



# Taming the Beast – Best Practices for zFS with CICS

[Phil\\_Wakelin@uk.ibm.com](mailto:Phil_Wakelin@uk.ibm.com)

*CICS Strategy & Design, IBM Hursley UK*



SHARE is an independent volunteer-run information technology association that provides education, professional networking and industry influence.

Copyright (c) 2014 by SHARE Inc. Except where otherwise noted, this work is licensed under <http://creativecommons.org/licenses/by-nc-sa/3.0/>



# Disclaimer

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.



# Why zFS?

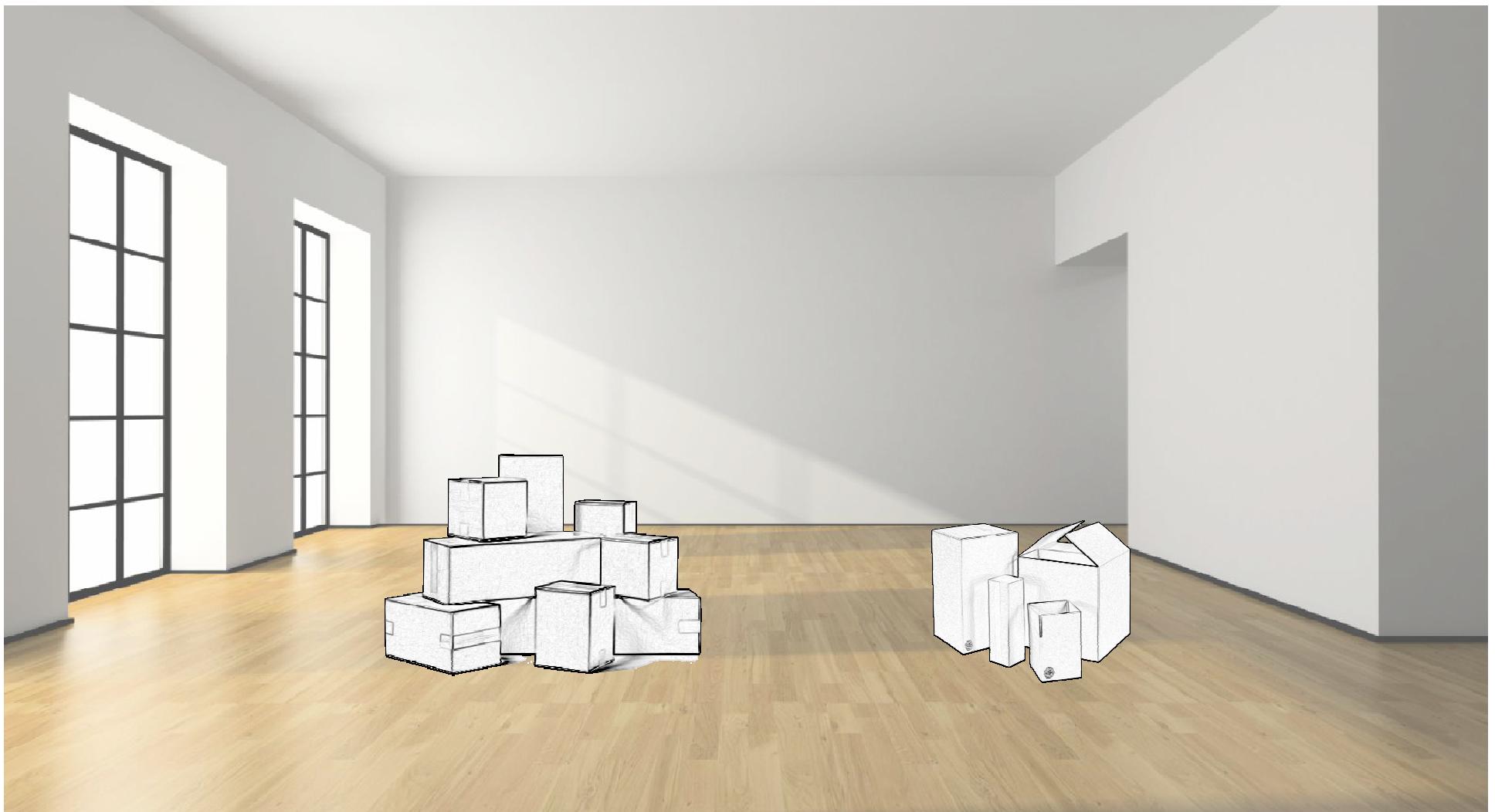
Complete your session evaluations online at [www.SHARE.org/Seattle-Eval](http://www.SHARE.org/Seattle-Eval)



# The Problem...





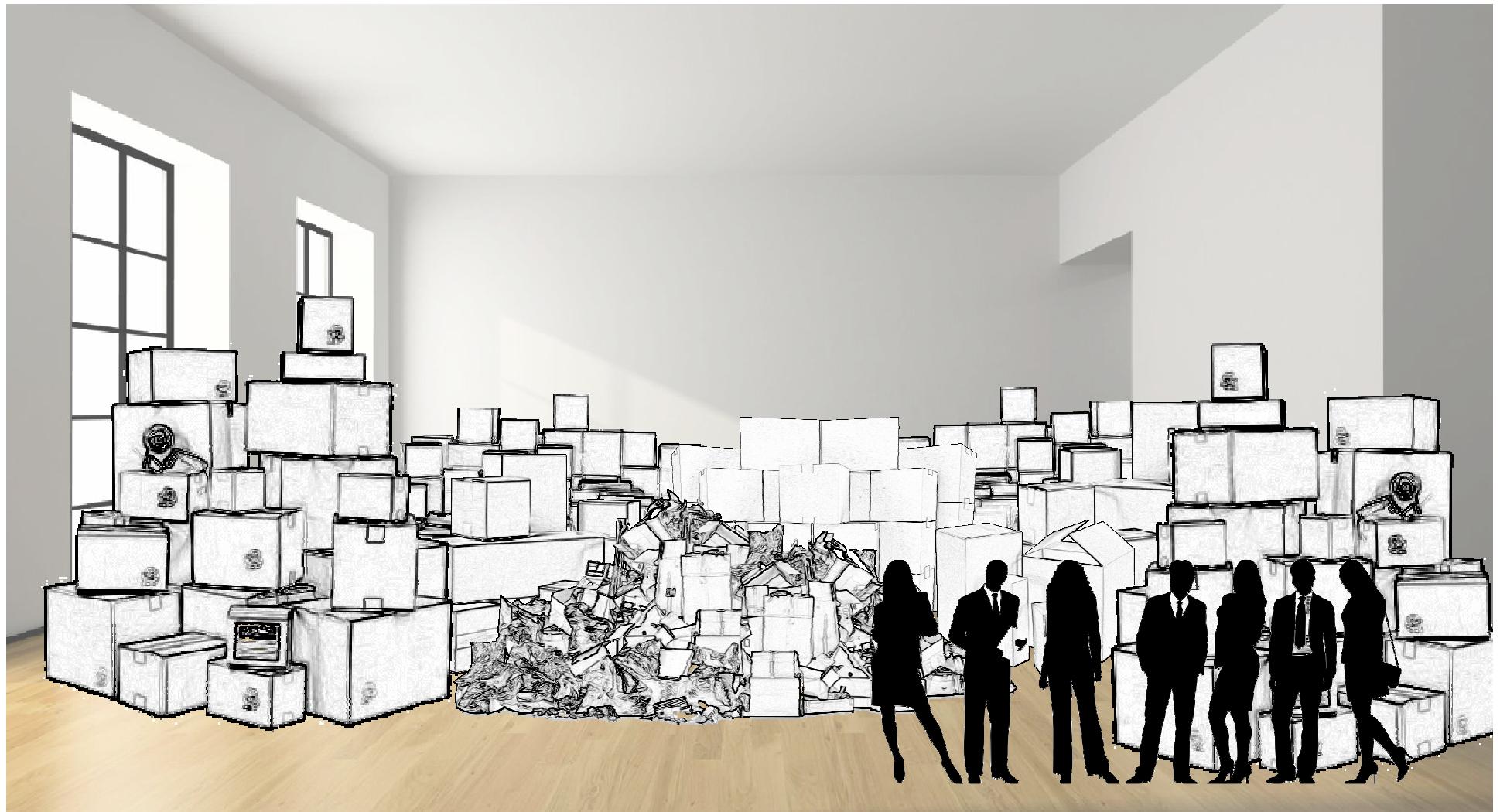


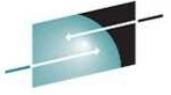




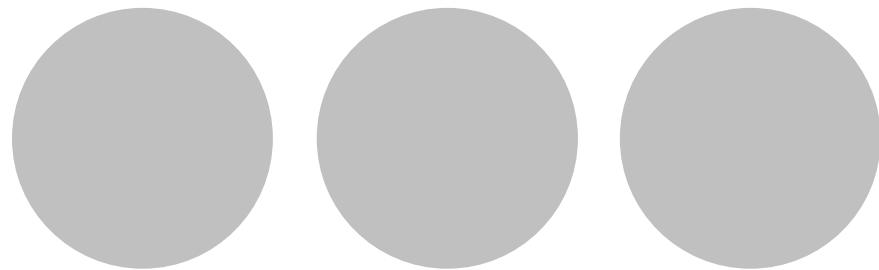








A place for  
everything



# Types of CICS zFS files

## ⬇️ Install

- CICS stuff provided by IBM
- CICS Programs and Samples
- May change during APARs

# Types of CICS zFS files

## Install

- CICS stuff provided by IBM
- CICS Programs and Samples
- May change during APARs

## Critical

- Things that define your applications and environments
- Programs, Properties and Config files etc
- Should be considered critical and be recoverable

# Types of CICS zFS files



## Install

- CICS stuff provided by IBM
- CICS Programs and Samples
- May change during APARs



## Critical

- Things that define your applications and environments
- Programs, Properties and Config files etc
- Should be considered critical and be recoverable



## Temp

- Things that applications create or write to at runtime
- Non critical, messages and logs files etc

# Where should they go?

## Install

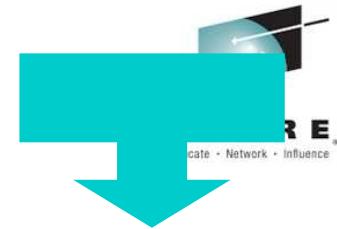
- Default: /usr/lpp/cicsts52
- SIT Parm: USSHOME

## Critical

- Suggest using /var/cicsts/...
- May have sub directories per CICSplex or region
- Use directories to group common assets eg all web files

## Temp

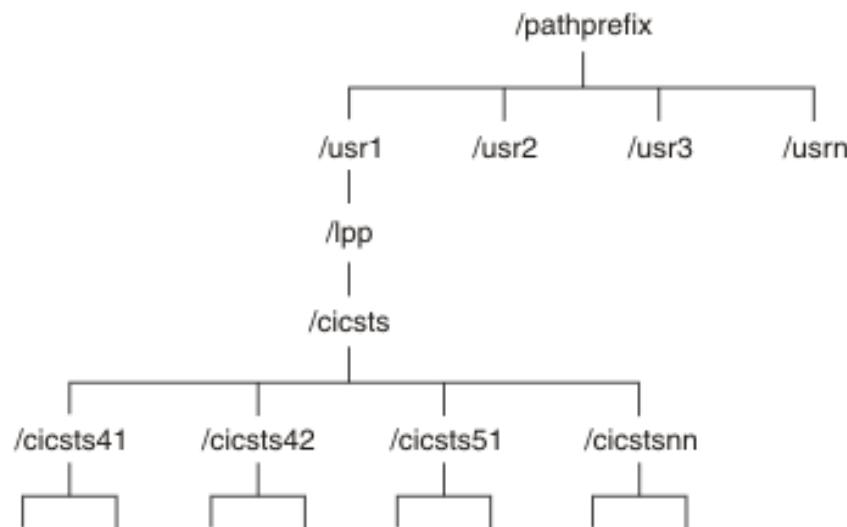
- Suggest using a separate place eg /cicslogs/<APPLID>...
- Typically have one per region
- May need to clean out periodically



# Setup for Install Dir

Create release specific directory structure

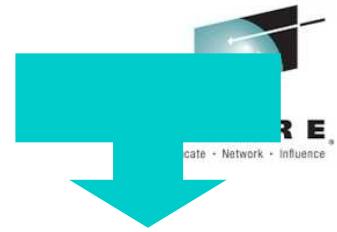
- /usr/lpp/cicsts is created ONCE for all releases (read only)
- Depending on SMP/E target zone structure
- You may create variants of /cicsts52





# Setup for Install Dir

- Directories under /cicsts52
  - docs
  - IBM
  - JVMPROFILES
  - lib
  - pipelines
  - samples
  - schemas
  - ...



## Setup for Install Dir

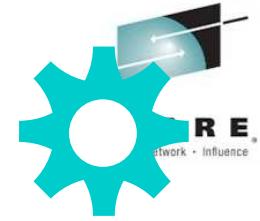
- SITParms
  - USSHOME
    - The name and path of the root directory for CICS files
    - Default: /usr/lpp/cicsts/cicsts52



# Setup for Critical Dir

- Create data set for usage as /var/cicsts
- If sharing zFS across a sysplex
  - Mount data set onto root filing system as /cicsts as a r/w filing system
  - On each LPAR create symbolic link to link /var/cicsts to /cicsts (/var is always a symlink to /<LPAR>/var)
  - > `In -s /cicsts /var/cicsts`
- If not sharing zFS
  - Mount data set onto /var as /var/cicsts

# Setup for Critical Dir



- SITParms
  - USSCONFIG
    - The name and path of the directory for CICS config files
    - Default: /var/cicsts/dfhconfig
  - JVMPROFILEDIR
    - The directory name for the JVM profiles
    - Suggest: /var/cicsts/jvmprofiles
    - Sample profiles DFHJVMAX and DFHOSGI must be copied to here (and edited) if you want to use them



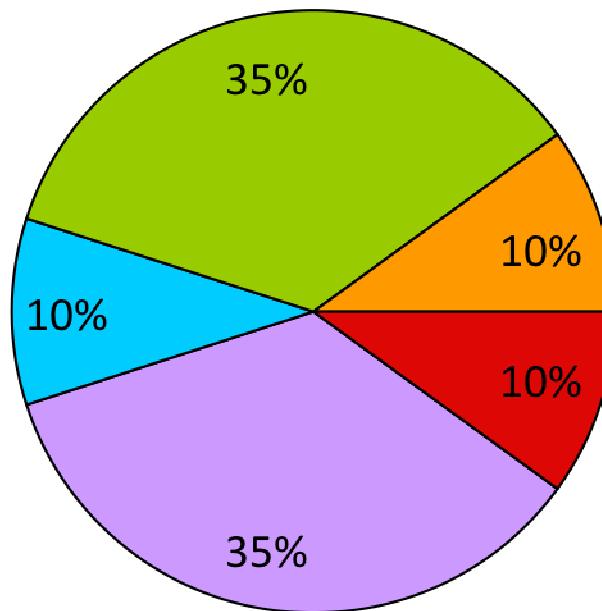
# Setup for Temp Dirs

- Give a z/OS UNIX User ID to each CICS region user ID
- Set up a Temp Dir on zFS for each of your CICS regions (eg /cicslogs/<APPLID>)
  - Each Temp Dir should be a separate filing system each backed by their own dataset
  - Give each region's User ID r/w permissions to their directory
- Choose a z/OS UNIX Group ID (GID) for the RACF group, and assign the it to the RACF Group
- Make sure that each CICS region user ID connects to the RACF group that you choose

Everything  
in its place

# CICS and zFS usage survey: types of resources

## Q2. What CICS application resources do you use today in HFS/zFS?



- Web services
- Static Web content
- Java
- Bundles
- N/A

CICS TS 5.1 introduces Application and Platform resources which are packaged as bundles on zFS...

# DOCTEMPLATES, URIMAPS and TCPIP SERVICES

- **DOCTEMPLATE** attribute
  - **HFSFILE**
    - zFS file containing the template (when it resides in zFS)
- **URIMAP** attribute
  - **HFSFILE**
    - zFS file that forms the body of a static response to an HTTP request from a Web Client
- **TCPIPSERVICE** attribute
  - **CIPHERS**
    - zFS file that contains the SSL cipher suite specification

# DOCTEMPLATES, URIMAPS and TCPIPSERVICES

- **DOCTEMPLATE** attribute

## **HFSFILE**

- zFS file containing the template (when it resides in zFS)

- **URIMAP** attribute

## **HFSFILE**

- zFS file that forms the body of a static response to an HTTP request from a Web Client

- **TCPIPSERVICE** attribute

## **CIPHERS**

- zFS file that contains the SSL cipher suite specification
- [Relative to USSCONFIG](#)

- Put all of these in the Critical Dir!

# WEBSERVICES

- **WEB SERVICE** attributes
  - **WSBIND**
    - zFS file name of web service binding file
  - **WSDLFILE**
    - zFS file name of the WSDL file
  - **ARCHIVEFILE**
    - zFS file name of the zip file that contains the WDSL

# WEBSERVICES

- **WEBSERVICE** attributes

## **WSBIND**

- zFS file name of web service binding file

## **WSDLFILE**

- zFS file name of the WSDL file

## **ARCHIVEFILE**

- zFS file name of the zip file that contains the WDSL

- Put all of these in the Critical Dir!

# PIPELINEs

- **PIPELINE** attributes
  - **CONFIGFILE**
    - ZFS file that defines processing nodes
  - **SHELF\***
    - Directory for CICS to store installed artifacts (may be shared by multiple CICS Regions)
  - **WSDIR**
    - Directory for WSDL and WSBIND files that are to be installed into CICS

# PIPELINEs

- **PIPELINE** attributes

## **CONFIGFILE**

- ZFS file that defines processing nodes

## **SHELF\***

- Directory for CICS to store installed artifacts (may be shared by multiple CICS Regions)

## **WSDIR**

- Directory for WSDL and WSBIND files that are to be installed into CICS

- Put all of these in the Critical Dir!

\*Recovery of shelf directory should be considered critical resource for WARM restart if using a CSD defined WEBSERVICE (as opposed to a pipeline scan)

# ATOMSERVICEs

- **ATOMSERVICE** attributes
  - **CONFIGFILE**
    - zFS file name that describes meta data for the feed
  - **BINDFILE**
    - ZFS file that describing XML transformation for the feed

# ATOMSERVICEs

- **ATOMSERVICE** attributes
  - **CONFIGFILE**
    - zFS file name that describes meta data for the feed
  - **BINDFILE**
    - ZFS file that describing XML transformation for the feed
- Put all these in the Critical Dir!

# BUNDLES

- **BUNDLE** attribute
  - **BUNDLEDIR** – zFS path to bundle directory

# BUNDLES

- **BUNDLE** attribute
-  **BUNDLEDIR** – zFS path to bundle directory
- Critical Dir again!

# Java Resources

- **PROGRAM** attribute
  - **JVMCLASS**
    - Name of service defined in OSGI bundle, in a CICS Bundle
    - Contained in BUNDLEDIR on the BUNDLE
- **JVMSERVER** attribute
  - **JVMPROFILE**
    - zFS file name of the JVMPROFILE
    - Relative to the SIT parm JVMPROFILEDIR
      - If the JVMSERVER is installed as a part of a bundle, JVMPROFILE is relative to BUNDLEDIR on the BUNDLE

# Java Resources

- **PROGRAM** attribute

## **JVMCLASS**

- Name of service defined in OSGI bundle, in a CICS Bundle
- Contained in BUNDLEDIR on the BUNDLE

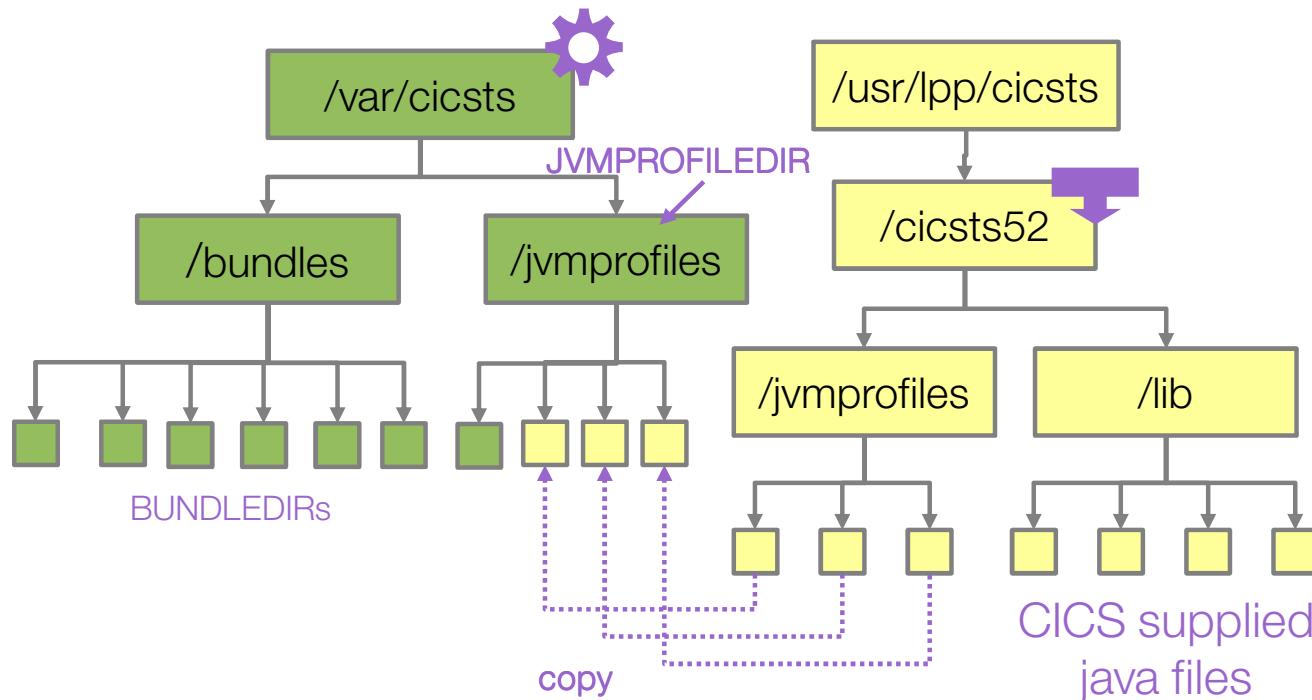
- **JVMSERVER** attribute

## **JVMPROFILE**

- zFS file name of the JVMPROFILE
- Relative to the SIT parm JVMPROFILEDIR
  - If the JVMSERVER is installed as a part of a bundle, JVMPROFILE is relative to BUNDLEDIR on the BUNDLE

- Critical Dir...

# Java Resources



You should copy sample JVMPROFILEs to the directories under `/var/cicsts` if you want to use them.

# JVMPROFILEs

- **JVMPROFILE** properties
  - **\$JAVA\_HOME** – zFS path to jre install
  - **\$WORK\_DIR** – zFS path to logs and dumps
- **JVMPROFILE (Liberty)** properties
  - **\$WLP\_INSTALL\_DIR** – install location wlp
  - **\$WLP\_USER\_DIR** – zFS path to server.xml
  - **\$WLP\_OUTPUT\_DIR** – Liberty logs and installedApps

# JVMPROFILEs

- **JVMPROFILE** properties
  - ⬇️ **\$JAVA\_HOME** – zFS path to jre install
  - ⬇️ **\$WORK\_DIR** – zFS path to logs and dumps
- **JVMPROFILE (Liberty)** properties
  - ⬇️ **\$WLP\_INSTALL\_DIR** – install location wlp
  - ⬇️ **\$WLP\_USER\_DIR** – zFS path to server.xml
  - ⬇️ **\$WLP\_OUTPUT\_DIR** – Liberty logs and installedApps

# JVMPROFILE suggestions

- **JVMPROFILE** properties
  - ➡ **\$JAVA\_HOME** – zFS path to jre install
  - ➡ **\$WORK\_DIR** – Use CICS Temp Dir: /cicslogs/&APPLID;
- **JVMPROFILE (Liberty)** properties
  - ➡ **\$WLP\_INSTALL\_DIR** – Use USSHOME: &USSHOME;/wlp
  - ➡ **\$WLP\_USER\_DIR** – Use JVMPROFILEDIR: /&CONFIGROOT;
  - ➡ **\$WLP\_OUTPUT\_DIR** – Use /cicslogs/&APPLID;/&JVM SERVER;

# JTA Logs in server.xml

- If you use JTA in Liberty, CICS needs r/w access to the JTA transaction logs.
- But they are critical for WARM restarts of CICS so should be recoverable
- Where do we put them?...

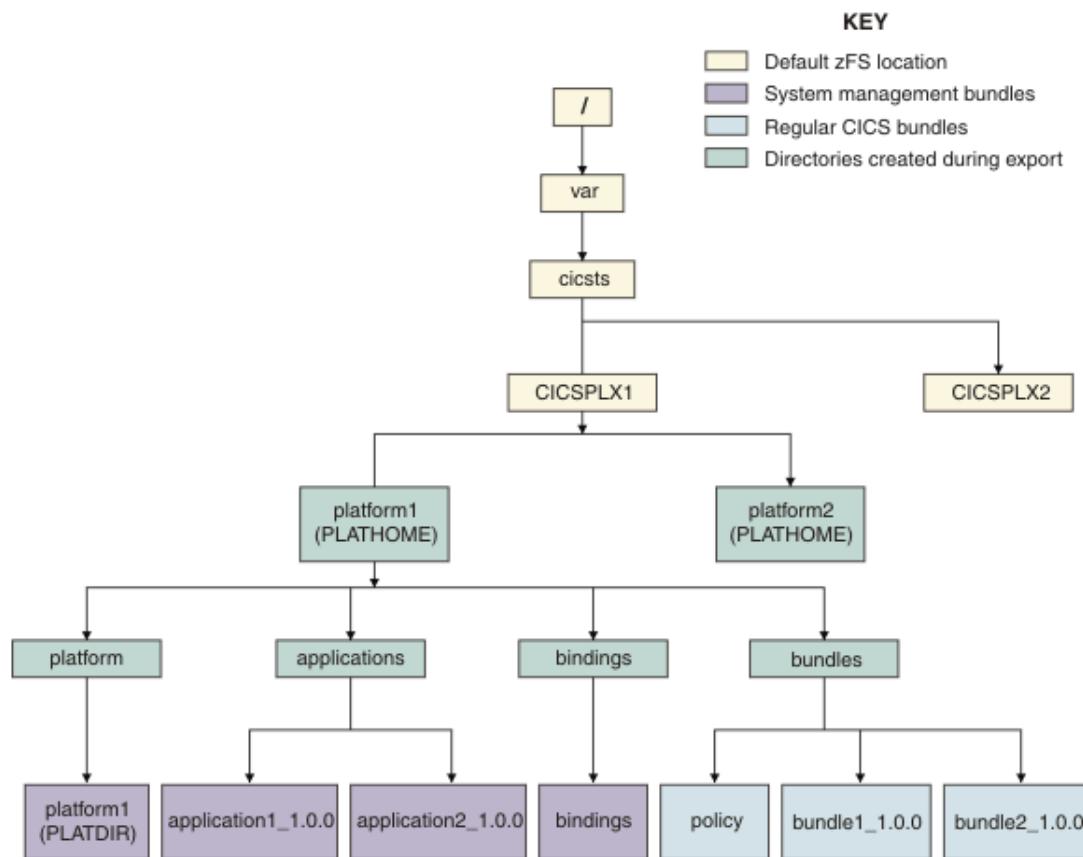
# JTA Logs in server.xml

- If you use JTA in Liberty, CICS needs r/w access to the JTA transaction logs.
- But they are critical for WARM restarts of CICS so should be recoverable
- Where do we put them?...
- Their own recoverable filing system under /cicsjta/...

```
<transaction transactionLogDirectory="/cicsjta/${com.ibm.cics.jvmserver.applid}/  
${com.ibm.cics.jvmserver.name}"/>
```

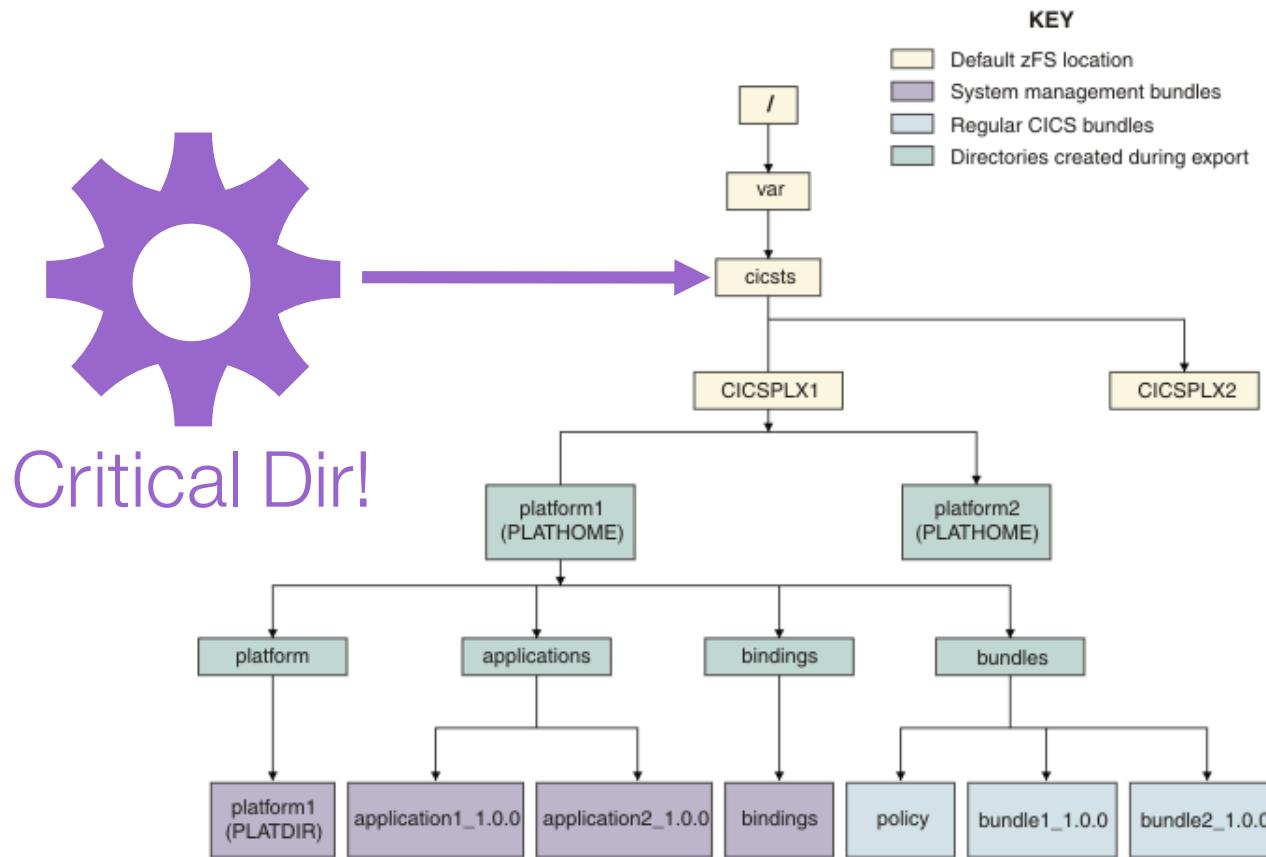
# CICS Platforms

- With the new CICS Platforms we've made it easy...



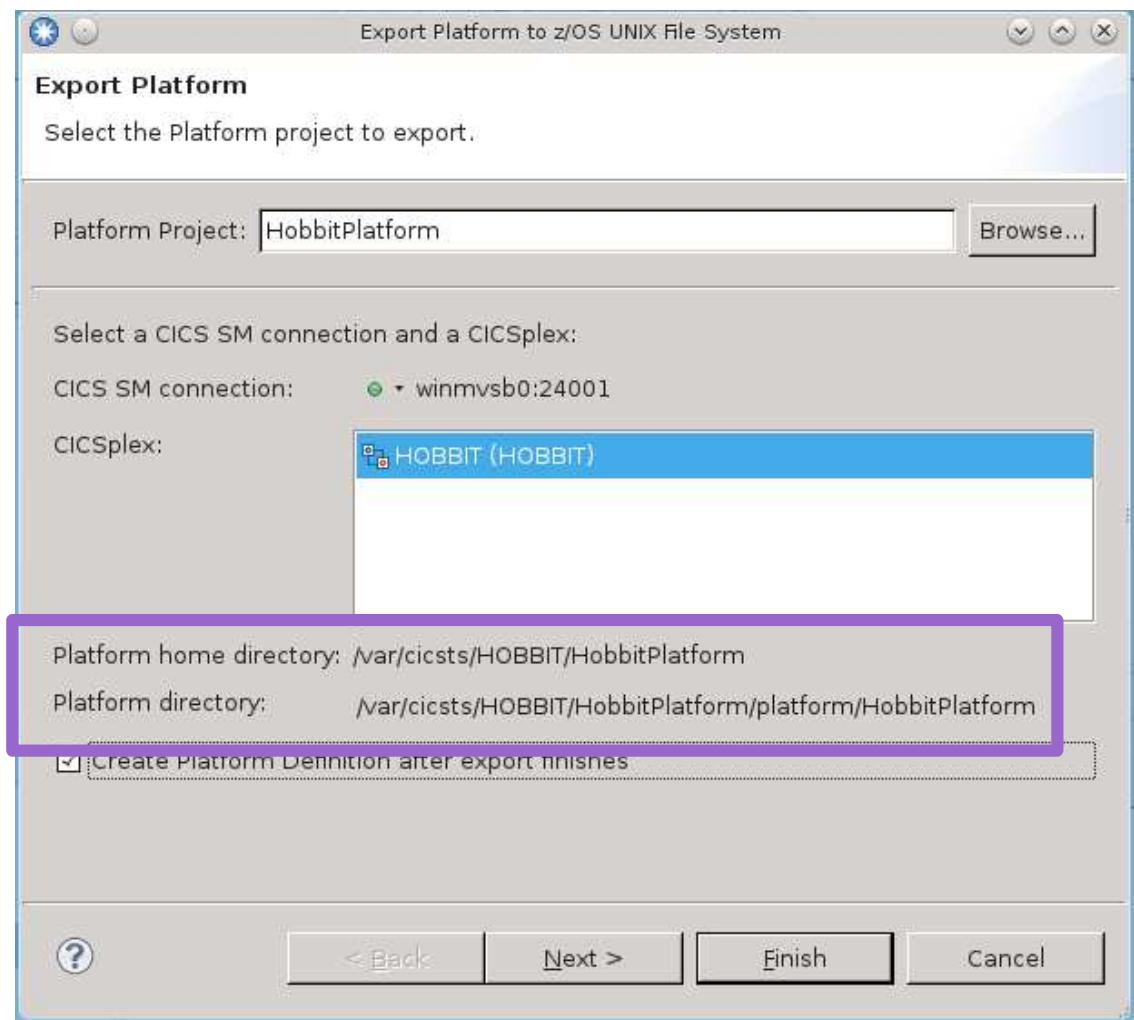
# CICS Platforms

- With the new CICS Platforms we've made it easy...



# CICS Platforms

- Explorer is aware of the platform home directory and takes care of the rest for you!



# Summary Table

Resources	Attribute
DOCTEMPLATE	HFSTFILE
URIMAP	HFSTFILE
TCPIPSERVICE	CIPHERS
WEBSERVICE	WSBIND WSDLFILE ARCHIVEFILE
PIPELINE	CONFIGFILE SHELF WSDIR
ATOMSERVICE	CONFIGFILE BINDFILE
PROGRAM (Java)	JVMCLASS
JVMSERVER	JVMPROFILE
JVMPROFILE	\$JAVA_HOME \$WORK_DIR \$WLP_INSTALL_DIR \$WLP_USER_DIR \$WLP_OUTPUT_DIR
server.xml	transactionLogDirectory
BUNDLE	BUNDLEDIR
CICS Platforms	Platform Home

Secure  
access

# Who needs access to what?

Directory	CICS Region	Admin
Install Dir	read	read
Critical Dir	read	read/write
Temp Dir	read/write	read/write

- Both user types would also have execute permissions on directories (not required for files)

# File permissions in zFS

Use the UNIX permission flags for Owner, Group and All to control access to your CICS resources on zFS

Here's an example of an entry you might see if you listed contents of a zFS directory

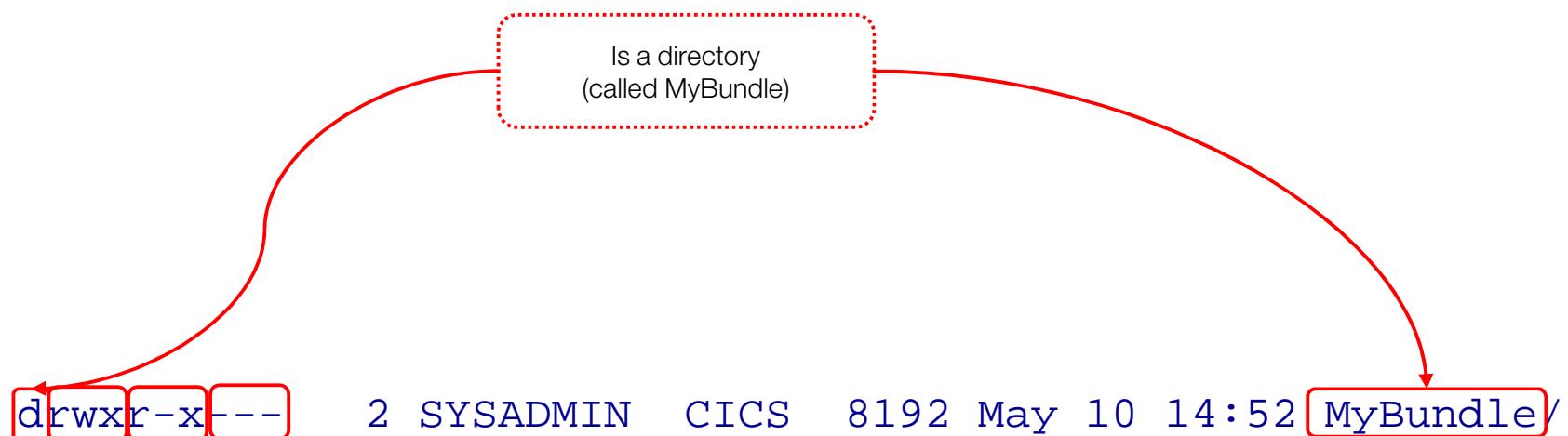
```
drwxr-x---    2 SYSADMIN  CICS  8192 May 10 14:52 MyBundle/
```

CICS zFS files may require access via 3 classes of user:

- CICS regions - region userid
- CICS system administrators (humans)
- Code management systems (tools)

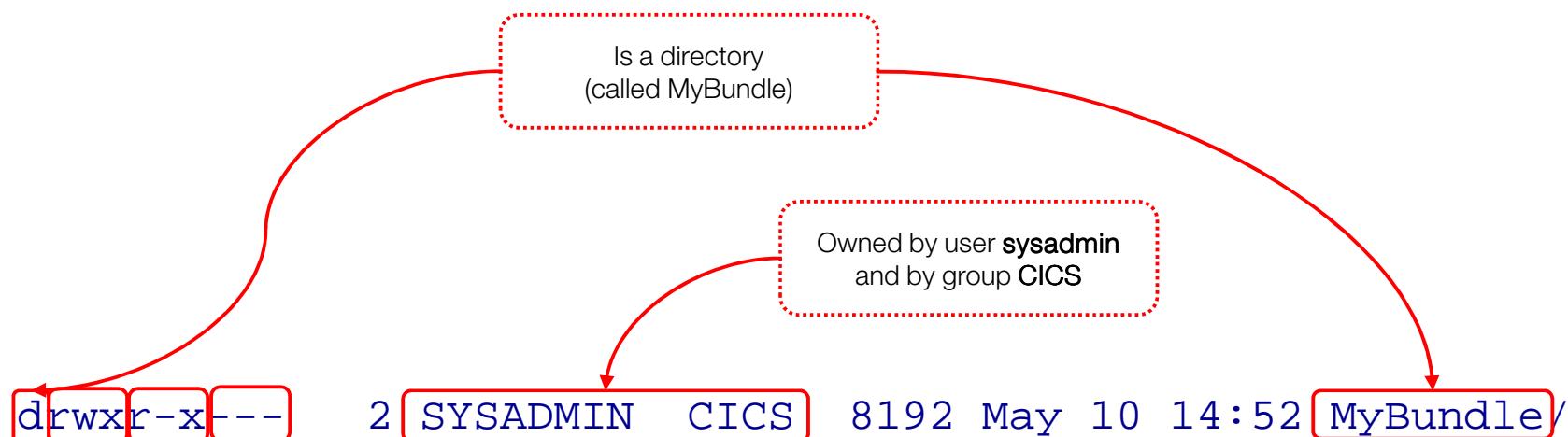
# File permissions in zFS

Use the UNIX permission flags for Owner, Group and All to control access to your CICS resources on zFS



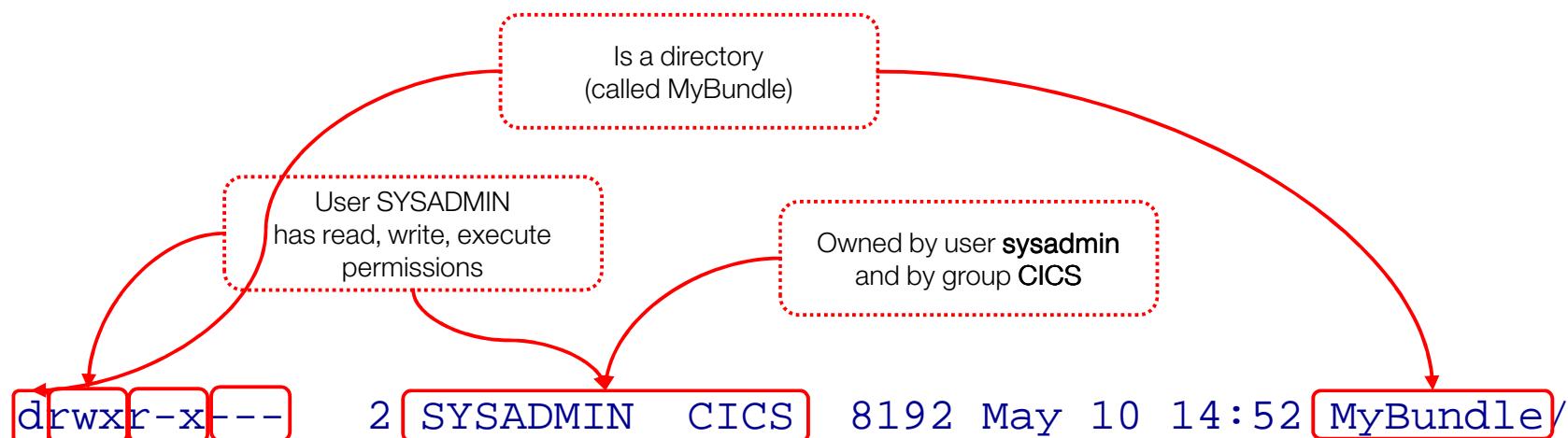
# File permissions in zFS

Use the UNIX permission flags for Owner, Group and All to control access to your CICS resources on zFS



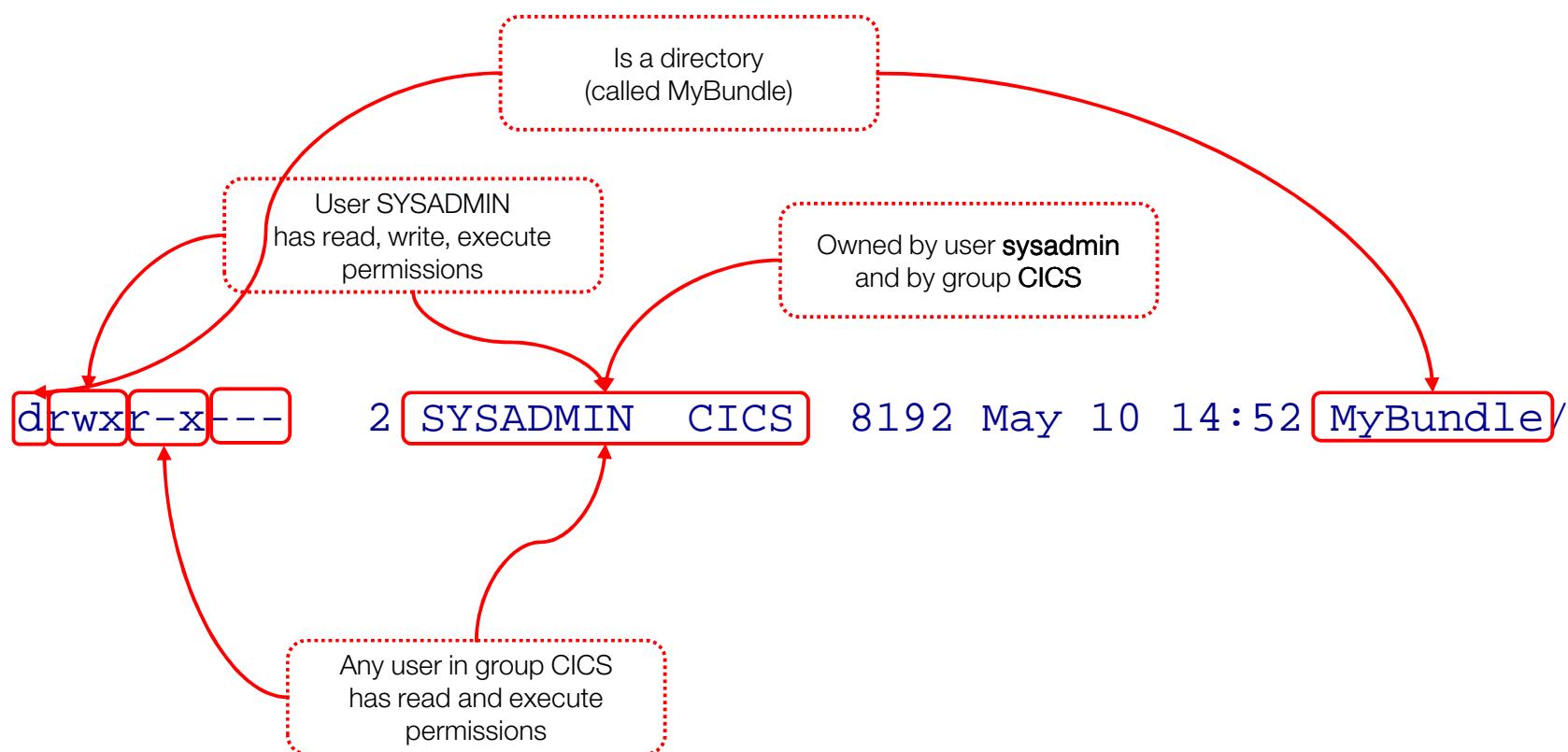
# File permissions in zFS

Use the UNIX permission flags for Owner, Group and All to control access to your CICS resources on zFS



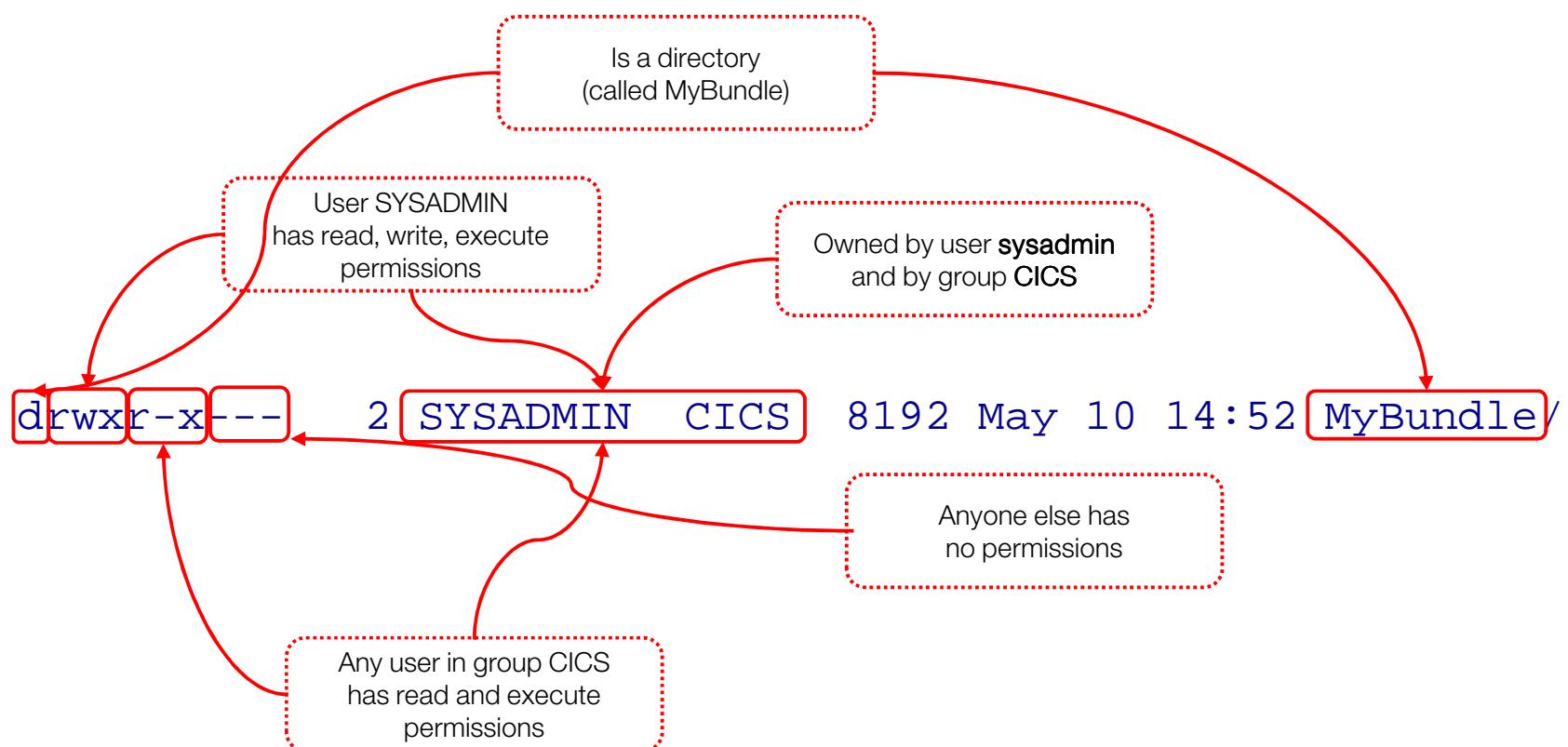
# File permissions in zFS

Use the UNIX permission flags for Owner, Group and All to control access to your CICS resources on zFS



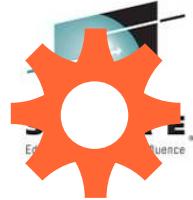
# File permissions in zFS

Use the UNIX permission flags for Owner, Group and All to control access to your CICS resources on zFS



# UMASK

- The file permission bits are set using the UMASK of the creating process, which signifies the bits that are not set
  - i.e a umask of 022 causes
    - Directories to be created with 755 (rwxr-xr-x) permissions
    - Files to be created with 644 (rw-r--r--) –
      - By default x permissions are not given for files



# Example Setup for Critical Dir

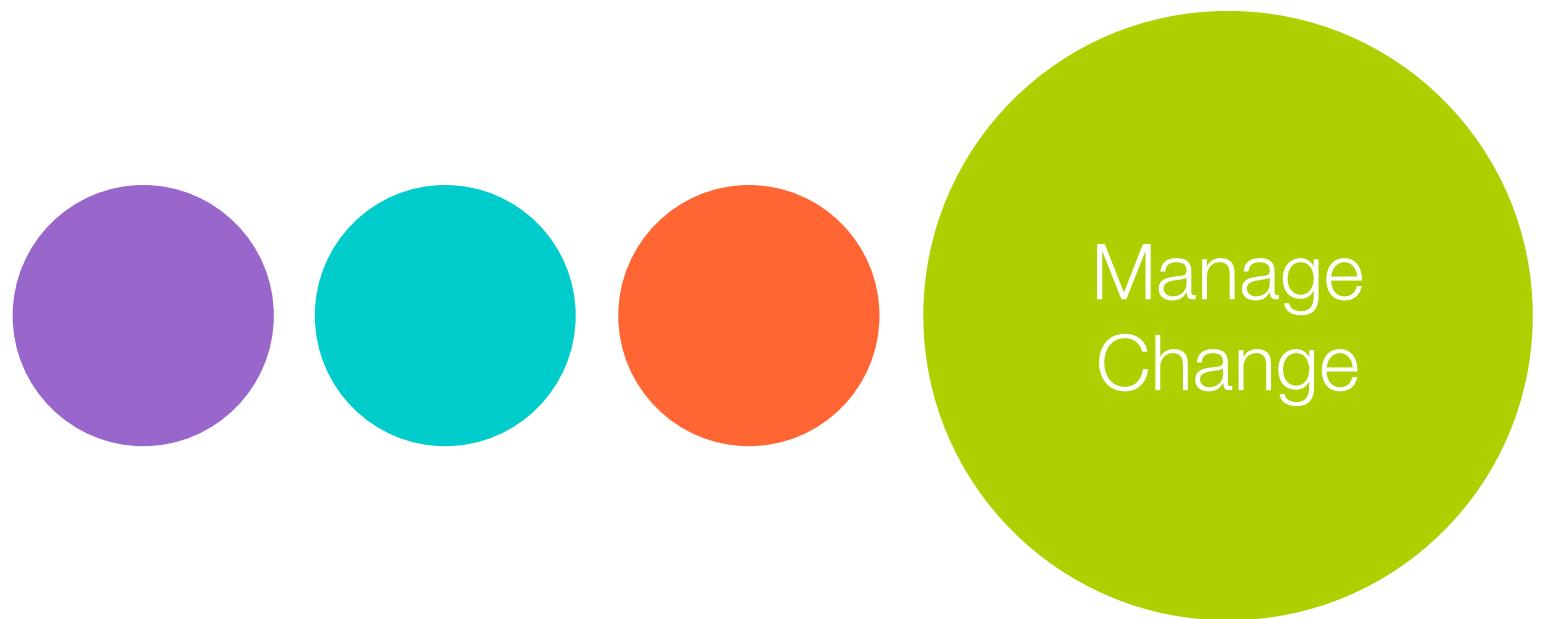
- Set permissions of /var/cicsts to allow access by multiple readers (CICS regions) and a common writer (administrator)
  - 1. Set the owner to have read/write/execute, this will be the userid required by zFS to export files into zFS
  - 2. Set the readers to have read/execute access
    - > chgrp -R <group> /cicsts
    - > chmod -R 750 /cicsts
  - 3. Set default file permission for the FTP daemon to give writers(owners) rw and readers(group) r
    - i.e UMASK 027
    - see SYS1.TCPPARMS(FTPDATA)
    - Used by CICS Explorer “Export Bundle Project to z/OS USS” process

# Multiple writer's problem

- A user can be in many groups, but a file has only one group permission
  - Meaning that if multiple users need to access the file they must be in that group, and will all share the same permissions
  - This means 2 logical groups of users (such as system admins and CICS regions) can not use UNIX permission bits to be granted access
- ACLs provide a solution to this as they allow a more flexible model
  - Multiple groups can have file permissions
  - ACL inheritance can be controlled
  - However, they may only restrict the access permissions that are defined by the UNIX permissions bits
  - RACF CLASS FSSEC
  - Control using setfacl USS command

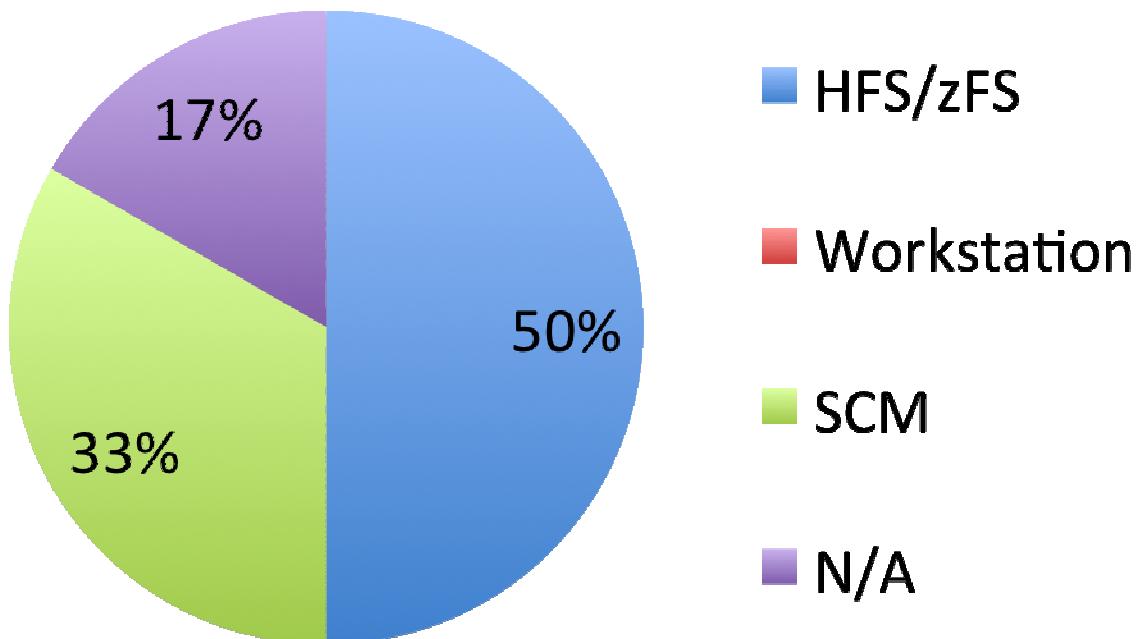
# Performance of zFS

- Performance of shared zFS mounted r/w filesystems has been regarded as an issue (in terms of XCF signalling costs) and function shipping of I/O between LPARs
  - Often a problem on JVM startup due to loading of JARs from zFS and writing of info to OSGi cache directory (under WORK\_DIR)
- Solutions:
  - V1R11 provides local read caching – removing overheads for reads
  - V1R13 provides direct I/O for read and write, removing need to function ship these commands to the owning LPAR
  - Or mounting file system locally removes need to function ship I/O
  - JVM class caching provides ability to cache Java byte codes in a shared memory area (i.e within LPAR)
    - > Requires APAR PM78799 on CICS TS V4.2 to support class caching



# zFS usage survey...

**Where do you store the master copy of CICS USS files?**

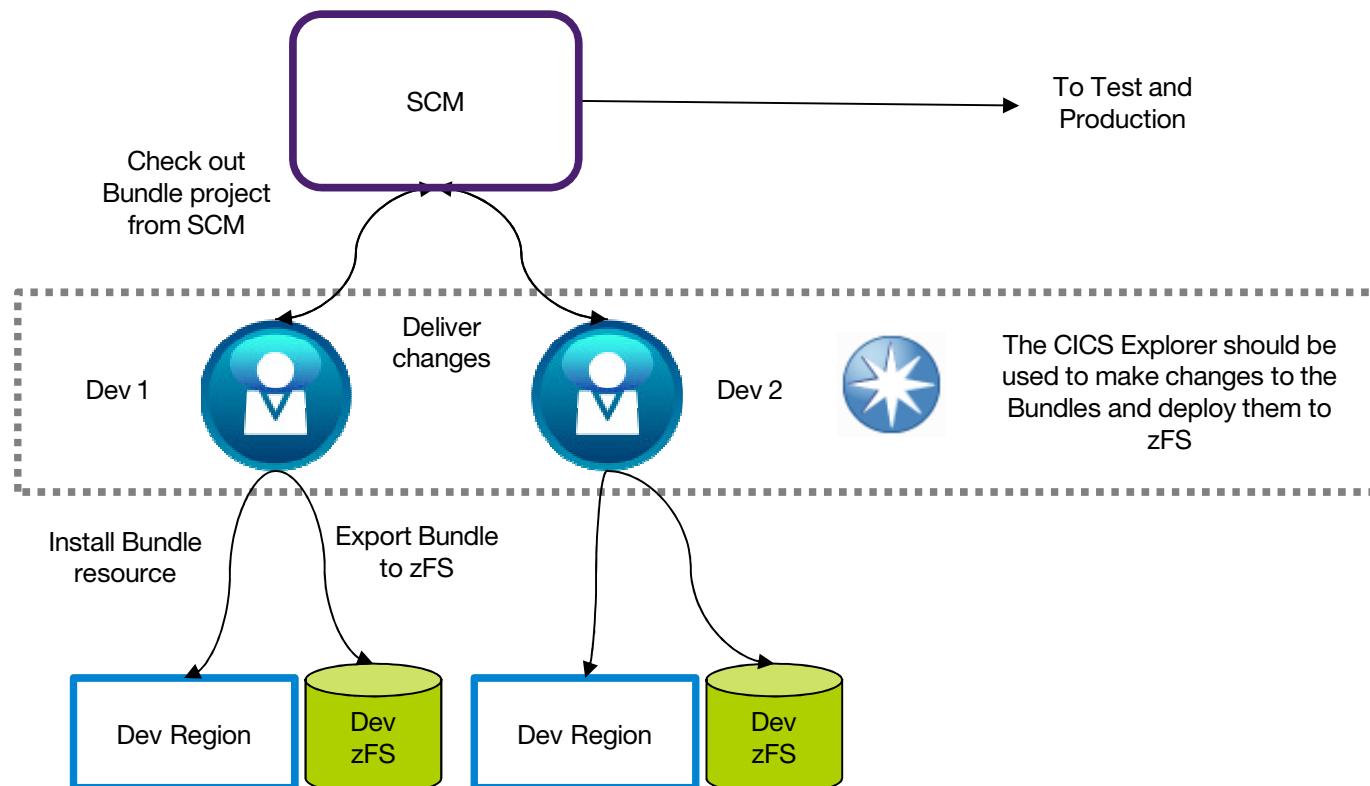


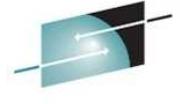
# Managing changes to CICS Bundles



CICS Bundle XML should be treated as **source code**

Changes should be managed and shared using a source code management (SCM) repository

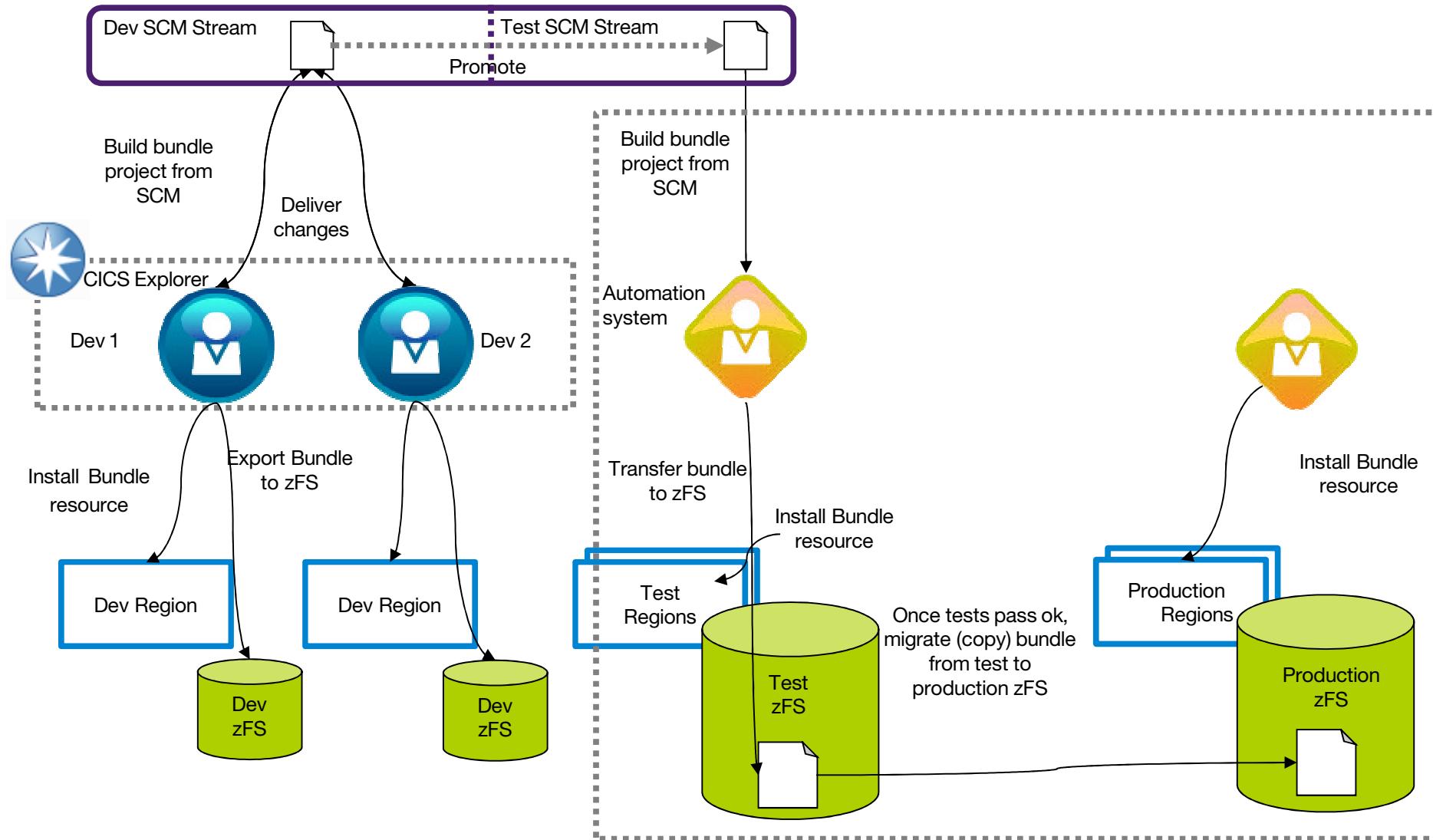




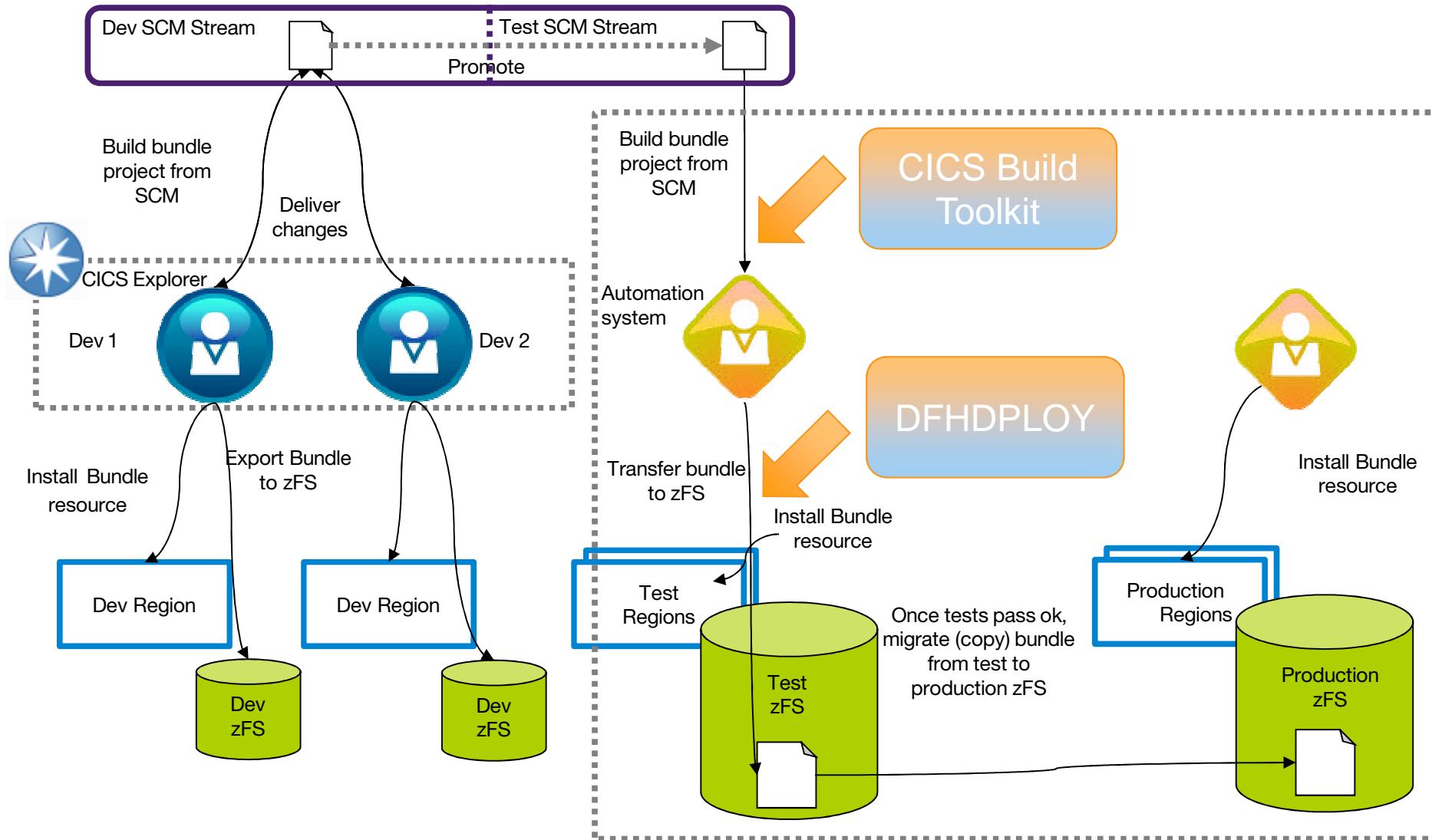
# Migrating CICS Bundles from Dev to Test to Production

- BUNDLES should be treated like any other CICS resource that has a reference to an artefact that lies outside the CSD eg:
  - PROGRAMS have load modules/java classes
  - WEBSERVICEs have wsbind files
- You should migrate the CICS Bundle XML **before** the BUNDLE resource
- You wouldn't migrate a new PROGRAM resource before you migrated the load module for it!

# Migrating CICS bundle XML from Dev to Production

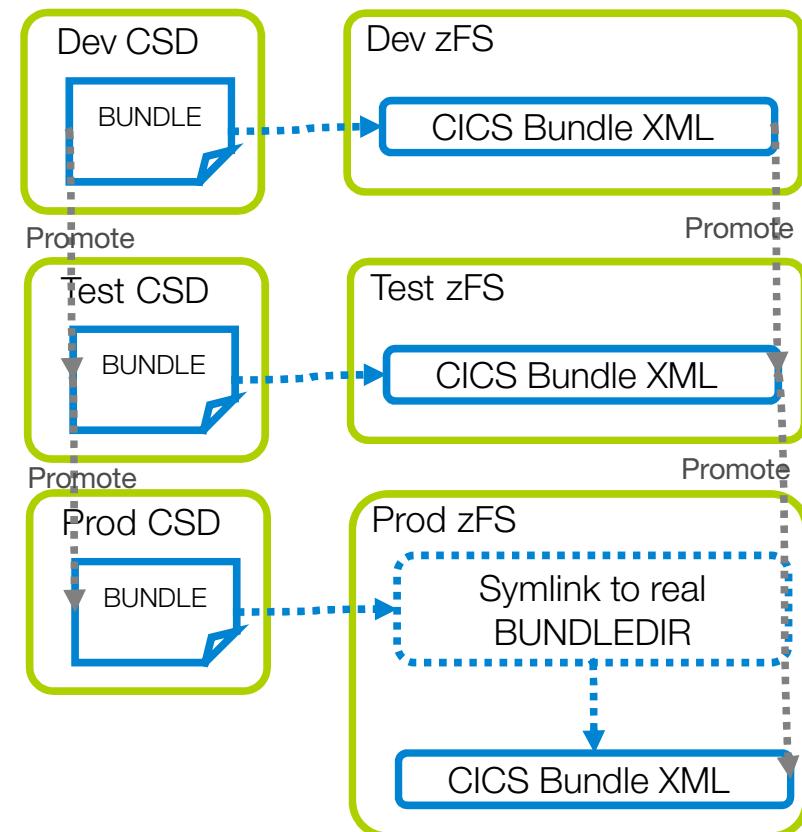


# Migrating CICS bundle XML from Dev to Production



# Migrating bundles

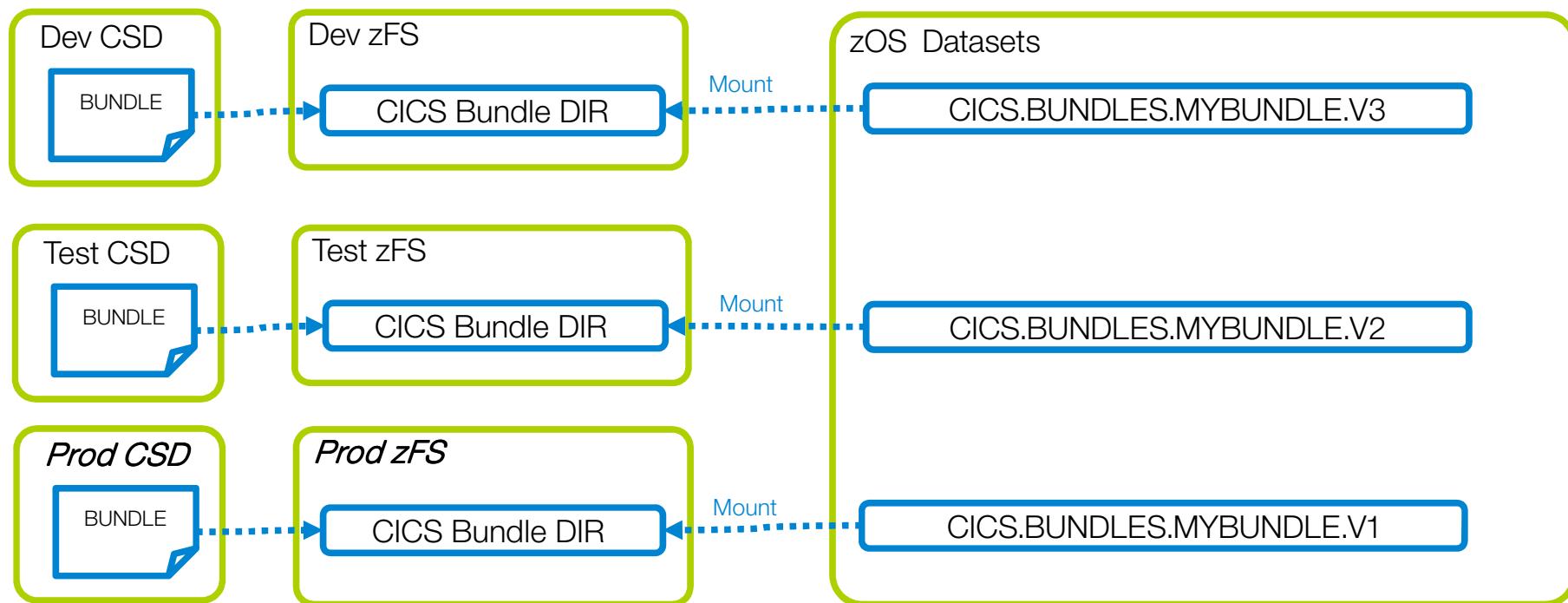
Option 1: Modify BUNDLEDIR to new zFS location



Option 2: Use Symlinks to point to the real bundle location

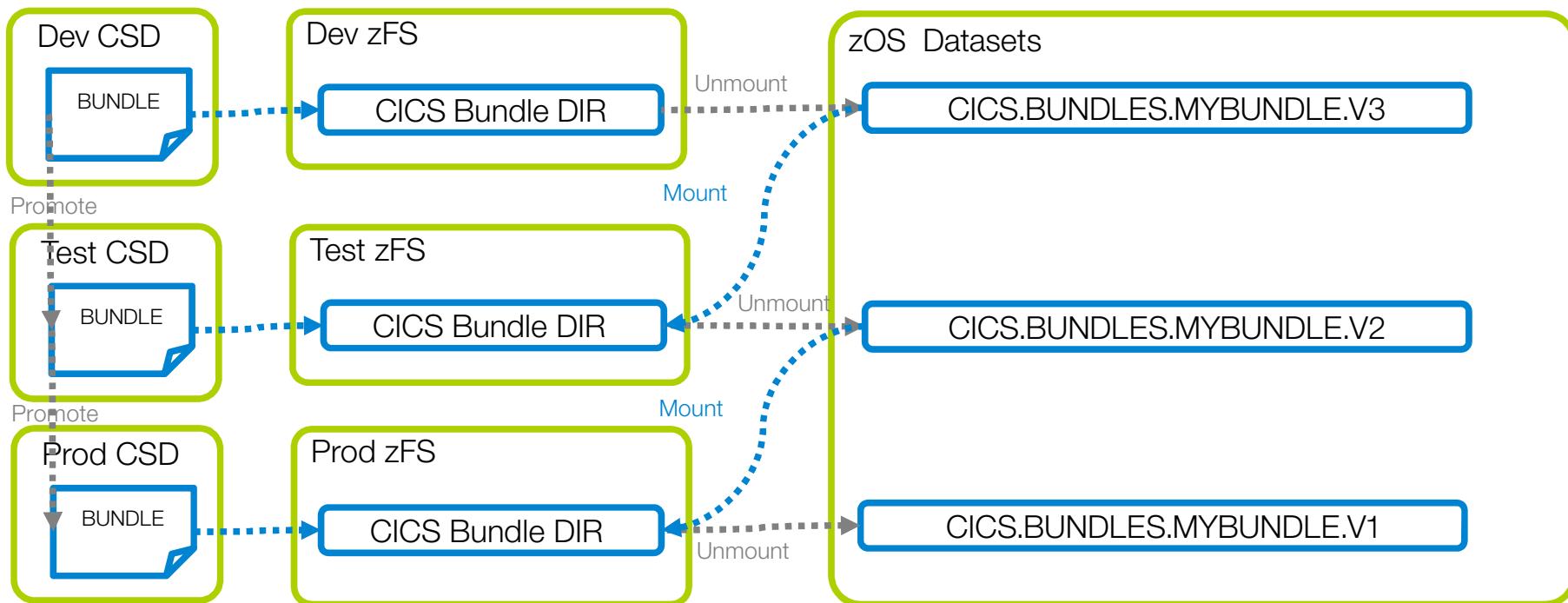
# Migrating bundles -2

## Option 3: Use zFS mounts to migrate bundles



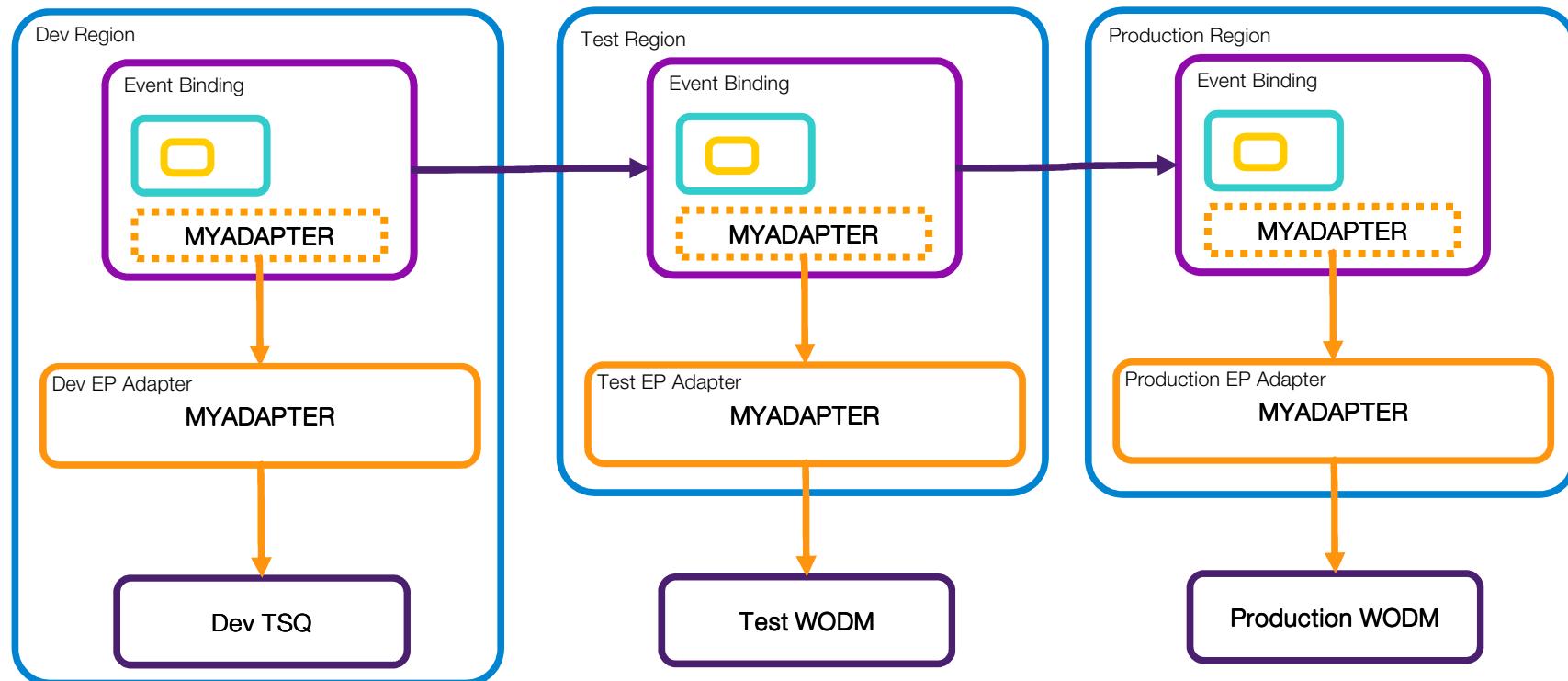
# Migrating bundles -3

## Option 3: Use zFS mounts to migrate bundles



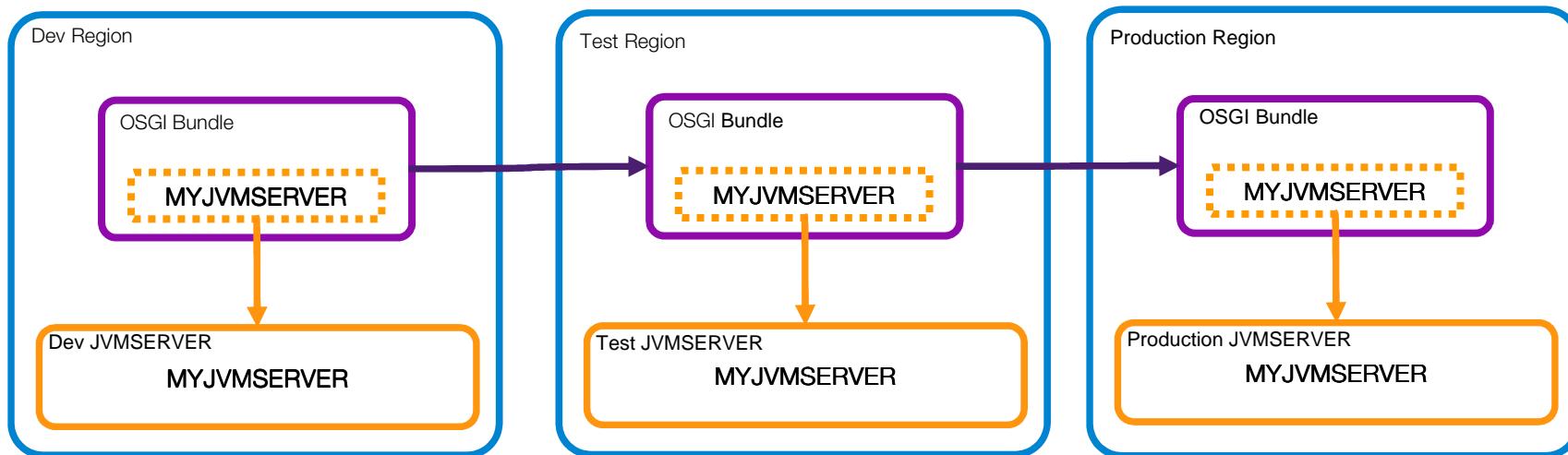
# Top Tip 2: Use separate EPADAPTERs on all EVENTBINDINGS

Use Separate EP Adapters in CICS TS V4.2 to ensure no changes are needed to CICS Bundles containing EVENTBINDINGS during migration



# Migrating bundles - JVMSERVER names

When using JVM server you need to have the same JVMSERVERs in all regions to ensure no changes are needed when migrating CICS bundles project containing OSGi bundles or Web project



Defined in the .osgibundle or .warbundle in CICS bundle project

```
<osgibundle symbolicname="com.ibm.cics.server.examples.hello"  
version="1.0.0" jvmserver="DFHJVMS"/>
```

# Summary



- A place for everything
  - Create separate file systems for install, critical and temp artifacts
  - Have places in those file system for each type of artifact
  - Consider requirements for sharing across LPARs
- Everything in it's place
  - Put each artifact into it's appropriate place
  - Use symlinks where necessary
  - Consider requirement for recovery and backup
- Secure access
  - Set permission bits on each file system to control access
  - Use ACLs to solve the multiple writer problem
  - Consider requirements for FTP access
- Manage change
  - Manage changes to zFS artifacts through a SCM
  - Consider references from zFS files, to minimise changes

Complete your session evaluations online at [www.SHARE.org/Seattle-Eval](http://www.SHARE.org/Seattle-Eval)

