API - IMPLEMENTED

data output(“DATA”)

host

PC

0

ip = output($host->getIP());

lease = output($host->getExpireDate());

band = output($stats['BW']);

mac = output($host->getMac());

default = output("ERROR: $cmd[4]/$q");

account.[Username]:[Password]

login output("Login Successful");

numberOfPC output(sizeof($userComputer));

PC

# = if( # >= sizeof($userComputer)) { die ("ERROR: $cmd[3]/$q");}

IP = output($userComputer[$num]->getIP());

MAC = output($userComputer[$num]->getMac());

lease = output($userComputer[$num]->getExpireDate());

name = output($userComputer[$num]->getName());

set = output("ERROR: $cmd[5]/$q");

<value> = output("Computer Name Was Successfully Updated");

default = output("ERROR: $cmd[5]/$q");

stats =

MTU = output($PCstats['MTU']);

band = output($PCstats['BW']);

delay = output($PCstats['delay']);

reliability = output($PCstats['reliability']);

transmitLoad = output($PCstats['transmitLoad']);

receiveLoad = output($PCstats['receiveLoad']);

duplex = output($PCstats['duplex']);

speed = output($PCstats['speed']);

5minuteInputBS = output($PCstats['5minuteInputBS']);

5minuteInputPS = output($PCstats['5minuteInputPS']);

5minuteOutputBS = output($PCstats['5minuteOutputBS']);

5minuteOutputPS = output($PCstats['5minuteOuputPS']);

inputByte = output($PCstats['inputByte']);

outputByte = output($PCstats['outputByte']);

inputPacket = output($PCstats['inputPacket']);

outputPacket = output($PCstats['outputPacket']);

description = output($PCstats['description']);

default: = output("ERROR: $cmd[5]/$q");

default = output("ERROR: $cmd[4]/$q");

addPCtoUser = output("This PC has been added to your account");

default = output("ERROR: $cmd[2]/$q");

default = output("ERROR: $cmd[0]/$q");

CLASSES

# class computer

function **computer** ($fcnIP , $fcnMac , $fcnExpireTimeStamp)

**Return**: Nothing **Arguments**: String, String, String

**Notes**: Computer class constructor

**Use**: Object setup

function **setIP** ($fcnIP)

**Return**: None **Arguments**: String

**Notes**: Sets the IP of the computer class (does not update MySQL).

**Use**: Update Information.

function **setMac** ($fcnMac)

**Return**: None **Arguments**: String

**Notes**: Set Mac to allow setting of the variable.

**Use**: Loading/Caching uses. Should not be used otherwise (unpredictable results)

function **setExpireTimeStamp** ($fcnExpireTimeStamp)

**Return**: None **Arguments**: String

**Notes**: Set ExpireTimeStamp variable

**Use**: Updating information

function **setName** ($fcnName)

**Return**: None **Arguments**: String

**Notes**: Sets the name variable

**Use**: Allow users to identify PCs based on names

function **setUser** ($fcnUser)

**Return**: None **Arguments**: String

**Notes**: Sets the User variable

**Use**: (not used yet) Device ownership would be possible

function **updateIP** ($fcnIP, $fcnExpireTimeStamp) - Public

**Return**: None **Arguments**: String

**Notes**: Updates both the IP and Expiration timer

**Use**: (not used yet)

function **getExpireTimeStamp** () – Public

**Return**: string of digits **Arguments**: None

**Notes**: Return the timestamp of the IP lease expiration

**Use**: To obtain Timestamp information, mostly used with the database class for writing to the database

function **getExpireDate** ()

**Returns**: string **Arguments**: None

**Notes**: Returns the date in H:M:S D/M/Y based on the predefined timestamp

**Use**: To obtain human readable date and time information

function **getUser**()

**Returns**: string **Arguments**: None

**Notes**: Returns username of device owner as seen in SQL DHCP table.

**Use**: (not used yet)

function **getName**()

**Returns**: string **Arguments**: None

**Notes**: Returns name of device, string

**Use**: Obtains the name of the device as seen in SQL DHCP table

function **getIP**()

**Returns**: String **Arguments**: None

**Notes**: Returns the ip of the device, string

**Uses**: Obtains the IP of the device

function **getMac**()

**Returns**: String **Arguments**: None

**Notes**: returns the MAC address of the device, string (all lower case, no dividers, IE 00ab1d50c00b00e0)

**Uses**: Obtains the MAC addresses of the device as seen in SQL DHCP table (used to identify the device)

function **PRINTCOMPUTER** ()

**Returns**: Nothing **Arguments**: None

**Notes**: Echo’s PC info (name, IP, Mac, ExpireDate, ExpireTS. Somewhat formatted)

**Uses**: Debugging/Development

function **getComputer**()

**Returns**: Array **Arguments**: None

**Notes**: Returns an array of the computer’s properties using the following indexes: Name, IP, MAC, ExpireDate, ExpireTimeStamp. All case sensitive

**Uses**: Caching

# class IOS\_device

function **IOS\_device** ($fcnIP, $fcnUsername, $fcnPassword, $fcnType, $fcnDHCP)

**Returns**: Array **Arguments**: None

**Notes**: Constructor, type is either IOS\_Switch, or IOS\_Router, DHCP indicates if the device is a DHCP server. Implementation of dynamic scanning of DHCP servers will follow. / Dynamic host to GW association

**Uses**: Class constructor

function **setPort** ($fcnPort)

**Returns**: None **Arguments**: Integer

**Notes**: Sets the SSH port on the SSH connection. Optional, default is 22.

**Uses**: Flexible port options/restoring from cache

function **getPort** ()

**Returns**: Integer **Arguments**: None

**Notes**: Returns the port of the SSH connection

**Uses**: Obtains the SSH port

function **setType** ($fcnType)

**Returns**: Nothing **Arguments**: String

**Notes**: sets the type variable, either IOS\_Switch or IOS\_Router. This is done as there is a slight difference in output formatting between the two platforms

**Uses**: Sets the device type

function **getType** ()

**Returns:** String **Arguments**: String

**Returns** device Type (IOS\_Switch or IOS\_Router)

**Uses**: Obtain the device type

function **setDHCP** ($fcnDHCP)

**Returns**: Nothing **Arguments**: String

**Notes**: sets the DHCP variable. True indicates the device is a DHCP server, false indicates the device is a transit device.

**Uses**: Sets DHCP variable, true/false 1/0

function **getDHCP** ()

**Returns**: Boolean **Arguments**: String

**Notes**: Returns the value of the DHCP variable

**Uses**: Return DHCP variable, true/false

function **connect** ()

**Returns**: Boolean **Arguments**: None

**Notes**: Calls the connect function of the sshconnection of the class (ssh class, connect)

**Uses**: Connects to the device

function **disconnect** ()

NOT CODE YET

function **getUsername** ()

**Returns**: String **Arguments**: None

**Notes**: Returns the username of the object

**Uses**: Obtain the username of the device, Caching

function **setUsername** ($fctUsername)

**Returns**: Nothing **Arguments**: None

**Notes**: Sets the username variable of the object. Should not be used

**Uses**: Sets the username of the device, used in the constructor

function **getPassword** ()

**Returns**: String **Arguments**: None

**Notes**: Returns the password variable of the object.

**Uses**: Obtains the username of the device, Caching

function **setPassword** ($fctUsername)

**Returns**: Nothing **Arguments**: String

**Notes**: Sets the password variable of the object

**Uses**: Sets the password, used in the constructor

function **setRID** ($fctRID)

**Returns**: Nothing **Arguments**: String

**Notes**: sets the RID, the main interface that will be used to identify the device, in addition, this IP Address will be used to communicate with the device.

**Uses**: Sets the RID, used in the constructor

function **getRID**()

**Returns**: String **Arguments**: None

**Notes**: Returns the RID, the main interface that will used to identify the device

**Uses**: Obtains the RIP/IP of the device, used for network redirection

function **addIP** ($fctIP)

**Returns**: Nothing **Arguments**: String

**Notes**: Adds an IP address to the list of IP Addresses assigned to each device. This list is used within the Network redirect object.

**Uses**: Used within the getInterface function

function **getIP** ()

**Returns**: Array of Strings **Arguments**: None

**Notes**: Returns array of IP addresses assigned to the device.

**Uses**: Used to obtain all the IP addresses that a device posses

function **getName** () //Does this return the SQL hostname (TH-Switch) or Location as the name

**Returns**: String **Arguments**: None

**Notes**: Returns the name of the device

**Uses**: Get the device name.

function **setName**($fcnName)

**Returns**: Nothing **Arguments**: String

**Notes**: Sets the hostname variable

**Uses**: Sets the name of the device, used within the constructor

function **sendCMD** ($cmd)

**Returns**: String **Arguments**: Array of Strings

**Notes**: Outdated, migrate to new sendMultiCMD. This function uses ssh2\_exec, and needs to create a new SSH connection for each command issued.

**Uses**: Issues commands to the device, and returns the output in a single string

function **sendMultiCMD**($cmd)

**Returns**: Array of Strings **Arguments**: Array of Strings

**Notes**: Sends multiple commands to a device over a single SSH connection, uses ssh2\_shell. More efficient then sendCMD

**Uses**: Communicate with the device

function **initilizeDevice** ()

NOT IMPLEMENTED CORRECTLY DO NOT USE. Future use: will allow for dynamic provisioning of type, ip, dhcp, and name variable by pulling the running configuration.

function **getClient** ()

**Returns**: Nothing **Arguments**: None

**Notes**: Queries the device for DHCP bindings and returns an array of computers who have been assigned an IP address by this device. Querying the devices is costly in time and should be done sparingly

**Uses**: Pulls information from the DHCP server allowing for updating of IP/MAC maps and adding new hosts to the databases

function **getStat**($fctComputer)

**Returns**: Array of port statistics **Arguments**: Computer Object

**Notes**: Returns an array with the following indexes: MTU, BW, delay, reliability, transmitLoad, receiveLoad, duplex, speed, 5minuteInputBS, 5minuteInputPS, 5minuteOutputBS, 5minuteOutputPS, inputByte, inputPacket, outputPacket, outputByte and description. This should be run sparingly as it is costly in time.

**Use**: Retrieves statistics about a particular host from the switch

function **getStatMultiComp** ($fcnComputers)

**Returns**: 2 dimensional Array of port statistics **Arguments**: Array of Computer Object

**Notes**: Returns a 2 dimensional array of statistics with the following indexes: MTU, BW, delay, reliability, transmitLoad, receiveLoad, duplex, speed, 5minuteInputBS, 5minuteInputPS, 5minuteOutputBS, 5minuteOutputPS, inputByte, inputPacket, outputPacket, outputByte and description. This should be run sparingly as it is costly in time.

**Use**: Retrieves statistics about a particular host from the switch

function **getOutput**($fctOutput)

**Returns** String **Arguments**: Strings

**Notes**: Outdated function. GetStats and getstatsMultiComp return the arrays that are accessible through this function.

**Use**: Retrieves values in the output array, using the argument as an index.

function **getInterface**()

**Returns**: Nothing **Arguments**: None

**Notes**: Queries the device and obtains IP addresses that are assigned to the device.

**Use**: Obtains associated with the device for use in networkRedirect path

function **getNextHop**($fcnIP)

**Returns**: String **Arguments**: String

**Notes**: Queries the device for the next hop IP Address of the argument address.

**Use**: Backtracking/adding devices that need to be updated to redirect a user’s traffic.

# class database

        function **database** ($fctUsername, $fctPassword, $fctAddress, $fctDbName)

**Returns** Nothing                                                      **Arguments**: String, String, String, String

**Notes**: Creates the DB class and initializes the connection

**Use**: Class constructor

        function **connect** ()

**Returns**: Nothing **Arguments**: None

**Notes**: Connects to the mySQL server. If the connection is unsuccessful a die command is issued.

**Use**: Called during the database class set, should not be called outside of the database class

        function **multiQuery** ($fctMultiQuery)

**Returns:** Array of results                                              **Arguments**: Array of Strings

**Notes**: Calls the query function in a looped fashion, taking the SQL result objects and pushing them to the end of an array

**Use**: Issuing a series of SQL commands to the database, typically only called from within the class.

function **query** ($fctQuery)

**Returns:** SQL results object                                      **Arguments**: String

**Notes**: Issues the provided command to the SQL database returning an SQL result object

**Use**: Issues SQL commands to the database, typically the multiQuery function is called.

function **close** ()

**Returns:** None  **Arguments**: None

**Notes**: Closes the SQL connection

**Use**: Closes the SQL connection

function **PRINTDB**()

**Returns:** String **Arguments**: None

**Notes**: Prints/Returns DB IP address

**Use**: Development/Debugging

function **addComputer** ($fctComputer)

**Returns:** SQL Result object  **Arguments**: Computer Object

**Notes**: Adds a computer object to the SQL DHCP table, no checks are performed. Checks should be performed by the caller.

**Use**: Called in the updateComputers function. Should not be called outside the class

function **updateComputers** ($fctComputers)

**Returns:** Array of Computer Objects **Arguments:** Array of Computer Object

**Notes:** Algorithm to detect and update changes within the network hosts. Firsts completes is update of the DHCP table, once complete, the function grabs additional information that is not yet present in the provided array (IE. device Name) from the DHCP table. This information is then added to the array. The updated array is then returned.

**Use:** Updates/returns a complete version of hosts with a valid DHCP binding.

function **getComputerMAC** ($fctMac)

**Returns:** Computer Object **Arguments:** String

**Notes:** Searches the SQL database for a host matching the MAC address provided. A Computer object is then initialized/populated based on the information found within the SQL table

**Use:** Obtain the Computer Object of a host based on its MAC address.

function **getComputerIP** ($fctIP)

**Returns:** Computer Object **Arguments:** String

**Notes:** Searches the SQL database for a host matching the IP addresses provided. A computer object is then initialized/populated based on the information found within the SQL table.

**Use:** Obtain the Computer Object of a host based on its IP address.

function **updateComputer** ($fctComputer)

**Returns:** Nothing **Arguments:** Computer Object

**Notes:** Takes a computer object and updates its MySQL row. The MAC addresses is used as a unique identifier (IE. Primary key).

**Use:** Update device information, IE IP address changes.

function **formatIP** ($fcnIP)

**Returns:** String **Arguments:** Array of Strings

**Notes:** Takes an array of strings as input and concatates them into a single string where each element of the array is separated by a colon (:). Will be switching over to a json encode/decode scheme.

**Use:** For storing IP information within MySQL tables.

function **addNetworkDevice** ($fcnDevice)

**Returns:** Nothing **Arguments:** IOS\_Device Object

**Notes:** Creates a new MySQL entries for a supplied IOS\_Device Object.

**Use:** Used to populate the networkDevice MySQL table.

function **getNetworkDevice** ($fcnIP)

**Returns:** IOS\_Device Object **Arguments:** String

**Notes:** Searches the MySQL networkDevice table to find a matching IP addresses and returns a initialize populated IOS\_device based on the information obtained from the MySQL query.

**Use:** Obtains the IOS\_device object based on the supplied IP address.

function **cacheCmd**($fcnID,$fcnCmd,$fcnData,$fcnTimestamp)

**Returns:** MySQLi result object **Arguments:** String, String, String, String

**Notes:** Checks if the command has already been stored, if so update the command by updating the data and timestamp. If the command is currently not cached, then a new row is added/populated for the cached command.

**Use:** Updates the caching table (netCache).

function **cacheCheck**($fcnID, $fcnCmd)

**Returns:** String/Boolean **Arguments:** String, String

**Notes:** Checks if a cached version of the command with the corresponding ID exists. If the cache entry exists it returns a string with the corresponding data. Otherwise false is returned.

**Use:** Retrieves cached information.

function **cleanCache**($fcnTimestamp)

**Returns:** MySQLi result object **Arguments:** String

**Notes:** Clears all cached commands who’s timestamp is greater then the current, minus the predefined expiration time.

**Use:** Removes stale cached entries.

# class ssh

function **ssh** ($fcnUsername, $fcnPassword, $fcnAddress)

**Returns:** Nothing **Arguments:** String, String, String

**Notes:** Initilizes the class

**Use:**

function **setUsername** ($fcnUsername)

**Returns:**  **Arguments:**

**Notes:**

**Use:**

function **setPassword** ($fcnPassword)

**Returns:**  **Arguments:**

**Notes:**

**Use:**

function **setAddress** ($fcnAddress)

**Returns:**  **Arguments:**

**Notes:**

**Use:**

function **setPort** ($fcnPort)

**Returns:**  **Arguments:**

**Notes:**

**Use:**

function **getPort** ()

**Returns:**  **Arguments:**

**Notes:**

**Use:**

function **getAddress**()

**Returns:**  **Arguments:**

**Notes:**

**Use:**

function **getUsername**()

**Returns:**  **Arguments:**

**Notes:**

**Use:**

function **getError** ()

**Returns:**  **Arguments:**

**Notes:**

**Use:**

function **PRINTSSH** ()

**Returns:**  **Arguments:**

**Notes:**

**Use:**

function **connect**()

**Returns:**  **Arguments:**

**Notes:**

**Use:**

function **close** ()

**Returns:**  **Arguments:**

**Notes:**

**Use:**

function **cmd** ($cmd)

**Returns:**  **Arguments:**

**Notes:**

**Use:**

function **multiCmd**($cmd)

**Returns:**  **Arguments:**

**Notes:**

**Use:**

# class redirect

function **redirect** ($fcnComputer, $fcnSource, $fcnDestIP, $fcnDb)

**Returns:**  **Arguments:**

**Notes:**

**Use:**

function **setDb** ($fcnDb)

**Returns:**  **Arguments:**

**Notes:**

**Use:**

function **getDb** ()

**Returns:**  **Arguments:**

**Notes:**

**Use:**

function **setComputer** ($fcnComputer)

**Returns:**  **Arguments:**

**Notes:**

**Use:**

function **getComputer** ()

**Returns:**  **Arguments:**

**Notes:**

**Use:**

function **setSource** ($fcnSource)

**Returns:**  **Arguments:**

**Notes:**

**Use:**

function **getSource** ()

**Returns:**  **Arguments:**

**Notes:**

**Use:**

function **setDestIP** ($fcnDestIP)

**Returns:**  **Arguments:**

**Notes:**

**Use:**

function **getDestIP** ()

**Returns:**  **Arguments:**

**Notes:**

**Use:**

function **getDevicePath**()

**Returns:**  **Arguments:**

**Notes:**

**Use:**

function **getNextHopPath**()

**Returns:**  **Arguments:**

**Notes:**

**Use:**

function **findPath** ()

**Returns:**  **Arguments:**

**Notes:**

**Use:**

function **applyPath** ()

**Returns:**  **Arguments:**

**Notes:**

**Use:**

function **removePath**()

**Returns:**  **Arguments:**

**Notes:**

**Use:**

# class myscripts

function **request**(cmd, id)

**Returns:** nothing **Arguments:** String, String

**Notes:** Adds the Ajax response into the specified HTML element (By ID); cmd is specified by the specific API commands

**Use:** table population, visual return of data

function **requestSp**(cmd, fct)

**Returns:** nothing **Arguments:** String, String

**Notes:** Adds the Ajax response into the specified function. cmd is specified by the specific API commands

**Use:** Login/Logout, Setting variables

function **publish** (responseText)

**Returns:** ajax response **Arguments:** String, String

**Notes:** Adds the Ajax response into the specified function. cmd is specified by the specific API commands

**Use:** Login/Logout, Setting variables

function **sendRequest** ()

**Returns:**  **Arguments:**

**Notes:**

**Use:**

function **pageLoad** ()

**Returns:** nothing **Arguments:**

**Notes:** currently not used. Used to be used

**Use:** call the page load

function **loginFunction**()

**Returns:**  **Arguments:**

**Notes:** Checks for username and password sends ajax request. Response sent to loginVerify().

**Use:** Starts the login process.

function **loginVerify**(login)

**Returns:**  **Arguments:** ajax response array

**Notes:** Verifies credentials and sets cookies, enables buttons.

**Use:** Do not call, used by loginFunction()

function **logoutFunction**()

**Returns:**  **Arguments:**

**Notes:** sets values to their default

**Use:** wipes log on data

function **register**()

**Returns:**  **Arguments:**

**Notes:** Allows for user registration. Grabs text input and sends a register request. Response is sent to registerVerify();

**Use:** user registration

function **registerVerify**()

**Returns:**  **Arguments:**

**Notes:** confirms user registration

**Use:**  Do not call, used by register()

function **removeComp**()

**Returns:**  **Arguments:**

**Notes:** Removes currently selected PC from active directory. Sends ajax request

**Use:** remove a pc

function **addComp**()

**Returns:**  **Arguments:**

**Notes:** Sends ajax request for addComp.

**Use:** add a PC

function **redirectFunction**()

**Returns:**  **Arguments:**

**Notes:**

**Use:** Redirect current selected PC

function **tableLoadComplete**()

**Returns:**  **Arguments:**

**Notes:** called by tablePop()

**Use:** watches for user click for table selection

function **tablePop**()

**Returns:**  **Arguments:**

**Notes:**

**Use:** Populates tables

function **detailPop**()

**Returns:**  **Arguments:**

**Notes:** called by tableLoadComplete();

**Use:** Populates detail table.

function **compName**()

**Returns:**  **Arguments:**

**Notes:**

**Use:** Set the current PC name

function **setCookie**(cname,cvalue,exdays)

**Returns:**  **Arguments:** string, string, integer

**Notes:** takes a variable name, variable value and cookie expiration time in days.

**Use:** create a cookie

function **getCookie**(cname)

**Returns:**  **Arguments:** string

**Notes:**

**Use:** gets the cookie of the specified name

function **checkCookie**()

**Returns:**  **Arguments:**

**Notes:**

**Use:** check for cookies