Summer 2012 SHARE Anaheim Session 11806: Recent z/OS Enhancements You Can Use to Reduce Down Time

www.ibm.com/redbooks

Frank Kyne
Karan Singh

Redbooks®

IBM ITSO - International Technical Support Organization

Intro

Who are we?

- Frank Kyne is an ex-sysprog that now works in the IBM Redbooks organization, with books and classes about Parallel Sysplex and HA.
- Karan Singh is an ex-sysprog (and ex-teacher and ex-you-name-it!) that is now a project leader in the IBM Redbooks organization, responsible for books and classes about core z/OS and security.

Why this topic?

- Now that we have a bit of breathing room between z/OS 1.13 and z/OS Version 2, we want to encourage customers to use the opportunity to exploit z/OS functions that can improve availability AND take very little time to implement.
- Thanks to Cheryl Watson for promoting the idea of this session!
- Thanks to a host of others for their help and patience.

Session objectives

The objective of this session is to provide a live demo to show that the implementation of many of these enhancements is something that you could tackle over your lunch break (note that no outage is required to implement any of this stuff)....

- It is not meant to teach you the details of the functions we will use - objective is just to illustrate the benefits they provide and how easy they are to implement.

Session objectives

In this session we will show you (time permitting) how to:

- Set up z/OS BCPii
  - Note that this is NOT the same as the BCPii function provided with Tivoli System Automation
  - BCPii is a pre-req for SSDPP
- Implement System Status Detection Partitioning Protocol (SSDPP)
  - Including a demo of the difference in how long it takes to partition a failed system time without and with SSDPP
- Implement AutoIPL for:
  - Taking an automatic standalone dump after a wait state
  - Automatically re-IPL z/OS after the SAD completes
- Exploit JES2 Dynamic Proclib
- Will NOT cover Auto Reply, MVS Message Flooding, z/OS HealthChecker, and SMF record flooding control this time - had originally planned to have 2 sessions to cover all this

ibm.com/redbooks

Base Control Program internal interface

Redbooks

IBM ITSO - International Technical Support Organization

Why BCPii and SSDPP

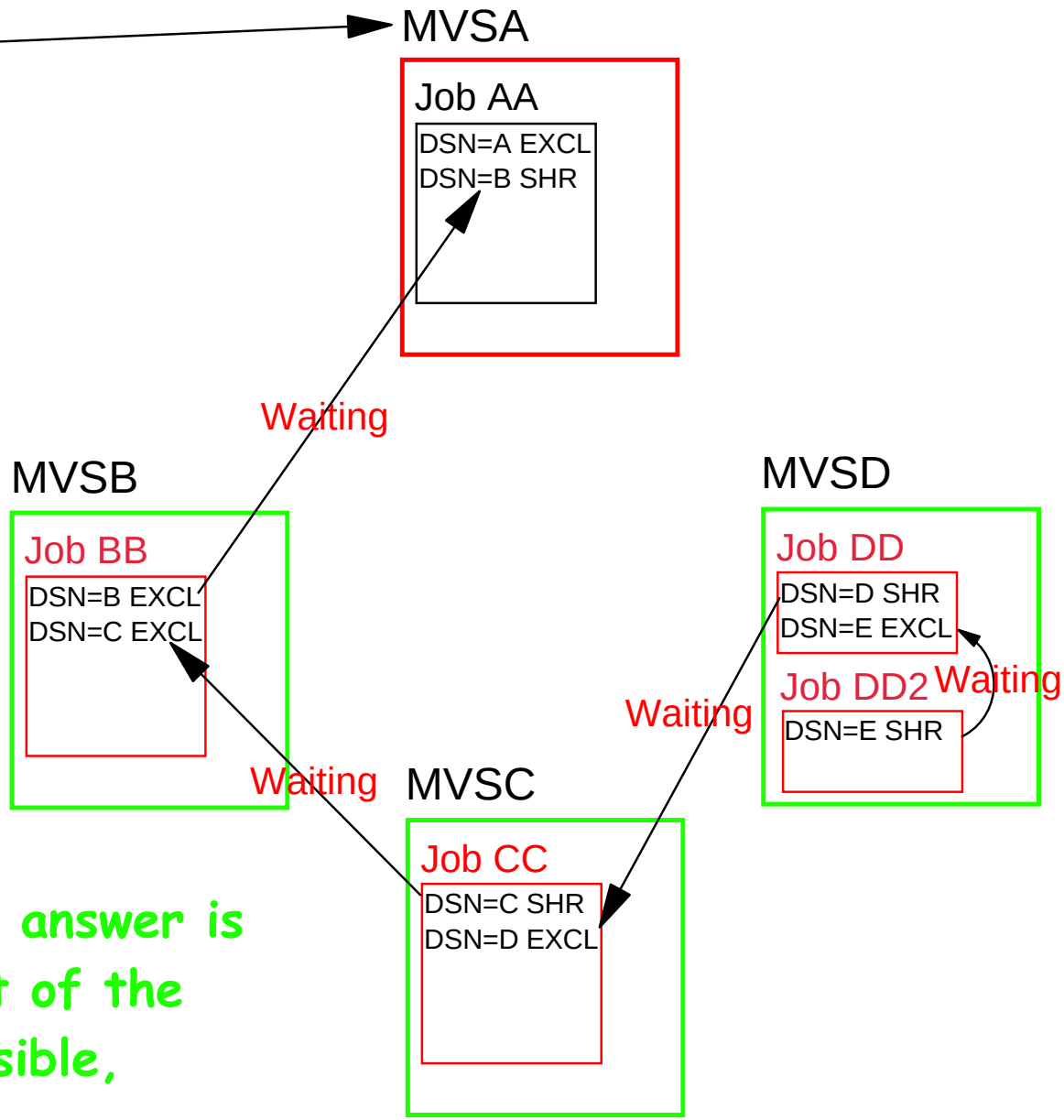
If a member of a sysplex dies, it is probably holding resources that will be required by other members of the sysplex.

- And the longer this situation lasts, the more units of work will be impacted.

If a system stops:

- It is probably holding resources that will be needed by another member of the sysplex.
- It will not release those resources until it recovers or is removed from plex

The longer a stalled system remains in the sysplex (holding resources), the larger is the impact on other systems.



So, OBVIOUSLY, the answer is to partition MVSA out of the plex as quickly as possible, right?

Why BCPii and SSDPP

Prior to z/OS 1.11, the only mechanism that z/OS had to determine the status of another member of the sysplex was to check that system's heartbeat in the sysplex CDS.

- If a system is going through recovery, it might not be able to update its heartbeat in the CDS. This means that you need to give a system some "reasonable" amount of time to recover before the system partitions the sick system out of the sysplex.
  - An IPL might take 30 minutes. Would you rather give a little more time for recovery to work, or kill it now and face an IPL? Your answer is probably "it depends on whether the system is dead or is in the middle of recovery".
  - Prior to z/OS 1.11, z/OS had no way to know whether another system was dead or trying to recover.
- SSDPP (and BCPii) changed that.

System partitioning actions

First, let's see how long it takes to partition a failed system out of the sysplex WITHOUT SSDPP....

For our demo, we will use our little 2-way sysplex. The systems are called #@$2 (LPAR A21) and #@$3 (LPAR A22) and both run z/OS 1.13 on a z196.

System partitioning actions

First, let's see how the system is currently set up:

Failure detection interval (FDI) (increased from 85 to 165) seconds by z/OS 1.11

SFM action when FDI is exceeded

Does sysplex CDS support SSDPP?

These fields would be populated if BCPii was working

```
D XCF,C
IXC357I  13.30.33  DISPLAY XCF 214
SYSTEM #@$2 DATA
   INTERVAL   OPNOTIFY     MAXMSG    CLEANUP       RETRY   CLASSLEN
     165        168        2000         15          10        956

   SSUM ACTION  SSUM INTERVAL  SSUM LIMIT     WEIGHT  MEMSTALLTIME
      ISOLATE              0         900         90           300

   CFSTRHANGTIME
            900

...

   SYSTEM STATUS DETECTION PARTITIONING PROTOCOL ELIGIBILITY:
     SYSTEM CANNOT TARGET OTHER SYSTEMS.
      REASON: SYSPLEX COUPLE DATA SET NOT FORMATTED FOR THE PROTOCOL
    SYSTEM IS NOT ELIGIBLE TO BE TARGETED BY OTHER SYSTEMS.
      REASON: SYSPLEX COUPLE DATA SET NOT FORMATTED FOR THE PROTOCOL

   SYSTEM NODE DESCRIPTOR: 002817.IBM.02.0000000B3BD5
                           PARTITION: 21    CPCID: 00

   SYSTEM IDENTIFIER: 3BD52817 21000008

   NETWORK ADDRESS: N/A

   PARTITION IMAGE NAME: N/A

   IPL TOKEN: N/A
```

System partitioning actions

Now let's wait-state the system and see how long we have to wait until we see the IXC101 Partitioning in Progress message... (should be a little under 3 minutes...)

Now we will set up BCPii and SSDPP and then repeat this exercise

What is BCPii?

Address space (HWIBCPII) that provides authorized programs running on z/OS with the ability to query, change, and perform HMC-like functions against the System z processors on the HMC network.

Program communication from z/OS directly to HMC - no need for TCP access from z/OS to HMC, so may help address security concerns about exposing HMC network beyond the machine room.

Delivered with z/OS 1.11, and rolled back to z/OS 1.10 with APAR OA25426.

BCPii

Starting with z/OS 1.11, system automatically tries to start BCPII address space at IPL time.

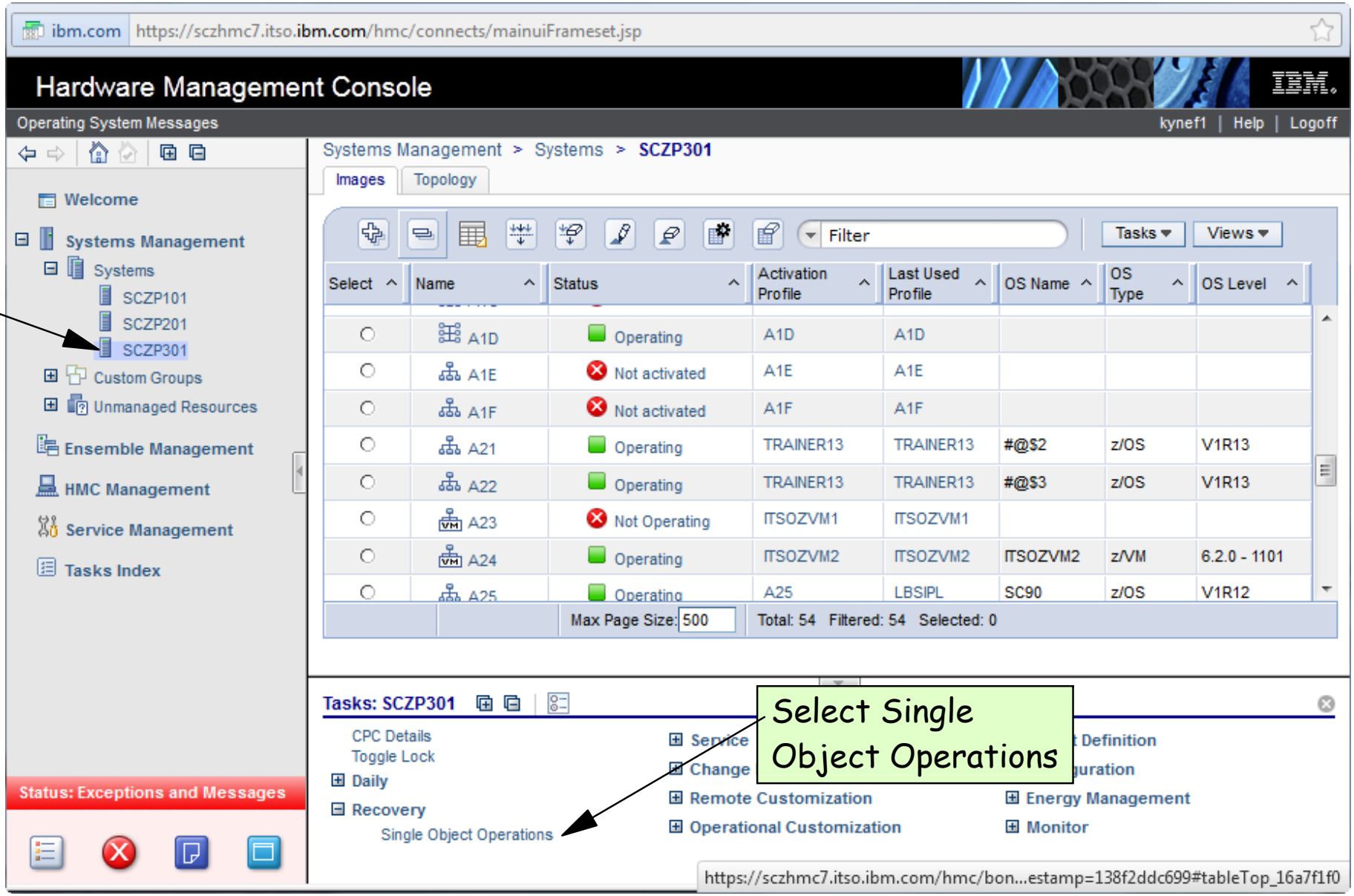
- You don't need to add anything to COMMNDxx, or automation.

Successful start requires that certain setup has been carried out:

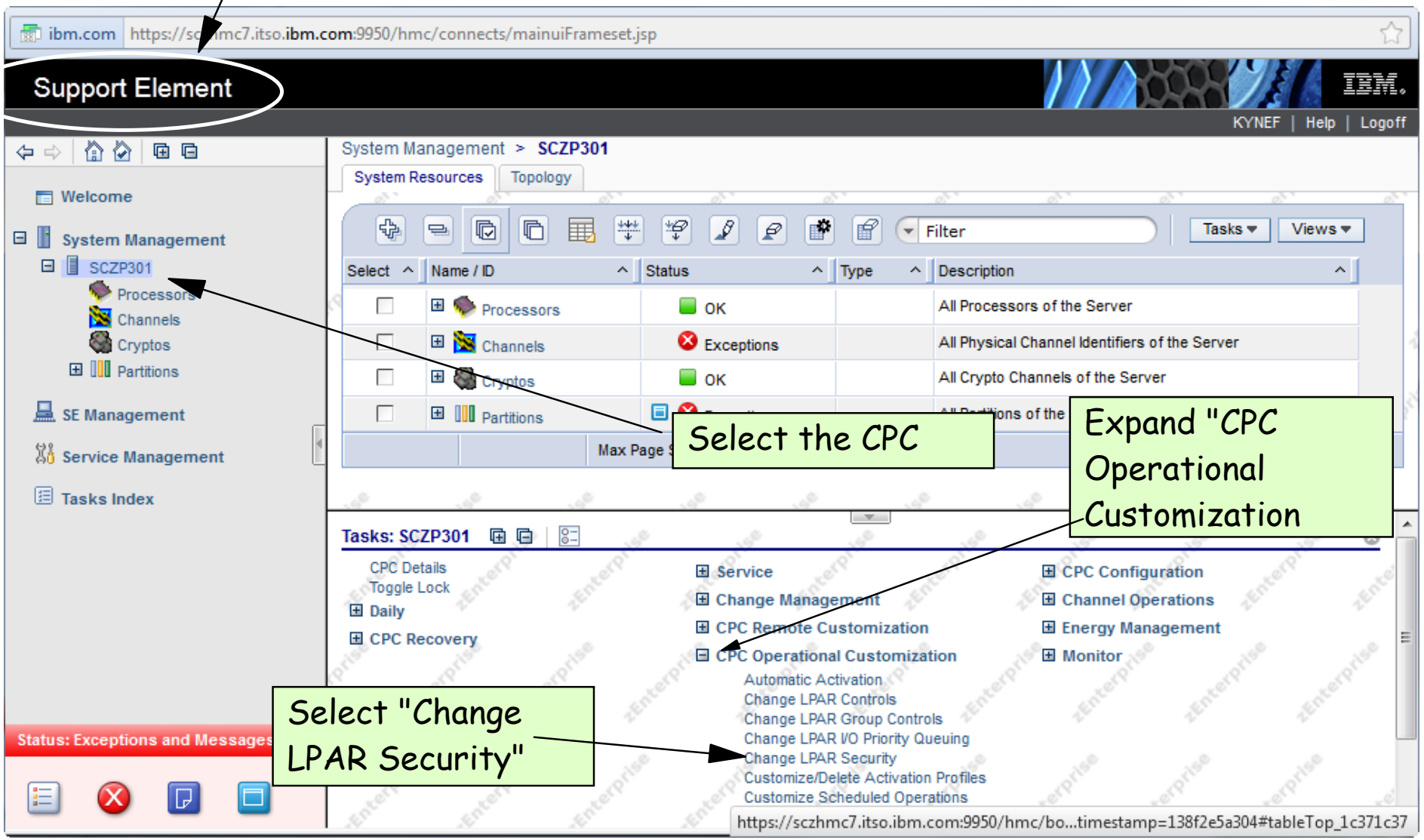
- Setup on the HMC:
  - Enable Cross Partition Authority for every LPAR that you want to be able to issue or be the target of BCPii commands.
  - Enable SNMP and define the Community Name.
    - Both of these can be changed non-disrputively if you wish
- Setup in z/OS
- SAF Security authorizations (in z/OS)

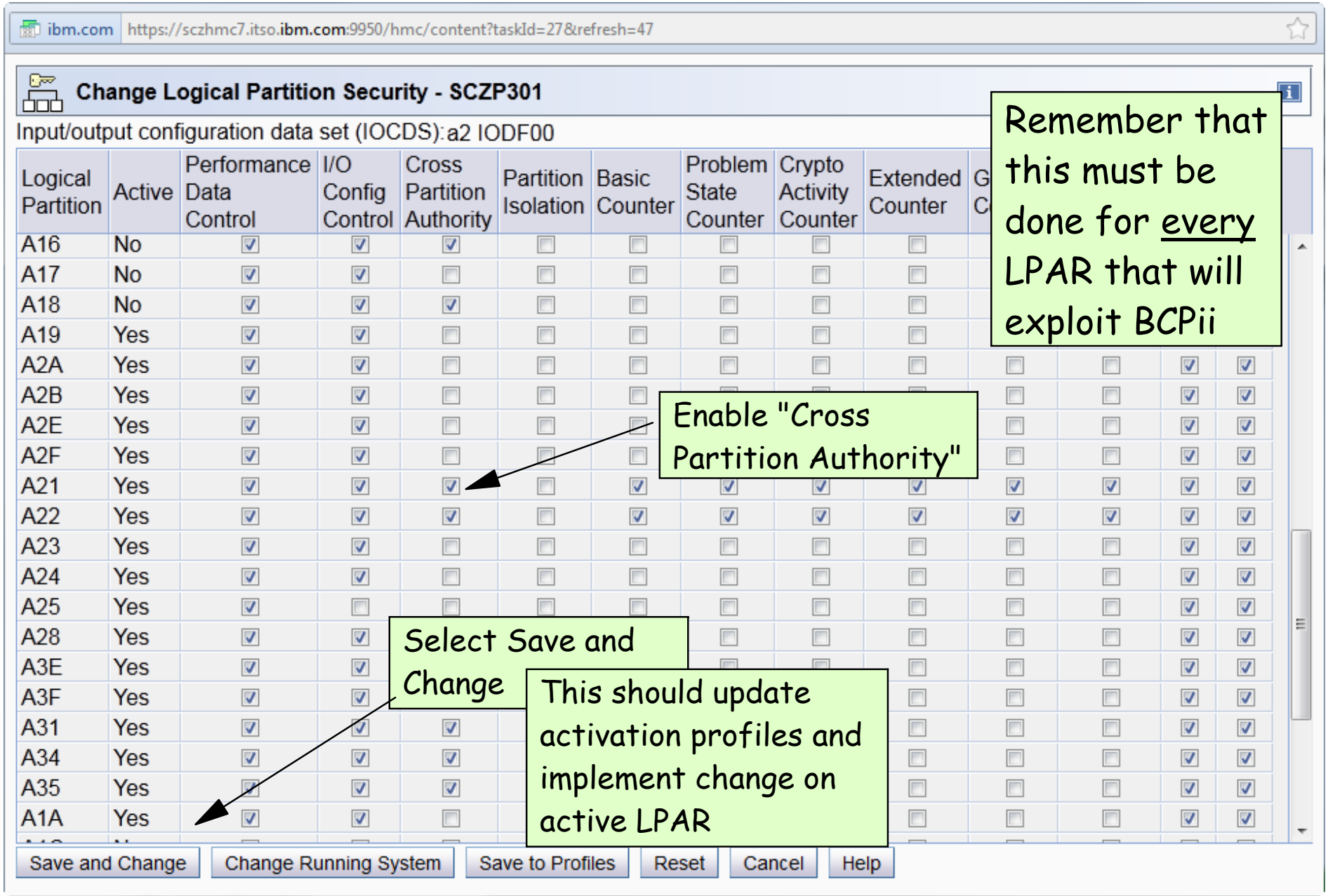
First step is to give LPARs authority to issue commands to other LPARs…

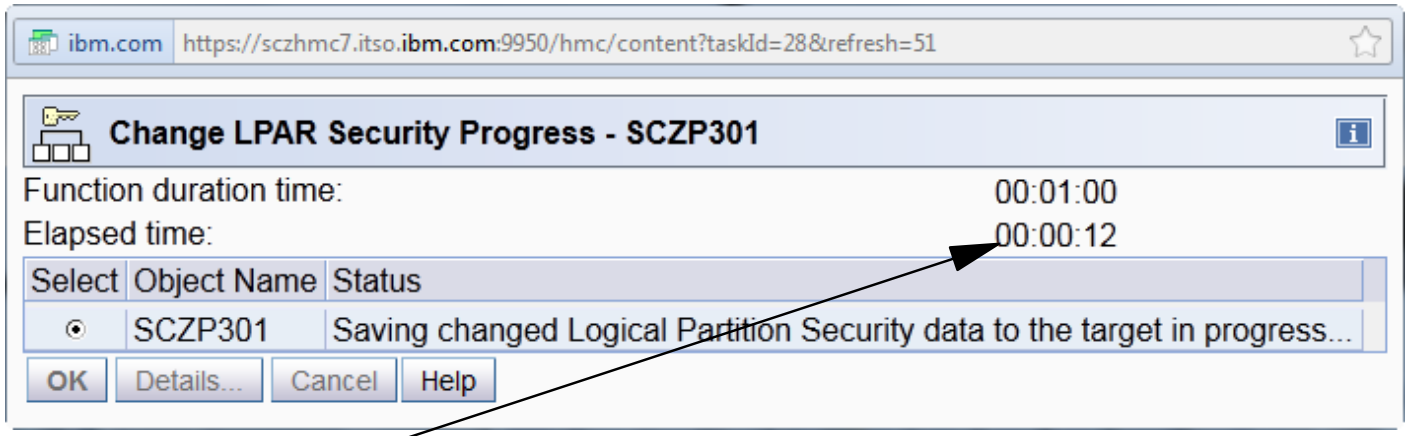
Select CPC you want to set up BCPii on



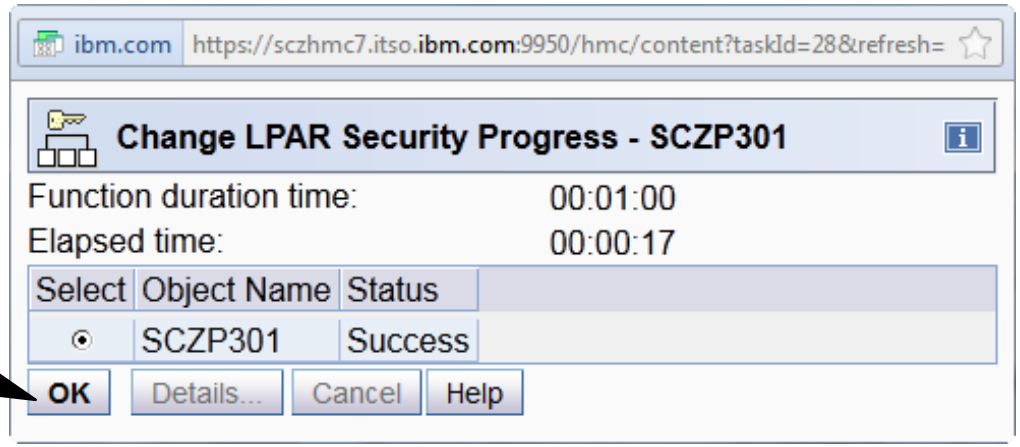
You are logged on to the SE







This may take a little while



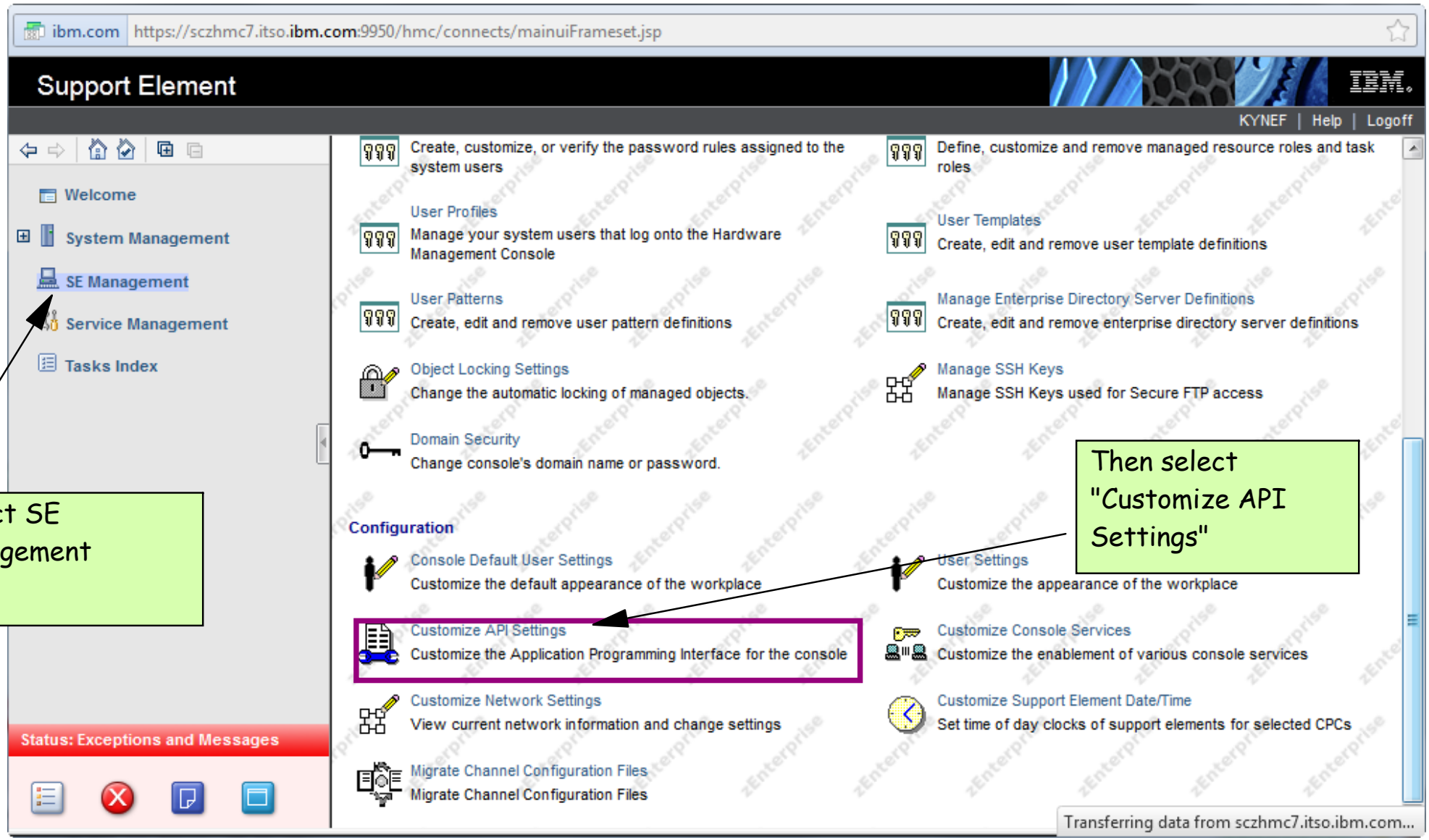
Press OK when finished

Recommend verifying that Activation Profiles were actually updated

Setting up BCPii - HW end

Next step is to add the SNMP definitions:

- These must be added in Single Object Operations for every CPC to be managed
- SE userid must have ACSADMIN authority to be able to do this....

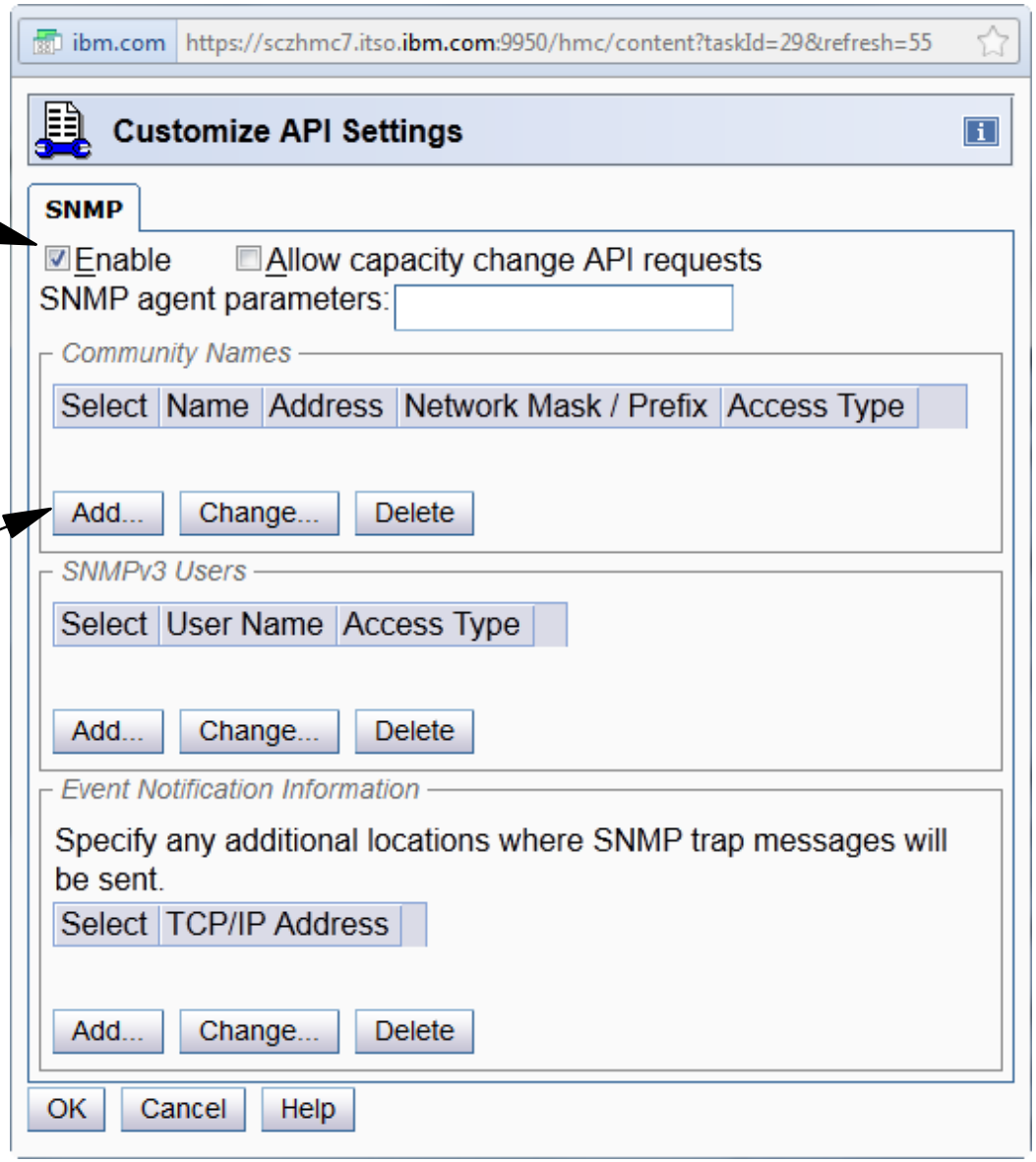


Select SE Management

Then select "Customize API Settings"

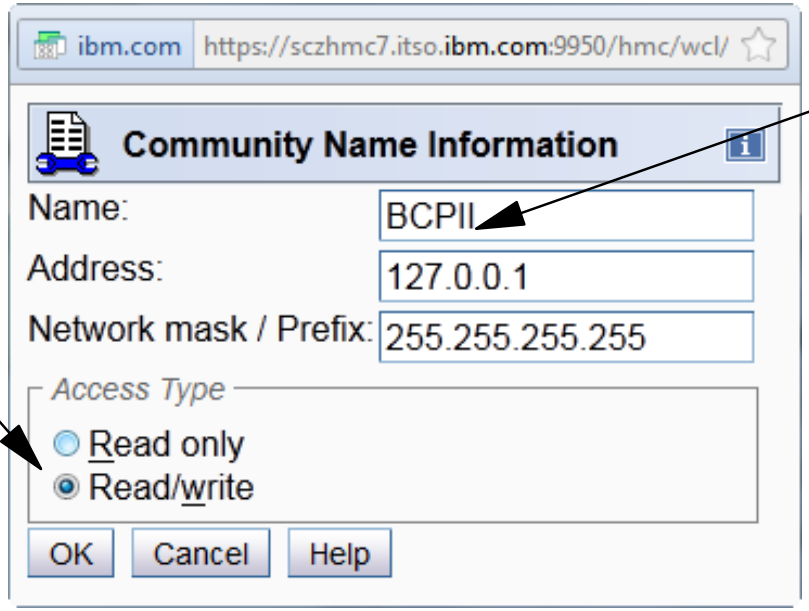
Select "Enable SNMP APIs"

Then click on Add in Community Names section



Fill in exactly as shown here. Remember to select Read/Write

Then press OK

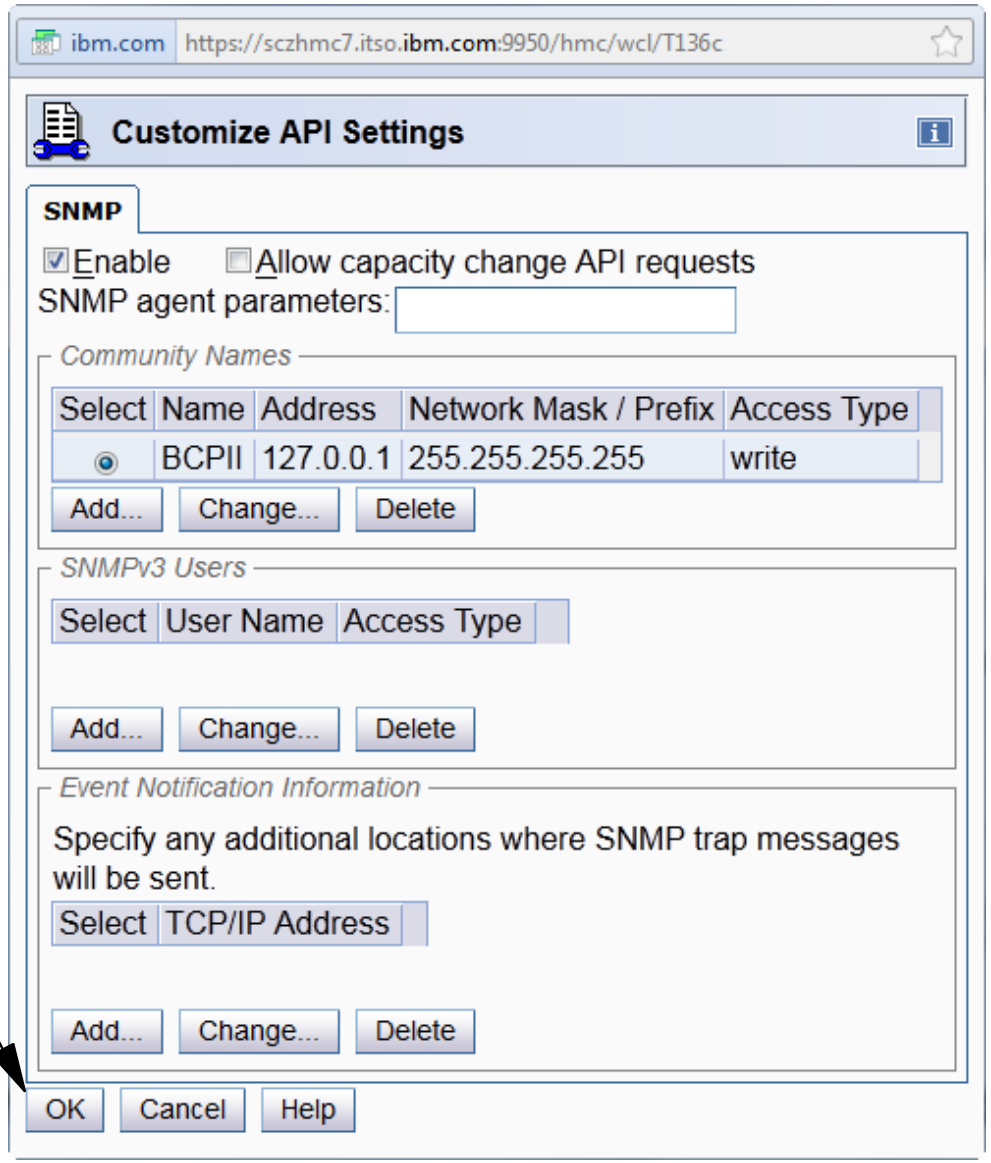


Name must be 1-16 chars, alphanumeric, no lower case.

Value you specify here must match name used in SAF CPC profile for this CPC

The Name value can be the same on every CPC, or different on every CPC. It is NOT necessary for each CPC to have a different Name value if you don't wish to.

Finally, click OK to apply and save the changes



The hardware setup for BCPii is now complete.....

BCPii - Security definitions

hlq.SCEERUN and hlq.SCEERUN2 must be in LNKLST.

Program authority:

- Program that will be calling BCPii services must reside in an APF-authorized library.

Issuing BCPii commands:

- The profile HWI.APPLNAME.HWISERV in the FACILITY resource class controls which applications can use BCPii services.
  - Anyone wishing to use BCPii must at least have READ access to this profile.
  - For XCF, simply have to ensure that the XCFAS started task is defined in RACF with the TRUSTED attribute - this is nearly always the case, but check to be sure.
- The FACILITY class must be RACLISTed.

BCPii - Security definitions

A BCPii application needs to have authority to the particular resource (CPC, Image, Capacity Record, Activation Profile) that it is trying to access (This is IN ADDITION to having access to the HWISERV FACILITY profile).

Profile names are:

- CPC: HWI.TARGET.netid.nau
- Image: HWI.TARGET.netid.nau.imagename
- Capacity Record: HWI.CAPREC.netid.nau.caprec
- Activation Profile: HWI.TARGET.netid.nau
- netid.nau is the 3-17 character SNA name for CPC (defined when you first define the SE to the HMC)

Level of access that is required depends on what you are trying to do - See Callable Services manual for details

BCPii - Security definitions

- When defining the CPC profiles, APPLDATA must match the community name you specified on the SE:
  - RDEFINE FACILITY HWI.TARGET.USIBMSC.SCZP301 UACC(NONE) APPLDATA('BCPII')

```
 BROWSE  - RACF COMMAND OUTPUT----------------------- LINE 00000000 COL 001 080
******************************* Top of Data ********************************
CLASS       NAME
-----       ----
FACILITY    HWI.TARGET.USIBMSC.SCZP301

LEVEL   OWNER      UNIVERSAL ACCESS  YOUR ACCESS   WARNING
-----   --------   ----------------  -----------   -------
 00     KYNEF           NONE              NONE      NO

INSTALLATION DATA
-----------------
NONE

APPLICATION DATA
----------------
BCPII

AUDITING
--------
FAILURES(READ)

NOTIFY
------
NO USER TO BE NOTIFIED
***************************** Bottom of Data ******************************
```

You will need one of these for EACH CPC that will be managed using BCPii

```
COMMAND ===> _                                              SCROLL ===> CSR
 F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
 F7=UP        F8=DOWN      F9=SWAP nex  F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

BCPii - z/OS end

System automatically tries to start BCPII address space at every IPL:

- Address space name is HWIBCPII.
- Address space shows up in SDSF DA, but not in D A,L output.

Address space can be stopped using P HWIBCPII command:

- Once the address space is stopped, no BCPII calls will be processed.
- ENF signal is broadcase to let any interested parties know that the interface is stopping.
- If P command doesn't work, you can use a CANCEL HWIBCPII

Address space can be started again using S HWISTART (HWISTART is delivered in SYS1.PROCLIB)

BCPii - z/OS end

There is currently no console command to check the status of BCPii.

If Pre-reqs are not in place at IPL time, address space will start and then stop.

So, if address space is active, that is at least a positive sign.

- Check for message HWI001I BCPII IS ACTIVE among IPL messages
- Doesn't guarantee that every CPC has been set up to support BCPII
- Currently the only way to check is from a program that uses the BCPII API

Start BCPii

Having completed the setup work on our CPC and in RACF, we now start BCPii address space:

```
  Display  Filter  View  Print  Options  Search  Help
 ------------------------------------------------------------------------------
 SDSF OPERLOG  DATE 08/04/2012     0 WTORS   1 FILTER            COLUMNS 52- 131
      -----                  ----PAGING COUNTS---
000210  -JOBNAME  STEPNAME PROCSTEP     RC    EXCP     CPU     SRB    VECT    VAFF
        CLOCK   SERV  PG    PAGE    SWAP     VIO SWAPS
000210  -HWISTART STARTING HWISTART     00       1     .00     .00     .00     .00
          .0      39    0       0       0       0      0
000210  -HWISTART ENDED.  NAME-                          TOTAL CPU TIME=    .00
        TOTAL ELAPSED TIME=     .0
000010  $HASP395 HWISTART ENDED
000200  IEA989I SLIP TRAP ID=X33E MATCHED.  JOBNAME=*UNAVAIL, ASID=012D.
000201  IEF196I           1 //IEESYSAS JOB MSGLEVEL=1
000201  IEF196I           2 //HWIBCPII EXEC IEESYSAS,PROG=HWIAMIN2
000201  IEF196I  STMT NO. MESSAGE
000201  IEF196I           2 IEFC001I PROCEDURE IEESYSAS WAS EXPANDED USING
        SYSTEM
000201  IEF196I LIBRARY SYS1.PROCLIB
000201  IEF196I           3 XXIEESYSAS PROC PROG=IEFBR14
000201  IEF196I           4 XXIEFPROC  EXEC PGM=&PROG
000201  IEF196I             XX* THE IEESYSAS PROCEDURE IS SPECIFIED IN THE
000201  IEF196I             XX* PARAMETER LIST TO IEEMB881 BY MVS COMPONENTS
000201  IEF196I             XX* STARTING FULL FUNCTION SYSTEM ADDRESS SPACES.
000201  IEF196I             IEFC653I SUBSTITUTION JCL - PGM=HWIAMIN2
000200  IEE252I MEMBER CTIHWI00 FOUND IN SYS1.IBM.PARMLIB
000201  IEF196I IEF285I   SYS1.PARMLIB                                  KEPT
000201  IEF196I IEF285I   VOL SER NOS= #@$#M1.
000201  IEF196I IEF285I   SYS1.IBM.PARMLIB                              KEPT
000201  IEF196I IEF285I   VOL SER NOS= Z1DRE1.
000010  HWI016I THE BCPII COMMUNICATION RECOVERY ENVIRONMENT IS 962
000010  NOW ESTABLISHED.
000210  HWI007I BCPII IS ATTEMPTING COMMUNICATION WITH THE LOCAL CENTRAL 963
000210  PROCESSOR COMPLEX (CPC).
000010  HWI001I BCPII IS ACTIVE.
000000  IXC104I SYSTEM STATUS DETECTION PARTITIONING PROTOCOL ELIGIBILITY: 965
000000    SYSTEM CANNOT TARGET OTHER SYSTEMS.
000000      REASON: SYSPLEX COUPLE DATA SET NOT FORMATTED FOR THE PROTOCOL
000000    SYSTEM IS NOT ELIGIBLE TO BE TARGETED BY OTHER SYSTEMS.
000000      REASON: SYSPLEX COUPLE DATA SET NOT FORMATTED FOR THE PROTOCOL
****************************** BOTTOM OF DATA ********************************
 COMMAND INPUT ===> _                                          SCROLL ===> CSR
  F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=IFIND     F6=BOOK
  F7=UP        F8=DOWN      F9=SWAP nex F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

BCPii Prerequisites

Software:

- z/OS 1.11 (included in the base)
- z/OS 1.10 with APAR OA25426

Hardware:

- The program *issuing* the BCPii calls must be running on any CPC supported by z/OS 1.11 (z900 or later)
  - It is always wise to keep CPCs (even old ones) at current microcode levels
- The HWICMD function can only be used against z9 or later with the following microcode levels:
  - z9: G40965.133
  - z10: F85906.116

BCPii further information

z/OS 1.11 MVS Programming: Callable Services for High-Level Languages:

- Primary BCPii documentation including installation instructions and BCPii API documentation.

z/OS 1.11 MVS System Commands:

- START HWISTART and STOP HWIBCPII commands.

z/OS 1.11 MVS Diagnosis: Tools and Service Aids:

- BCPii's CTRACE documentation.

z/OS MVS Programming: Authorized Assembler Services Reference, Volume 2 (EDT-IXG):

- BCPii's ENF68 documentation.

Various SHARE presentations - see www.share.org

ibm.com/redbooks

System Status Detection Partitioning Protocol

Redbooks

IBM ITSO - International Technical Support Organization

System Status Detection Partitioning Protocol

System Status Detection Partitioning Protocol (SSDPP) is an enhancement to failed-system handling designed to partition a failed system from the sysplex in a more timely way and with improved data integrity.

SSDPP achieves this by exploiting the z/OS BCPii support to communicate with the SE to obtain the current status of an LPAR.

System Status Detection Partitioning Protocol

When a z/OS 1.11 or later system is IPLed using a correctly formatted Sysplex CDS, it writes new information about itself into the CDS. It gets this information from BCPii:

- The network name of the CPC it is running on (netid.nau).
- The name of the LPAR it resides in.
- An IPL Token.
  - Both the hardware and the software know the IPL Token:
    - The IPL token is valid for the life of the IPL, as long as the system is still functioning.
    - If the LPAR is RESET, the IPL Token in the hardware will change.
    - If the LPAR waitstates (non-restartable), the IPL Token in the hardware will change.
    - If the LPAR is IPLed, the IPL token will change.

All of this information is available to the other members of the sysplex via the Sysplex CDS and the BCPii.

System Status Detection Partitioning Protocol

```
D XCF,C
IXC357I  13.30.33  DISPLAY XCF 214
SYSTEM #@$2 DATA

...

  SYSTEM STATUS DETECTION PARTITIONING PROTOCOL ELIGIBILITY:
    SYSTEM CANNOT TARGET OTHER SYSTEMS.
      REASON: SYSPLEX COUPLE DATA SET NOT FORMATTED FOR THE PROTOCOL
    SYSTEM IS NOT ELIGIBLE TO BE TARGETED BY OTHER SYSTEMS.
      REASON: SYSPLEX COUPLE DATA SET NOT FORMATTED FOR THE PROTOCOL

  SYSTEM NODE DESCRIPTOR: 002817.IBM.02.0000000B3BD5
                          PARTITION: 21    CPCID: 00

  SYSTEM IDENTIFIER: 3BD52817 21000008

  NETWORK ADDRESS: N/A

  PARTITION IMAGE NAME: N/A

  IPL TOKEN: N/A
```

Obtained via BCPii (if SSD is active)

System Status Detection Partitioning Protocol

What do I need to do to enable SSDPP?

- The systems that will drive the System Status Detection Partitioning Protocol processing, or be the target of such processing, MUST be running on z10 EC GA2 or z10 BC GA1 or later.
- BCPii must be configured and functioning.
- XCFAS must be defined as TRUSTED to RACF or must have access to the required BCPii SAF profiles.
- Only z/OS 1.11 or later systems can exploit SSDPP, but previous levels can tolerate the new Sysplex CDS format that is required for SSDPP.

System Status Detection Partitioning Protocol

Let's check the format of our current sysplex CDS....

```
D XCF,C,TYPE=SYSPLEX
IXC358I  15.24.12  DISPLAY XCF 977
SYSPLEX COUPLE DATA SETS
PRIMARY    DSN: SYS1.XCF.CDS03
           VOLSER: #@$#X1     DEVN: D20F
           FORMAT TOD         MAXSYSTEM MAXGROUP(PEAK) MAXMEMBER(PEAK)
           04/12/2012 14:31:32        4     500   (42)      303    (8)
           ADDITIONAL INFORMATION:
            ALL TYPES OF COUPLE DATA SETS ARE SUPPORTED
            GRS STAR MODE IS SUPPORTED
ALTERNATE  DSN: SYS1.XCF.CDS04
           VOLSER: #@$#X2     DEVN: D30F
           FORMAT TOD         MAXSYSTEM MAXGROUP       MAXMEMBER
           04/12/2012 14:31:36        4     500            303
           ADDITIONAL INFORMATION:
            ALL TYPES OF COUPLE DATA SETS ARE SUPPORTED
            GRS STAR MODE IS SUPPORTED
```

No mention of SSDPP support, so we need to move to correctly formatted Sysplex couple data sets.

System Status Detection Partitioning Protocol

Format 3 new Sysplex CDSs (primary, alternate, and spare) using the SSTATDET keyword:

```
//DEFCOUP  JOB (0,0),'DEF XCF CDSS',NOTIFY=&SYSUID,
//    CLASS=A,MSGCLASS=X,REGION=0M
//STEP1    EXEC PGM=IXCL1DSU
//STEPLIB  DD   DSN=SYS1.MIGLIB,DISP=SHR
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
     DEFINEDS SYSPLEX(#@$#PLEX)
              DSN(SYS1.XCF.CDS05) VOLSER(#@$#X1)
              MAXSYSTEM(4)
              CATALOG
         DATA TYPE(SYSPLEX)
              ITEM NAME(GRS) NUMBER(1)
              ITEM NAME(GROUP) NUMBER(500)
              ITEM NAME(MEMBER) NUMBER(303)
              ITEM NAME(SSTATDET) NUMBER(1)
...
/*
```

System Status Detection Partitioning Protocol

Enabling SSD (cont)...

- Issue the SETXCF COUPLE,ACOUPLE=dsn and SETXCF COUPLE,PSWITCH commands to roll the new CDSs into production.
  - Note that after you activate a new CDS formatted for SSD, it may take a few seconds before you see:

```
IXC103I SYSTEM IDENTIFICATION INFORMATION 033
  CONNECTION STATUS:    CONNECTED
  SYSTEM NAME:          #@$2
  SYSTEM NUMBER:        0100000E
  IMAGE NAME:           A21
  NODE DESCRIPTOR:      002817.IBM.02.0000000B3BD5
  PARTITION NUMBER:     21
  CPC ID:               00
  NETWORK ADDRESS:      USIBMSC.SCZP301
  IPL TOKEN:            C9F849E0 890FC7A5
IXC104I SYSTEM STATUS DETECTION PARTITIONING PROTOCOL ELIGIBILITY: 034
  SYSTEM CAN TARGET OTHER SYSTEMS.
  SYSTEM IS ELIGIBLE TO BE TARGETED BY OTHER SYSTEMS.
IXC111I LOGICAL PARTITION REMOTE CONNECTION INFORMATION 035
  CONNECTION STATUS:    CONNECTED
  SYSTEM NAME:          #@$3
  SYSTEM NUMBER:        0200000F
  IMAGE NAME:           A22
  NETWORK ADDRESS:      USIBMSC.SCZP301
  IPL TOKEN:            C9F84E37 44695DEB
  DIAG INFO:            N/A
```

System Status Detection Partitioning Protocol

Check Sysplex CDS format now:

```
D XCF,C,TYPE=SYSPLEX
IXC358I  15.43.54  DISPLAY XCF 046
SYSPLEX COUPLE DATA SETS
PRIMARY    DSN: SYS1.XCF.CDS05
           VOLSER: #@$#X1     DEVN: D20F
           FORMAT TOD         MAXSYSTEM MAXGROUP(PEAK) MAXMEMBER(PEAK)
           08/04/2012 15:33:31        4     500   (42)      303    (8)
           ADDITIONAL INFORMATION:
            ALL TYPES OF COUPLE DATA SETS ARE SUPPORTED
            GRS STAR MODE IS SUPPORTED
           SYSTEM STATUS DETECTION PROTOCOL IS SUPPORTED
ALTERNATE  DSN: SYS1.XCF.CDS06
           VOLSER: #@$#X2     DEVN: D30F
           FORMAT TOD         MAXSYSTEM MAXGROUP       MAXMEMBER
           08/04/2012 15:33:33        4     500             303
           ADDITIONAL INFORMATION:
            ALL TYPES OF COUPLE DATA SETS ARE SUPPORTED
            GRS STAR MODE IS SUPPORTED
           SYSTEM STATUS DETECTION PROTOCOL IS SUPPORTED
```

Remember to update COUPLExx to reflect new CDS names

System Status Detection Partitioning Protocol

Time to wait-state #@$2 again and see how long recovery takes this time.....

```
2012217 15:49:07.51 JOB19311 00000010  $HASP373 LOADWAIT STARTED - INIT 1    - CLASS A - SYS #@$2
2012217 15:49:07.51 JOB19311 00000010  ZTT JOB#=00000001: LOADWAIT EXECUTION STARTED -- LEVEL ZOS1C.06.001
                                       08/30/10 19.23
2012217 15:49:07.57          00000201  IEF196I IEF237I D057 ALLOCATED TO SYS00076
2012217 15:49:07.57          00000201  IEF196I IEF285I   MSPCT.ZOS1CZTT.LOADLIB                      KEPT
2012217 15:49:07.57          00000201  IEF196I IEF285I   VOL SER NOS= #@$#W1.
2012217 15:49:11.75 INTERNAL 00000010  IST1494I PATH SWITCH STARTED FOR RTP CNR00003 TO USIBMSC.#@$2M 284
                        284  00000010  IST1818I PATH SWITCH REASON: SHORT REQUEST RETRY LIMIT EXHAUSTED
                        284  00000010  IST314I END
2012217 15:49:16.52          00000000  IXC106I SYSTEM #@$2 285
                        285  00000000  RESET OR NEW IMAGE LOADED
2012217 15:49:16.52          00000000  IXC101I SYSPLEX PARTITIONING IN PROGRESS FOR #@$2 REQUESTED BY 286
                        286  00000000  XCFAS. REASON: SYSTEM RESET OR NEW IMAGE LOADED
2012217 15:49:16.53          00000200  IXC113I BCPII CONNECTION TO SYSTEM #@$2 RELEASED 287
                        287  00000200    DISCONNECT REASON:    SYSTEM REMOVED FROM SYSPLEX
                        287  00000200    IMAGE NAME:           A21
                        287  00000200    NETWORK ADDRESS:      USIBMSC.SCZP301
                        287  00000200    SYSTEM NUMBER:        0100000E
                        287  00000200    IPL TOKEN:            C9F849E0 890FC7A5
```

So it took about 30 minutes to implement and it saved about 2.5 minutes on every unplanned outage

System Status Detection Partitioning Protocol

Anything else?

- You can turn the use of SSDPP on or off dynamically at the system level using the SETXCF FUNCTIONS command and/or in COUPLExx member if you wish:
  - Default is ENABLED - this is the recommended setting
  - If you DISABLE SSDPP on a system, that system cannot be the target of any BCPii-related actions and will not use BCPii to initiate actions against any other systems.

System Status Detection Partitioning Protocol

Summary:

- Prereqs:
  - z10 GA2 or later
  - z/OS 1.11
  - Correctly formatted Sysplex CDS
  - Implement BCPii
- System Status Detection Partitioning Protocol is a significant step forward. This is the most fundamental change to handling of system failures since the introduction of SFM.
- Easy to implement.
- You can start to enable it as soon as your first z10 z/OS system moves to z/OS 1.11 - no need to wait for the whole sysplex to be upgraded.

ibm.com/redbooks

AutoIPL

Redbooks

IBM ITSO - International Technical Support Organization

AutoIPL overview

AutoIPL feature was delivered with z/OS 1.10 and supports z9 or later CPCs.

Provides the ability to:

- Automatically IPL Stand Alone Dump and/or z/OS following certain wait states
- Tell a system to take a stand alone dump on the V XCF,sysnm,OFFLINE command. Removes need to use HMC.
- Tell a system to shutdown and then automatically re-IPL itself on the V XCF,sysnm,OFFLINE command.
  - No interaction with HMC required.
  - Can IPL from existing sysres or a different sysres.

All of this function requires..... ONE extra line in your DIAGxx member...

AutoIPL overview

Each system is responsible for telling the CPC that it is running on what actions should be taken if it enters certain wait states

- Each system reads the DIAGxx member that is pointed to by IEASYS00, or by a SET DIAG=xx command.
- System then passes that information over to the hardware.
- Remember that the information that is provided in DIAGxx will be used FOR THE NEXT IPL. So if you want to change what happens at the next IPL, you MUST update DIAGxx and issue the SET DIAG command NOW. If you wait for the system to read that information as part of the IPL, it is too late to influence how that IPL was handled.

AutoIPL

First, let's go through a typical IPL scenario:

- Shut down all applications on a system
- Issue V XCF,sysname,OFFLINE command
- Wait for system to go into a wait state
- Logon to HMC
- Select right CPC and right LPAR (hopefully!) and Activate it.

What was the elapsed time from the V XCF,OFFLINE to the point where the system is IPLed and coming back up (msg IEE389I)?

AutoIPL

Now let's enable AutoIPL and use that for the IPL...

In the DIAGxx member, add the following line:

- AUTOIPL SADMP(NONE) MVS(LAST)
- This indicates that z/OS should be auto-IPLed off the same sysres as the last time, using the same parms as the last time.

Issue RO *ALL,SET DIAG=xx

- Note that you cannot concatenate DIAGxx members on the SET command

Now issue V XCF,sysname,OFFLINE,REIPL

How long does it take from V XCF,OFFLINE to IEE389I this time?

AutoIPL

In order to have the system automatically take a standalone dump and then RE-IPL, set up SAD with job like this:

```
//KARANASM JOB (0,0),CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID
//OSG      EXEC  PGM=AMDSAOSG
//SYSLIB   DD DISP=SHR,DSN=SYS1.MACLIB,UNIT=3390,VOL=SER=Z1DRS1
//         DD DISP=SHR,DSN=SYS1.MODGEN,UNIT=3390,VOL=SER=Z1DRS1
//TRK0TEXT DD DSN=&TRK0TEXT,DISP=(,PASS),
//            SPACE=(4096,(2,1)),UNIT=SYSALLDA
//DSFSYSIN DD DSN=&DSFSYSIN,DISP=(,PASS),
//            SPACE=(80,(5,5)),UNIT=SYSALLDA
//GENPRINT DD SYSOUT=*
//GENPARMS DD *
         AMDSADMP IPL=D3390,VOLSER=#@$#M1,                              X
               CONSOLE=(SYSC),DDSPROMPT=NO,                             X
               OUTPUT=D9C08,NOPROMPT,MINASID=PHYSIN
         END
/*
//DPLTEXT  DD  DISP=SHR,DSN=SYS1.NUCLEUS(AMDSADPL)
//DVITEXT  DD  DISP=SHR,DSN=SYS1.NUCLEUS(AMDSADVI)
//IPITEXT  DD  DISP=SHR,DSN=SYS1.NUCLEUS(AMDSAIPI)
//IPLTEXT  DD  DISP=SHR,DSN=SYS1.NUCLEUS(AMDSAIPD)
//PGETEXT  DD  DISP=SHR,DSN=SYS1.NUCLEUS(AMDSAPGE)
//PUTIPL   EXEC PGM=ICKDSF
//IPLDEV   DD DISP=OLD,UNIT=SYSALLDA,VOL=(PRIVATE,RETAIN,SER=#@$#M1)
//TRK0TEXT DD DSN=&TRK0TEXT,DISP=(OLD,DELETE)
//SYSIN    DD DSN=&DSFSYSIN,DISP=(OLD,DELETE)
//SYSPRINT DD SYSOUT=*
//DSFDUMP  DD SYSOUT=*
```

AutoIPL

Then set up DIAGxx member with:

- AUTOIPL SADMP(dddd,SNSYSC4) MVS(LAST)

Issue RO *ALL,T DIAG=xx

Next time your system goes into a disabled wait state, it should automatically take a SAD and then re-IPL

- For info about how AutoIPL handles various wait states, see the section titled "Wait state action table (WSAT)" in Planning: Operations book.

To test this, issue V XCF,sysnm,OFFLINE,SADMP,REIPL

AutoIPL

Summary:

- Delivered with z/OS 1.10.
- Works on z9 and later.
- Can be used to IPL z/OS from the same sysres as last time OR from a different sysres (if you are moving to a new service level, for example)
- Highly recommended to use this to automate taking of standalone dumps.
- Should NOT be used if you are using GDPS/PPRC, because GDPS wants to manage all IPLs.

ibm.com/redbooks

JES2 Dynamic Proclib

Redbooks

IBM ITSO - International Technical Support Organization

JES2 Dynamic Proclib

Who amongst you can honestly say that you never had a JCL error in your JES2 proc?

Did you ever have someone delete a JES2 proclib and only find out that it is gone the next time you tried to IPL?

How much fun is it to do a MAS-wide restart of JES2 so you can add a proclib to JES2?

The answer to your problems is here (and has been here for the last 10 years!) thanks to those nice JES2 Development people - Dynamic Proclib support

JES2 Dynamic Proclib

What can you do with Dynamic Proclib?

- Change proclib concatenations without touching JES2 JCL.
- Bypass errors in proclib definitions.
- Display PROCxx definitions.
- Dynamically add PROCxx definitions.
- Dynamically MODIFY existing PROCxx definitions.
- Add a new PROCxx definition, test it, and then rename it.
- Delete PROCxx definitions.

Let's see some examples...

JES2 Dynamic Proclib

Here is what we started with:

```
//JES2     PROC M=J2USECF
//IEFPROC EXEC PGM=HASJES20,TIME=1440,DPRTY=(15,14)
//HASPLIST  DD DDNAME=IEFRDER
//HASPPARM  DD DSN=SYS1.PARMLIB(&M),DISP=SHR
//PROC00    DD DSN=SYS1.DIST.PROCLIB,DISP=SHR
//           DD DSN=SYS1.PROCLIB,DISP=SHR
//           DD DSN=SYS1.IBM.PROCLIB,DISP=SHR
```

JES2 Dynamic Proclib

To add a new data set to PROC00, we need to update JCL:

```
//JES2     PROC M=J2USECF
//IEFPROC EXEC PGM=HASJES20,TIME=1440,DPRTY=(15,14)
//HASPLIST  DD DDNAME=IEFRDER
//HASPPARM  DD DSN=SYS1.PARMLIB(&M),DISP=SHR
//PROC00    DD DSN=SYS1.DIST.PROCLIB,DISP=SHR
//           DD DSN=SYS1.PROCLIB,DISP=SHR
//           DD DSN=SYS1.IBM.PROCLIB,DISP=SHR
//           DD DSN=SYS1.KYNEF.PROCLIB,DISP=SHR
```

And do a MAS-wide JES2 restart.

JES2 Dynamic Proclib

What happens if we mess up the JCL?

```
S JES2,PARM='NOREQ'
IEF196I         1 //JES2     JOB MSGLEVEL=1
IEF196I         2 //STARTING EXEC JES2,PARM='NOREQ'
IEF196I  STMT NO. MESSAGE
IEF196I         2 IEFC001I PROCEDURE JES2 WAS EXPANDED USING SYSTEM
IEF196I LIBRARY SYS1.PROCLIB
IEF196I         3 XXJES2    PROC M=J2USECF
IEF196I         4 XXIEFPROC EXEC PGM=HASJES20,TIME=1440,DPRTY=(15,14)
IEF196I         5 XXHASPLIST  DD DDNAME=IEFRDER
IEF196I         6 XXHASPPARM  DD DSN=SYS1.PARMLIB(&M),DISP=SHR
IEF196I           IEFC653I SUBSTITUTION JCL - DSN=SYS1.PARMLIB(J2USECF
),
IEF196I DISP=SHR
IEF196I         7 XXPROC00    DD DSN=SYS1.DIST.PROCLIB,DISP=SHR
IEF196I         8 XX          DD DSN=SYS1.PROCLIB,DISP=SHR
IEF196I         9 XX          DD DSN=SYS1.IBM.PROCLIB,DISP=SHR
IEF196I        10 XX          DD DSN=SYS1.KYNEFPROCLIB,DISP=SHR
IEF196I        10 IEF642I EXCESSIVE PARAMETER LENGTH IN THE DSNAME
FIELD
IEF677I WARNING MESSAGE(S) FOR JOB JES2     ISSUED
IEF196I        10 IEF686I DDNAME REFERRED TO ON DDNAME KEYWORD IN
PRIOR
IEF196I STEP WAS NOT RESOLVED
IEF452I JES2     - JOB NOT RUN - JCL ERROR
IEE122I START COMMAND JCL ERROR
```

Oops.....

JES2 Dynamic Proclib

So how would we do this using Dynamic Proclib?

- This is what we had in the JES2 Proc:

```
//JES2    PROC M=J2USECF
//IEFPROC EXEC PGM=HASJES20,TIME=1440,DPRTY=(15,14)
//HASPLIST  DD DDNAME=IEFRDER
//HASPPARM  DD DSN=SYS1.PARMLIB(&M),DISP=SHR
//PROC00    DD DSN=SYS1.DIST.PROCLIB,DISP=SHR
//          DD DSN=SYS1.PROCLIB,DISP=SHR
//          DD DSN=SYS1.IBM.PROCLIB,DISP=SHR
//          DD DSN=SYS1.KYNEF.PROCLIB,DISP=SHR
```

- This is how we do the same thing in the JES2 Parm member

```
PROCLIB(PROC00) DD(1)=(DSN=SYS1.DIST.PROCLIB),
                DD(2)=(DSN=SYS1.PROCLIB),
                DD(3)=(DSN=SYS1.IBM.PROCLIB),
                DD(4)=(DSN=SYS1.KYNEF.PROCLIB)
```

JES2 Dynamic Proclib

What happens if we mess up the JES2 parm?

```
PROCLIB(PROC00) DD(1)=(DSN=SYS1.DIST.PROCLIB),
                DD(2)=(DSN=SYS1.PROCLIB),
                DD(3)=(DSN=SYS1.IBM.PROCLIB),
                DD(4)=(DSN=SYS1.KYNEFPROCLIB)
```

Automatic replies

```
$HASP466 PARMLIB    STMT    11  DD(4)=(DSN=SYS1.KYNEFPROCLIB)
$HASP003 RC=(03),DD(4)  - INVALID PARAMETER STATEMENT
REPLY 13,END
013 $HASP469 REPLY PARAMETER STATEMENT, CANCEL, OR END
IEE600I REPLY TO 013 IS;END
IEF196I IEF285I   SYS1.PARMLIB                                KEPT
IEF196I IEF285I   VOL SER NOS= #@$#M1.
$HASP451 ERROR ON JES2 PARAMETER LIBRARY
REPLY 14,Y
014 $HASP441 REPLY 'Y' TO CONTINUE INITIALIZATION OR 'N' TO TERMINATE
IEE600I REPLY TO 014 IS;Y
IEF196I IEF237I D056 ALLOCATED TO SYS00007
$HASP478 INITIAL CHECKPOINT READ IS FROM CKPT1 779
         (STRNAME JES2CKPT_1)
         LAST WRITTEN MONDAY,  6 AUG 2012 AT 21:41:21 (GMT)
$HASP493 JES2 MEMBER-#@$2 HOT START IS IN PROGRESS - z11 MODE
```

JES2 Dynamic Proclib

How do we add a new PROCxx concatenation?

```
$ADD PROCLIB(PROC02),DD1=DSN=SYS1.KYNEF.PROCLIB
$HASP319 PROCLIB(PROC02)    DD(1)=(DSNAME=SYS1.KYNEF.PROCLIB)
RO #@$2,$D PROCLIB(PROC02)
$D PROCLIB(PROC02)
$HASP319 PROCLIB(PROC02)    DD(1)=(DSNAME=SYS1.KYNEF.PROCLIB)
```

JES2 Dynamic Proclib

For more information, refer to:

- JES2 Commands
- JES2 Initialization and Tuning Reference
- z/OS 1.2 Implementation, SG24-6235

IBM Redbooks

The ITSO is looking at what we can do to make IBM Redbooks deliverables more valuable to both our Generation Y audience AND our "more experienced" users:

- More smaller, architecture-level, books (RedGuides)
- How-to videos
- Blog entries to publicize new books, residencies, workshops, and so on.
- More face-to-face classes
- More online classes
- Twitter and Facebook presence

Please let us know what YOU think we need to do to help you perform your job more effectively. Send an email to kyne@us.ibm.com or karansin@us.ibm.com with your suggestions and comments

Wrap up

Any questions?

Related sessions:

- Thursday 08:00-09:00 Session 11713, Steve Warren (Mr BCPii), "BCPii Programming Beyond the Basics" for everything you could ever want to know about BCPii
- Thursday 09:30-10:30, Session 11722: "z/OS Planned Outages - Control them, instead of the other way around" (repeat of session from Atlanta)

If you have any suggestions for improving this material, please let us know.

This is NOT Poughkeepsie.....

Neither is this.....

Shameless advertising

Come to Poughkeepsie to take part in a project with other subject matter experts from all over the world to write a Redbook.....

- IBM covers all travel expenses, hotel, meal allowance, car, etc...
- Your mission is to learn as much as you can about the latest and greatest IBM technology and document your experiences
- Gain fame and fortune (well, at least, you will get your name on the front cover of a Redbook)

If you think you might be interested, keep an eye on
http://www.redbooks.ibm.com/residents.nsf/ResIndex/
or sign up for automatic notification at
https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

Thank you

Please remember to hand in your session evaluations:

Session 11806