IBM z Systems

# z/VM System Configuration Let's Talk About all the Choices

Bruce Hayden

IBM z/VM and Linux Solutions

Washington Systems Center

# Trademarks

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

Not all common law marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within its relevant market.
Those trademarks followed by ® are registered trademarks of IBM in the United States; all others are trademarks or common law marks of IBM in the United States.

For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml:

*, AS/400®, e business(logo)®, DBE, ESCO, eServer, FICON, IBM®, IBM (logo)®, iSeries®, MVS, OS/390®, pSeries®, RS/6000®, S/30, VM/ESA®, VSE/ESA, WebSphere®, xSeries®, z/OS®, zSeries®, z/VM®, System i, System i5, System p, System p5, System x, System z, System z9®, BladeCenter®

**The following are trademarks or registered trademarks of other companies.**

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

* All other products may be trademarks or registered trademarks of their respective companies.

**Notes**:
Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.
IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.
All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.
This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.
All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.
Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

# Agenda

- Basic rules on configuring CP
- SYSTEM CONFIG rules and overall guidelines
- System identifiers
- Devices statement
- Features statement
- Commands and Privilege classes
- Operators
- Syntax checking
- IPL Parameters

# z/VM System Configuration

- There are 2 primary files that configure CP
  - SYSTEM CONFIG
  - LOGO CONFIG (referenced in SYSTEM CONFIG)
  - These files reside on a PARM disk
- The CONFIG files are only read when you IPL the system
  - Dynamic system changes are made via CP commands
  - Logo configuration changes using the CP REFRESH command
- The User Directory configures virtual machines
  - It is read often by the system and can be dynamically updated

# What are the defaults?

- Some defaults are in the old configuration files (remember these?)
  - HCPSYS, HCPRIO, HCPBOX
  - They still exist and can still be customized, but it is not recommended
  - Any changes require the HL Assembler and for you to rebuild CP
- Can you IPL CP without a SYSTEM CONFIG file?
  - No.  CP requires:
    - CP_Owned statement to define the residence volume
    - System_Residence statement to define warm start and checkpoint areas
    - Operator_Consoles statement to find a console (or an IPL override)
- For defaults on each setting – you must read the documentation!
  - I'll cover some of the statements and defaults

# Configuration file rules

- General rules
  - Fixed or variable length file
  - Rexx style comments (start with “/*”, end with “*/”, can span lines)
  - Rexx style continuation (comma at the end of the line)
  - Blank lines are ignored – they do not affect continuation
  - Case does not matter.  Lines are uppercased except what is quoted
    - Feel free to make your configuration readable.  Please!
- Order of statements and duplicates
  - Order, for most statements, does not matter
    - The System Identifier is referenced in other statements, so usually this is early in the file.
  - Duplicates - for most statements, the last one wins.  Some are combined.

# Using multiple files

- The Imbed statement can imbed another file that is on the same disk
  - **Imbed *fn ft***
    - Where *fn* or *ft* can be “=” to use the file name or type of the source file
  - Special file name or type of “-SYSTEM-” is replaced with the system identifier
- Imbedded files can also contain Imbed statements
  - No limit(!) but you can’t create a circular imbed.

- This can help you create common configurations for your LPARs
  - Some people make extensive use of this

# Setting the system identifier

- It can be set by CPU model and serial number, or LPAR name
  - **`System_Identifier 2965 02BC957 ZVM01`**
  - **`System_Identifier LPAR VM01 ZVM01`**
- Wildcard characters are allowed
  - **`System_Identifier LPAR VM* ZVM01`**
- A default can be specified if no statement matches
  - **`System_Identifier_Default ZVMV6R30`**
  - You could also specify a default as **`System_Identifier * * ZVMV6R30`**
- If multiple statements match, the last one sets the identifier.
- You can also set it to the match the LPAR name
  - **`System_Identifier LPAR ZVM* &LPARNAME`**

# SSI requirements

- SSI systems share a common CONFIG for all members
  - Not a technical requirement, but recommended by IBM
- The correct system identifier must be set for each member
  - Using System_Identifier_Default is not recommended
- SSI systems also require
  - SSI statement to declare the PDR volume and slot number of each member
    - Only 1 SSI statement is allowed and all member's statements much match!
  - ISLINK statements to define ISLinks (CTC) to all other members
  - A statement to enable the SSI feature
- Because SSI systems use the System Identifier in several places, associating the correct one with the LPAR is very important.

# Qualified records

- Lines and blocks of lines can be qualified by system
  - These must follow System_Identifier statements, of course!
  - You are familiar with these if you have an SSI cluster
- Multiple qualifiers are allowed on a single statement; wildcards are allowed.
    - Wildcard rules are like CMS; “%” for single character, “*” for multiple
        - Example: LINUXVM1: LINUXVM2: LINUXVM3: Multithreading Enable
        - Or: LINUXVM*: Multithreading Enable
- Several systems can be Equated to a new name
    - Equate LINUXSYS LINUXVM1 LINUXVM2 LINUXVM3 TESTSYS%
    - LINUXSYS: VMlan Limit Transient 0

# Qualifying blocks of statements

- Begin and End qualify groups of statements
  - Qualifier required on Begin, optional on End
  - No nesting and no qualifiers allowed on statements in the block
  - If an error with Begin and End is found during IPL, hard wait code 1689!
    - Make sure you have syntax checked your file.
  - End is required in the same file as the Begin
- Example
    - ```
      LINUXVM1:  Begin
          CP_Owned  Slot 1  M01RES
          End
      ```
      ```

# Creative uses of the System Identifier

- You can redefine the System Identifier throughout your file
- I've used this to set up systems that may run on multiple LPARs
  - Note: This example is non-SSI. SSI may add a bit more complexity.
  - First, I set the identifier based on the LPAR name.
    - **System_Identifer LPAR * &LPARNAME**
      - Or more specific: **System_Identifier LPAR ZVM01 ZVM01**
  - Qualify statements based on the LPAR environment
    - Operator consoles, MAC prefix, machine features like Multithreading
  - Then set the "real" system name later on in the file
    - **System_Identifier LPAR ZVM01 BRUCETST**

# The sample configuration file, from installation

- The SYSTEM CONFIG from a fresh install gets you started
  - It doesn't have examples of everything you may need
    - I hope this presentation will tell you more
  - It includes some statements that just re-state the default.
    - For example, the Priv_Classes statement

```
Priv_Classes  ,
  Operator     A ,
  IOCP_Read    CE ,
  IOCP_Write   C ,
  HW_Service   F ,
  User_Default G
```

    - All of these are the default

# Sample configuration, continued

- You may think some statements just show the defaults, but they may not
  - System_Userids statement in the sample

```
System_Userids   ,
  Operator  OPERATOR  ,
  Account   DISKACNT  ,
  Dump      OPERATNS  ,
  Erep      EREP
```

  - The actual defaults

```
System_Userids   ,
  Operator  OPERATOR  Disconnect,
  Account   OPERACCT  ,
  Dump      OPERATNS  ,
  Erep      OPEREREP  ,
  Startup   AUTOLOG1  ,
  Symptom   OPERSYMP
```

  - I don’t know why the default user ids don’t match conventional use

# Devices statement

```
Devices ,
  Online_at_IPL 0000-FFFF,
  Sensed        0000-FFFF
```

- The sample statement shows the defaults
  - CP accepts and senses all devices.  It initializes all devices it can identify
    - Devices that cannot be sensed also need an RDEV statement
    - Note:  This assumes there is no IODF statement in the config file
- Devices statement can be specified more than once
  - Basically, multiple Devices statements are merged together
  - If the same parameter is specified again for the same device, last one wins
- References to “Online” and “Initialized” mean the same thing
  - **Online_at_IPL** and **Initialized_at_IPL** are synonyms
  - Also **Offline_at_IPL** and **Notinitialized_at_IPL**

# Devices statement, continued

- "New" parameter **Sensed_but_Offline**
  - Better than "Offline_at_IPL" because a non-initialized device is not sensed
    - A query will show "DEV 1000 OFFLINE" instead of "DASD 1000 OFFLINE"
    - You also see it when you issue QUERY DASD OFFLINE
- The parameter **NotAccepted** means no real device block is built
  - Harder to dynamically bring the device online later on
- Specifying DASD as "Shared" is required if the devices are shared
  - SSI clusters set this automatically for shared disks in the cluster
  - It must be set for some other uses, such as a shared RACF database
  - The RDEV statement can also set this attribute; also CP SET SHARED

# Devices statement, continued

- Other parameters which are used less often
  - **Assign_at_IPL**
    - Automatically assign a tape drive
  - **Dynamic_I/O**
    - (Default) allow dynamic I/O changes, must also be allowed via the Features statement
  - **SCmeasured**
    - (Default) collect subchannel measurement data
  - **Throttled**
    - Limit the rate of I/O to the devices
- The converse of all these is also valid
  - **NOassign_at_IPL, NOTdynamic_I/O, NOTscmeasured, NOTthrottled**

# The Features statement

- A lot of parameters on this statement
  - *Automatic IPL*
  - *Disconnect time*
  - *Maximum users*
  - *Showing Passwords*
  - *Retrieve settings*
  - *Virtual disk*
  - And, of course, *Enable* or *Disable* of CP features
- The sample from the install has a mixture of defaults and overrides
  - These sample settings are different from the default:
    - Retrieve, the default is 7 commands and the maximum is 7 instead of 20 and 255.
    - Vdisk user limit default is 0; the sample defines the limit as 140000 blocks
    - Passwords on commands. This one is interesting!
      - The documentation states the default is “No” for each one (**As it should be!)**
      - But actually, the default is “Yes”, and this agrees with the sample.
        - Development has been made aware that the code does not match the documentation

# The Features statement, continued

- Enable and Disable of features
  - There are 20 items that can be enabled (most are disabled by default)
    - The 2 that are enabled by default are:
      - ***New_Devices_Initialized_When_Added*** and ***IPL_Messages***
    - New releases and other enhancements have added to the list
  - The 6.3 install sample only lists 3 of the items as disabled
  - Some you would never enable unless you are developing CP code
    - Such as **CPchecking, XRC_Test, Cross_System_Timeouts**
  - Others are very good ideas to enable
    - Usability of the system, security, or are required for the system environment

# Features Statement – what to enable?

- My suggestions
  - **Auto_Warm_IPL** - You can always override with IPL parameter PROMPT
  - **Clear_Tdisk** - Your security policy probably requires it
  - **STP_Timezone** - If you have z/OS, it is probably enabled on your CPC
  - **Validate_Shutdown** - Prevent accidental system shutdowns!
- Optional
  - **PCI** - If you have PCI features, it is required
  - **Set_Devices** - Dynamically change the Devices list
  - **Set_Dynamic_IO** - If you may have to enable dynamic I/O

# Features statement

- Others to know about
  - **Logmsg_From_File** - Show a system logon message
  - **Prompt_After_Restart** - Sometimes enabled for a short time
  - **Prompt_After_Shutdown_RelPL**
  - **Set_Privclass** - Useful for testing, auditors may not like it!
  - **STP_Timestamping** - Required if you must timestamp I/Os
    - **XRC_Optional** goes along with this
  - **Throttle_All** - I doubt you'd want to enable this one!

# Other Features statement keywords

- Automatic IPL statements
  - **Auto_IPL**
  - **Auto_IPL_After_Restart**
  - **Auto_IPL_After_Shutdown_ReIPL**
    - These all accept as parameters the usual IPL prompt keywords
      - WARM, FORCE, COLD, CLEAN
        - Optionally NOENABLE, DRAIN, NOAUTOLOG, NODIRECT
    - These can be useful for test systems
    - I set up second level systems with "**Auto_IPL Force**" so they come up unattended
- **Disconnect_Timeout**
  - Default is 15 minutes before logoff after a forced disconnect
  - You want to specify this as "**Disconnect_Timeout Off**" !

# Features Statement

- VDisk limits
  - The default system limit is calculated based on your storage size
  - The default user limit is zero
  - Both can be changed dynamically with the SET VDISK command
  - “Infinite” is a valid setting, meaning there is no limit
- Maxusers
  - Can be from 1 to 99999
  - Dynamic changes via the SET MAXUSERS command
  - Users with OPTION IGNMAXU can always log on

# The SET statement

- This is where you set the time allowed to shut down guests
  - **`Set Shutdowntime nn`**
  - **`Set Signal Shutdowntime nn`**
- The Shutdowntime is the time reserved for CP to complete its shut down
  - Defaults to 30 seconds, which should be plenty except for very large systems
  - If WITHIN is specified on a command (such as SIGNAL or SHUTDOWN):
    - The system time is subtracted from the WITHIN time to give the guest shutdown time
    - The same is true for a hardware deactivation, which is 300 seconds
- The **Signal Shutdowntime** is the default time allowed for guests
  - Ignored if WITHIN is specified, and the default is zero
  - The default total system shutdown time is the addition of both intervals
- Both values can be set or changed with CP commands

# Other enabling and setup statements

- **Crypto APVirtual AP *a* Domain *d***
  - This is a somewhat new statement, to reserve shared crypto domains
  - It is recommended if you are using Crypto with Linux
    - Without this statement, crypto statements in the user directory determine the usage
    - CP will assign shared domains in the order specified.  Multiple statements are allowed.
    - CP only uses 1 crypto type for sharing (accelerator or co-processor)
    - A range of AP numbers and Domain numbers are allowed
- **Enforce_by_Volid** *ON* or *OFF*
  - Requires you to specify the volume id (label) for Attach or Dedicate
    - If your DASD pool is available to many systems, this may avoid mistakes!

# Other setup

- **Multithreading** *Enable* or *Disable*
  - Required to use multithreading on IFLs on z13 and z13s.  Disabled by default
  - There is no (current) CP command to change the running system
    - See the z/VM 6.4 preview announce for an statement of direction on “Dynamically managed thread activation levels”
- **SRM** statement
  - Lets you set some SET SRM values at IPL time
    - Such as CPU Polarization mode to Horizontal or
    - DSPWDMethod (Dispatcher Work Distribution Method) to Rebalance
  - Normally, the defaults are what you want
    - But – if you’re one of those “special systems” that need a different setting..

# Privilege classes of CP commands

- What are they?
  - CP commands have 1 or more privilege classes assigned to them by IBM
  - This is the IBMCLASS. IBM defines classes A-G and reserves class H
  - The system supports 32 privilege classes. A-Z and 0-6
  - CP commands may also be class “Any”; available to any user, like LOGOFF
  - G is the general “unprivileged” user class. Other IBM classes are privileged
- You can freely modify the class(es) assigned to a command
    - or to a subcommand of a QUERY or SET command
  - Create new classes for users or servers with only the commands and diagnoses needed for the task
    - Usually known as “least privilege” – Give no more privilege than needed

# Modifying CP Commands

- **Modify Command *command* IBMclass *x* Privclasses *classes***
- This allows you to modify the privilege classes of a command
  - You specify the command and the IBM assigned privilege class
  - You may completely change the classes or just add to the IBM assigned ones
  - If there is a duplicate statement, only the first one is used
- Other command modifications
  - Create an alias to an existing CP command
    - **Define Alias *alias* For *command* Abbrevlength *nn***
  - Disable a CP command
    - **Disable Command *command***
  - These commands also work with Query and Set subcommands

# Storage statement

- CP calculates many allocations based on the LPAR memory size
- The **Storage** statement lets you override those calculations
  - Some options you may need when using PCIe functions:
    - IOAT – a storage subpool.  See the documentation on what to specify
    - LOCKING – Issue messages when size of lock requests exceed usage percentages
  - SCMBK – Subchannel measurement blocks
    - More space available for adding new I/O devices
  - EDEVICE – emulated FBA devices on FCP channels
    - Sets the reserved memory pool for expected EDEVs that will be added
  - Others sizes you can specify:
    - CP trace area, Reserved pages maximum, and the Agelist size.

# Real Devices

- This is the **RDEVICE** statement, and **CU** statement for DASD
- The CU statement allows you to restrict PAV by control unit
  - The default is to enable the highest (best) level of support
- Rdevice allows you to specify these things for devices
  - Define devices that cannot be sensed
  - Additional characteristics such as shared DASD, minidisk cache, or spooling
  - For unsupported devices, the device class (DASD, printer, tape, etc.)
  - EQID (equivalency ID) is required for some devices in an SSI cluster
    - OSA ports, Hipersockets, FCP channels, and channel to channel devices
    - It is required so that guest relocations can occur for guests using these devices
    - EQID is allowed on any Rdevice statement, but these are the ones that need it

# Speaking of real device numbers..

- Did you know you can now specify them for system disks?
  - **CP_Owned Slot *nnn volid* RDEV *rdev***
    - The RDEV operand is optional; with it, the label on *rdev* must be *volid*
    - Any other DASD with label *volid* is ignored (but an IPL message is issued.)
  - **User_Volume_RDEV *volid* RDEV *rdev***
    - This is a new statement, used instead of the other User_Volume statements
    - Each statement only defines 1 volume, no wildcards
    - Maybe a good use for an imbedded file.
- This avoids any “duplicate volser” problems at IPL!
  - If you share DASD with many systems, be sure to think about this
  - If users (meaning Linux admins) have the ability to write labels, you really want to use this.

# Operators and Operator consoles

- CP requires a console to log on the OPERATOR at IPL time
  - Otherwise it loads a disabled wait PSW with code x‘1010’
- The **Operator_Consoles** statement specifies a list of addresses of locally attached 3270 displays to choose from
  - The HMC 3270 function is specified as “System_3270”
  - The HMC operating system messages panel is “System_Console”
  - The device addresses are OSA-ICC defined devices
  - CP searches for a working device in the order specified
- The IPL parameter “CONS=” overrides this statement
- Tip:
  - Placing “System_Console” at the end of the list ensures your system will always be able to IPL.

# Emergency message consoles

- The statement is **Emergency_Message_Consoles**
- This is where CP will send shutdown, abend, and dump messages
- Same syntax as **Operator_Consoles** except “System_3270” not allowed
- Not required – the default is the Operator_Consoles list
  - But only the consoles that are operational at IPL time
  - An address on the IPL parm CONS= is added to the list
  - The “System_Console” is always included
- Limits:  100 devices can be specified.
  - Note that you can have up to 500 Operator_Consoles.

# Who is the System Operator?

- The "**`System_Userids Operator`** ***`user`***" is the system operator at IPL
  - Even if that user does not exist in the directory!
  - This is the first user logged on to the system
- What if that user logs off?
  - A user id on the **Alternate_Operators** statement is made the system operator
    - If the user is logged on and has the required privilege class
    - Otherwise, the system has no system operator (QUERY SYSOPER)
  - The system operator will be designated when one of these occur:
    - The default OPERATOR logs on
    - Any user with the required operator privilege class logs on (like MAINT630 or MAINT)
    - The SET SYSOPER command successfully sets a new operator

# The Journaling statement

- Lock out logon or CP LINK after invalid attempts
  - Attempts are also written as accounting records
  - Messages about attempts can be sent to the Operator or another user
- Only really useful if you don’t have an ESM, such as RACF
  - ESM controls logon attempts and automatic disable
  - No link passwords with an ESM, so no need to lock out link
- Facility is off by default.
  - CP commands to enable and disable are also off by default

# Syntax check

- The CPSYNTAX utility checks the syntax of your file
  - Always run it after an update!  It is found on MAINT 193.
  - Incorrect statements, bad syntax, unknown keywords
  - It does not check every statement for duplicate operands or interactions with other statements
  - It is a good check, but no guarantee it will find all problems an IPL would find
- If you use qualified statements and multiple identifiers:
  - Run it for each LPAR and/or CPU specification you have.  The options are:
    - **CPUID *model serial***
    - **LPAR *lparname***
    - **SYSTEM *sysname***
  - Wildcards and multiple options are allowed

# IPL Parameters for CP

- These are specified in the IPLPARMS area of the stand alone loader
  - They can also be set with SET IPLPARMS for a SHUTDOWN REIPL
- The valid parms are documented in *z/VM System Operation*
  - CONS=*addr* or CON=*addr*
    - Override the Operator_Consoles statement and use this console address
  - FN=*filename* and FT=*filetype*
    - The system configuration file. Default is SYSTEM CONFIG
  - PDNUM=*n*, PDOFF=*offset*, PDVOL=*raddr*
    - Where to find the parm disk for SYSTEM CONFIG. Which parm disk or cylinder location
  - PROMPT
    - Prompt for startup, even if Auto_Warm_IPL or Auto_IPL is specified
    - Only valid on the stand alone loader screen. Can be changed on the Features statement

# CP IPL parameters, continued

- Diagnostic parameters
  - NOEXITS, NOHCD
    - Don’t load any exits for this IPL, or ignore the IODF statement
  - REPAIR
    - Used very carefully if no members of your SSI cluster will start.
      - It bypasses many SSI management functions, so no other members must be active!
    - Forces start-up parameters NOAUTOLOG and DISABLE (even non-SSI)
- STORE= parameter
  - Specifies the amount of memory CP will use, up to the LPAR amount
    - The SET STORAGE can be used to dynamically increase memory
    - Not needed for dynamic increases; usually just for testing

# Specifying IPL parameters

- Using the stand alone loader, enter a 3270 console addr as loadparm
  - Change or specify the parameters in the IPL PARAMETERS area
- Or, use these special loadparms:
  - **CONS*xxxx*** creates iplparm CONS=*xxxx*. HMC 3270 is device “SYSG”
  - **FN*xxxxxx*** creates iplparm FN=*xxxxxx*
  - **I*rrrrr*** or **I*rrrrr.p*** overrides the load device (*rrrrr*) or device and parm extent (*p*)
    - The first letter is I for IPL device.
- FN*xxxxxx* allows you to use conditional IPL parameters. An example:
  - Loadparm of `FNSITE2`, IPL parms: `FN:SITE1(PDVOL=1111)FN:SITE2(PDVOL=2222)`
  - IPL parms passed to CP: PDVOL=2222 FN=SITE2

# What did you learn?

- There are probably more statements and parameters than you knew about!
  - Hopefully you now know more and will make use of some of them
- I didn't cover several statements
  - Many are ones you're used to working with
  - Many are covered elsewhere, such as vswitch statements
  - Some are for special situations, like loading new code and exits
  - Please give me feedback if there are ones I should have covered
- Next year, we'll talk about z/VM 6.4!
  - What is new in SYSTEM CONFIG, IPL parameters, and so forth

# The End

## Thank you for listening!

Contact information:
Bruce Hayden
bjhayden@us.ibm.com

IBM Washington Systems Center
Endicott, NY / Clinton, IA





41