IBM Education Assistance for z/OS V2R1

Item: Enhancements
REXX Support
Getbulk Performance Improvement
Element/Component: z/OS BCPii

Material is current as of March 2013

Agenda

- Trademarks
- Presentation Objectives
- Overview – REXX Support
- Usage & Invocation – REXX Support
- Installation – REXX Support
- Overview – GetBulk
- Usage & Invocation – GetBulk
- Interactions & Dependencies - GetBulk
- Presentation Summary
- Appendix

Trademarks

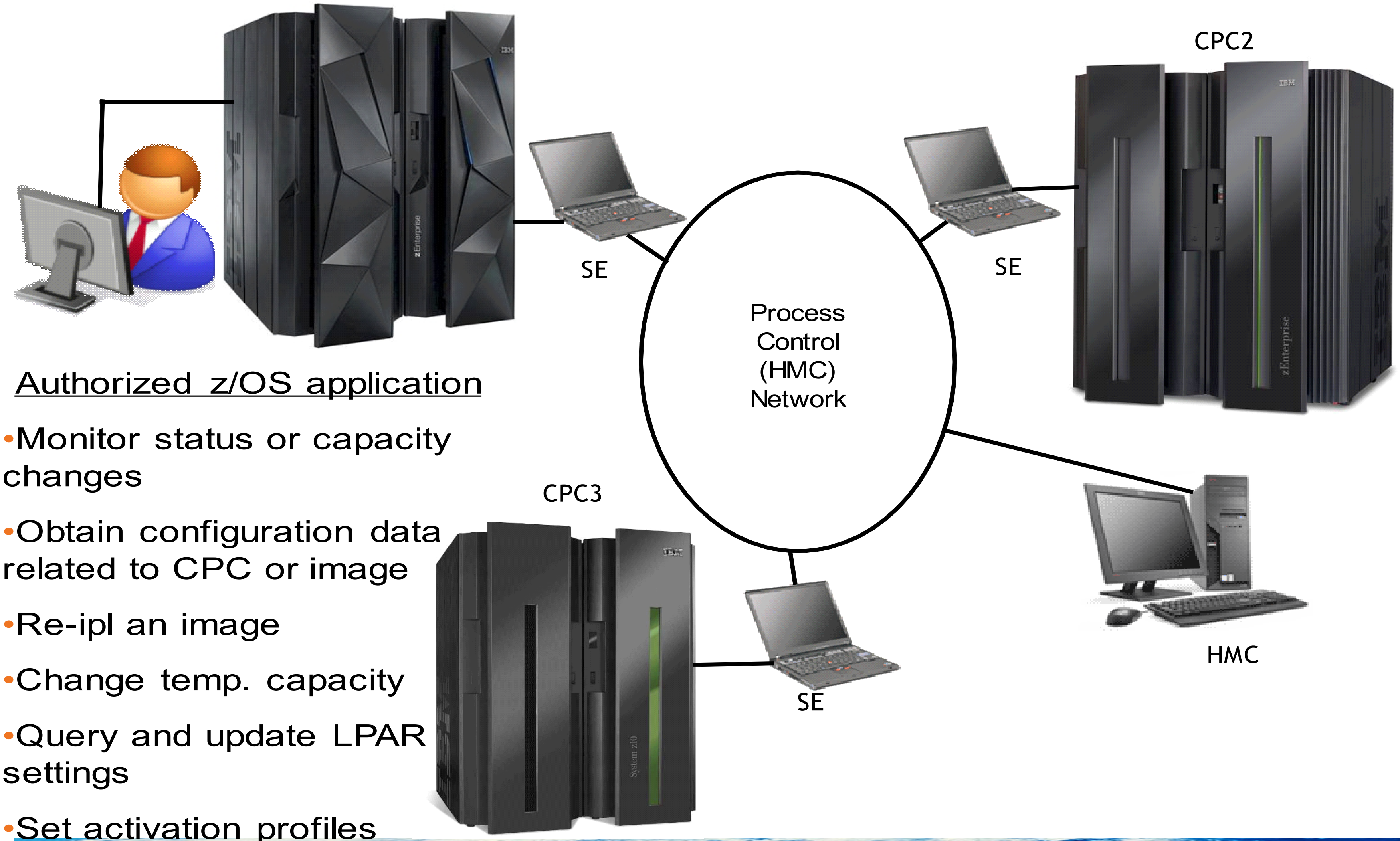
- See url http://www.ibm.com/legal/copytrade.shtml for a list of trademarks.

Presentation Objectives

- Quick overview of z/OS BCPii

- Overview of new functionality introduced in V2R1
  - z/OS BCPii support of REXX
  - z/OS BCPii performance enhancements in HWIQUERY and HWILIST

What is z/OS 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

Overview – BCPii Support for REXX

- Problem Statement / Need Addressed
  - z/OS BCPii supports only C and Assembler callers today
  - Several customer requirements have been written and numerous other requests have been given to IBM to ask for BCPii support of the REXX programming language
- Solution
  - BCPii is providing comprehensive REXX support
    - System REXX
    - TSO REXX
    - ISV-provided REXX environments
- Benefit / Value
  - BCPii application programming just got even simpler!

Usage & Invocation (REXX Support)

- *BCPii is providing a new REXX host command environment for System REXX, TSO REXX and ISV REXX environments. (address bcpii)*
  - Same authorization requirements as current BCPii applications
  - Simpler programming model than in C or Assembler
    - Programming style is intuitive for REXX programmer
    - Use of stem variables for variable number of items output
  - Parameter lists for BCPii services using REXX are simpler than C or Assembler parameter lists
    - Differences documented in the publications
  - BCPii REXX programs compatible with the different REXX environments*
  - Built-in RC return will indicate if BCPii processed the host command successfully. If zero, the BCPii return code should be consulted.

* For the common services supported by BCPii in the different environments

Usage & Invocation (REXX Support)

- *z/OS BCPii APIs supported*

Services	System REXX	TSO REXX	ISV REXX
HWICONN	x	x	x
HWIDISC	x	x	x
HWILIST	x	x	x
HWIQUERY	x	x	x
HWISET	x	x	x
HWIEVENT	x		
HWICMD	x		

Usage & Invocation (REXX Support)

- *z/OS BCPii System REXX support*
  - Full support of BCPii API suite
    - Command and event require non-REXX event exit and a program to wait on an ECB based on event activity
  - Ability for REXX BCPii applications to work with other C or Assembler BCPii applications
    - The Connect Token can be passed to and from the REXX exec and the other compiled BCPii applications.
  - Address space affinity connections
    - When AXREXX macro invoker's address space terminates, BCPii will implicitly disconnect all connections
  - TSO=YES and TSO=NO environments supported
    - TSO=YES allows REXX to interpret the IBM-supplied REXX include file
    - TSO=NO requires the IBM-supplied include file to be copied into the exec
  - TIMELIMIT keyword can be used to throttle BCPii exec execution time
    - The default 30 seconds value may need to be adjusted

Usage & Invocation (REXX Support)

- *z/OS BCPii System REXX support (continued)*
  - Two methods of execution of BCPii REXX execs
    - Code an assembler program to invoke the AXREXX macro
      - Specify the name of BCPii REXX exec and any of the myriad of AXREXX options
    - New BCPii helper program HWIREXX
      - IBM-supplied helper program shipped in SYS1.LINKLIB that authorized users can invoke to launch their System REXX execs
      - Simple REXX execs can be invoked directly without the need to code the AXREXX assembler macro
      - A set of input parameters allows minor customization
        - Samplib JCL member HWIXMRJL provides list of parameters HWIREXX takes as input (supports a subset of AXREXX options)

Usage & Invocation (REXX Support)

- *z/OS BCPii TSO REXX Support*
  - Support of all BCPii APIs except HWIEVENT and HWICMD
  - Task affinity connections
    - All connections created by the REXX exec are automatically cleaned-up by BCPii when exec completes
    - Connections cannot be shared with other BCPii applications or REXX execs
  - Same SAF authorization requirements as other BCPii applications
  - Setup required for TSO REXX support
    - IKJTSOxx parmlib member must have the following update:
      - `AUTHTSF NAMES(HWIC1TRX)`

Usage & Invocation (REXX Support)

- *z/OS BCPii ISV REXX Support*
  - Support of all BCPii APIs except HWIEVENT and HWICMD
  - Task affinity connections
    - All connections created by the REXX exec are automatically cleaned-up by BCPii when exec completes
    - Connections cannot be shared with other BCPii applications or REXX execs
  - Same program and SAF authorization requirements as other BCPii applications
    - Must be invoked from an authorized address space
  - To get the “bcpii” host command environment, the REXX exec must issue the following statement:
    - `rc = hwihost("ON")`

Usage & Invocation (REXX Support)

- *Example of z/OS BCPii REXX exec in action:*

```
ListType = HWI_LIST_CPCS

address bcpii "hwilist
               ReturnCode
               ConnectToken
               ListType
               CPCList.
               DiagArea."

If rc <> 0 | ReturnCode <> 0 Then
  /* Error handling code here */
Else
  Do
    Say 'Number of CPCs returned = ' CPCList.0
    /* Write the list of CPCs returned.                        */
    Do i = 1 to CPCList.0
      say 'CPC '|| i ' = ' CPCList.i
    End
  End
```

Usage & Invocation (REXX Support)

- *Example of z/OS BCPii REXX exec in action (continued):*

```
QueryParm.0 = 2
QueryParm.1.ATTRIBUTEIDENTIFIER = HWI_MMODEL
QueryParm.2.ATTRIBUTEIDENTIFIER = HWI_SNAADDR

address bcpii "hwiquery
               ReturnCode
               ConnectToken
               QueryParm.
               DiagArea."

If rc <> 0 | ReturnCode <> 0 Then
    /* Error handling code here */
Else
    Do
      say 'MModel      = ' QueryParm.1.ATTRIBUTEVALUE
      say 'SNAAddr     = ' QueryParm.2.ATTRIBUTEVALUE
    End
```

Usage & Invocation (REXX Support)

- *BCPii REXX Samples provided in Samplib:*
  - *HWIXMRS1*
    - Sample REXX program that uses the most commonly used services
      - HWIXMRJL – Sample JCL invoking HWIXMRS1 using the HWIREXX helper program
  - *HWIXMRS2*
    - Sample REXX program that additionally uses HWIEVENT and HWICMD services.
      - Requires assembler sample program HWIXMRA1
        - Sets up common storage accessible to both ENF Exit and waiting program.
        - Provides example of using the AXREXX macro to invoke the BCPii REXX exec
      - Uses existing Metal C BCPii Sample ENF Exit HWIXMCX1

Installation (REXX Support)

- For System REXX enablement of z/OS BCPii host command environment:
  - Verify current AXRxx parmlib member is configured correctly. (e.g. (parameters such as REXXLIB and AXRUSER)
- For TSO/E enablement of z/OS BCPii host command environment:
  - Current IKJTSOxx parmlib member must be updated with:
    - `AUTHTSF NAMES(HWIC1TRX)`
- Potential security product configuration changes
  - Access lists for existing security profiles will need to be updated to add all new users that now wish to use z/OS BCPii in the System REXX, ISV REXX, and TSO REXX environments.

Overview – BCPii GetBulk Performance Improvements

- Problem Statement / Need Addressed
  - BCPii retrieval requests can be slow, especially when multiple attributes are retrieved. The connection between z/OS and the SE that z/OS BCPii uses (internal proprietary interface) has both high latency and low bandwidth. Single simple query requests can average between 0.3 and 0.5 seconds on the wall clock due to various factors.
  - Example (Today):
    - HWILIST ListImage: requires n+1 HwmcaGet* requests to the SE where n = number of LPARs on the CPC
    - HWIQUERY specifying 6 attributes: requires 6 HwmcaGet requests to the SE

* - Part of the System z API

Overview – BCPii GetBulk Performance Improvements

- Solution
  - Use HwmcaGetBulk* to package requests on one request to the SE and other improved algorithms in retrieving data
  - Example (Using z/OS 2.1):
    - HWILIST ListImage: requires 1 HwmcaGetBulk request to the SE (2 the first time listing child objects for a CPC)
    - HWIQUERY specifying 6 attributes: requires 1 HwmcaGetBulk request
- Benefit
  - Significant performance improvements for certain types of z/OS BCPii requests

* - Part of the System z API

Overview – BCPii GetBulk Performance Improvements

- Sample Scenario:
  - z/OS BCPii application wishes to list all image names on a CPC (60 partitions) or query multiple attributes regarding a particular LPAR (image)
  - HWILIST (HWI_LIST_IMAGES):
    - Today: 61 HwmcaGet calls = Approx 8.5 seconds
    - New: 1 HwmcaGet call + 1 HwmcaGetBulk call = Approx 0.65 seconds on 1st HWILIST and 0.27 seconds on subsequent HWILIST calls.
    - **13x up to 31x improvement**
  - HWIQUERY (HWI_OPERSTAT, HWI_OSNAME, HWI_OSTYPE, HWI_OSLEVEL, HWI_SYSPLEX, HWI_PARTITIONID)
    - Today: 6 HwmcaGet calls at approx 0.4 secs each = Approx 2.43 seconds
    - New: 1 HwmcaGetBulk call = 0.52 secs
    - **4.7x improvement**

Note: Performance benefits will vary depending on the attribute being queried, number of attributes being queried simultaneously, the load of the SE, load of the z/OS image and the hardware configuration.

Usage & Invocation (GetBulk)

- *No changes to z/OS BCPii configuration or applications required.*
  - Note: If HWIQUERY requests were called separately for each attribute in the past, a modification to the application to combine the attribute queries into a single HWIQUERY call can improve performance significantly

Interactions & Dependencies (GetBulk)

- Software Dependencies
  - None
- Hardware Dependencies
  - Must target z9 (running at latest microcode level) or higher to take advantage of performance improvements
- Exploiters
  - None

Presentation Summary

- A quick glimpse of z/OS BCPii functionality
- Overview of new functionality introduced in V2R1
  - z/OS BCPii support of REXX
  - z/OS BCPii performance enhancements in HWIQUERY and HWILIST

Appendix

- Publication References:
  - SA22-7613 - z/OS MVS Programming: Callable Services for High Level Languages
    - Contains all of the z/OS BCPii REXX documentation and a few new return codes in support of the GetBulk performance improvements support.