IBM® Power® Draw Installation and Usage Guide

Overview

IBM® Power® Draw is a chargeable product created by the Technology Expert Labs team. IBM® Power® Draw creates interactive diagrams that detail the hardware and configuration of a Power virtualized environment based on capture files collected by the capture module. IBM® Power® Draw is distributed as a Multiplatform Java JAR file that can be executed on any platform that has a version 1.8 or higher Java Runtime Environment, including IBM i, AIX, Linux, and Windows. On IBM i, an install process provides commands to facilitate running the process in that environment. A license key is required for each Power system serial number that you wish to draw.

The IBM® Power® Draw product includes a separate capture module, which is distributed in a separate install jar file. Please see the PowerCapture Installation and Usage Guide for instructions.

IBM® Power® Draw licensing is based on the serial numbers of the systems that are drawn, rather than the system where the IBM® Power® Draw code is executed. You may install IBM® Power® Draw on a single central computer or distribute the installation to multiple computers depending on where you want to create the drawings.

IBM i Installation

To install or update IBM® Power® Draw on IBM i:

- Transfer the distribution jar file (typically named pwrdraw.jar) to the IBM i IFS using the method of your choice: ACS, FTP, SCP, etc. The file must be transferred as a binary file.
- Open the PASE execution environment with the command: CALL QP2TERM
- Change to the directory where you uploaded the jar file: cd directoryname
- Execute the jar file: java -jar pwrdraw.jar
- The install process will look similar to:

```
 $
> cd /home/vgreene
 $
>  java -jar pwrdraw.jar
Looking up existing version
No current properties installed
Installing new version
Creating Save file
CRTSAVF FILE(QZRDPWRDRW/QZRDPWRDRW) TEXT('PowerCapture')
CPC7301: File QZRDPWRDRW created in library QZRDPWRDRW.

228096 bytes copied to QZRDPWRDRW.FILE
```

RSTLIB SAVLIB(QZRDPWRDRW) DEV(*SAVF) SAVF(QZRDPWRDRW/QZRDPWRDRW) MBROPT(*ALL) ALWOBJDIF(*ALL) RSTLIB(QZRDPWRDRW)
CPC3703: 3 objects restored from QZRDPWRDRW to QZRDPWRDRW.

DLTF FILE(QZRDPWRDRW/QZRDPWRDRW)
CPC2191: Object QZRDPWRDRW in QZRDPWRDRW type *FILE deleted.

182 bytes copied to draw-installer.properties
158019 bytes copied to QZRDPWRDRW.jar

Install Completed

- Press F3 to exit the PASE terminal environment.
- IBM® Power® Draw will have created or updated the library QZRDPWRDRW and created the directory /QIBM/QZRDPWRDRW if it does not already exist.

IBM i License Codes

IBM® Power® Draw licensing is based on the serial numbers of the systems that are drawn, rather than the system where the IBM® Power® Draw code is executed. To create a diagram for a specific system, you must register/store the license code for that system's serial number with this option.

Option 1 – If you have the PowerHA Toolkit (library QZRDHASM):

- Use the QZRDHASM/ADDPRDACS command to add the license codes for each serial number.

Option 2

- Open the PASE execution environment with the command: CALL QP2TERM
- Execute the command:
  java -jar /QIBM/QZRDPWRDRW/QZRDPWRDRW.jar -k serialnum,liccode
- Repeat for each serial number and license code.
- Press F3 to exit the PASE environment.
- On any given system, you should only need to run this command once per serial number.

IBM i Usage

On IBM i, the PWRDRW command is installed in the QZRDPWRDRW library. You can add QZRDPWRDRW to your library list or run it directly as QZRDPWRDRW/PWRDRW.

The command parameters are:

CAPTURE (required) – Capture File Name - This is the input filename for the capture file that contains all the collected information that will be used to create the diagram. This is a text file that is produced by the capture module PWRCAP command as it retrieves Power system configuration data from the HMC. The capture file should be stored in the /QIBM/QZRDPWRDRW/draw_captures/ directory. By default,

the capture module will store the capture files in this directory. You can press F4 to get a list of the files in the draw_captures directory.

DRAWING (required) – Select the drawing types that will be created.

*ALLAVL – Creates all available drawing types (all of those listed below)
*NETWORK creates drawings that include all virtualized network elements, including Shared Ethernet Adapters and associated virtual adapters and virtual switches, SR-IOV adapters, physical, and logical ports, and vNIC connections.

*VNIC creates drawings that include SR-IOV adapters, physical, and logical ports, and vNIC connections.

*STORAGE creates drawings that show all storage connections: Virtual Fibre Channel (NPIV), and Virtual SCSI (vSCSI).

*ALL creates drawings that show all network and storage connections.

You can specify multiple values to draw multiple drawing types. The default is the *ALL drawing.

OUTPREFIX (optional) – Output prefix – This is the prefix used for all output (drawing) file names created. The output file names created are in the form prefix_systemname_type.svg where systemname is the name of the system from the capture file, type is network, vnic, storage, or all depending on the DRAWING option, and prefix is the value specified for this parameter. If the prefix is omitted, the prefix and "_" underscore separator are omitted. The SVG diagrams will be created in the directory /QIBM/QZRDPWRDRW/draw_graphics/ The directory will be created if it does not exist.

MNGSYS - Managed Systems list - defaults to *ALL. You may specify a list of one or more managed system names to draw. If specified, only system names that match the specified case sensitive names will be drawn. The default of *ALL draws all systems in the specified capture file.

LPAR - LPAR List - defaults to *ALL. You may specify a list of one or more client lpar names matching regular expressions to draw. If specified, the drawing will only include client lpars with names that match the regular expressions in the supplied list. The match is case sensitive. The default of *ALL draws all the lpars on the selected systems. For example, a simple substring match is a regular expression – for example, specifying "PROD" would match lpar names PROD01, PRODLPAR, and ACCTPROD. More complex regular expressions are also possible e.g. "PROD[0-9]{2}" to match lpar names containing PROD and two numbers. The regular expression syntax used is Java 1.8. Please note that it is not necessary to understand regular expressions as a simple list of lpar names will also work.

USERDATA – User data files (optional) – Provide a list of userdata text files that contain instructions to customize colors and/or add custom text elements and popup data to the generated diagrams. See the Custom User Data section of the installation guide for instructions. These files must be located in the /QIBM/QZRDPWRDRW/draw_userdata/ directory. This directory will be created when this command is run if it does not exist.

COMPACT – Compact Mode (optional-recommended) – Compact mode draws the virtualization diagrams with one vertical line per VIOS rather than the original one vertical line per physical connection. This produces a diagram that is more compact and easier to navigate interactively than the

non-compact mode which produces much larger diagrams that show all connections without overlap. This defaults to *YES.

The PWRDRW command executes very quickly, even for very detailed capture files, so it is suitable for interactive execution, but it may also be run in batch as needed.

PWRDRW produces log files in the /QIBM/QZRDPWRDRW/logs directory in files stdout.txt and stderr.txt. If there are any problems in the generated diagram, please check these log files for relevant messages. The WRKLNK and DSPF commands can be used to examine the logs, which are simple IFS text files. It is also worth noting that the text popup for the heading at the top of the generated diagrams also includes any warnings from the capture and draw processes.

Upon completion, the SVG diagrams will be in the /QIBM/QZRDPWRDRW/draw_graphics/ directory. These are SVG/XML text files that can be viewed in any modern browser. Typically, you will want to transfer these files to a PC or web server using any IFS capable IBM i tool: ACS, FTP, SCP, etc.

The diagrams can be sent via email but be aware that they contain detailed information about the collected environment, including IP addresses, hardware serial numbers, wwns, mac addresses, system and partition names, etc. that may be considered private.

AIX/Linux/Windows Installation

- Transfer the distribution jar file (typically named pwrdraw.jar) to the location of your choice in the filesystem. There is no installation required -- you can run the process from the jar file in any directory as long as you have a Java runtime environment (JRE) at 1.8 level or higher available.

AIX/Linux/Windows License Codes

IBM® Power® Draw licensing is based on the serial numbers of the systems that are drawn, rather than the system where the IBM® Power® Draw code is executed. To create a diagram for a specific system, you must register/store the license code for that system's serial number with this option.

On any system with a version 1.8 or higher Java Runtime environment, you can execute the pwrdraw.jar file using the Java command with options:

java -jar pwrdraw.jar -k serialnum,liccode

-k serialnum,liccode – Validates and store the license key for a specific serial number. On any given system, you should only need to run this command once per serial number.

AIX/Linux/Windows Usage

On any system with a version 1.8 or higher Java Runtime environment, you can execute the pwrdraw.jar file using the Java command with options:

java -jar pwrdraw.jar [-f [draw]|[drawnetwork][,drawstorage][,drawvnic][,drawall] [-i inputCaptureFile]
[-o outputFileName] [-m managedSystemList] [-l partitionList] [-a userAttributeFiles] [-c]]
|[-k serialnum,liccode]

-f [draw]|[drawnetwork][,drawstorage][,drawvnic][,drawall] (optional) – Draw Type –
Specify a comma-separated list of the drawings to create:

draw – Creates all the available diagrams. If the -f parameter is omitted, this is the default.

drawnetwork - Creates diagrams that include all virtualized network elements, including Shared Ethernet Adapters and associated virtual adapters and virtual switches, SR-IOV adapters, physical, and logical ports, and vNIC connections.

drawvnic - Creates diagrams that include SR-IOV adapters, physical, and logical ports, and vNIC connections.

drawstorage - Creates diagrams that show all storage connections: Virtual Fibre Channel (NPIV), and Virtual SCSI (vSCSI).

drawall - Creates diagrams that show all network and storage connections.

-i – Input Capture File Name - This is the input filename for the capture file that contains all the collected information that will be used to create the diagram. This is a text file that is produced by the capture module as it retrieves Power system configuration data from the HMC. The capture file must be located in the draw_captures/ directory under the current directory. By default, the capture module will store the capture files in this directory.

-o (optional) – Output prefix – This is the prefix used for all output (drawing) file names created. The output file names created are in the form prefix_systemname_type.svg where systemname is the name of the system from the capture file, type is network, vnic, storage, or all depending on the -f option, and prefix is the value specified for this parameter. If the prefix is omitted, the prefix and "_" underscore separator are omitted. The SVG diagrams will be created in the directory draw_graphics/ below the current directory. The directory will be created if it does not exist.

-m - Managed Systems list – (optional). You may specify a list of one or more comma-separated managed system names to draw. If specified, only system names that match the specified case sensitive names will be drawn. The default if omitted draws all systems in the specified capture file.

-l LPAR List (optional) You may specify a comma-separated list of one or more client lpar names matching regular expressions to draw. If specified, the drawing will only include client lpars with names

that match the regular expressions in the supplied list. The match is case sensitive. The default if omitted draws all the lpars on the selected systems. A simple substring match is a regular expression – for example, specifying "PROD" would match lpar names PROD01, PRODLPAR, and ACCTPROD. More complex regular expressions are also possible e.g. "PROD[0-9]{2}" to match lpar names containing PROD and two numbers. The regular expression syntax used is Java 1.8. Please note that it is not necessary to understand regular expressions as a simple list of lpar names will also work.

-a User data files (optional) – Provide a comma delimited list of userdata text filenames that contain instructions to customize colors and/or add custom text elements and popup data to the generated diagrams. See the Customized User Data section of the installation guide for instructions. These files must be located in the draw_userdata directory under the current directory at the time the command is run. This directory will be created when this command is run if it does not exist.

-c Compact mode (optional-recommended) – Compact mode draws the virtualization diagrams with one vertical line per VIOS rather than the original default of one vertical line per physical connection. This produces a diagram that is more compact and easier to navigate interactively than the non-compact mode which produces much larger diagrams that show all connections without overlap. Omitting this flag results in the generation of the original non-compact diagrams.

Customized User Data

A new feature in release 1.2 of IBM Power Draw is support for customization of the generated diagram with user-provided attributes that can be added to the pop-up data or included in the diagrams as custom text with or without associated popups or hyperlinks to other documents.

A good example of how this might be used is to add O/S information such as IP addresses and configuration file or object names that not contained in the HMC collected data, or to include details such as the ethernet or SAN switch port that a specific connection uses.

Command Syntax Changes

User data customization files will be read from the draw_userdata directory. For IBM i the PWRDRW command has been changed to add a new USERDATA parameter. For all other operating systems, the command syntax has been changed to include the '-a' flag that can either be repeated for multiple files or process a comma delimited list of files.

Userdata Syntax with examples

Here are some examples to demonstrate the syntax of each of the new User Data directives and how they will affect the diagrams. The examples below are mostly are based on the BIGSYSTEM demonstration on the main IBM Power Draw web page.

GRAPHIC

Syntax:

```
GRAPHIC,graphicURL
```

GRAPHIC

Constant that defines that this entry is a GRAPHIC declaration. There can only be one GRAPHIC entry defined.

graphicURL

The name of the graphics file that will be displayed in the top left corner of each view. If this is a filename, it must reside in the same folder as the graphic svg files in order for it to be resolved when it is opened by the browser. If this is a fully resolved URL, it will be linked by your web browser and scaled to 50x50 in a 75px square area.

GRAPHIC example

```
GRAPHIC,ibm.gif
```

Displays as:

COLOR

Syntax:

```
COLOR,objectType=htmlColor
```

COLOR

Constant that defines that this entry is a COLOR declaration.

objectType

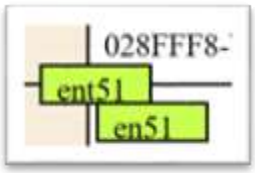
The name of the type of object that will be set to the specified HTML Color. The htmlColor can be HTML named colors, Hex, or rgb specified. Specifying red as hex **#FF0000**, **red**, or **rgb(255,0,0)** are equivalent.

For a list of object types and the currently active colors for each, click on the “Legend” text on any rendered drawing to see the currently active colors with the syntax that would produce them. These will be the default values if no user data was used for the drawing.

COLOR Example

If we want our vritual ethernet connectors to really stand out we can change the default light blue to a greenyellow color:

```
COLOR,VIRTUAL_ETHERNET=greenyellow
```



URL

URL userdata allows you to add a hyperlink to each client lpar or vios lpar with a link that is generated using the lpar name. This can be used to link to static information pages about an LPAR, inventory management systems, help desk ticket systems, Navigator or DCM GUI, etc as long as the URL can be created from the LPAR name.

URL will add the specified link to every Client Lpar or every VIOS Lpar globally. These entries can appear in any order in the userdata files and only one per line. It builds the url based on the urlref, prefix, lpar_name, and suffix. If the urlref=http://www.ibm.com, prefix=test_,suffix=.htm, then the resulting url=http://www.ibm.com/test_''lpar_name''.htm.

Providing multiple entries for the same client will appear as stacked objects within view in the client partition object. Creating entries for software levels, partition functionality or perhaps physical location are a few ideas of the type of data that may be defined.

URLs can be anything your browser will recognize http: https: ftp: etc. You can also use simple filenames to hyper link to other files in the same directory as the SVG file when running locally.

Syntax:

```
URL,dataType,linktext:url=linkurl
```

Or

```
URL,dataType,linktext:urlref=linkstart[,prefix=linkprefix][,suffix=linksuffix]
```

URL

Constant that defines that this entry is a global URL declaration that will be generated for every CLIENTLPAR or VIOLPAR if defined.

dataType

Which lpar type to add the link to. Valid values are:

CLIENTLPAR

Provides the ability to associate add a link to every client lpar.

VIOLPAR

Provides the ability to associate a link to every vios lpar.

linktext

Name to display for the link. This will appear under the profile name and above the connections in the block for each lpar.

url

The link URL is used as the href of the link. You may substitute the lpar name into the output as ${ name}, so the url **http://${lpar_name}.yourdomain.com:2002/Navigator/login** would resolve to **http://LPAR01.yourdomain.com:2002/Navigator/login** for lpar name LPAR01. Note that any XML entity references such as & < > ‘ and “ must be encoded with the associated entity: & < > ' and &qout; or invalid SVG will be produced.

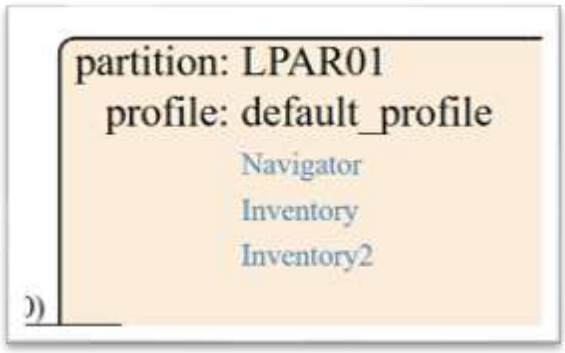
Urlref,prefix,suffix

For compatibility with older user data, you can also use urlref, prefix, and suffix to create a URL in parts. When this format is used, it builds the url based on the urlref, prefix, lpar_name, and suffix. If the urlref=http://www.ibm.com, prefix=test_,suffix=.htm, then the resulting url=http://www.ibm.com/test_''lpar_name''.htm. The url method is much easier and is the recommended syntax.

URL Examples

```
URL,CLIENTLPAR,Navigator:url=http://${name}.mydomain.com:2002/Navigator/login

URL,CLIENTLPAR,Inventory:urlref=http://myinventory.mydomain.com,prefix=lookup
host,suffix=?detail=true

URL,CLIENTLPAR,Inventory2:url=http://myinventory.mydomain.com/lookuphost?host
name=${name}&amp;detail=true
```

These three lines in your userdata file will render like this on each client lpar:



Each of the blue text areas will be a hyperlink to the specified URL

For LPAR01, the "Navigator" link will go to **http://lpar01.mydomain.com:2002/Navigator/login**

The Inventory link will go to **http://myinventory.mydomain.com/lookuphost/LPAR01?detail=true**

The Inventory2 link will go to
http://myinventory.mydomain.com/lookuphost?hostname=LPAR01&detail=true

These links will repeat on each client lpar with the appropriate changes to the link targets.

USERDATA

Userdata elements allow adding text or popup content to most of the graphical elements on the IBM Power Draw diagram. See the examples to understand how it is used. The basic syntax of a USERDATA specification is:

```
USERDATA,managedSystem,objName or Location,DataType,newName[,rel-x][,rel-y][,KEY|URL]:userDefinedData
```

USERDATA

Constant that defines that this entry is a USERDATA declaration. These entries can appear in any order in the userdata files and only one per line.

managedSystem

The name or serial number of the managed system this data belongs to.

objName or Location

The name of the object in the graphical representation to associate the given data. For CLIENTLPAR and VIOLPAR data types, this should be the name of the LPAR to manipulate. For all other types, this is a hardware (or virtual) location that identifies a particular object to alter. The special value ANY specifies that the data should be associated with all objects of a given type.

dataType

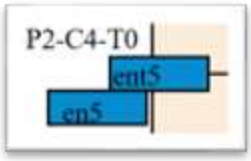
This is the graphical element type to which the changed data will be associated. Valid values are:

ENT

ENT boxes are the top graphical box in a ethernet connection that is represented as a pair of connected boxes. It represents the entx device on a AIX or VIOS connection, or the CMNxx device on an IBM i lpar. If only one box is shown for an ethernet connection, it is the ENT box.

EN

EN boxes are the bottom graphical box in a ethernet connection that is represented as a pair of connected boxes. It represents the enx device on a AIX or VIOS connection. This may not be displayed for all ethernet connections on client lpars.

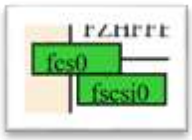


FCS

FCS boxes are the top graphical box in a fiber channel connection that is represented as a pair of connected boxes. It represents the fcsx device on a AIX or VIOS connection. FCS box popups tend to display physical connectivity characteristics. FCS boxes can represent physical or virtual fiber channel connections.

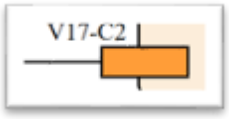
FSCSI

FSCSI boxes are the bottom graphical box in a fiber channel connection that is represented as a pair of connected boxes. It represents the fscsix device on a AIX or VIOS connection. FSCSI box popups tend to display logical or protocol level characteristics. FSCSI boxes can represent physical or virtual fiber channel connections.



VSCSI

VSCSI boxes are the graphical boxes on a client lpar that represent CLIENT VSCSI connections in a virtualized SCSI environment.



VNICCARD

VNICCARD boxes are the leftmost box in the graphical representation of a VNIC connection that represents the adapter level of a VNIC adapter.

VNICPORT

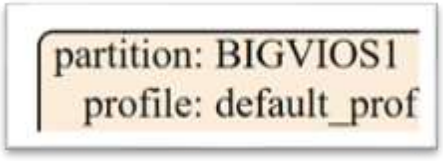
VNICPORT boxes are the middle box in the graphical representation of a VNIC connection that represents the port level of a VNIC adapter. The rightmost boxes that represent individual logical port (virtual function) connections for an LPAR are ENT boxes.

CLIENTLPAR

Provides the ability to associate any user defined data with an existing client partition object based on the name of the partition provided. Providing multiple entries for the same client will appear as stacked objects within view in the client partition object. This provides similar functionality to the URL tag except it is specified for specific lpar names rather than generated for all lpars.

VIOLPAR

VIOLPAR provides the same functionality as CLIENT LPAR above but applies to a VIOS lpar instead of a client lpar.



newName or ADD

This is the text that will be displayed in or near the specified graphical element. If the name here matches the default text for the specified graphical element, the provide attributes will be added to the existing popup data. Otherwise, a new text element will be added to the right of the element (or below for CLIENTLPAR or VIOLPAR). Any additional elements attached to the same datatype and location are stacked below the previous userdata. There is no collision avoidance, so it is possible that userdata may overwrite existing graphical elements or other userdata – use rel-x and rel-y to adjust if needed.

If the newName starts with '$', it is interpreted as a key name that will be replaced by the corresponding value from the key=value combinations that are associated with the datatype and location. This can be used to include values such as wwpns for type FSCSI or mac_addr for type ENT as text directly on the diagram. If the matching value name is not found in the attributes, there will be no output.

ADD

Add is a special value for newName that provides the ability to add additional attributes to an existing object's popup attributes that are already being displayed in the view. The location field indicates the object(s) that should have attributes added to their popup display. ADD normally does not make any changes to the default view except when the object/location specified is normally empty, in which case it will create a text named "Notex" where x is a number that increments for each new userdata element.

rel-x

This is an optional offset to the x coordinate (left-right) of the new element. If specified, the offset (positive or negative) is added to the normally calculated location and will affect the positioning of each additional object added to the same datatype and location. This can be used to fine tune where the user data is positioned.

rel-y

This is an optional offset to the y coordinate (up-down) of the new element. If specified, the offset (positive or negative) is added to the normally calculated location and will affect the positioning of each additional object added to the same datatype and location. This can be used to fine tune where the user data is positioned.

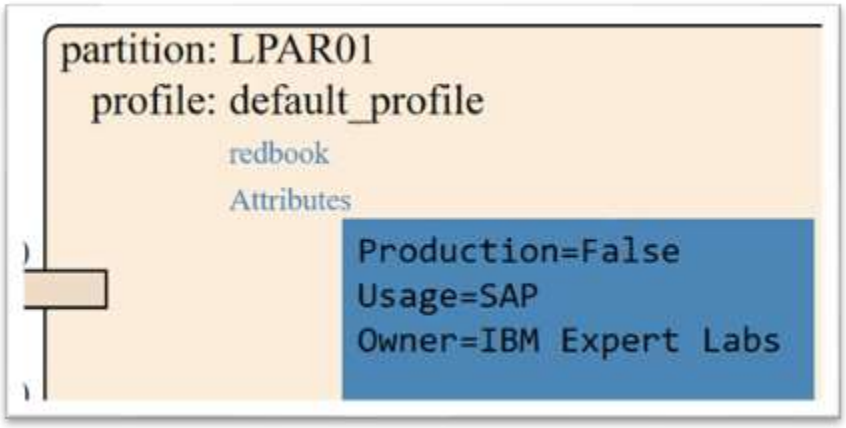
KEY | URL

This is an optional constant that specifies how the user defined data below is interpreted. If KEY is specified, or omitted, the user defined data should be a comma separated list of key=value pairs that will be added to a popup window. If URL is specified, the user defined data should be a URL that will be presented as a hyperlink to another document.

userDefinedData

If dataType KEY or not specified, then userDefinedData is comma separated list of key=value pairs. If dataType URL, then userDefinedData is the url to launch.

USERDATA CLIENTLPAR Examples




```
USERDATA,BIGSYSTEM,LPAR01,CLIENTLPAR,redbook,,,URL:https://www.redbooks.ibm.com/redpapers/pdfs/redp4815.pdf
```

This entry causes the CLIENTLPAR named LPAR01 on managed system BIGSYSTEM to get a text hyperlink added that will link to the PowerVM Redbook

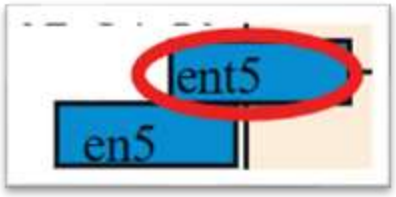
```
USERDATA,BIGSYSTEM,LPAR01,CLIENTLPAR,Attributes:Production=False,Usage=SAP,Owner=IBM Expert Labs
```

This entry causes the CLIENTLPAR named LPAR01 on the managed system BIGSYSTEM to get a new text element “Attributes” added below the other existing userdata. Hovering over the attributes with the cursor will show a popup with the content in the list.

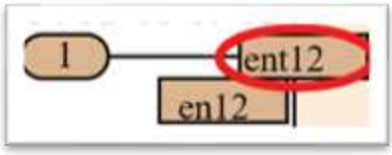
USERDATA ENT type objects

ENT objects are the top boxes in a pair of boxes that represent an Ethernet connection, either physical or virtual. Often IBM Power Draw will only include a single box when drawing an ethernet connection. In these cases that box is always considered an ENT type. All of these can be altered with USERDATA specifications. ENT boxes look like these:

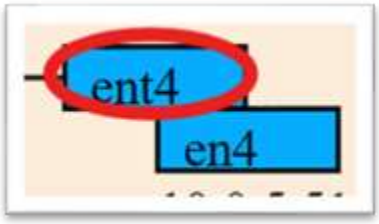
Physical Ethernet Connections:



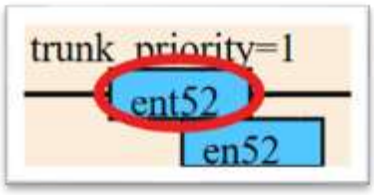
SRIOV Logical ports (not used as VNIC):



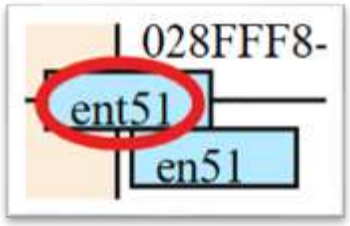
Etherchannel Ports(for vios):



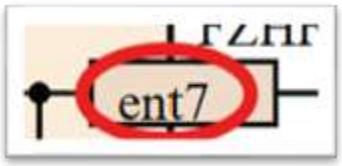
VIOS SEA Adapters:



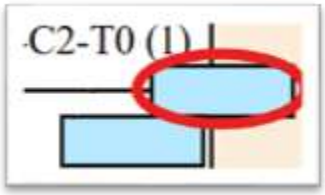
VIOS Virtual Ethernet Adapters:



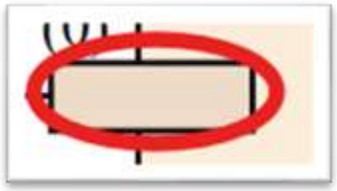
VIOS SRIOV Logical ports adapters:



Client Lpar Virtual Ethernet Adapters:



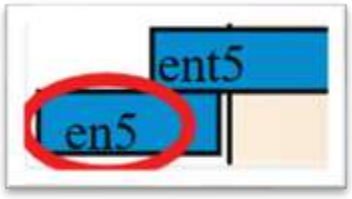
Client Lpar VNIC Adapters:



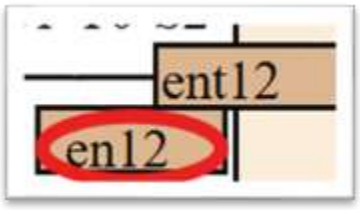
USERDATA EN type objects

EN objects are the bottom boxes in a pair of boxes that represent an Ethernet connection, either physical or virtual. These do not appear on all Ethernet connections. For example, VNIC server adapters on VIOS do not have EN interfaces. In the cases where there is no EN (bottom) box, any userdata specified for EN will be ignored. Where an EN box is included, these can be altered with USERDATA specifications. EN boxes look like these:

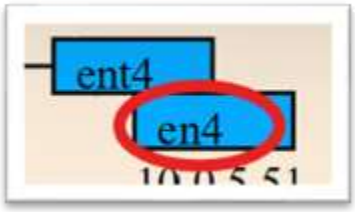
Physical Ethernet Connections:



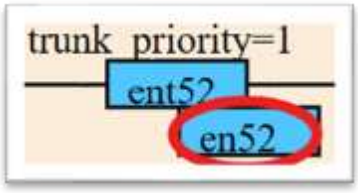
SRIOV Logical ports (not used as VNIC):



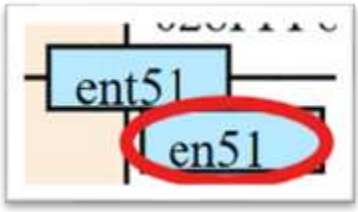
Etherchannel Ports(for vios):



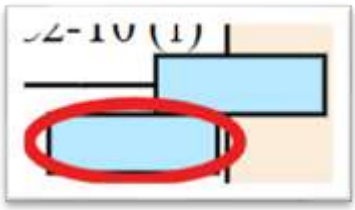
VIOS SEA Adapters:



VIOS Virtual Ethernet Adapters:



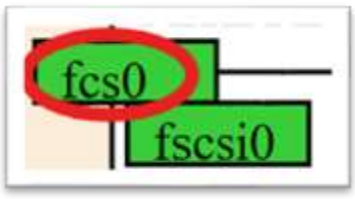
Client Lpar Virtual Ethernet Adapters:



USERDATA FCS type objects

FCS objects are the top boxes in the pair of boxes that represent a Fiber Channel connection, either physical or virtual. All of these can be altered with USERDATA specifications. FCS boxes look like these:

VIOS Shared Physical Fiber Channel:

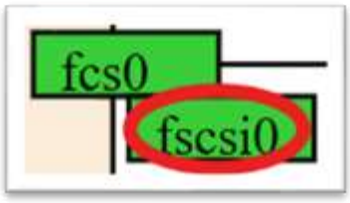


Client Shared (NPIV) Virtual Fiber Channel adapter:

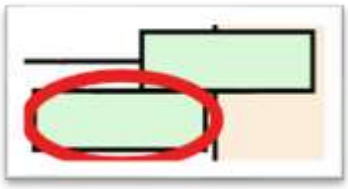
USERDATA FSCSI type objects

FSCSI objects are the bottom boxes in the pair of boxes that represent a Fiber Channel connection, either physical or virtual. All of these can be altered with USERDATA specifications. FSCSI boxes look like these:

VIOS Shared Physical Fiber Channel:



Client Shared (NPIV) Virtual Fiber Channel adapter:

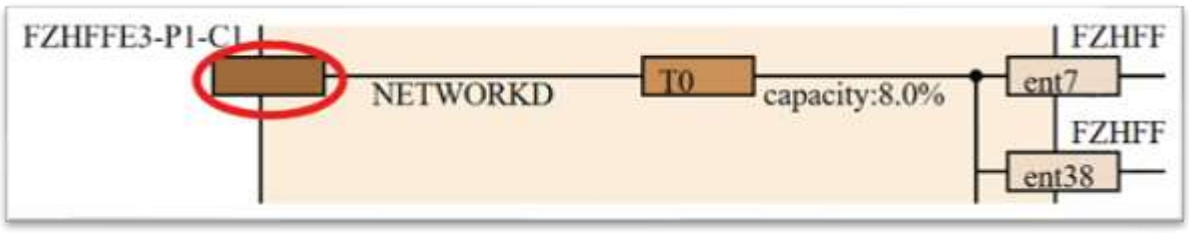


USERDATA VSCSI type objects

VSCSI objects are the client side objects that represent the client virtual adapters that the client uses to present VSCSI resources such as disk or tape mapped to fiber channel, iscsi or storage pools, or file backed virtual optical from a VIOS repository. All of these can be altered with USERDATA specifications. VSCSI boxes look like this:

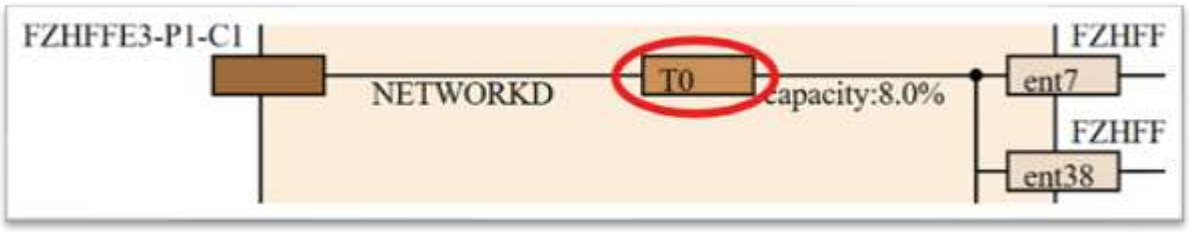
USERDATA VNICCARD type objects

VNICCARD objects represent the adapter level for a SR-IOV interface that is used for VNIC. These appear as the left most adapter in the VNIC section of the VIOS. All of these can be altered with USERDATA specifications. Note that these are normally empty, so adding a new text element will appear by default inside the empty box. VNICCARD boxes look like these:



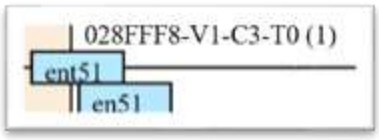
USERDATA VNICPORT type objects

VNICPORT objects represent the Port level for a SR-IOV interface that is used for VNIC. These appear as the middle adapter in the VNIC section of the VIOS. The text in these boxes is the Tx part that indicates a physical port on the SR-IOV adapter. All of these can be altered with USERDATA specifications.
VNICPORT boxes look like these:



USERDATA ADD Examples

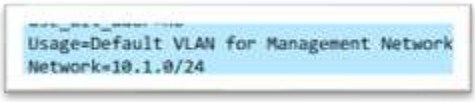
For the default ENT element from the diagram like this (right side of VIOS – Virtual Ethernet)



We can use the following userdata specification to add attributes to the popup for the ent51 entry:

```
USERDATA,BIGSYSTEM,028FFF8-V1-C3-T0,ENT,ADD:Usage=Default VLAN for Management
Network,Network=10.1.0/24
```

This results in no change to the diagram, but adds the data to the popup we get when hovering over ent51:



We can stack multiple entries with ADD, or even use a URL entry as follows to add a hyperlink to the popup box:

```
USERDATA,BIGSYSTEM,028FFF8-V1-C3-
T0,ENT,ADD,URL:https://www.ibm.com/docs/en/power9/0000-FUL?topic=networking-
shared-ethernet-adapters
```

```
Usage=Default VLAN for Management Network
Network=10.1.0/24
link=https://www.ibm.com/docs/en/power9/0000-FUL?topic=networking-shared-ethernet-adapters
```

If we click on the ent51 entry to bring up a separate window with the attributes, the link in the separate window is now a hyperlink to the documentation on Shared Ethernet Adapters.

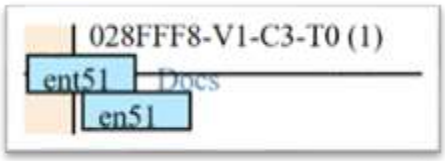
USERDATA new text element Examples

Working with the same entry as above, another alternative for adding the documentation is to add a text link:

```
USERDATA,BIGSYSTEM,028FFF8-V1-C3-
T0,ENT,Docs,URL:https://www.ibm.com/docs/en/power9/0000-FUL?topic=networking-
shared-ethernet-adapters
```

Everything is the same here, except instead of the reserved word ADD, we used our own text “Docs”

That will show like this:



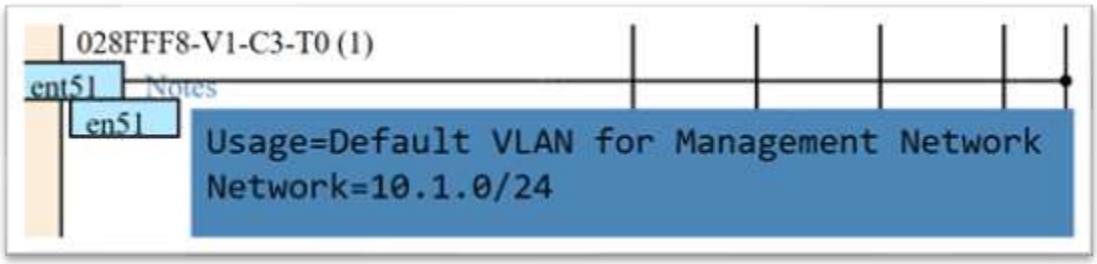
The new hyperlink “Docs” is added in the default location to the right of the anchor graphic.

Hovering over the word shows this (example truncated)

And clicking on the word “Docs” will open a new browser window to the documentation.

Rather than a Hyperlink, you can also use arbitrary key=value combinations to create your own popup. Notice that this is similar to the ADD example above, but has the word “Network” instead of ADD:

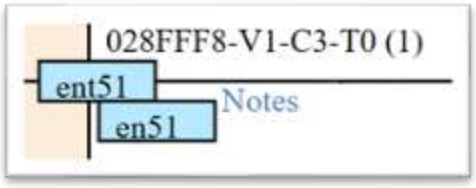
```
USERDATA,BIGSYSTEM,028FFF8-V1-C3-T0,ENT,Notes:Usage=Default VLAN for
Management Network,Network=10.1.0/24
```



USERDATA text element positioning and ANY Examples

The default positioning of the added userdata text above could be improved a bit by specifying a rel-x,rel-y offset (in this case right 13 pixels and down 4:

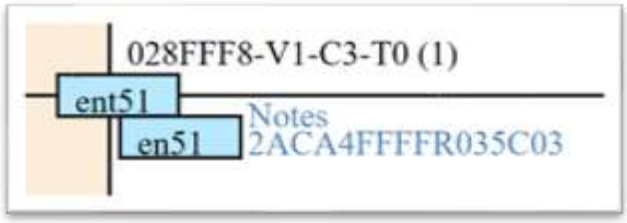
```
USERDATA,BIGSYSTEM,028FFF8-V1-C3-T0,ENT,Notes,13,4:Usage=Default VLAN for
Management Network,Network=10.1.0/24
```



And finally, to show how changing the positioning (rel-x,rel-y) of an added element affects other elements, we can demonstrate using ANY and a variable retrieved from the attributes of the ent adapter. This repeats the previous statement to add “Notes” and includes two more statements to add either mac_addr or mac_address from the attributes – depending on the port type one or the other may be appropriate, and the one that is not correct will be bypassed if it is missing. That gives us:

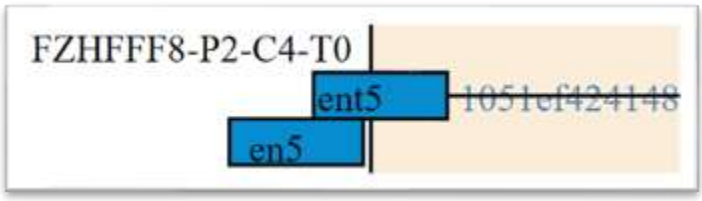
```
USERDATA,BIGSYSTEM,028FFF8-V1-C3-T0,ENT,Notes,13,4:Usage=Default VLAN for
Management Network,Network=10.1.0/24

USERDATA,BIGSYSTEM,ANY,ENT,$mac_addr

USERDATA,BIGSYSTEM,ANY,ENT,$mac_address
```

Which results in:



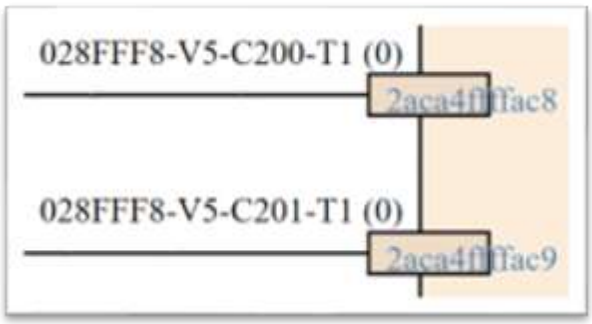
The “Notes” still has a popup as before and is still shifted using our rel-x,rel-y values. The mac address is substituted from the default attributes of the location popup “mac_addr=2ACA4FFFFR035C03 “ and has no popup attributes of its own because those were omitted on the USERDATA specification. You could include :URL=link or :key=value to include attributes associated with the ANY specification.

Note that ANY includes the attributes on EVERY ENT element, so we also see it added here on the physical interfaces for our VIOS:



And here on our VIOS VNIC adapters:

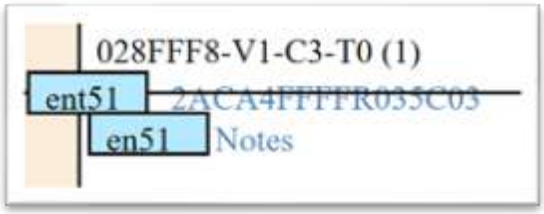
And even here on our client lpars virtual interfaces, where they defaulted to the ENT box text because the box did not have any default text:



USERDATA order of specifications

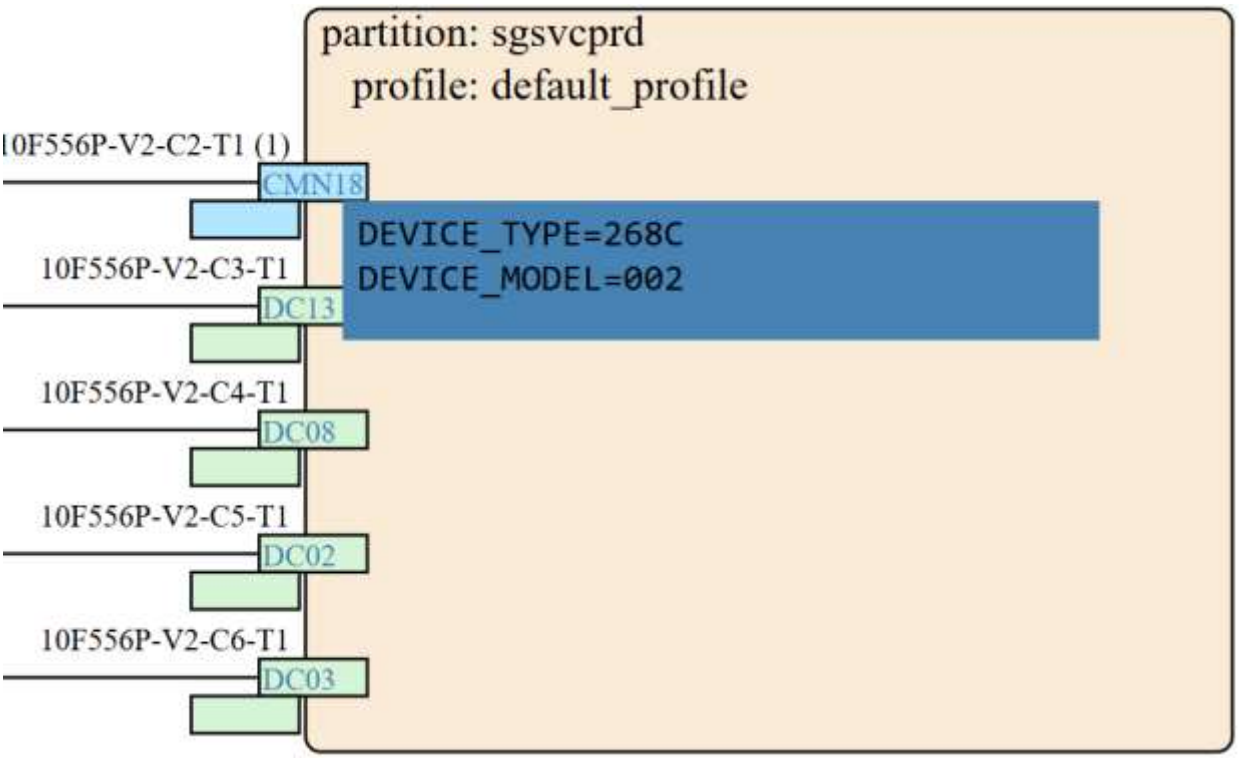
It is important to consider that the order the USERDATA elements are processed controls the order they are displayed. For these ANY cases, the display location was different for the location that had specific notes added before the ANY specification. If those same statements appeared in a different order, they would have displayed differently. This example puts the ANY first, and then the specific location:

```
USERDATA,BIGSYSTEM,ANY,ENT,$mac_addr

USERDATA,BIGSYSTEM,ANY,ENT,$mac_address

USERDATA,BIGSYSTEM,028FFF8-V1-C3-T0,ENT,Notes,13,4:Usage=Default VLAN for
Management Network,Network=10.1.0/24
```



IBM i SQL Services Example

As an example, if you want to populate the ENT and FCS boxes for an IBM i lpar with the resource names (CMNxx and DCxx) to help when tracing connections, you can easily create the user data content using SQL services. In this example, we put the resource name in the ENT/FCS box and include data that includes the device type and model information.



This example code is a qshell script that can be uploaded to IFS and executed as:

QSH CMD('ibmi-userdata.qsh')

For the diagram above, the script generated the following output:

```
USERDATA,10F556P,U9119.MHE.10F556P-V2-C2-
T1,ENT,CMN18:DEVICE_TYPE=268C,DEVICE_MODEL=002

USERDATA,10F556P,U9119.MHE.10F556P-V2-C3-
T1,FCS,DC13:DEVICE_TYPE=6B25,DEVICE_MODEL=001

USERDATA,10F556P,U9119.MHE.10F556P-V2-C5-
T1,FCS,DC02:DEVICE_TYPE=6B25,DEVICE_MODEL=001

USERDATA,10F556P,U9119.MHE.10F556P-V2-C6-
T1,FCS,DC03:DEVICE_TYPE=6B25,DEVICE_MODEL=001

USERDATA,10F556P,U9119.MHE.10F556P-V2-C4-
T1,FCS,DC08:DEVICE_TYPE=6B25,DEVICE_MODEL=001
```

Ibmi-userdata.qsh Example Code

```
# Qshell Script to create an IBM Power Draw userdata file with CMNxx
#  and DCxx resource names

# Setup some variables
myhostname=$(hostname -s)
OUTFILE=$myhostname.userdata

# outputs CMNxx & DCxx resources using SQL service
#    QSYS2.HARDWARE_RESOURCE_INFO
( db2 -ti <<"EOF"
-- CMNxx resources
select 'USERDATA,'||STRIP(CAST(CURRENT_CHARACTER_VALUE AS CHAR(10)))||
','||LOCATION_CODE||',ENT,'||RESOURCE_NAME||
':DEVICE_TYPE='||DEVICE_TYPE||',DEVICE_MODEL='||DEVICE_MODEL
AS TEXT from QSYS2.HARDWARE_RESOURCE_INFO
JOIN QSYS2.SYSTEM_VALUE_INFO ON SYSTEM_VALUE_NAME = 'QSRLNBR'
WHERE RESOURCE_NAME like 'CMN%';
-- DCxx resources
select 'USERDATA,'||STRIP(CAST(CURRENT_CHARACTER_VALUE AS CHAR(10)))||
','||LOCATION_CODE||'-T1,FCS,'||RESOURCE_NAME||
':DEVICE_TYPE='||DEVICE_TYPE||',DEVICE_MODEL='||DEVICE_MODEL
AS TEXT from QSYS2.HARDWARE_RESOURCE_INFO
JOIN QSYS2.SYSTEM_VALUE_INFO ON SYSTEM_VALUE_NAME = 'QSRLNBR'
WHERE RESOURCE_NAME like 'DC%' and LOCATION_CODE is not NULL;
exit;
EOF
) | grep '^USERDATA,' >> $OUTFILE
echo file name $OUTFILE written
```

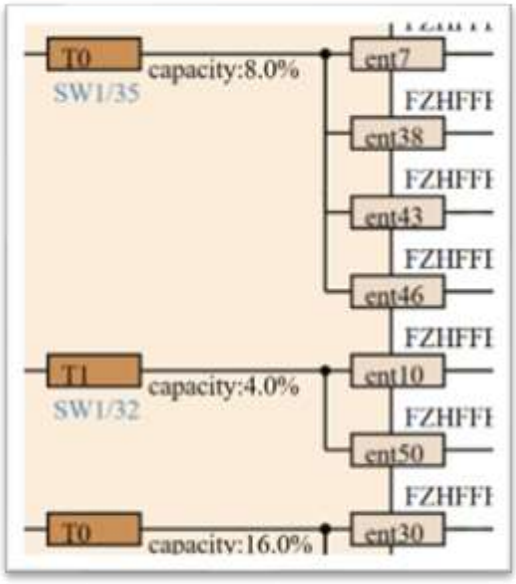
Static Connectivity Information Example

Another helpful application of custom user data would be to create a static file mapping Power hardware locations to their external connections, such as Ethernet switch ports or SAN Switch ports.

If you create the following user data based on your cable maps. This could be very easy to do if it the data is already in a spreadsheet. Note the rel-x of -40 and rel-y of 13 that causes these to be positioned under the port box instead of to the right of the box where it would be harder to read:

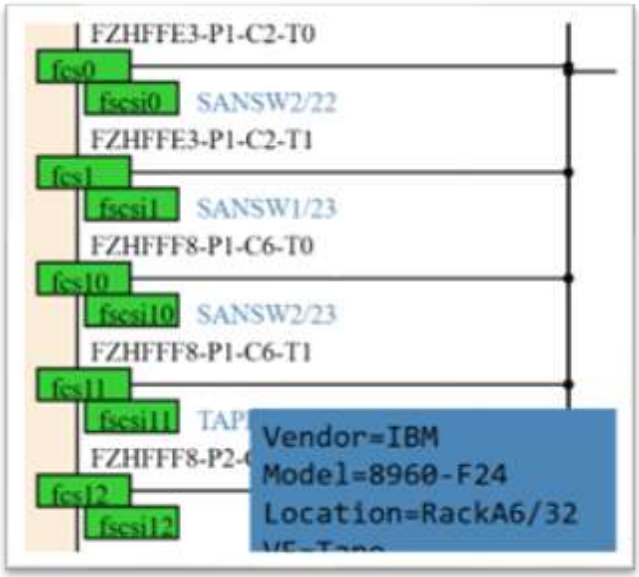
```
USERDATA,BIGSYSTEM,FZHFFE3-P1-C1-T0,VNICPORT,SW1/35,-40,13

USERDATA,BIGSYSTEM,FZHFFE3-P2-C1-T1,VNICPORT,SW1/32,-40,13

USERDATA,BIGSYSTEM,FZHFFF3-P1-C1-T0,VNICPORT,Sw2/32,-40,13
```

You'll get port labels on your VNIC ports:



The same logic can also apply to the Fiber channel connections. In this case FSCSI boxes were used to so the

```
USERDATA,BIGSYSTEM,FZHFFE3-P1-C1-T0,VNICPORT,SW1/35,-40,13

USERDATA,BIGSYSTEM,FZHFFE3-P2-C1-T1,VNICPORT,SW1/32,-40,13

USERDATA,BIGSYSTEM,FZHFFF3-P1-C1-T0,VNICPORT,Sw2/32,-
40,13:Vendor=IBM,Model=8960-F24,Location=RackA6/32,VF=Tape
```


FZHFFE3-P1-C2-T0
fcs0
fcsi0
SANSW2/22
FZHFFE3-P1-C2-T1
fcs1
fcsi1
SANSW1/23
FZHFFF8-P1-C6-T0
fcs10
fcsi10
SANSW2/23
FZHFFF8-P1-C6-T1
fcs11
fcsi11
Vendor=IBM
Model=8960-F24
Location=RackA6/32
fcs12
fcsi12