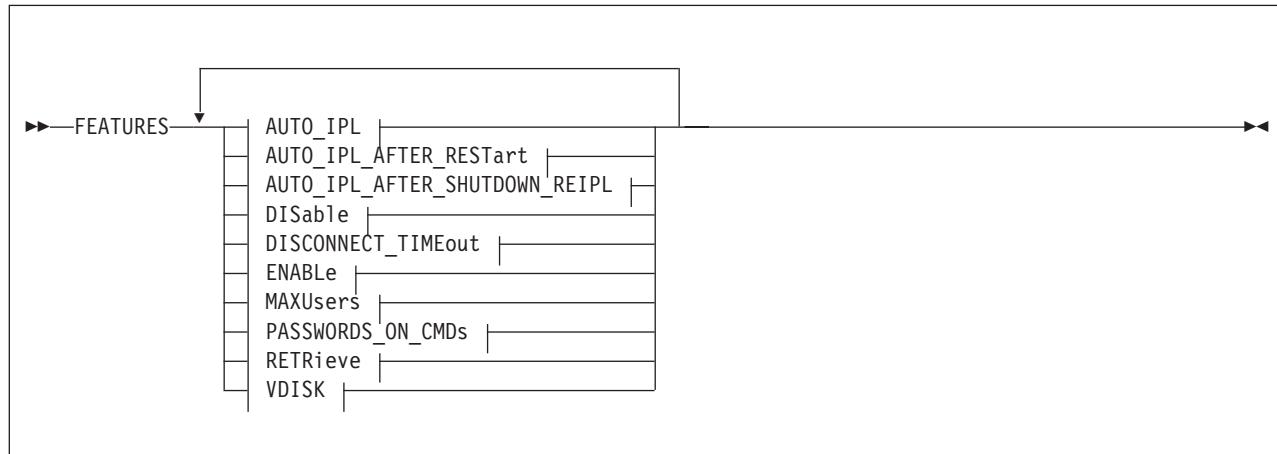
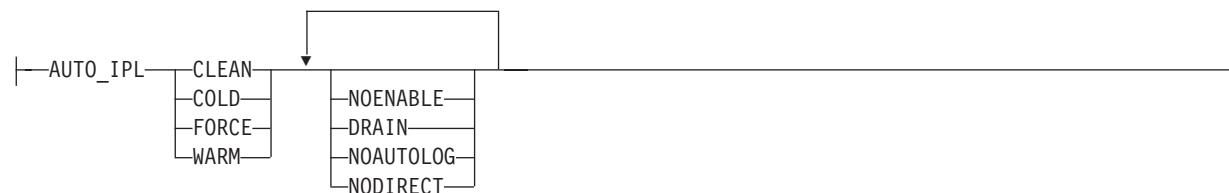


## FEATURES

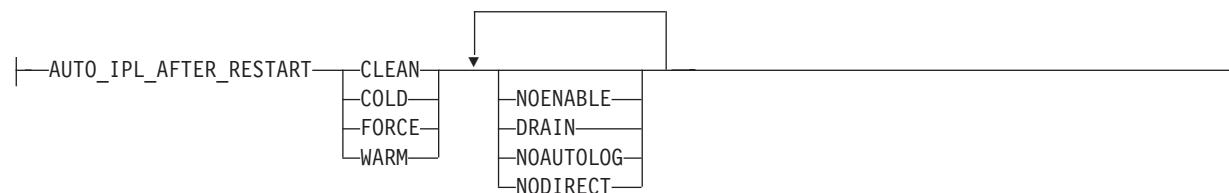
### FEATURES Statement

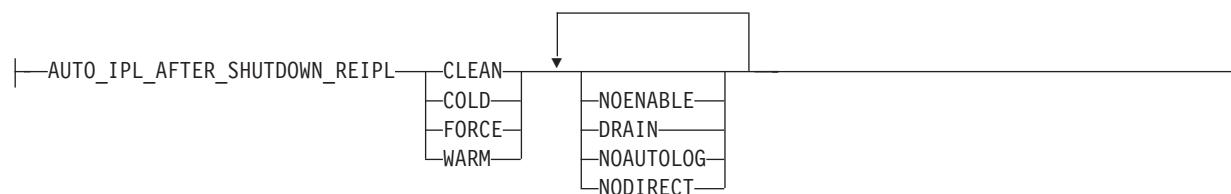
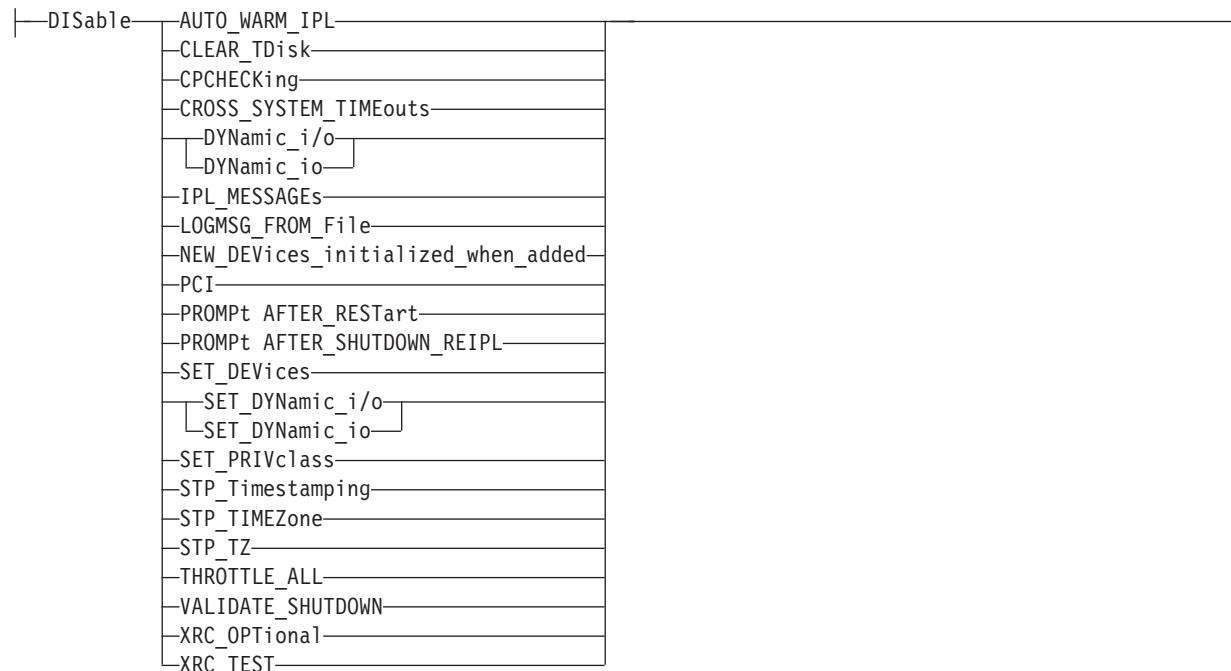
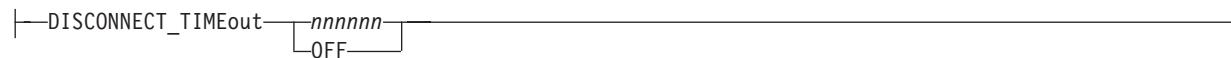


#### AUTO\_IPL:



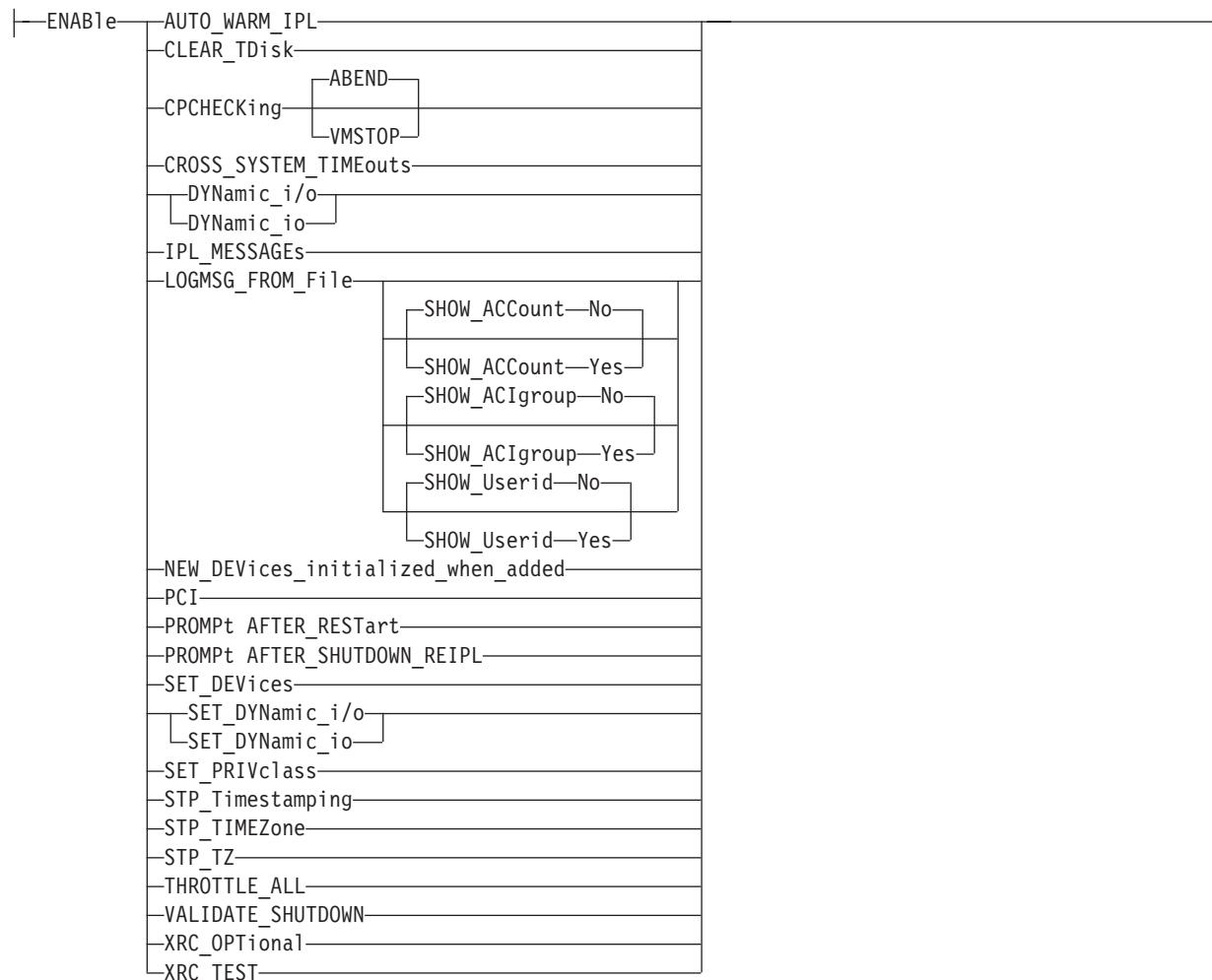
#### AUTO\_IPL\_AFTER\_RESTART:



**AUTO\_IPL\_AFTER\_SHUTDOWN\_REIPL:****DISable:****DISCONNECT\_TIMEout:**

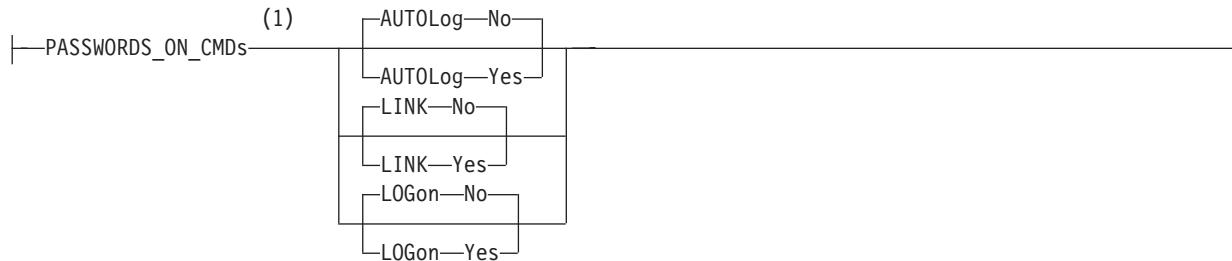
## FEATURES

### ENABLE:



### MAXUsers:

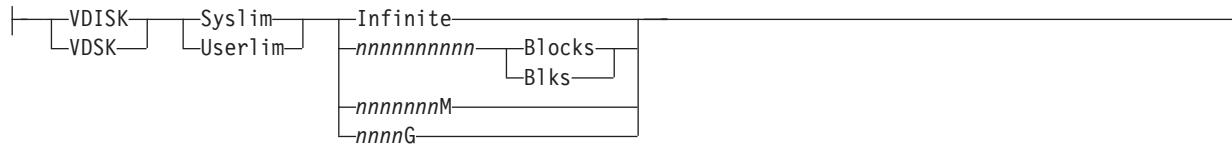


**PASSWORDS\_ON\_CMDS:****Notes:**

- 1 You must specify at least one of the operands.

**RETRieve:****Notes:**

- 1 You must specify at least one of the operands.

**VDISK:****Purpose**

Use the FEATURES statement to set certain attributes of the system at system initialization.

**How to Specify**

Include as many statements as needed; they are optional. You can place FEATURES statements anywhere in the system configuration file. If you specify more than one statement with the same operands, the last operand definition overrides any previous specifications.

## FEATURES

### Operands

#### **AUTO\_IPL**

defines the kind of start to be performed when the system is initialized. The system will be started without prompting the system operator and without changing the TOD clock.

#### **AUTO\_IPL\_AFTER\_RSTArt**

defines the kind of start to be performed when the system is initialized after a system restart. The system will be restarted without prompting the system operator and without changing the TOD clock

#### **AUTO\_IPL\_AFTER\_SHUTDOWN\_REIPL**

defines the kind of start to be performed when the system is initialized after a SHUTDOWN REIPL. The system will be restarted without prompting the system operator and without changing the TOD clock.

#### **COLD**

tells CP to perform a cold start. This will purge all spool files, accounting records, error recording records, symptom records, and the system log message. System data files will not be purged.

If spooling errors that could result in the loss of system data files (NSS, DCSS, TRF, IMG, UCR, NLS) are encountered, the system operator will be prompted and given the opportunity to stop. If no spooling errors are encountered, the system operator will not be prompted and there will be no opportunity to stop.

#### **CLEAN**

tells CP to perform a clean start. This will purge all spool files, system data files, accounting records, error recording records, symptom records, and the system log message. The system operator will not be prompted and there will be no opportunity to stop.

#### **FORCE**

tells CP to perform a force start. If spooling errors are encountered, the spool files and system data files in error will be purged. The system operator will not be prompted and there will be no opportunity to stop.

#### **WARM**

tells CP to perform a warm start. If spooling errors are encountered, the system operator will be prompted and given the opportunity to stop.

#### **NOENABLE**

tells CP not to enable terminal access after system initialization.

#### **DRAIN**

tells CP to drain unit-record devices after system initialization.

#### **NOAUTOLOG**

tells CP to bypass automatic logon of virtual machines after system initialization.

#### **NODIRECT**

tells CP to bring up the system without a User Directory.

#### **DISable**

disables the following system attributes during IPL. Note that initially each option is disabled until enabled using the ENABLE operand.

#### **AUTO\_WARM\_IPL**

tells CP to go through all the usual prompts in the IPL process.

**CLEAR\_TDisk**

tells CP to clear only cylinder 0 or the first eight blocks on the temporary minidisk when it detaches the minidisk.

**CPCHECKing**

tells CP that internal CP checking is not to be executed.

**CROSS\_SYSTEM\_TIMEOUTs**

specifies that timeouts detected at the device and logical link level within the ISFC inter-system communications component are ignored. Also, in an SSI cluster environment, CROSS\_SYSTEM\_TIMEOUTS specifies that missing or late heartbeat signals between members in the SSI cluster are ignored. Disabling CROSS\_SYSTEM\_TIMEOUTS prevents the SSI member from recognizing unresponsive member systems and from initiating normal error recovery.

This option is intended for use in diagnostic situations only—typically, when running members of an SSI cluster as z/VM guests, in which a member might be stopped for long periods of time when using the CP TRACE command or similar facilities to debug code within the CP nucleus or a CPXLOADED nucleus extension. Without this option in effect, manual commands such as SET SSI DOWN, DEACTIVATE ISLINK, and ACTIVATE ISLINK might be required to restore proper operation.

In order to join an SSI cluster, a member's CROSS\_SYSTEM\_TIMEOUTS setting (either ENABLE CROSS\_SYSTEM\_TIMEOUTS or DISABLE CROSS\_SYSTEM\_TIMEOUTS) must be the same as that of the first member who joined; a nonconforming member is not permitted to join the cluster.

**DYNAMIC\_i/o****DYNAMIC\_io**

tells CP not to allow dynamic I/O changes on this processor.

**IPL\_MESSAGES**

tells CP not to display IPL or SHUTDOWN messages or prompts during system initialization.

If IPL\_MESSAGES are disabled and spooling errors are encountered during an automatic WARM or COLD IPL, wait state HCP2516W will be issued.

**LOGMSG\_FROM\_File**

tells CP not to look for any LOGMSG files on disk. Instead, CP should use information from the class B CP SET LOGMSG command. For more information on the CP SET LOGMSG command, see the *z/VM: CP Commands and Utilities Reference*.

**NEW\_DEVICES\_initialized\_when\_added**

tells CP to create a real device control block (RDEV) for a new I/O device, but not to initialize (bring online) that device when your system receives an I/O machine check (IOMCK) for adding a new device to the system. To bring the device online, use the CP VARY (Real Device) command. For more information about the CP VARY (Real Device) command, see the *z/VM: CP Commands and Utilities Reference*.

**PCI**

tells CP to not allow any PCI functions to come online to the system for use. This is the default.

**PROMPT\_AFTER\_RSTArt**

tells CP not to force a prompt to the operator when CP bounces.



## FEATURES

### **PROMPt AFTER\_SHUTDOWN\_REIPL**

tells CP not to force a prompt to the operator when CP is performing a SHUTDOWN REIPL.

### **SET\_DEVICES**

tells CP not to allow users to execute CP SET DEVICES commands to change the way CP handles specific real devices after initialization. For more information about the CP SET DEVICES command, see the *z/VM: CP Commands and Utilities Reference*.

### **SET\_DYNAMIC\_i/o**

### **SET\_DYNAMIC\_io**

tells CP not to allow users to execute CP SET DYNAMIC\_I/O commands to enable or disable CP's ability to dynamically change the processor's I/O configuration after initialization. For more information about the CP SET DYNAMIC\_I/O command, see the *z/VM: CP Commands and Utilities Reference*.

### **SET\_PRIVclass**

tells CP not to give end users the authority to use the CP SET PRIVCLASS command to change their own privilege classes, and tells CP not to give the system operator the authority to use the CP SET PRIVCLASS command to change the privilege classes of users logged on to the system. For more information about the CP SET PRIVCLASS command, see the *z/VM: CP Commands and Utilities Reference*.

### **STP\_Timestamping**

tells CP not to enable the STP protocol for timestamping purposes.

### **STP\_TIMEZone / STP\_TZ**

tells CP not to enable the STP protocol in order to obtain timezone information automatically from the STP server.

### **THROTTLE\_ALL**

tells CP to allow throttling of ALL devices on the system except CP OWNED DASD.

### **VALIDATE\_SHUTDOWN**

tells CP that the SYSTEM operand on the SHUTDOWN command is not required. This is the default.

### **XRC\_OPTIONal**

tells CP not to allow non-timestamped I/O to be issued whenever STP is in an unsynchronized state.

### **XRC\_TEST**

tells CP not to timestamp I/O when CP is running as a virtual machine guest.

### **DISCONNECT\_TIMEOUT nnnnnn**

sets the interval between a forced disconnect of a virtual machine and its logoff to the specified number of minutes. The default is 15 minutes.

### **DISCONNECT\_TIMEOUT OFF**

disables the automatic logoff of a virtual machine that is forcibly disconnected.

### **ENABLE**

enables the following system attributes during IPL. Note that initially each option is disabled until it is enabled.

### **AUTO\_WARM\_IPL**

tells CP to attempt a warm start without changing the TOD clock. If there

is no warm start data, if the TOD clock is not set, or if spooling errors are encountered, CP will prompt the operator for more information.

#### **CLEAR\_TDIsK**

tells CP to automatically clear all previously written data and directory areas on TDISK DASD space. CP will change TDISK DASD space to binary zeros during CP initialization, when attaching a CP-owned volume that contains TDISK allocations, and when a user detaches a TDISK minidisk. By clearing this space, you prevent users from accidentally accessing old temporary disk space and provides your system with additional security.

#### **CPCHECKing**

tells CP to activate internal CP checking to confirm assertions established in the CP code and take the appropriate action based on whether ABEND or VMSTOP is specified.

#### **ABEND**

Indicates that an abend should occur. Whether it is a hard abend or a soft abend is controlled by the specific assertion case. It is typically a hard abend. This is the default when internal CP checking is activated.

#### **VMSTOP**

This parameter is permitted only when CP is itself running in a virtual machine, and is generally useful only for debugging CP. Specifying this parameter causes CP to issue DIAGNOSE code X'8' when an untrue assertion is encountered, specifying a command string length of zero. This causes the virtual machine to stop and for CP to post a read to the console. A message is sent to the console (using DIAGNOSE code X'8' and the CP MESSAGE \* command) providing information about the cause of the stop.

#### **CROSS\_SYSTEM\_TIMEOUTs**

specifies that timeouts caused by device and logical link failures within the ISFC inter-system communications component are treated as errors. Also, in an SSI cluster environment, CROSS\_SYSTEM\_TIMEOUTS causes error processing to occur for missing or late heartbeat signals between members.

This option is the default setting and is what should be used in a production environment in order to ensure that the ISFC collection or SSI cluster reacts properly to network and member failures.

In order to join an SSI cluster, a member's CROSS\_SYSTEM\_TIMEOUTS setting (either ENABLE CROSS\_SYSTEM\_TIMEOUTS or DISABLE CROSS\_SYSTEM\_TIMEOUTS) must be the same as that of the first member who joined; a nonconforming member is not permitted to join the cluster.

#### **DYNamic\_i/o**

#### **DYNamic\_io**

tells CP to allow dynamic I/O changes on this processor.

#### **IPL\_MESSAGES**

tells CP to display IPL or SHUTDOWN messages and prompts during system initialization (the default).

#### **LOGMSG\_FROM\_File**

tells CP to display the contents of the SYSTEM LOGMSG file on the lowest accessed CP disk. You can optionally choose to have CP read and display additional files at a user's terminal using one or more of the following operands:

## FEATURES

### **SHOW\_ACCOUNT No**

(the default) tells CP not to display a file called *accountid* LOGMSACC (*accountid* is the account ID of a specific user) in response to a CP QUERY LOGMSG command.

### **SHOW\_ACCOUNT Yes**

tells CP to display a file called *accountid* LOGMSACC (*accountid* is the account ID of a specific user) in response to a CP QUERY LOGMSG command.

### **SHOW\_ACIGROUP No**

(the default) tells CP not to display a file called *acigroup* LOGMSACI (*acigroup* is the ACI group of a specific user) in response to a CP QUERY LOGMSG command.

### **SHOW\_ACIGROUP Yes**

tells CP to display a file called *acigroup* LOGMSACI (*acigroup* is the ACI group of a specific user) in response to a CP QUERY LOGMSG command.

### **SHOW\_USERID No**

(the default) tells CP not to display a file called *userid* LOGMSUSR (*userid* is the user ID of a specific user) in response to a CP QUERY LOGMSG command.

### **SHOW\_USERID Yes**

tells CP to display a file called *userid* LOGMSUSR (*userid* is the user ID of a specific user) in response to a CP QUERY LOGMSG command.

For more information about the CP QUERY LOGMSG command, see the *z/VM: CP Commands and Utilities Reference*.

### **NEW\_DEVICES\_initialized\_when\_added**

tells CP to automatically create a real device control block (RDEV) and initialize (bring online) the associated I/O device when you add a new device, causing an I/O machine check (IOMCK).

### **PCI**

tells CP to allow PCI functions to come online to the system for use.

### **PROMPT\_AFTER\_RESTART**

tells CP to force a prompt to the operator when CP bounces, so that REIPL can be stopped or the type of start desired (WARM, FORCE, etc.) can be specified.

### **PROMPT\_AFTER\_SHUTDOWN\_REIPL**

tells CP to force a prompt to the operator when CP is performing a SHUTDOWN REIPL, so that REIPL can be stopped or the type of start desired (WARM, FORCE, etc.) can be specified.

Note that in this case, the user will not be prompted to change the Time of Day (TOD) clock.

### **SET\_DEVICES**

tells CP to allow users to execute CP SET DEVICES commands to change the way the way CP handles specific real devices after initialization. For more information about the CP SET DEVICES command, see the *z/VM: CP Commands and Utilities Reference*.

### **SET\_DYNAMIC\_I/O**

### **SET\_DYNAMIC\_IO**

tells CP to allow users to execute CP SET DYNAMIC\_I/O commands to

enable or disable CP's ability to dynamically change the processor's I/O configuration after initialization. For more information about the CP SET DYNAMIC\_I/O command, see the *z/VM: CP Commands and Utilities Reference*.

#### **SET\_PRIVclass**

tells CP to give end users the authority to use the CP SET PRIVCLASS command to change their own privilege classes, and tells CP to give the system operator the authority to use the CP SET PRIVCLASS command to change the privilege classes of users logged on to the system. For more information about the CP SET PRIVCLASS command, see the *z/VM: CP Commands and Utilities Reference*.

#### **STP\_Timestamping**

tells CP to enable the STP protocol (if the STP facility is installed) and apply timestamps to all XRC-capable DASD devices.

#### **STP\_TimeZone / STP\_TZ**

tells CP to enable the STP protocol (if the STP facility is installed) and obtain timezone information automatically from the STP server.

#### **THROTTLE\_ALL**

tells CP to allow throttling of ALL devices on the system including CP OWNED DASD.

#### **VALIDATE\_SHUTDOWN**

tells CP to require the user to specify the SYSTEM operand on the SHUTDOWN command.

#### **XRC\_OPTIONal**

when STP\_Timestamping is also enabled, this will allow non-timestamped I/O to be issued whenever STP is in an unsynchronized state, as opposed to deferring I/O until STP synchronization completes.

#### **XRC\_TEST**

tells CP to timestamp I/O regardless of STP availability. This option is meant only for vendor and testing purposes, and can only be specified for systems running within a virtual machine.

#### **MAXusers NOLimit**

(the default) tells CP that the number of users who can log on at one time is not limited by the installation. Therefore, the system-defined limit of 99,999 logged on users is the maximum number allowed.

#### **MAXusers nnnnn**

defines the maximum number of users who can log on to the system at one time. The variable *nnnnn* is a decimal number from 1 through 99999.

#### **PASSWORDS\_ON\_CMDS**

tells CP whether to accept passwords in the command syntax (in clear text) when users issue the CP AUTOLOG, XAUTOLOG, LINK, or LOGON commands. If the setting is NO, CP accepts the command only without the password, then prompts the user for the password and masks the input field.

#### **AUTOLog No**

(the default) tells CP to not accept passwords entered by users who are issuing the CP AUTOLOG or XAUTOLOG command.

#### **AUTOLog Yes**

tells CP to accept passwords entered by users who are issuing the CP AUTOLOG or XAUTOLOG command. For AUTOLOG, users *must* enter the password as part of the command.

## FEATURES

### **LINK No**

(the default) tells CP to not accept passwords entered by users who are issuing the CP LINK command.

### **LINK Yes**

tells CP to accept passwords entered by users who are issuing the CP LINK command.

### **LOGon No**

(the default) tells CP to not accept passwords entered by users who are issuing the CP LOGON command.

### **LOGon Yes**

tells CP to accept passwords entered by users who are issuing the CP LOGON command.

For more information about the CP AUTOLOG, XAUTOLOG, LINK, and LOGON commands, see the *z/VM: CP Commands and Utilities Reference*.

### **RETRieve**

defines the default and maximum number of retrieve buffers allowed per user on your system. These numbers determine how many console input lines a user can retrieve. Before retrieving a buffer, users must define a program function (PF) key as a retrieve key, using the CP SET PF<sub>nnn</sub> RETRIEVE command. After defining a retrieve key, users can press that PF key when they want to retrieve a command that they issued previously. For more information about the CP SET PF<sub>nnn</sub> RETRIEVE command, see the *z/VM: CP Commands and Utilities Reference*.

### **DEFault nnn**

tells CP to define *nnn* default retrieve buffers per user on your system. The variable *nnn* is a decimal number from 0 to 255. If omitted, the default is 7.

**Note:** The number of default retrieve buffers must be less than or equal to the number of maximum retrieve buffers.

### **MAXimum nnn**

tells CP to define *nnn* maximum retrieve buffers per user on your system. The variable *nnn* is a decimal number from 0 to 255. If omitted, the default is 7.

### **VDISK**

### **VDSK**

defines installation defaults for the system and user limits on the maximum amount of host storage available for allocation as virtual disks in storage. If an installation default is not defined, CP uses a built-in default.

You can supersede the installation or built-in default by using the CP SET VDISK command to set the current system limit or user limit. The DEFAULT operand on the SET VDISK command resets the current limit to the installation default, or, if none is defined, to the built-in default. The CP QUERY VDISK command displays current and default system and user limits. For more information about the CP SET and QUERY VDISK commands, see the *z/VM: CP Commands and Utilities Reference*.

**Note:** Use of virtual disks in storage increases the load on system paging, so you should set limits in proportion to the availability of paging space.

**Syslim**

sets the total resource available for allocating virtual disks in storage on the system.

If an installation default is not defined, CP calculates the built-in default from the available host storage. It takes 2050 non-pageable pages of page and segment data tables to support each gigabyte of address space for virtual disks in storage. The limit is set so these non-pageable structures can consume no more than 1/4 of the DPA pages. The limit is further reduced, if necessary, so the virtual storage pages used for address spaces for virtual disks in storage can consume no more than 1/4 of the total available paging space.

**Userlim**

sets the maximum resource available for virtual disks in storage created by a single user using the CP DEFINE command. This limit does not apply to virtual disks in storage defined by MDISK statements in the directory. If an installation default is not defined, the built-in default is 0.

**Infinite**

indicates that there is no limit. All available host storage may be allocated to virtual disks in storage.

*nnnnnnnnn Blocks*

*nnnnnnnnn Blks*

specifies the number of 512-byte blocks of storage available for virtual disks in storage. If the number specified is equal to or greater than 2147483648, the limit is set to INFINITE.

*nnnnnnnnM*

specifies the number of megabytes of storage available for virtual disks in storage. If the number specified is equal to or greater than 1048576, the limit is set to INFINITE.

*nnnnG*

specifies the number of gigabytes of storage available for virtual disks in storage. If the number specified is equal to or greater than 1024, the limit is set to INFINITE.

**Usage Notes**

1. If an IODF statement is defined in the system configuration file, then the hardware I/O configuration will be controlled by HCD. In this case, the following FEATURES statements are ignored if they are specified:

- FEATURES ENABLE DYNAMIC\_I/O
- FEATURES ENABLE DYNAMIC\_IO
- FEATURES ENABLE SET\_DYNAMIC\_I/O
- FEATURES ENABLE SET\_DYNAMIC\_IO
- FEATURES DISABLE DYNAMIC\_I/O
- FEATURES DISABLE DYNAMIC\_IO
- FEATURES DISABLE SET\_DYNAMIC\_I/O
- FEATURES DISABLE SET\_DYNAMIC\_IO

If the *osconfig* parameter is specified on the IODF statement, then the software I/O configuration will be controlled by HCD. In this case, the following FEATURES statements are ignored if they are specified:

- FEATURES ENABLE SET\_DEVICES
- FEATURES DISABLE SET\_DEVICES

## FEATURES

Refer to “IODF Statement” on page 165 for more information.

2. FEATURES ... SET\_PRIVCLASS sets the parameters for the CP SET PRIVCLASS command, which general users can use to change their privilege classes to be all or a subset of the classes specified in their virtual machine definitions. For more information about the CP SET PRIVCLASS command, see the *z/VM: CP Commands and Utilities Reference*.
3. At logon, CP gives each user enough storage for the default number of retrieve buffers. If users want to retrieve more than the default number of console input lines, they can use the CP SET RETRIEVE command to increase their number of retrieve buffers to some number less than or equal to the maximum number of retrieve buffers that you are defining for the system using this FEATURES statement. To find out the maximum number of retrieve buffers allowed on a running system, users can issue the CP QUERY RETRIEVE command. For more information about the CP QUERY and SET RETRIEVE commands, see the *z/VM: CP Commands and Utilities Reference*.
4. If you enable AUTO\_WARM\_IPL and no warm start data has been saved, or if invalid warm start data is encountered, CP will only ask the operator whether it should try a FORCE start instead.
5. If you enable AUTO\_WARM\_IPL in the system configuration file and you want to go through the full series of prompts during the IPL, you can specify the PROMPT keyword as part of the IPL parameters on the Stand-Alone Program Loader (SAPL). For more information, see “Passing IPL Parameters” in *z/VM: System Operation*.
6. To change whether new devices are initialized when they are added to the system after IPL, use the CP SET NEW\_DEVICES command. For more information about the SET NEW\_DEVICES command, see the *z/VM: CP Commands and Utilities Reference*.
7. To allow users to dynamically change a device's I/O configuration after IPL, you must specify the ENABLE DYNAMIC\_I/O operand on the FEATURES statement in the system configuration file or you must issue the CP SET DYNAMIC\_I/O ON command after IPL. If you do not turn this function on either during or after IPL, CP will reject all attempts to dynamically change the I/O configuration. This means that CP rejects the following dynamic I/O commands:

DEFINE CHPID	DEFINE CU	DEFINE DEVICE
DELETE CHPID	DELETE CU	DELETE DEVICE
MODIFY CHPID	MODIFY CU	MODIFY DEVICE
DEFINE PATH	DEFINE CNTLUNIT	DEFINE IODEVICE
DELETE PATH	DELETE CNTLUNIT	DELETE IODEVICE
MODIFY PATH	MODIFY CNTLUNIT	MODIFY IODEVICE
QUERY CONFIGMODE		QUERY DYNAMIC_I/O
SET CONFIGMODE	SET DEVICES	SET DYNAMIC_I/O

8. To change whether I/O operations from guest operating systems can be limited (or controlled) after IPL, use the CP SET DEVICES command. For more information about the SET DEVICES command, see the *z/VM: CP Commands and Utilities Reference*.
9. Because the SYSLimit does not apply during system initialization, it is possible to find more LAN segments in the system than the number defined as the system limit.
10. Although the PASSWORDS\_ON\_CMDS operand can be set to prevent passwords from being entered with the commands (in clear text) on the terminal screen, passwords may still be included on the command line by REXX execs using the DIAG() and DIAGRC() functions. If you require access

security without the possibility of passwords being stored in clear text, you should consider installing an external security manager (ESM).

11. In order to use the STP\_Timestamping feature, STP must be set up on the Central Processor Complex (CPC), where the CPC is either a member of an STP-only Collaborative Timing Network (CTN) or a stratum 2 or higher member of a mixed CTN. In order to use the STP\_TIMEZone / STP\_TZ feature, STP must be set up on the CPC, and the CPC must be a member of an STP-only CTN.
12. The ENABLE and DISABLE options control attributes during IPL. This is defined as an IPL from the HMC. These attributes might not apply during CP bounce processing (that is, as initiated by the SHUTDOWN REIPL command, or restarting after an abend). For example, if you disable the AUTO\_WARM\_IPL attribute, you will be prompted for the type of start you want to perform and given the opportunity to change the TOD clock, but only if you have IPLED from the HMC.

## Examples

1. To have CP:
  - Enable all log message support
  - Allow end users and the operator to issue the SET PRIVCLASS command
  - Clear all temporary disk space after users are through with it
  - Allow operators to dynamically change the I/O configuration
  - Prompt the operator for startup information after a software restart (bounce)
  - Allow operators to limit the I/O operations from guest operating systems
  - Initialize new devices when they are added to the system
  - Disable the automatic warm start feature
  - Give users a default of 7 retrieve buffers and let them go up to a maximum of 255 buffers
  - Allow an unlimited number of users to be logged on at one time
  - Prompt users for passwords on the CP AUTOLOG, XAUTOLOG, and LOGON commands, but not the CP LINK command
  - Set the installation default system limit on space for virtual disks in storage to '32GB' and the installation default user limit to 800 blocks

use the following FEATURES statement:

```
----- Features Statement -----
Features,
Enable,
Logmsg_From_File,      /* Allow log messages from files      */
Show_ACCount Yes,      /* If account logmsg exists, show it */
Show_ACIGroup Yes,     /* If acigroup logmsg exists, show it */
Show_Userid Yes,       /* If userid logmsg exists, show it */
Set_Privclass,          /* Let the SET PRIVCLASS command be
                         executed. If this option is disabled,
                         then the SET PRIVCLASS command is not
                         allowed. */
Clear_Tdisk,            /* Clear all temporary disk space after
                         use */
Dynamic_I/O,           /* Allow dynamic I/O changes on this
                         system */

```

## FEATURES

```
Prompt_on_Bounce,      /* If CP bounces, prompt operator for
                           startup conditions */  
  
Throttling,           /* Allow people to limit I/O from guests */  
  
New_Devices_Initialized_when_Added, /* Initialize devices if we get
                                         an IPI I/O machine check */  
  
Disable,  
  
Auto_Warm_Ipl,       /* Do not perform Auto_Warm_IPL's, let
                           operator see the CP start-up message
                           prompts and time message */  
/* Set some additional features ... */  
  
Retrieve,  
  
Default 7,            /* Give each user 7 default retrieve buffers */
Maximum 255,           /* Set the maximum system buffers per user to 255 */  
  
MaxUser NoLimit,      /* Don't set a maximum number of user limit */  
  
Passwords_On_Cmds,  
  
AutoLog No,           /* Prompt user for password on AUTOLOG command */
Link Yes,              /* Do not prompt for password on LINK command */
Logon No,              /* Prompt user for password on LOGON command */  
  
Vdisk,  
  
Syslim 32G,           /* Set total virtual disks in storage to 32GB */
Userlim 800 Blks,      /* User can define max 800 blocks using DEFINE */  
  
/*----- End of Features Statement -----*/
```

---

## Chapter 16. Using PCIe Functions for z/VM Guests

This chapter describes:

- The location of industry standards for PCI functions
- System programmer tasks
- Debug aids
- Recovery from errors.

---

### PCI Industry Standards

PCI functions are IO devices that conform to the PCI bus specification. This specification is controlled by the PCI-SIG. For more information, see PCI SIG ([www.pcisig.com](http://www.pcisig.com)).

---

### z/VM Support

In some ways, PCIe function support on z/VM is similar to channel attached devices such as DASD. This includes defining the function in the IOCDs and the VARY, ATTACH, and QUERY commands. For more information about these commands, see *z/VM: CP Commands and Utilities Reference*.

Guests with a dedicated PCIe function are not eligible for Live Guest Relocation. In addition, if in an SSI environment, a guest will only be able to use PCIe functions if they are in a singleton domain or a domain where the appropriate PCIe facilities hardware is available on all members in the domain and PCI is enabled via the FEATURES ENABLE PCI statement in the system configuration file on all members in the domain. For additional details, see “Using Relocation Domains” on page 756.

A list of PCI function supported by z/VM can be found in *z/VM: General Information*.

---

### Steps to Using a PCIe Function on VM

The following should be done on a test system first. This will allow tuning the system configuration file statements. Refer to *z/VM: CP Commands and Utilities Reference*, for more information on the listed commands.

1. Plug in the PCIe function cards.
2. Define the PCIe functions in the IO gen, either dynamically or via HCD/IOCDs. See “PCIe Functions Defined in IOCP” on page 444.
3. Enable PCI support in the system configuration file. See “FEATURES Statement” on page 142.
4. Define the IOAT subpool in the system configuration file. See “STORAGE Statement” on page 254.
5. Define storage locking warning and fail percentages in the system configuration file using the STORAGE statement.
6. IPL the VM system.
7. Verify the expected PCI functions are available with the QUERY PCIFUNCTION command.
8. Verify the IOAT subpool settings with the QUERY FRAMES command.

## Setting Up PCIe

9. Verify the storage locking and warning settings with the QUERY FRAMES command.
10. Log on the guests that are using PCIe functions
11. Attach PCIe functions to guests with the ATTACH command. These can be included in the user directory.
12. Verify the PCIe functions are attached to the guests with the QUERY VIRTUAL PCIFUNCTION command.
13. IPL the guest operating system. For more information, see the IPL command.
14. Start the applications in the guest operating system that are using the PCIe functions.
15. Watch the usage of the IOAT subpool and the locked frame percentage. This is done with the QUERY FRAMES command and PERFKIT. For more information, see *z/VM: CP Commands and Utilities Reference* and *z/VM: Performance Toolkit Reference*.
16. The IOAT usage will increase as a guest begins using a function. The usage for that function will level out and hold until the tables are reregistered. If, after putting another function into use, the IOAT subpool is 100% used, then the customer will need to increase the subpool size to enable the long term use of that number of functions with those guests.
17. Adjust the storage locking warning and fail percentages set in step 4 based on usage. See “Protecting VM from Excessive Memory Use” on page 445 for guidelines. Tell the VM lab what the results of your testing are so that publications can be improved.

---

## PCIe Functions Defined in IOCP

PCI functions are dedicated reconfigurable entities. This means that a PCI function can only be configured to one system at a time, but multiple systems can be specified as having potential access to the PCI function. In the IOCP one or no systems can be specified as having initial access to the PCI function and one or multiple systems can be specified as having access to the PCI function (Refer to *System z: Input/Output Configuration Program User's Guide for ICP IOCP*, SB10-7037). If a system is specified as having initial access, then that system will see that PCI function in the configured (online) state (see QUERY PCIFUNCTION command) while the other systems will not see the PCI function at all. If there are no systems specified as having initial access, then all the systems in the access list will see the PCI function in the standby (offline) state (see QUERY PCIFUNCTION command). In this case, the first system that varies the PCI function online (see VARY PCIFUNCTION command) will see it in the configured state, while all other systems will no longer be able to see the PCI function. If the first system varies off that PCI function, then all system with access to that PCI function will now see it in standby (offline) state.

Refer to *z/VM: CP Commands and Utilities Reference* for more information on these commands.

---

## Dynamic I/O and PCI Functions

PCI function's can be dynamically defined/modified/deleted via CP's dynamic I/O commands:

- DEFINE PCIFUNCTION
- MODIFY PCIFUNCTION
- DELETE PCIFUNCTION

Refer to *z/VM: CP Commands and Utilities Reference* for information on these commands and *z/VM: I/O Configuration* for general dynamic I/O information.

You can also use zVM HCD/HCM support to dynamically define/modify/delete PCI functions. For more information, see *z/VM: I/O Configuration* and *z/OS and z/VM: Hardware Configuration Manager User's Guide*.

**Note:** In order not to lose dynamic I/O changes, your IOCP should be updated with the dynamic changes that were made, see *IOCP User's Guide for ICP, IOCP, IBM System z*. The new IOCP should be made active via the SET IOCDS active command.

## Protecting VM from Excessive Memory Use

PCIe functions are very different from other of z/Architecture IO devices. Data is transferred directly from the function to guest memory using Dynamic Memory Access (DMA). The application code and device driver running in the guest control how data is transferred; z/VM has no control.

The number of active PCIe functions will have to be limited to maintain acceptable system operation. This will depend on the type of function and the system load. Every installation will have to experiment to determine how best to tune the system.

When a large amount of storage is locked for PCIe data transfer, the size of the Dynamic Paging Area (DPA) is reduced, resulting in less real storage available for use by other guests. You should take this reduction in DPA into account when calculating the total system workload capacity and storage over-commitment targets. A small DPA can negatively affect overall system performance and possibly result in a system outage when insufficient memory is available for system use.

If your system is storage over-committed, it may also be useful to have a SET RESERVED value in effect for the guests using the PCIe functions, particularly during initialization of the PCIe application. The “SET RESERVED command” is used to specify the amount of storage a guest is to have resident. The specific guest pages contained within its reserved frame allotment is dependent on the reference pattern of the guest. If the pages to be locked by the PCIe application are not resident, they must be paged in during the locking operation. This can result in a noticeable delay. Having a reserved setting may improve the performance of the lock operation if the pages are already resident in real storage. However, because SET RESERVED restricts the amount of frames that can be stolen from a guest by CP's frame replenishment algorithm, overuse of the command can also negatively affect overall system performance. For more information, see *z/VM: CP Commands and Utilities Reference*.

Another tuning task is to determine the fail percent to specify in the system configuration file. If it is set too high, the response time for guests will be affected too much or the system could abend. If too low, guest PCIe function use will be stopped when it was not necessary. It may be appropriate to deliberately cause problems on a test system as part of this process. QUERY FRAMES or PERFKIT can be used to watch how much memory is locked. For more information, see *z/VM: CP Commands and Utilities Reference* and *z/VM: Performance Toolkit Reference*.

After the fail percent is determined, PCIe function use should be added in a controlled way, watching the amount of memory locked. This will determine the number of functions that VM can support.

## Setting Up PCIe

The warning percent should be set low enough to allow preventive action, such as stopping some low priority work, when the percentage of locked pages is getting too high.

---

## Recovering from Permanent Errors on PCIe Function

If a PCI function goes to permanent error state, an attempt to recover can be made. A class G user can use the RESET command. See the “RESET PCIFUNCTION command” for more details. A class B user can use the VARY command. See the “VARY PCIFUNCTION command” for more details. If this doesn't work, there is probably a hardware problem that must be corrected.

For more information, see *z/VM: CP Commands and Utilities Reference*.

---

## PCIe Debugging Aids

There are two types of PCI debug aids that are provided to help IBM service resolve problems. Customers will be told how to gather traces, which will be interpreted by IBM.

The “SET CPTRACE command” and “TRSOURCE TYPE PCI” operand control what will be traced. For more information, see *z/VM: CP Commands and Utilities Reference*.