Installation & Operating Guide

OMS Series Matrix Switch
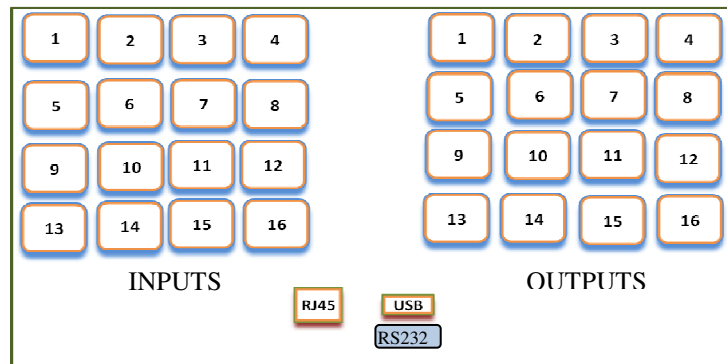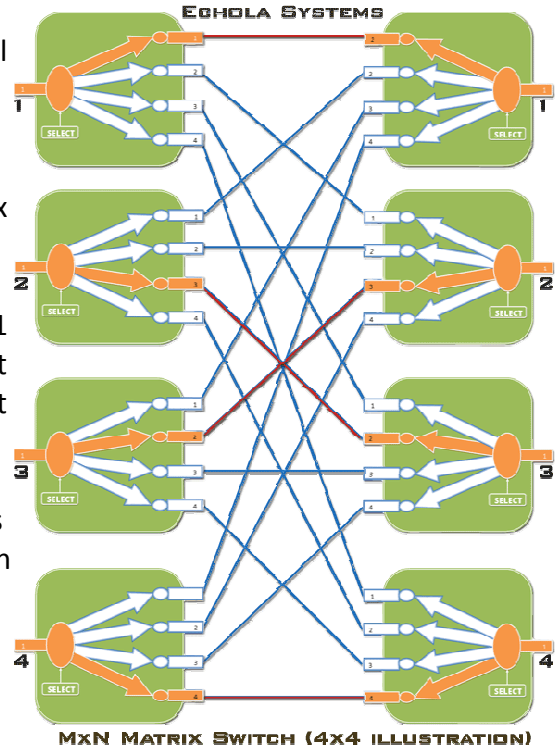
Installation

The OMS series switch is a non blocking MxN all-optical cross-connect switch that allows light to be switched from any input port to any output port.

The figure on right side illustrates how a 4x4 matrix switch works. The 4x4 matrix is essentially eight 1x4 switches connected as shown. It shows simple config where the switch is programmed to connect input port 1 to output port 1, input port 2 to output port 3, input port 3 to output port 2 and input port 4 to output port 4. But it can be programmed to switch light to any input port to any output port through GUI or CLI command. The switching time is about 1 second. The typical applications include patch panel/test lab automation. We offer both single-mode and multi-mode versions of the switch.

MxN Matrix Switch (4x4 illustration)

It comes in different port configurations but above figure on right shows a 16x16 for illustration purpose. There are 16 input optical connectors on left and 16 output optical connectors on right. You can connect up to 16 single strand optical fibers to left connectors where you want the light to be switched from and similarly 16 on the output side where light to be switched to. If you want to connect both Rx/Tx pairs then you can connect up to 8 such pairs on input side and 8 on output. Even though ports are grouped as input and output ports they are bi-directional ports which means the input and output can be swapped.

OMS series switch provides a serial (RS232) and an Ethernet (RJ45) port connectivity for management. Serial port is only used in special situations such as to debug network connectivity if OMS series switch is not reachable through Ethernet.

Configuring OMS series switch

You might require a Laptop or a PC to configure OMS series switch with an IP address before connecting to your network. You can use either serial or Ethernet port to configure IP. If you use serial port using supplied null modem cable then you would need to configure serial for **38400 baud rate with 8-N-1** to access OMS series switch. If you use Ethernet then you would need to configure the PC or Laptop's IP to match OMS series switch's default network. The default network configuration and Login information is as follows

> **IP Address: 10.1.1.100**
> **Mask: 255.255.255.0**
> **Gateway: 10.1.1.1**
> **DNS: 10.1.1.1**
>
> **"root" password:** *omsctl*

Other non-root default user:

> **Username:** *omsctl*
> **Password:** *omsctl*

Configuring Static IP

The management module runs an embedded version of Linux, so anyone familiar with Linux will find it easy to use. But we provide a CLI command which can be used to configure and use most of the features. The name of the CLI is "omsctl". So, to configure a static IP address you can use following syntax.

> **# omsctl -i 192.168.1.10 -m 255.255.255.0 -g 192.168.1.1**

> "**omsctl -?**" shows detailed omsctl command options with examples.

The above command configures IP address of OMS series switch as 192.168.1.10 with network mask 255.255.255.0 and gateway & DNS as 192.168.1.1. Once the IP is configured from a PC or a Laptop using Ethernet or serial port, you can then connect OMS series switch to your network and access it using "telnet" or "ssh".

Configuring Dynamic IP

If you have a DHCP server running on your network and you may want to dynamically assign an available IP address to OMS series switch by using following option.

> **# omsctl -D**

Make sure you know the assigned IP address to login using "telnet" or "ssh".

Configuring Hostname

You can also change the hostname of OMS series switch switch using "omsctl".

omsctl −h OMS500-SW-1

The above command changes hostname of OMS series switch to OMS series switch-SW-1.

Configuring DNS/Nameserver

To configure a DNS or Nameserver,   you can use "-n" option of omsctl.

    # omsctl −n 192.168.1.11

Configuring Syslog Server

To send system generated events to an external syslog server, use "−S" option of omsctl. You have to specify the address of the syslog server which will receive these event logs. (Note: this option only available in software version 2.0 and above).

    # omsctl −S 192.168.1.200

Version Info

The following command shows hardware and software versions and serial number of the switch.

    # omsctl −v

Other Administrative commands

Most of other administrative functions can be done using standard Linux commands. For example, to change the password use "passwd" command from the Linux prompt and to add new user use "useradd" command. You have to be "root" user in order to add a new user.

Operation

To simply the operation, all functions are provided in a single command called "omsctl". omsctl provides 3 major functions

1. Network & Host configuration (allowed only as "root" user)
2. Switch Control & Status
3. Port group Management

Network & Host configuration

The following are the commands used to configure network.

1. To configure static IP
   # **omsctl -i \<ip\> -m \<mask\> -g \<gw\>**
2. To use DHCP (dynamic IP)
   # **omsctl -D**
3. To configure hostname alone
   # **omsctl -h \<hostname\>**
4. To configure a DNS or Nameserver,   you can use "-n" option of omsctl.
   # **omsctl –n \<dns_server_ip\>**
5. To configure syslog server to receive OMS series switch's events
   # **omsctl –S \<syslog_server_ip\>**

For details on above commands refer to above installation section.

Switch Control

In order to cross connect an input port to an output port or a group of ports, the following command can be used.

$ **omsctl -p { \<input_port#\> \<output_port#\> } | \<port_group\> [-t \<secs\>]**

Note that the "{}" (braces) groups options and "|" is equivalent to "or". If the options are in square brackets "[]" then it is optional.

For example:
> $ **omsctl -p 1 4**
> > ➔ switch input port 1 to output port 4. Note, there should be space between p,1 and 4.
> $ **omsctl -p config1**
> > ➔ switch ports mentioned in group named config1. Refer to port group management section for creating port groups.
> $ **omsctl -p 2 1 -t 120**
> > ➔ switch input port 2 to output port 1 after 120 secs (delayed switching)

Switch Status

Following command shows the status of ports, which input port is mapped to which output port or status of ports in a pre-defined group.

$ **omsctl -s [<input_port#>]**

For example:

$ **omsctl -s**
➔ shows all ports status as follows. It shows which input port connected to which output and also shows the active group.

```
        All Ports Status
=========================
  In Port  |  Out Port
=========================
    1       |    4
    2       |    1
    3       |    5
    4       |    12
    5       |    14
    6       |    2
    7       |    6
    8       |    7
    9       |    9
   10       |    11
   11       |    10
   12       |    8
   13       |    13
   14       |    3
   15       |    16
   16       |    15
------------------------------------
     Active Group: config1
------------------------------------
```

$ **omsctl -s 2**
➔ shows which output port the port2 is cross connected to.
   $ 1

Port group Management

Port group management commands provide convenience of switching bunch of ports together identified by a name. Group name can be any alpha-numeric name. Group name can not be just a number or start with a number. For example group name can not be "10" or 10gp1.

1. To create a new group
   $ **omsctl -c <port_group> [-f <map_file>]**
2. To update existing group
   $ **omsctl -u <port_group> {-a | -r <input_port#> <output_port#>}**
3. To delete a group
   $ **omsctl -d <port_group>**
4. To delete all groups
   $ **omsctl -R**
5. To list ports in a group
   $ **omsctl -l [<port_group> | <input_port#>]**

For example:
   $ **omsctl -c config1**
   ➔ creates empty group named "config1"

   $ **omsctl -u config1 -a "1 4"**
   ➔ add a port map where input port 1 would connect to output port 4 to config1 group
   ➔ if you want to add may port maps at a time, use **-f** option as shown in one of following example.

   $ **omsctl -u config1 -a "2 1"**
   ➔ add a port map where input port 2 would connect to output port 1 to config1 group

   $ **omsctl –l config1**
   ➔ show the port map for the port group config1
```
========================
        In Port  | Out Port
========================
        1        |   4
        2        |   1
-----------------------------------
```
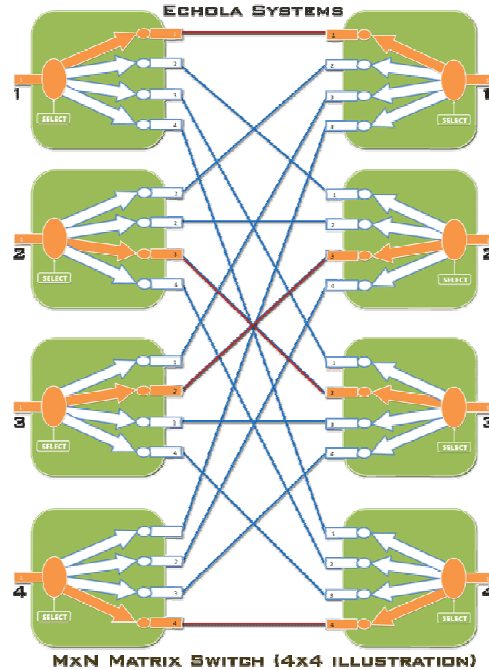 $ **omsctl -u config1 -r "1 12"**
   ➔ replace input port 1 entry in the port map of config1 to make now connect to output port 12 instead of previous 4

   $ **omsctl –l config1**
```
========================
        In Port  | Out Port
========================
        1        |   12
        2        |   1
-----------------------------------
```

The following example shows how to create group with multiple port maps in one go. Let use the following 4x4 configuration for illustration purpose.

MxN Matrix Switch (4x4 illustration)

$ omsctl -c config1 -f testconfig.txt

➔ create port group config1 with port maps defined in the file testconfig.txt. you can use **vi** editor and create this file. In the above example, the input port 1 is connected to output port1, input port 2 is connected to output port 3, input port 3 is connected to output port 2 and input port 4 is connected to output port 4. So the file testconfig.txt would have 4 lines with input and corresponding output ports separated by space as shown below.

```
1   1
2   3
3   2
4   4
```

Now listing the config1 would show

$ omsctl –l config1

```
========================
    In Port  |  Out Port
========================
     1       |    1
     2       |    3
     3       |    2
     4       |    4
----------------------------------
```

Now if you want to apply this group so that it connect all ports as per this group, just issue the command **omsctl -p config1**.

$ **omsctl -d config1**
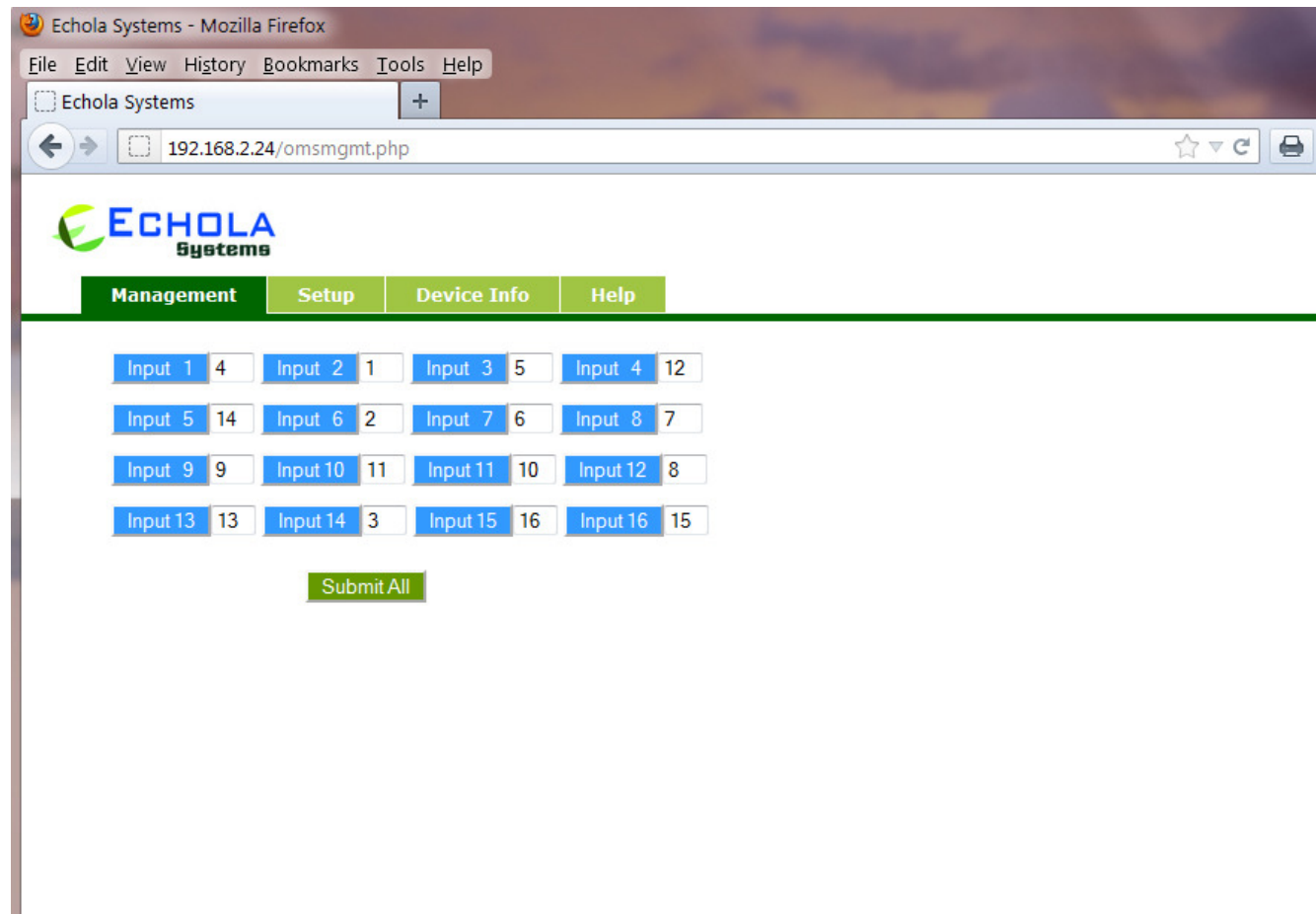➔ deletes group config1

$ **omsctl –R**
➔ delete all groups in the database.

Web interface

In order to access OMS series switch web interface you will have to use its IP address in URL address bar of the web browser. The first page you will see is the management page where you control all ports. Web interface has a main menu tab with following pages…

Management page

The management page of OMS series switch allows controlling the matrix. You can either individually switch each input port to output port or all by using 'Submit All' button at once.

Setup page

Setup page allows you to change Hostname, Static IP address configuration and Dynamic IP (DHCP). Initially it shows the current set values. You can change them by clicking on the corresponding field. Note that the changing IP, Gateway, Mask or DHCP setting would make switch to reboot!

Device Info page

Device Info page shows hardware and software information such as model and version.

Help page

This page shows the complete user's manual.

Automating OMS Switch using TCL scripts

In order to automate Echola's Layer 1 switches you would need to write tcl/expect based script. The tcl and expect scripting languages are easy to learn. We have given an example script written for OMS series switch at the end which you can modify to suit your need. There are tons of online sources for learning tcl & expect. The following provides quick high level overview of tcl and expect http://cplug.org/t/uploads/2009/02/tcl-expect.pdf. There is a good book from O'Reilly which provides great insight into expect language itself: "Exploring Expect: A Tcl-based Toolkit for Automating Interactive Programs (Nutshell Handbooks)".

Running scripts from Unix/Linux systems

If you want to run the script from a Unix/Linux based machines then there is possibility that you may be already having these tools on your system. Check if it's already been installed by typing "expect" from Unix/Linux prompt. If it is not then you will have to install it using package install tool for that particular flavor of Unix/Linux. For instance, on Fedore core Linux, you can use "yum install tcl expect" to install tcl and expect.

Running scripts from Windows

For windows based systems you can install windows free community version of ActiveTcl from Activestate http://www.activestate.com/activetcl/downloads. Expect is not available yet for 64bit version of Windows 7/Vista. So you will need to download 32bit version for ActiveTcl first and then make sure to install "expect" using command "teacup install Expect".
Also you need to enable "telnet" client on Windows before running any scripts. In order to enable telnet on Windows follow these steps

> ➔ Start
> ➔ Control Panel
> ➔ Programs And Features
> ➔ Turn Windows features on or off
> ➔ Check Telnet Client
> ➔ Hit OK

After that you can start Telnet via Command Prompt to check if it works.

The following sample script actually login into OMS series switch/fc811 switch and issue a switch command then check whether the switch command was successful and return the result before terminating the telnet session. This script takes argument (port number and state of the port (on/off)) from commands line argument. Cut and Paste the following script on to any editor and save as "romsctl". Then you can run the script by issuing romsctl -p <port#> on|off. For instance, to switch port 2 to ON, you can call script as romsctl -p 2 on. Make sure you have proper path set for expect on first line "#!/usr/bin/expect" for Unix/Linux based systems. For windows you will have to uncomment ' exec' and 'package' commands as mentioned in the script. All comments inside '#' provide more info on what the script is doing.

The Sample Tcl/Expect script

```
#!/usr/bin/expect
################################################################################
# This script switches the given port and verifies if the port is switched from a remote machine
# that's why its called romsctl!
#          Command Usage: romsctl -p <input_port#> <output_port#>
################################################################################
# For windows uncomment following
#   exec tclsh "$0" ${1+"$@"}
#   package require Expect

# Check number of arguments passed to this command if < 3 then spit out error & exit

    if { $argc < 3 } {
        puts "Usage: romsctl -p <input_port#> <output_port#>\n"
        exit 1
    }

# Set telnet host, username, password and other parameters, modify these to reflect your setup

    set hostname "192.168.2.20"
    set username "omsctl"
    set password "omsctl"
    set prompt "omsctl@.*\$"
    set input_port [lindex $argv 1]
    set output_port [lindex $argv 2]
    set commandcontrol "omsctl -p $input_port $output_port"   # use the CLI to control
    set commandstatus "omsctl -s $input_port "                # use the CLI to check status

# Display info.

    puts "Connecting to $hostname."

# Connect to the telent server using the "spawn" command.

    spawn telnet $hostname
    #spawn C:\Putty\putty.exe -telnet $hostname


# Wait for a login prompt.

    expect -re "(Name|login|Login|Username).*:.*" {

      # Login prompt received. Send user name  to OMS series switch.

      send "$username\r"

    } eof {

      # No login prompt received. Display an error.

      puts "could not connect\n"
    }
```

```
# Wait for a password prompt from the Unix server.

    expect "Password:" {

        # Password prompt received. Send the password.

        send "$password\r"
    }

# Wait for the switch prompt.

    expect -re $prompt {

        # Issue omsctl command to switch given port

        send "$commandcontrol\r"
    }

# Wait for the switch prompt again to check status.

    expect -re $prompt {

        # Issue omsctl command to check status

        send "$commandstatus\r"
    }

# Discard echoed command - we need only the status

    expect "$commandstatus\r"

# Discard unwanted prompt as well

    expect -re "(.*)$prompt"

    #Debug
    #puts "\nGOT*****$expect_out(buffer)**************\n"
    #puts "\n GOTS ####$expect_out(1,string)#####\n"

# Save remaining to buffer 'data'

    set data $expect_out(1,string)

# Check return status and display result accordingly
    switch -re $data {
        $input_port { puts "Input Port $input_port is connected" }
        default { puts "Port $input_port is not connected" }
    }

# Terminate telnet

    send "exit\r"
```

OMS Series Switch Specifications

OMS Specifications	Min.	Typical	Max.
Operation Wavelength (nm)	350	1260~1650	1800
Optical Interface	LC Connectors		
Insertion Loss (excluding connectors) (dB)	0.7	1	2
Cross Talk (dB)	70		
Switching Speed (sec)			1
Durability (cycles)	10000000		
Polarization Dependent Loss (dB)		0.04	0.2
Wavelength Dependence Loss (dB)		0.1	0.3
Return Loss (dB)	45		
Repeatability (dB)		0.1	0.3
Power Consumption (Watts)	3.5	7	10
Power Supply (V)	100		240
Storage Temperature (°C)	-40		85
Switch Type	Latching		
Electrical Interface	RS232/USB/Ethernet		
Software	Web Interface/Telnet/SSH/Syslog		

Standard Configurations

Model	Ports	Ports in Tx/Rx pairs	Size
OMS64	32 x 32 = 64	16 x 16 = 32	2RU
OMS128	64 x 64 = 128	32 x 32 = 64	3RU
OMS192	96 x 96 = 192	48 x 48 = 96	4RU
OMS256	128 x 128 = 256	64 x 64 = 128	6RU
OMS384	192 x 192 = 384	96 x 96 = 192	8RU
OMS500	250 x 250 = 500	125 x 125 = 250	10RU

Troubleshooting

➔ If you use telnet and is very slow, it could be due to name server / DNS issue. Make sure you have name server configured correctly (-n option). Try "ping <name_server_ip>" to see if name server configured can be reached from fc811. If you don't have dns on your network then you can just remove file "/etc/resolv.conf" and see if it works normal.

➔ All the group information is stored in a hidden XML database file. If for some reason this file is corrupted, the system will recover from this error by trying to copy the backup database file. If this happens it will throw a warning message but it is not guaranteed that all the group information will be restored correctly. In that case you may need to recreate missing groups.

➔ For adding new users Linux's "useradd" command can be used. You have to be "root" in order to add new user to the system.

Contact Info

If you have any technical questions and need help you can send email to support@echola.com or call us at 408-321-9663 ext 2. You can also download latest documents and software from our website www.echola.com/optical.