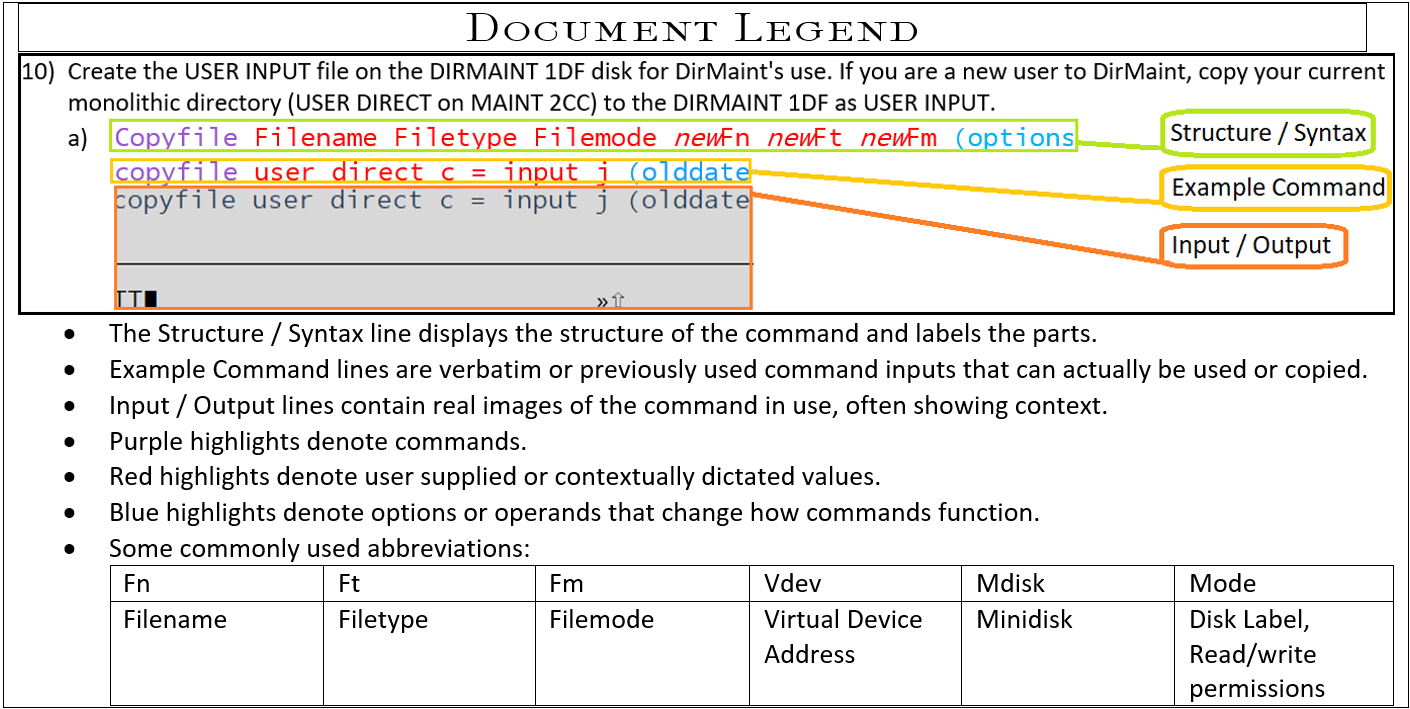
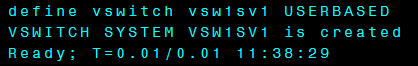
Creating a QDIO Virtual Switch

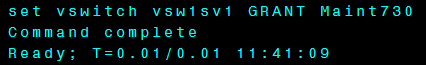
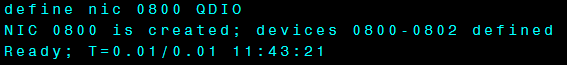


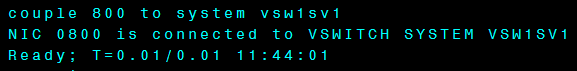
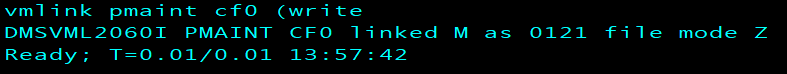
Task Overview

1. Create a virtual switch to act as the network between virtual machines.
2. Set access for management of the switch
3. Create a simulated network interface card (NIC) on each virtual machine to be connected to the virtual switch.
4. Couple each virtual machine’s simulated NIC to the virtual switch. (repeat for relevant VM’s.)
5. Set Sysconfig Statements for persistence through IPL.

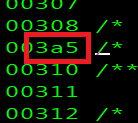
Detailed Steps

1. Create([Define](#Define_Vswitch)) the virtual switch.
   1. Define vswitch vswitchname options  
      Define vswitch vsw1sv1 USERBASED  
      

1. [Set](#Set_Vswitch) the switch to grant Maint730 management privileges.
   1. Set vswitch vswitchname operands options  
      Set vswitch vsw1sv1 GRANT MAINT730  
      
2. Create([Define](#Define_NIC)) the simulated NIC for the virtual switch to connect to.
   1. Define NIC vdev operands options  
      Define NIC 0800 QDIO  
      

1. [Couple](#Couple) the NIC and switch.
   1. Couple vdev to system vswitchname  
      Couple 800 to system vsw1sv1  
      
2. Edit the System Configuration file on PMAINT CF0 to include the virtual switch information.
   1. VMLINK ownerid vdev (options  
      Vmlink PMAINT CF0 (Write  
      
   2. Xedit filename filetype filemode  
      Xedit System Config z  
      
   3. /Searchterm  
      /System\_Alias  
      
      1. Navigate through xedit with UP/Down and # of lines. Alternatively use F7 and F8.

UP #  
UP 10

* + 1. Add lines as needed by typing A# where A stands for add and # will be the number of lines to add. This can be submit from the command line as shown above with / or in the prefix column number area.  
       

00301 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

00302 /\* ADD VSWITCH \*/

00303 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

00304 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

00305 DEFINE VSWITCH VSW1 RDEV 00CD

00306 VMLAN MACPREFIX 020001

00307

00308 /\* The following System\_Alias statement is an example. \*/

00309 /\* Remove the comment delimiters to make the statment active. \*/

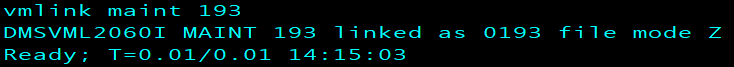
00310 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

00311

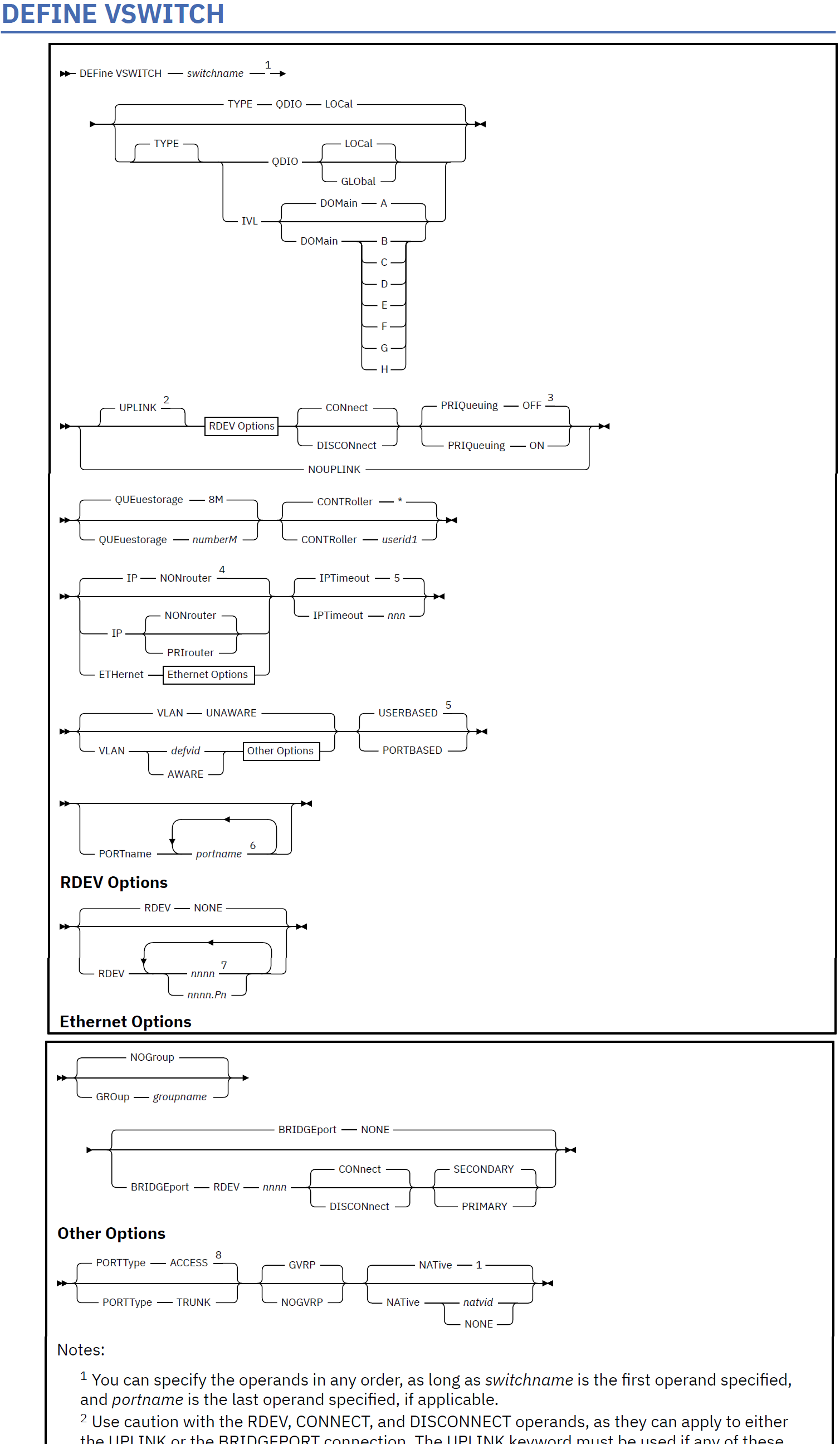
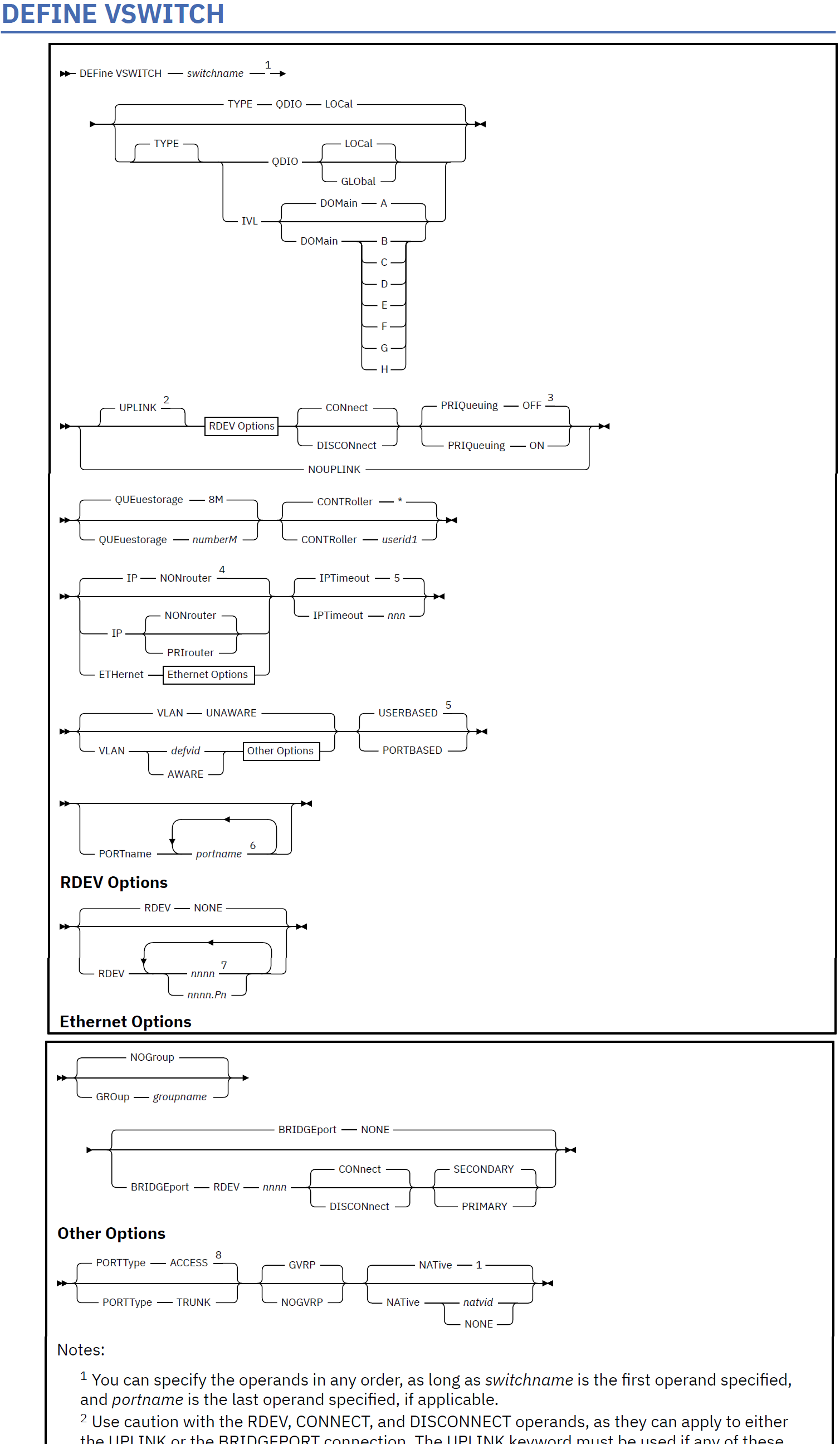
00312 /\* System\_Alias 0000-FFFF \*/

00313

00314

1. Verify there are no issues with the System Config changes by linking MAINT 193 to run the CPSYNTAX executable.
   1. VMlink ownerid vdev  
      Vmlink MAINT 193  
      
   2. CPSYNTAX filename filetype filemode  
      CPSYNTAX system config x  
      
2. Shutdown and REIPL your system.
   1. Shutdown options  
      Shutdown REIPL
3. Monitor for any errors or unexpected outputs.

**[Define Vswitch](#Define_the_virtual_Switch)**



**Authorization**

Privilege Class: B

**Purpose**

Use the DEFINE VSWITCH command to create a CP system-owned switch (a virtual switch) to which virtual machines can connect. Each switch is identified by a *switchname*. A z/VM user can create the appropriate QDIO network interface card (NIC) and connect it to this switch. For information about related commands, see “COUPLE” on page 83 and “DEFINE NIC” on page 194.

For more information about Virtual Networking Options, see *z/VM: Connectivity*.

The SET VSWITCH command can be used to modify attributes of the virtual switch. See “SET VSWITCH”

on page 1755 for more information on this command.

You can also define a virtual switch during system initialization using the DEFINE VSWITCH configuration

file statement. For more information about the DEFINE VSWITCH Statement, see *z/VM: CP Planning and*

*Administration*.

**Operands**

***switchname***

Is the name of the new virtual switch. The *switchname* is a single token (1–8 alphanumeric characters) that identifies this virtual switch for subsequent commands.

**TYPE**

Specifies the type of virtual switch to be created, specifically the hardware and protocol the virtual switch will emulate.

**DEFINE VSWITCH**

**232** z/VM: 7.3 CP Commands and Utilities Reference

**QDIO**

Defines a simulated Ethernet or IP virtual switch. A QDIO virtual switch creates a network comprised of both simulated QDIO devices residing on the same z/VM system, with real network devices located on an external or physical network. A QDIO type virtual switch can only accept connections from a simulated QDIO adapter. External connectivity to network devices on a physical network is achieved using the RDEV or GROUP keywords.

**IVL**

Defines an Inter-VSwitch Link which provides the communication facility to implement an IVL Domain. An IVL domain is a grouping of up to 16 systems running z/VM connected by an IVL LAN segment. All the active members within an IVL domain provide control operations that support the creation and management of shared virtual networking components such as Shared Port Groups.

The IVL virtual switch provides communication infrastructure to exchange control information and data necessary to manage global networking objects that can span multiple systems running z/VM. The IVL virtual switch must be defined with external connectivity (using the RDEV or GROUP keywords) and its UPLINK connection must remain operational in order to support global networking objects.

Only one IVL virtual switch may be created on a system running z/VM.

Guest NICs may not be coupled to an IVL virtual switch.

**DOMain [A | B | C | D | E | F | G | H]**

Defines the domain to which the IVL virtual switch belongs. See “19” on page 242 for more information about an IVL virtual switch.

**GLObal**

Identifies this virtual switch as a member of a global virtual switch. A global virtual switch is a collection of virtual switches that share the same name and the same networking characteristics. This collection of virtual switches spans multiple systems running z/VM but logically operates as a single switch.

Global virtual switches using a shared port group must reside on LPARs in the same CEC.

See usage note 18 for more information about a global virtual switch.

**LOCal**

LOCAL means that the virtual switch is not a member of a global virtual switch.

**UPLINK**

Enables and specifies connectivity for a virtual switch UPLINK port. The UPLINK port is a special port that typically is used to connect the virtual switch to a physical switch, essentially bridging the virtual switch's simulated network to a physical network.

**RDEV *nnnn***

**RDEV *nnnn*.P*n***

Is a real device number or a real device number and OSA-Express port number to be used as an UPLINK port to connect the virtual switch to the appropriate model OSA-Express device. The device selected must be compatible to the type of virtual switch created. Connectivity to the physical network will be prevented when the specified device doesn't match the virtual switch's type.   
Specify each real device number as a hexadecimal number between X'0001' and X'FFFD'. If specifying a real device and an OSA-Express port number, specify the real device number as a hexadecimal number between X'0001' and X'FFFD' followed by a period (.), the letter 'P' (or 'p') followed by the port number as a hexadecimal number between X'0' and X'F'. For example, to specify port 1 for RDEV 300, specify "300.P1". The value of the port number depends on how many ports the OSA-Express hardware adapter supports. If the port number is not specified, it will default to port 0.

You can specify a maximum of three real device numbers. If you specify more than one device number, each must be separated from the others by at least one blank. When the virtual switch has been defined with the GROUP attribute, any devices identified by the RDEV keyword are used for failover in the event of a real switch failure of the link aggregation group. Failover in this environment will be to a single OSA-Express device connected to a second real switch.

**DEFINE VSWITCH**

Chapter 2. CP Commands **233** Each real device number represents a trio of devices. For example, specifying **RDEV 111 222 333** means that the first devices, 111-113, are used to provide the connection to the real hardware LANsegment. When issuing the VSWITCH command without the GROUP attribute, if there is a problemwith the connection, devices 222-224 are used next to provide the connection, and if those devicesfail to connect, devices 333-335 are used. This feature provides dynamic recovery for OSA-Expressdevice failures when issuing the VSWITCH command without the GROUP attribute or for real switchfailure when issuing the VSWITCH command with the GROUP attribute. (Failure of an OSA in anaggregated group is automatically handled by the virtual switch; that is, the virtual switch will transferthe data flow to the remaining OSAs in the group.)

RDEV NONE means that the virtual switch should not be connected to the real LAN segment when defined with NOGROUP. When the virtual switch has been defined with GROUP, RDEV NONE means that there is no link aggregation group failover in the event the real switch should fail. The RDEV operand may not be used to specify device numbers when the virtual switch is configured to use a shared port group. MAC address takeover is managed by the shared port group to maintain connectivity following a failure.

**CONnect**

Indicates that the currently configured virtual switch UPLINK port must be activated, and traffic may flow through the specified UPLINK ports device(s).

**DISCONnect**

indicates that the currently configured virtual switch UPLINK port must not be activated, and that no

traffic is to flow through the specified UPLINK ports device(s).

A virtual switch can be functional without a connection to a real LAN segment, and traffic flows only

between virtual machines coupled to the virtual switch.

**PRIQueuing** Enables or disables guest priority queuing support on all outbound data transmissions from the virtualswitch uplink port to an external network. Priority queuing is a capability of the OSA-Express featurewhere multiple output queues are defined for a single network connection with each queue weightedby a priority on how often it gets serviced by the OSA-Express feature. The highest priority queue isserviced first and more often followed by the next highest priority and so on. No queue is starved andall will get serviced at some point by the OSA-Express feature. See Virtual Switch Priority QueuingFunction in *z/VM: Connectivity* for more information on virtual switch exploitation of priority queuing.If PRIQueuing is to be enabled on a virtual switch, then the OSA-Express features configured to thevirtual switch's uplink port must be configured by IOCP to enable the feature within the adapter(PQ\_ON). (See “DEFINE CHPID / PATH” on page 132 for more information.

**OFF**

The virtual switch will not exploit priority queuing. A single input queue is established for inbound transmissions from the external network and a single output queue is established for outbound transmissions. All outbound data to the external network is transmitted with equal priority. This is the default for TYPE QDIO virtual switches.

The OFF option is not allowed for a TYPE IVL virtual switch.

**ON**The virtual switch will exploit priority queuing. If the customer configured the OSA-Expressfeatures used by a virtual switch uplink port for priority queuing via IOCP (PQ\_ON), then z/VMwill establish one input queue and four output queues when activating its network connection.This will allow z/VM to transmit data to the external network at four different priorities. CP willuse the highest priority queue for control and management traffic. The other three queues (low,normal and high) can be used for virtual NICs' network connections. This is the default for TYPEIVL virtual switches.For an IVL virtual switch, z/VM will attempt to establish an active network connection with the firstOSA device specified. For migration reasons, if the OSA-Express features (used by the IVL virtualswitch uplink port) are configured without priority queuing (PQ\_OFF), z/VM will establish an active

**DEFINE VSWITCH**

**234** z/VM: 7.3 CP Commands and Utilities Reference network connection, and force priority queuing off. A warning message will be displayed to inform the customer to configure priority queuing via IOCP. If the OSA-Express features used by a non-IVL virtual switch uplink port are configured without priority queuing (PQ\_OFF), z/VM will not establish an active network connection, and an error message will be displayed. The policy used to select which priority a specific datagram is transmitted to the external network is determined by the type of virtual switch. For an IVL virtual switch, the priority of outbound transmissions is handled by z/VM. IVL management traffic will be queued and transmitted on a high priority queue, and encapsulated production data failover traffic will be sent on a lower priority queue. The default for TYPE IVL virtual switches is PRIQueuing ON so that IVL network management data and encapsulated production data can be prioritized appropriately. Only an OSA-Express feature configured for priority queuing should be used on an IVL virtual switch's uplink port. For all other virtual switch types, the priority can be set to low, normal or high for all packets sent from a guest NIC's network connection to an external network via the SET VSWITCH command. See the PQUPLINKTX operand on “SET VSWITCH” on page 1755 for more information.

**NOUPLINK**

Indicates the virtual switch will never have connectivity to a physical network through the UPLINK port. This option removes the UPLINK port from the virtual switch. Once the UPLINK port is removed, it can never be added back to the virtual switch. Defining a virtual switch UPLINK port with either the RDEV or GROUP operands while also removing the UPLINK port with the NOUPLINK operand will cause the command to fail. The NOUPLINK option is not allowed for a TYPE IVL virtual switch.

**QUEuestorage *number*M**

Indicates the upper limit of the amount of fixed storage CP and Queued Direct I/O Hardware Facility will use for buffers for each OSA-Express data device. *number* defines the maximum number of megabytes of storage that can be consumed for QDIO queues on a single OSA port. When multiple OSA devices are defined in a link aggregation group, then each OSA port within the group will use the specified amount of storage. Fixed storage is allocated as needed based on network traffic, until the maximum of *number*M are allocated. *number* is a number from 1 to 8. 8 is the default value.

**CONTRoller \***

**CONTRoller *userid1***

Identifies the z/VM user ID that controls the OSA-Express device connected at the device number identified by *rdev*. CONTROLLER \* means CP selects from any of the eligible z/VM TCP/IP stacks. See usage note “3” on page 238 for more information about the function of a controller. If you specify multiple real devices on the RDEV keyword, or through the GROUP keyword, then specify CONTROLLER \*, or allow it to default. The controller functions are then spread across multiple z/VM TCP/IP stacks, providing more flexibility in case of a failure. You can also specify a pool of specific controllers to be chosen from by specifying a list of user IDs after the CONTROLLER keyword with the SET VSWITCH command or the MODIFY VSWITCH statement.

**IP**

**ETHernet**

Indicates whether the transport for the virtual switch is ETHERNET or IP. The ETHERNET transport type is Data Link (Layer 2) based, where the ETHERNET frame is used as a point of reference for source and destination Media Access Control (MAC) addresses in transporting ETHERNET frames on the LAN. The IP transport type is Network (Layer 3) based, where the IP packet is used as the point of reference for source and destination IP addresses in transporting IP packets on the LAN. If this option is omitted for a type QDIO virtual switch, then the transport is set to IP NONROUTER. For an IVL virtual switch the ETHERNET setting is used. An IVL virtual switch can function correctly only

**DEFINE VSWITCH**

Chapter 2. CP Commands **235**

At the Data Link (layer 2) level. Specifying anything other than ETHERNET for an IVL virtual switch will cause the command to fail.

**NONrouter**

Indicates that the OSA-Express device identified by the RDEV keyword will not act as a router to the virtual switch. If a datagram is received at this device for an unknown IP address, the datagram will be discarded. This is the default.  
  
  
  
  
**PRIrouter**

Indicates that the OSA-Express device identified by the RDEV keyword will act as a primary router to the virtual switch. If a datagram is received at this device for an unknown IP address, the datagram will be passed to the virtual switch.

**GROup *groupname***Indicates that the virtual switch UPLINK port is to be configured to use IEEE 802.3ad LinkAggregation. The *groupname* is a single token (1-8 alphanumeric characters) that identifies the group.Use the SET PORT GROUP command to specify the attributes of the group and the OSA-Expressdevices that will make up the group. This option can only be specified when the virtual switch hasbeen defined with the ETHERNET transport attribute. **NOGROUP** means that the virtual switch will notuse Link Aggregation. The port group must be defined using the SET PORT GROUP command beforecreating a GLOBAL virtual switch. For a LOCAL virtual switch, if the group specified has not beendefined, it will be created with this command.

**NOGROUP**

Means that the virtual switch will not use Link Aggregation.

**BRIDGEport**

Configures and specifies connectivity for a virtual switch Bridge Port. The Bridge Port is a special port that is used to connect the virtual switch to a HiperSockets CHPID, essentially bringing the HiperSockets CHPID to the virtual switch's simulated and physical networks. Configuring a virtual switch Bridge Port is only supported when the Bridge facility is supported by the processor, the virtual switch transport type is ETHERNET, and the virtual switch TYPE is QDIO. Additionally, the ISOLATION feature cannot be used when the virtual switch has a Bridge Port. Only guest operating systems running in a virtual machine under z/VM and exploiting QDIO Enhanced Buffer State Management (QEBSM) are eligible to be bridged from the HiperSockets CHPID to the virtual switch's simulated and physical networks.

**NONE**

Specifies that the virtual switch does not have a Bridge Port.

**RDEV *nnnn***

Is a real device number to be used as a Bridge Port to connect the virtual switch to a HiperSockets CHPID. Only devices that have been configured with the HiperSockets CHPID parameter EXTERNAL\_BRIDGED can be specified. Use these parameters on the DEFINE CHPID command or the appropriate CHPARM in the IOCP (x4 for EXTERNAL\_BRIDGED). The device selected must be compatible to the type of virtual switch created; for example EXTERNAL\_BRIDGED for a QDIO type virtual switch. Connectivity to the HiperSockets CHPID will be prevented when the specified device does not match the virtual switch's type. You must specify a single real device number as a hexadecimal number between X'0001' and X'FFFD'. The real device number specified represents a trio of devices. For example, specifying BRIDGEPORT RDEV 508 means the devices, 508-50A, are used to provide the connection to the HiperSockets CHPID.

**CONnect**

Indicates that the device identified by the RDEV keyword will be immediately activated, allowing the connection to be used as the active or standby Bridge Port.

**DISCONnect**

Indicates that the device identified by the RDEV keyword will be placed in the inactive state. If this connection is the active Bridge Port connection, another virtual switch with a Bridge Port on the same CHPID in standby state will take over the active Bridge Port connection.

**DEFINE VSWITCH**

**236** z/VM: 7.3 CP Commands and Utilities Reference

**SECONDARY**

Indicates that this virtual switch should be assigned a role as a SECONDARY Bridge Port for the HiperSockets CHPID. This is the default. See usage note “17” on page 241 for more information about the SECONDARY Bridge Port role.

**PRIMARY**

Indicates that this virtual switch must be assigned the role as the PRIMARY Bridge Port for the HiperSockets CHPID. See usage note “17” on page 241 for more information about the PRIMARY Bridge Port role.

**IPTimeout *nnn***

Indicates the length of time in minutes that a transient IP address table entry remains in the IP address table for the virtual network. A transient entry is an entry added to the virtual network by one guest on behalf of another. This operand has no meaning for ETHERNET networks. *nnn* is a number from 1 to 240. 5 minutes is the default value.

**VLAN**

Used to enable and configure IEEE standard 802.lQ VLAN support for the virtual switch being defined. A VLAN-aware virtual switch provides VLAN controls at the user level (with SET VSWITCH GRANT, SET VSWITCH PORTNUMBER and SET VSWITCH VLANID commands) that cannot be overridden by a guest host. If this option is omitted for a type QDIO virtual switch, then the virtual switch is set to VLAN UNAWARE.  
  
**UNAWARE**

Indicates the virtual switch does not support IEEE standard 802.lQ. All frames flow within the virtual switch regardless of presence or absence of Virtual Local Area Network (VLAN) tags. Any VLAN tags present in the frames will be ignored within the switch (however the guest hosts may perform VLAN filtering at the virtual NIC level).

***defvid***

Defines the virtual switch as a VLAN-aware switch supporting IEEE standard 802.lQ. The *defvid* defines the default VLAN ID to be assigned to guest ports when no VLAN ID is coded on the SETVSWITCH GRANT VLAN command, MODIFY VSWITCH GRANT VLAN statement, the SET VSWITCHPORTNUMBER command, the SET VSWITCH VLANID command, or through an ESM. It is a numberfrom 1 to 4094.

**AWARE**

Defines the virtual switch as a VLAN-aware switch supporting IEEE standard 802.lQ without a default VLAN ID. When a virtual switch is specified as VLAN AWARE, one or more VLAN IDs must be assigned to each guest port by either a SET VSWITCH GRANT VLAN command, MODIFY VSWITCH GRANT VLAN statement, the SET VSWITCH PORTNUMBER command, the SET VSWITCH VLANID command, or through an ESM. Failure to assign an explicit VLAN ID causes all untagged frames transmitted from the port to be discarded. In the case of a VLAN-unaware guest using a PORTTYPE ACCESS all outbound frames will be discarded until a VLAN ID is set for the port.

**PORTType ACCESS**

Defines the default porttype attribute for guests authorized for the virtual switch. For PORTTYPE ACCESS, the guest is unaware of VLAN IDs and sends and receives only untagged traffic.

**PORTType TRUNK**

Defines the default porttype attribute for guests authorized for the virtual switch. For PORTTYPE TRUNK, the guest is VLAN aware and sends and receives tagged traffic for those VLANs to which the guest is authorized. If the guest is also authorized to the *natvid*, untagged traffic sent or received by the guest is associated with the native VLAN ID (*natvid*) of the virtual switch. PORTTYPE TRUNK is not allowed for a TYPE IVL virtual switch.

**GVRP**

Indicates that the VLAN IDs in use on the virtual switch should be registered with GVRP-aware switches on the LAN. This provides dynamic VLAN registration and VLAN registration removal for networking switches. This eliminates the need to manually configure the individual port VLAN assignments.

**DEFINE VSWITCH**

Chapter 2. CP Commands **237**

**NOGVRP**

Do not register VLAN IDs with GVRP-aware switches on the LAN. When NOGVRP is specified VLAN port assignments must be configured manually.

**NATive *natvid***

**NATive NONE**The *natvid* option is available only on a VLAN-aware switch. When *natvid* is a number from 1 to 4094,this operand defines the native VLAN ID that is to be associated with untagged frames received andtransmitted by the virtual switch. NATIVE NONE causes all untagged frames to be discarded, insteadof being delivered with a native VLAN ID. If this option is omitted, 1 is used as the *natvid* for a typeQDIO virtual switch.

**USERBASED**

Operational differences between USERBASED and PORTBASED VSwitches have been eliminated. USERBASED specifies that user based rules will be applied when relocating a guest in an SSI. A user-defined port may or may not be preserved over a relocation. Port numbers assigned by z/VM will not be preserved. New port numbers will be assigned on the destination system. If portnumber predictability across an SSI is required, use the PORTBASED option on the VSwitch in each SSI member. Refer to What Is the Difference Between a Port Based and User Based Virtual Switch? in *z/VM:* *Connectivity* for more information.

**PORTBASED**

Operational differences between USERBASED and PORTBASED VSwitches have been eliminated. PORTBASED specifies that port based rules will be applied when relocating a guest in an SSI. Userdefined port numbers will be preserved over a relocation. Port numbers assigned by z/VM will not be preserved. In this case a new port number will be allocated on the destination system. Refer to What Is the Difference Between a Port Based and User Based Virtual Switch? in *z/VM:* *Connectivity* for more information.

**PORTname *portname***

Is a 1- to 8-character name that identifies the OSA-Express adapter. You can specify a maximum of three port names. Multiple port names are used when different port names are needed for the multiple *rdevs* specified on the RDEV operand. See usage note “10” on page 240 for more information.

**Note:** The PORTNAME operand is ignored when the GROUP operand is used.

**Usage Notes**

1. The DEFINE VSWITCH command creates a virtual switch. The SET VSWITCH command can be used to modify a virtual switch by authorizing users to use the switch. Authorization to a virtual switch may also be provided by an External Security Manager.

2. Accounting is set for the switch based on the default accounting state as set by the SET VMLAN ACCOUNT SYSTEM command or configuration statement. If accounting is turned on after the virtual switch is defined, the virtual switch will need to be redefined for accounting to take effect.

3. A virtual switch's connection to a real hardware LAN segment is not operational until an eligible z/VM TCP/IP stack is selected to be the controller for the OSA-Express device. CP selects an eligible z/VM TCP/IP stack to be the controller by either:

* If CONTROLLER *userid1* is specified on the DEFINE or SET VSWITCH commands or the DEFINE or MODIFY VSWITCH System Configuration statements, with either a single user ID or a list of user IDs, only those user IDs are selected.
* If CONTROLLER \* is specified or allowed to default, CP selects from any eligible z/VM TCP/IP stacks.

A z/VM TCP/IP stack becomes eligible when:

**DEFINE VSWITCH**

**238** z/VM: 7.3 CP Commands and Utilities Reference

* The TCPIP MODULE running in the controller is at a release level that supports the function required for the virtual switch. It is recommended that the VSWITCH controller be at the same release level as CP, although all supported releases are allowed.
* An IUCV \*VSWITCH statement is included in its CP directory entry.
* The TCP/IP VSWITCH CONTROLLER statement is coded, and has defaulted to be ON or is explicitly set to ON in the TCP/IP configuration file or through an OBEYFILE command.
* The stack has completed initialization.
* The stack has virtual device numbers available for CP to attach the control device. The virtual device range used by CP is specified in the VSWITCH CONTROLLER TCP/IP configuration statement. If no VDEV range is specified, CP uses the virtual device number (*vdev*) that matches the *rdev* number specified on the DEFINE VSWITCH or SET VSWITCH command. See the VSWITCH CONTROLLER Statement in *z/VM: TCP/IP Planning and Customization* for more information.
  + **Note:** Do not code DEVICE and LINK TCP/IP configuration statements for the device. Do not attach the device to a TCP/IP controller virtual machine. These steps are handled by DEFINE VSWITCH processing when a controller is selected. If an eligible stack is not found, or none of the *rdevs* are operational, you receive a message, and the virtual switch operates in a local LAN environment.

4. The virtual switch supports two modes of operation for data transport in support of both TCP/IP and non-IP based applications. In deciding which mode to deploy for your network, some things to consider about deploying an ETHERNET virtualized LAN segment are:

* Do your servers or applications need to have their own unique MAC addresses (load balancers)?
* Do you plan to deploy non-IP based applications on your network (SNA or NetBIOS, for example)?
* Do you want to build a virtual LAN segment that operates closely to its physical counterpart?

The key attributes of each transport mode with their operational characteristics are as follows: **ETHERNET (Layer 2)**

– Supports all applications that deploy ETHERNET (IEEE 802).

– ETHERNET frames are transported on the LAN segment.

– All destinations are identified by MAC address.

– MAC addresses are locally administered by the LAN administrator through z/VM CP commands or

configuration statements.

– Each host connection is identified by a single MAC address.

– This is a Link Layer transport, in which all hosts maintain their respective ARP caches.

– VLAN tagging resides within the ETHERNET frames per IEEE 802.1Q specifications.

– When GROUP attribute is specified, the frames can be transported as part of the IEEE 802.3ad

standard.

– When a Bridge Port is defined the virtual switch provides physical LAN connectivity for the target HiperSocket channel.

• **IP (Layer 3)**

– Supports IP for TCP/IP applications only.

– IP packets are transported on the LAN segment.

– All destinations are identified by IP addresses.

– IP address assignments are set by the host running in the guest virtual machine.

– Each host may have more than one IP address (multi-homed).

– This is Link Layer independent (that is, no MAC addresses), and ARP processing is offloaded onto

the OSE-Express adapter.

– VLAN tagging resides in internal QDIO headers.

**DEFINE VSWITCH**

Chapter 2. CP Commands **239**

– All hosts share the OSA-Express MAC address. A virtual switch operates in only one mode for a given instance. For example, if the switch is configured as IP, then all communications on the LAN segment must be IP based. The same is also true when the switch is configured in ETHERNET mode. These transport modes affect the method of data transfer for the virtual switch. The other operations of the switch, such as guest authorization, failover, controller configuration, and so on, function the same for both modes.

5.   
The IP transport type is IPv4 only. In order to support IPv6 through the virtual switch UPLINK port, the ETHERNET transport is required.

6.  
 SET VSWITCH cannot be used to change the type of transport. The virtual switch will need to be redefined.

7.  
 Use the QUERY CONTROLLER command output to find the z/VM TCP/IP stacks that are the virtual switch controllers. Use the QUERY VSWITCH command to display information about the virtual switch.

8.  
 CP manages the devices used to control a virtual switch's connection to a real LAN segment through an OSA-Express device. CP attaches the devices to the z/VM TCP/IP virtual machine. CP also defines devices of type VSWITCH-OSD to the z/VM TCP/IP stack, concatenating *switchname* with *vdev* and "DEV" to form the device name and *switchname* with *vdev* and "LINK" to form the link names. These names appear in the TCP/IP query and trace information. DEVICE and LINK statements must not be included in the TCP/IP configuration file for these devices. The port number specified by *nnnn*.P*nn* will be used by the TCP/IP stack (controller) when initializing the device. If the port number is not specified, it will default to port 0. If the port number is not supported, initialization of the device will be terminated.

9.  
 Multiple real devices and portnames can be specified on the RDEV and PORTNAME parameters. This feature allows failover to an alternate real device in the event of a failure with the current OSAExpress device or link aggregation group. All real devices specified must be active and connected to the same hardware LAN in order to effectively and dynamically failover to an alternate device. In addition, the alternate devices must be defined on separate CHPIDs. If your OSA-Express device requires a portname, specify one portname for each real device number.

10.  
 When the real device identified by one of the *rdev*s is started using the VSWITCH command with the NOGROUP attribute, TCP/IP assigns the port name as the hardware adapter name. If an adapter name was already assigned by a previous connection, then the port name must be the same as assigned by any other connection in order to share the adapter. This includes sharing the OSAExpress adapter with this logical partition or all other partitions. The PORTNAME operand is optional. However, some levels of the OSA-Express adapters require that the PORTNAME operand is specified. When such an adapter is in use and the PORTNAME operand is omitted, an error message is displayed during switch initialization. If the device is already started, you must stop it by issuing the SET VSWITCH *switchname* DISCONNECT command before changing the port name.

11.   
PRIROUTER is required only when IP forwarding (routing) nodes will be coupled to the switch. Router nodes provide connectivity for their LAN segments (remote nodes) through their switch connection. When Router nodes are deployed, their switch connection must be configured as PRIROUTER. In addition to this, the switch itself must also be configured as PRIROUTER to the OSA-E adapter. This will insure delivery of datagrams destined for LAN segments that are connected through routers coupled to a switch. Only one connection on each OSA-Express card can be designated as PRIROUTER. If the switch is successful in establishing PRIROUTER on the OSA-Express card, no other node (or switch) sharing the same OSA-Express card will be able to act as PRIROUTER. If another connection has already been established as PRIROUTER, the switch will be left with NONROUTER status (which is reflected in the QUERY VSWITCH response).

12.   
NONROUTER is the default mode for the switch. Every node is directly coupled to the switch and the associated IP destinations are registered with the OSA-Express connection. This is the most efficient way to use the virtual switch. In this mode, packets with an unrecognized IP destination are automatically sent out through the switch connection.

**DEFINE VSWITCH**

**240** z/VM: 7.3 CP Commands and Utilities Reference

13.   
For a VLAN aware virtual switch, you should specify the same native VLAN ID (*natvid*) as specified in your configuration of any physical switches in your network. Most hardware manufacturers use a default of 1.

14.   
SET VSWITCH cannot be used to change the VLAN awareness attribute. To change the attribute, detach the virtual switch and define it again with the correct attribute.

15.   
If the virtual switch is defined as VLAN UNAWARE, any attempts to define a VLAN membership will fail. When the CP access list is used, SET or MODIFY VSWITCH GRANT with VLAN membership list fails. In addition, SET or MODIFY VSWITCH VLANID fails and SET or MODIFY VSWITCH PORTNUMBER with VLAN members fails. When an ESM is used, the COUPLE command fails if a VLAN list is returned by the ESM.

16.   
SET VSWITCH cannot be used to change the USERBASED/PORTBASED attribute. The virtual switch will need to be redefined.

17.   
A virtual switch's Bridge Port can be configured to take on the role as either the PRIMARY or as a SECONDARY Bridge Port for the HiperSockets CHPID. SECONDARY is the default role assigned when defining a virtual switch with a Bridge Port. A single PRIMARY Bridge Port and up to four SECONDARY Bridge Ports can be defined per HiperSockets CHPID. A functional virtual switch configured for the PRIMARY role is always selected as the *active* Bridge Port connection. Functional virtual switches configured with the SECONDARY role provide backup (standby) capability in the event of failure on the part of the active Bridge Port connection. The first virtual switch Bridge Port successfully connecting to the HiperSockets CHPID (whether PRIMARY or SECONDARY) will take on the responsibility as the active Bridge Port connection. When the active Bridge Port connection is a virtual switch with a SECONDARY role, then its responsibility as the active Bridge Port connection will be relinquished to a connection made by a virtual switch requesting the PRIMARY role.

18.   
A Global virtual switch is a collection of virtual switches that share the same networking characteristics. This collection of virtual switches spans multiple systems running z/VM but logically operates as a single switch. Virtual switches defined with the same name and the GLOBAL option are said to be members of the same global virtual switch when they reside in systems running z/VM that belong to the same IVL domain. (Definition and activation of an IVL virtual switch allows a system running z/VM to join an IVL domain. See usage note 20 for more information about the IVL virtual switch.)

The following conditions must be fulfilled in order to create or change a global virtual switch member:

* The system's IVL virtual switch must be defined and its UPLINK port connected. In other words, the system must be an active member of an IVL domain. Creation of Global virtual switches specified in the SYSTEM CONFIG file will be deferred until the IVL virtual switch UPLINK port connects the system to the IVL domain. Message HCP3178I will be displayed for each deferred Global virtual switch.
* If any other virtual switch exists in the IVL domain with the same name the following attributes must match or the DEFINE VSWITCH or SET VSWITCH will fail with message HCP3170E.

– TYPE QDIO

– IP or ETHERNET

– ISOLATION

– VEPA

– VLAN AWARE or UNAWARE

– NATIVE *natvid*

– USERBASED or PORTBASED

* The following additional restrictions exist if a global virtual switch is configured to use a shared port group:

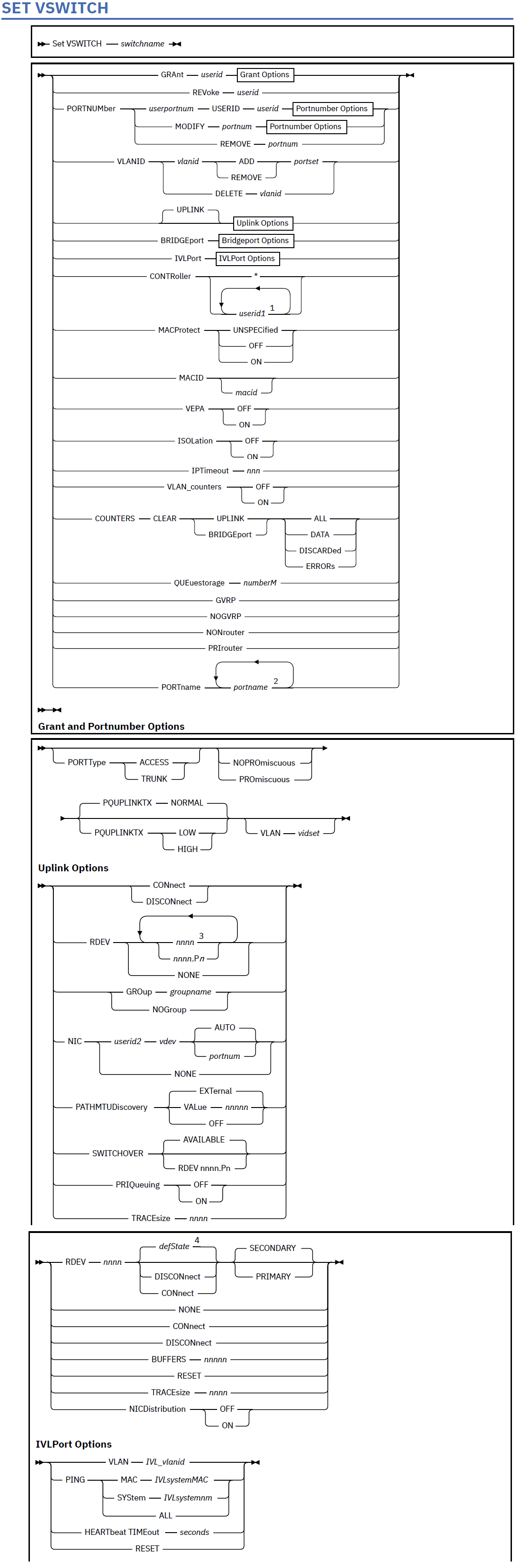
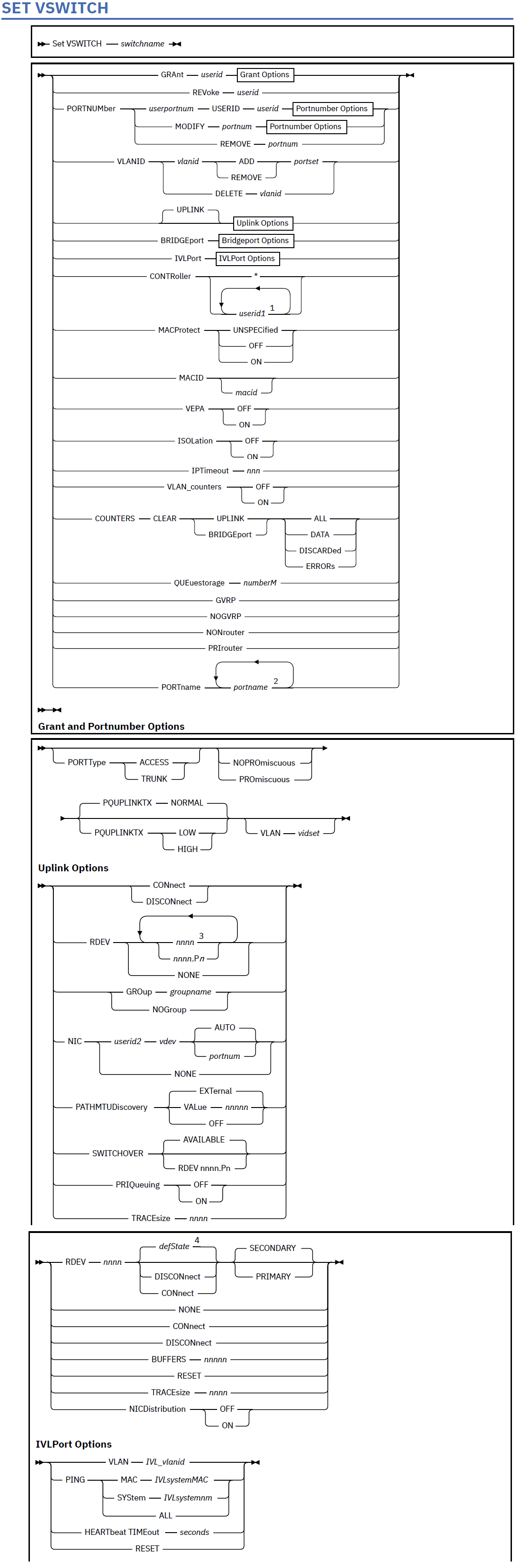
– RDEV *device\_addr* is not allowed

**DEFINE VSWITCH**

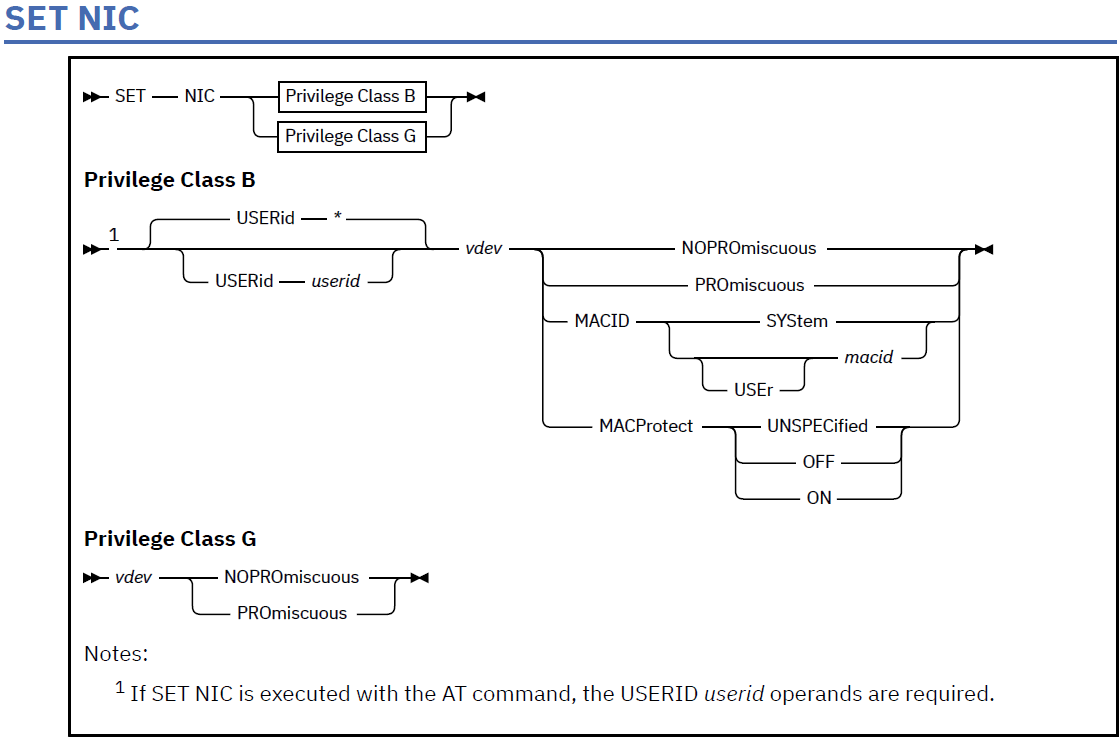
Chapter 2. CP Commands **241**

19.   
A virtual switch defined to be TYPE IVL is an Inter-VSwitch Link which provides communication infrastructure to exchange control information and data necessary to manage global networking objects that can span multiple systems running z/VM. A single IVL virtual switch accommodates all inter-LPAR control traffic for a system. This virtual switch is defined by the system administrator using a CP command or a statement in the System Configuration file. The DEFINE VSWITCH specifies the RDEV or GROUP operand to configure OSAExpress adapters to provide the required connectivity to systems running z/VM in other LPARs so they can all join the same IVL domain. An IVL domain is a group of systems running z/VM that have each defined an IVL VSwitch with an UPLINK port configured and connected to the same external LAN segment. Configuration of an IVL virtual switch defines the system's IVL domain membership. These domains allow isolation of global networking traffic through the enumeration of different domain letters A-H. Each is assigned a separate, reserved multicast MAC address. For example, communications for default Domain A is assigned to 03-FF-FF-FF-FF-01. (MAC address prefix 03-FFFF is reserved for IVL communications for all systems running z/VM.) The VLAN associated with a VLAN-aware IVL virtual switch can be modified using the IVLPORT VLAN operand on the SET VSWITCH command. Up to 8 IVL domains can be defined per VLAN. Note that up to 16 systems can be members of the same IVL domain. When a global virtual switch is in use the IVL virtual switch UPLINK port must remain operational in order to support full function for a global virtual switch that spans multiple systems. See IVL (Inter-VSwitch Link) Overview in *z/VM: Connectivity* for more information.

**[Set Vswitch](#Set_the_switch_to_grant_privs)**



**SET NIC**

****

**Authorization**

Privilege Class: B, G

**Purpose**

Use SET NIC to perform various settings for a specific network device.

A class B user can:

* Toggle promiscuous mode ON or OFF for a specified user ID's virtual network data device. In order for the guest to participate in promiscuous mode, the NIC must be authorized for promiscuous mode and coupled to a guest LAN or virtual switch.
* Specify the mechanism to be used when assigning a MAC address to a real or simulated network data device.
* Enable or disable MAC address protection for a specific network data device. Enabling MAC address protection ensures only the MAC address assigned by z/VM or the system administrator is used when a guest establishes a layer 2 (ETHERNET) network connection. A class G user can only toggle promiscuous mode ON and OFF for a particular virtual network data device.

**Operands**

**USERid *userid***

**USERid \***

Is the user ID of the virtual machine for which you want to set NIC settings. If omitted or specified as an asterisk (\*), CP uses the user ID of the virtual machine issuing the command. The userid specified must be logged on.

**SET NIC**

Chapter 2. CP Commands **1581**

***vdev***

Is the virtual device number to be modified. Typically, this is required to be the device number of what is the data device (or what will be the data device) for the NIC.

**NOPROmiscuous**

**PROmiscuous**

specifies if promiscuous mode is to be toggled ON or OFF. Current Linux distributions enable or disable Promiscuous mode as part of the device driver alleviating the need for this SET command. Authorization to enable Promiscuous mode is still required using SET VSWITCH.

**MACID SYStem**

**MACID [USEr] *macid***

specifies the mechanism to be used when assigning a MAC address to a real or simulated network data device. Specify SYSTEM to have a locally administered MAC address automatically assigned to this device number. The first 3 bytes (6 hexadecimal digits) of the MAC address will be the MACPREFIX value set on the VMLAN system configuration statement and the last 3 bytes will be the next available number in the MACIDRANGE SYSTEM range specified on the VMLAN statement. Specify USER *macid* to have a static MAC address assigned. The *macid* value must be a unique 3-byte (6 hexadecimal digits) identifier. It will be appended to the system MACPREFIX or USERPREFIX value set on the VMLAN system configuration statement to form a unique user-defined locally administered MAC address for this device. When MACPREFIX and USERPREFIX are set to the identical value, the value specified for *macid* must be in the USER subset of the MACIDRANGE SYSTEM range. When MACPREFIX and USERPREFIX are set to different values, the MACIDRANGE settings are ignored. This allows the entire range of MACIDs (000001 - FFFFFF) to be used by either the system or the user when making MAC address assignments. Use SET NIC with the SYSTEM operand to remove a previously assigned USER MACID.

**MACProtect UNSPECified|ON|OFF**

Turns MAC address protection on or off for the specified network device number. Turning MAC address protection on prevents a guest from using a MAC address that is not assigned to the user's network device. MAC address protection is valid only for simulated network devices when coupled to an ETHERNET virtual switch or guest LAN. MAC address protection is not supported for HiperSockets Bridge Capable guest ports that are being bridged by this virtual switch. There are three levels of inheritance used when determining the MAC address protection level for a MAC address assigned to a network device. The highest is the system level, followed by the virtual switch or guest LAN level. The lowest level is the protection set for a specific network data device. The MAC address protection level assigned is determined first by the device, then virtual switch or guest LAN and lastly the system level. If MAC address protection is set at the device level using the SET NIC CP command, then its current setting is used. When the device is set to UNSPECIFIED, it will inherit the MACPROTECT setting of the virtual switch. When the virtual switch MACPROTECT setting is set to UNSPECIFIED, the SYSTEM's MAC address protection specified by the SET VMLAN CP command or VMLAN configuration statement is assigned.

**UNSPECified**

Allows the MAC address protection to be determined by the next higher level. UNSPECIFIED is the default setting for a network device.

**OFF**

Will set the device MAC address protection level to unrestricted or off. The MAC address assigned to this device is not protected. A guest can override the assigned MAC address when establishing a network connection.

**ON**

Will set the device MAC address protection level to restricted or on. The MAC address assigned to this device is protected and cannot be overridden by the guest when establishing a network connection.

**SET NIC**

**1582** z/VM: 7.3 CP Commands and Utilities Reference

**Usage Notes**

1.   
The virtual device number (*vdev*) does not have to exist to set a MACID or the MACPROTECT option. In this case, the option will be preassigned and will be inherited when the virtual NIC is created comprising the device number.

2.   
A SET NIC *vdev* MACID command has no effect when a real device is attached as *vdev*, unless the device driver uses VM MAC address support. That is, the device driver must make a call to subcode X'30' of Diagnose X'26C' to retrieve a MAC address from CP and use that MAC address on link initialization.   
The z/VM TCP/IP stack's OSD device driver provides this support for a link defined to use the ETHERNET transport type. The link must be restarted in order to use the new MAC address. Until the link is restarted, a QUERY VIRTUAL OSA or QUERY VIRTUAL NIC command issued for the device shows a status of "Pending" for the newly assigned MAC address.

3.  
A SET NIC *vdev* MACID command sets the MAC ID for a virtual device. For the z/VM OSD device driver, the link must be restarted in order to use the newly assigned MAC address, unless it is SET before the link is first initialized..

4.   
For the z/VM TCP/IP stack, the *network data device number* is the first device number in a network adapter triplet. That is, a DEFINE NIC 501 creates an adapter with devices 501, 502, and 503. 501 is the data device number and should be used in the SET NIC command. (This information applies to z/VM stacks with code from release z/VM 5.4.0 and later.)

**Responses**

Response 1:

Promiscuous Mode has been enabled for device *vdev*

Response 2:

Promiscuous Mode has been disabled for device *vdev*

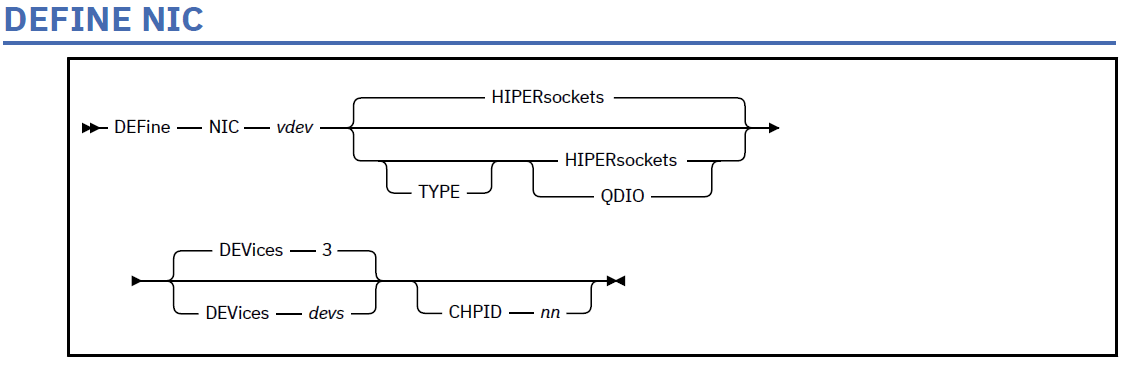
Response 3:

MAC address *macaddr* has been assigned to device *vdev*

Response 4:

A MAC address is automatically assigned to device *vdev*

are the normal responses from this command.

**[Define NIC](#Create_Define_simulated_NIC)  
**

**Authorization**

Privilege Class: G

**Purpose**

Use the DEFINE NIC command to install a simulated network interface card (NIC) in the invoker's virtual machine configuration. A successful definition creates a contiguous range of virtual OSA-type devices which all belong to the same simulated adapter. After the network interface card (NIC) is installed, use the **COUPLE** command to connect the adapter to a guest LAN or virtual switch. VM guests connected to the same LAN segment or virtual switch can exchange messages using the same communication software that they would use to drive a physical network adapter. For more information on the COUPLE command, see “COUPLE” on page 83.

**Operands**

***vdev***

Specifies the base virtual device address for the new adapter. The network interface card (NIC) is represented by a series of I/O devices in your virtual machine configuration. CP messages and responses will use this first device address, the base device address, to refer to the adapter.

**TYPE**

Specifies the type of NIC adapter to be created, specifically the hardware and protocol that the adapter is to emulate.

**HIPERsockets**

Defines this adapter as a simulated HiperSockets NIC. This adapter will function like the HiperSockets internal adapter (device model 1732–05). A HiperSockets NIC can function without a guest LAN connection, or it can be coupled to a HiperSockets guest LAN. You will receive an error if you attempt to connect a simulated HiperSockets adapter to a virtual

switch.

**QDIO**

Defines this adapter as a simulated QDIO NIC. This adapter will function like OSA-Express (QDIO) adapter (device model 1732–01). A QDIO NIC is functional only when it is coupled to a QDIO guest LAN or a QDIO virtual switch.

**DEVices 3**

**DEVices *devs***

Determines the number of virtual devices associated with this adapter. For a simulated HiperSockets adapter, *devs* must be a decimal value between 3 and 3,072 (inclusive). For a simulated QDIO adapter, *devs* must be a decimal value between 3 and 240 (inclusive). The **DEFINE NIC** command will create a range of virtual devices from *vdev* through *vdev+devs-1* to represent this adapter in your virtual machine configuration.

**DEFINE NIC**

**194** z/VM: 7.3 CP Commands and Utilities Reference

If the **DEVices** operand is omitted, the default is **3** for any adapter type.

**CHPID *nn***

Is the two-digit hexadecimal number that represents the CHPID number the invoker wants to allocate for this simulated adapter. If the requested CHPID number is available, all of the virtual devices belonging to this adapter will share the same CHPID number.

**TYPE**

This is an optional keyword that you can specify with HIPERsockets or QDIO.

**Usage Notes**

1. The CP User Directory can be used to automatically define a simulated network interface card (NIC) during LOGON processing. The **NICDEF** statement supports device type **HIPERs** (to create a simulated HiperSockets adapter) or **QDIO** (to create a simulated QDIO adapter).
2. The HiperSockets adapter is functional immediately after it is defined, but it can only communicate with its own internal LAN until it is connected to a guest LAN by the COUPLE command. If an attempt is made to connect to a guest LAN via the NICDEF statement or the COUPLE command, the default internal LAN is destroyed and the adapter becomes "LAN NOTREADY" whenever it is uncoupled.
3. The virtual devices created by the DEFINE NIC command cannot be moved or detached by virtual device number. Each device will retain its assigned virtual device number until the virtual adapter is destroyed by the DETACH NIC command (or by LOGOFF processing).
4. The **CHPID** *xx* operand allows the invoker to request a specific CHPID number for this adapter. If the designated real CHPID number *xx* is available on the LPAR, it will be allocated for the adapter, and assigned to each virtual device that is created as part of this virtual adapter. This option is useful if you need to configure a virtual environment with predictable CHPID numbers for your simulated adapters. It is also useful if you need to carry certain attributes associated with the real CHPID to the virtual adapter. An example would be the NETID associated with an OSD CHPID used as a VSWITCH uplink port. These attributes would be picked up by the internal discovery code run by the guest operating system using the virtual adapter.
5. z/VM supports virtual QDIO networking connections comprised of one read control device, one write control device, and up to eight data devices. This provides the ability to configure up to ten virtual devices per host QDIO connection.

**Examples**

Create a simulated HiperSockets adapter with (3) devices.

cp define nic 500 hipersockets devices 3

NIC 0500 is created; devices 0500-0502 defined

cp define nic 5120 qdio devices 16

NIC 5120 is created; devices 5120-512F defined

**Responses**

Response 1:

NIC *vdev* is created; devices *vdev*-*lastdev* defined

***vdev***

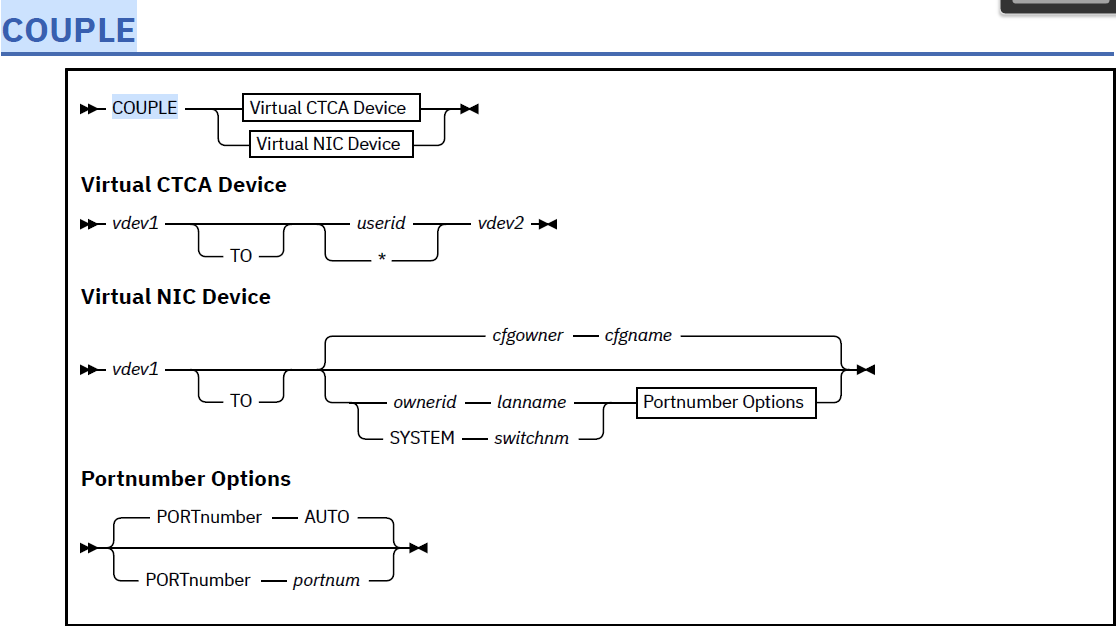
Is the base device address

***lastdev***

Is the last device address in the range created for this adapter (*lastdev* = *vdev* + *devs* - 1).

This is the normal response for this command.

**[COUPLE](#Couple_the_NIC_and_switch)**



**Authorization**

Privilege Class: G

**Purpose**

Use COUPLE to connect a virtual channel-to-channel adapter (CTCA) to a compatible virtual CTCA, or to connect a simulated network interface card (NIC) to a compatible guest LAN or virtual switch.

**Operands**

***vdev1***

Is the virtual device number (hex) of a CTCA, or network device in the invoker's virtual machine configuration. If *vdev1* is a virtual CTCA, it can be connected with a compatible virtual CTCA (designated by *userid vdev2*). If *vdev1* is a fully-simulated virtual OSA device type (created by the DEFINE NIC command), the adapter can be connected to a compatible guest LAN (designated by *ownerid* and *lanname*).

***userid vdev2***

Is the target 'remote' virtual CTCA when *vdev1* is a virtual CTCA. If *userid* is specified as \*, it represents the user ID of the invoker. Use this form of the command to connect two virtual CTCA devices to emulate a single adapter. The invoker should be able to communicate with the 'remote' virtual machine using software appropriate for a compatible physical CTCA.

***cfgowner cfgname***

if a network is configured on the NICDEF user directory statement, the COUPLE command defaults to the configured network (*cfgowner cfgname*).

***ownerid lanname***

Identifies a guest LAN as the target of the connection when *vdev1* is a virtual OSA device. The subject adapter is connected to the LAN named *lanname* owned by *ownerid*. Use this form of the command to connect a simulated network interface card (NIC) to a guest LAN. After a successful connection,

**COUPLE**

Chapter 2. CP Commands **83**

The invoker should be able to communicate with other virtual machines on the same guest LAN using software appropriate for a comparable physical network adapter.

**SYSTEM *switchnm***

Identifies a virtual switch as the target of the connection when *vdev1* is a virtual OSA device. Use this form of the command to connect a simulated network interface card (NIC) to a virtual switch.

**PORTnumber *AUTO***

CP determines the virtual switch port number to be used. Connect using an automatically assigned port number. If a NICDEF PORTNUMBER is configured for this device, the configured port and associated NICDEF attributes will be used. If a single port has been reserved for this user (using SET VSWITCH), the port and its associated port settings will be used. If more than one port has been reserved for this user, then the COUPLE command will fail because it cannot automatically resolve port number.

**PORTnumber *portnum***

The NIC is coupled to the specified *portnum* on the virtual switch. The specified *portnum* must be configured by SET VSWITCH, or the NICDEF must include network attributes. Otherwise the COUPLE command will fail. The *portnum* is a value from 1 to 2048.

**Usage Notes**

1.   
If you do not specify TO, the *userid* you specify cannot be T or TO.

2.  
 If *vdev1* is already connected to another virtual device, a guest LAN, or a virtual switch, that connection is broken. However, when a target virtual CTCA is already connected to another virtual device, the existing connection is **not** broken.

3.  
 If an external security manager is installed on your system, you may not be authorized to enter this command. For additional information, contact your security administrator.

4.   
The connection established by the **COUPLE** command is automatically broken when the adapter, the guest LAN, or the virtual switch is detached. If you need to break this connection without detaching the adapter, the guest LAN, or the virtual switch, use the **UNCOUPLE** command.

5.  
A simulated network interface card (NIC) can be coupled only to a compatible guest LAN or virtual switch. For example, a QDIO NIC can be coupled only to a QDIO LAN or a QDIO virtual switch. Any attempt to connect to an incompatible network will be rejected and reported by message HCP6024E.

6.  
 To connect a NIC to a virtual switch:

* The connection may be configured by the NICDEF user directory statement that defines this virtual NIC. This method binds the port configuration to a specific virtual NIC in the user directory. When NICDEF LAN is configured, it also authorizes the connection (without the need for a SET VSWITCH command). If the network is protected by an External Security Manager (ESM) then use the appropriate ESM interface to authorize this connection.
* A user-defined port may be configured with the SET VSWITCH PORTNUMBER command. This method binds the network configuration to a specific port (in the range 1-2048) and allows the user to decide which virtual NIC to connect with each port. Use the appropriate External Security Manager (ESM) interface to grant authorization if the virtual switch is protected by an ESM.
* A port may be configured with the SET VSWITCH GRANT command where the port number is assigned by z/VM. This method binds the network configuration to a user ID. Any virtual NIC belonging to this user will be connected to a port (in the range 2176-4095) and will adopt the port configuration. Use the appropriate External Security Manager (ESM) interface to grant authorization if the virtual switch is protected by an ESM.

7.  
 After the virtual NIC is coupled to a virtual network, the port configuration can be changed dynamically:

* Use a SET VSWITCH GRANT or PORTNUMBER command to change the configuration, including the configured Virtual Local Area Network (VLAN) ID set, dynamically. The guest does not have to uncouple and re-couple to activate this change. This method only overrides the NICDEF configuration until the next time this virtual NIC is coupled to the virtual network.

**COUPLE**

**84** z/VM: 7.3 CP Commands and Utilities Reference

* If the NICDEF statement (in the user directory) provides network configuration, update the directory with a new configuration and refresh the configuration by having the guest uncouple and re-couple to the virtual network.

8.   
For a virtual CTCA the specified user must be logged on, and the virtual machine must have a virtual CTCA defined. The virtual machine may be operating in disconnect mode.

9.   
If you want to connect the *vdev1* CTCA to another CTCA in your own virtual machine, specify *userid* asan asterisk (\*).

10.   
The *vdev2* user must have authorized the connection using the DEFINE CTCA command or the SPECIAL directory entry.

11.   
The originator of the command and the owner of the coupled CTCA (if different from the originator) receive messages indicating that the CTCAs have been coupled.

12.   
It is possible to "UNCOUPLE" a virtual CTCA device by using an obscure form of the **COUPLE** command (**COUPLE** *vdev1* **TO** \* *vdev1*). This method is supported for compatibility, but shouldbe replaced by the **UNCOUPLE** command. Message HCPCPL058E (RC58) results when using thismethod.

13.   
The virtual ESCON® devices (defined with subclass SCTC, BCTC or CNC) cannot be coupled to virtual 3088 devices (defined with subclass CTCA or 3088), or to virtual FICON devices (defined with subclass FCTC). A virtual ESCON connection requires one controller (SCTC or BCTC) coupled with one non-control device (the CNC). The following combinations are permitted for ESCON devices:

* COUPLE a subclass SCTC device with a subclass CNC device to get a virtual ESCON connection in Standard Mode
* COUPLE a subclass BCTC device with a subclass CNC device to get a virtual ESCON connection in Basic Mode.

A virtual FICON device (defined with FCTC) cannot be coupled to virtual 3088 devices (defined with

subclass CTCA or 3088) or to virtual ESCON devices (defined with subclass SCTC, BCTC, or CNC).

Any other combination of virtual device types will be rejected and the invoker will receive message

HCP6024E.

**Responses**

Response 1:

CTCA *vdev1* COUPLED TO *userid vdev2*

is the normal response sent to the invoker of the COUPLE command when *vdev1* represents a virtual

CTCA.

***vdev1***

Is the virtual device address of the invoker's CTCA.

***userid vdev2***

Is the user ID and virtual device address of the targeted CTCA.

Response 2:

CTCA *vdev2* COUPLED BY *invoker vdev1*

Is the normal response sent to the owner of the target CTCA.

***vdev2***

Is the virtual device address of the targeted CTCA.

***invoker vdev1***

Is the user ID and virtual device address of the originating CTCA.

Response 3:

CTCA *vdev1* DROPPED FROM *userid vdev2*

**COUPLE**

Chapter 2. CP Commands **85**

Is the normal response sent to the invoker (and to the other userid) if the originating CTCA was already coupled to a different device.

***vdev1***

Is the virtual device address of the "local" CTCA.

***userid vdev2***

Is the user ID and virtual device address of the "remote" CTCA.

Response 4:

NIC *vdev1* is connected to *ownerid lanname*Is the normal response sent to the invoker if *vdev1* is part of a simulated network interface card (NIC).

***vdev1***

Is the virtual device address that is part of a simulated network interface card (NIC).

***ownerid lanname***

Is the fully-qualified name of the coupled guest LAN.

Response 5:

NIC *vdev1* is connected to SYSTEM *switchnm*

Is the normal response sent to the invoker if *vdev1* is part of a simulated network interface card (NIC) connected to the virtual network via a port assigned by z/VM (2176-4095).

***vdev1***

Is the virtual device address that is part of a simulated network interface card (NIC).

***switchnm***

Is the name of the virtual switch.

Response 6:

NIC *vdev1* is connected to SYSTEM *switchnm* port portnumber

Is the normal response sent to the invoker if *vdev1* is part of a simulated network interface card (NIC) connected to the virtual network via a user-defined port (1-2048).

***vdev1***

Is the virtual device address that is part of a simulated network interface card (NIC).

***switchnm***

Is the name of the virtual switch.

***portnumber***

Is the port number assigned to the connection.