z/VM Memory Management Update

2026 VM Workshop

Romney White

romneyw@us.ibm.com

Agenda

- **Active Drain**
- **Memory Terminology Simplification**
- **Memory Scalability Infrastructure**
- **Memory Overcommitment Control**

Active Drain

Active Drain – Overview

- **Enables a system administrator to remove a PAGE volume on a running z/VM system, without requiring an IPL**
- **Allows a customer to move PAGE volumes to a new storage server without requiring a planned z/VM outage to remove the volumes from the previous storage server, enhancing existing z/VM dynamic configuration capabilities**
- **New ACTIVE option on existing CP DRAIN command**
- **Available in z/VM 7.4 Feature Pack 4**

Active Drain – Example Use

1. CPFMTXA replacement PAGE volumes
2. Update SYSTEM CONFIG file adding/modifying CP_OWNED statements so the new PAGE volumes will be available after the next system IPL
3. Issue corresponding CP DEFINE CPOWNED commands to identify these new PAGE volumes to the running system
4. CP ATTACH <rdev> SYSTEM to bring these new PAGE volumes online
5. Remove old PAGE volumes from SYSTEM CONFIG CP_OWNED statements (mark as RESERVED) to ensure they will not be used after the next system IPL
6. Issue CP DRAIN <rdev1> <rdev2> … PAGE PASSIVE to allow natural relocation of in-use PAGE space to non-draining PAGE volumes (optional)
7. Issue CP DRAIN <rdev1> <rdev2> … PAGE ACTIVE to force any remaining pages off the old PAGE volumes to memory or other non-draining PAGE volumes
8. CP DETACH <rdev> SYSTEM to remove the old PAGE volumes from the system
9. Issue CP DEFINE CPOWNED commands to mark the slots RESERVED for the old PAGE volumes

Active Drain – Prior Limitation

- **Prior DRAIN functionality (now referred to as PASSIVE DRAIN) stopped new allocations on PAGE volumes, but did not forcibly remove active use of PAGE volumes**
- **Natural paging activity or virtual machines logging off would often remove most use of draining PAGE volumes**
- **Often a small number of in-use slots remained, preventing the PAGE volume from being detached from SYSTEM, requiring an IPL to safely clear any remaining use of the PAGE volume**

Active Drain – New Functionality

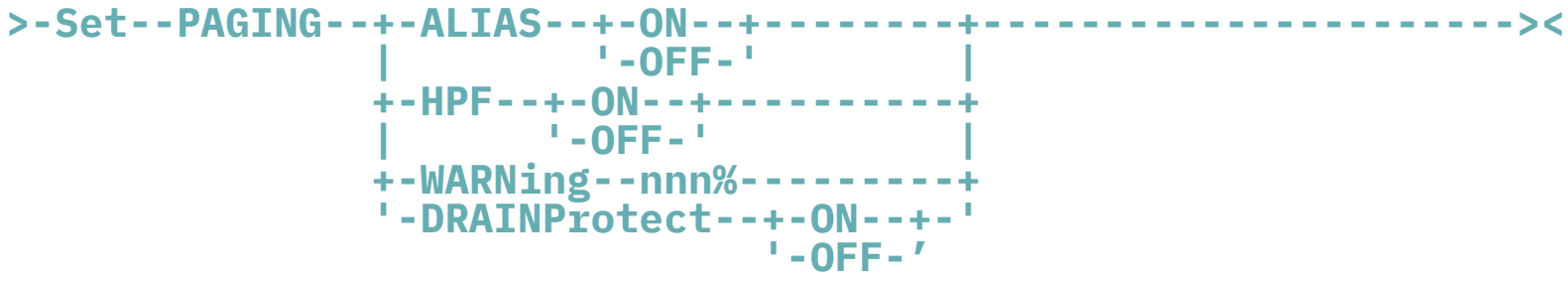
- **New ACTIVE option on CP DRAIN command will locate all existing use of slots on the draining PAGE volume**
- **These pages will be brought into memory via a read I/O, freeing the PAGE slots**
  - **And might be subsequently written to a different, non-draining, PAGE volume**

Active Drain – What It Does (for the curious)

- z/VM maintains page state information in DAT structures that it builds for all virtual address spaces
- z/VM maintains bit masks to track which slots are in-use vs available on a PAGE volume
- There is no mapping from an in-use PAGE slot to the virtual page whose contents reside on the disk
- To find all the in-use PAGE slots on a volume (or set of volumes), the control program must step through all DAT tables for all virtual address spaces looking for pages on draining volumes
  - When KEEPSLOT=YES, resident pages might also have a PAGE slot assigned. These can be dropped without a read I/O
  - For non-resident pages with a PAGE slot assigned, a read I/O must be performed to bring the page into memory before releasing the slot

Active Drain – Additional Safety Measures

- **Insufficient PAGE space can cause a system outage (PGT004 Abend)**
- **Removing existing/in-use PAGE space can lead to insufficient PAGE space if steps to ensure availability of replacement PAGE volumes are omitted**
- **A new DRAINPROTECT setting:**



- **When ON (default setting), z/VM will reject DRAIN commands when in-use PAGE space exceeds or is predicted to exceed the system warning percentage**
- **Regardless of this setting:**
  - **the system will issue a warning message for a DRAIN command when in-use PAGE space exceeds or is predicted to exceed the system warning percentage**
  - **the system will automatically select and START a draining PAGE volume (if one exists) until the predicted in-use PAGE space is below the warning threshold**

Memory Terminology Simplification

Memory Terminology Simplification – Overview

- **Historically, IBM Z has used the term “storage” to refer to what the industry and most other architectures refer to as “memory”**
- **This has been a source of confusion, particularly for those new to the platform, as the term is overloaded to also refer to non-volatile disk and flash technologies**
- **z/VM is taking it’s first step towards transitioning to using industry-standard terminology, while respecting long-standing expectations of compatibility on the platform**
- **Availability: Base of z/VM 7.5**

Memory Terminology Simplification – What's New

- **New alternative command and configuration statement names and operands to allow "memory" to be used where "storage" (and related/similar words) was previously used to refer to what the industry understands to be memory**

- **New commands/configuration statements**
  - **to control CP's terminology choice on responses/messages when "storage" is specified**
  - **to control whether an informational message is produced when "storage" is used, to help identify existing use of the "storage" terminology**

Memory Terminology Simplification – What's Not (yet) Changed

- **DIRECTXA statements not updated**
- **CMS, SMAPI, and other components not updated**

- **Reminder, this is a first step, expect additional changes may come later**

Memory Terminology Simplification – What's Next

- **The "storage" commands/statements/operands are deprecated, and you should begin using the corresponding "memory" alternatives**
- **In a future release, we will likely change the default terminology to memory and enable deprecation warnings by default**
- **Eventually "storage" terminology will be removed**

Memory Terminology Simplification – Examples

Q TERMINOLOGY SYSTEM MEMORY

System terminology memory is OFF
Ready; T=0.01/0.01 11:28:06

Q DEPRECATION SYSTEM WARNING

System deprecation warnings are OFF
Ready; T=0.01/0.01 11:28:18

Q V STOR

STORAGE = 64M
Ready; T=0.01/0.01 11:28:42

Q V MEM

MEMORY = 64M
Ready; T=0.01/0.01 11:28:46

SET TERMINOLOGY SYSTEM MEMORY ON

Ready; T=0.01/0.01 11:29:26

Q V STOR

MEMORY = 64M
Ready; T=0.01/0.01 11:29:30

Q V MEM

MEMORY = 64M
Ready; T=0.01/0.01 11:29:32

SET DEPRECATION SYSTEM WARNING ON

Ready; T=0.01/0.01 11:30:12

Q V STOR

MEMORY = 64M
HCPQVP1398I A deprecated operand was specified: STORAGE
HCPQVP1398I Processing continues
Ready; T=0.01/0.01 11:30:17

Memory Scalability Infrastructure

Memory Scalability Infrastructure – Overview

- **Release level infrastructure (internals) changes in preparation for future real memory scalability efforts**
- **4GB minimum real memory required to IPL (regardless of whether running in a partition or virtual machine)**
  - **6GB PERM memory required before adding RECONFIG memory**
- **Real memory rounded to a multiple of 2GB, any remainder will be ignored**
  - **INITIAL and RESERVED memory should sum to a value evenly divisible by 2GB to avoid unusable memory in the partition**

Memory Scalability Infrastructure – Overview ...

- 2GB of memory will be allocated to CP for future exploitation
  - Intended to be used for System eXecution Space (SXS) constraint relief (SXA004 avoidance) and better support for allocation of larger buffers
  - Reconsider partition memory size at z/VM release upgrade time, rather than in the service stream

- Availability: Base of z/VM 7.5

Memory Scalability Infrastructure – Motivation

- **Rapid processor memory growth over the past 10 years – 2000%**
  - **From 3TB on zEC12 (2014) to 64TB on z17 (2025)**
- **105% PU count and 230% MIPS growth in the same time frame**
- **z/VM currently supports up to 4TB of real memory**
  - **Supporting more real memory expected to reduce systems management costs (fewer, larger partitions) and allow for supporting larger virtual machines**

Memory Scalability Infrastructure – What's Next?

- **Current memory management algorithms based on 4K frame/page size limits z/VM's ability to scale beyond 4TB**
  - **4TB is over 1 billion 4K frames to manage**

- **IBM Z hardware supports managing virtual memory in 1MB or 2GB granularity**
  - **Fewer things to manage**
  - **More efficient TLB usage**
  - **Paging not feasible with current architectural restrictions**

Memory Scalability Infrastructure – What's Next?

- Pursuing a solution to split real memory within a partition
  - One section managed as 4K frames, like it is today, up to 4TB in size, as currently supported
  - A new section managed as 2G region-frames which will be used to back the virtual memory of a new type of virtual machine, as well as for CP internal purposes
  - Goal is to support up to 16TB real memory

- A new type of virtual machine whose backing memory is managed in 2G granularity
  - Non-pageable and backed in a fixed location at LOGON
  - Additional restrictions on functionality expected
  - Expect some performance benefit from more efficient TLB usage with smaller DAT structures

Memory Scalability Infrastructure – What's Next? ...

- 2GB region CP internal area reserved for System eXecution Space (SXS) relief
  - Currently the area of SXS used for dynamic allocation of memory buffers for CP use must be below 2G
  - Growing workload tends to grow CP structures as well, but the below-2G SXS can't get any bigger
  - Plan to move an existing heavy use of this space into a new (>2G) space that will be backed by this real memory
  - Additional infrastructure improvements in support of larger (bigger than 4K, up to 1MB) allocations envisioned

Memory Overcommitment Control

Memory Overcommitment Control – As-Is Scenario

- **Allowing a virtual machine to log on, relocate, change its storage size, or create an address space when resources might not be available to support it can lead to trouble (performance problems, denial of service, missed SLAs, system failures)**
  - **For the virtual machine attempting to perform the action**
  - **For those already logged on**

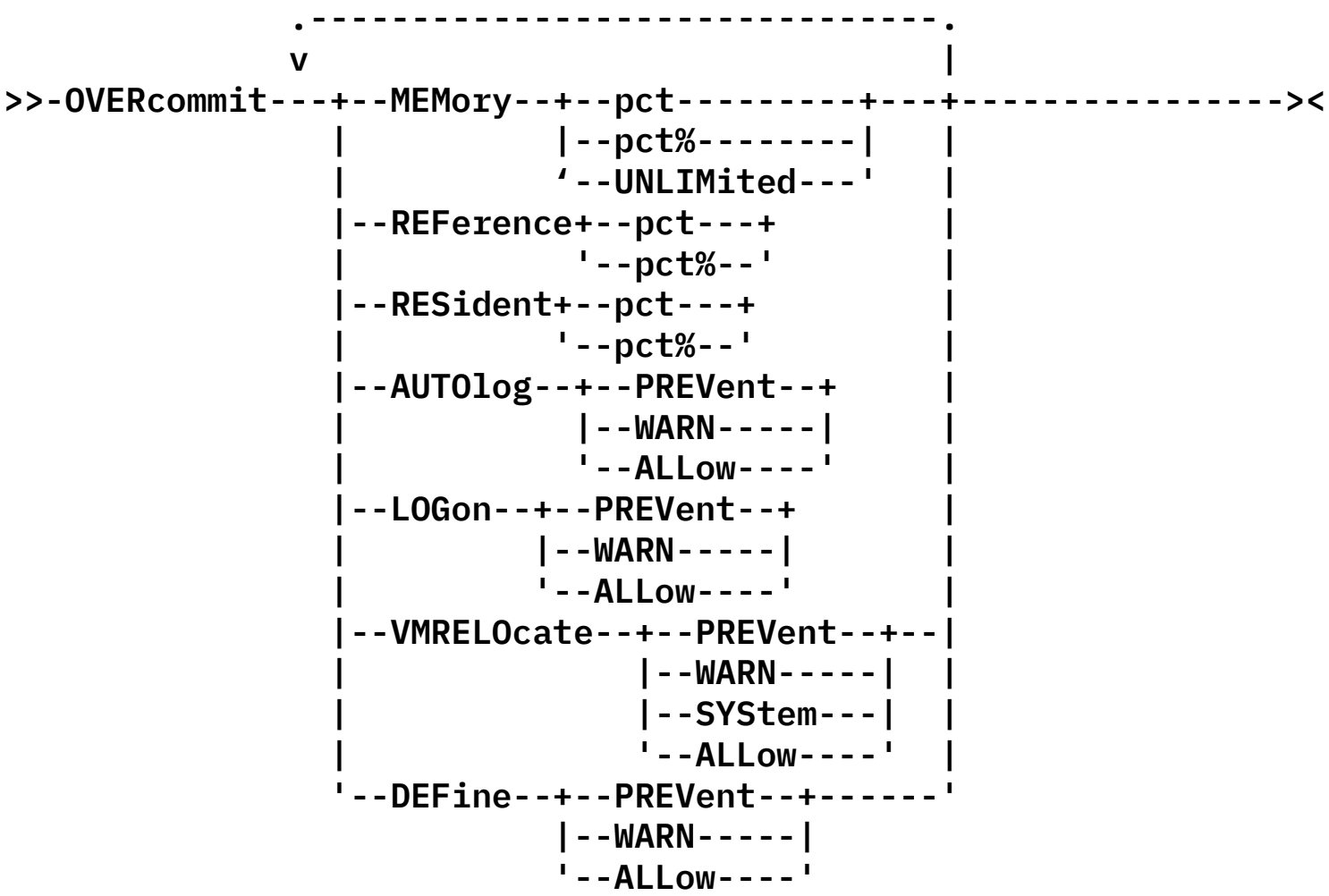
Memory Overcommitment Control – To-Be Scenario

- **Establish acceptable level of memory overcommitment as a percentage of pageable real memory (e.g., 150%)**
- **For each memory-related commitment request (e.g., LOGON, DEFINE STORAGE), calculate resident and auxiliary page counts based on estimated percentages of referenced and resident pages**
  - **Determine if projected memory use associated with request would cause overcommitment level to be exceeded**
  - **Determine if paging resources would be exhausted (avoid PGT004 abend)**
  - **If overcommitment of either memory or paging is predicted, take designated action (Prevent, Warn, Allow)**
- **Maintain running total of projected resident and auxiliary memory commitments**

Memory Overcommitment Control – Externals Changes

- OVERCOMMIT command and system configuration statement
- QUERY OVERCOMMIT command
- XAUTOLOG command FORCE option
- DEFINE and SET RESPOOL command overcommitment options
- User Directory OPTION statement RESPOOL option
- ADRSPACE CREATE overcommitment checking
- New and changed messages
- New Storage Domain “Memory Overcommitment Check Result” Monitor Event record
- New *VMEVENT overcommitment check result message

Externals Details – OVERCOMMIT Command



OVERCOMMIT Command ...

Authorization

Privilege Class: C

Purpose

Use OVERCOMMIT to set or change memory overcommitment checking parameters.

Operands

MEMory pct
pct%
specifies the overcommitment level, which is the total amount of virtual memory that guests are allowed to consume in aggregate, expressed as a percentage of total system pageable memory. It must be an integer value between 100 and 9999, optionally followed by a percent sign (%).

OVERCOMMIT Command ...

MEMory UNLIMited
deactivates memory overcommitment checking.

REFerence pct
pct%
specifies the percentage of a guest's defined memory to assume it will reference, to be used for overcommitment checking purposes. It must be an integer value between 1 and 100, optionally followed by a percent sign (%). If this operand is not specified, its previous setting or the initial default of 100% is retained.

RESident pct
pct%
specifies the percentage of a guest's referenced memory to assume it will need kept resident in real memory, to be used for overcommitment checking purposes. It must be an integer value between 1 and 100, optionally followed by a percent sign (%). If this operand is not specified, its previous setting or the initial default of 50% is retained.

OVERCOMMIT Command ...

AUTOlog PREVent
AUTOlog WARN
AUTOlog ALLow

specifies the action to take when a guest logon is attempted using the AUTOLOG or XAUTOLOG command and it would cause memory to exceed the memory overcommitment level. PREVENT will prevent the guest from being logged on. WARN will allow the guest to log on but will produce a warning message on the system operator's console. ALLOW will allow the guest to log on. If a guest is XAUTOLOGged with the FORCE option and would exceed the memory overcommitment level, a warning message is sent to the system operator's console and the logon logon is allowed. If this operand is not specified, its previous setting or the initial default of ALLOW is retained.

LOGon PREVent
LOGon WARN
LOGon ALLow

specifies the action to take when a guest logon is attempted using the LOGON command and it would cause memory to exceed the memory overcommitment level. PREVENT will prevent the guest from being

OVERCOMMIT Command ...

```
        logged on. WARN will allow the guest to log on but will produce a
        warning message on the system operator's console. ALLOW will allow
        the guest to log on. If a guest is XAUTOLOGged with the FORCE option
        and would exceed the memory overcommitment level, a warning message is
        warning message on the system operator's console. ALLOW will allow
        the guest to log on. If this operand is not specified, its previous
        setting or the initial default of ALLOW is retained.

VMRELOcate PREVent
VMRELOcate WARN
VMRELOcate SYStem
VMRELOcate ALLow
        specifies the action to take when a guest relocation is attempted to
        another member of an SSI cluster using the VMRELOCATE command and it
        would cause memory to exceed the memory overcommitment level on the
        target system. PREVENT will prevent the guest from relocating. WARN
        will allow the guest to relocate but will produce a warning message
        on the system operator's console. ALLOW will allow the guest to
        relocate. SYSTEM will bypass memory overcommitment checking and
        perform standard VMRELOCATE command memory checking. If SYSTEM is
```

OVERCOMMIT Command ...

not in effect and a guest relocated with the FORCE STORAGE option would exceed the memory overcommitment level, a warning message is sent to the system operator's console and the relocation is allowed. If this operand is not specified, its previous setting or the initial one of SYSTEM is retained.

DEFine PREVent
DEFine WARN
DEFine ALLow

specifies the action to take when a guest virtual storage size redefinition is attempted using the DEFINE STORAGE command and it would cause memory to exceed the memory overcommitment level. PREVENT will prevent the storage redefinition. WARN will allow the storage to be redefined but will produce a warning message on the system operator's console. ALLOW will allow the storage to be redefined. If this operand is not specified, its previous setting or the initial default of ALLOW is retained.

OVERCOMMIT Command ...

Usage Notes

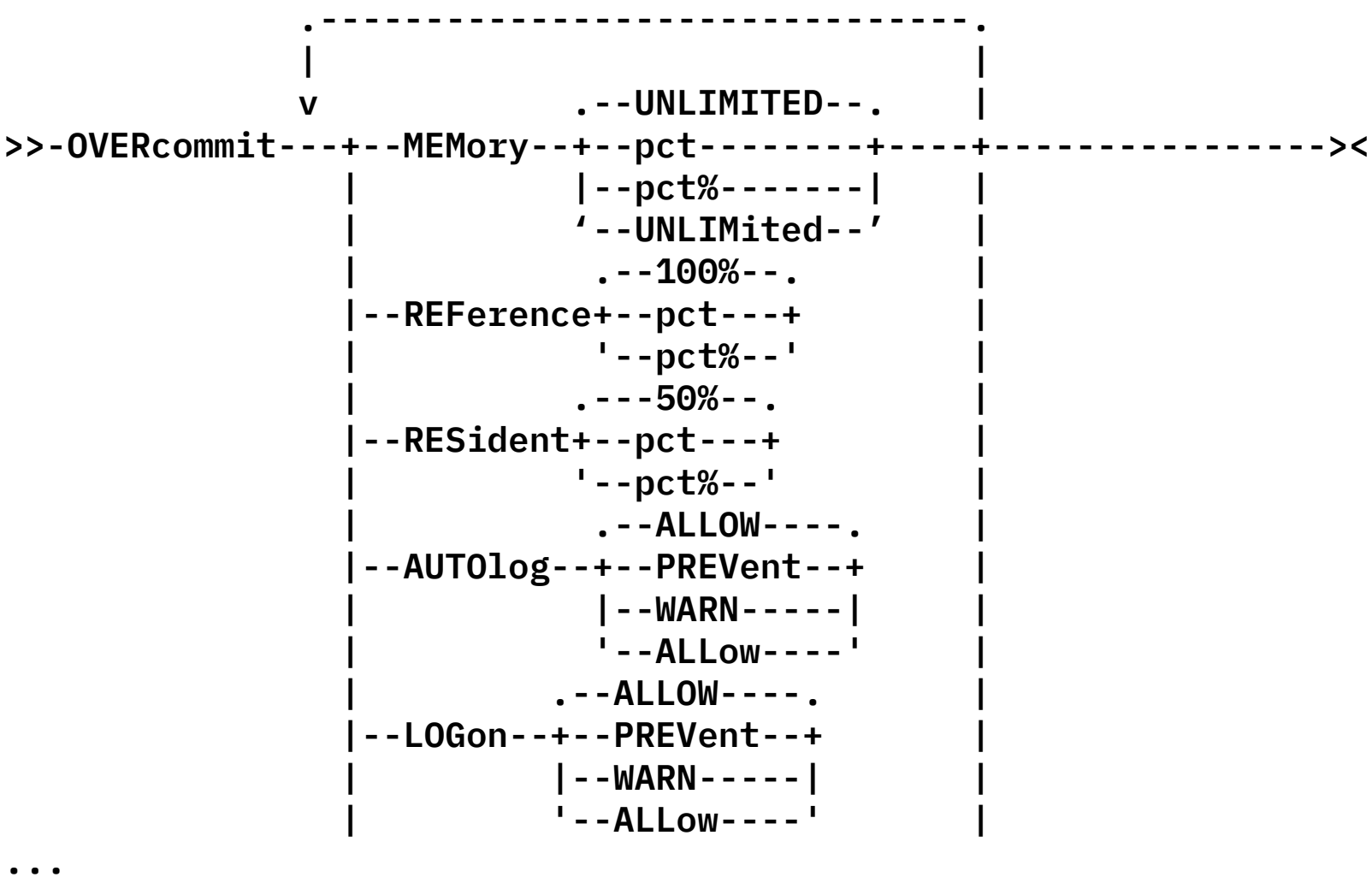
1. The amount of pageable memory is displayed as part of the response to the QUERY FRAMES command.

2. The OVERCOMMIT system configuration statement can establish default memory overcommitment settings during z/VM IPL.

Messages

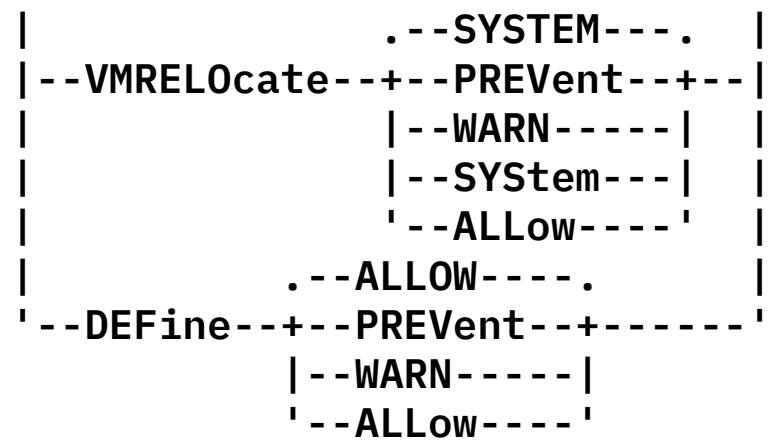
* HCP002E Invalid operand - operand
* HCP003E Invalid option - option
* HCP013E Conflicting option - option
* HCP026E Operand missing or invalid
* HCP2541E New OVERCOMMIT settings would cause excess memory overcommitment.
* HCP6704E Missing token at end of line
* HCP6706E Invalid string – string

Externals Details – OVERCOMMIT Statement



OVERCOMMIT Statement ...

...



...

Externals Details – QUERY OVERCOMMIT

```
>>-Query--OVERcommit--+--------+------------------------------><
                      '-userid-'
```

Authorization

Privilege Class: B

Purpose

Use QUERY OVERCOMMIT to determine the memory overcommitment checking settings and current projected memory and paging space use for the z/VM system or the projected memory and paging space use for a specific virtual machine.

Operands

userid

specifies the user whose projected memory and paging space use are to be displayed. If this operand is omitted, the system memory overcommitment settings and projected memory and paging use are displayed.

QUERY OVERCOMMIT

```
Responses

Response 1:

If a memory overcommitment level is not defined, the response displayed is

   Memory overcommitment checking is not enabled.

Response 2:

If a memory overcommitment level is defined, then the response displayed is

   Memory overcommitment llll%  Reference rrr%  Resident sss%
                                AUTOLOG action  LOGON action
                                DEFINE action   VMRELOCATE action
   Projection: Resident memory mmm Page space ppp Overcommitment ooo%

llll
         specifies the overcommitment level, which is the amount of memory
         allowed to be consumed by guests as a percentage of pageable memory.
```

QUERY OVERCOMMIT ...

rrr

specifies the percentage of a guest's defined memory to assume it will reference, to be used for overcommitment checking purposes.

sss

specifies the percentage of a guest's referenced memory to assume it will need kept resident in real memory, to be used for overcommitment checking purposes.

action

specifies the action to be taken if memory overcommitment is detected for the designated command or commands, and is one of the following:

Prevent	will prevent the guest from being logged on.
Warn	will allow the guest to log on but will produce a a warning message on the system operator's console.
Allow	will allow the guest to log on.
System	is reported for VMRELOCATE only and causes the standard relocation memory checks to be performed

QUERY OVERCOMMIT ...

mmm
is the amount of resident memory that current virtual machines are projected to consume in units of megabytes (MB), gigabytes (GB) or terabytes (TB).

ppp
is the amount of page space that current virtual machines are projected to consume in units of megabytes (MB), gigabytes (GB) or terabytes (TB).

ooo
is the current projected memory overcommitment level.

Response 3:

If a user identifier is specified, then the response displayed is

```
   User: xxxxxxxx Projection: Resident memory mmm Page space ppp
```

xxxxxxxx is the user identifier.

QUERY OVERCOMMIT ...

mmm

is the amount of resident memory that the user's virtual machine is projected to consume in units of megabytes (MB), gigabytes (GB) or terabytes (TB).

ppp

is the amount of page space that the user's virtual machine is projected to consume in units of megabytes (MB), gigabytes (GB) or terabytes (TB).

Messages

* **HCP003E Invalid option - command contains extra option(s) starting with option**
* **HCP045E *userid* not logged on**

Externals Details – XAUTOLOG

XAUTOLOG

```
>>-XAUTOLog--userid--+------------------+---------------------->
                    +-Ipl--+-vdev----+-+
                    |      '-sysname-' |
                    '-NOIpl------------'

...

>--+-------+--+------------------+--+-------+------------------>
   '-SYNch-'  |    (1)           |  '-FORCE-'
              '-ON------+-rdev-+-'
                        '-ldev-'

...
```

Notes:

1. The PASSWORD, PROMPT, FORCE, and ON operands are valid only for a Class A or Class B user.

XAUTOLOG ...

...

FORCE
specifies that any possible memory overcommitment that is estimated to occur because of logging on the guest should not cause the command to terminate. The FORCE option is available only to Class A or Class B users.

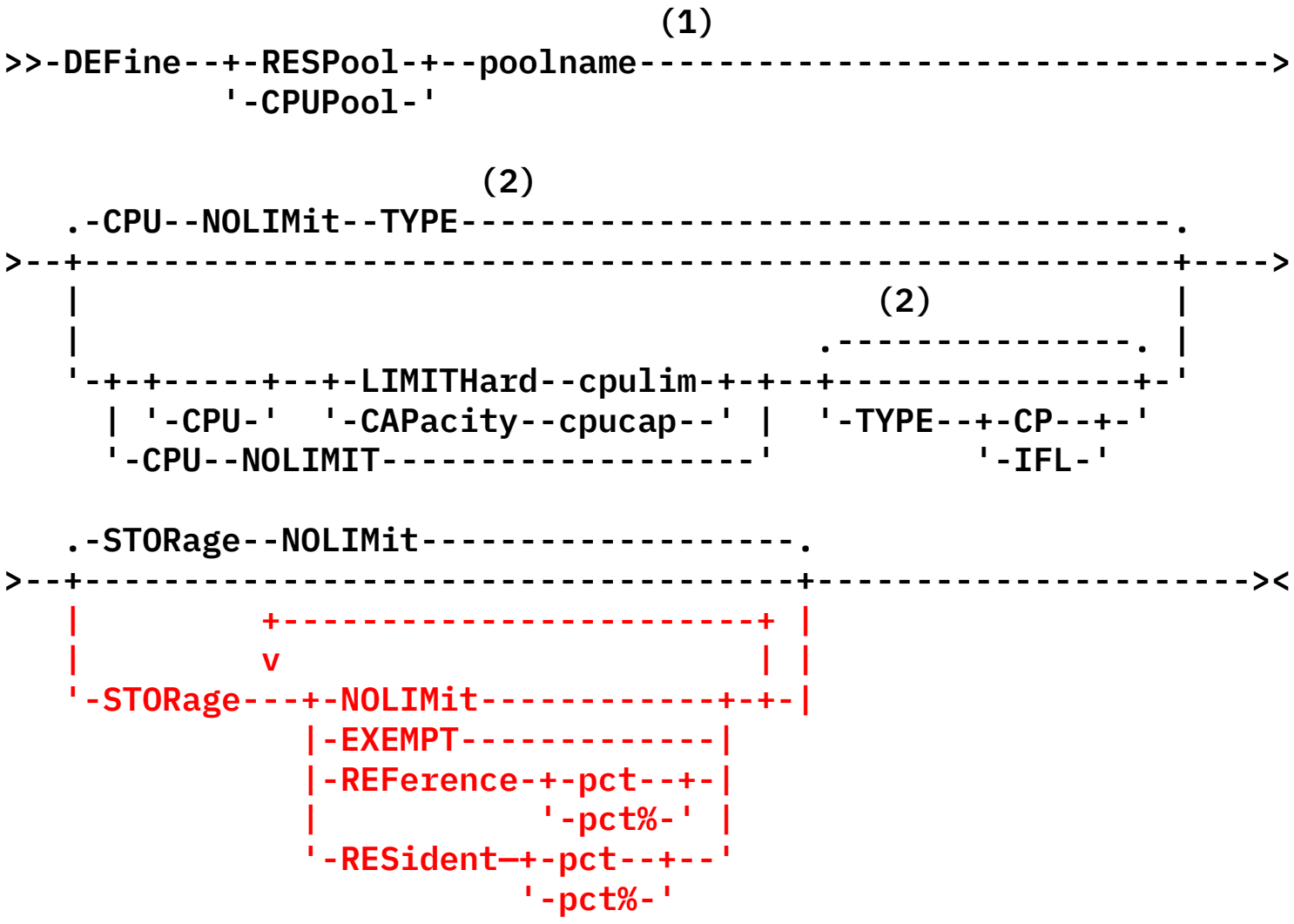
...

Messages

...

* HCP059E {XAUTOLOG|AUTOLOG} failed for userid - {IPL missing|IPL failed|accounting errors|incorrect password|memory overcommitment level would be exceeded}

Externals Details – DEFINE RESPOOL



DEFINE RESPOOL ...

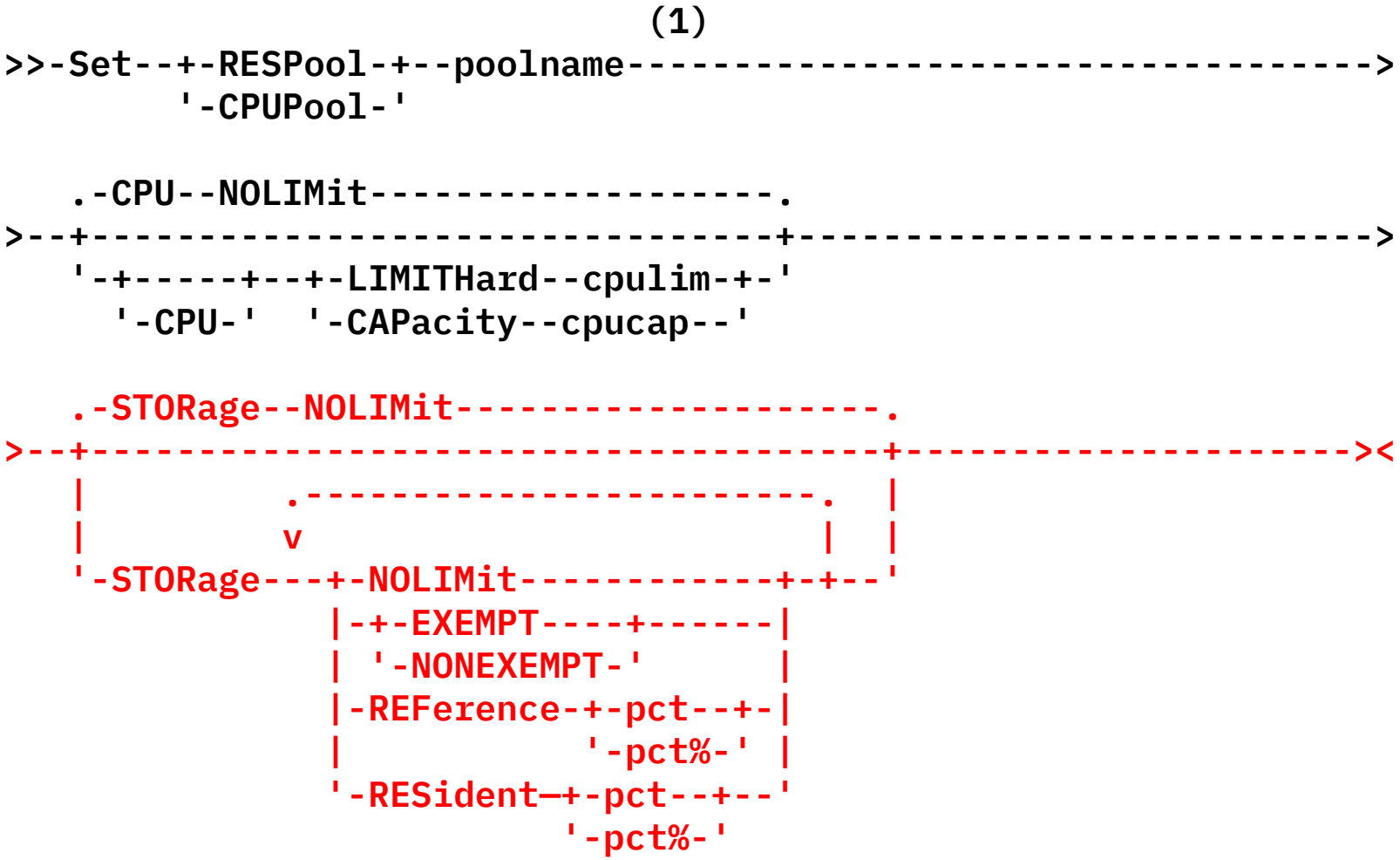
STORage NOLIMit
prevents the enforcement of memory limiting for the defined pool.

STOrage EXEMPT
exempts guests in the pool from memory overcommitment enforcement.

STOrage REFerence pct
pct%
for a guest in the pool, specifies the percentage of defined memory to assume it will reference, to be used for overcommitment checking purposes. It must be an integer value between 1 and 100, optionally followed by a percent sign (%). If this operand is not specified, the current OVERCOMMIT setting is used.

STOrage RESident pct
pct%
for a guest in the spool specifies the percentage of referenced memory to assume it will need kept resident in real memory, to be used for overcommitment checking purposes. It must be an integer value between 1 and 100, optionally followed by a percent sign (%). If this operand is not specified, the current OVERCOMMIT setting is used.

Externals Details – SET RESPOOL



SET RESPOOL ...

STORage NOLIMit
prevents the enforcement of memory limiting for the defined pool.

STOrage EXEMPT
NONEXEMPT
determines if guests in the pool are exempt from memory overcommitment enforcement.

STOrage REFerence pct
pct%
for a guest in the pool, specifies the percentage of defined memory to assume it will reference, to be used for overcommitment checking purposes. It must be an integer value between 1 and 100, optionally followed by a percent sign (%). If this operand is not specified, any previous setting or the OVERCOMMIT default is used.

STOrage RESident pct
pct%
for a guest in the spool specifies the percentage of referenced memory to assume it will need kept resident in real memory, to be used for overcommitment checking purposes. It must be an integer value between 1 and 100, optionally followed by a percent sign (%). If this operand is not specified, any previous setting or the OVERCOMMIT default is used.

Externals Details – QUERY RESPOOL

...

Response 2:

The following example shows the format of the response to QUERY RESPOOL ALL when resource pools are defined. Each line under the heading contains the definition of a resource pool, which includes the name of the pool, the pool limits, the type of CPU resource that is limited, the number of users assigned to the pool, the estimated percentage of memory referenced and resident for pool members, and whether pool members are exempt from overcommitment control.

Pool name	CPU	Type	Storage	Trim	Members	Refer	Resid	Exempt
LINUXP2	8.00 Cores	IFL	NoLimit	----	0	100%	50%	No
LINUXP1	NoLimit	IFL	NoLimit	----	6	30%	50%	Yes
CPPOOL10	12 %	CP	NoLimit	----	8	25%	20%	No
LINUXP3	30 %	IFL	NoLimit	----	20	50%	32%	No

Externals Details – ADRSPACE CREATE

- **ADRSPACE CREATE (Diagnose X'23C' Create-Space Function) will give return code 12 (total address space size exceeds maximum permitted) if creation is predicted to cause paging overflow**
  - **Allows User Directory XCONFIG settings (e.g., for SFS servers) to remain unchanged**
  - **SFS will report inability to create data space and continue operation**

- **If memory overcommitment detected, warning is issued**
  - **Respects authorization and limits configured in User Directory**

Externals Details – Dynamic Storage Reconfiguration

- **DSR functions will be rejected if action is predicted to cause paging overflow**
- **If memory overcommitment detected, warning is issued**
  - **Respects authorization by virtue of User Directory configuration**

External Details – User Directory OPTION Statement

```
>>-Option--...-+------------------+--...
               '-RESPool-poolname-'
```

...

RESPool poolname

specifies the name of a resource pool to which the virtual machine is to be assigned.

The RESPOOL option is not allowed in a subconfiguration entry.

...

User Directory OPTION Statement ...

- **DIRMAINT SETOPTN command supports RESPOOL operand (Query, ADD, CHANGE, DELETE)**
- **SMAPI IMAGE_ACTIVATE API provides more specific reason codes for activation failures, including memory overcommitment**

Externals Details – Message HCP054E (new variation)

```
HCP054E

The variations of this message are explained below.

...

-------------------------------------------------------------------------
- Variation 4

LOGON not permitted because memory overcommitment level would be exceeded.

Explanation:

Meeting the estimated memory requirements of the user you are logging on to
would cause the acceptable system memory overcommitment level to be exceeded.

System action:

The command is not executed; system operation continues.
```

Message HCP054E ...

User response:

Try to log on again specifying a smaller storage size using the STORAGE option of the LOGON command or after waiting for system resource demand to decrease. If the problem persists, contact your system administrator to determine if Memory overcommitment settings are appropriate or if the user's memory size should be adjusted.

Externals Details – Message HCP059E (new failure reason)

```
HCP059E        {XAUTOLOG | AUTOLOG} failed for userid - {IPL missing | IPL
               failed | accounting errors | incorrect password |
               Memory overcommitment level would be exceeded }
...
```

Memory overcommitment level would be exceeded	Satisfying the memory required by the user being logged on would cause memory over-commitment beyond the level that is allowed	Contact your system administrator to determine if memory overcommitment settings are appropriate or if the user's memory size should be adjusted. For XAUTOLOG, reissue the command specifying a smaller storage size via the STORAGE option or use the FORCE option to override the overcommitment limit.

Externals Details – Message HCP991I (new)

HCP991I

User *userid command|function* may cause memory overcommitment.

Explanation:

The *userid* was the subject of a *command* - (X)AUTOLOG, LOGON, XAUTOLOG FORCE, VMRELOCATE or VMRELOCATE FORCE – or invoked a *function* - ADRSPACE CREATE or Dyn Mem Reconfig (Dynamic Memory Reconfiguration) - whose estimated (for a new user or memory redefinition) or current (for a relocation) memory footprint (in memory and on auxiliary storage) could cause memory overcommitment to exceed its acceptable level. Allowing the command or function to complete could result in degraded system performance.

Message HCP991I ...

System action:

XAUTOLOG FORCE and VMRELOCATE FORCE commands will always continue processing. If there is insufficient auxiliary storage, ADRSPACE CREATE functions will terminate with return code 12, which indicates that the memory limit has been exceeded, and Dynamic Memory reconfiguration functions will be rejected. Otherwise, depending on system overcommitment settings the command or function will either continue processing or be terminated.

Operator response:

Additional operator console messages will indicate whether a command has continued processing. If it has, be aware that performance may be affected and might require remedial action to be taken, such as increasing system memory dynamically, adding paging capacity, or relocating guests to other systems.

Externals Details – Message HCP1810I (new variation)

HCP1810I

The variations of this message are explained below.

Explanation:

...

5. *userid*: Storage use (*size*) would cause excess memory overcommitment on destination (*system*)

...

Variation 5:

The current storage footprint of this virtual machine (central storage, and auxiliary storage) would cause the memory overcommitment on the destination system to exceed the acceptable level. Allowing such a virtual machine to relocate could result in degraded system performance.

Message HCP1810I ...

System action:

If the VMRELOCATE command FORCE STORAGE option was used then the relocation continues. Otherwise, it is terminated with no further action and the virtual machine continues to run on the source system.

User response:

Contact your system administrator to determine if memory overcommitment settings on the destination system are appropriate or reissue the VMRELOCATE command and specify the FORCE STORAGE option to override the overcommitment limit. Alternatively, take steps on the destination system to reduce the system load or increase paging capacity, or relocate this virtual machine to a different system with adequate capacity in the SSI cluster.

Externals Details – Message HCP2541E (new)

HCP2541E New OVERCOMMIT settings would cause excess memory overcommitment.

Explanation:

An OVERCOMMIT command was issued but would cause memory overcommitment limits to be exceeded or paging space to be exhausted.

System action:

The memory overcommitment settings remain unchanged.

User response:

Several actions can be taken to allow the setting to be increased:

1. **Decrease virtual machine memory sizes to alleviate the demand for paging space.**
2. **Increase the amount of available paging space by adding appropriately formatted volumes to the system-owned list.**
3. **Increase the amount of real memory available to the z/VM system.**

Externals Details – Monitor Record

```
MRSTOOVC Prolog
 DSECT NAME - STOOVC
 FUNCTION - Map a Monitor record.
 LOCATED By -
        Through the Monitor Control Area
 REFERENCED CONTROL BLOCKS -
        MRRECHDR    Monitor Record Header
 NAME - MRSTOOVC
 DESCRIPTIVE NAME - Monitor Event Record
                    Domain 3 - Storage Domain
                    Record 28 – Memory Overcommitment Check Result
 DESCRIPTION - Generated whenever memory overcommitment checking is
               performed
 NOTES:
```

Monitor Record ...

MRSTOOVC Control Block Contents

Offsets Dec	Hex	Type	Len	Name (Dim)	Description
0	0	Structure	xxx	STOOVC	Start of monitor record
0	0	Character	0	STOOVC_MRHDR	Record header - see MRRECHDR for details
0	0	Character	20	MRHDR	
0	0	Unsigned	2	MRHDRLEN	
2	2	Unsigned	2	MRHDRZER	
4	4	Unsigned	1	MRHDRDM	
5	5	Unsigned	1	*	
6	6	Unsigned	2	MRHDRRC	
8	8	Character	8	MRHDRTOD	
16	10	Character	4	*	
20	14	Character	0	MRHDR_END	
20	14	Character	8	STOOVC_ISSUER	Userid that issued the command that initiated overcommitment checking
28	1C	Character	8	STOOVC_VMDUSER	Userid that is the target of the command
36	24	Character	8	STOOVC_LIMPOOL	Associated resource pool name or blank
44	2C	Unsigned	16	STOOVC_CALCMND	Command name

Monitor Record ...

60	3C	Unsigned	2	STOOVC_CALREFPC	Referenced storage percent in effect
62	3E	Unsigned	2	STOOVC_CALRESPC	Resident storage percent in effect
64	40	Unsigned	4	STOOVC_CALESRES	Estimated guest resident storage, in MB
68	44	Unsigned	4	STOOVC_CALESAUX	Estimated guest auxiliary storage, in MB
72	48	Unsigned	4	STOOVC_VMDDPRES	Current estimated guest resident storage, in MB
76	4C	Unsigned	4	STOOVC_VMDPRAUX	Current estimated guest auxiliary storage, in MB
80	50	Unsigned	8	STOOVC_SYSOCPRS	Current system projected resident storage use, in MB
88	58	Unsigned	8	STOOVC_SYSOCAUX	Current system projected auxiliary storage use, in MB
96	60	Unsigned	8	STOOVC_CALRSRVD	Current unclaimed reserved storage, in pages
104	68	Unsigned	4	STOOVC_CALOVERC	Overcommitment level allowed

Monitor Record ...

108	6C	Bitstring	1	STOOVC_CALFLAGS	Overcommitment flags
		1...		STOOCV_SYSOCPRE	Prevent
		.1..		STOOVC_SYSOCWRN	Warn
		00..			If neither SYSOCPRE nor SYSOCWRN is set then the action is Allow
		..1.		*	
		...1		*	
		.... 1...		*	
		.... .1..		STOOVC_CALPGOVF	Allowing command would overflow paging space
		.... ..1.		STOOVC_CALEXEMP	User exempt from overcommitment control
		1		STOOVC_CALNOOVC	No overcommitment detected
109	6D	Char	3	*	Reserved for IBM use
112	70	Char	0	STOOVC_END	

Externals Details – VM Event Message

Class	Type	Event	Data
0	30	Memory Overcommitment	The following table shows the offset and length of the data elements. The values are decimal numbers of bytes.

Offset	Length	Element
0	8	User ID
8	8	Subject User ID
16	8	Resource pool name
24	1	Function code
25	1	Action flag

VM Event Message ...

The data elements are as follows:

User ID
Character string, 8 bytes. The ID of the user who invoked the function.

Subject user ID
Character string, 8 bytes. The ID of the user causing memory overcommitment.

Resource pool name
Character string, 8 bytes. The name of the resource pool of which the user is a member.

VM Event Message ...

Function code

1-byte unsigned binary integer indicating the function being performed, as follows:

Value	Function
0	AUTOLOG
1	LOGON
2	XAUTOLOG FORCE
3	VMRELOCATE
4	VMRELOCATE FORCE
5	DEFINE STORAGE
6	ADRSPACE CREATE
7	Dyn Mem Reconfig

VM Event Message ...

Action flag
1-byte hexadecimal value indicating action to be taken, as follows:

Value	Meaning
X'80'	Prevent overcommitment
X'40'	Warn of overcommitment
X'00'	Allow overcommitment
X'04'	Allowing function would overflow paging space
X'02'	User exempt from overcommitment action
X'01'	No overcommitment detected

Memory Overcommitment Control – Delivery

- **Delivered as a technology preview as part of z/VM 7.5**