS system (without Parallel Sysplex functions) may be used, but not at the same time as the Parallel Sysplex systems. The base can be IPLed in a simple zPDT environment (without z/VM) in the usual manner.<sup>14</sup> It can also be IPLed under z/VM using the BASEAD guest definition, as follows:

<sup>13</sup> The QUIESCE and DOWN commands may not be needed. After you vary XCF offline to SOW2, z/VM will eventually detect that the CF links are disconnected and place SOW2 in a wait state. Likewise for the XCF offline for SOW1.

<sup>14</sup> See the full IBM Redbooks publications for more information. These are publication numbers SG24-7721, SG24-7722, and SG24-7723.

1. Start zPDT and several 3270 sessions in the normal manner.
2. IPL z/VM (zPDT command **ip1 200 parm 0700**).
3. Manage z/VM startup in the normal manner.
4. Log on to z/VM guest BASEAD (the password is BASEAD).
5. Enter a TERM CONMODE 3270 command.
6. Enter an **ip1 A80 LOADPARM 0A8200** command (using whatever IPL parameter is appropriate for your tasks.)
7. After z/OS is up, enter a DIAL BASEAD command on another 3270 session.

## 2.4 Rebuilding coupling data sets

**Important:** The steps we describe here (deleting and recreating the coupling data sets in order to change structure definitions) are not likely to be acceptable on a larger system.

In a production Parallel Sysplex operation, changes to the coupling data sets are usually done by changing the alternate coupling data set, switching processing to the alternate, changing the primary data set, and then switching back to the primary. However, on our small system we often cheat and use an easier and more primitive method. Our excuse for doing this is that we want to always have the coupling data set allocation jobs reflect the latest changes, and we often have a number of changes we want to make. The jobs are in data set OGDEN.SYSPLEX.CNTL on volume CF0001. The process is as follows:

1. Shut down the Parallel Sysplex and z/VM.<sup>15</sup>
2. IPL the basic AD-CD system (not under z/VM), with an IPL parameter such as 0A82CS.
3. Using ISPF option 3.4, delete all the coupling data sets on volume CF0001. These data sets have names beginning with SYSPLEX.ADCDPL. (Do *not* delete the coupling data sets used by the basic AD system. These have names beginning with SYS1.ADCDPL and are on volume HDRES2.)  
  
You may also want to delete the logger offload data sets, as noted in “Logger data sets” on page 19.
4. Change whatever parameters you want in the coupling definitions in jobs COUPLE2, CFRMPOL3, and LOGS6.
5. Run job COUPLE2 to create new coupling data sets and verify that the job has return code zero. Run job CFRMPOL3 to create the basic policy and structures and verify return code zeros.
6. Change the GRS parameter in data set ADCD.Z113H.PARMLIB member IEASYSX1 from GRS=STAR to GRS=TRYJOIN. Also, *remember the comma after the parameter.*
7. Do any other functions that are easier in an unshared system, such as compressing the AD-CD PARMLIB. (You may need to issue F OMVS,STOPPFS=ZFS to compress PARMLIB.)
8. Shut down the basic AD-CD system
9. Start z/VM, the two Coupling Facilities, and IPL the S0W1 system (with IPL parameter 0A82X1) as described earlier.

<sup>15</sup> This could also be done using the BASEAD guest under z/VM. When using this approach, it is better to terminate and restart the CF guests as part of the process.

There will be a number of unexpected error messages in the log. These occur because the various logstreams are not defined yet. They will be added by the LATER6 job that is run later.

10. Once MVS command processing is available, enter these two commands:

```
SETXCF START,POLICY,POLNAME=CFRM01,TYPE=CFRM
SETGRS MODE=STAR
```

11. Run job LATER6 to complete various LOGR definitions. (Remember to use the z/VM command DIAL S0W1 to access a VTAM session.) This job is in data set OGDEN.SYSPLEX.CNTL on volume CF0001.

12. Edit PARMLIB member IEASYSX1 again and change GRS=TRYJOIN to GRS=STAR.

At this point the additional Coupling Facility structures are available, but your S0W1 system is not using the ones that were added by the LATER6 job. You can use the structures immediately by issuing the following MVS commands:

```
V OPERLOG,HARDCPY          (starts OPERLOG operation)
SETLOGRC LOGSTREAM         (moves LOGREC to the log stream)
SET SMF=X1                 (reinitialize SMF; will use log stream)
```

The next time you IPL (with IPL parameter X1 or X2), these log streams will be used automatically.

## Logger data sets

Be aware that the coupling data sets may contain data that is needed across IPLs. The most obvious case involves LOGGER data sets and their retention periods. In normal usage, LOGGER automatically deletes its offloaded data sets when the retention period for all the records in the data set has expired. If you delete and recreate the coupling data sets this retention period information for existing LOGGER data sets is lost and the data sets will remain on your disks until someone explicitly deletes them. These are VSAM data sets and are slightly more difficult to delete than simple sequential or partitioned data sets. The LOGGER data sets have the high level qualifier IXGLOGR and have full names such as:

```
IXGLOGR.IFASMF.GENERAL.A0000000          <==base cluster name
IXGLOGR.IFASMF.GENERAL.A0000000.DATA
IXGLOGR.SYSPLEX.LOGREC.ALLRECS.A0000000  <==base cluster name
IXGLOGR.SYSPLEX.LOGREC.ALLRECS.A0000000.D
```

The A0000000 qualifiers in these examples are for the first LOGGER data set in each group. This number is incremented when new LOGGER data sets are created.

z/OS 1.13 provides a convenient extension to the “traditional” ISPF 3.4 function. You can use a D line command against the base cluster name of a VSAM data set to delete it.<sup>16</sup> This takes you through two additional ISPF panels, but is more convenient than other methods of deleting VSAM data sets.

## 2.5 Adding 3390 volumes

You may want to add more emulated 3390 volumes to one or both z/OS systems. The distributed z/VM with this starter system has directory definitions for z/OS 3390 volumes at addresses A80 - ABF; these may be sufficient for your additional volumes.

<sup>16</sup> The base cluster name is the one without the .DATA or .D in the name.

In general, proceed as follows:

1. Create (or restore) the volumes. Creating a new volume involves use of the zPDT **a1cckd** command.
2. Add the new volume to the devmap. Devmap devmap1 is used in our examples. Use an address in the A80-ABF range if possible. If this is not possible, you need to modify the z/VM directory, as follows:
  - a. Start zPDT (**awsstart devmap1**) and IPL z/VM (**ip1 200 parm 0700**).
  - b. Log on to user maint (password is maint).
  - c. Edit the VM directory (**xedit user direct c**).
  - d. Scroll to the section beginning with userid MVSDUMMY.
  - e. Using the existing patterns for userids MVSDUMMY, BASEAD, S0W1, and S0W2, add additional lines for your new volumes, if needed. Add these lines to all four userids.
  - f. Exit from Xedit (**file**).
  - g. Rebuild the active z/VM directory (**directxa user direct c**).
3. Start one of the z/OS systems. If you created new 3390 volumes (as opposed to restoring the volumes from a DVD), you will need to submit an ICKDSF job from TSO to initialize the volumes. (z/OS automatically varies offline any uninitialized volumes it finds during IPL. After you initialize the volumes with ICKDSF you can vary them online to z/OS.)

## 2.6 Adding more 3270 terminals

You can add more 3270 sessions for z/VM or for each z/OS. There are several considerations involved.

1. z/VM must know about the additional local 3270 devices. This is done by adding more 3270 definitions in the devmap. z/VM recognizes 3270 terminals regardless of the addresses used. That is, it is not necessary to use addresses in the 7xx range (although our z/VM customization requires a local 3270 session at address 700 for the initial operator session). Remember that a maximum of 32 terminals may be defined for the aws3274 device manager.
2. Once you have additional z/VM 3270 sessions, you can use these sessions to DIAL to a z/VM guest. For example, DIAL ADCD to connect to our first z/OS system.
3. The z/VM userid definition for a guest (S0W1, S0W2, BASEAD) must have sufficient SPECIAL statements for all the 3270 sessions (except the session at address 700 that is used for the MVS console). You can add more SPECIAL statements for these userids by editing the z/VM directory, as outlined in “Adding 3390 volumes” on page 19. These SPECIAL connections are, in effect, hardware connections between a 3270 session and the guest z/OS virtual machine.
4. z/OS must recognize the 3270 session. The current AD systems recognize local 3270 connections at addresses 701-73F. However, the A0600 member in VTAMLST allows only ten TSO sessions through local 3270 connections. You would need to modify VTAMLST to increase this number; this is a normal z/OS systems programming activity.
5. You can connect 3270 sessions to z/OS TCP/IP. The current AD systems allow up to 30 such connections (via definitions in the TCPIP PROFILE and in VTAMLST). These sessions<sup>17</sup> (directly to z/OS TCP/IP) do not require any modifications to z/VM or the devmap.

<sup>17</sup> The limit of 30 is due to TN3270 and VTAM definitions. These can be increased if desired.

## 2.7 Scaling up

It is important to understand some of the limitations of our starter system. It is a relatively small configuration and a variety of steps are required to expand it to a larger framework. In particular, the following areas would need attention:

- ▶ Our two z/OS systems are defined (in the z/VM directory) with modest memory for each. Some application frameworks will require much larger z/OS systems.
- ▶ Our sample devmap defines an 8 or 10 GB zPDT system. This would need to be increased to accommodate the larger z/OS systems.
- ▶ Larger z/OS systems would probably require larger paging data sets. If large virtual memory applications (such as DB2 applications) are used, and if SVC dumps might be expected, then considerably larger paging areas are required for each z/OS. The MVS command D ASM is useful for determining the amount of paging space used. The D A function, under SDSF, can be used to informally sample dynamic paging rates.
- ▶ Larger z/VM and z/OS memory should have correspondingly larger PC server memory, of course.
- ▶ Our z/VM system has one 3390-3 volume for paging. Additional z/VM paging volumes may be needed. You can monitor z/VM system information to determine this.
- ▶ Our Coupling Facilities (in the z/VM directory) are defined with 1024 MB each. Depending on your coupling structures, these may need to be considerably larger.
- ▶ Our z/OS systems have very limited STORAGE disk space (as defined in the VATLST00 members in PARMLIB). You may need to add disk volumes defined for STORAGE.
- ▶ SVC dumps are directed to HDSYS1 and HDSYS2. You may want to direct them to larger volumes that you add.
- ▶ You should monitor paging and swap space in your base Linux. The **xosview** program is convenient for this if you installed it with your Linux.

Remember that paging (by the base Linux, by z/VM, or by z/OS) is generally a poor use of the limited disk I/O resources in a zPDT system. Larger PC memory, combined with a small amount of monitoring to best balance memory usage, can be effective. In spite of these efforts you are likely to see base Linux paging and swap usage; these appear to be mostly due to a strong Linux preference to use real memory for the disk cache. An effective disk cache is important for zPDT performance, of course, and your best operation may be to define a somewhat smaller zPDT memory than you might first think about.

## 2.8 Useful commands

A few commands may be entered directly on a CFCC console. Since we use a common z/VM interface to the CFCC virtual machines, we need to use the z/VM SEND command. The following are examples of direct CFCC commands (these are entered on the CFCONSOL terminal window):

```
SEND CFCC1 DISPLAY MODE
SEND CFCC2 DISPLAY RESOURCES
SEND CFCC1 DISPLAY CHPIDS
SEND CFCC2 HELP
```

A number of z/OS commands are directly related to CF usage. In principle, these commands are issued through the MVS operator console. In practice, some of the commands produce

too much output to be useful on the MVS console panel and are best used via the SDSF LOG interface:

<b>D CF</b>	Best use SDSF interface
<b>D XCF</b>	Brief XCF status
<b>D XCF,CF</b>	More detailed information
<b>D XCF,COUPLE</b>	Lots of output; use SDSF interface
<b>D XCF,POLICY</b>	
<b>D XCF,STR</b>	List defined structures
<b>D XCF,STR,STRNAME=ISGLOCK</b>	Details about specific structure; use SDSF
<b>D LOGREC</b>	LOGREC status
<b>D LOGGER,L</b>	LOGGER status
<b>V OPERLOG,HARDCPY</b>	If not already using LOGGER for some reason
<b>SETLOGRC LOGSTREAM</b>	Change LOGREC recording
<b>SET SMF=xx</b>	Switch to new SMFPRMxx member

The following z/OS commands are not directly related to CF usage, but are helpful in determining the state of the system:

<b>D C,HC</b>	Hardcopy log status
<b>D ETR</b>	External timer mode status
<b>D C</b>	MVS console status
<b>D IPLINFO</b>	
<b>D M=CPU</b>	
<b>D IOS,CONFIG</b>	
<b>D M=STOR</b>	
<b>\$D MASDEF</b>	
<b>\$D MEMBER</b>	

## 2.9 Multiple users and 3270 sessions

z/OS is a multi-user system, of course, and a Parallel Sysplex installation is somewhat pointless without multiple users. This section reviews methods of connecting multiple users to z/OS running under zPDT and the topic is not limited to Parallel Sysplex usage. In this context we are considering only 3270 connections, typically for TSO usage but also for CICS, IMS, and any other application that works with 3270 terminals.

The 3270 terminals are provided by 3270 emulators that communicate through TCP/IP.<sup>18</sup> Examples are x3270 (under Linux), the IBM PCOMM products, and so forth.

### 2.9.1 Methods to connect multiple 3270 sessions

There are two methods of connecting 3270 emulators to z/OS running on a zPDT system:

- Through the aws3274 device manager

This device manager accepts TCP/IP connections (from a 3270 emulator) through the base Linux and transforms them to appear to z/OS as local, channel-attached 3270 terminals. The MVS master console(s) must be connected this way, and this is the most basic connection method for TSO sessions. The 3270 emulator programs may run in the base Linux or be connected through Linux TCP/IP to other systems. The aws3270 device manager provides a maximum of 32 connections.

No z/OS networking is required for this usage.

---

<sup>18</sup> Ethernet SNA connections are not supported by zPDT.



- Through the awsosa device manager

This device manager provides an Ethernet connection to z/OS and can accept 3270 TCP/IP connections. There is no particular limit on the number of connections. The 3270 emulators may run on the base Linux system or on other systems connected to z/OS TCP/IP.

z/OS TCP/IP networking must be configured and active for this usage.

The same 3270 emulator programs may be used for either connection method.

## 2.9.2 Multiple user configurations

The following discussion covers several possible configurations for multiple user operation. This is a very basic discussion and is intended for readers not yet familiar with typical zPDT usage.

### Base Linux system

There is a difference between multiple users and multiple 3270 sessions. A basic zPDT system may have multiple 3270 sessions, each with a separate logon to TSO for example. This is indicated in Figure 2-6.

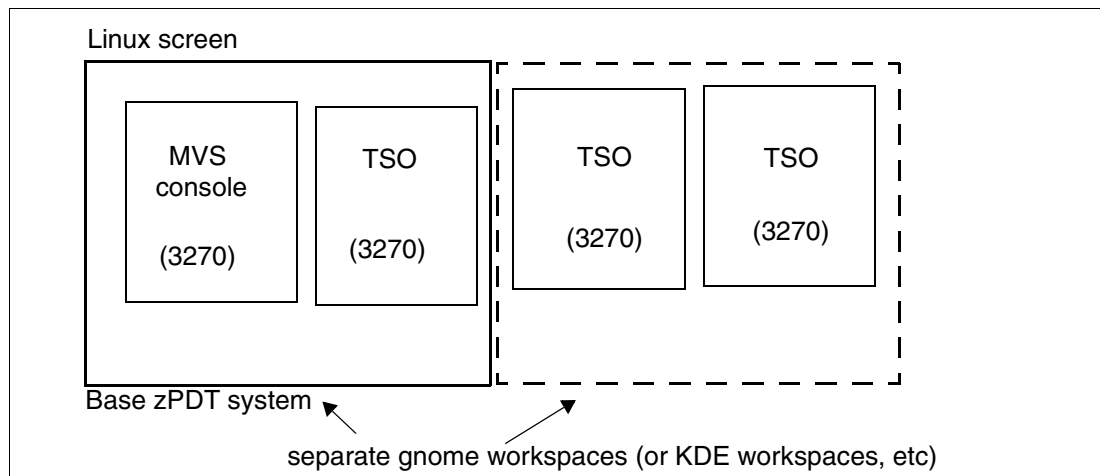


Figure 2-6 Base system, multiple 3270 sessions

To z/OS, this represents multiple users. To a human, this is a single user with multiple 3270 sessions. Multiple people could use this configuration only by taking turns at the single PC keyboard. This is similar to the basic window layout for initial Parallel Sysplex operation that is suggested in Figure 2-4 on page 12. More than two 3270 sessions may be used in each workspace, but the panel becomes confusing if too many 3270 windows are open.

The advantages of this configuration include:

- No networking is required, other than the internal TCP/IP within the base Linux.
- Up to 32 3270 sessions (including the MVS console function) may be used.

The disadvantages of this configuration include:

- It is not suitable for multiple human users (all at the same time).
- Switching between workspaces may be needed to access the desired 3270 session.
- Multiple workspaces prevent the user from seeing all the sessions at the same time.
- Too many sessions in one workspace can be confusing and it is very easy to type into the wrong 3270 session.

## Private local router

A private router (wired or wireless) can provide a simple networked environment, as illustrated in Figure 2-7.

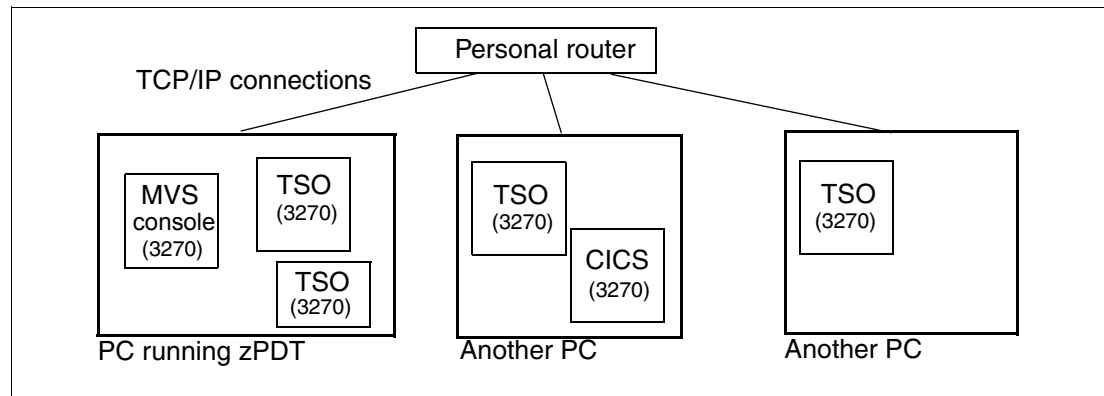


Figure 2-7 Multiple PCs with private router

The advantages of this configuration include:

- ▶ This is suitable for multiple human users.
- ▶ The private router can be configured with fixed IP addresses, allowing simple addressing of the IP connections.
- ▶ Logically separate users and users can have their own panels, keyboards, and 3270 sessions. This reduces confusion created by too many 3270 sessions on a PC window.
- ▶ There are generally no communication security concerns with this configuration.

The disadvantages of this configuration include:

- ▶ Some networking skill is required. This is a very simple network environment to someone who is familiar with network management, but it could be a challenge for someone with no networking experience.
- ▶ Multiple PCs are needed and each must have a 3270 emulator and each must be configured for network use.
- ▶ With a private router users are typically located within a relatively small area, although this is not required.

## Internet network

This is similar to the previous example, with the private router replaced by an Internet cloud; it is illustrated in Figure 2-8 on page 25.

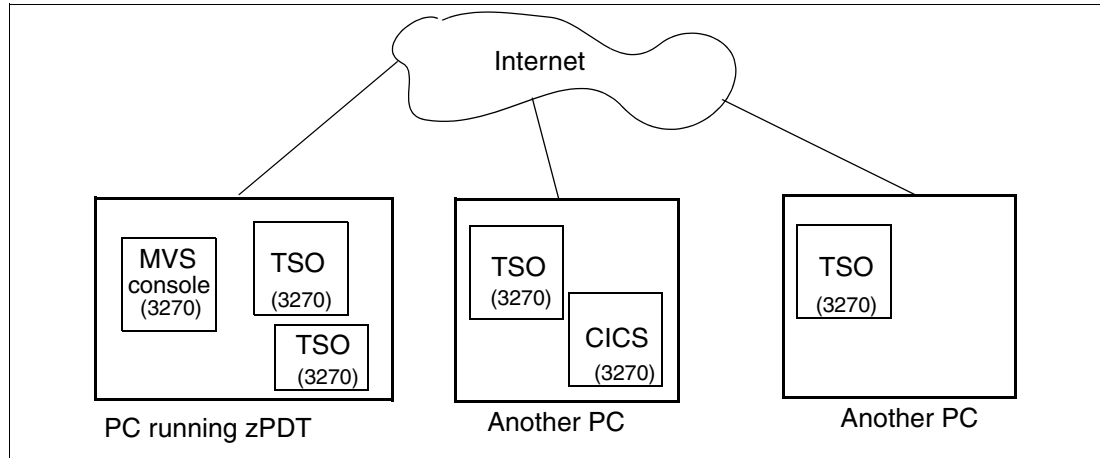


Figure 2-8 Multiple PCs connected through an Internet

The advantages of this configuration include:

- ▶ This is suitable for multiple human users.
- ▶ Logically separate users and users can have their own panels, keyboards, and 3270 sessions. This reduces confusion created by too many 3270 sessions on a PC window.
- ▶ Users (on the remote PCs) may be located anywhere that is connected to the network.
- ▶ The networking skills needed are minimal, but some setup is needed.

The disadvantages of this configuration include:

- ▶ Multiple PCs are needed and each must have a 3270 emulator and each must be configured for network use.
- ▶ DHCP addressing is probably used for the connections. This means that the IP address needed to connect 3270 sessions is changeable; a remote user must be notified of the current IP address. The DHCP address may change.
- ▶ Security concerns may exist, depending on the applications and usage involved.

### Base Linux with thin client

The thin client we used is a *Pluggable Technologies USB 2.0 Docking Station* ordered through Amazon. The manufacturer has several versions available or in development and more information may be obtained at:

<http://pluggable.com/products/ud-160-m/>

or by a web search.<sup>19</sup>

This thin client (Figure 2-9 on page 26) is supported by Fedora 17 and 18.<sup>20</sup> It uses a simple USB connection to the Linux machine running zPDT and no other configuration is required. When connected, it presents a Linux logon window.<sup>21</sup> The user simply logs into Linux and starts x3270 sessions. These are local TCP/IP connections, the same as the x3270 sessions

<sup>19</sup> This is not an IBM product and there is no IBM endorsement of the product or company.

<sup>20</sup> At the time of writing Fedora 17 or 18 is not an “officially supported” base for zPDT. We installed it from DVD plus an online connection. The online connection was needed to obtain a required 32-bit support library (libstdc++-4.7.0-5.fc17.i686). We were pleased to find that x3270 was also available. It may be possible to add these remotely-supplied items later, using a command such as `sudo yum install x3270-x11`.

<sup>21</sup> This thin client adapts to window resolutions up to 1920x1080.

started on the base PC. The same Linux userid that was used on the base machine may be used, or a different Linux userid may be used.

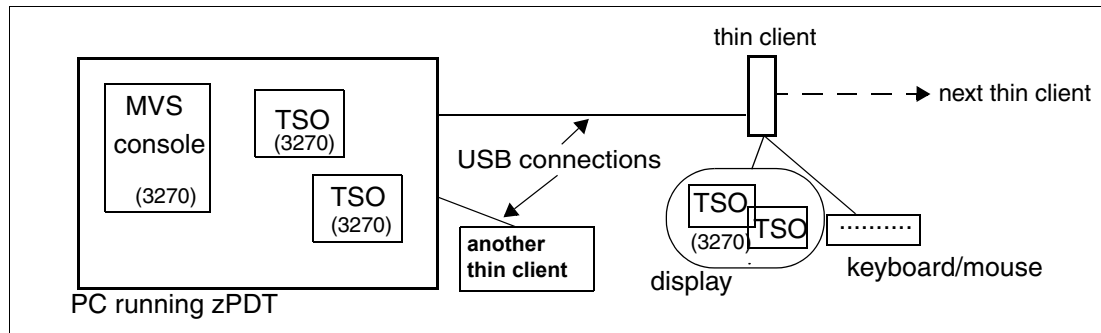


Figure 2-9 Base Linux with thin clients

The current implementation in Fedora 17 allows a maximum of seven thin clients, with a maximum of four “daisy chained” together and others connected to USB hubs.

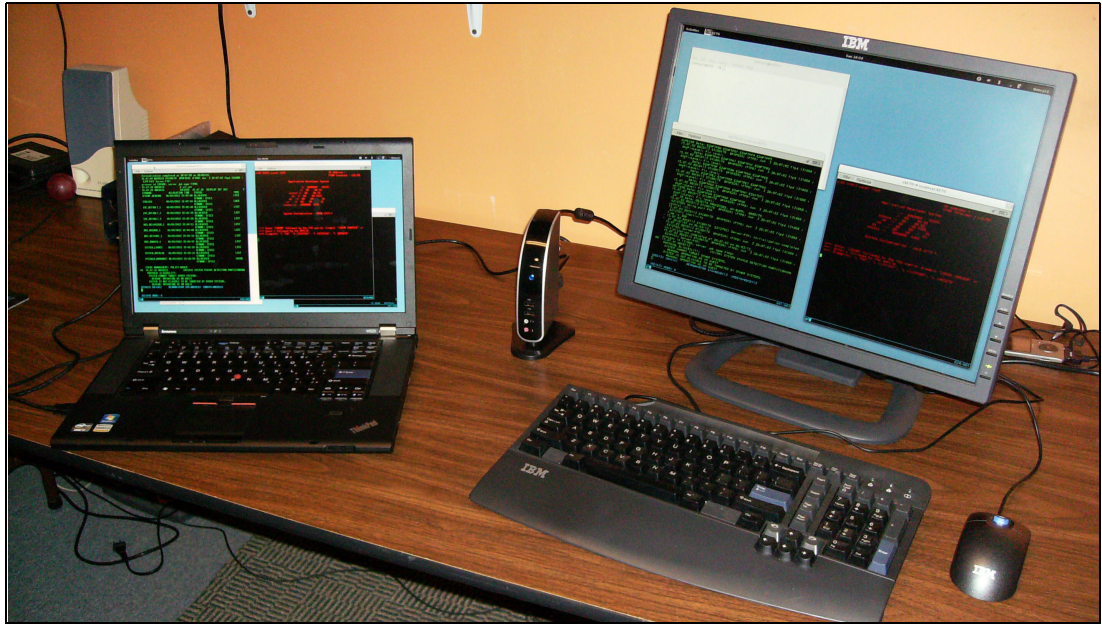
The advantages of this configuration include:

- ▶ No networking is needed; no network skills are needed.
- ▶ No additional PCs are needed and no additional software is needed. This removes all the potential cost and administrative time associated with additional computers. The configuration is very inexpensive.
- ▶ There should be no security concerns.

The disadvantages of this configuration include:

- ▶ The maximum USB cable length is something like 20 feet, making this arrangement usable only within a single room.
- ▶ At the time of writing, only the Fedora 17 and 18 Linux distributions had the necessary support for these thin clients. However, the support is being designed for the Linux kernel and we expect it will eventually be a standard Linux function. The Linux implementation is quite new and minor details will need to be resolved over time.

Figure 2-10 on page 27 depicts a thin client in use with an earlier version of our Parallel Sysplex starter system. In this photo we had our Parallel Sysplex starter system running. The laptop has Linux (Fedora 17), zPDT, an x3270 window (partly hidden) for CFCC control, the MVS console and a TSO session connected to the first z/OS. The thin client has the MVS console and a TSO session for the second z/OS. (We later started more TSO sessions on the thin client.) We found this arrangement easy to use and reduced errors associated with typing into the wrong x3270 session.



*Figure 2-10 Thin client in use*

The thin client is the box between the laptop and the display. It has a USB connection to the laptop and uses additional USB ports to connect to the thin client keyboard and mouse. The thin client has a standard display port. The laptop in this photo is a Lenovo W520, with four cores and 16 GB of memory.



## Building the starter system

In this chapter we describe how we created our Parallel Sysplex system, starting with the distributed AD-CD z/OS 1.13H<sup>1</sup> system. This information may be useful for migration to new releases, and as background information for expanding the scope of our Parallel Sysplex starter system. We do not claim to have used the best methods or the most efficient techniques. We almost always elected to use the simplest way to accomplish our goals. All of the batch jobs we used are distributed on volume CF0001 in data set OGDEN.SYSPLEX.CNTL. Most of the following work was done while running under the base (non-sysplex) AD-CD system.

Note that the files on your DVD or download may differ slightly from the following listings, due to tuning and updates after this document was written. The purpose of the following listings is to provide an overview of what was done to create the system.

Remember that we essentially have three z/OS systems resident in one set of volumes. The naming conventions in Table 3-1 may help clarify this. The X1 and Y1 IPL parameters are for a cold start and the X2 and Y2 IPL parameters are for warm starts.<sup>2</sup>

*Table 3-1 Naming conventions*

z/OS system	IPL parameters	PARMLIB members suffix
sysplex S0W1 (under z/VM)	X1 or X2	X1 or X2
sysplex S0W2 (under z/VM)	Y1 or Y2	Y1 or Y2
BASEAD (under z/VM)	all others	all others
base ADCD (not under z/VM)	all others	all others

The BASEAD system and the base ADCD are essentially the same and do not use Coupling Facility functions. The sole difference is that a z/VM guest is defined for BASEAD. This permits you to IPL a basic z/OS (no Parallel Sysplex) while still under z/VM.

<sup>1</sup> The 1.13H notation indicates the z/OS 1.13 AD-CD update that was made available in January 2013. The volume serial numbers all begin with the letter H.

<sup>2</sup> Solely for the AD-CD design, a cold start includes a CLPA and JES2 cold start. A warm start does not include CLPA or a JES2 cold start. This is the only difference between cold and warm starts for the AD-CD z/OS implementations. The Y1 parameters (for S0W2) produce a CLPA but not a JES2 cold start.

The jobs referenced in the following material are found on the CF0001 volume.

## 3.1 Volumes and coupling data sets

We used normal zPDT functions (the **alckkd** command) to create five 3390 volumes: CF0001, HDSYS2, WORK01, HDPAGY, and HDPAGZ. CF0001 is a 3390-1 and the others are 3390-3 volumes. We then used ICKDSF to initialize the volumes.

### 3.1.1 Clone BDCD specific data sets

The VSAM1 job creates the paging, VIO, SMF, and LOGREC data sets for S0W2. These are all placed on volser HDSYS2. The data sets are cataloged in the master catalog. The ADCD system (as part of the sysplex) uses the paging, SMF, VIO, and LOGREC data sets of the base (non-sysplex) z/OS system.

When using logstreams (LOGR) for LOGREC data the LOGREC data set created here is not used. We created it so that you have a choice of using a data set or a log stream for LOGREC data. Likewise, when using logstreams for SMF data the SYS2.MANx data sets are not used. Previous AD-CD releases used three SYS1.MANx data sets and we allocate three here, although the 1.13H release uses only the first two.

```
MEMBER NAME VSAM1
//VSAM1 JOB 1,OGDEN,MSGCLASS=X
//S1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//DD1 DD VOL=SER=HDPAGY,UNIT=3390,DISP=SHR
//DD2 DD VOL=SER=HDPAGZ,UNIT=3390,DISP=SHR
//SYSIN DD *
  DEFINE PGSPC(NAME(SYS2.PLPA.PAGE) -
    FILE(DD1) CYLINDERS(100) VOLUME(HDPAGY))
  DEFINE PGSPC(NAME(SYS2.COMMON.PAGE) -
    FILE(DD1) CYLINDERS(50) VOLUME(HDPAGY))
  DEFINE PGSPC(NAME(SYS2.LOCALX.PAGE) -
    FILE(DD1) CYLINDERS(3000) VOLUME(HDPAGY))
  DEFINE PGSPC(NAME(SYS2.LOCALY.PAGE) -
    FILE(DD2) CYLINDERS(3100) VOLUME(HDPAGZ))

/*
//S2 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//DD1 DD VOL=SER=HDSYS2,UNIT=3390,DISP=SHR
//SYSIN DD *
  DEFINE CLUSTER(NAME(SYS2.MAN1) REUSE NIXD SPND SPEED SHR(2) -
    VOLUME(HDSYS2) CYLINDERS(40) RECSZ(4086,32767) CNVSZ(4096))
  DEFINE CLUSTER(NAME(SYS2.MAN2) REUSE NIXD SPND SPEED SHR(2) -
    VOLUME(HDSYS2) CYLINDERS(10) RECSZ(4086,32767) CNVSZ(4096))
  DEFINE CLUSTER(NAME(SYS2.MAN3) REUSE NIXD SPND SPEED SHR(2) -
    VOLUME(HDSYS2) CYLINDERS(10) RECSZ(4086,32767) CNVSZ(4096))
  DEFINE CL(NAME(SYS2.STGINDEX) VOLUME(HDSYS2) CYLINDERS(10) -
    KEYS(12 8) BUFSPC(20480) RECSZ(2041 2041) FILE(DD1) REUSE) -
    DATA (CNVSZ(2048)) INDEX (CNVSZ(4096))
/*
//S3 EXEC PGM=IFCDIP00
```



```

//SERERDS DD DSN=SYS2.LOGREC,UNIT=3390,VOL=SER=HDSYS2,
//      DISP=(NEW,CATLG),SPACE=(TRK,(90),,CONTIG)
//SYSPRINT DD SYSOUT=*
/*

```

## Couple data sets

We used the COUPLE2 job to create the coupling data sets, all placed on volume CF0001. A larger Parallel Sysplex system would not have these data sets all on the same volume for both performance and recovery reasons.

```

MEMBER NAME COUPLE2
//COUPLE2 JOB 1,OGDEN,MSGCLASS=X,REGION=40M
// EXEC PGM=IXCL1DSU
//STEPLIB DD DSN=SYS1.MIGLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD DATA,DLM='##'
/* SYSPLEX COUPLE DATASETS */
  DEFINEDS SYSPLEX(ADCDPL)
    MAXSYSTEM(4)
    DSN(SYSPLEX.ADCDPL.XCF.CDS01) VOLSER(CF0001)
    CATALOG
    DATA TYPE(SYSPLEX)
    ITEM NAME(GRS) NUMBER(1)
    ITEM NAME(GROUP) NUMBER(48)
    ITEM NAME(MEMBER) NUMBER(32)
  DEFINEDS SYSPLEX(ADCDPL)
    MAXSYSTEM(4)
    DSN(SYSPLEX.ADCDPL.XCF.CDS02) VOLSER(CF0001)
    CATALOG
    DATA TYPE(SYSPLEX)
    ITEM NAME(GRS) NUMBER(1)
    ITEM NAME(GROUP) NUMBER(48)
    ITEM NAME(MEMBER) NUMBER(32)
/* CFRM COUPLE DATASETS */
  DEFINEDS SYSPLEX(ADCDPL)
    MAXSYSTEM(4)
    DSN(SYSPLEX.ADCDPL.CFRM.CDS01) VOLSER(CF0001)
    CATALOG
    DATA TYPE(CFRM)
    ITEM NAME(POLICY) NUMBER(4)
    ITEM NAME(CF) NUMBER(2)
    ITEM NAME(STR) NUMBER(16)
    ITEM NAME(CONNECT) NUMBER(8)
    ITEM NAME(SMREBLD) NUMBER(1)
    ITEM NAME(SMDUPLEX) NUMBER(1)
  DEFINEDS SYSPLEX(ADCDPL)
    MAXSYSTEM(4)
    DSN(SYSPLEX.ADCDPL.CFRM.CDS02) VOLSER(CF0001)
    CATALOG
    DATA TYPE(CFRM)
    ITEM NAME(POLICY) NUMBER(4)
    ITEM NAME(CF) NUMBER(2)
    ITEM NAME(STR) NUMBER(16)
    ITEM NAME(CONNECT) NUMBER(8)
    ITEM NAME(SMREBLD) NUMBER(1)

```

```

        ITEM NAME(SMDUPLEX) NUMBER(1)
/*      OMVS COUPLE DATASETS      */
DEFINEDS SYSPLEX(ADCDPL)
        MAXSYSTEM(4)
        DSN(SYSPLEX.ADCDPL.OMVS.CDS01) VOLSER(CF0001)
        CATALOG
        DATA TYPE(BPXMCD)
        ITEM NAME(MOUNTS) NUMBER(100)
        ITEM NAME(AMTRULES) NUMBER(50)
DEFINEDS SYSPLEX(ADCDPL)
        MAXSYSTEM(4)
        DSN(SYSPLEX.ADCDPL.OMVS.CDS02) VOLSER(CF0001)
        CATALOG
        DATA TYPE(BPXMCD)
        ITEM NAME(MOUNTS) NUMBER(100)
        ITEM NAME(AMTRULES) NUMBER(50)
/*      LOGR COUPLE DATASETS      */
DEFINEDS SYSPLEX(ADCDPL)
        MAXSYSTEM(4)
        DSN(SYSPLEX.ADCDPL.LOGR.CDS01) VOLSER(CF0001)
        CATALOG
        DATA TYPE(LOGR)
        ITEM NAME(LSR) NUMBER(200)
        ITEM NAME(LSTRR) NUMBER(100)
        ITEM NAME(DSEXTENT) NUMBER(20)
        ITEM NAME(SMDUPLEX) NUMBER(1)
DEFINEDS SYSPLEX(ADCDPL)
        MAXSYSTEM(4)
        DSN(SYSPLEX.ADCDPL.LOGR.CDS02) VOLSER(CF0001)
        CATALOG
        DATA TYPE(LOGR)
        ITEM NAME(LSR) NUMBER(200)
        ITEM NAME(LSTRR) NUMBER(100)
        ITEM NAME(DSEXTENT) NUMBER(20)
        ITEM NAME(SMDUPLEX) NUMBER(1)
/*      WLM COUPLE DATASETS      */
DEFINEDS SYSPLEX(ADCDPL)
        MAXSYSTEM(4)
        DSN(SYSPLEX.ADCDPL.WLM.CDS01) VOLSER(CF0001)
        CATALOG
        DATA TYPE(WLM)
        ITEM NAME(POLICY) NUMBER(10)
        ITEM NAME(WORKLOAD) NUMBER(35)
        ITEM NAME(SRVCLASS) NUMBER(200)
        ITEM NAME(APPLENV) NUMBER(200)
        ITEM NAME(SCHENV) NUMBER(100)
        ITEM NAME(SVDEFEXT) NUMBER(100)
        ITEM NAME(SVDCREXT) NUMBER(100)
        ITEM NAME(SVAEAEXT) NUMBER(100)
        ITEM NAME(SVSEAEXT) NUMBER(100)
DEFINEDS SYSPLEX(ADCDPL)
        MAXSYSTEM(4)
        DSN(SYSPLEX.ADCDPL.WLM.CDS02) VOLSER(CF0001)
        CATALOG
        DATA TYPE(WLM)

```

```

ITEM NAME(POLICY) NUMBER(10)
ITEM NAME(WORKLOAD) NUMBER(35)
ITEM NAME(SRVCLASS) NUMBER(200)
ITEM NAME(APPLENV) NUMBER(200)
ITEM NAME(SCHENV) NUMBER(100)
ITEM NAME(SVDEFEXT) NUMBER(100)
ITEM NAME(SVDCREXT) NUMBER(100)
ITEM NAME(SVAEAEXT) NUMBER(100)
ITEM NAME(SVSEAEXT) NUMBER(100)

##

```

After creating the coupling data sets, we used job CRFMPOL3 to define the Coupling Facility policies and structures. The assignment of the structures to the two CFCCs (in the PREFLIST parameters) is quite arbitrary.

```

MEMBER NAME CRFMPOL3
//CRFMPOL3 JOB 1,OGDEN,MSGCLASS=X
// EXEC PGM=IXCMIAPU
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSIN DD *
DATA TYPE(CFRM) REPORT(YES)
DSN(SYSPLEX.ADCDPL.CFRM.CDS01)
DEFINE POLICY NAME(CFRM01) REPLACE(YES)
CF NAME(CFCC1)
TYPE(SIMDEV)
MFG(IBM)
PLANT(EN)
SEQUENCE(0000000CFCC1)
PARTITION(0)
CPCID(00)
DUMPSPACE(2000)

CF NAME(CFCC2)
TYPE(SIMDEV)
MFG(IBM)
PLANT(EN)
SEQUENCE(0000000CFCC2)
PARTITION(0)
CPCID(00)
DUMPSPACE(2000)

STRUCTURE NAME(IXC_DEFAULT_1)
SIZE(16128)
PREFLIST(CFCC1,CFCC2)

STRUCTURE NAME(IXC_DEFAULT_2)
SIZE(16128)
PREFLIST(CFCC2,CFCC1)

STRUCTURE NAME(IXC_DEFAULT_3)
SIZE(16128)
PREFLIST(CFCC2,CFCC1)

STRUCTURE NAME(SYSTEM_OPERLOG)
INITSIZE(32768)

```

```

SIZE(131072)
PREFLIST(CFCC1,CFCC2)
FULLTHRESHOLD(90)

STRUCTURE NAME(SYSTEM_LOGREC)
SIZE(65636)
PREFLIST(CFCC1,CFCC2)

STRUCTURE NAME(ISGLOCK)
SIZE(17000)
PREFLIST(CFCC1,CFCC2)
REBUILDPERCENT(1)

STRUCTURE NAME(SYSZWLM_WORKUNIT)
SIZE(12000)
INITSIZE(10000)
PREFLIST(CFCC2,CFCC1)

STRUCTURE NAME(RRS_RMDATA_1)
INITSIZE(3000)
SIZE(4096)
PREFLIST(CFCC1,CFCC2)

STRUCTURE NAME(RRS_MAINUR_1)
INITSIZE(9216)
SIZE(9216)
PREFLIST(CFCC2,CFCC1)

STRUCTURE NAME(RRS_DELAYEDUR_1)
INITSIZE(10240)
SIZE(24576)
PREFLIST(CFCC1,CFCC2)

STRUCTURE NAME(RRS_RESTART_1)
INITSIZE(10240)
SIZE(28672)
PREFLIST(CFCC2,CFCC1)

STRUCTURE NAME(SMF_GENERAL))
INITSIZE(9216)
SIZE(28672)
PREFLIST(CFCC1,CFCC2)

```

### 3.1.2 SMS changes

Three changes are needed for SMS:

- ▶ The second z/OS system (S0W2) and the Parallel Sysplex name (ADCDPL) must be added.
- ▶ A data class must be added for the automatic allocation of LOGR datasets.
- ▶ SMS control data sets must have their share option changed to 3,3.

#### Add BDCD and ADCDPL to SMS

1. Go to the ISMF panels in ISPF. (This is option M.2 in recent AD-CD systems.)

2. Ensure that you are in Administrator mode. (If you can see option 8 (Control Data Set) in the panel, then you are in Administrator mode. If not, select option 0 (Profile), then option 0 (User Mode), then option 2 (Administrator); then exit completely out of ISMF (using PF3s) and restart it.)
3. Select option 8 (Control Data Set).
4. Set CDS name to 'SYS1.SCDS' at the top of the panel.
5. Select option 3 (Alter).
6. Page (PF8) to second panel.
7. Specify option 1 (Add), specify System Name S0W2.
8. Specify option 1 (Add), specify Sys Group Name ADCDPL.
9. Exit this panel (PF3).
10. Select option 4 (Validate the SCDS) and look for SUCCESSFUL VALIDATION.
11. Select option 5 to activate the CDS. Enter / (slash) to request activation.
12. You should see an ACTIVATION SCHEDULED message and the MVS console should have a NEW CONFIGURATION ACTIVATED message.
13. Use PF3 to exit from ISMF.

### Create data class for LOGR

1. Go to the ISMF panels in ISPF.
2. Select option 4 (Data Class).
3. Set CDS name to SYS1.SCDS at the top of the panel.
4. Enter LOGR4K as the Data Class Name.
5. Select option 3 (Define).
6. Use PF8 to scroll through a number of panels. Make *only* the following changes:
7. RECORG = LS
8. Clsize Data = 4096<sup>3</sup>
9. Shareoptions Xregion = 3
10. Xsystem = 3
11. Use PF3 to return to the main ISMF menu.
12. Use option 8 (Control Data Set) and option 5 (Activate the CDS) to activate the modified CDS.

Job ACDS5 changes the share options of several SMS data sets to allow use by both S0W1 and S0W2.

```

MEMBER NAME  ACDS5
//ACDS5  JOB 1,OGDEN,MSGCLASS=X
// EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
ALTER SYS1.ACDS.DATA SHAREOPTIONS(3 3)
ALTER SYS1.SCDS.DATA SHAREOPTIONS(3 3)
ALTER SYS1.COMMDS.DATA SHAREOPTIONS(3 3)
/*

```

<sup>3</sup> Production Parallel Sysplex systems would typically have this value set to 24K. However, there are side issues with values larger than 4K.

### 3.1.3 HFS and zFS changes

We needed to clone the HFS and zFS data sets specific to each z/OS system. We used job HFS4 to clone these data sets and place them on volume HDSYS2. The naming in our starter system is poor. We probably should have changed the ADCD113H and BDCD113H qualifiers to SOW1 and SOW2, but this involved too many other changes in the base AD-CD system.

```
MEMBER NAME  HFS4
//HFS4 JOB 1,OGDEN,MSGCLASS=X,REGION=OK
/*
/** COPY INSTANCE HFS AND ZFS DATASETS
/**
// EXEC PGM=ADRDSSU
//SYSPRINT DD SYSOUT=*
//FROMDASD DD UNIT=3390,DISP=SHR,VOL=SER=HDUSS1
//TODASD DD UNIT=3390,DISP=SHR,VOL=SER=HDSYS2
//SYSIN DD *
COPY DATASET(INCLUDE(ZFS.ADCD113H.DEV, -
                     ZFS.ADCD113H.VAR, -
                     ZFS.ADCD113H.ETC, -
                     ZFS.ADCD113H.SYSTEM, -
                     ZFS.ADCD113H.VARWBEM, -
                     ZFS.ADCD113H.WEB, -
                     HFS.ADCD113H.USERS, -
                     HFS.ADCD113H.USR.MAIL, -
                     HFS.ADCD113H.WEB.SAV, -
                     HFS.ADCD113H.TMP )) -
INDD(FROMDASD) OUTDD(TODASD) -
TOLERATE(ENQF) -
RENAMEU((ZFS.ADCD113H.DEV,ZFS.BDCD113H.DEV), -
        (ZFS.ADCD113H.VAR,ZFS.BDCD113H.VAR), -
        (ZFS.ADCD113H.ETC,ZFS.BDCD113H.ETC), -
        (ZFS.ADCD113H.SYSTEM,ZFS.BDCD113H.SYSTEM), -
        (ZFS.ADCD113H.VARWBEM,ZFS.BDCD113H.VARWBEM), -
        (ZFS.ADCD113H.WEB,ZFS.BDCD113H.WEB), -
        (HFS.ADCD113H.USERS,HFS.BDCD113H.USERS), -
        (HFS.ADCD113H.USR.MAIL,HFS.BDCD113H.USR.MAIL), -
        (HFS.ADCD113H.WEB.SAV,HFS.BDCD113H.WEB.SAV), -
        (HFS.ADCD113H.TMP,HFS.BDCD113H.TMP)) -
SHARE ALLEXCP ALLDATA(*) CANCELERROR CATALOG
/*
```

The AD-CD 1.13H system already had the sysplex root and instance file systems created. The HFS4A job (on previous versions of volume CF0001) is not needed. If it were needed, this job would need to be altered to use SYS1.SAMPLIB(BPXISYZR) and SYS1.SAMPLIB(BPXISYZS) as the bases for the job.

### 3.1.4 CF structure definitions

We later used the LATER66 job to complete structure and log definitions for OPERLOG, LOGREC, RRS,<sup>4</sup> and SMF operation; this just must be run under a target system (in Parallel Sysplex operation).

**MEMBER NAME LATER6 - run this job later, in Parallel Sysplex mode**

<sup>4</sup> We defined RRS logstreams but did not use them in our minimal Parallel Sysplex operation.

```

//LATER6 JOB 1,OGDEN,MSGCLASS=X
//*
/* MUST HAVE CFRM SET BEFORE RUNNING THIS JOB
/* RUN THIS JOB WHILE IN PARALLEL SYSPLEX MODE.
/*
//POLICY EXEC PGM=IXCMIAPU
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DATA TYPE(LOGR)
DEFINE STRUCTURE NAME(SYSTEM_OPERLOG)
  LOGSNUM(1)
  MAXBUFSIZE(4092)
  AVGBUFSIZE(512)

DEFINE LOGSTREAM NAME(SYSPLEX.OPERLOG)
  STRUCTNAME(SYSTEM_OPERLOG)
  LS_DATACLAS(LOGR4K)
  HLQ(IXGLOGR)
  LS_SIZE(1024)
  LOWOFFLOAD(0)
  HIGHOFFLOAD(80)
  STG_DUPLEX(NO)
  RETPD(7)
  AUTODELETE(YES)
/*
/*
//DEF      EXEC PGM=IXCMIAPU
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DATA TYPE(LOGR)
DEFINE STRUCTURE NAME(SYSTEM_LOGREC)
  LOGSNUM(1)
  AVGBUFSIZE(4068)
  MAXBUFSIZE(4068)

DEFINE LOGSTREAM NAME(SYSPLEX.LOGREC.ALLRECS)
  STRUCTNAME(SYSTEM_LOGREC)
  LS_DATACLAS(LOGR4K)
  HLQ(IXGLOGR)
  LS_SIZE(1024)
  LOWOFFLOAD(0)
  HIGHOFFLOAD(80)
  STG_DUPLEX(NO)
  RETPD(7)
  AUTODELETE(YES)
/*
/*
//RRS EXEC PGM=IXCMIAPU
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  DATA TYPE(LOGR) REPORT(YES)

  DEFINE STRUCTURE NAME(RRS_RMDATA_1) LOGSNUM(1)
  MAXBUFSIZE(1024) AVGBUFSIZE(252)

```

```

DEFINE STRUCTURE NAME(RRS_MAINUR_1) LOGSNUM(1)
MAXBUFSIZE(65276) AVGBUFSIZE(158)

DEFINE STRUCTURE NAME(RRS_DELAYEDUR_1) LOGSNUM(1)
MAXBUFSIZE(65256) AVGBUFSIZE(158)

DEFINE STRUCTURE NAME(RRS_RMDATA_1) LOGSNUM(1)
MAXBUFSIZE(65256) AVGBUFSIZE(158)

DEFINE LOGSTREAM
NAME(ATR.ADCDPL.RM.DATA) STRUCTNAME(RRS_RMDATA_1)
LS_DATACLAS(LOGR4K)
HLQ(IXGLOGR) MODEL(NO) LS_SIZE(1024)
LOWOFFLOAD(60) HIGHOFFLOAD(80) STG_DUPLEX(NO)
RETPD(0) AUTODELETE(NO)

DEFINE LOGSTREAM
NAME(ATR.ADCDPL.MAIN.UR) STRUCTNAME(RRS_MAINUR_1)
LS_DATACLAS(LOGR4K)
HLQ(IXGLOGR) MODEL(NO) LS_SIZE(1024)
LOWOFFLOAD(60) HIGHOFFLOAD(80) STG_DUPLEX(NO)
RETPD(0) AUTODELETE(NO)

DEFINE LOGSTREAM
NAME(ATR.ADCDPL.DELAYED.UR) STRUCTNAME(RRS_DELAYEDUR_1)
LS_DATACLAS(LOGR4K)
HLQ(IXGLOGR) MODEL(NO) LS_SIZE(1024)
LOWOFFLOAD(60) HIGHOFFLOAD(80) STG_DUPLEX(NO)
RETPD(0) AUTODELETE(NO)

DEFINE LOGSTREAM
NAME(ATR.ADCDPL.RESTART) STRUCTNAME(RRS_RESTART_1)
LS_DATACLAS(LOGR4K)
HLQ(IXGLOGR) MODEL(NO) LS_SIZE(1024)
LOWOFFLOAD(60) HIGHOFFLOAD(80) STG_DUPLEX(NO)
RETPD(0) AUTODELETE(NO)
/*
/**
//SSMF EXEC PGM=IXCMIAPU
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DATA TYPE(LOGR)

DEFINE STRUCTURE NAME(IFASMF_GENERAL)
LOGSNUM(1)
MAXBUFSIZE(33024)
AVGBUFSIZE(512)

DEFINE LOGSTREAM NAME(IFASMF_GENERAL)
STRUCTNAME(IFASMF_GENERAL)
LS_DATACLAS(LOGR4K)
HLQ(IXGLOGR)
LS_SIZE(1024)
LOWOFFLOAD(0)
HIGHOFFLOAD(80)

```



```

STG_DUPLEX(YES)
RETPD(7)
AUTODELETE(YES)
/*

```

## 3.2 PARMLIB changes

Many PARMLIB changes are needed. These mostly consisted of adding members to ADCD.Z113H.PARMLIB. An alternative would be to add these members in USER.PARMLIB.

### 3.2.1 IEASYSxx members

New IEASYSxx members are the key elements here. Significant changes in these members are shown in bold.

<b>MEMBER NAME</b>	<b>IEASYSX1</b> (based on IEASYSXS)
<b>CLOCK=X1,</b>	<b>MODIFIED</b>
CLPA,	
CMB=(UNITR,COMM,GRAPH,CHRD),	ADDITIONAL CMB ENTRIES
<b>CMD=X1,</b>	<b>MODIFIED</b>
<b>CON=(X1,NOJES3),</b>	<b>MODIFIED</b>
<b>COUPLE=X1,</b>	<b>MODIFIED</b>
CSA=(3000,200000),	CSA RANGE
DIAG=00,	SELECT DIAG00, DIAGNOSTIC COMMANDS
DUMP=DASD,	PLACE SVC DUMPS ON DASD DEVICES
FIX=00,	SELECT IEAFIX00, FIX MODULES SPECIFIED
<b>GRS=STAR,</b>	<b>MODIFIED; TRYJOIN FOR INITIAL STARTUP</b>
<b>GRSRNL=00,</b>	<b>ADDED</b>
ILMMODE=NONE,	EXPLICITLY DISABLE LICENSE MANAGER
LNKAUTH=LNKLST,	AUTHORIZE LNKLST00, APFTAB IS ALTERNATE
LOGCLS=L,	WILL NOT BE PRINTED BY DEFAULT
LOGLMT=999999,	MAX WTL MESSAGES QUEUED, MUST BE 6 DIGITS
<b>LOGREC=LOGSTREAM,</b>	<b>COULD BE CHANGED TO SYS1.LOGREC</b>
<b>LPA=01,</b>	1.13 AD-CD ERRATIC NAMING HERE
MAXUSER=250,	SYS TASKS PLUS INITS PLUS TSUSERS
MLPA=00,	SELECT IEALPA00, MLPA PARAMETERS
MSTRJCL=00,	SELECT MSTJCLEX, MASTER JCL
OMVS=CS,	NOT MODIFIED
OPI=YES,	ALLOW OPERATOR OVERRIDE TO IEASYS00
<b>PAGE=(SYS1.PLPA.PAGE,</b>	
<b>      SYS1.COMMON.PAGE,</b>	
<b>      SYS1.LOCALA.PAGE,</b>	
<b>      SYS1.LOCALB.PAGE,L),</b>	
PAK=00,	SELECT IEAPAK00
PLEXCFG=ANY,	
PROG=01,	1.13 AD-CD ERRATIC NAMING HERE
REAL=128,	ALLOWS 2 64K OR 1 128K JOB TO RUN V=R
RSU=0,	NO RECONFIG STORAGE UNITS
RSVNONR=100,	RESERVED ASVT ENTRIES
RSVSTRT=5,	RESERVED ASVT ENTRIES
SCH=00,	SELECT SCHED00
<b>SMF=X1,</b>	<b>MODIFIED</b>

SQA=(15,64),	SQA SIZE APPROX 640K
SSN=00,	SELECT IEFSSN00, SUBSYSTEM NAMES
PROD=(00,01),	PLEASE NOTE IFAPRD01
SVC=00,	SELECT IEASVC00, USER SVCS
VAL=00,	SELECT VATLST00
VIODSN=SYS1.STGINDEX,	VIO DS
VRREGN=64	DEFAULT REAL-STORAGE REGION SIZE
 <b>MEMBER NAME IEASYSY1</b> (based on IEASYSCS)	
<b>CLOCK=X1,</b>	<b>MODIFIED USE X1</b>
CLPA,	
CMB=(UNITR,COMM,GRAPH,CHRD),	ADDITIONAL CMB ENTRIES
<b>CMD=Y1,</b>	<b>MODIFIED</b>
<b>CON=(Y1,NOJES3),</b>	<b>MODIFIED</b>
<b>COUPLE=X1,</b>	<b>MODIFIED USE X1</b>
CSA=(3000,200000),	CSA RANGE
DIAG=00,	SELECT DIAG00, DIAGNOSTIC COMMANDS
DUMP=DASD,	PLACE SVC DUMPS ON DASD DEVICES
FIX=00,	SELECT IEAFIX00, FIX MODULES SPECIFIED
<b>GRS=STAR,</b>	<b>MODIFIED</b>
<b>GRSRNL=00,</b>	<b>ADDED</b>
ILMMODE=NONE,	EXPLICITLY DISABLE LICENSE MANAGER
LNKAUTH=LNKLST,	AUTHORIZE LNKLST00, APFTAB IS ALTERNATE
LOGCLS=L,	WILL NOT BE PRINTED BY DEFAULT
LOGLMT=999999,	MAX WTL MESSAGES QUEUED, MUST BE 6 DIGITS
<b>LOGREC=LOGSTREAM,</b>	<b>COULD BE SET TO SYS2.LOGREC</b>
<b>LPA=01,</b>	1.13 AD-CD ERRATIC NAMING
MAXUSER=250,	SYS TASKS PLUS INITS PLUS TSUSERS
MLPA=00,	SELECT IEALPA00, MLPA PARAMETERS
MSTRJCL=00,	SELECT MSTJCLEX, MASTER JCL
OMVS=CS,	NOT MODIFIED
OPI=YES,	ALLOW OPERATOR OVERRIDE TO IEASYS00
<b>PAGE=(SYS2.PLPA.PAGE,</b>	
<b>    SYS2.COMMON.PAGE,</b>	
<b>    SYS2.LOCALX.PAGE,</b>	
<b>    SYS2.LOCALY,PAGE,L),</b>	
PAK=00,	SELECT IEAPAK00
PLEXCFG=ANY,	
PROG=01,	1.13 AD-CD ERRATIC NAMING
REAL=128,	ALLOWS 2 64K OR 1 128K JOB TO RUN V=R
RSU=0,	NO RECONFIG STORAGE UNITS
RSVNONR=100,	RESERVED ASVT ENTRIES
RSVSTRT=5,	RESERVED ASVT ENTRIES
SCH=00,	SELECT SCHED00
<b>SMF=Y1,</b>	<b>MODIFIED</b>
SQA=(15,64),	SQA SIZE APPROX 640K
SSN=00,	SELECT IEFSSN00, SUBSYSTEM NAMES
PROD=(00,01),	PLEASE NOTE IFAPRD01
SVC=00,	SELECT IEASVC00, USER SVCS
VAL=00,	SELECT VATLST00
<b>VIODSN=SYS2.STGINDEX,</b>	<b>MODIFIED</b>
VRREGN=64	DEFAULT REAL-STORAGE REGION SIZE

New member **IEASYSX2** is exactly the same as IEASYSX1 except:

- The CLPA line is removed,

- **CMD=X2**

New member **IEASYSY2** is exactly the same as IEASYSY1 except:

- The CLPA line is removed.

You might want to change the LOGREC parameter to LOGREC=IGNORE in all four IEASYSxx members if you are certain you will never use LOGREC data. Always remember the *comma* in IEASYSxx parameter lines!

## 3.2.2 Clock

The same CLOCK member is used by both S0W1 and S0W2. It invokes a simulated sysplex clock. A Parallel Sysplex requires an appropriate sysplex clock. However, this type of clock parameter is valid only under z/VM.

```
MEMBER NAME  CLOCKX1
OPERATOR NOPROMPT
TIMEZONE W.05.00.00
ETRMODE YES
ETRZONE YES
SIMETRID 00
ETRDELTA 10
STPMODE NO
STPZONE NO
```

## 3.2.3 COMMNDxx members

The COMMNDxx members control the JES2 startup mode, the VTAM startup parameters, and any VTAMAPPL started tasks.

```
MEMBER NAME  COMMNDX1
COM='S JES2,PARM='COLD,NOREQ','
COM='S VLF,SUB=MSTR'
COM='S VTAM,,,(LIST=X1)'
COM='S VTAMX1'
COM='S DLF,SUB=MSTR'
COM='DD ADD,VOL=HDSYS1'
COM='DD NAME=SYS1.&SYSNAME..DMP&SEQ'
COM='DD ALLOC=ACTIVE'
```

```
MEMBER NAME  COMMNDY1
COM='S JES2,PARM='WARM,NOREQ','
COM='S VLF,SUB=MSTR'
COM='S VTAM,,,(LIST=Y1)'
COM='S VTAMY1'
COM='S DLF,SUB=MSTR'
COM='DD ADD,VOL=HDSYS2'          <-- this could be on ZCSYS2 or elsewhere
COM='DD NAME=SYS1.&SYSNAME..DMP&SEQ'
COM='DD ALLOC=ACTIVE'
```

The COMMNDX2 member is exactly the same as COMMNDX1 except that the JES2 parameter is WARM instead of COLD. There is no COMMNDY2 member; the Y1 member is used again. With this setup, JES2 may be cold started only with the X1 (S0W1) option.

### 3.2.4 Consoles

Each member of a Parallel Sysplex must have uniquely named MVS consoles.

```
MEMBER NAME  CONSOLX1
INIT         CMDDELIM(“)
             MLIM(1500)
             MONITOR(DSNAME)
             MPF(00)
             PFK(00)
             RLIM(10)
             UEXIT(N)
             CNGRP(00)
DEFAULT ROUTCODE(ALL)
CONSOLE
  DEVNUM(700)
  AREA(NONE)
  AUTH(MASTER)
  CON(N)
  DEL(RD)
  MFORM(J,T,S)
  MONITOR(JOBNAME$-T)
  NAME(CXC1)
  PFKTAB(PFKTAB1)
  RNUM(19)
  ROUTCODE(ALL)
  RTME(1)
  SEG(14)
  UNIT(3270-X)
CONSOLE  DEVNUM(SYSCONS) LEVEL(ALL)
         NAME(HWCI) AUTH(MASTER) ROUTCODE(ALL)
CONSOLE
  DEVNUM(SUBSYSTEM)
  NAME(CXS1)
  AUTH(ALL)
HARDCOPY
  DEVNUM(SYSLOG,OPERLOG)
  CMDLEVEL(CMDS)
  ROUTCODE(ALL)
```

```
MEMBER NAME  CONSOLY1
INIT         CMDDELIM(“)
             MLIM(1500)
             MONITOR(DSNAME)
             MPF(00)
             PFK(00)
             RLIM(10)
             UEXIT(N)
             CNGRP(00)
DEFAULT ROUTCODE(ALL)
CONSOLE
  DEVNUM(700)
  AREA(NONE)
  AUTH(MASTER)
  CON(N)
  DEL(RD)
```

```

MFORM(J,T,S)
MONITOR(JOBNAME-T)
NAME(CYC1)
PFKTAB(PFKTAB1)
RNUM(19)
ROUTCODE(ALL)
RTME(1)
SEG(14)
UNIT(3270-X)
CONSOLE  DEVNUM(SYSCONS) LEVEL(ALL)
          NAME(HWCI) AUTH(MASTER) ROUTCODE(ALL)
CONSOLE
  DEVNUM(SUBSYSTEM)
  NAME(CYS1)
  AUTH(ALL)
HARDCOPY
  DEVNUM(SYSLOG,OPERLOG)
  CMDLEVEL(CMDS)
  ROUTCODE(ALL)

```

### 3.2.5 Couple data sets

The same couple data set definitions are used by SOW1 and SOW2. (Note that these are different couple data sets than are used by the non-sysplex AD-CD.)

```

MEMBER NAME COUPLEX1
COUPLE  SYSPLEX(&SYSPLEX)
        PCOUPLE(SYSPLEX.ADCDPL.XCF.CDS01)
        ACOUPLE(SYSPLEX.ADCDPL.XCF.CDS02)
        CLEANUP(15)
        MAXMSG(4096)
        RETRY(10)
DATA  TYPE(CFRM)
        PCOUPLE(SYSPLEX.ADCDPL.CFRM.CDS01)
        ACOUPLE(SYSPLEX.ADCDPL.CFRM.CDS02)
DATA  TYPE(LOGR)
        PCOUPLE(SYSPLEX.ADCDPL.LOGR.CDS01)
        ACOUPLE(SYSPLEX.ADCDPL.LOGR.CDS02)
DATA  TYPE(BPXMCD)
        PCOUPLE(SYSPLEX.ADCDPL.OMVS.CDS01)
        ACOUPLE(SYSPLEX.ADCDPL.OMVS.CDS02)
DATA  TYPE(WLM)
        PCOUPLE(SYSPLEX.ADCDPL.WLM.CDS01)
        ACOUPLE(SYSPLEX.ADCDPL.WLM.CDS02)
LOCALMSG MAXMSG(512) CLASS(DEFAULT)
PATHIN  STRNAME(IXC_DEFAULT_1,IXC_DEFAULT_2)
PATHOUT STRNAME(IXC_DEFAULT_1,IXC_DEFAULT_2)
CLASSDEF CLASS(BIG) CLASSLEN(62464) MAXMSG(4096) GROUP(UNDESIG)
PATHOUT STRNAME(IXC_DEFAULT_3) CLASS(BIG) MAXMSG(4096)
PATHIN  STRNAME(IXC_DEFAULT_3)

```

### 3.2.6 SMF parameters

Different SMF parameters are needed for SOW1 and SOW2. The following definitions allow you to use SMF data set recording or log stream recording, or switch between the modes.

```
MEMBER NAME SMFPRMX1
ACTIVE                               /* ACTIVE SMF RECORDING          */
RECORDING(LOGSTREAM)                 /* USE LOGSTREAM RECORDING      */
DEFAULTLSNAME(IFASMF.GENERAL)
DSNAME(SYS1.MAN1,
        SYS1.MAN2)
NOPROMPT                             /* DO NOT PROMPT OPERATOR      */
REC(PERM)                             /* TYPE 17 PERM RECORDS ONLY    */
MAXDORM(3000)                         /* WRITE IDLE BUFFER AFTER 30 MIN */
STATUS(010000)                       /* WRITE SMF STATS AFTER 1 HOUR  */
JWT(0400)                             /* 522 AFTER 4 hours            */
SID(ADCD)
LISTDSN                              /* LIST DATA SET STATUS AT IPL  */
SYS(NOTYPE(14:19,62:69),EXITS(IEFU83,IEFU84,IEFACTRT,
                              IEFUSI,IEFUJI,IEFU29),NOINTERVAL,NODETAIL)
SUBSYS(STC,EXITS(IEFU29,IEFU83,IEFU84,IEFUJP,IEFUSO))

MEMBER NAME SMFPRMY1
ACTIVE                               /* ACTIVE SMF RECORDING          */
RECORDING(LOGSTREAM)                 /* USE LOGSTREAM RECORDING      */
DEFAULTLSNAME(IFASMF.GENERAL)
DSNAME(SYS2.MAN1,
        SYS2.MAN2)
NOPROMPT                             /* DO NOT PROMPT OPERATOR      */
REC(PERM)                             /* TYPE 17 PERM RECORDS ONLY    */
MAXDORM(3000)                         /* WRITE IDLE BUFFER AFTER 30 MIN */
STATUS(010000)                       /* WRITE SMF STATS AFTER 1 HOUR  */
JWT(0400)                             /* 522 AFTER 4 hours            */
SID(BDCD)
LISTDSN                              /* LIST DATA SET STATUS AT IPL  */
SYS(NOTYPE(14:19,62:69),EXITS(IEFU83,IEFU84,IEFACTRT,
                              IEFUSI,IEFUJI,IEFU29),NOINTERVAL,NODETAIL)
SUBSYS(STC,EXITS(IEFU29,IEFU83,IEFU84,IEFUJP,IEFUSO))
```

Members SMFPRMX3 and SMFPRMY3 are exactly the same as the X1 and Y1 members except that ACTIVE is changed to NOACTIVE. These members are used to stop (and reset) SMF recording.

### 3.2.7 Other PARMLIB changes

The EXSPAT00 member can be set by an operator command. It deals with SPINLOOP timeouts.

```
MEMBER NAME EXSPAT00      (this member created)
SPINRCVY SPIN,ABEND
SPINTIME=30
```

Each z/OS must have its own system symbols defined.

```
MEMBER NAME IEASYMX1
SYSDEF SYSNAME(SOW1)
```

```

SYSCLONE(1X)
SYMDEF(&UNIXVER='Z113H')
SYMDEF(&SYSVER='Z113H')
SYMDEF(&ADCDLVL='ADCD113H')
SYMDEF(&SYSP1.='HDPRD1')
SYMDEF(&SYSP2.='HDPRD2')
SYMDEF(&SYSP3.='HDPRD3')
SYMDEF(&SYSR2.='HDRES2')
SYMDEF(&SYSS1.='ZDSYS1')

```

```

MEMBER NAME IEASYMY1
SYSDEF  SYSNAME(SOW2)
        SYSCLONE(1Y)
        SYMDEF(&UNIXVER='Z113H')
        SYMDEF(&SYSVER='Z113H')
        SYMDEF(&ADCDLVL='BDCD113H')
        SYMDEF(&SYSP1.='ZDPRD1')
        SYMDEF(&SYSP2.='ZDPRD2')
        SYMDEF(&SYSP3.='ZDPRD3')
        SYMDEF(&SYSR2.='ZDRES2')
        SYMDEF(&SYSS1.='ZDSYS1')      <== Do not change this

```

JES2 must be changed to a Multi-Access (MAS) configuration. The whole JES2PARM is not listed here. Only the changed lines for MAS are shown. This change also affects the non-sysplex ADCD operation, but has no unwanted effects there. (We also changed the number and classes of started initiators, but this is optional.)

```

MEMBER NAME JES2PARM (partial)
/*
                                *-----*
                                *   Multi-Access Spool   *
                                *-----*
                                */

MASDEF  SHARED=CHECK,
        RESTART=YES,CKPTLOCK=ACTION,
        DORMANCY=(0,100),
        HOLD=0,
        LOCKOUT=500

MEMBER(1) NAME=SOW1
MEMBER(2) NAME=SOW2

```

We modified VATLST to use ZDSYS2 and WORK01 as storage volumes. This is optional, but there should be more than one storage volume for the two z/OS systems.

```

MEMBER NAME VATLST00
VATDEF IPLUSE(PRIVATE),SYSUSE(PRIVATE)
ZDSYS1,0,0,3390    ,Y
ZDSYS2,0,0,3390    ,N
WORK01,0,0,3390    ,N

```

### 3.2.8 VTAMAPPL scripts

The following members are in PARMLIB (for convenient storage) although they are not actually part of MVS parameter library processing. They are command scripts that are executed by the VTAMAPPL program. Each of these scripts has a corresponding procedure in PROCLIB. For example, the SHUTX1 procedure in PROCLIB uses the SHUTX1 member in

PARMLIB. This association is not automatic and is not required, but it is consistently used in the AD-CD systems.

This is the shutdown command script for S0W1.

```
MEMBER NAME  SHUTX
CommandPrefix=None
F TSO,USERMAX=0
P TSO
C INETD4
C NFSC
C NFSS
C DFSCM
P LLA
SETRRS SHUTDOWN
P VLF
* C HTTPD1
MODIFY DLF,MODE=Q
P DLF
* %CSQ7 STOP CHINIT
* PAUSE 5
* %CSQ7 STOP QMGR
P TN3270
P TCPIP
P SDSF
Z NET,QUICK
PAUSE 10
F OMVS,SHUTDOWN
#STOP
```

We provided a SHUTY member for S0W2; it is exactly the same as SHUTX. This is to allow easy insertion of different shutdown options for the two z/OS systems.

The VTAMX and VTAMY startup command scripts start many standard z/OS functions. We removed HTTPD1 and SSHD from the active list. You might want to restore them (by removing the leading asterisk in the following list).

```
MEMBER NAME  VTAMX
CommandPrefix=None
PAUSE 10
S RRS,SUB=MSTR
S TSO
S IRRDPTAB
S SDSF
S EZAZSSI,P=S0W1
S TCPIP
PAUSE 5
S TN3270
S INETD
*S HTTPD1
S SSHD
D T
```

Member VTAMY1 (for S0W2) is exactly the same as VTAMX1 except for the S EZAZSSI,P=S0W2 command.



## 3.3 PROCLIB changes

A few PROCLIB changes were needed. These changes (or additions) were made in ADCD.Z113H.PROCLIB.

### 3.3.1 VTAMAPPL started tasks

A procedure exists for each of the VTAMAPPL command scripts that are stored in PARMLIB.

```
MEMBER NAME  SHUTX
//SHUTX PROC M=SHUTX,PREFIX=NONE  <== Change as required
//VTAMAPPL EXEC PGM=VTAMAPPL,PARM='COMMANDPREFIX=&PREFIX',
//          REGION=512K,TIME=1440
//STEPLIB DD DSN=ADCD.Z113H.LINKLIB,DISP=SHR
//PARMLIB DD DSN=ADCD.Z113H.PARMLIB(&M),DISP=SHR
//SYSABEND DD SYSOUT=H,HOLD=YES

MEMBER NAME  SHUTY
//SHUTY PROC M=SHUTY,PREFIX=NONE  <== Change as required
//VTAMAPPL EXEC PGM=VTAMAPPL,PARM='COMMANDPREFIX=&PREFIX',
//          REGION=512K,TIME=1440
//STEPLIB DD DSN=ADCD.Z113H.LINKLIB,DISP=SHR
//PARMLIB DD DSN=ADCD.Z113H.PARMLIB(&M),DISP=SHR
//SYSABEND DD SYSOUT=H,HOLD=YES

MEMBER NAME  VTAMX1
//VTAMX1 PROC M=VTAMX1,PREFIX=NONE  <== CHANGE AS REQUIRED
//VTAMAPDB EXEC PGM=VTAMAPPL,PARM='COMMANDPREFIX=&PREFIX',
//          REGION=512K,TIME=1440
//STEPLIB DD DSN=ADCD.Z113H.LINKLIB,DISP=SHR
//PARMLIB DD DSN=ADCD.Z113H.PARMLIB(&M),DISP=SHR
//SYSABEND DD SYSOUT=H,HOLD=YES

MEMBER NAME  VTAMY1
//VTAMY1 PROC M=VTAMY1,PREFIX=NONE  <== CHANGE AS REQUIRED
//VTAMAPDB EXEC PGM=VTAMAPPL,PARM='COMMANDPREFIX=&PREFIX',
//          REGION=512K,TIME=1440
//STEPLIB DD DSN=ADCD.Z113H.LINKLIB,DISP=SHR
//PARMLIB DD DSN=ADCD.Z113H.PARMLIB(&M),DISP=SHR
//SYSABEND DD SYSOUT=H,HOLD=YES
```

### 3.3.2 TCPIP procedure

The TCPIP procedure was changed to reference-specific TCPIP profiles and data for each of the two z/OS systems. This change also affects the non-sysplex AD system, and the same PROFILE/DATA is used by both the non-sysplex and the sysplex ADCD systems. The odd naming (such as ADCD113P for a profile) is due to the long length of the &SYSNAME symbolic value.

```
MEMBER NAME  TCPIP
//TCPIP PROC PARMS='CTRACE(CTIEZB00)'
//TCPIP EXEC PGM=EZBTCPIP,REGION=0M,TIME=1440,
//          PARM='&PARMS'
//*          PARM=('&PARMS',
```

```

/*          'ENVAR("RESOLVER_CONFIG=/'"TCPIP.TCPPARMS(TCPDATA)'")')
/*
//STEPLIB DD DSN=ADCD.&SYSVER.VTAMLIB,DISP=SHR
/*
//SYSPRINT DD SYSOUT=H,DCB=(RECFM=VB,LRECL=132,BLKSIZE=136)
//ALGPRINT DD SYSOUT=*,DCB=(RECFM=VB,LRECL=132,BLKSIZE=136)
//CFGPRINT DD SYSOUT=*,DCB=(RECFM=VB,LRECL=132,BLKSIZE=136)
//SYSOUT DD SYSOUT=H,DCB=(RECFM=VB,LRECL=132,BLKSIZE=136)
//CEEDUMP DD SYSOUT=*,DCB=(RECFM=VB,LRECL=132,BLKSIZE=136)
//SYSERROR DD SYSOUT=*
//PROFILE DD DISP=SHR,DSN=ADCD.&SYSVER.TCPPARMS(&SYSNAME.P)
//SYSTCPD DD DISP=SHR,DSN=ADCD.&SYSVER.TCPPARMS(&SYSNAME.D)

```

### 3.4 SYS1.IPLPARM changes

We added four members to SYS1.IPLPARM

MEMBER NAME	LOADX1	
IODF	99 SYS1	
SYSCAT	HDSYS1113CCATALOG.Z113H.MASTER	
SYSPARM	X1	
IEASYM	X1	
NUCLST	00	
PARMLIB	USER.PARMLIB	HDSYS1
PARMLIB	ADCD.Z113H.PARMLIB	HDRES1
PARMLIB	SYS1.PARMLIB	HDRES1
NUCLEUS	1	
SYSPLEX	ADCDPL	

Members LOADX2, LOADY1, and LOADY2 are exactly the same, except that X2, Y1, and Y2 are used in the SYSPARM statements. The IEASYM parameter is X1 for the X1 and X2 members, and it is Y1 for the Y1 and Y2 members.

### 3.5 TCPIP profiles and data

These members are in ADCD.Z113H.TCPPARMS. You may need to adjust these parameters to match your LAN requirements. *Do not change these member names* unless you also change the TCPIP JCL in PROCLIB.

```

MEMBER NAME SOW1P
ARPAGE 5
...
(all the normal profile statements are not listed here)
...
DEVICE PORTA MPCIPA
LINK ETH1 IPAQENET PORTA
HOME 10.1.1.2 ETH1

DEVICE PORTB MPCIPA
LINK ETH2 IPAQENET PORTB
HOME 192.168.0.61 ETH2

BEGINRoutes

```

```

;      Destination  SubNetMask    FirstHop    Link    Size
ROUTE 192.168.0.0 255.255.255.0    =          ETH1    MTU 1492
ROUTE 10.0.0.0   255.0.0.0          =          ETH2    MTU 1492
ROUTE DEFAULT                    192.168.0.1  ETH1    MTU 1492
ENDRoutes

```

ITRACE OFF

START PORTA  
START PORTB

This is the corresponding member for SOW2.

```

MEMBER NAME SOW2P
ARPAGE 5
...
(all the normal profile statements are not listed here)
...
DEVICE PORTA MPCIPA
LINK ETH1 IPAQENET PORTA
HOME 10.1.1.3 ETH1

DEVICE PORTB MPCIPA
LINK ETH2 IPAQENET PORTB
HOME 192.168.0.63 ETH2

BEGINRoutes
;      Destination  SubNetMask    FirstHop    Link    Size
ROUTE 192.168.0.0 255.255.255.0    =          ETH1    MTU 1492
ROUTE 10.0.0.0   255.0.0.0          =          ETH2    MTU 1492
ROUTE DEFAULT                    192.168.0.1  ETH1    MTU 1492
ENDRoutes

ITRACE OFF

START PORTA
START PORTB

```

The following two members are the corresponding TCPIP DATA members. If you connect your z/OS systems to an external network, the DOMAINORIGIN values in these members should be changed. Likewise, if you intend to use a name server, the NSINTERADDR values should be provided.

```

MEMBER NAME SOW1D
TCPIPJOBNAME TCPIP
HOSTNAME SOW1
DOMAINORIGIN XXX.YYY.COM
DATASETPREFIX TCPIP
;NSINTERADDR xx.xx.xx.xx
RESOLVEVIA UDP
RESOLVERTIMEOUT 10
RESOLVERUDPRETRIES 1
ALWAYSWHO NO

```

```

MEMBER NAME SOW2D
TCPIPJOBNAME TCPIP
HOSTNAME SOW2

```

```

DOMAINORIGIN   XXX.YYY.COM
DATASETPREFIX  TCPIP
;NSINTERADDR   xx.xx.xx.xx
RESOLVEVIA     UDP
RESOLVERTIMEOUT 10
RESOLVERUDPRETRIES 1
ALWAYSUTO      NO

```

### 3.6 VTAMLST changes

We removed unused members from the VTAM major node list and added TRL entries for our OSA OSD configuration. The X symbols on the right-hand side of some lines should be in column 72 of the statement.

```

MEMBER NAME  ATCCONOX
A0600,NSNA70X,DYNMODEL,COSAPPN,
AOTCP,OSATRLX,IMS11APL,IMS12APL,DB9GLU,CICSAPPL,DBAGLU

MEMBER NAME  ATCCON0Y
A0600,NSNA70X,DYNMODEL,COSAPPN,
AOTCP,OSATRLY,IMS11APL,IMS12APL,B9GLU,CICSAPPL,DBAGLU

MEMBER NAME  ATCSTRX1
CONFIG=0X,SUPP=NOSUP,
SSCPID=06,NOPROMPT,
HOSTSA=6,MAXSUBA=31,
SSCPNAME=SOW1SSCP,HOSTPU=SOW1$PU,
NETID=SOW1,
NODETYPE=NN,
CRPLBUF=(208,,15,,1,16),
IOBUF=(400,508,19,,1,20),
LFBUF=(104,,0,,1,1),
LPBUF=(64,,0,,1,1),
SFBUF=(163,,0,,1,1)

MEMBER NAME  ATCSTRY1
CONFIG=0Y,SUPP=NOSUP,
SSCPID=07,NOPROMPT,
HOSTSA=7,MAXSUBA=31,
SSCPNAME=SOW1SSCP,HOSTPU=SOW1$PU,
NETID=SOW1,
NODETYPE=NN,
CRPLBUF=(208,,15,,1,16),
IOBUF=(400,508,19,,1,20),
LFBUF=(104,,0,,1,1),
LPBUF=(64,,0,,1,1),
SFBUF=(163,,0,,1,1)

MEMBER NAME  OSATRLX
OSATRLX VBUILD TYPE=TRL
OSATRL1E TRLE LNCTL=MPC,READ=(0400),WRITE=(0401),DATAPATH=(0402),
PORTNAME=PORTA,
MPCLEVEL=QDIO

```

0

```

OSATRL2E  TRLE  LNCTL=MPC,READ=(0404),WRITE=(0405),DATAPATH=(0406), X
          PORTNAME=PORTB, X
          MPCLEVEL=QDIO

```

**MEMBER NAME OSATRLY**

```

OSATRLY  VBUILD TYPE=TRL

```

```

OSATRL3E  TRLE  LNCTL=MPC,READ=(0408),WRITE=(0409),DATAPATH=(040A), X
          PORTNAME=PORTA, X
          MPCLEVEL=QDIO

```

```

OSATRL4E  TRLE  LNCTL=MPC,READ=(040C),WRITE=(040D),DATAPATH=(040E), X
          PORTNAME=PORTB, X
          MPCLEVEL=QDIO

```

## 3.7 TCPIP RESOLVER

TCPIP automatically starts a domain name resolver. By default, it starts a procedure named RESOLVER (and this is in ADCD.Z113H.PROCLIB). The RESOLVER uses several small data sets containing parameters.

We modified the RESOLVER procedure as follows:

```

//RESOLVER  PROC  PARM=CTTRACE(CTIRES00)
//EZBREINI  EXEC  PGM=EZBREINI,REGION=OM,TIME=1440,PARM=&PARMS
//SETUP DD  DISP=SHR,CLOSE=FREE,DSN=ADCD.Z113H.TCPPARMS(&SYSNAME.RES)

```

We then created several new members in ADCD.Z113H.TCPPARMS:

**Member SOW1RES:**

```

DEFAULTTCPIPDATA('ADCD.Z113H.TCPPARMS(GBLTDAT1')
GLOBALTCPIPDATA('ADCD.Z113H.TCPPARMS(GBLTDAT1')
DEFAULTIPNODES('ADCD.Z113H.TCPPARMS(GBLIPNO1')
GLOBALIPNODES('ADCD.Z113H.TCPPARMS(GBLIPNO1')
COMMONSEARCH
CACHE
CACHESIZE(200M)
MAXTTL(2147483647)
UNRESPONSIVETHRESHOLD(25)

```

Member SOW2RES is exactly the same as SOW1RES except that the GBLTDAT1 and GBLIPNO1 values are changed to GBLTDAT2 and GBLIPNO2.

**Member GBLIPNO1:**

```

192.168.0.61  XXX.YYY.COM SOW1

```

**Member GBLIPNO2:**

```

192.168.0.63  XXX.YYY.COM SOW2

```

**Member GBLTDAT2:**

```

TCPIPJOBNAME TCPIP
SOW2: HOSTNAME SOW2
DOMAINORIGIN XXX.YYY.COM
: SEARCH YOUR.DOMAIN.NAME MY.DOMAIN.NAME
DATASETPREFIX TCPIP

```

```

: NSINTERADDR x.x.x.x
: NSPORTADDR 53
: RESOLVEVIA UDP
: RESOLVERTIMEOUT 30
: RESOLVERUDPRETRIES 1
: LOOKUP DNS LOCAL
: ALWAYSSTO NO

```

Member GBLTDAT1 is the same as GBLRDAT2 except that the HOSTNAME line contains S0W1 instead of S0W2. Also, all the original comments were left in this member.

## 3.8 Cross reference

Table 3-2 may be useful in relating key PARMLIB members to the IPL parameter that invokes that member.

Table 3-2 Relating key PARMLIB members to the IPL parameter that invokes that member

	monoplex S0W1		sysplex S0W1		sysplex S0W2	
IPL Parameter ----->	CS	00	X1	X2	Y1	Y2
SYSPARM (in LOADxx)	CS	01	X1	X2	Y1	Y2
IEASYM (in LOADxx)	00	00	X1	X1	Y1	Y1
PARMLIB CLOCKxx	00	00	X1	X1	X1	X1
PARMLIB CONxx	00	00	X1	X1	Y1	Y1
PARMLIB IEASYSxx	CS	01	X1	X2	Y1	Y2
PARMLIB COMMNDxx	CS	00	X1	X2	Y1	Y1
VTAMAPPL started	APPL	APPL	X1	X1	Y1	Y1
PARMLIB COUPLExx	00	00	X1	X1	X1	X1
PARMLIB GRS	none	none	<sup>a</sup>	STAR	STAR	STAR
PARMLIB LOGREC	SYS1	SYS1	SYS1	SYS1	SYS2	SYS2
PARMLIB OMVSxx	CS	CS	CS	CS	CS	CS
PARMLIB PAGE	SYS1	SYS1	SYS1	SYS1	SYS2	SYS2
PARMLIB SMFxx	00	00	X1	X1	Y1	Y1
PARMLIB VIODSN	SYS1	SYS1	SYS1	SYS1	SYS2	SYS2
TCPIP PROFILE	ADCD113P	ADCD113P	ADCD113P	ADCD113P	BDCD113P	BDCD113P
TCPIP DATA	ADCD113D	ADCD113D	ADCD113D	ADCD113D	BDCD113D	BDCD113D
PARMLIB SHUTDOWN	SHUT00	SHUT00	SHUTX	SHUTX	SHUTY	SHUTY
VTAM ATCSTRxx	00	00	X1	X1	Y1	Y1
VTAM ATCONxx	00	00	0X	0X	0Y	0Y

a. TRYJOIN when first bringing up the Parallel Sysplex, and STAR thereafter.

## 3.9 VM directory

We IPLed z/VM and added the following entries to the directory, and then activated the directory. The MVSDUMMY userid (which is never used for a logon) is needed in order to share the z/OS volumes among multiple other guests (ADCD and BDCD). The IBMSYS1 userid that is included is not required, but may be convenient for some users. The BASEAD guest definition may be used to IPL the AD-CD (without Parallel Sysplex functions) under z/VM. The BASEAD guest must not be used at the same time as the ADCD and/or BDCD guests.

```
*****
* FOLLOWING ADDED BY OGDEN *****
*****
*
*MVSDUMMY OWNS VOLUMES SO THEY CAN BE SHARED
*
USER MVSDUMMY NOLOG 1024M 1024M G
MDISK A80 3390 DEVNO A80 MWV
MDISK A81 3390 DEVNO A81 MWV
MDISK A82 3390 DEVNO A82 MWV
MDISK A83 3390 DEVNO A83 MWV
MDISK A84 3390 DEVNO A84 MWV
MDISK A85 3390 DEVNO A85 MWV
MDISK A86 3390 DEVNO A86 MWV
MDISK A87 3390 DEVNO A87 MWV
MDISK A88 3390 DEVNO A88 MWV
MDISK A89 3390 DEVNO A89 MWV
MDISK A8A 3390 DEVNO A8A MWV
MDISK A8B 3390 DEVNO A8B MWV
MDISK A8C 3390 DEVNO A8C MWV
MDISK A8D 3390 DEVNO A8D MWV
MDISK A8E 3390 DEVNO A8E MWV
MDISK A8F 3390 DEVNO A8F MWV
MDISK A90 3390 DEVNO A90 MWV
MDISK A91 3390 DEVNO A91 MWV
MDISK A92 3390 DEVNO A92 MWV
MDISK A93 3390 DEVNO A93 MWV
MDISK A94 3390 DEVNO A94 MWV
MDISK A95 3390 DEVNO A95 MWV
MDISK A96 3390 DEVNO A96 MWV
MDISK A97 3390 DEVNO A97 MWV
MDISK A98 3390 DEVNO A98 MWV
MDISK A99 3390 DEVNO A99 MWV
MDISK A9A 3390 DEVNO A9A MWV
MDISK A9B 3390 DEVNO A9B MWV
MDISK A9C 3390 DEVNO A9C MWV
MDISK A9D 3390 DEVNO A9D MWV
MDISK A9E 3390 DEVNO A9E MWV
DASDOPT WRKALLEG
MDISK A9F 3390 DEVNO A9F MWV
DASDOPT WRKALLEG
MDISK AA0 3390 DEVNO AA0 MWV
MDISK AA1 3390 DEVNO AA1 MWV
MDISK AA2 3390 DEVNO AA2 MWV
```

```

MDISK AA3 3390 DEVNO AA3 MWV
MDISK AA4 3390 DEVNO AA4 MWV
MDISK AA5 3390 DEVNO AA5 MWV
MDISK AA6 3390 DEVNO AA6 MWV
MDISK AA7 3390 DEVNO AA7 MWV
MDISK AA8 3390 DEVNO AA8 MWV
MDISK AA9 3390 DEVNO AA9 MWV
MDISK AAA 3390 DEVNO AAA MWV
MDISK AAB 3390 DEVNO AAB MWV
MDISK AAC 3390 DEVNO AAC MWV
MDISK AAD 3390 DEVNO AAD MWV
MDISK AAE 3390 DEVNO AAE MWV
MDISK AAF 3390 DEVNO AAF MWV
MDISK AB0 3390 DEVNO AB0 MWV
MDISK AB1 3390 DEVNO AB1 MWV
MDISK AB2 3390 DEVNO AB2 MWV
MDISK AB3 3390 DEVNO AB3 MWV
MDISK AB4 3390 DEVNO AB4 MWV
MDISK AB5 3390 DEVNO AB5 MWV
MDISK AB6 3390 DEVNO AB6 MWV
MDISK AB7 3390 DEVNO AB7 MWV
MDISK AB8 3390 DEVNO AB8 MWV
MDISK AB9 3390 DEVNO AB9 MWV
MDISK ABA 3390 DEVNO ABA MWV
MDISK ABB 3390 DEVNO ABB MWV
MDISK ABC 3390 DEVNO ABC MWV
MDISK ABD 3390 DEVNO ABD MWV
MDISK ABE 3390 DEVNO ABE MWV
MDISK ABF 3390 DEVNO ABF MWV
*
*
* BASEAD ALLOWS IPLING ADCD IN NON-SYSPLEX MODE
*
USER BASEAD BASEAD 4000M 4000M G
MACH ESA
CPU 0
SPOOL 00C 2540 READER *
SPOOL 00D 2540 PUNCH A
SPOOL 00E 1403 A
CONSOLE 700 3215 T
LINK MAINT 0190 0190 RR
LINK MAINT 019D 019D RR
LINK MAINT 019E 019E RR
SPECIAL 701 3270          <==These provide DIAL TSO sessions
SPECIAL 702 3270
SPECIAL 703 3270
SPECIAL 704 3270
SPECIAL 705 3270
SPECIAL 706 3270
DEDICATE 400 400          <==Note the OSA addresses
DEDICATE 401 401
DEDICATE 402 402
DEDICATE 404 404
DEDICATE 405 405
DEDICATE 406 406

```



```

LINK MVSDUMMY A80 A80 MW
LINK MVSDUMMY A81 A81 MW
LINK MVSDUMMY A82 A82 MW
    (More LINK statements are repeated for addresses A83 through A9D)
LINK MVSDUMMY A9E A9E MW      <==DISKOPT parameter not needed for LINK
LINK MVSDUMMY A9F A9F MW
    (More LINK statements to match MVSDUMMY)
*
*
* The SOW1 guest is the "1" SYSPLEX z/OS
*
USER SOW1 SOW1 4000M 4000M G
OPTION CFUSER TODENABLE
MACH ESA
CPU 0
SPOOL 00C 2540 READER *
SPOOL 00D 2540 PUNCH A
SPOOL 00E 1403 A
CONSOLE 700 3215 T
LINK MAINT 0190 0190 RR
LINK MAINT 019D 019D RR
LINK MAINT 019E 019E RR
SPECIAL 1400 MSGP CFCC1      <==These connect to Coupling Facilities
SPECIAL 1500 MSGP CFCC2
SPECIAL 701 3270             <==For DIAL TSO terminals
SPECIAL 702 3270
SPECIAL 703 3270
SPECIAL 704 3270
SPECIAL 705 3270
SPECIAL 706 3270
DEDICATE 400 400             <==OSA address for ADCD
DEDICATE 401 401
DEDICATE 402 402
DEDICATE 404 404
DEDICATE 405 405
DEDICATE 406 406
MDISK 191 3390 2300 5 610W02
LINK MVSDUMMY A80 A80 MW
LINK MVSDUMMY A81 A81 MW
LINK MVSDUMMY A82 A82 MW
    (More LINK statements are repeated for addresses A83 through A9D)
LINK MVSDUMMY A9E A9E MW
LINK MVSDUMMY A9F A9F MW
    (More LINK statements to match MVSDUMMY)
*
*
* The Sow2 guest is the sysplex "2" system
*
USER SOW2 SOW2 4000M 4000M G
OPTION CFUSER TODENABLE
MACH ESA
CPU 0
SPOOL 00C 2540 READER *
SPOOL 00D 2540 PUNCH A
SPOOL 00E 1403 A

```

```

CONSOLE 700 3215 T
LINK MAINT 0190 0190 RR
LINK MAINT 019D 019D RR
LINK MAINT 019E 019E RR
SPECIAL 1400 MSGP CFCC1
SPECIAL 1500 MSGP CFCC2
SPECIAL 701 3270
SPECIAL 702 3270
SPECIAL 703 3270
SPECIAL 704 3270
SPECIAL 705 3270
SPECIAL 706 3270
DEDICATE 408 408          <==Note different OSA addresses for BDCD
DEDICATE 409 409
DEDICATE 40A 40A
DEDICATE 40C 40C
DEDICATE 40D 40D
DEDICATE 40E 40E
MDISK 191 3390 2305 5 610W02
LINK MVSDUMMY A80 A80 MW
LINK MVSDUMMY A81 A81 MW
LINK MVSDUMMY A82 A82 MW
      (More LINK statements are repeated for addresses A83 through A9D)
LINK MVSDUMMY A9E A9E MW
LINK MVSDUMMY A9F A9F MW
      (More LINK statements to match MVSDUMMY)
*
*
*CFCC1 IS FIRST COUPLING FACILTY
*
USER CFCC1 CFCC1 1024M 1024M G
XAUTOLOG CFCONSOL
OPTION CFVM TODENABLE
MACH ESA
CONSOLE 009 3215 T CFCONSOL
*
*
*CFCC2 IS SECOND COUPLNG FACILITY
*
USER CFCC2 CFCC2 1024M 1024M G
XAUTOLOG CFCONSOL
OPTION CFVM TODENABLE
MACH ESA
CONSOLE 009 3215 T CFCONSOL
*
*
*CFCONSOL PROVIDES SHARED CONSOLE FOR CF'S
*

USER CFCONSOL CFCONSOL 4M 4M ABCDEFG
MACH ESA
CONSOLE 009 3215 T
*
*
* User BILL is to illustrate adding a simple VM userid

```

```

*
USER BILL W2W0 128M 128M G
MACH ESA
CPU 0
SPOOL 00C 2540 READER *
SPOOL 00D 2540 PUNCH A
SPOOL 00E 1403 A
CONSOLE 700 3215 T
LINK MAINT 0190 0190 RR
LINK MAINT 019D 019D RR
LINK MAINT 019E 019E RR
MDISK 191 3390 1667 10 VMCOM2 MR    <--- note cylinder address
*
*****
*****END OF OGDEN ADDITIONS*****
*****

```

Notice that the z/OS guest userids do not have a MDISK 191 statement; that is, they do not have a CMS “A” disk. User BILL does have a MDISK 191 statement. The cylinder address used (1667) was appropriate for the base z/VM we used but is unlikely to be correct for any other z/VM system. The cylinder number (1667) was the next available cylinder in a USER DISKMAP listing. It was placed on the VMCOM2 volume because there was considerable free space on this volume. User BILL is included solely as an example; it is not needed or used when running Parallel Sysplex operations.

## 3.10 DEVMAPs

This devmap (named devmap1 on the DVD) contains all the definitions needed for z/VM and the z/OS systems. You may want to change the processors statement to reflect one or three processors, depending on your PC configuration.

```

[system]
memory 10000m
3270port 3270
processors 3
command 2 x3270 localhost:3270
command 2 x3270 -geometry +1100+100 localhost:3270

[manager]
name aws3274 0002
device 0700 3279 3274
device 0701 3279 3274
device 0702 3279 3274
device 0703 3279 3274
device 0704 3279 3274
device 0705 3279 3274
device 0706 3279 3274
device 0707 3279 3274
device 0708 3279 3274
device 0709 3279 3274
device 070B 3279 3274
device 070C 3279 3274
device 070D 3279 3274
device 070E 3279 3274

```

```

device 070F 3279 3274
#
# VM volumes
#
[manager]
name awsckd 0101
device 0200 3390 3990 /z/M01RES
device 0201 3390 3990 /z/610RL1
device 0202 3390 3990 /z/610RL2
device 0203 3390 3990 /z/M01W01
device 0204 3390 3990 /z/M01S01
device 0205 3390 3990 /z/M01SP1
device 0206 3390 3990 /z/VMCOM1
device 0207 3390 3990 /z/VMCOM2
#
# z/OS volumes
#
[manager]
name awsckd 0001
device 0A80 3390 3990 /z/HDRES1
device 0A81 3390 3990 /z/HDRES2
device 0A82 3390 3990 /z/HDSYS1
device 0A83 3390 3990 /z/HDUSS1
device 0A84 3390 3990 /z/HDUSS2
device 0A85 3390 3990 /z/HDPRD1
device 0A86 3390 3990 /z/HDPRD2
device 0A87 3390 3990 /z/HDPRD3
device 0A90 3390 3990 /z/HDPAGA
device 0A91 3390 3990 /z/HDPAGB
device 0A92 3390 3990 /z/HDPAGY
device 0A93 3390 3990 /z/HDPAGZ
device 0A95 3390 3990 /z/WORK01
device 0A94 3390 3990 /z/HCSYS2
device 0A9C 3390 3990 /z/SARES1
device 0A9F 3390 3990 /z/CF0001 #Must be A9E or A9F to match VM

[manager]
name awsosa 0019 --path=A0 --pathtype=OSD --tunnel_intf=y
device 400 osa osa #Tunnel for SOW1
device 401 osa osa #Tunnel for SOW1
device 402 osa osa #Tunnel for SOW1
device 408 osa osa #Tunnel for SOW2
device 409 osa osa #Tunnel for SOW2
device 40A osa osa #Tunnel for SOW2

[manager]
name awsosa 0009 --path=F0 --pathtype=OSD
device 404 osa osa #Ethernet for SOW1
device 405 osa osa #Ethernet for SOW1
device 406 osa osa #Ethernet for SOW1
device 40C osa osa #Ethernet for SOW2
device 40D osa osa #Ethernet for SOW2
device 40E osa osa #Ethernet for SOW2

```

## Frequently asked questions

**Q:** What is the purpose of VM guest BASEAD?

**A:** You may need to make adjustments to z/OS in order to fix a problem in Parallel Sysplex mode. One way to do this is to stop z/VM and then IPL the base z/OS (using an IPL parameter such as CS), and use this base z/OS to correct parameters necessary for Parallel Sysplex operation. You would then start z/VM again and bring up the Parallel Sysplex system. Instead of stopping z/VM in order to run the base z/OS, you can run it under z/VM guest BASEAD; this eliminates the need to stop z/VM and bring it up again later. Operation requires logging onto z/VM user BASEAD and using the TERM CONMODE 3270 and IPL commands. In this case you would *not* use IPL parameters X1, X2, Y1, or Y2. You would probably use IPL parameter 00 or CS.

**Q:** How do I IPL the SARES1 volume?

**A:** With our sample devmap it is mounted at A9C. The command is `ip1 A90 parm 0A9CSA`. Use `S SHUTSA` to shut down the SARES1 system.

**Q:** I am running S0W1 as a member of my Parallel Sysplex. I need to shut down the Parallel Sysplex and IPL basic AD-CD (perhaps using the BASEAD guest). Do I need to do a cold start? Likewise, when switching back to Parallel Sysplex mode, will I need a cold start?

**A:** With the AD-CD systems, a cold start means CLPA and JES2 cold start. We found we could switch between S0W1 (in basic mode) and S0W1 (as a Parallel Sysplex member) without doing a cold start.<sup>1</sup> During IPL a message appears asking if the existing LPA paging data set should be used; reply `continue` to this message.

**Q:** Why do you use awkward names such as S0W1? Why not simply use A and B?

**A:** The standard AD-CD z/OS 1.13H release uses the “S0W1” form and we did not want to change more than was required for sysplex operation.

**Q:** Why is the first part of the z/OS IPL sequence a little slow under z/VM? After this portion is complete, z/OS seems to run at a more normal speed under z/VM.

**A:** This is due to memory initialization and management functions being initialized through multiple virtualization paths (VM, SIE, Linux). The time seems related to the defined z/OS guest memory size and disk paging performance. The effect is less apparent when larger PC memory is available.

<sup>1</sup> This assumes that the same LPA options are used. In the starter system this is true if the basic AD-CD is IPLed with a 00 IPL parameter, or initially IPLed with a CS parameter.

**Q:** Can I make a sysplex system using CTCs?

**A:** Yes. You can make a basic sysplex system using CTCs instead of a Coupling Facility. A basic sysplex and a Parallel Sysplex are two different designs, with many common elements. This document (and the *System z Coupling Application Developer Controlled Distribution* downloads) are concerned only with Parallel Sysplex systems. Most System z customers today are oriented toward Parallel Sysplex systems because they offer more functionality than basic sysplex.

**Q:** Is a Parallel Sysplex system practical on a laptop?

**A:** The implementation described in this book was on a Lenovo W520 ThinkPad with 16 GB of memory, working under openSUSE 12.2. This is quite practical for reasonably small tasks.

**Q:** I want to connect my complete Parallel Sysplex to the company LAN so that my coworkers can access it. How should I do this?

**A:** There is no simple answer. We suggest you study Chapter 3 in the IBM Redbooks SG24-7722 publication. This includes a discussion of Linux NAT functions for external IP addressing plus the use of fixed IP addresses.

**Q:** How many “local” 3270 sessions can I use with zPDT?

**A:** At the time of writing, the limit is 32 sessions with the aws3274 device manager. There is no particular limit on the number of sessions through OSA and z/OS TCPIP.

**Q:** I want to place DB2 buffers in the Coupling Facilities. Also, I want to put the JES2 checkpoint data and RACF® data there. How do I do this?

**A:** This is beyond the scope of this document; you need to study the appropriate manuals and/or seek help from a systems programmer with experience in this area. We intentionally limited our project to a very basic implementation in order to reduce complexity as much as possible.

**Q:** Can I run z/VM, two z/OS guest machines, and a CP guest machine with only one PC processor (“core”) and one zPDT CP?

**A:** Yes. z/VM shares the processor with all the guest systems. However, this configuration may produce timeouts within z/OS. We consider it below the minimum level for practical usage.

**Q:** Can I use 3270 sessions on other PCs? Your example has all the sessions on the zPDT machine.

**A:** Yes, of course. Simply point your 3270 emulator (on your external PC) to the Linux IP address (192.168.0.60, in our example) and port 3270. We strongly suggest using a relatively modern 3270 emulator, such as recent versions of PCOMM or x3270. Older, “free” 3270 emulators have created problems during some of the zPDT test cycles.

**Q:** Can I use VSWITCH instead of the separate OSA interfaces you show?

**A:** Probably, but we did not try it.

**Q:** I am unable to compress the ADCD PARMLIB. It always says the data set is busy.

**A:** This is not related to Parallel Sysplex usage, but is a characteristic of some releases of the z/OS AD-CD system. You can issue F OMVS,STOPPFS=ZFS to free PARMLIB for compression. (You need to do this on both z/OS systems, if both are active.) This is best done on a single system just before shutting it down. Another way to compress PARMLIB and PROCLIBs is to run under the SARES1 system and address the target PARMLIB and PROCLIBs on their specific volsters.

**Q:** Why are the z/VM volumes at addresses 2xx?

**A:** No particular reason. z/VM adapts to whatever addresses are used for its volumes. We elected to use addresses that have no meaning to z/OS.

**Q:** I usually just crash my zPDT system (possibly with the `awsstop` command). Can I continue to do this with the Parallel Sysplex running?

**A:** You can do it, but starting the Parallel Sysplex again may be painful. We strongly suggest you follow the shutdown procedure we listed, or something similar. In particular, use the `V XCF,xxx,OFFLINE` command to stop additional z/OS members of the sysplex.

**Q:** You included RRS structures in the Coupling Facilities. How do I use these?

**A:** Setting up and using RRS is typically part of a larger environment, with DB2, CICS, and so forth. This is beyond the scope of this document. We included the RRS structures as a convenience for future expansion.

**Q:** I notice that you have three processors defined in your sample devmap, but only one processor defined for each z/OS guest machine under z/VM. Is this configuration needed?

**A:** No. However, we believe it is a reasonable configuration. We did not experiment with multiple processors defined to each z/OS guest.

**Q:** Could I run Linux for System z in this environment?

**A:** Linux for System z does not participate in Parallel Sysplex operation and does not use Coupling Facility functions. You could run Linux for System z as another z/VM guest and communicate with z/OS guests through TCP/IP.

**Q:** The distributed z/OS 1.13H AD-CD has more paging volumes than your system. Why was this changed?

**A:** We felt that two paging volumes per z/OS was ample for most usage and we wanted to keep the total number of emulated DASD volumes to a reasonable number.

**Q:** Can I have some or all of the 3390 volumes on a USB drive?

**A:** Yes, although it *might* be a bit slower than an internal PC drive.

**Q:** What are the next steps? I want to run DB2 and WebSphere in Parallel Sysplex modes.

**A:** We suggest more education is the next step. Your goals are beyond what we can provide in a simple cookbook distribution. Always plan your memory usage to minimize paging at all levels. Also, be aware of the WebSphere Application Server performance limitations on these small systems. We question whether running multiple z/OS WebSphere instances would be practical.

**Q:** Your z/OS uses the default WLM. How do I install a more advanced WLM?

**A:** To keep our installation process as basic as possible, we did not want to get into WLM discussions. You can use the `IWMARIN0` program (issue this command while in ISPF option 6) to try to install WLM definitions.

**Q:** Your z/VM directory definitions list many DASD devices in the definitions for `MVSDUMMY`, `S0W1`, `S0W2`, and `BASEAD`. Must I have all these volumes present to use the system?

**A:** No. You may have DASD definitions in the directory entries for `MVSDUMMY`, `BASEAD`, `S0W1`, and `S0W2` that are not present on your system. These devices or addresses could be used for future expansion, such as adding your own local volumes. However, you normally have all the volumes that are defined in your devmap.

**Q:** Why does your sample devmap say that the `CF0001` volume must be at address `A9F`?

**A:** The z/VM directory for our base z/OS volumes has a special parameter with addresses `A9E` and `A9F` that is used with volumes containing coupling data sets. You can change this configuration, of course.<sup>2</sup>

**Q:** If I try to log on as `IBMUUSER` on both z/OS systems, the second login cannot find the ISPF profile data set. A subsequent login as `IBMUUSER` on the first z/OS system also fails to find the

<sup>2</sup> If you happen to know *exactly* what this special parameter does, please send us a note!

ISPF profile. How can I recover this?

**A:** The login on the second z/OS (while logged in with the same userid on the first z/OS) apparently uncatalogs the ISPF profile. On a subsequent TSO login try this:

```
READY
alloc da('ibmuser.ispf.ispprof') f(ispprof) shr vol(hdsys1) unit(3390)
READY
ispf
```

Once you have ISPF access again, go to volume HDSYS1 (or whatever is appropriate on your system) and recatalog the `ibmuser.ispf.ispprof` data set. We suggest you do not log onto TSO with the same userid on multiple systems unless you manage the ISPF profile data set in a compatible manner.

**Q:** Why did you use openSUSE 12.2 for the base Linux? Is this required?

**A:** It is not required. We use a variety of SUSE and Red Hat base systems.

**Q:** Why did you define the two CF guests with 1024 MB of memory?

**A:** No particular reason; it was an arbitrary choice. You can change to z/VM directory definitions to match your needs. However, we suggest you keep the total guest sizes (the two z/OS guests and the two CF guests and whatever other guests you define) less than the zPDT size (in the devmap) and this should be less than the real PC memory size. The goal is to avoid paging whenever possible by Linux or z/VM.

**Q:** You frequently mention that paging should be avoided. Can I install more PC disks to reduce bottlenecks in this area?

**A:** We do not know. We doubt that another disk in a laptop or USB port would have much effect. Additional SCSI drives on an xServer, especially on multiple SCSI adapters, would probably help. We would be interested in any documented experiences in this area.

**Q:** Can I use your z/VM for other purposes? Is there anything special about it?

**A:** There is nothing special about our z/VM 6.2 for the “starter system,” other than the obvious directory definitions for S0W1, S0W2, CFCC1, and so forth. We changed the z/VM logo slightly and specified device number 0700 as the console, but these are optional changes. However, at the time of writing, some of the new and advanced features of z/VM 6.2 have not been tested under zPDT systems.

**Q:** I made minor changes to the CF structures and rebuilt the coupling data sets according to your instructions. I can start S0W1 correctly (with XCF, and so forth) but S0W2 simply hangs. If I shut down S0W1, then S0W2 startup works. Why?

**A:** Try making the CF memory larger by changing the VM guest definitions and rebuilding the VM directory.



# Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

## IBM Redbooks

For information about ordering these publications, see “How to get Redbooks” on page 63. Note that some of the documents referenced here may be available in softcopy only. Be certain to use the latest editions of these documents.

- ▶ *IBM System z Personal Development Tool: Volume 1 Introduction and Reference*, SG24-7721
- ▶ *IBM System z Personal Development Tool: Volume 2 Installation and Basic Use*, SG24-7722
- ▶ *IBM System z Personal Development Tool: Volume 3 Additional Topics*, SG24-7723
- ▶ *System Programmer's Guide to: z/OS System Logger*, SG24-6898
- ▶ *SMF Logstream Mode, Optimizing the New Paradigm*, SG24-7919

## Other publications

These publications are also relevant as further information sources:

- ▶ *MVS Setting Up a Sysplex*, SA22-7625

## How to get Redbooks

You can search for, view, or download Redbooks, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks publications, at this website:

[ibm.com/redbooks](http://ibm.com/redbooks)

## Help from IBM

IBM Support and downloads

[ibm.com/support](http://ibm.com/support)

IBM Global Services

[ibm.com/services](http://ibm.com/services)



# Index

## Numerics

1090 tokens 2  
1091tokens 2  
3270 sessions 60  
3270 sessions, more 20  
3390 volumes, adding 19

## A

ACDS5 job 35  
alcckd command 20, 30  
awsstart devmap1 12  
awsstop command 61

## B

BASEAD guest 59  
basic sysplex system 60  
BPXI070E USE SETOMVS.... 17

## C

CF guests, size 62  
CF memory 62  
CF structures 62  
CF0001 volume address 61  
CFCC program 6  
CFCC1 13  
CFCC1 DISPLAY MODE 21  
CFCC2 13  
CFCC2 DISPLAY RESOURCES 21  
CFCONSOL 2, 13  
CFCONSOL guest 2  
CFRMPOL3 job 18  
CLOCK member 41  
CLOCK=X1 39  
CMD=X1 39  
COMMNDxx members 41  
compress PARMLIB 60  
Consoles, MVS 42  
Couple datasets 43  
COUPLE=X1 39  
COUPLE2 job 18, 31  
cpuopt zVM\_CouplingFacility 2  
CRFMPOL3 job 33  
CTC usage 60

## D

D SMF command 16  
D XCF 22  
D XCF,STR command 14  
DASDOPT WRKALLEG 3  
DB2 61  
DB2 buffers 60  
devmap statements 2

devmap, edit 11  
devmap, sample 61  
DEVMAPs 57  
DHCP addresses 10  
directxa 20  
domain name resolver 51

## G

gnome workspace 12  
GRS 8  
GRS=STAR 39  
GRSRNL=00 39

## H

hardware required 3  
HFS changes 36  
HFS4 job 36

## I

IBMUSER password 12  
IEASYSxx members 39  
IEBCOPY program 8  
IP addresses 9  
IPL parameters 11  
IPL sequence, speed 59  
IPL, cold start 59  
ISMF panels 34  
ISPF profile 61

## J

JES2 checkpoint data 60  
JES2 parms 45

## L

LAN and TCP/IP configuration 9  
laptop usage 60  
LATER6 job 36  
Linux for System z 61  
Linux NAT functions 60  
local 3270 sessions 60  
LOG O command 15  
LOG S command 15  
LOGGER datasets 19  
LOGREC 8  
LOGREC data 15  
LOGREC, start in LOGGER 19  
LOGS6 job 19

## M

MAS JES2 8  
memory size, PC 3

## N

naming conventions 29  
NAT functions 60  
ntel-compatible computer 6

## O

OMVS=CS 39  
openSUSE 12.1 62  
OPERLOG 8, 15  
OPERLOG operation, start 19  
OPTION CFUSER TODENABLE 3  
OPTION CFVM TODENABLE 2

## P

PAGE= 39  
paging data sets 21  
paging volumes 5, 61  
paging, disks 62  
PARMLIB changes 39  
PARMLIB, compress 18  
password for IBMUSER 12  
PC memory, partition 3  
performance notes 4  
PROCLIB changes 47

## R

RACF® data 60  
Rebuilding coupling datasets 18  
Red Hat base system 62  
Redbooks Web site 63  
    Contact us viii  
RESOLVER procedure 51  
retention periods 19  
RRS structures 61

## S

SARES1 volume 5  
SETGRS MODE=STAR 19  
SETXCF START 19  
shared spool (MAS) 8  
Shutdown 17  
SHUTX1 script 46  
SMF 8  
SMF recording 15  
SMF, start in LOGGER 19  
SMF=X1 39  
SMS changes 34  
starter system 5  
STORAGE disk space 21  
SYS1.IPLPARM changes 48  
SYS1.SCDS 35  
SYSLOG 15  
sysplex clock 41  
system symbols 44

## T

TCP/IP configuration 9

TCP/IP customization for z/VM 11  
TCPIP procedure 47  
TCPIP profiles and data 48  
TCPIP Resolver 51  
TSO logon 62

## U

USB drive 61  
user direct c 20  
USER.PARMLIB 39  
USS file systems 8

## V

V XCF,BDCD,OFFLINE 17  
V XCF,xxx,OFFLINE 61  
VATLST parm 45  
VATLST00 member 21  
VM Directory 53  
VSAM1 job 30  
VSWITCH 60  
VSWITCH functions 10  
VTAMAPPL scripts 45  
VTAMAPPL started tasks 47  
VTAMLST changes 50  
VTAMLST definitions 10

## W

WebSphere 61  
WLM, default 61

## X

X1 and Y1 IPL parameters 29  
X2 and Y2 IPL parameters 29  
x3270 12  
xosview program 21

## Z

z/VM 6, as included 62  
z/VM 6.1 volumes 5  
z/VM directory entry 3  
z/VM overhead 4  
z/VM paging volumes 21  
z/VM usage 4  
z/VM volume addresses 60  
zPDT release level 6  
zVM\_CouplingFac option 2









# IBM System z

## Personal Development Tool Vol. 4

### Coupling and Parallel Sysplex



**Redbooks®**

#### **Basic use of Coupling Facility functions with zPDT**

This IBM Redbooks publication describes the usage of Coupling Facility (CF) functions with the IBM System z Personal Development Tool (zPDT). It describes the System z Coupling Application Developer Controlled Distribution, which is a Parallel Sysplex “starter system” based on the AD-CD package and lists the exact steps taken to turn the normal AD-CD z/OS system into a Parallel Sysplex base.

#### **Using the Parallel Sysplex AD-CD starter system**

This document assumes that the reader is familiar with basic zPDT usage and terminology, with z/OS, with the z/OS AD-CD system, with basic z/VM usage, and with general Parallel Sysplex concepts. It is not intended as an introduction to any of these topics.

#### **Details for building the starter system**

### **INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION**

### **BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE**

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

**For more information:**  
[ibm.com/redbooks](http://ibm.com/redbooks)