z/OS V2R4 BCPii Update

Stephen Warren
Senior Technical Staff Member
Client Architect, IBM Worldwide Client Experience Center

November 2019
Session BB

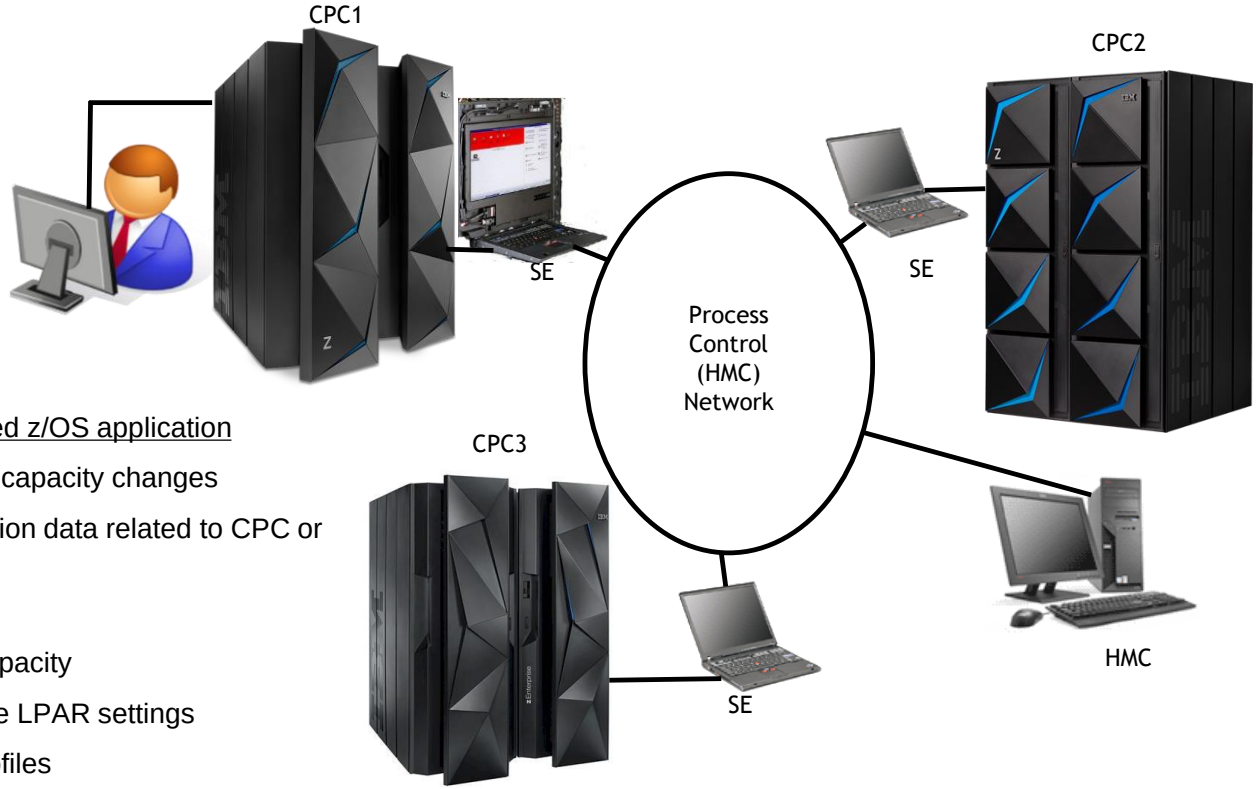



Agenda

- Quick overview of BCPii
- Major new BCPii enhancement (V2R4)
  - New LPAR Group support
- New BCPii enhancements (V2R3 and hardware)
  - New BCPii Security Controls (z14 & higher)
  - BCPii performance improvements
  - Absolute capping base support
  - Absolute capping group support
  - New more flexible HWICMD2 service
  - Support for larger data returned from SE
  - Support for dynamic CPC name change
  - New HWISET2 service (multiple attribute set)

Quick overview of z/OS BCPii

Overview - What is BCPii?



Authorized z/OS application

- Monitor status or capacity changes
- Obtain configuration data related to CPC or image
- Re-ipl an image
- Change temp. capacity
- Query and update LPAR settings
- Set activation profiles

Quick overview - What is BCPii?

- Base Control Program internal interface
  - Allows authorized z/OS applications to have HMC-like control over systems in the process control (HMC) network
  - A set of authorized APIs provided
- Does not use any external network.
  - Communicates directly with the SE rather than going over an IP network.
- A z/OS address space that manages authorized interaction with the interconnected hardware

Quick overview - Who uses BCPii?

- **z/OS operating system components**
  - System Status Detection (SSD) provided in Sysplex Failure Manager (SFM)
  - Capacity Provisioning Manager (CPM)
  - Hardware Configuration Definition (HCD)
- **Vendor applications**
  - Control center, system management applications
  - Many vendor products use BCPii
- **In-house (customer-written) applications**

Quick overview – BCPii Installation Steps

- 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 steps have been carried out:
  - Setup on the HMC/SE:
    - Give partition BCPii authority from one or more other partitions (z14) or Enable Cross Partition Authority for every LPAR that you want to be able to issue or be the target of BCPii commands. (Prior to z14)
    - Enable SNMP and define the Community Name.
      - Both of these can be changed non-disruptively if you wish
  - Setup in z/OS
  - Setup with SAF Security authorizations (in z/OS)

New BCPii Enhancement (V2R4)

BCPii LPAR Group Controls

Current BCPii hardware entities that can be read or modified

- BCPii connection types
  - CPCs
  - Images
  - Capacity Records
  - Activation Profiles
    - Reset
    - Image
    - Load
  - User Defined Image Groups

Z Hardware entities not “well-supported” in BCPii prior to 2.4

LPAR Group (**LPAR Capacity Group**)

- Allows soft capping across multiple LPARs defined in the same LPAR group.
- Uses z/OS workload management (WLM) to manage the four-hour rolling average utilization of all of the LPARs within the same group
- Does not enforce relative weight (no PR/SM™ hard cap set)
- Use the Hardware Management Console (HMC) to define the capacity group

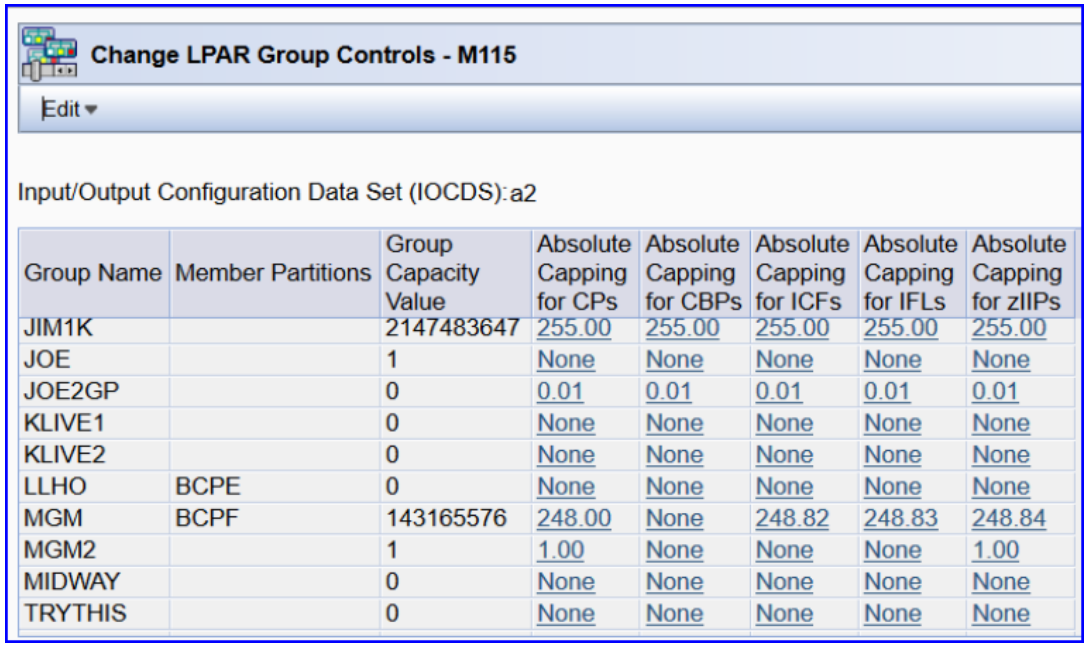
Group Profile (another activation profile type)

- Allows an installation to save a desired LPAR group configuration that will be utilized when an image is activated.

LPAR Capacity Groups on the SE

Operational Customization ->

Change **LPAR Group** Controls



Change LPAR Group Controls - M115

Edit

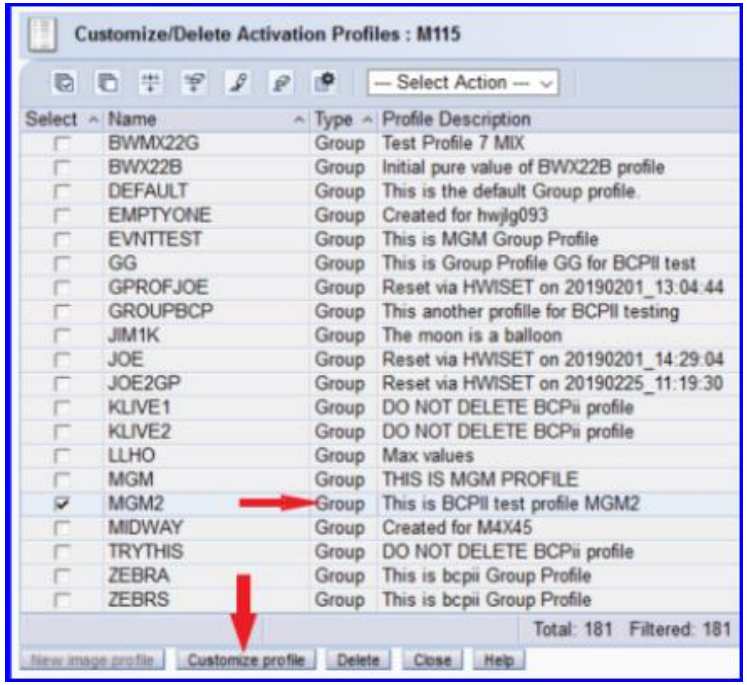
Input/Output Configuration Data Set (IOCDS): a2

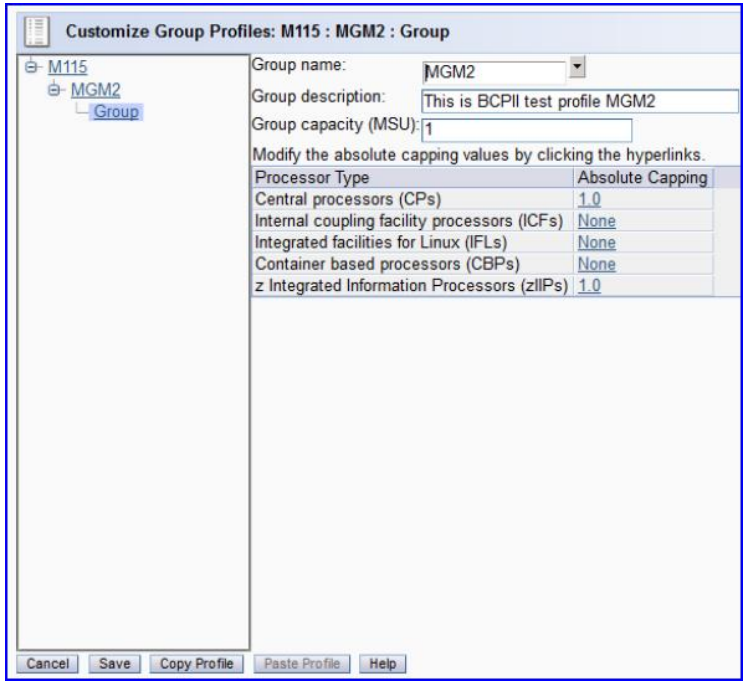
Group Name	Member Partitions	Group Capacity Value	Absolute Capping for CPs	Absolute Capping for CBPs	Absolute Capping for ICFs	Absolute Capping for IFLs	Absolute Capping for zIIPs
JIM1K		2147483647	255.00	255.00	255.00	255.00	255.00
JOE		1	None	None	None	None	None
JOE2GP		0	0.01	0.01	0.01	0.01	0.01
KLIVE1		0	None	None	None	None	None
KLIVE2		0	None	None	None	None	None
LLHO	BCPE	0	None	None	None	None	None
MGM	BCPF	143165576	248.00	None	248.82	248.83	248.84
MGM2		1	1.00	None	None	None	1.00
MIDWAY		0	None	None	None	None	None
TRYTHIS		0	None	None	None	None	None

Group Activation Profiles on the SE

Operational Customization ->

Customize/Delete **Activation Profiles**





What does BCPii support today for LPAR groups?

- **HWI_GROUP_PROFILE_CAPACITY (Query, Set) – retrieve or set the dynamic workload unit capacity for a group of images**
  - Requires an image connection
  - Requires knowledge that this image is in the particular group of images
- **Absolute capping attributes for all processor types (Query, Set) – retrieve or set the dynamic workload absolute capping value for a group of images**
  - For example, for CPs: HWI_GROUP_PROF_ABSCAP, HWI_GROUP_PROF_ABSCAPVAL
  - Requires an image connection
  - Requires knowledge that this image is in a particular group of images

Requires knowledge that may not be known and only sets live values

Introducing LPAR Group Support in BCPii

- What (Solution)
  - Enhancements will allow an application to:
    - Connect to LPAR capacity groups (live values) and group profiles (saved configuration)
    - List corresponding group members
    - Query and set attributes specific to the group type.

- Wow (Benefit / Value, Need Addressed)
  - To allow simplified programmatic control over LPAR Groups, both live groups and saved profiles, in the same manner that BCPii already provides for other hardware configuration entities.

Two new BCPii connection types!

- **HWI_LPAR_GROUP (LPAR Capacity Group)**
  - *Provides the user with real time information regarding what is currently configured for the group*
  - *Updates will take effect immediately for all active images associated with the group*
  - *Only available when targeting a z14 or higher machine*
- **HWI_GROUP_PROFILE (Group Profile)**
  - *These profiles allow the user to provide and alter information that will be used when an image is activated*
  - *The updates will not take effect until all active CPC images that correspond to the referenced Group Profile are deactivated and then re-activated*

Two new BCPii List types!

- **HWI_LIST_LPAR_GROUPS** (list of LPAR Capacity Groups)
  - List the LPAR capacity groups currently active on the specified CPC
  - Each name returned can be used on subsequent HWICONN requests with connection type of HWI_LPAR_GROUP
  - Only available when targeting a z14 or higher
- **HWI_LIST_GROUP_PROFILES** (list of Group activation profiles)
  - List the group profile names defined on a particular CPC
  - Each name returned can be used on subsequent HWICONN requests with connection type of HWI_GROUP_PROFILE

Listing LPARs within the 2 new connection types

- **HWI_LPAR_GROUP** connection
  - **HWI_LIST_IMAGES list type**
    - Each name returned can be used on a subsequent HWICONN call with connection type of HWI_IMAGE
- **HWI_GROUP_PROFILE** connection
  - **HWI_LIST_IMAGE_ACTPROF list type**
    - Each name returned can be used on a subsequent HWICONN call with connection type of HWI_IMAGE_ACTPROF
- Both only available when targeting a z14 or higher

LPAR Group and Group Profile Security Setup

- Both LPAR Groups and Group Profiles are entities associated with a particular CPC.
- BCPii needs a profile defined in the FACILITY class that represents the particular CPC:

```
HWI.TARGET.netid.nau
     where netid.nau represents the 3- to 17-character SNA name of the particular CPC
```

- Access level required for the particular LPAR Capacity Group or Group Profile
  - **READ** access for application to LIST, CONNECT, or QUERY
  - **UPDATE** access for application to SET
- Example: Joe needs to QUERY attributes associated with a specific Group Profile on CPC NET1.CPC001:

```
RDEFINE FACILITY HWI.TARGET.NET1.CPC001 UACC(NONE)
PERMIT HWI.TARGET.NET1.CPC001 CLASS(FACILITY) ID(JOE) ACCESS(READ)
SETROPTS RACLIST(FACILITY) REFRESH
```

LPAR Group and Group profile attributes

Attribute definitions specific for LPAR Group and Group Profile support:

Attributes	Description
HWI_NAME	The name of the object the LPAR Group or Group profile.
HWI_OBJTYPE	The type of object the group profile object represents.
HWI_PROFILE_DESCRIPTION	The description of the profile.
HWI_GROUP_PROFILE_CAPACITY	The capacity value of the object the LPAR Group or Group profile represents.
HWI_GROUP_PROF_ABSCAP	Used to enable/disable absolute capping for General Purpose Processors.
HWI_GROUP_PROF_ABSCAPVAL	The value used for absolute capping for General Purpose Processors.
HWI_GROUP_PROF_ICFABSCAP	Used to enable/disable absolute capping for Internal Coupling Facility (ICF) processors.
HWI_GROUP_PROF_ICFABSCAPVAL	The value used for Internal Coupling Facility (ICF) absolute capping.
HWI_GROUP_PROF_IFLABSCAP	Used to enable/disable absolute capping for Integrated Facility for Linux (IFL) processors.
HWI_GROUP_PROF_IFLABSCAPVAL	The value used for Integrated Facility for Linux (IFL) absolute capping.
HWI_GROUP_PROF_IIPABSCAP	Used to enable/disable absolute capping for z Integrated Information Processor (zIIP) processors.
HWI_GROUP_PROF_IIPABSCAPVAL	The value used for z Integrated Information Processor (zIIP) absolute capping.

LPAR Group Usage

LPAR Groups

- Use **HWILIST** with ListType **HWI_LIST_LPAR_GROUPS** to list the LPAR Groups available on a CPC
- Use **HWICONN** with ConnectType of **HWI_LPAR_GROUP** to connect to a specific LPAR Group available on a CPC
- Use **HWILIST** with ListType **HWI_LIST_IMAGES** to list the CPC images (LPARs) associated with a specific LPAR Group
- The following attributes are supported for **HWIQUERY / HWISET/ HWISET2** request associated with an LPAR Group

Attributes	Supported for HWIQUERY	Supported for HWISET/ HWISET2
HWI_NAME	x	
HWI_OBJTYPE	x	
HWI_GROUP_PROFILE_CAPACITY	x	x
HWI_GROUP_PROF_ABSCAP	x	x
HWI_GROUP_PROF_ABSCAPVAL	x	x
HWI_GROUP_PROF_ICFABSCAP	x	x
HWI_GROUP_PROF_ICFABSCAPVAL	x	x
HWI_GROUP_PROF_IFLABSCAP	x	x
HWI_GROUP_PROF_IFLABSCAPVAL	x	x
HWI_GROUP_PROF_IIPABSCAP	x	x
HWI_GROUP_PROF_IIPABSCAPVAL	x	x

Group Profile Usage

Group Profiles

- Use **HWILIST** with ListType **HWI_LIST_GROUP_PROFILES** to list the Group Profiles available on a CPC
- Use **HWICONN** with ConnectType of **HWI_GROUP_PROFILES** to connect to a specific Group Profile available on a CPC
- Use **HWILIST** with ListType **HWI_LIST_IMAGE_ACTPROF** to list the image activation profiles associated with a specific Group Profile
- The following attributes are supported for **HWIQUERY / HWISET/ HWISET2** request associated with an Group Profile

Attributes	Supported for HWIQUERY	Supported for HWISET/ HWISET2
HWI_NAME	x	
HWI_OBJTYPE	x	
HWI_PROFILE_DESCRIPTION	x	x
HWI_GROUP_PROFILE_CAPACITY	x	x
HWI_GROUP_PROF_ABSCAP	x	x
HWI_GROUP_PROF_ABSCAPVAL	x	x
HWI_GROUP_PROF_ICFABSCAP	x	x
HWI_GROUP_PROF_ICFABSCAPVAL	x	x
HWI_GROUP_PROF_IFLABSCAP	x	x
HWI_GROUP_PROF_IFLABSCAPVAL	x	x
HWI_GROUP_PROF_IIPABSCAP	x	x
HWI_GROUP_PROF_IIPABSCAPVAL	x	x

LPAR Group Example

Example code snippets querying and setting an LPAR Capacity group attribute:

```
...
LPAR_Group = 'LLHO'
ConnectType = HWI_LPAR_GROUP
ConnectTypeValue = left( strip(LPAR_Group), 8 )
address bcpii "hwiconn ",
              "ReturnCode ",
              "CPCConnectToken ",
              "OutConnectToken ",
              "ConnectType ",
              "ConnectTypeValue ",
              "DiagArea."

...

LPARGroupConnectToken = OutConnectToken
QueryParm.0 = 2
QueryParm.1.ATTRIBUTEIDENTIFIER = HWI_GROUP_PROFILE_CAPACITY
QueryParm.2.ATTRIBUTEIDENTIFIER = HWI_GROUP_PROF_ABSCAPVAL
address bcpii "hwiquery ",
              "ReturnCode ",
              "LPARGroupConnectToken ",
              "QueryParm. ",
              "DiagArea."
say 'Group Profile Capacity is ' QueryParm.1.ATTRIBUTEVALUE
say 'Group Profile AbsCapVal is ' QueryParm.2.ATTRIBUTEVALUE
...

SetType = HWI_GROUP_PROFILE_CAPACITY
SetTypeValue = 0
address bcpii "hwiset ",
              "ReturnCode ",
              "LPARGroupConnectToken ",
              "SetType ",
              "SetTypeValue ",
              "DiagArea."
```

***Connection** to the desired CPC has already been done earlier. We will connect to the LPAR Capacity Group LLHO.*

*Once connected to the LPAR Capacity Group, identify the attributes to query. Issue the **query**. After query, the values are stored in the stem variable.*

*Now, **set** the LPAR group capacity to 0.*

Group Profile Example

Example code snippets listing the image activation profile associated with a specific Group Profile:

```
...
ListType = HWI_LIST_GROUP_PROFILES
address bcpii  "hwilist",
               "ReturnCode",
               "CPCConnectToken",
               "ListType",
               "GRPList.",
               "DiagArea."
...
GRPName = GRPList.4
...
```

First you want to list the Group Profiles available on the CPC and obtain the name of the profile you're interested in

```
ConnectType = HWI_GROUP_PROFILE
ConnectTypeValue = left( strip(GRPName), 8 )
address bcpii "hwiconn",
              "ReturnCode",
              "CPCConnectToken",
              "GRPConnectToken",
              "ConnectType",
              "ConnectTypeValue",
              "DiagArea."
...
```

Once the specific Group Profile is identified, you want to connect to it which will provide you with a connection token associated with the Group Profile

```
ListType = HWI_LIST_IMAGE_ACTPROF
address bcpii "hwilist",
              "ReturnCode",
              "GRPConnectToken",
              "ListType",
              "myQueryParm.",
              "DiagArea."
```

Now you can issue a request to list the Image Activation Profiles associated with the specified Group Profile

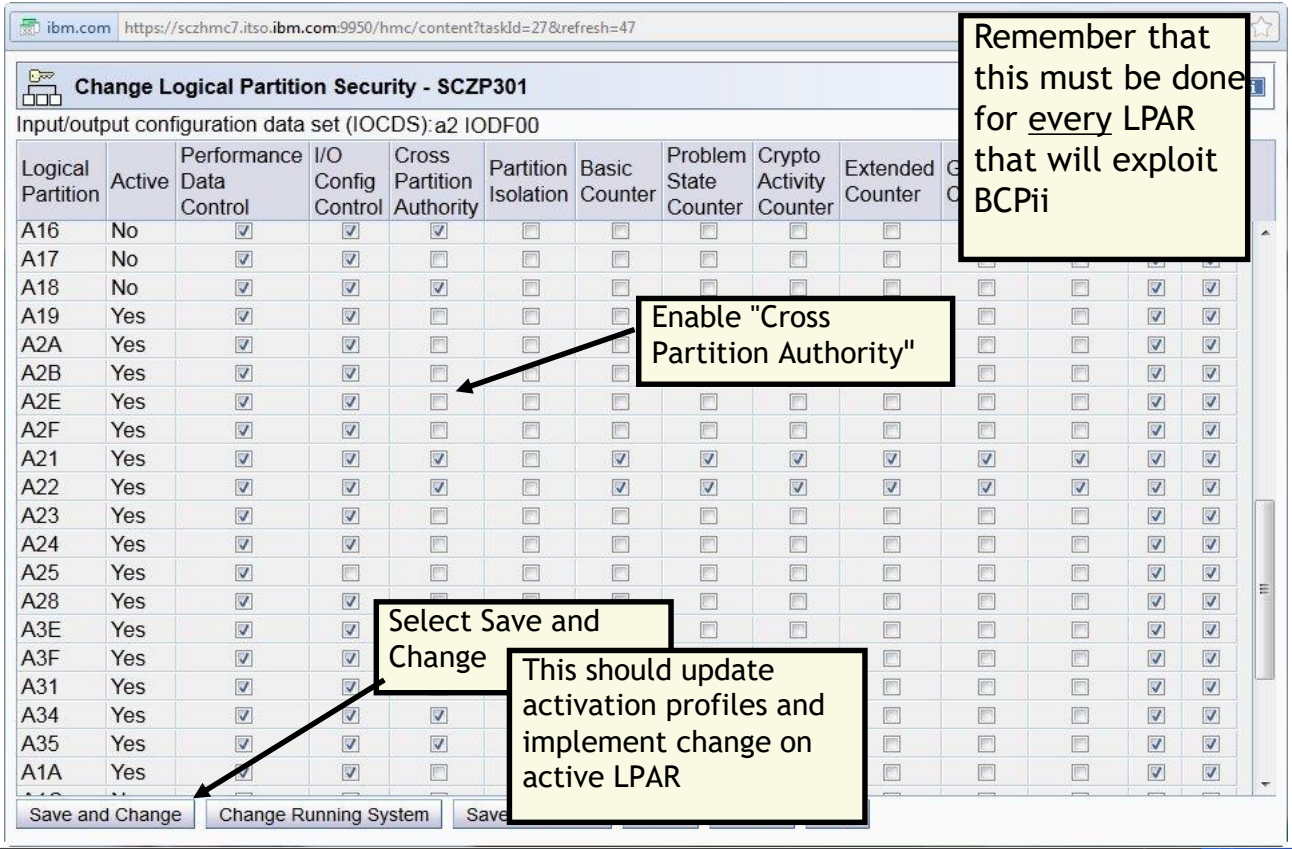
Diagnostics – HWISET / HWISET2 behaviors

Scenario	Result targeting LPAR Group connection	Result targeting Group Profile connection
No image / image activation profiles associated with the LPAR Group / Group Profile	RC = 101/CommErr = HWMCA_DE_INVALID_TARGET (requires active image)	Successful Change is immediate in: • LPAR Group Controls and Group Profile panels • HWIQUERY group profile attribute value(s)
ONLY INACTIVE images associated with the LPAR Group / Group Profile	RC = 101/ CommErr = HWMCA_DE_INVALID_TARGET (requires active image)	Successful Change is immediate in: • LPAR Group Controls and Group Profile panels • HWIQUERY group profile attribute values(s)
At least ONE ACTIVE image associated with the LPAR Group / Group Profile	Successful Change is immediate in: • LPAR Group Controls panel • HWIQUERY LPAR Group attribute values(s)	Successful • Change **IS NOT immediate** in LPAR Group Controls panel because there is an active image • Change **IS immediate** in HWIQUERY group profile attribute value(s) and Group Profile panel. Note: Later when all images transition to INACTIVE, group profile attribute values will take effect on both LPAR Group Controls panel and HWIQUERY of LPAR Group (also implies Group Profile)

New BCPii Enhancements (V2R3)

New BCPii Security Controls (z14 and higher)

BCPii Security Setup on SE/HMC (pre-z14)



Problems with pre-z14 BCPii security mechanism

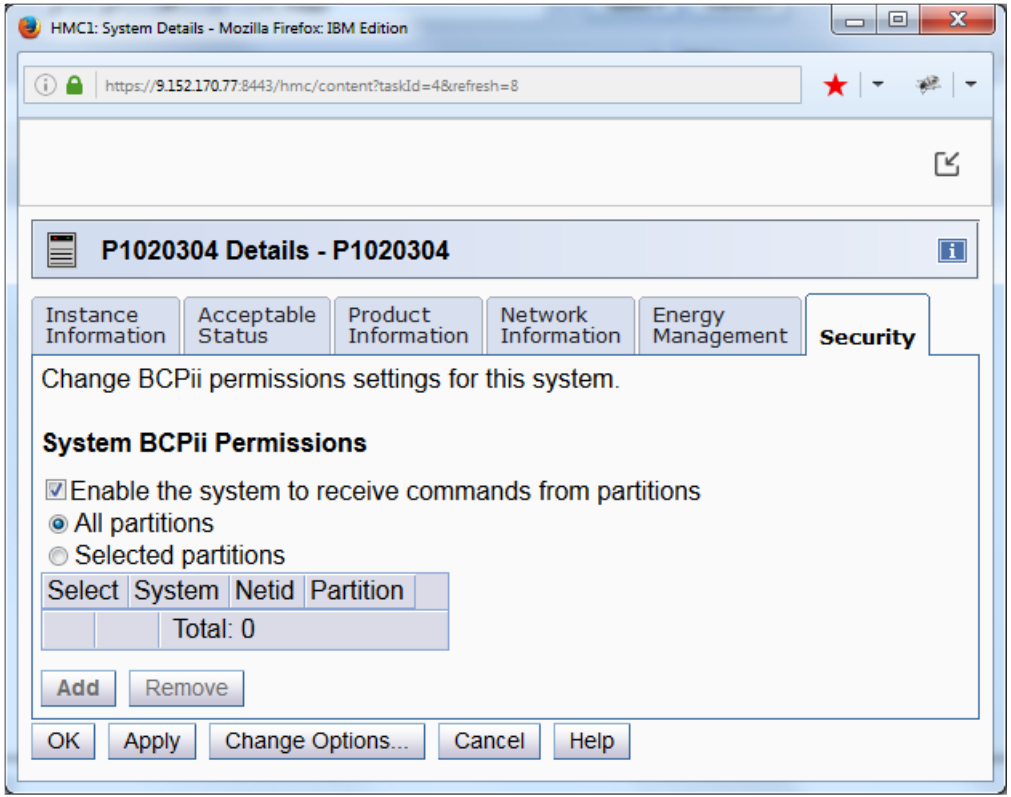
- **“Cross Partition Authority” not granular**
  - LPARs that have checkbox checked have unrestricted freedom to perform BCPii functionality against any LPAR that also has its checkbox checked
  - 100% reliant on each LPAR “playing nice” and having BCPii lockdown security in SAF to restrict access to other partitions
    - Unsuitable for service providers installations
    - Unsuitable for installations that have test LPARs that want to enable BCPii to test their applications but don’t have lockdown security
- **SHARE Top 50 requirement**
  - Close BCPii security exposure

- “z/OS 1.11 introduced a powerful new capability whereby XCF can use BCPii to query the status of an LPAR that has stopped updating its heartbeat in the CDS. There are many advantages to this new capability - it enables faster reaction to failed systems increases the likelihood of spin loop recovery being successful and protects you from operator error. To exploit this capability you have to enable BCPii and the only way to control the scope of BCPii actions is currently RACF. However if the service delivery department doesn’t have full control over RACF for every system running in the environment it would be possible for an installation-written program running in one LPAR to impact any LPAR that has been enabled for BCPii.”

New z14 BCPii Security Enhancements

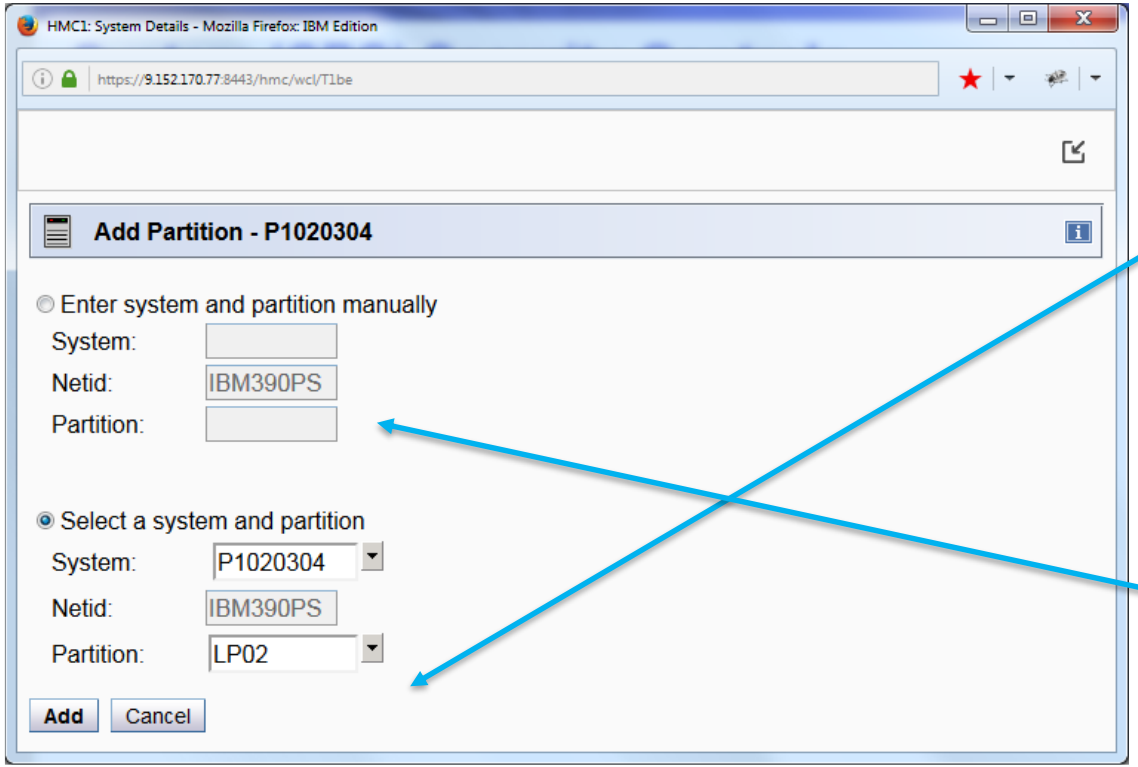
- **Granular system/partition access**
  - Enable/disable send capability per image
  - Enable/disable receive capability per image
    - Enable to receive from all partitions
    - Enable to receive from a list of selected partitions
  - Can limit which images can access CPC via System Details task (HMC or SE)
  - Can limit which images can access other images via Image activation profile and Change LPAR Security task (HMC or SE)
- **System Upgrade migration path**
  - Cross Partition Authority setting remains the same
    - Enabled maps to “BCPii Permission” of send enabled, receive enabled from all partitions
    - Disabled maps to “BCPii Permission” of send/receive disabled.

z14 BCPii CPC Security Controls



- Defaults to allow the CPC to receive commands from any BCPii-enabled partition.

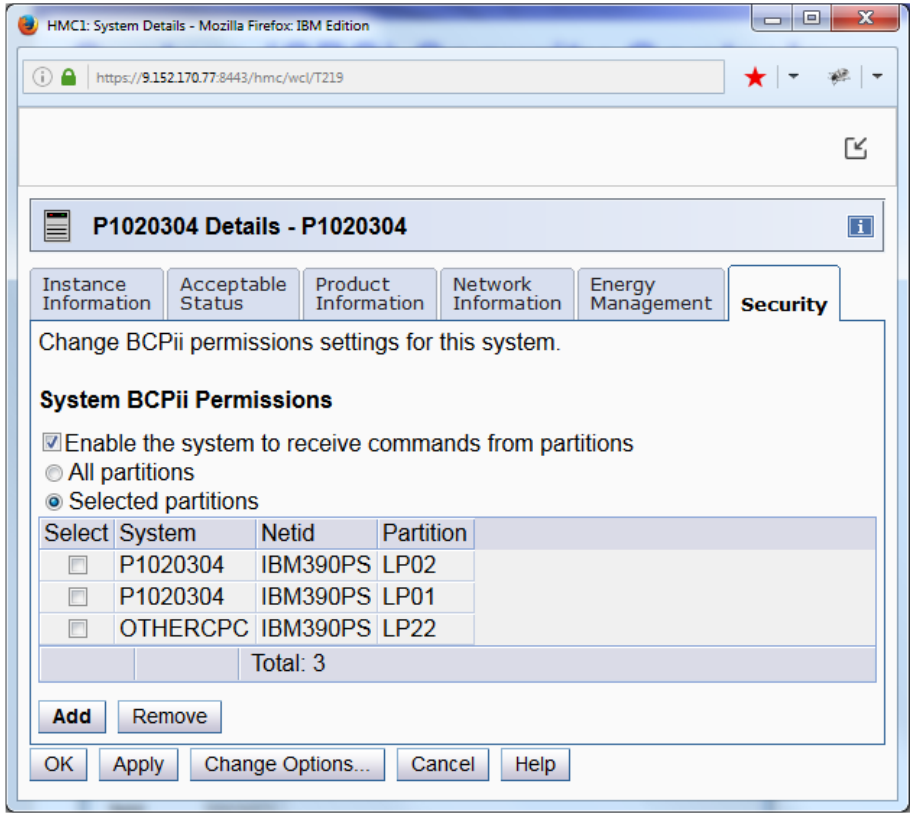
z14 BCPii CPC Security Controls – adding partitions



1. Can select from the set of known partitions
   - Known partitions are from the point of view of the system/partition being configured
   - “Local” partitions based on the currently defined image profiles.
   - “Remote” partitions learned from managing HMCs.

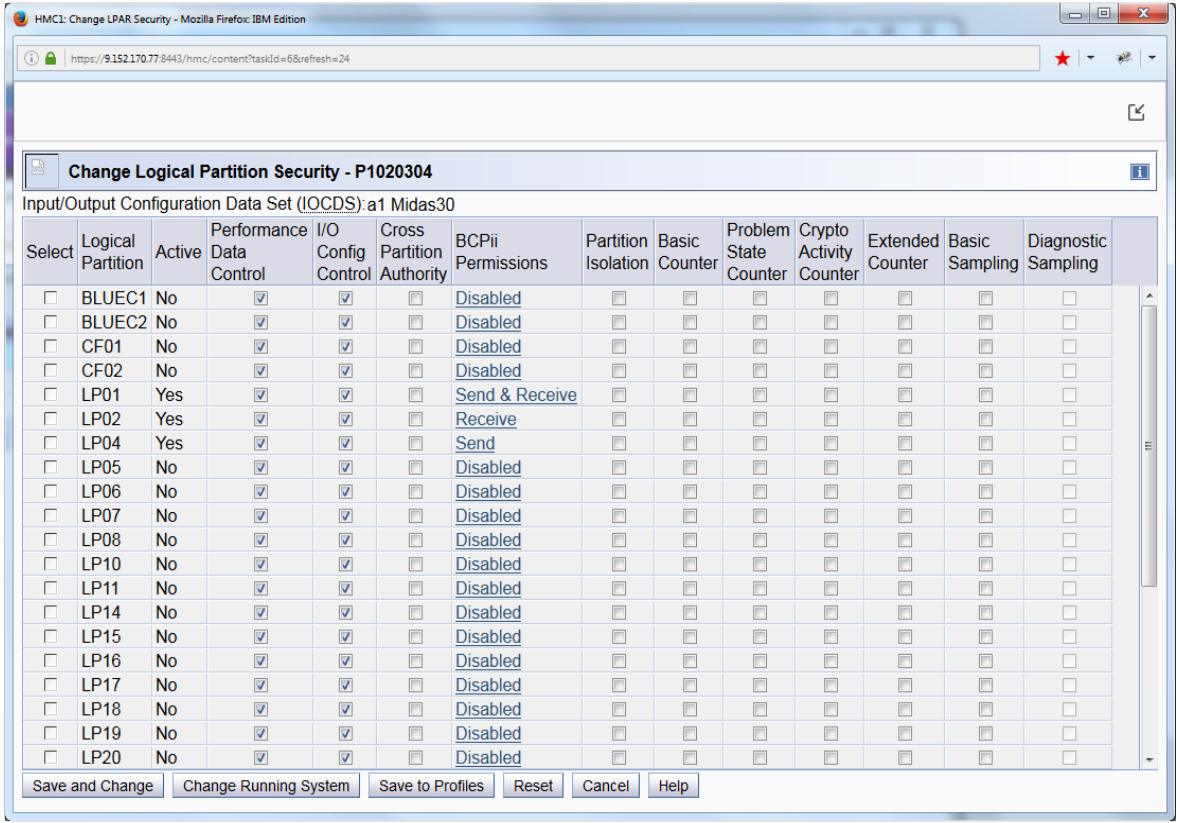
2. Can enter a partition manually for cases like pre-planning.

z14 BCPii CPC Security Controls – adding partitions



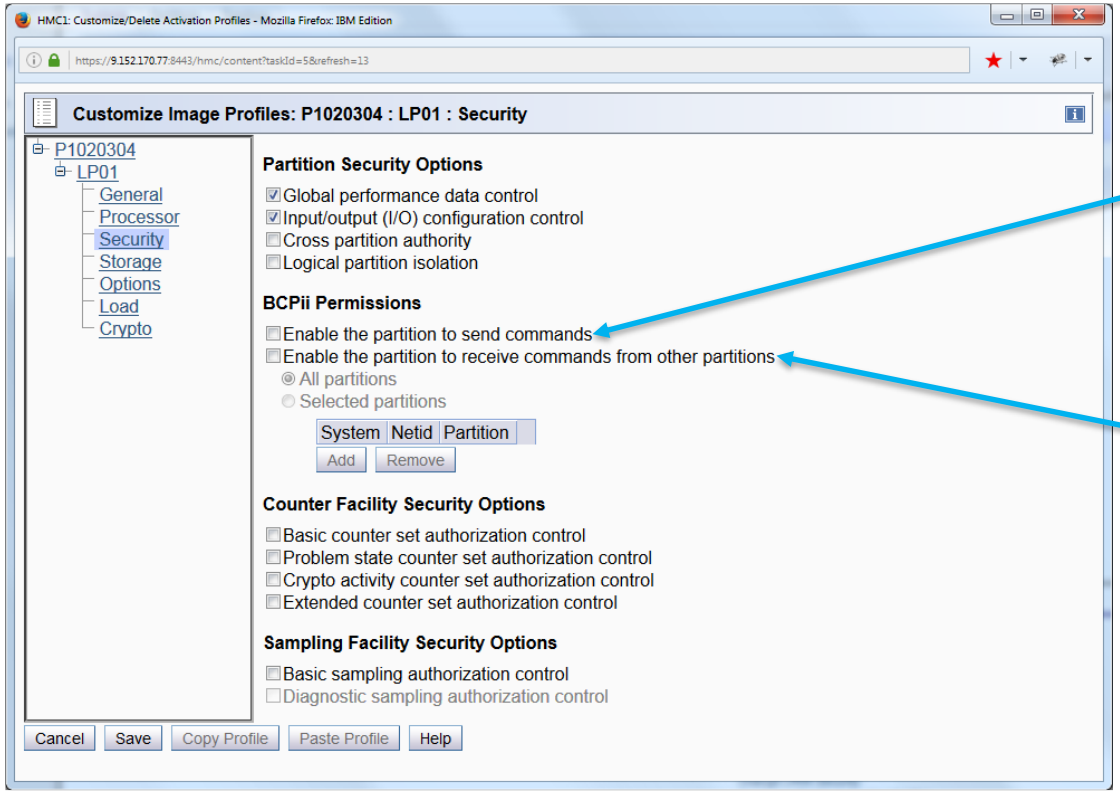
- Can also allow only selected partitions.
- Check the check box only to remove a CPC/Image from the list.

z14 BCPii LPAR Security Controls – Change LPAR Security Task



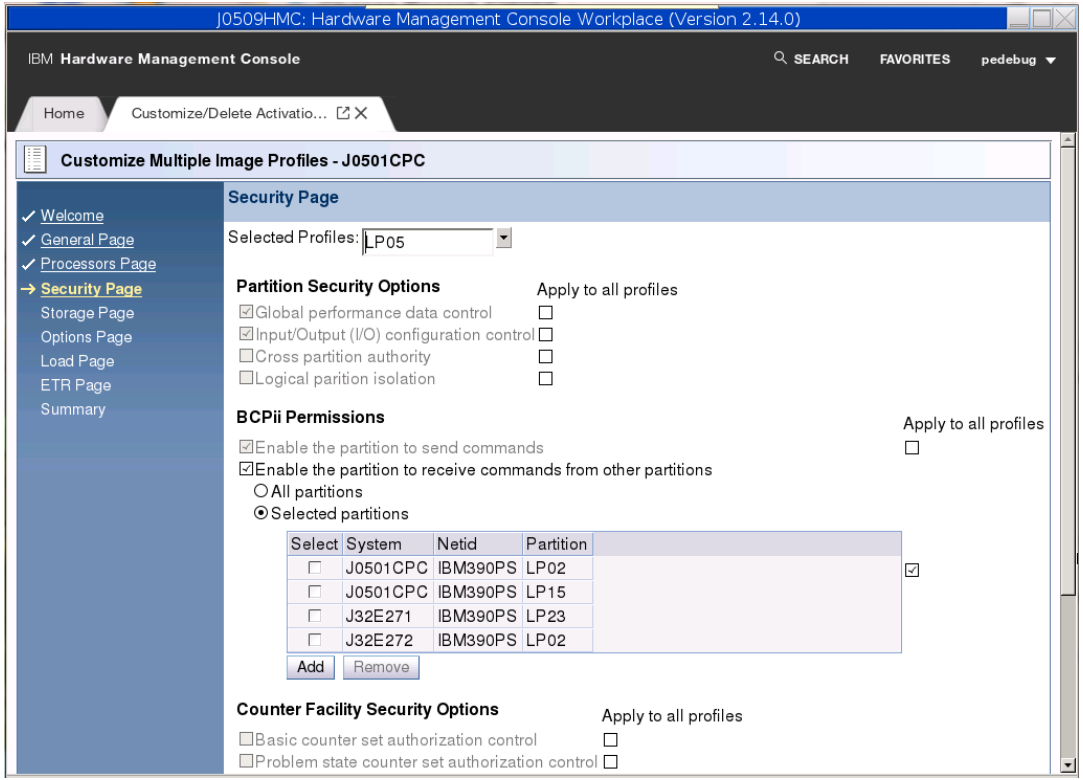
- Used to view/alter LPAR BCPii security controls.
- Available on HMC and SE. HMC only supports z14 and higher
- Click “BCPii Permissions” link to view/alter BCPii security control setting.

z14 BCPii LPAR Security Controls – Image Profile Security Controls



- Allows BCPii to be enabled on this image and to be able to potentially access other entities.
- Allows BCPii requests to target this partition from all or a select set of local and remote partitions.

z14 BCPii LPAR Security Controls – Multiple Image Profile Security Controls



- Can change multiple image activation profiles at the same time to easily configure the same BCPii permission settings.

z14 BCPii LPAR Security Controls – z/OS Support

- z/OS Migration from z/OS V2R2 and z/OS V2R1 to z/OS V2R3 publication updated
  - See “Enable BCPii communications on the support element” section for more details on the procedure to upgrade your BCPii security on the new z14.
- z/OS BCPii APAR OA53580
  - Update publication *z/OS MVS Programming: Callable Services for High-Level Languages* to explain the new security controls
  - To fix possible vague error return codes when some obscure security operations are performed.
  - This APAR available on V2R1 and higher.

BCPii Performance Enhancements

BCPii Performance

- BCPii performance has been an issue for some customers and applications for years
  - Particularly for HWIQUERY and HWILIST operations
    - Especially if querying activation profiles
  - Accentuated if many requests are performed by the application

BCPii Performance Improvements

- BCPii performance improvements when targeting a post-z13 GA2 machine
  - z13 - BCPii Performance & Request Enhancements
    - **MCL P00339.090 (Bundle S09)**
  - z14 and higher – in the base microcode
- Significant performance improvements across the board
- z/OS BCPii also leverages hardware update to improve activation profile HWIQUERY and HWILIST performance
  - In base V2R3
  - APAR OA51493 available on V2R2

BCPii Absolute Capping Support

Absolute capping “base” support

- Allows to limit an LPAR’s maximum usage of any CPU type to a specific amount of hardware processor capacity
  - Requires target system to be at **zEC12 GA2** and higher
  - Limits only “not dedicated” processors
- New z/OS BCPii support targeting both live image and image activation profile connections for the various engine types available on CPC
  - An LPAR’s Absolute Capping enablement/disablement to be both queried and set for every processor type (GPP, IFL, IIP, etc..)
  - An LPAR’s Absolute Capping value to the hundredth's of a processor
- Available on z/OS V2R2 via APAR OA49720

Absolute capping “group” support

- Allows to limit a group of LPARs of any CPU type to a specific amount of hardware processor capacity (a particular LPAR group)
  - Requires target system to be at **z13 GA2** and higher
  - Limits only the “not dedicated” processors
- New z/OS BCPii support targeting live image connections for the various engine types available on CPC
  - An LPAR group’s Absolute Capping enablement/disablement to be both queried and set for every processor type (GPP, IFL, IIP, etc..)
  - An LPAR group’s Absolute Capping value to the hundredth's of a processor
- Available on z/OS V2R3 and higher

New more flexible HWICM2 service

Current HWICMD service

- Current HWICMD allows the authorized z/OS application to perform major actions against a CPC or image
  - Load, Operating System commands, Activate, Deactivate, etc..
- Parameters are structured as the following:

```
CALL HWICMD  (
  ReturnCode
 ,ConnectToken
  ,CmdType
 ,CmdParm_Ptr
  ,DiagArea)
```

New HWICMD2 service

- CmdParm_Ptr points to a fixed-size data structure mapping which is unique for the command desired
  - This allows each possible BCPii command to have a customized “parameter list”, where each command-specific parm is in this unique data structure
  - e.g. If CmdType = HWI_CMD_LOAD, the mapping pointed by CmdParm_Ptr would look like this:

```
typedef char  HWI_LOADADDR_TYPE[4];
typedef struct {
  HWI_LOADADDR_TYPE LoadAddr;
  HWI_LOADPARM_TYPE LoadParm;
  HWI_FORCE_TYPE    ForceType;
} HWI_CMD_LOADORDUMP_PARM_TYPE;
```

New HWICMD2 service

- What happens when BCPii needs to support a 5-digit load address?
  - It can't use this parameter list

```
typedef struct {
  HWI_LOADADDR_TYPE LoadAddr;
  HWI_LOADPARM_TYPE LoadParm;   <- 5th character first char in field
  HWI_FORCE_TYPE    ForceType;
} HWI_CMD_LOADORDUMP_PARM_TYPE;
```

New HWICMD2 service

- HWICMD2 solves this by having the concept of versioned parameter lists
  - New Version number parameter passed into service
  - Version 2 of the parmlist for Load, SCSI Load and SCSI Dump commands will have an extra byte to allow for 5-char IPL device addresses.
  - Eliminates the need for any foreseeable time int the future where the command service will need to be changed again.
  - Available in V2R2 via APAR OA51250

New HWICMD2 service

- The syntax of the new service in non-REXX will be as follows:
  - `CALL HWICMD2(`
  - `ReturnCode,`
  - `ConnectToken,`
  - `CmdType,`
  - `CmdParm_Ptr,`
  - `CmdParmVersion,`
  - `DiagArea);`
- CmdParmVersion refers specifies the version of the parameter list

New HWICMD2 service

- Comparison of mapping of version 1 and version 2 parameter list for Load:

```
/* Version 1 of the Load Command structure */
 typedef struct ??<
    HWI_LOADADDR_TYPE LoadAddr; <- 4-char length
    HWI_LOADPARM_TYPE LoadParm;
    HWI_FORCE_TYPE    ForceType;
 ??> HWI_CMD_LOADORDUMP_PARM_TYPE;

/* Version 2 of the Load Command structure */
typedef struct ??<
    HWI_LOADADDR5_TYPE LoadAddr; <- 5-char length
    HWI_LOADPARM_TYPE LoadParm;
    HWI_FORCE_TYPE    ForceType;
 ??> HWI_CMD2_LOADORDUMP_PARM_TYPE;
```

New HWICMD2 service - REXX

- The BCPii REXX interface for the command service will remain unchanged.
- BCPii will invoke HWICMD2 at the highest cmd version level, regardless if REXX program chooses Hwicmd or Hwicmd2
- The syntax of the new service in REXX remains the same:

```
CmdType = HWI_CMD_OSCmd
CmdParm.PriorityType = Hwi_Cmd_Priority
CmdParm.OSCmdString = 'D GRS'

address bcpii "hwicmd ",          ← or hwicmd2
              "ReturnCode ",
              "ConnectToken ",
              "CmdType ",
              "CmdParm. ",

              "DiagArea."
```

Support for larger data returned from SE

BCPii Constraint Relief

- Currently, some larger sized attributes can sometimes exhaust BCPii's max buffer allowed allowed
  - BCPii limits the maximum number of data chunks that can come back from the SE
  - Verbose attributes (especially ones using XML) such as HWI_EC_MCL_INFO most susceptible to reaching the arbitrary BCPii limit.
- z/OS BCPii will now handle virtually any sized data returning from SE
- **No application changes required**
- Available on V2R2 via APAR OA53268.

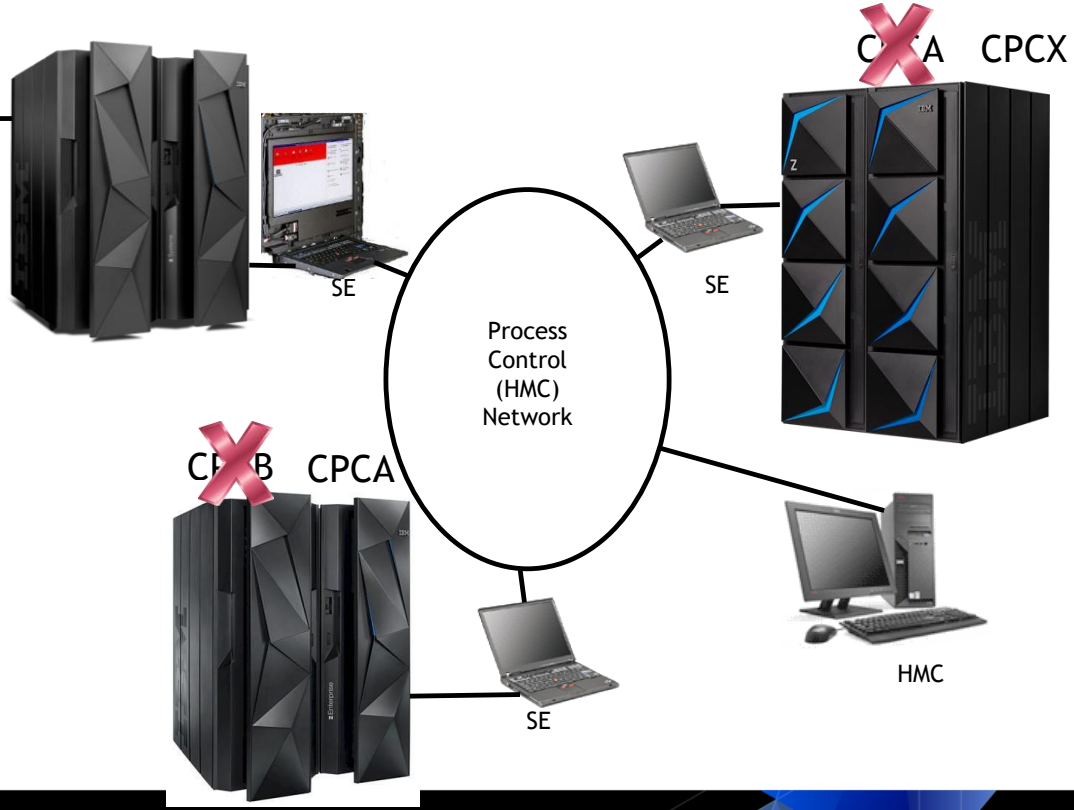
Support for Dynamic CPC name change

Dynamic CPC Name Change

- Today, z/OS BCPii does not react properly when an installation dynamically changes the name of their CPC and reboots the SE to make the change effective.
- BCPii has no easy way of knowing that a request targeting one CPC physical machine could suddenly now be arriving at another CPC physical machine.

Dynamic CPC Name Change

```
HWICONN CPCA
HWIQUERY CPCA
HWIQUERY CPCA
```



Dynamic CPC Name Change

- BCPii provides support for changing the name of a CPC with ACTIVE images.
- When a CPC name change is detected, BCPii takes the following actions:
  - Invalidates outstanding connections to the affected CPC.
  - Issues an ENF68 informing interested parties of the name change ((hardware event HWIENF68_HWEVENT_NAMECHG)
  - Reconnects to the local CPC (if the local CPC name is changed)
- Applications targeting the CPC using the old name will get a return code indicating the connection is no longer valid (e.g, HWI_CONNECT_TOKEN_INV or HWI_TARGET_CPC_CHANGED).

Dynamic CPC Name Change

- HWI_TARGET_CPC_CHANGED return code
  - **Meaning:** The CPC name represented by the specified token is valid, but does not represent the same physical machine targeted by the initial HWICONN call. All connections established prior to the name change can no longer be used.
  - **Action:** The application should cease from using this connect token. If the application intends to target the CPC using the name represented by the specified connect token, they must first re-connect to the CPC before issuing any BCPii service call.

New HWISET2 (multiple attribute set) service

BCPii Set service (HWISET) functionality

- One CPC, image, or activation profile attribute can be set at a time
  - Higher latency connection between z/OS and the SE means that there are short time lags between each HWISET call
  - Configuration likely to have temporarily incompatible values set for attributes
- Example:
  - Image A – defined capacity value (HWI_DEFCAP) = 10
  - Image B – defined capacity value = 5
  - Want to set Image A to 8 and Image B to 8.
  - Application issues
    - HWISET image A – HWI_DEFCAP = 8
    - HWISET image B – HWI_DEFCAP = 8
  - **There is a short time period of configuration inconsistency, where A is 8 and B is 5, when the end goal is 8 and 8.**

New HWISET2 (Multiple attribute set) service

- Multiple attributes can be set at the same time
  - Provides better adjustments to the hardware configuration
  - Shorter periods of configuration
- Example:
  - Image A – defined capacity value (HWI_DEFCAP) = 10
  - Image B – defined capacity value = 5
  - Want to set Image A to 8 and Image B to 8.
  - Application issues
    - HWISET2 with HWI_DEFCAP = 8 for image A and HWI_DEFCAP = 8 for image B
  - **Much shorter period of configuration inconsistency. When call returns to application, both values have been adjusted.**

HWISET2 particulars

- New HWISET2 service similar to the HWIQUERY interface for querying multiple attributes on a single call
- Additional benefit of having a “two-phase commit” like behavior where either all changes will be made or none of the changes will be made when targeting at z13 GA2 or higher machine.
- HWISET2 can target z13 GA1 and lower successfully.
  - If a failure occurs with one of the attributes, a return code of HWI_SET2_PARTIAL_UPDATE will be returned along with an indicator of the first failing attribute and the reason for the failure.
  - User will either have to manually back out the attributes already updated or attempt to rectify the partial update in another manner.

HWISET2 particulars (continued)

- All resources specified must be located on the same CPC.
  - Each attribute to be set specifies a ConnectToken representing that resource. The ConnectToken must either be:
    - Identical to the CPC ConnectToken on the HWISET2 parameter list
      - i.e. the CPC attribute is being set
    - A child connection of the CPC ConnectToken on the HWISET2 parameter list.
      - i.e. the ConnectToken is an image or activation profile on that CPC

HWISET2 – Syntax

- The syntax of the new service in non-REXX will be as follows:
  - `CALL HWISET2(`
  - `ReturnCode,`
  - `InConnectToken,`
  - `SetParm_Ptr,`
  - `NumOfAttributes,`
  - `DiagArea);`
  - `InConnectToken` must represent a BCPii CPC connection.
  - `SetParm_Ptr` points to an array of Set structures.
  - `NumofAttributes` specifies the # of elements in the array (up to 9 attributes can be set at one time

HWISET2 Data structure

- SetParm_Ptr points to a structure mapped by SetParm

```
/*-------------------------------------------------------------------*/
/* Mapping of the SetParm structure, whose address is supplied by    */
/* as the SetParm_Ptr on a call to HWISET2.                          */
/* An array of these structures can be supplied to set multiple      */
/* attributes.                                                       */
/*-------------------------------------------------------------------*/
typedef struct {
    HWI_CONNTOKEN_TYPE Set2_CToken;
    int Set2_SetType;
    char * Set2_SetValue_Ptr;
    int Set2_SetValueLen;
}HWI_SET2_SETPARM_TYPE;
```

HWISET2 – Syntax (REXX)

- The BCPii REXX interface for the SET2 service will use constant stem variable tails ala HWIQUERY to set the various set values required.

```
stemName.0 = 3
stemName.1.SET2_Ctoken = <image1ConnToken on CPC1>
stemName.1.SET2_SetType = HWI_DEFCAP
stemName.1.SET2_SetValue = 5
stemName.2.SET2_Ctoken = <image2ConnToken on CPC1>
stemName.2.SET2_SetType = HWI_DEFCAP
stemName.2.SET2_SetValue = 3
stemName.3.SET2_Ctoken = <imageActProfConnToken on CPC1>
stemName.3.SET2_SetType = HWI_DEFCAP
stemName.3.SET2_SetValue = 5
```

- Note that the connect tokens must all be targeted to the same CPC as specified on the InConnectToken parameter.

HWISET2 new return codes and abend codes

- New return codes:
  - HWI_SET2_SETPARM_INACCESSIBLE – setparm structure not addressable
  - HWI_SET2_NUMOFATTRIB_INV – specified a value <1 or >9
  - HWI_SET2_CONNECT_TOKEN_INV – connect token specified in the data structure either not associated with InConnectToken or bad value
  - HWI_SET2_PARTIAL_UPDATE – mentioned on previous slide
- New abend codes:
  - HWI_SET2_PRIM_ADDR – parms not in primary AS
  - HWI_SET2_NOT_ADDR – parms not addressable
  - HWI_SET2_BAD_PARM – number of parms bad

BCPii Enhancement in V2R2 (BCPii SMF Logging)

The need for SMF support in z/OS BCPii

- **Problem Statement / Need Addressed**
  - Whenever a BCPii application issues an API that modifies hardware resources, there are not sufficient audit trails to keep track of which application/user modified the resources (SHARE Top 50 requirement SSMVSE12018)
- **Solution**
  - BCPii now cuts SMF 106 records for **successful** HWISET and HWICMD API calls
- **Benefit / Value**
  - Sufficient audit information to know what resources were modified by BCPii applications

BCPii SMF Support – How to cut the records

- New SMF 106 record
  - IEASYSxx points to SMFPRMxx member
  - SMFPRMxx member
    - SYS(TYPE(106))
- 2 subtypes
  - Subtype 1 (HWISET) SYS(TYPE(106(1)))
    - Cuts detailed information about each successful HWISET call
  - Subtype 2 (HWICMD) SYS(TYPE(106(2)))
    - Cuts detailed information about each successful HWICMD call

BCPii SMF support – How to read the records

- Mapped by BCPii SMF Type 106 mapping macro
  - SYS1.MACLIB(HWISMF6A)
- Supplied sample formatting JCL
  - HWI6AFMT
    - Copies BCPii SMF Type 106 records from a data set or logstream to a temporary dataset
    - Using the DFSORT-provided ICETOOL:
      - Sorts the type 106, subtype 1 and subtype 2 records
      - Produces a summary report for the type 106 records
      - Produces a detailed report for type 106 for subtype 1 and subtype 2
  - HWIRPTMP
    - SMF type 106 JCL variable map as input to the DFSORT-provided ICETOOL

What is actually in the SMF Data

- What is actually cut by BCPii in the SMF Record Type 106?
  - Connection Type of the HWISET or HWICMD request
    - CPC, Image, Reset activation profile, Image activation profile, or Load activation profile
  - CPC Name
  - Request parameter
    - Either the image name or activation profile name specified by the requester
  - ASID
  - Jobname
  - User ID

What specifically is in SMF records for HWISET?

- Detailed information in Subtype 1 (HWISET):
  - SetType
    - Resource that was modified
  - Set Type Value Length
    - Length of the value being set
  - Set Parameter
    - The actual value being set

What specifically is in the SMF records for HWICMD?

- Detailed information in Subtype 2 (HWICMD):
  - Command Type
    - Command that was issued
  - Command parameter list passed to BCPii
  - Optional XML data sent on request

Example output

- Sample report output:

SUBTYPE 1 RECORDS SUMMARY REPORT 02/03/15 10:18:52

```
CPC Name  IBM390PS.H87

   LEN      SEG   FLG    RTY        TME          DTE    SID    WID       STP
------   ------   ---   ----   --------   ----------   ----   ----   ------
   397        0   DE     106   10:45:05   2015/02/03   BCPJ   JES2         1

VERSION   PROD NAME   MVS PROD   SYSTEM NAME   CONN TYPE   CPC
-------   ---------   --------   -----------   ---------   -----------------
01        BCPII       SP7.2.2    BCPJ                  2   IBM390PS.H87

RTN                  ASD   JOBNAME    USER        SET TYPE     SET PARM LEN
----------------   ------   --------   --------   -----------   ------------
LP8              2D     HWASEY04   SWARREN           116              4

TDA
--------------------------------
000003E700000000000000000000000
```

Usage & Invocation

- Sample report output:

SUBTYPE 2 RECORDS SUMMARY REPORT 02/13/15 14:47:02 - 1

```
CPC Name  IBM390PS.H87

   LEN      SEG    FLG    RTY        TME          DTE    SID    WID       STP
------   ------   ---   ----   --------   ----------   ----   ----   ------
  3955        0    DE     106   14:34:05   2015/02/13   BCPJ   JES2         2

 VERSION   PROD NAME   MVS PROD   SYSTEM NAME   CONN TYPE   CPC
 -------   ---------   --------   -----------   ---------   ----------------
 01        BCPII       SP7.2.2    BCPJ                  1   IBM390PS.H87

RTN                  ASID   JOBNAME    USER       COMMAND TYPE
----------------   ------   --------   --------   ------------
                       47   HWAECX16   IBMUSER              14
CMD XML or IPLToken
--------------------------------
0000000123DE04100000005C00000000
```

Sample output

- Sample report output:

HWI_CMD_TEMPCAP SMF RECORDS REPORT 02/13/15
14:47:02 - 1

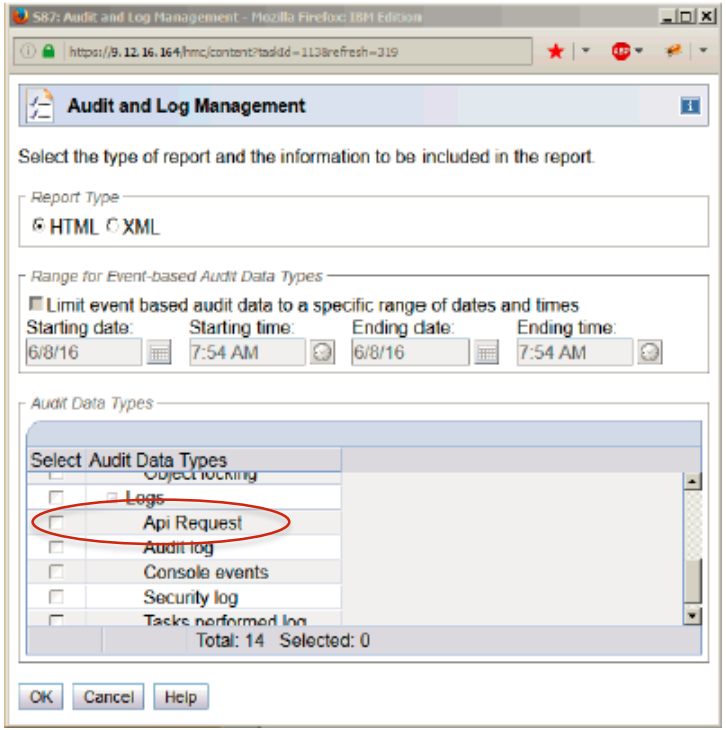
```
   LEN      SEG   FLG    RTY       TME          DTE    SID    WID       STP
------   ------   ---   ----   --------   ----------   ----   ----   ------   -
  3955        0   DE     106   14:34:05   2015/02/13   BCPJ   JES2        2

        CTY   TEMPCAPTYPE   CAPXMLPTR    CAPXMLSIZE   XMLData
----------   -----------   ---------   -----------   ----------------
         14             1   23DE0410            5C   <add><recordid>0
```

Enabling BCPii SMF recording

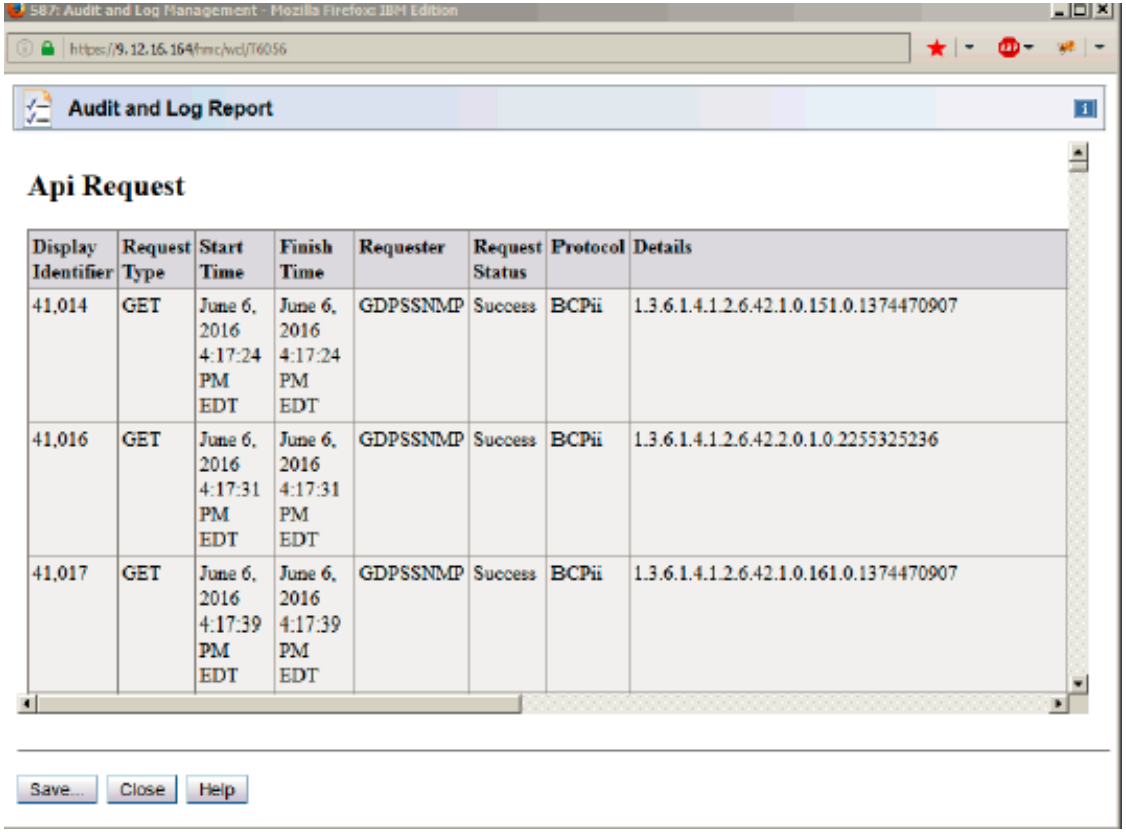
- To activate BCPii SMF recording:
  - Parmlib method
    - Add the necessary statements to the SMFPRMxx parmlib member for SMF Type 106
      - SYS(TYPE(106))
      - SYS(TYPE(106(1)))
      - SYS(TYPE(106(2)))
      - SYS(TYPE(106(1:2)))
    - Issue the SET SMF=xx command to activate the parmlib changes
  - On the fly method
    - SETSMF command

Debugging Programming Errors



- BCPii API request history logging available on z13 GA2
  - MCL P00339.185 (Bundle S20)
- Shows BCPii requests coming into the SE
- API Request Log Option
  - Single Object Operations → Support Element Management → Audit and Log Management task (in the security section)
  - Select Api Request checkbox

Debugging Programming Errors



Display Identifier	Request Type	Start Time	Finish Time	Requester	Request Status	Protocol	Details
41,014	GET	June 6, 2016 4:17:24 PM EDT	June 6, 2016 4:17:24 PM EDT	GDPSSNMP	Success	BCPii	1.3.6.1.4.1.2.6.42.1.0.151.0.1374470907
41,016	GET	June 6, 2016 4:17:31 PM EDT	June 6, 2016 4:17:31 PM EDT	GDPSSNMP	Success	BCPii	1.3.6.1.4.1.2.6.42.2.0.1.0.2255325236
41,017	GET	June 6, 2016 4:17:39 PM EDT	June 6, 2016 4:17:39 PM EDT	GDPSSNMP	Success	BCPii	1.3.6.1.4.1.2.6.42.1.0.161.0.1374470907

BCPii Reference Material

BCPii Reference Materials

- **z/OS 2.4 MVS Programming: Callable Services for High-Level Languages**
  - Primary BCPii documentation including installation instructions and BCPii API documentation (including BCPii REXX support)
- **z/OS 2.4 MVS System Commands:**
  - START HWISTART and STOP HWIBCPII commands.
- **z/OS 2.4 MVS Diagnosis: Tools and Service Aids:**
  - BCPii's CTRACE documentation.
- **z/OS 2.4 MVS Programming: Authorized Assembler Services Reference, Volume 2 (EDT-IXG)**
  - BCPii's ENF68 documentation.
- **z/OS 2.4 MVS System Management Facilities (SMF)**
  - BCPii's primary SMF documentation on SMF Type 106 records
- **z/OS 2.4 MVS Initialization and Tuning Reference**
  - Miscellaneous documentation
- **z/OS 2.4 MVS System Codes**
  - BCPii abend '042'x documentation

Yet More BCPii Information!

- Cheryl Watson's Tuning Letter, 2013, No.2
  - Focus: Exploiting z/OS, Part 3 - BCPii
- IBM Redbooks (http://www.redbooks.ibm.com)
  - System z Parallel Sysplex Best Practices

Yet More BCPii Information (Older but useful info)!

- z/OS Hot Topics
  - August 2013: Quick and Easy: BCPii (pg. 63)
  - August 2012: Seeing BCPii with new eyes (pg. 7)
  - August 2009: The application doesn't fall far from the tree (*BCPii: Control your HMC and support element directly from z/OS apps)*

Quick and Easy: BCPii!

BY STEVE WARREN AND RITA BEISEL

It's time to check out the Base Control Program internal interface (BCPii) in z/OS Version 2 Release 1 (V2R1). The improvements in BCPii function might be the quick and easy recipe to help you start using this base function of the z/OS operating system. If you are already using BCPii, you can now use it more efficiently than ever.

BCPii at your service

In z/OS V2R1, BCPii supports applications written in the REXX programming language, known for its ease of use. BCPii also minimized the traffic to the support element (SE). Less traffic to the SE might equal improved performance for you. Let's first take a step back and look at BCPii

BCPii is a cool way to access System z hardware controls from any z/OS authorized application running in any address space. For example, you might want to:

- Find out what is going on with the hardware
- Perform powerful tasks like re-IPL or load an LPAR
- Receive notification when certain hardware events occur.

Do you want to do all these things from the convenience of your z/OS application? If so, BCPii is at your service! It's not necessary to install a suite of products or complete a complicated install process to start using it.

Ready for REXX?

Before z/OS V2R1, the BCPii APIs supported applications using either the C or assembler programming languages. Over the years, there has been a growing and vocal demand for REXX programming language support in BCPii API.

We listened and delivered

In z/OS V2R1, the BCPii support for REXX and a much simpler programming model than either the C or assembler programming languages, you can get applications up and running quickly and easily.

BCPii APIs support applications using REXX in the z/OS System REXX, TSO/E REXX, and independent software vendor (ISV) provided REXX programming environments. Not only does writing with the REXX programming language allow you to develop BCPii applications in record time, but also maintains your investment in your existing BCPii applications written in C or assembler. These REXX applications can work right along side them.

Sample BCPii REXX exec

Here is a simple BCPii REXX exec that lists all the interconnected processors in your Hardware Management Console (HMC) network

Notice the intuitive programming style. Just specify the list type and voilà, BCPii returns the data in a stem variable. The zero element of the stem variable contains the number of items returned and the 1 to n elements contain the actual names of the processors connected to the system. This is only an example, but the other BCPii API calls are just as intuitive and easy to use.

```
LISTTYPE = HWI_LIST_CPCS

ADDRESS BCPII "HWILIST
        RETURNCODE
        CONNECTTOKEN
        LISTTYPE
        CPCLIST.
        DIAGAREA."

IF RC <> 0 | RETURNCODE <> 0 THEN
 /* IF THE REXX RC IS NOT GOOD OR THE BCPII RETURN
 CODE IS NOT GOOD, HAVE ERROR HANDLING CODE HERE */
ELSE
 DO
  SAY 'NUMBER OF CPCS RETURNED = ' CPCLIST.0
  /* WRITE THE LIST OF CPCS RETURNED. THE .0
  ELEMENT CONTAINS THE NUMBER OF ITEMS RETURNED */
  DO I = 1 TO CPCLIST.0
   SAY 'CPC '|| I ' = ' CPCLIST.I
  END
 END
```

Figure 1. Sample BCPii REXX exec

IBM Systems Worldwide Client Experience Centers

IBM Systems Worldwide Client Experience Centers *maximize IBM Systems competitive advantage in the Cloud and Cognitive era by providing access to world class technical experts and infrastructure services to assist Clients with the transformation of their IT implementations..*

9 Worldwide Locations (* also Infrastructure Hubs):

Austin TX , *Poughkeepsie NY, Rochester MN, Tucson AZ, *Beijing CHINA, Boeblingen GERMANY, Guadalajara MEXICO,*Montpellier FRANCE, Tokyo JAPAN

Client Experience	*Architecture & Design*	*Infrastructure Solutions*	*Content*
Tailored, in-depth technology Innovation Exchange Events Relationship building Demonstrations Meetups Solution workshops Remote options	Advise clients, "Art of the Possible" Discovery & Design Workshops, Consulting, Showcases, Reference Architectures, Co-Creation of assets	Benchmarks, MVP & Proof of Technology "Test Drives" Demonstrations Infrastructure Services Certify ISV solutions Hosting Cloud Environment	Content Development IBM Redbooks Training Courses Video courses "Test Drives" Demonstrations
(Inbound & Outbound)	(Inbound & Outbound)	(Inbound to Centers)	

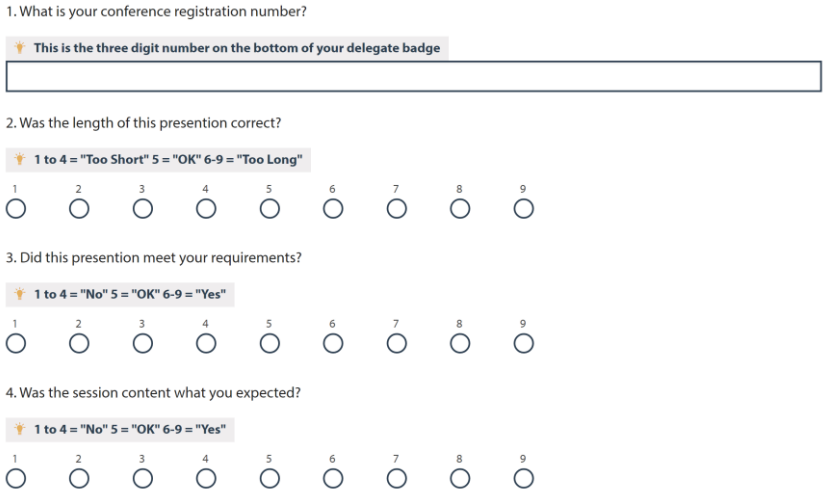
NEW: *Co-Creation Lab; CEC Cloud; RedHat Center of Competency*

For further information, please contact the Centers via email at: *ccenter@us.ibm.com*

Please submit your session feedback!

- Do it online at http://conferences.gse.org.uk/2019/feedback/bb

- This session is BB



Questions?