

MPD Series RCM Software[™] for Switched Outlets

Owner's Guide and Reference Software Version 1.0.x

July 2013 : P/N 501014-000



© 2013, Marway Power Systems, Inc. All rights reserved.

RCM Software[™] is a trademark of Marway Power Systems, Inc. MPD Series[™] is a trademark of Marway Power Systems, Inc. All other trademarks are property of their respective owners.

Global Support Contacts

Web: www.marway.com Email: support@marway.com sales@marway.com Phone: 800-462-7929 (7am-5pm PST)

There may be updates to this documentation and the software it describes at: www.marway.com/mpd/rcm_software



MPD Series RCM Software

Owner's Guide and Reference Software Version 1.0.x

Contents3
Getting Started7
Initial TCP/IP Configuration
Feature Tour10
Web Browser Interface10
Command Line Interface10
Power Terms
Browser Interface Tour11
Main Menu
Web Dashboard12
Outlet Control12
Outlet Settings13
User Settings14
Authentication14
Access Methods14
Personal Profile14
Permissions14

Network Settings	15
System Settings	. 16
Command Line Interface Tour	17
Login	17
Built-in Help	18
Command Syntax	18
Aliases and Variants	19
FTP and the File System	20
Software Operation	22
Login	. 22
HTTP/HTTPS ("Web") Login	. 23
Startup Notices	23
Documentation and Support	. 23
Telnet Login	24
SSH Login	24
Viewing Outlet Status	. 25
Outlet Status on the Web Dashboard	25
Outlet Status by Command Line	. 25
Outlet Control	26
Outlet Switching and Adjustable Delays	. 26



User Management
Root User
Authentication
Passwords in the Settings File
Authorization34
Access Methods
Permissions
Auditing
Activity Logging
Adding Deleting Editing Licers 26
Adding, Deleting, Editing Osers
Software Settings
Adjust Settings with the Web UI
Adjust Settings with the CLI
Adjust Settings in the Files
System Settings40
Network Settings41
User Settings44
Outlet Settings49
Labels
Outlet Switch Delays49
Outlet Startup State49
Command Line Reference52
Built-in Help52
Command Syntax

Command Aliases53
Command Variants53
Outlet Commands54
getOutlet54
setOutlet54
User Commands55
getProfile (a.k.a getMy)55
makeLoginPswd55
setProfile (a.k.a setMy)55
getUser56
setUser56
addUser57
deleteUser57
Network Commands58
getTcp58
setTcp58
getHttp59
setHttp59
getHttps60
setHttps60
getSntp61
setSntp61
getFtp62
setFtp62
Log Commands63
getLog63
getStartupLog63

System Commands
RESTful API Reference
What is REST and RESTful API? 66 The Gist of REST 66 Programming for a RESTful API 67 Why Use the RESTful API? 67
HTTP Verbs67
Responses
Authentication API69
API Resources69
/session69
POST /session69
DELETE /session69
/outlet70
GET /outlet/:id/label70
GET /outlet/:id/switch70
GET /outlet/:id/rated_amps70
GET /outlet/:id/rated_volts71
GET /outlet/:id/startup_state71
GET /outlet/:id/on_delay71
GET /outlet/:id/off_delay72
GET /outlet/:id/cycle_delay72
PATCH /outlet/:id/switch72
PATCH /outlet/:id/startup_state73
PATCH /outlet/:id/on_delay73



PATCH /outlet/:id/off_delay73
PATCH /outlet/:id/cycle_delay74
/system74
GET /system/1/label74
GET /system/1/location74
Settings Files Reference76
Know What You're Doing76
Accessing Settings Files76
File Format and Syntax77
Name/Value Pairs77
Objects and IDs77
Object Groups of Settings78
File Identity and Version78
systemSettings.txt79
networkSettings.txt79
userSettings.txt79
powerSettings.txt80
factorySettings.txt81
Firmware Updates83



MPD Series RCM Software Quick Start



Getting Started

You will need:

- An Ethernert LAN with DHCP services enabled.
- The Marway PDU connected to the LAN above.
- A computer connected to the LAN above.
- The RCM Finder.exe software installed—which should have accompanied the PDU, or which can be downloaded from Marway's web site at: http://www.marway.com/mpd/rcm_software

Initial TCP/IP Configuration

Configuring TCP/IP the first time, or any time the network settings of the PDU are not known, must be done using the following procedure:

- Power up the PDU.
- Run the RCM Finder.exe application mentioned above. It should automatically start a scan for devices on the network.
- One or more devices should appear in the list. Look for those identified as Marway RCM. If there is more than one Marway RCM device, each PDU will be differentiated by the MAC address shown in the list, and identified on a label on the PDU.
- Having identified the IP address of the PDU, enter that address in a web browser (e.g. http://192.168.1.10/) and open that site.
- Enter the factory default user account credentials to log into the PDU web site.
- After the Dashboard page loads, click the Network link at the top of the web page.

	Linuu	[1	
	IP Address	MAU Address	Name	Product DCM
Configure IP Settings	132.166.1.10	00:40:30:43:35:37		maiway num
Device Info				
Behont Device				
Refresh List				

FACTORY DEFAULT USER ACCOUNT

Account: root

Password: pP8*word

TCP/IP NETWORKING IS ON BY DEFAULT.

It is highly recommended, for security purposes, that the root user password be updated before or immediately when the system is connected to a network. When possible, use an Ethernet connection local to a single computer, or private LAN, to do the initial setup.

If the PDU is connected to a network without the password being updated, the PDU may be accessible by any person or script familiar with the default password of Marway's PDUs which may result in unintended and unauthorized access.



- Use the Network page to change protocol setups as needed. It is highly recommended that IPv4 DHCP be disabled, and that a manual IPv4 address be used so that it will always be known when logging into the PDU by web or by command line.
- Save the form.
- Open the System web page, and click the System Restart button.
- Change the IP address in your browser to use the PDU web site at its new address.

Change root User Password

The factory default password should be changed immediately. Log into the PDU using the factory default password, then use the web browser interface or the command line via Telnet or SSH to change the password of the user named root.

Using the web interface, log into the PDU, open the Users list from the main menu, then edit the user named root.

Using the command line interface, log into the PDU using Telnet or SSH, then execute the command setUser root password "abc" where abc is replaced with the new password.

Reset Password to Factory Default

If the system has been configured before, and the root password or any other password with full privileges is not known, the system will have to be reset. To do this, first find the reset button near the Ethernet port. There will be a pinhole requiring a paper clip or a small tipped object to push the button behind the panel. Power down the PDU, push and hold the reset button, power up the PDU, and hold the reset button for 30 seconds. Wait another 30 seconds, then log into the PDU using the factory default password.

Time Servers (SNTP)

The factory default SNTP servers are public volunteer servers known to work at the time of shipment, but are not run by Marway, and are not guaranteed to be available. Check with your IT administrator to see if there's an alternative preferred set of servers to use.

TimeZone Settings

For details on time zone settings, find the settings in the section "Network Settings" on page 41.



MPD Series RCM Software

Feature Tour



Feature Tour

Marway's RCM Software for MPD Series PDUs offers multiple interfaces to monitor and/or control the unit:

- a web browser based graphical interface ("web UI")
- a command line interface ("CLI") over Telnet or SSH
- a RESTful API over HTTP for automated outlet control

Additionally, a utility named RCM Finder.exe (for Microsoft Windows[®]) is used for the first-time setup of TCP/IP settings. See "Initial TCP/IP Configuration" on page 7 for details.

This section of the *User Guide* is an introductory tour of the major features of the web UI and CLI. It presents screenshots along with short descriptions of several of the interface and operation features of the RCM Software.

Web Browser Interface

The browser interface has been designed for reasonably broad compatibility with several of the most commonly used browsers in versions released within the past couple of years. If you notice any specific compatibility problems, please report them to Marway support.

The browser interface can be used over HTTP or HTTPS. It allows for remote relay switching, user management, and editing all adjustable settings of the PDU. Note that many UI elements (fields and buttons) will display help tips if you let the mouse pointer hover over the element for a second or two. Additionally, many display panels on the web pages have their own help button on the right side of the panel title bar.

Command Line Interface

The command line interface works over Telnet and SSH. It allows for remote relay switching, user management, and editing all adjustable settings of the PDU.

Power Terms

In Marway's RCM products, the following terms are used throughout the documentation and interfaces.

- Outlet : a female power connector used to provide point-of-use power for the user's application.
- Circuit : a branch in the power distribution protected by a circuit breaker or fuse (or the whole unit if there is no breaker or fuse).
- Line : a pair of conductors (wires) across which exists a voltage potential.
- Input : a single conductor on which a current flows.

In Marway's products, Lines and Inputs are the wires directly connected to the power feed which we call the power Source. A PDU will have one or more power Sources. Each one is made of Lines and Inputs. Each Source may be branched into one or more Circuits, and each circuit would have one or more Outlets (with a variety of possible connector styles).



Browser Interface Tour

Login

The login page is fairly straight forward. One noteworthy feature aside from the obvious login functionality is the Startup Notices panel. Any errors or warnings generated during startup will be listed in this panel. If startup went as expected, the "OK" will be displayed as the screenshot shows.

Main Menu

After the login page, the web browser interface is organized into the following major sections which are labeled in the browser's main menu tabs as:

Data Viewing

- Dashboard : provides outlet control. Other than logs, Dashboard is where virtually all information viewing is performed. All other main menu tabs are primarily for adjusting settings.
- Logs : display of the system and startup logs

Settings Adjustments

- Outlets : settings for outlet labels, switching delays, and startup state
- Users : settings for user profiles, authentication, and authorization
- Network : settings for supported networking protocols and features
- System : settings for unit labeling and clock, command for restart, and miscellaneous system-level information

	() http://testbay2-pdu.local/logir	explorer	√ × Societ √ × Societ	
Favorites	() PDU Login : Test Bay 2 PDU		👌 🔻 🔝 👻 📑 🖶 👻 Page 👻 Safety 🕶 To	ols 🔻
(MARWAY POWER SOLUTIONS			
Te Lo	est Bay 2 PDU ocation: R&D Building :	112		
L	ogin			
		Account Password	Login	
S	Startup Notices : OK		?	
C T	Documentation and Su hese links open new windows to	external resources.		
		Factory Support Contacts Email : support@marway.com Phone : 800-462-7929	Other Links and Downloads Product Info for MPD 99000-003 Marway Website	
			© 2013, Marway Power Solution	ns
			Internet Protected Mode: On	.00%





Web Dashboard

The Dashboard is where all remote outlet control is located. Other than logs, Dashboard is where virtually all information viewing is performed. All other main menu tabs are primarily for adjusting settings.

Below the main menu, the top of the dashboard includes a user-defined unit label and location on the left. These are defined on the System page, and allow the PDU to have a unique and meaningful identification. On the right, the Help and Logout buttons sit above a time stamp of when the viewed web page was last loaded. This top section of the page is common to all web pages.

Outlet Control

The web Dashboard offers three ways to manipulate outlet state. The first is an All Outlets set of buttons to affect all outlets with a single command to turn them on, turn them off, or cycle them.

The second method for switching outlets is the Action column of the Dashboard Outlets panel. Each outlet has a popup menu to choose On, Off, or Cycle. Each outlet can be set to any of these options. When all choices have been made, pressing the Apply button at the top of the column will issue each command in sequence.

The third method for switching outlets is the Switch column buttons which provide simple and immediate manual On and Off switching control.

In addition to outlet control, the Dashboard displays the electrical load specifications of the outlet, its panel name, and an arbitrary user-entered label to identify the purpose of, or equipment connected to, the outlet.

Test Bay	2 PC	U uilding 110			2 Logout
Location: F		unding 112			3417/2010 1110/14074
Outlets :	All				?
All Outlet	ts Show	n Below			Apply Actions
On	Off	Cycle			Apply
Switch	ID	Outlet Label	Panel Name	Rating	Action
• On	1	Utility Outlet	J1	90 to 125 VAC, 20 A max.	None ‡
• On	2	Control Panel & Logic	32	90 to 125 VAC, 20 A max.	None ‡
• On	3	Hydraulics Controller	J3	90 to 125 VAC, 15 A max.	None ‡
• On	4	M1 Motor Controller	34	90 to 125 VAC, 15 A max.	None ‡
• On	5	M2 Motor Controller	J5	90 to 125 VAC, 15 A max.	None ‡
• On	6	Battery Backup	J6	90 to 125 VAC, 15 A max.	None ‡
• On	7	DAQ Server	37	48 VDC, 30 A max.	None ‡
• On	8	DB Server	J8	48 VDC, 30 A max.	None ‡
Documon	tation	and Support			
These links of	pen new	windows to external resources.			
		Factory Support Co	ontacts	Other Links and Downloads	
		Email : support@ma	rway.com	Product Info for MPD 99000-003	



Outlet Settings

The main menu Outlets tab displays the adjustable settings for all outlets. This includes a descriptive label, delays for on, off, and cycle actions, and a startup state configuration. The Startup State option determines what the outlet state will be set to shortly after the PDU is powered up, or after a software restart.

🏉 PDU Outle	et Setting	gs : Test Bay 2 PDU - Windows Internet Ex	plorer							- •	×
G•	(†) ht	tp://testbay2-pdu.local/power					- 🛛 + 🗙	Google			• م
🚖 Favorite:	s (I)	PDU Outlet Settings : Test Bay 2 PDU					🖄 🔻 🖻) • 🖃 🖶 •	Page 🔻 Safety 🔻	Tools 🔻	0-
	(')	MARWAY POWER SOLUTIONS		C	Dashbo	ard	Outlets User	s Network L	ogs System		
l	Test Locati	Bay 2 PDU on: R&D Building 112						? Jul 17 20	Logout)	
	Outle	et Settings : All						? Revert	Save		
	ID	Outlet Label	Panel Name	S On	witch De in Secor Off	elays nds Cycle	Startup State				
	1	Utility Outlet	J1	0.2	0.0	5.0	Last Known 💌				
	2	Control Panel & Logic	32	0.2	0.0	5.0	Last Known 💌				
	3	Hydraulics Controller	33	0.2	0.0	5.0	Last Known 💌]			
	4	M1 Motor Controller	34	0.2	0.0	5.0	Last Known 💌				
	5	M2 Motor Controller	35	0.2	0.0	5.0	Last Known 🔻				
	6	Battery Backup	J6	0.2	0.0	5.0	Last Known 💌				
	7	DAQ Server	37	0.2	0.0	5.0	Last Known 💌]			
	8	DB Server	J8	0.2	0.0	5.0	Last Known 💌				
	Use thi configu same v outlets actuall Enter " Blank f	age All Outlets is form to set one or more ration fields of all outlets to the adue. Press Apply to update the form above. Values are not y saved until Save is pressed. to reset to a default value. fields are ignored.	Switch Delays in Seconds On Off Cycle	Startu State Choose	p •				Apply		
	Docu These	imentation and Support links open new windows to external re Factor Email : su Phone	y Support Contacts pport@marway.com a : 800-462-7929		Proc	Other Linl duct Info Marv	ks and Downloads for MPD 99000-00: way Website	3			
Done							Internet Prot-stad	© 2013, M	larway Power Soli	utions € 100%	
Done	_						Thernet Protected	wide: Un	 ₩ 	-4 100 %	



User Settings

Authentication

Authentication determines the credentials used to login into the PDU (via web or CLI). This is an account name and a password.

Access Methods

The Access Methods section determines whether the user can access the system via the web UI, the CLI, and the settings files. If the none of the options are selected, this effectively disables a user without having to delete him.

Personal Profile

This section allows an administrator to keep minimal contact information on board if needed. The Company and Job Role field can be used to for third-party vendors who may be allowed access perhaps for support.

Permissions

The RCM software does not assign permissions based on user "groups." Rather, each user can be assigned unique permissions to allow greater flexibility.

Show L	List A	dd User					
Users :	List						?
Edit Log	gin Name	User Name	Email	Access	Logins	Last Login	Del
roc 🖉	ot :	System Admin	sysmin@test.com	Web CLI FTP	104	2013-07-17 12:02:44	
les	ter	Lester Tester	les@test.com	Web	137	2013-07-12 13:32:10	





Network Settings

This page presents the settings details for all supported network protocols. Assuming you're familiar with these protocols, most settings should be self explanatory. One unique feature the RCM has is an auto-off for the FTP server. Since the need to have FTP on would be for rare access to the log file or settings files, FTP can be enabled, when needed and set to auto disable after a period time. It's a small convenience and potential security feature.

PDU N <u>E</u> dir	Network Settings : Test Bay 2 PDU t⊥View History <u>B</u> ookmarks	J - Mozilla Firefox Iools Help					×
	Network Settings : Test Bay 2 PDU			v C Sooale		م .	â
PDU	Dashboard						
		AY	Dashboard	Outlets Users Network	Logs System		
	Test Bay 2 PDU Location: R&D Bui	lding 112		Jul 17	2013 02:08:42 PM		
	Network Settings			? Revert	Save		
	All settings require a Settings marked wit	a Restart to take effect, except those h * are required.	e noted. See System page	for Restart.			
	TCP/IP		SNTP and Timez	one			
	IPv4 DHCP Enabled	Yes	Server 1 IP	192.168.1.99			
	IPv4 Address	192.168.1.10	Server 2 IP	192.168.1.99	í l		
	IPv4 Subnet Mask	255.255.255.0	Sync Frequency	12 hours			
	IPv4 Gateway	192.168.1.1	STD GMT Offset	-8.0 hours			
	IPv6 DHCP Enabled	Yes	DST GMT Offset	-7.0 hours			
	IPv6 Address	FE80::240:9DFF:FE6A:8554	DST Starts	3.2.0/02:00:00			
	IPv6 Prefix Length	64	DST Ends	11.1.0/02:00:00	Ĵ		
	HTTP and HTTPS		FTP				
	HTTP Port *	80	Enabled	🗷 Yes (restart not requ	ired)		
	HTTPS Enabled	Yes (disables HTTP)	Port *	21	-		
	HTTPS Port *	443	Auto Off	0 minutes			
	User Session Time	0 minutes					
	Telnet and SSH						
	Telnet Enabled	Ves.					
	Telnet Port	23					
	SSH Enabled	Ves					
	SSH Port	22					
	Documentation	nd Support					
	These links open new wi	ndows to external resources.					
		Factory Support Contacto	Other Links	and Downloads			
		Empil: cupport@manway.com	Droduct Info fo				



System Settings

The System page offers the place to set the system label and location. It also provides tools to restart the system. A Restart is needed after most Network setting changes.

U System Settings : Test Bay 2 PDU +	습 マ C ↓ Soogle	P 🏦
U Dashboard		
MARWAY POWER SOLUTIONS	Dashboard Outlets Users Network Logs System	
Test Bay 2 PDU Location: R&D Building 112	2 Logout Jul 17 2013 02:15:24 PM	
Unit Identification ? Revert Save	System Information	
System Label Test Bay 2 PDU System Location R&D Building 112	Model Number MPD 99000-003 Serial Number MPS 0099999-12 IPv4 Address 192.168.1.10 MAC Address 00:40:9D:43:35:97	
System Clock ?		
Year-Month-Day 2013-07-17 Hour:Min:Sec 02:15:24 PM	Firmware Versions / Updates	
The system clock is kept updated by SNTP.	Main Board 0.9.0 beta (r11) Outlet Board 1 1.99.0 (r299)	
Restart and Reboot ? System Restart	For details on updating the software, click the help link above. To get software updates, use the Product Info link in the Support panel below.	
The button above will restart the system software, but hardware is not rebooted.		
Documentation and Support		
These links open new windows to external resources.		
Factory Support Contacts Email : support@marway.com Phone : 800-462-7929	Other Links and Downloads Product Info for MPD 99000-003 Marway Website	



Command Line Interface Tour

After login, the CLI is organized into several get/set command sets such as these examples listed below:

- getOutlet / setOutlet
- getUser / setUser and addUser / deleteUser
- getTcp / setTcp
- getHttp / setHttp and getHttps / setHttps
- getSntp / setSntp
- getFtp / setFtp
- getLog / getStartupLog
- getSystem / setSystem

Login

The login task is fairly straight forward, for Telnet and SSH, enter a user account name at the login prompt, and a password.

nmand Line		

et Commands	Set Commands	Misc Commands
etOutlet	setOutlet	getOutlets
etUser etProfile	setUser setProfile	getUsers getMy, setMy addUser, deleteUser, makeLoginPswd
etNetwork etTcp etHttp etHttps etSntp etFtp	setTcp setHttp setHttps setSntp setFtp	
etLog etStartupLog		
etSystem	setSystem	?, help, quit
ore any command lp for that spe	d above, or type ecific command.	? after any command above
	tt Commands ttOutlet ttUser ttProfile ttTcp ttHttps ttHttps ttHttps ttStartupLog etSystem pre any comman up for that sp	t Commands Set Commands tOutlet setOutlet tUser setUser tProfile setProfile tTCp setTcp tHttp setHttp tHttp setHttp tEStartupLog tSystem setSystem ore any command above, or type tp for that specific command.



Built-in Help

At any point after login, typing help or ? at the prompt will display the toplevel help which is a list of commands. Each command can also be followed by a ? such as getSystem? (with or without a space) to display detailed usage help for that command. The command help will identify the syntax of the command, various attribute options, and some examples.

Command Syntax

The majority of commands follow a common set or get pattern. The following are typical get commands:

- getOutlet 8 switch
- getUser lester email
- getSystem location

These are typical set commands:

- setOutlet 1 switch on
- setUser lester email "les@example.com"
- setSystem location "Aisle 6 Rack 4"

You'll notice the pattern of get and set are identical except for the last parameter of the set command which passes the new value. Otherwise, both use the same syntax of:

commandName instanceID attribute [setValue]

Where commandName is something like getOutlet, getUser, etc. The instanceID is the numerical ID of an outlet, or the login name of a user. A few commands do not need an instance ID such as getSystem/setSystem where there's always one implied instance. The attribute is the specific detail being requested or set. Finally, if the command is a set, the fourth parameter is the new value.

0	0	☆ telnet — 82×42	M
#> (getSystem ?		
	Syntax: getSystem	[attribute]	
٧	Where:		
	[] : ir	ndicates an optional parameter	
	attribute : is	s one of the following (or its alias):	
	label		
	location		
	model_number	(model, model_no)	
	serial_number	(serial, serial_no)	
	mac_address	(mac, mac_addr)	
	version		
	start_time	(started)	
	time		
E	Examples:		
	getSystem label	: gets one attribute of the system	
	getSystem	: gets all attributes of the system	



Aliases and Variants

Commands are flexible in two ways. First, many command attributes have aliases. These aliases are revealed in the command help. For example, the getSystem command has an attribute model_number. That attribute name may be substituted with model_no or even model. So, getSystem model is the same command as getSystem model_number.

The second flexible feature is command variants. These are different ways of using the command to get different results—namely whether to get/set information for a single instance or multiple instances.

A get command is used like these examples:

getOutlet 4 switch

getUser lester email

will return one attribute for one specific instance.

A get command is used like these examples:

getOutlet 4

getUser lester

will return all attributes for one specific instance.

A get command is used like these examples: getOutlet getUser will return all attributes for all instances.

00	☆ telnet — 82×42	M.
<pre>#> getOutlet 1 switch</pre>		
Outlet ID	= 1: Utility Outlet	
Panel Name	= J1	
Switch State	= on	
#> getOutlet 1		
Outlet ID	= 1: Utility Outlet	
Panel Name	= J1	
Rated Volts	= 90 to 125 V	
Maximum Rating	= 20 A	
Continuous Rating	= 16 A	
Switch State	= on	
Startup State	= last known	
On Delay	= 0.2 s	
Off Delay	= 0.0 s	
Cycle Delay	= 5.0 s	
#>		



FTP and the File System

An on-board file system is used to store files containing all of the settings, and a small log of recent events. This file system is accessible through FTP.

Settings files can be downloaded for archiving, and even uploaded after being edited offline. This allows preconfiguration of many details which can then be uploaded to multiple PDUs.

A special file named factorySettings.txt can be downloaded, but cannot be edited. This file defines exactly which features the PDU has.





MPD Series RCM Software Software Operation



Software Operation

This chapter covers the typical operation features of the RCM Software which primarily revolves around the web interface Dashboard page, and a few CLI commands. If you have not yet read them, you may want to read these sections for an overview:

- For Setup instructions, refer to "Getting Started" on page 7.
- For an overview of the organization and features of the web browser interface refer to "Browser Interface Tour" on page 11.
- For an overview of the command line interface refer to ""Command Line Interface Tour" on page 17.

Login

Access to the PDU features and settings is available using HTTP, HTTPS, Telnet, or SSH. Additionally, access to the file system is available using FTP.

Each protocol has a unique login interface, but all use the same account and password. The account is sometimes referred to as the user name or login name. Regardless of the label used, it refers to the PDU account as entered in the User Settings form.

For discussion purposes, assume the PDU has been configured with an IP address of 192.168.1.10. We will further assume the predefined account root will be used for the examples.

CHANGE THE ROOT PASSWORD BEFORE DOING ANYTHING ELSE.

It is highly recommended, for security purposes, that the root user password be updated before or immediately when the system is connected to a network. When possible, use an Ethernet connection local to a single computer, or private LAN, to do the initial setup.

If the PDU is connected to a network without the password being updated, the PDU may be accessible by any person or script familiar with the default password of Marway's MPD PDUs which may result in unintended and unauthorized access to the PDU.

DNS Names vs IP Addresses

Your network administrator may have a domain name assigned to the PDU IP address, and you may have been instructed to use that name. It may be something like www.pduname.com or pdu.company.com to access a PDU across the Internet, or something like deptserverspdu.local to access the PDU on your local building network. These are DNS names, and can be used instead of the numeric IP address.



HTTP/HTTPS ("Web") Login

To access a PDU by the web, type the IP address in a web browser's address field such as http://192.168.1.10. If the PDU has been configured to use https, then use https:// instead of http:// at the start of the address.

The web login form includes four panels of information. Below the Marway logo, the top of the main white panel identifies the PDU's label and location. In the example screen capture, the label displays as Test Bay 2 PDU. This label and the location description below it are assigned by a PDU administrator.

Below the PDU label is the Login panel which includes the form to enter the user's account and password. The account is a unique name assigned to the user specifically for logging in. There is one default account named root which automatically has access to every feature and setting. The root account must be used to do the initial setup of all settings since there would be no other users defined yet.

Startup Notices

This panel will show any errors encountered during the startup of the PDU. The sample login page shown did not have any startup errors, and the panel title shows an "OK" status.

The types of errors that may show up here would include problems with missing or malformed settings files in the file system, hardware errors detected by the system, and network connectivity errors.

Documentation and Support

This panel is repeated on every page of the web interface for the PDU. It simply offers useful links to support services and online reference material.

Login : Test Bay 2 PDU - Windows Internet Explorer		
> C http://testbay2-pdu.local/login	👻 😽 🔀 🖓 Goog	ie ,
rorites PDU Login : Test Bay 2 PDU	👌 🕶 🖻 🔻 🖻	🛯 🖶 🔻 Page 🕶 Safety 🕶 Tools 🕶
MARWAY POWER SOLUTIONS		
Test Bay 2 PDU Location: R&D Building 112		
Login		
	Account Password Login	
Startup Notices : OK		?
Documentation and Support These links open new windows to external resou	ces.	
Factory St	port Contacts Other Links and Downloads tr@marway.com Product Info for MPD 99000-003	

🔥 Startup Notice	S			?
Date Time	Severity	Category	Message	
1970-01-21 08:27:00	Error	Time	SNTP may not be connected. There may be a delay, or the time server may not be reachable.	
Documentation and	Support			



Telnet Login

Telnet is a command line program already installed on most Linux, Unix, and Mac OS X computers. It is also installed on Windows XP, but not on newer versions of Windows. If you're using Windows Vista or newer, search the Internet for instructions on how to install and enable the Telnet client.

To log into the PDU using Telnet, open your operating system's command line interface (e.g. Windows' Command Prompt program, or OS X's Terminal application). Type the word telnet followed by an IP address or DNS name for the PDU like these examples:

telnet 192.168.1.10
telnet pdu.mycompany.com
telnet deptserverspdu.local

SSH Login

SSH is a command line program already installed on most Linux, Unix, and Mac OS X computers. It is not installed by default on Windows computers. For Windows computers, you will need to install an SSH program. There are several popular programs for this, consult your IT or network administrator if you have one. The rest of this discussion assumes the installation has been completed.

To log into the PDU via SSH, open your operating system's command line interface. Type the word ssh followed by a space, the PDU account name, an @ symbol, and an IP address or DNS name like these examples:

ssh root@192.168.1.10
ssh lester@pdu.mycompany.com
ssh jsmith@deptserverspdu.local

The Difference Between Telnet and SSH

The short version is that SSH is much more secure than Telnet. With Telnet, the information exchanged between your computer and the PDU is not encrypted. It is possible, under specific circumstances, for a skilled person to intercept that information. A worst case scenario would allow them to capture your login credentials and gain access to the PDU as though they were you.

With SSH, all traffic including the login is encrypted. This makes interception of the information pretty much useless (though nothing in the digital world is ever absolutely secure).

The Difference Between HTTP and HTTPS

If you substitute HTTP for Telnet above, and HTTPS for SSH, the story is pretty much identical. HTTPS is secure, where HTTP is not. However, the PDU determines whether you need to use HTTP or HTTPS based on the Network settings.

Which Should You Use?

Wherever possible, using SSH and HTTPS are the more secure choices. Yet, the others still exist. Many people choose to use Telnet and HTTP over the more secure options when they communicate over a closed, local network where the risk of someone capturing data to perform an attack is low (depending on the environment of course). If you're accessing the PDU across the Internet, use HTTPS and SSH. If you choose to use HTTP or Telnet, we assume you know why you're doing that.



Viewing Outlet Status

The status of all outlets are available to view on the web Dashboard, and through the CLI commands getOutlet.

Outlet Status on the Web Dashboard

The web interface Dashboard page is the focal point for reviewing all "remote" features on the PDU. There is no equivalent view on the command line, so the web interface is the most convenient way to see the most data at one time.

The screen capture at the right shows the default Dashboard layout for a PDU with 8 remote-controlled outlets. The left-most column displays an "indicating switch" for each outlet.

On and Off are normal operational values for the indicators. If the button displays ???, this indicates the system cannot detect the control circuitry for the outlet switching. Another possible display is a red blinking dot on the button which indicates that the actual current state of the outlet does not match what the software expects it to be.

For outlet switching capabilities, refer to "Outlet Control" on page 26 for details. The label for each Outlet (set to "Outlet 1" etc. by default) can be edited. See "Outlet Settings" on page 49 for details on changing the label as well as the amps setpoint settings.

Outlet Status by Command Line

Access to outlet status on the command line is available with the getOutlet command. The getOutlet command displays several attributes. Refer to

Test Bay Location: F	2 PD R&D BI	U uilding 112			Jul 17 2013 11:57:40 AM
Outlets :	All				?
All Outle	Off	Below			Apply Actions Apply
Switch	ID	Outlet Label	Panel Name	Rating	Action
• On	1	Utility Outlet	J1	90 to 125 VAC, 20 A max.	None ‡
• On	2	Control Panel & Logic	32	90 to 125 VAC, 20 A max.	None \$
• On	3	Hydraulics Controller	33	90 to 125 VAC, 15 A max.	None 🗘
• On	4	M1 Motor Controller	34	90 to 125 VAC, 15 A max.	None \$
• On	5	M2 Motor Controller	35	90 to 125 VAC, 15 A max.	None \$
• On	6	Battery Backup	36	90 to 125 VAC, 15 A max.	None ‡
• On	7	DAQ Server	37	48 VDC, 30 A max.	None ‡
• On	8	DB Server	38	48 VDC, 30 A max.	None \$
Documen These links o	tation pen new	and Support windows to external resources.			
		Factory Support C Email : support@ma	Contacts arway.com	Other Links and Downloads Product Info for MPD 99000-003	



"getOutlet" on page 54 for additional command details, but the basic command formats to view power data are:

getOutlet [id] returns all attributes
getOutlet [id] switch returns only switch status

The label for each Outlet may be edited. See "setOutlet" on page 54 for details on changing the label and the amps setpoint settings.

Outlet Control

Outlets may be controlled using the web Dashboard, and through the CLI command setOutlet. Additionally, adjustable delays in response to on and off commands can be set using the web Outlets page or the CLI setOutlet command. See "Outlet Settings" on page 49.

Outlet Switching and Adjustable Delays

Outlets may be switched on, off, and cycled. Cycle switches an outlet off, then back on with a single command. If an outlet was already off, it will be switched on. In addition to the obvious on and off action, switching also includes the ability to add delays to these actions.

An On Delay will delay the on action. If the On Delay of Outlet 1 were set to 1.5 seconds, then when an operator sends an on command to that outlet (by web or CLI), the system would count out an approximate 1.5 seconds before dispatching that command to the outlet controller.

An Off Delay will delay the off action. It works just like an on delay in adding a delay between the time a command is given and when it is performed.



<pre>#> getOutlet 1</pre>	· · · · · · · · · · · · · · · · · · ·
Outlet ID	= 1: Utility Outlet
Panel Name	= J1
Rated Volts	= 90 to 125 V
Maximum Rating	= 20 A
Continuous Rating	= 16 A
Switch State	= on
Startup State	= last known
On Delay	= 0.2 s
Off Delay	= 0.0 s
Cycle Delay	= 5.0 s



A Cycle Delay is a unique value used only between the off and on steps of a cycle command. During a Cycle command, the off step will be delayed by the Off Delay setting. The on step will be delayed by the larger of the On Delay or Cycle Delay. This can be confusing, but the intent is to prioritize the needs of attached equipment which is sensitive to startup order. This may result in longer then necessary delays for some scenarios, but it ensures the minimum delays are honored for sequence-sensitive cases.

Staggering Startups to Curb Inrush

The most common use of On Delay is to stagger the power up of attached devices to avoid an excessive inrush which might trip a circuit breaker if all devices started at the same time. Most electrical systems suitable for the PDU have very short in rush periods, and the default On Delay value should work for most applications.

Staggering Startups for Dependencies

Another use of On Delay is to intentionally connect several devices in sequential outlets, then stagger the on times to meet the needs of a startup dependency between those particular pieces of equipment.

For example, suppose we have 3 devices. Device A depends on Device B, and Device B depends on Device C, and each device should be powered for 20 seconds before the next unit is started. Set the devices as follows:

- Device C in Outlet 1 with an On Delay of 0.
- Device B in Outlet 2 with an On Delay of 20 seconds.
- Device A in Outlet 3 with an On Delay of 20 seconds.

The PDU powers each outlet in sequence. Therefore, in the above scenario, after the PDU is powered or rebooted, the sequence of events will be:

- Outlet 1 is switched on immediately
- A delay of 20 seconds elapses for the On Delay of Outlet 2.



- Outlet 2 is switched on.
- A delay of 20 seconds elapses for the On Delay of Outlet 3.
- Outlet 3 is switched on.

Using Cycle Delay

The Cycle Delay is most typically used to ensure that when a piece of equipment is switched off, enough time passes to allow capacitors to fully discharge, hard drives to stop spinning, and other factors to come to a "full stop" before power is reapplied. Those details and the time needed, if any, are specific to the equipment involved.

Switching Sets of Outlets

When multiple outlets are targeted to be switched (for example, the web UI's Action controls), each outlet is handled in sequence. That is, outlet 1 will be addressed first, then outlet 2, 3, etc. With each outlet, any applicable delay is handled in sequence. The delays build up cumulatively from a starting point. Assume outlets 1 through 4 each have a 5 second On Delay. If these outlets start out as off, and are all commanded at one time to turn on, then after 5 seconds outlet 1 is turned on, after 10 seconds outlet 2 is turned on, after 15 seconds outlet 3 is turned on, etc. as each outlet's own 5-second delay is accounted for.

Outlet Control from the Web

Outlet Switching

The web Dashboard offers three ways to manipulate outlet state. The first is an All Outlets set of buttons to affect all outlets with a single command to turn them on, turn them off, or cycle them. When using these controls, each outlet is handled in sequence as described in "Switching Sets of Outlets" above.

Outlets : A	All					?
All Outlets	Show	n Below Cycle			Apply Actions Apply	
Switch	ID	Outlet Label	Panel Name	Rating	Action	
• On	1	Utility Outlet	J1	90 to 125 VAC, 20 A max.	None 🗘	
• On	2	Control Panel & Logic	32	90 to 125 VAC, 20 A max.	None 🗘	
• On	3	Hydraulics Controller	33	90 to 125 VAC, 15 A max.	None +	
• On	4	M1 Motor Controller]4	90 to 125 VAC, 15 A max.	None ‡	
• On	5	M2 Motor Controller	35	90 to 125 VAC, 15 A max.	None ‡	
• On	6	Battery Backup	36	90 to 125 VAC, 15 A max.	None ‡	
• On	7	DAQ Server	37	48 VDC, 30 A max.	None 🗘	
• On	8	DB Server	J8	48 VDC, 30 A max.	None ‡	



The second method for switching outlets is the Action column of the Dashboard Outlets panel. Each outlet has a popup menu to choose On, Off, or Cycle. Each outlet can be set to any of these options. When all choices have been made, pressing the Apply button at the top of the column will issue each command in sequence.

The third method for switching outlets is the Switch column buttons which provide simple and immediate manual On and Off switching control.

Outlet Delays

Adjusting outlet delays is done in the Outlets section of the web interface.

Outlet Control from the Command Line

Outlet Switching

For all outlet actions, the command line offers the setOutlet command. Using the available variants and attributes, the following commands are possible:

- setOutlet 1 switch on affects only the identified outlet number
- setOutlet 1 switch off
- setOutlet 1 switch cycle
- setOutlet switch on affects all outlets
- setOutlet switch off
- setOutlet switch cycle

Whenever a specific outlet is identified, the command is issued immediately. That is, setOutlet 1 switch off works just like a manual switch, and the on/ off delays are ignored. However, when switching all outlets, the on/off/cycle delays are honored.

```
#> setOutlet 1 switch off
```

Setting switch of outlet 1 to: off... OK.

#> setOutlet switch on

Setting switch of outlet 1 to: on... OK. Setting switch of outlet 2 to: on... OK. Setting switch of outlet 3 to: on... OK. Setting switch of outlet 4 to: on... OK. Setting switch of outlet 5 to: on... OK. Setting switch of outlet 6 to: on... OK. Setting switch of outlet 7 to: on... OK. Setting switch of outlet 8 to: on... OK.

#> setOutlet 3 on_delay 2

Setting on_delay of outlet 3 to: 2... OK.

#> setOutlet off_delay 0

Setting off_delay of outlet 1 to: 0... OK. Setting off_delay of outlet 2 to: 0... OK. Setting off_delay of outlet 3 to: 0... OK. Setting off_delay of outlet 4 to: 0... OK. Setting off_delay of outlet 5 to: 0... OK. Setting off_delay of outlet 6 to: 0... OK. Setting off_delay of outlet 7 to: 0... OK. Setting off_delay of outlet 8 to: 0... OK.



Outlet Delays

Adjusting outlet delays is also done using the setOutlet command.

- setOutlet 3 on_delay 2 sets only the identified outlet value
- setOutlet 3 off_delay 0
- setOutlet 3 cycle_delay 15
- setOutlet switch on_delay 2 sets all outlets to the same value
- setOutlet switch off_delay 0
- setOutlet switch cycle_delay 15

Refer to "setOutlet" on page 54 for details.

Viewing Log Data

Log entries are viewable on the web interface using the main menu item Logs where two tabs are available: System Log and Startup Notices. Additionally, the CLI commands getLog and getStartupLog may be used.

Logged Events

The RCM Software maintains an event log intended for tracking recent event history. Logged events include:

- System startup
- Successful user login
- Failed user login
- Outlet switched on/off or cycled (via web or CLI, but not by RESTful API)
- Settings updated
- Internal errors

Log Format

The log file is an ASCII text file with tab delimited fields and "Windows-style" line endings (\r\n). Each log entry includes the following data:

- Severity
- DateTime
- PDU IP
- PDU Location
- Category

- Message
- PDU Model
- PDU Version
- PDU Extra

The Severity field describes an escalating scale of importance using these keywords:

Info : normal event activity information

Warn : something which is not a problem, but may be more important to know about than Info (e.g. a failed user login)

Error : errors are not expected. An error message is an indication of something not working correctly. Occasional errors may not be a sign of a larger problem, but multiple errors should be investigated.

Critical : a serious error has occurred. The system should be restarted as soon as possible.

The DateTime field will be in the format of YYYY-MM-DD HH:MM:SS using 24-hour time. This format is naturally sortable.

The PDU IP field is the IPv4 address of the PDU. This will be the same for all entries, but allows logs to be combined, and keeps logs distinct if the IP address changes.



The PDU Location field is the user-entered location field.

The Category field is a set of keywords useful for searching or filtering the log for specific types of events. Not all categories apply to all RCM Software versions or PDUs. The keywords, in no particular order, include:

Startup : used when recording that a user issued a startup command, and also the startup event itself.

Login : used when recording successful and failed login attempts.

Alarm : used when recording that a setpoint or other alarmable condition has been triggered.

Email : used when logging alert preparations, and other activities of the SMTP system.

Switch : used when recording changes to an outlet state (on, off, or cycled).

Settings : used when recording changes to any of the setting files.

Time : used when logging activities of the internal clock system and SNTP protocol.

Files : used when logging activities of the file system, which will almost always be for an error.

System : used when logging activities of the core operating system, which will almost always be for an error.

The Message field is the text of the event message.

The PDU Model field is the model number of the PDU.

The PDU Version field is the version number of the PDU software.

The PDU Extra field provides additional internal details for errors which will be useful to factory technician when troubleshooting.

System Log vs. Startup Log

The system log has a fixed number of allowed entries. Once the available log space becomes full, the oldest message is deleted to make room for the newest message.

How long a period of time the log covers depends on how many events have been recorded. A session which includes a number of settings changes, switch state changes, etc. will write more events to the log, so the total period of time covered could be shortened to hours or even minutes. Periods where the PDU is operating normally with little or no settings changes, and no alarm events could cover weeks or months.

With such a system, it's possible that any error messages logged during the PDU startup process could be pushed out of the log at some point. To prevent this, an error message which occurs during the startup process is also logged to a separate "startup log."

The startup log is retained until the system is restarted. The startup log is visible on the PDU web login page, the Logs section of the web interface, and by using the getStartupLog command. The events in this log remain viewable at any time until the next startup, during which the startup log is replaced by that startup's events (if any).

It is normal for the startup log to be empty.



Web Viewing of Log Data

The Logs section of the web interface include two tabs: System Log and Startup Notices. The System Log list is not sortable, but the log viewer allows filtering based on the Severity and Category fields making it easier to seek out specific events. Selecting an option from one or both of the filters will remove from the display (not from the log itself) everything except those items matching the selected options.

Command Line Viewing of Log Data

Use the command getLog to view the system log, and getStartupLog to view the startup notices. The getLog command accepts filter parameters to help seek out a specific type of log message. See "getLog" on page 63.

Fest Bay 2 .ocation: R&E	PDU) Buildin	g 112		? Logout Jul 17 2013 12:02:46 PM		
System Log	Startup	Notices				
Local System	n Log			? Show All Severities + Show All Categories + Update		
Date Time		Severity	Category	Message		
2013-07-17 11	:59:55	Info	Startup	Startup completed		
2013-07-17 12	:01:09	Info	Login	Web access granted to System Admin (root)		
2013-07-17 12	:01:18	Info	Settings	System Identify Settings updated by System Admin (root)		
2013-07-17 12	:01:25	Info	Settings	ps Power Outlet Settings updated by System Admin (root)		
2013-07-17 12	:01:38	Info	Login CLI access granted to System Admin (root)			
2013-07-17 12	:02:06	Info	Startup	RESTART initiated by System Admin (root)		
2013-07-17 12	:02:20	Info	Startup	RESTART initiated by System Admin (root)		
2013-07-17 12	:02:37	Info	Startup	Startup completed		
2013-07-17 12	:02:44	Info	Login	Web access granted to System Admin (root)		
Documentati	ion and new windov	Support vs to externa Far	al resources. ctory Support	Contacts Other Links and Downloads		
		Ph	ione : 800-46	22-7929 Marway Website		



MPD Series RCM Software

User Management



User Management

An RCM Software user record includes several features:

- authentication (a login name and password),
- access methods (which network protocols the user is allowed to login to),
- a personal profile (name, company, email, and phone information),
- permissions (specific capabilities the user is allowed to use), and
- status (last login time and login count).

The RCM Software will store up to 20 locally-defined user records. One of those records is predefined as the root user (which cannot be deleted).

Root User

A user with a login name of "root" is predefined. This user cannot be deleted, and the login name cannot be changed. The permissions cannot be edited (all permissions are permanently allowed). However, the password and profile for the user may be edited. The password can be updated through the web interface or command line.

Authentication

Generally speaking, authentication deals with the identification of a user through some means of information that only the user should have. In the RCM Software, this means having both an account login name and a password.

Name and Password

Authentication requires user name and password credentials. The name is an arbitrary text value. The password requirements help to enforce moderately strong passwords, and with up to 32 characters, allows for very strong passwords.

Passwords in the Settings File

The settings file stores passwords salted an encrypted. It is not possible to simply edit a user's password in the settings file. To generate a salt and encryption string for the settings file, use the command line command makeLoginPswd. See "makeLoginPswd" on page 55 for details.

Authorization

Authorization deals with allowing the user to see, utilize, or edit specific data in or features of the software. In the RCM Software, authorization includes whether the user in enabled to use one or more protocols to log into the PDU, and a specific set of selected permissions once logged in.

Access Methods

The root user is inherently allowed to use all login protocols. All other users are allowed to web access only (they cannot use command line interfaces nor FTP).



Permissions

A number of specific permissions have been predefined in the software to allow an administrator to have granular control over what each user is allowed to see and do. Users do not have "roles" or belong to "groups." These paradigms are useful for batch editing of users, but our experience in user management suggests there are always exceptions to these paradigms which results in the overprescribing of permissions to many users. Given the relatively few users needing access to a PDU, the assignment of explicit permissions uniquely to each user allows for more control.

Auditing

Auditing deals with logging user activity so that an audit trail is available to identify exactly what a user has done in the system.

Activity Logging

Though not built with full-feature auditing capabilities, the RCM Software provides some user activity logging. The on-board log keeps track of several critical activities, and is able to stamp those activities with user identification. Software restarts, outlet switching, and settings file changes are all stamped with user login names which performed those actions. The log file size is limited, and is intended more for short-term troubleshooting than long-term audit trails.

A + testbay	2-pdu.local/users	PDU Use	er Update : Test Bay 2 PDU	C Reader
DWARWAY Power Solutions Test Bay 2 PDU Location: R&D Building 112			Dashboard Outlets	Users Network Logs System
			? Logout Jul 17 2013 12:04:31 PM	
Show List	Add User			
Users : lester (Le	ster Tester)			? Cancel Save
Settings marked wi	th * are requir	ed.		
Authentication			Permissions	
Login Name	lester		Outlet Settings	Network Settings
Password * (Rules)			View Outlet Settings	View Network Settings
Password Again *			Control Outlet On/Off	Edit TCP/IP
Access Mathada			Edit Outlet Labels	
Access Methous			Eult Outlet Delays	Edit FTP
Web Login	Enabled (http, https)	Users Settings	
Personal Profile			View Users	System Settings
First & Last Name	Lester	Tester	Edit Users	View System Settings
Email Address	les@test.com	1	Logs Settings	E care by seein bettings
Company Name	Acme Tect A	utomation	View Logs	
Job Role	Test Technic	ian		
Company Phone	800-999-43	21		
Direct Phone				
Status				
Last Login	2013-07-12	13:32:10		



Profile

The Personal Profile fields of the user record allow a user to be identified by name, the company represented, and with email and phone contact information.

Adding, Deleting, Editing Users

Users can be added, deleted, and edited using either the web interface or command line. Using the web interface, press the Add User button or select a user to edit or delete from the users list. The web interface for adding and editing users is shown on the previous page. Deleting a user via the web interface is a two-step process. First press the trash button for the user to be deleted. A non-editable form will display the details of that user, and provide a set of confirmation Delete and Cancel buttons.

Separate commands on the command line are used to add, delete, edit, and display user records. Refer to the addUser, deleteUser, setUser, and getUser commands in "User Commands" on page 55.

Show List	Add User					
Users : List						?
Edit Login Name	User Name	Email	Access	Logins	Last Login	Del
root	System Admin	sysmin@test.com	Web CLI FTP	104	2013-07-17 12:02:44	
lester	Lester Tester	les@test.com	Web	137	2013-07-12 13:32:10	-


MPD Series RCM Software Software Settings



Software Settings

Adjustable values for outlets, users, network protocols, etc. are called settings, and are accessible through the web UI, the command line, and also in files accessed through FTP.

Editing the files directly would not be an ideal way to make adjustments normally, but can be a convenient way to set up or update multiple PDU units quickly by preparing replacement files and uploading them to each PDU (followed by a restart to activate the new settings).

In this chapter, the purpose and acceptable values of each setting will be identified. Additionally, how to use the web UI, CLI, and files to edit them will also be covered.

Adjust Settings with the Web UI

Settings are adjusted in all sections of the web interface except the Dashboard and Logs. Changing settings is done through web forms. Values can be edited, and the form must be saved. If invalid values are entered, the web interface will highlight those entries and explain valid entry options. Additionally, most fields will display a range and/or default value tooltip by hovering a mouse pointer over the field and pausing for a short time.

Adjust Settings with the CLI

Settings are adjusted on the command line using various set commands specific to the setting being set (e.g. setOutlet or setUser). If an invalid value is submitted, the CLI will usually offer some text as to the valid entry



Outle	t Settings : All			? Revert Save				
One or more fields below has an input error.								
ID	Outlet Label	Panel Name	Switch Delays in Seconds On Off Cycle	Startup State				
The field	d On Delay must be between 0 and 60 se	conds.						
1	Utility Outlet	J1	99999 0.0 5.0	Last Known \$				
2	Control Panel & Logic	J2	0.2 0.0 5.0	Last Known \$				
3	Hydraulics Controller	13	02 00 50	Last Known 💠				

Create New User			? Cancel Save
	One or more fields	s below has an input error.	
Settings marked with	* are required.		
Authentication		Permissions	
The field Account Name mu	ist be 3-32 characters of a-z, A-Z, 0-9,	Outlet Settings	Network Settings
underscores, hyphens, per	iods, and @ symbol.	View Outlet Settings	View Network Settings
Login Name *	John Doe	Control Outlet On/Off	Edit TCP/IP
Password * (Rules)	•••••	Edit Outlet Labels	Edit SNTP
Password Again *		Edit Outlet Delays	Edit HTTP



requirements, and at least provide feedback that the value was not valid. Note that the CLI can be used to set multiple objects to the same value. For example:

setOutlet cycle_delay 5.0

will set the cycle delay for every outlet whereas

setOutlet 3 cycle_delay 5.0

will set the value for outlet 3 only. See "Command Syntax" on page 52 for other command line options.

Adjust Settings in the Files

Settings files may be copied from FTP, edited, then uploaded back to FTP. Many FTP client applications will open a text file directly and save it back to the same place. Settings files are simple ASCII text files saved with "Windows-style" line endings (that is, the \r\n sequence).

Use a text editor, not a word processor, to edit these files. On Windows, the simple Notepad utility will work (though a more sophisticated text editor will provide better editing tools). Do not use WordPad, or any of the "office" type applications like Microsoft Word, OpenOffice Writer, etc.

While this chapter discusses the individual settings within these files, refer to "Settings Files Reference" on page 76 for details about the formatting of each settings file.

The file at /factory/factorySettings.txt can be copied for archiving and reference, but it cannot be edited, deleted, or replaced. This file is what defines the configuration of the PDU product, and tells the software what the PDU features and specifications are.

#> setHttp port 999999

Error: port value must be between 1 and 65535.

RESTART THE PDU AFTER EDITING SETTINGS FILES

It is possible to edit settings files in place (using an FTP client application that opens text files), or by copying, editing, and uploading the settings file. It is necessary to restart the PDU after the new files have been saved/uploaded. First, this is the only way to have the new setting stake effect. Second, if another user makes any changes to settings via web or CLI, or even changes the state of an outlet, the uploaded settings files will be overwritten with their previous values saved in the PDU's memory. Always use a software restart command by web or CLI after uploading settings files.



System Settings

System settings are for items universal to the whole PDU, and which otherwise doesn't have another appropriate home. Refer to the screen capture in "System Settings" on page 16.

Setting	Description	CLI Command / Attribute	File / Setting	Туре	Validation
Label	A label to uniquely identify the PDU. It is displayed on all web pages just below the logo, and in response to the getSystem CLI command.	setSystem label	systemSettings.txt label =	Text	All English letters, numerals, and punctuation except double quotes.
Location	A label to uniquely identify the location of the PDU. It is displayed on all web pages just below the system label at the top of the page, and in response to the getSystem CLI command.	setSystem location	systemSettings.txt location =	Text	All English letters, numerals, and punctuation except double quotes.
Time	Date and time values for when the PDU is not connected to a networked time server (SNTP).	setSystem time	(none)	Time	YYYY-MM-YY HH:MM:SS (24 hr)



Network Settings

Network settings define the network protocols such as IP addresses, ports, and other values. This guide will not attempt to explain the protocols themselves. Assistance from an experienced network administrator may be necessary to help with unfamiliar protocols or settings. Refer to the screen capture in "Network Settings" on page 15.

Setting	Description	CLI Command / Attribute	File / Setting	Туре	Validation
IPv4 DHCP Enabled	Determines whether the TCP/IPv4 address will be supplied by a DHCP server ("on") or entered manually ("off").	setTcp ipv4_dhcp	networkSettings.txt ipv4dhcp =	Text	"on" or "off"
IPv4 Address	The TCP/IPv4 address. Leave empty for DHCP.	setTcp ipv4_address	networkSettings.txt ipv4address =	Text	0.0.0.0 format
IPv4 Subnet Mask	The TCP/IPv4 subnet mask. Usually something like 255.255.255.nnn (where nnn depends on the IP address range of the subnet). On large networks, the 3rd element may not be 255.	setTcp ipv4_subnet	networkSettings.txt ipv4subnet =	Text	0.0.0.0 format
IPv4 Gateway	The TCP/IPv4 subnet gateway address.	setTcp ipv4_gateway	networkSettings.txt ipv4gateway =	Text	0.0.0.0 format
IPv6 DHCP Enabled	Determines whether the TCP/IPv6 address will be supplied by a DHCP server ("on") or entered manually ("off").	setTcp ipv6_dhcp	networkSettings.txt ipv6dhcp =	Text	"on" or "off"
IPv6 Address	The TCP/IPv6 address. Leave empty for DHCP.	setTcp ipv6_address	networkSettings.txt ipv6address =	Text	IPv6 format



Setting	Description	CLI Command / Attribute	File / Setting	Туре	Validation
IPv6 Prefix Length	The TCP/IPv6 prefix length. Almost always /64.	setTcp ipv6_prefix_length	networkSettings.txt ipv6PrefixLength =	Integer	0-128 (default = 64)
HTTP Port	The TCP port for the service.	setHttp port	networkSettings.txt httpPort =	Integer	1-65535 (default = 80)
HTTPS Enabled	Indicates whether HTTPS should be used instead of HTTP (only one runs at a time).	setHttps enabled	networkSettings.txt httpsEnabled =	Text	"on" or "off"
HTTPS Port	The TCP port for the service.	setHttps port	networkSettings.txt httpsPort =	Integer	1-65535 (default = 443)
HTTP Session Time	The maximum time in minutes a web session can be inactive before the login session is flagged as invalid.	setHttp session_minutes	networkSettings.txt httpSessionMins =	Integer	15–1440 (default 30) 0 = disabled (no expiration)
SNTP Server 1 IP	The TCP/IPv4 address of a network time server.	setSntp server1	networkSettings.txt sntpServerIp1 =	Text	0.0.0.0 format
SNTP Server 2 IP	The TCP/IPv4 address of a network time server.	setSntp server2	networkSettings.txt sntpServerIp2 =	Text	0.0.0.0 format
SNTP Sync Frequency	Defines how often in hours the PDU should contact the SNTP server for an updated time.	setSntp sync_interval	networkSettings.txt sntpSyncInterval =	Integer	1–168 hours (default = 12) 0 = disabled
Timezone STD GMT Offset	Defines the local standard time offset in hours from Greenwich Mean Time (GMT).	setSntp std_offset	networkSettings.txt sntpStdOffset =	Text	GMT-H.H or GMT+H.H Where H.H is the offset in hours such as GMT-8.0 or GMT+9.5
Timezone DST GMT Offset	Defines the local daylight savings time offset in hours from Greenwich Mean Time (GMT).	setSntp dst_offset	networkSettings.txt sntpDstOffset =	Text	GMT-H.H or GMT+H.H Where H.H is the offset in hours such as GMT-8.0 or GMT+9.5



Setting	Description	CLI Command / Attribute	File / Setting	Туре	Validation
Timezone DST Starts	Defines when daylight savings time starts.	setSntp dst_starts	networkSettings.txt sntpDstStarts =	Text	Enter format like 11.1.0/02:00:00 meaning Month.Week.Day/Hr:Min:Sec where : Month = 1 to 12 Week of the month = 1 to 5 Day = 0 (Sun) to 6 (Sat) Hour = 0 to 24 Minute = 0 to 59 Second = 0 to 59 The example above means Sun first week of Nov at 2AM.
Timezone DST Ends	Defines when daylight savings time ends.	setSntp dst_ends	networkSettings.txt sntpDstEnds =	Text	Enter format like 11.1.0/02:00:00 (see row above for details)
FTP Enabled	Determines whether FTP is enabled or not.	setFtp enabled	networkSettings.txt ftpEnabled =	Text	"true" or "false"
FTP Port	The TCP port for the service.	setFtp port	networkSettings.txt ftpPort =	Integer	1-65535 (default = 21)
FTP Auto Off	When non-zero, this sets how many minutes after FTP is enabled that it will remain active before being automatically disabled. When zero, FTP remains active indefinitely.	setFtp auto_off	networkSettings.txt ftpAutoOff =	Integer	5-60 minutes (default 60) 0 = no expiration



User Settings

User settings define the authentication, profile, and permissions of each PDU user. Most of these settings are adjustable through the web or command line interfaces. Some status data such as the number of logins, most recent login date, and others are stored in the settings file. Generally, these would not be manually adjusted, and there is not a way to use the web or CLI to adjust these values. However, their presence in the settings file means technically they could be adjusted. Refer also to "User Management" on page 34.

Setting	Description	CLI Command / Attribute	File / Setting	Туре	Validation
Login Name	The account name a user will provide to login into the PDU.	getUser auth addUser	userSettings.txt loginName =	Text	3-32 characters of a-z, A-Z, spaces, or underscores
Login Password	The password for the account name a user provides to login into the PDU. The userSettings.txt file requires an encrypted version of the password and a salt. Use the command line utility makeLoginPswd to generate the encrypted loginPswd value and the pswdSalt value to store in the file.	addUser setUser password	userSettings.txt loginPswd =	Text	8-32 characters, at least one from A-Z, at least one from a-z, at least one from 0-9. A minimum of four letters (i.e. it can't be mostly numerals). Should include one from !@#\$%^&* but no other punctuation marks.
Password Salt	This value is generated internally when passwords are added or updated via the web UI or command line and saved to the settings file. However, to manually add to or modify the userSettings.txt file use the command line utility makeLoginPswd to generate the encrypted loginPswd value and the pswdSalt values.	(no get command) (no set command) Use makeLoginPswd to create values for the settings file.	userSettings.txt pswdSalt =	Text	12 random characters from a-z, A-z, 0-9



Setting	Description	CLI Command / Attribute	File / Setting	Туре	Validation
Password Is Temporary	This value is generated internally to indicate whether the password needs to be replaced by the user on the next login. [This is not yet in use.]	getUser auth (no set command)	userSettings.txt pswdIsTemp =	Text	"true" or "false"
Allowed Client IPs	A definition of which IPv4 addresses the PDU will allow client logins from. [This is not yet in use.]	(not yet available)	userSettings.txt allowedClientIPs =	Text	A list of IPv4 addresses and/ or slash notation addresses.
Login Count	This value is maintained internally to indicate how many times the user has logged into the PDU.	getUser auth (no set command)	userSettings.txt loginCount =	Integer	0-65535
Most Recent Login	This value is maintained internally to indicate the last date and time the user logged into the PDU.	getUser auth (no set command)	userSettings.txt mostRecentLogin =	ISO Datetime	YYYY-MM-YY HH:MM:SS (24 hr)
Failed Attempts	This value is maintained internally to indicate the how many times the users account has attempted a login with a wrong password. After 3 failed attempts, the account will be blocked for three minutes.	getUser auth (no set command)	userSettings.txt failedAttempts =	Integer	0-65535
Lockout Time	This value is maintained internally to indicate when a user's account was set to be blocked for excessive failed login attempts. The time is ignored and reset after it the inerval exceeds three minutes.	getUser auth (no set command)	userSettings.txt lockoutTime =	ISO Datetime	YYYY-MM-YY HH:MM:SS (24 hr)
First Name	The user's first name (given name).	getUser profile setUser first_name	userSettings.txt firstName =	Text	0–24 characters of a-z, A-Z, spaces, periods, hyphens, and apostrophes.
Last Name	The user's last name (surname).	getUser profile setUser last_name	userSettings.txt lastName =	Text	0–24 characters of a-z, A-Z, spaces, periods, hyphens, and apostrophes.
Company Name	The company name the user represents.	getUser profile setUser company_name	userSettings.txt companyName =	Text	All English letters, numerals, and punctuation except double quotes.



Setting	Description	CLI Command / Attribute	File / Setting	Туре	Validation
Job Role	A descriptive title or role the user performs for the PDU or for the company he represents.	getUser profile setUser job_role	userSettings.txt jobRole =	Text	All English letters, numerals, and punctuation except double quotes.
Company Phone	A general phone number for the company the user represents.	getUser profile setUser company_phone	userSettings.txt companyPhone =	Text	Up to 29 characters of 0-9, spaces, periods, and $+()-/x$ (write extensions like x000).
Direct Phone	A direct phone number to reach the user.	getUser profile setUser direct_phone	userSettings.txt directPhone =	Text	Up to 29 characters of 0-9, spaces, periods, and $+()-/x$ (write extensions like x000).
Email	An email address to reach the user. This same address is used to send the user alerts (subject to the Alerts Settings).	getUser profile setUser email	userSettings.txt email =	Text	A valid email address.
SMS	A mobile phone number in an email addressable form to send the user alert messages by SMS. This is generall in the form of mobile_number@carrier.tld such as 2125551212@txt.att.net	getUser profile setUser sms	userSettings.txt sms =	Text	A valid email address.
Login Via Web	Determines whether the user is allowed to access the PDU via the web interface.	getUser auth permissions setUser login_via_web	userSettings.txt loginViaWeb =	Text	"true" or "false"
Login Via CLI	Determines whether the user is allowed to access the PDU via Telnet and SSH. Note: as of this version, only the root user can access Telnet and SSH. Changing this setting for other users is ignored.	getUser auth permissions setUser login_via_cli	userSettings.txt loginViaCLI =	Text	"true" or "false"
Login Via FTP	Determines whether the user is allowed to access the PDU via FTP. Note: as of this version, only the root user can access FTP. Changing this setting for other users is ignored.	getUser auth permissions setUser login_via_ftp	userSettings.txt loginViaFTP =	Text	"true" or "false"



Setting	Description	CLI Command / Attribute	File / Setting	Туре	Validation
View Power Settings	Determines whether the user is allowed to view power settings. If any of the power setting edit permissions is true, then this will be forced true as well.	getUser auth permissions setUser view_power_settings	userSettings.txt viewPowerSettings =	Text	"true" or "false"
Control Outlets	Determines whether the user is allowed to switch outlets on/off.	getUser auth permissions setUser control_outlets	userSettings.txt controlOutlets =	Text	"true" or "false"
Edit Outlet Labels	Determines whether the user is allowed to edit outlet labels.	getUser auth permissions setUser edit_outlet_labels	userSettings.txt editOutletLabels =	Text	"true" or "false"
Edit Outlet Delays	Determines whether the user is allowed to edit outlet delays.	getUser auth permissions setUser edit_outlet_delays	userSettings.txt editOutletDelays =	Text	"true" or "false"
View Logs	Determines whether the user is allowed to view log entries.	getUser auth permissions setUser view_logs	userSettings.txt viewLogs =	Text	"true" or "false"
View Network Settings	Determines whether the user is allowed to view network protocol settings.	getUser auth permissions setUser view_network_settings	userSettings.txt viewNetworkSettings =	Text	"true" or "false"
Edit TCP/IP	Determines whether the user is allowed to edit TCP/ IP protocol settings.	getUser auth permissions setUser edit_tcpip	userSettings.txt editTCPIP =	Text	"true" or "false"
Edit SNTP	Determines whether the user is allowed to edit SNTP protocol settings.	getUser auth permissions setUser edit_sntp	userSettings.txt editSNTP =	Text	"true" or "false"
Edit HTTP	Determines whether the user is allowed to edit HTTP(S) protocol settings.	getUser auth permissions setUser edit_http	userSettings.txt editHTTP =	Text	"true" or "false"
Edit FTP	Determines whether the user is allowed to edit FTP protocol settings.	getUser auth permissions setUser edit_ftp	userSettings.txt editFTP =	Text	"true" or "false"



Setting	Description	CLI Command / Attribute	File / Setting	Туре	Validation
View Users	Determines whether the user is allowed to view user settings.	getUser auth permissions setUser view_users	userSettings.txt viewUsers =	Text	"true" or "false"
Edit Users	Determines whether the user is allowed to edit user settings.	getUser auth permissions setUser edit_users	userSettings.txt editUsers =	Text	"true" or "false"
View System Settings	Determines whether the user is allowed to view system settings.	getUser auth permissions setUser view_system_settings	userSettings.txt viewSystemSettings =	Text	"true" or "false"
Edit System Settings	Determines whether the user is allowed to edit system settings.	getUser auth permissions setUser edit_system_settings	userSettings.txt viewSystemSettings =	Text	"true" or "false"



Outlet Settings

Outlet settings include labels and some items specific to switching behavior. Refer also to the screen captures in "Outlet Settings" on page 13.

Labels

The RCM Software defaults to labeling each power device by its device name along with a sequential numerical identification. This results in default labels such as Outlet 1, Outlet 2, etc. To allow labels with more meaningful descriptions, the labels for Outlets can be changed.

Valid characters for labels include all English letters, numerals, and most standard keyboard punctuation marks. Quotes are not allowed in a label, nor are extended range, escaped, or Unicode characters.

Outlet Switch Delays

Outlet switches feature three action delay settings. Details about these features can be found in "Outlet Delays" on page 29. In summary:

The On Delay causes a delay between the moment when an On action is initiated and when the outlet actually switches on.

The Off Delay causes a delay between the moment when an Off action is initiated and when the outlet actually switches off.

The Cycle Delay causes a delay in the cycle process after the outlet is switched off before it is switched on.

The values for these delays are in seconds with a valid range of 0 to 60. A decimal value in tenths such as 0.2 is acceptable.

Outlet Startup State

If the PDU loses facility power, or is rebooted with the reset switch in the onboard keypad, the outlets will be switched off. Software restart commands do not affect outlet state. If the outlets are switched off, the PDU needs to know what to do with the outlets after startup. This is the job of the Startup State setting. It has three values: on, off, and last known. When "on" the outlet will be switched on after startup, and "off" will leave it off. The "last known" option will restore the same state the outlet was in before power was removed.



Setting	Description	CLI Command / Attribute	File / Setting	Туре	Validation
Outlet Label	A label to identify the outlet (defaults to "Outlet 1").	getOutlet setOutlet label	powerSettings.txt (outlet~ID section) label =	Text	All English letters, numerals, and punctuation except double quotes.
Outlet On Delay	Defines the length of a delay between the moment when an On action is initiated and when the outlet actually switches on.	getOutlet setOutlet on_delay	powerSettings.txt (outlet~ID section) onDelay =	Decimal	0-60.0 seconds
Outlet Off Delay	Defines the length of a delay between the moment when an Off action is initiated and when the outlet actually switches off.	getOutlet setOutlet off_delay	powerSettings.txt (outlet~ID section) offDelay =	Decimal	0-60.0 seconds
Outlet Cycle Delay	Defines the length of a delay in the cycle process after the outlet is switched off and before it is switched on.	getOutlet setOutlet cycle_delay	powerSettings.txt (outlet~ID section) cycleDelay =	Decimal	0-60.0 seconds
Outlet Startup State	After the PDU is rebooted (or power is restored after a facility power loss), this setting determines whether the outlet should be switched on, off, or set to the last known state.	getOutlet settings setOutlet startup_state	powerSettings.txt (outlet~ID section) startupState =	Text	One of the keywords: "on," "off," or "last known"
Outlet Switch State	Indicates the current state of the outlet (or, in the settings file the last known state).	getOutlet settings setOutlet switch	powerSettings.txt (outlet~ID section) switchState =	Text	One of the keywords: "on" or "off"



MPD Series RCM Software Command Line Reference



Command Line Reference

This chapter includes an overview of the command line syntax, and the entire contents of the built-in help for every command.

Built-in Help

At any point after login, typing help or ? at the prompt will display the top-level help which is a list of commands.

Each command can also be followed by a ? such as getSystem? (with or without a space) to display detailed usage help for that command. The command help will identify the syntax of the command, various attribute options, and some examples.

Command Syntax

The majority of commands follow a common set or get pattern. The following are typical get commands:

- getOutlet 8 switch
- getUser lester email
- getSystem location

Marway PDU	Command Line		
Login: roo [.] Password:	t *****		
#> ?			
Workspace	Get Commands	Set Commands	Misc Commands
POWER	getOutlet	setOutlet	getOutlets
USERS	getUser getProfile	setUser setProfile	getUsers getMy, setMy addUser, deleteUser, makeLoginPs
NETWORK	getNetwork getTcp getHttp getHttps getSntp getFtp	setTcp setHttp setHttps setSntp setFtp	
LOG	getLog getStartupLog		
SYSTEM	getSystem	setSystem	?, help, quit



These are typical set commands:

- setOutlet 1 switch on
- setUser lester email "les@example.com"
- setSystem location "Aisle 6 Rack 4"

You'll notice the pattern of get and set are identical except for the last parameter of the set command which passes the new value. Otherwise, both use the same syntax of:

commandName instanceID attribute [setValue]

Where *commandName* is something like getOutlet, getUser, setSystem, etc. The *instanceID* is going to be the numerical ID of a power device such as an outlet, or the login name of a user. A few commands do not need an instance ID such as getSystem/setSystem where there's always one implied instance. The *attribute* is the specific detail being requested or set. Finally, if the command is a set, the fourth parameter is the new value.

Command Aliases

Many command attributes have aliases (synonyms). These aliases are revealed in the command help. For example, the getSystem command has an attribute model_number. That attribute name may be substituted with model_no or even model. So, getSystem model is the same command as getSystem model_number. The idea behind aliases is to enable commands to be flexible so people can use words and phrases they might naturally think of (without going overboard and having aliases for every word).

Command Variants

Most commands have variants, which are different ways of using the command to get different results. This primarily applies to get/set commands to address whether the command applies to a single instance or multiple instances of the command target.

A get command used like these examples: getOutlet 4 switch getUser lester email will return one attribute for one specific instance.

A get command used like these examples: getOutlet 4 getUser lester will return all attributes for one specific instance.

A get command is used like these examples: getOutlet getUser will return all attributes for all instances.

Command variants are explained with each command's built-in help by identifying optional parameters in square brackets like [name] in the example help display below:

#> getUser?

Syntax: getUser [name] [attribute]



Outlet Commands

getOutlet

Syntax: getOu	tlet [id] [attribute]	
Where: [] id	<pre>: indicates an optional parameter : is the number of the outlet (e.g. 1 or 3, etc.) or leave it empty to refer to all outlets</pre>	
attribute	: is one of the following (or its alias):	
label switch startup_st rated_amps rated_volt on_delay off_delay cycle_dela power settings	<pre> (relay) ate (startup) (ra, ratedamps, max_amps, maxamps) s (rv, ratedvolts) y y (pwr)</pre>	
Examples: getOutlet 6 getOutlet 6 getOutlet sw getOutlet	<pre>switch : gets one attribute for one outlet</pre>	

setOutlet

 Syntax: setOutlet [id] at	tribute value
Where: [] : indicate id : is the n or leave attribute : is one o	s an optional parameter umber of the outlet (e.g. 1 or 3, etc.) it empty to refer to all outlets f the following (or its alias):
label switch startup_state on_delay off_delay cycle_delay	 (relay) (startup)
value for label value for switch value for startup_state value for delays	: is a quoted string (60 chars max) : is one of: on, off, reboot cycle bounce : is one of: on, off, last_known last : is a decimal of seconds, or: default
Examples: setOutlet 6 switch reboo setOutlet cycle_delay 2. setOutlet 3 on_delay def	t : sets one outlet 5 : sets all outlets to 2.5 secs ault : resets to the default value



User Commands

getProfile (a.k.a getMy)

Syntax: getProfile attribute or getMy attribute

Where attribute is one of the following:

status- record management status detailsauth- user authentication detailsprofile- user profile detailspermissions- user permissions details

Examples:

getProfile status getProfile auth getProfile profile getProfile permissions getMy permissions

makeLoginPswd

Encrypts a provided string (presumed to be a password) using RIPEMD-160. Returns a salt and encryption string suitable for adding to the userSettings.txt file stored on the file system.

#> makeLoginPswd "abc123"

salt = 8BnM0Qd6hCTJ
pswd = 6f23b84a083fbbbf08ae1d7d4c7c5e4d9590543d

setProfile (a.k.a setMy)

Syntax: setProfile attribute "value" setMy attribute "value"

Where attribute is one of the following:

password	 following these rules: minimum of 8 characters, maximum of 32 characters, must include at least 4 letters (case sensitive), must include at least 1 digit (0-9), should include at least one of these !@#\$^&* symbols for higher security, but not required
first name	- the user's first name
last_name	- the user's last name
company_name	- the user's companyname
job_role	- the user's job role or title
company_phone	- 888-555-1818 x12345
<pre>mobile_phone</pre>	- 212-555-1010
email	- the user's email address
Examples: setPro setPro setPro setPro setPro setMy	ofile password "Qwer!234" ofile webadmin enabled true ofile company_name "Silicon Farmers" ofile job_role "Server Support Contractor" job_role "Server Support Contractor"



getUser

Syntax: getUser [name] [attribute] Where: [] : indicates an optional parameter : is the login name of the user (e.g. john doe) name or leave it empty to refer to all users attribute : is one of the following attribute groups: status = record management status details auth = user authentication details profile = user profile details permissions = user permissions details : or one of the following profile attributes: enabled, first_name, last_name, company_name, job_role, company_phone, direct_phone, email : or one of the following permission attributes login via web view_power[_settings] view_network[_settings] control_outlets edit tcpip edit_outlet_labels edit_sntp edit_outlet_delays edit http view_logs edit_ftp view_system[_settings] view users edit_system[_settings] edit_users Examples: getUser john_doe edit_http : gets one attribute for one user getUser john doe : gets all attributes for one user getUser edit_http : gets one attribute for all users : gets all attributes for all users getUser

setUser

Svntax: setUser [name] attribute "value" Where: : indicates an optional parameter [] : is the login name of the user (e.g. john doe) name or leave it empty to refer to all users attribute : is one of the following profile attributes: enabled - enter true/false or yes/no password - look at setMy? for details first name - the user's first name last name the user's last name company name - the user's companyname - the user's job role or title iob role company phone - 888-555-1818 x12345 direct phone - 212-555-1010 email - the user's email address : or one of the following permission attributes with a value of true/false or yes/no: login via web view power[settings] view network[settings] control outlets edit tcpip edit outlet labels edit sntp edit outlet delays edit http view logs edit ftp view users view system[settings] edit users edit system[settings] Examples: setUser john doe password "Qwer!234" setUser webadmin enabled true setUser john doe company name "Silicon Farmers" setUser dbAdmin job role "Server Support Contractor" setUser webadmin edit system true



addUser

Syntax:	addUser "loginName" password "loginPswd"		
Where:	<pre>loginName : the account name, following these rules: - minimum of 3 characters, - maximum of 32 characters, - letters (case sensitive), numerals, underscores, hyphens, periods, and @ symbols. leading/trailing spaces will be trimmed, the name cannot already be used.</pre>		
	<pre>loginPswd : the account password, following these rules: - minimum of 8 characters, - maximum of 32 characters, - must include at least 4 letters (case sensitive), - must include at least 1 digit (0-9), - should have one or more symbols from !@#\$^&* for higher security, but not required</pre>		
Examples:	addUser "JohnDoe" password "Qwer!234" addUser "mail_admin" password "wu7jsKhgr3poi" addUser "admin@example.com" password "wu7jsKhgr3poi"		

deleteUser

Syntax:	deleteUser	"name" password "pswd"
Where:	name pswd	: the account name for the user : the account password for login
Examples:	deleteUser	"John Doe" password "abc#4mPwr"

addUser "admin@example.com" password "wu7jsKhgr3poi"



Network Commands

getTcp

[]

Syntax: getTcp [attribute] Where: : indicates an optional parameter attribute : is one of the following (or its alias): ipv4 dhcp (dhcpv4, dhcp) ipv4_address (ipv4, ip) ipv4 subnet (subnet, mask) ipv4_gateway (gateway) ipv6 dhcp (dhcpv6) ipv6 address (ipv6) (prefixlen, prefix) ipv6_prefix Examples: getTcp : displays all attributes getTcp subnet : displays a single attribute's value _____

setTcp

Syntax: setTcp attribute value Where: [] : indicates an optional parameter attribute : is one of the following (or its alias): ipv4 dhcp (dhcpv4, dhcp) ipv4 address (ipv4, ip) ipv4 subnet (subnet, mask) ipv4 gateway (gateway) ipv6 dhcp (dhcpv6) ipv6 address (ipv6) ipv6 prefix (prefixlen, prefix) value for ipv4 dhcp : is on off value for ipv4 address : is is formatted like 0.0.0.0 value for ipv4 subnet : is is formatted like 0.0.0.0 value for ipv4 gateway : is is formatted like 0.0.0.0 value for ipv6 dhcp : is on off value for ipv6 address : defaults to :: for ::0.0.0.0 compatibility value for ipv6 prefix : almost always an integer of 64 Examples: setTcp ip 192.168.0.15 : sets the ipv4 address setTcp subnet 255.255.255.0 : sets the ipv4 subnet mask setTcp dhcpv6 on : enables DHCP for ipv6



getHttp

setHttp

setHttp session 45 : session invalid after 45 mins of inactivity

Syntax: getHttp [attribute]	Syntax: setHttp attribute value
Where: [] : indicates an optional parameter attribute : is one of the following (or its alias):	Where: [] : indicates an optional parameter attribute : is one of the following (or its alias):
port session_minutes (session, session_mins)	port session_minutes (session, session_mins)
Examples: getHttp : displays all attributes getHttp port : displays a single attribute's value	<pre>value for port : is an integer (default is 80) value for session_minutes : is an integer of minutes. Applies to both</pre>
	Examples: setHttp port 8080 : binds HTTP to port 8080



getHttps

Syntax: getHttps [attribute]

Where:

[]	:	indicates	an optional parameter
attribute	:	is one of	the following (or its alias):

enabled		
port		
session_minutes	(session,	session_mins)

Examples:

getHttps	: displays all attributes	
getHttps enable	: displays a single attribute's value	

setHttps

- - -

_____ Syntax: setHttps attribute value Where: : indicates an optional parameter [] attribute : is one of the following (or its alias): enabled - port - session_minutes (session, session_mins) value for enabled : is true false or on off value for port : is an integer (default is 443) value for session_minutes : is an integer of minutes. Applies to both http and https user sessions. (default is 30, min is 15, max is 1440) Examples: setHttps port 8443 : binds HTTP to port 8443

setHttps enabled on	: enables https, which disables http	
setHttps enabled off	: disables https, which enables http	
setHttps session 45	: session invalid after 45 mins of inactivity	



getSntp

Syntax: getSntp [attribute]	Syntax: setSntp attribute value		
Where:	Where:		
[] : indicates an optional parameter	[] : indicates an optional parameter		
attribute : is one of the following (or its alias):	attribute : is one of the following (or its alias):		
server1	server1		
server2	server2		
sync_interval (sync)	sync_interval (sync)		
<pre>std_gmt_offset (std_offset, std_gmt)</pre>	<pre>std_gmt_offset (std_offset, std_gmt)</pre>		
dst gmt offset (dst offset, dst gmt)	dst gmt offset (dst offset, dst gmt)		
dst start	dst start		
dst_end	dst_end		
Examples:	value for servers : is an IP address like 0.0.0.0		
getSntp : displays all attributes	value for sync : is between 0 and 168 hours (default is 12)		
getSntp server1 : displays a single attribute's value	value for gmt offset : is from -12 to +14		
	value for dst_start/end : is in format of 3.2.0/02:00:00		

Examples:

setSntp

<pre>setSntp sync_interval 12</pre>	: updates local clock time every 12 hours
<pre>setSntp dst_zone GMT-7</pre>	: sets DST to 7 hrs before GMT

getFtp

Syntax: getFtp [attribute]	Syntax: setFtp attribute value
Where: [] : indicates an optional parameter attribute : is one of the following (or its alias):	Where: [] : indicates an optional parameter attribute : is one of the following (or its alias):
enabled port auto_off	enabled port auto_off
Examples: getFtp : displays all attributes getFtp enabled : displays a single attribute's value	<pre>value for enabled : is true false or on off value for port : is an integer (default is 21) value for auto_off : is integer of minutes, 0 = never</pre>
	Examples: setFtp enabled true : enables FTP service setFtp port 2100 : binds FTP to port 2100

setFtp

setFtp auto_off 20 : FTP disables 20 mins after being enabled



Log Commands

getLog

Syntax: getLog [severity option] [category option] Where: [] : indicates an optional parameter : is one of the following (or its alias): attribute severity (sev) category (cat) severity option is one of the following: critical, error, warn, info, debug category option is one of the following: alarm, email, switch, config, startup, login, files, time, log, system, kernel Examples: : shows all log lines getLog getLog severity error : shows error entries for all categories getLog category switch : shows only switch category log lines getLog sev error cat files : shows only errors related to the file system

getStartupLog

Syntax: getStartupLog Examples: getStartupLog : shows all log lines



System Commands

getSystem

Syntax: getSystem [attribute]	Syntax: setSystem attribute [value]
Where:	Where:
[] : indicates an optional parameter	[] : indicates an optional parameter
attribute : is one of the following (or its alias):	attribute : is one of the following (or its alias):
label	label
location	location
model_number (model, model_no)	restart
serial number (serial, serial no)	
mac address (mac, mac addr)	value for label : is a quoted string (30 chars max)
version	value for location : is a guoted string (30 chars max)
start time (started)	value for time : is formatted like "YYYY-MM-DD HH:MM:SS" (24 hr)
time	value for restart : none
Examples:	Examples:
getSystem label : gets one attribute of the system	setSystem label "WWW Servers" : sets label of the system
getSystem : gets all attributes of the system	setSystem restart : restarts the system software

setSystem



MPD Series RCM Software RESTful API Reference



RESTful API Reference

This chapter includes an overview of what a RESTful API is, and the details of using the RCM Software implementation.

The RESTful API for RCM Software is designed to support optimized machine-to-machine control of a PDU by a remote script, primarily in the application of automated test and evaluation ("ATE"). Therefore, the focus of the API is to provide access to power management features such as switching outlets and collecting power data where available. The API does not support access to non-power-related resources such as Users, Logs, configuration settings, etc. It is possible to automate access to these resources using Telnet commands.

What is REST and RESTful API?

The Gist of REST

REST is an acronym for REpresentational State Transfer which at an academic level boils down to being an architectural style of how to represent information on a distributed hypermedia system. In other words, it's a particular style of how to design links and connections to information resources on a computer network (such as the internet).

It was originally described in 2000 (see the reference material sidebar), but has gained rapid popularity in recent years. During that time, there's been significant chatter amongst software developers about the academic and pragmatic nuances. For the purposes the RCM Software, the objectives and implications are far simpler than what has to be considered for more complex applications.

Reference Materials for REST

The canonical academic paper which defined REST: http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm

Some additional internet discussions about REST: http://en.wikipedia.org/wiki/Representational_state_transfer http://www.ibm.com/developerworks/webservices/library/ws-restful/



One academic point to cover is the difference between the terms REST and RESTful. REST is an architecture not a protocol (such as HTTP, FTP, etc.). Therefore, a system does not access information via REST. Rather, the data is accessed in a REST-like or RESTful manner. The latter term has the broader usage, and thus we end up with the term RESTful API—a programmer's interface to something using a RESTful design style.

Programming for a RESTful API

Since the REST architecture was originally conceived in the context of hypermedia (i.e. the world wide web), a fundamental tenet of REST is the use of HTTP as the communication protocol.

There are plenty of good resources online describing REST and RESTful API design. However, the bottom line for the RCM Software is that "messages" are sent to the PDU using a combination of HTTP URIs, verbs, and form data. For example to switch an outlet on, the verb and URI combination of:

PATCH http://192.168.1.10/outlet/3/switch

would be sent along with name-pair form data of an existing session and the new state value for the outlet switch like this:

session_id=Ygv0PkCheJrl4UuyPfd10FUpD807XQ4Y
new_value=on

The response to this request will be a standard HTTP response with the body having a simple string. Since all response to RCM requests are single values, the RCM Software uses plain text responses rather than incur unneeded overhead of XML, JSON, or HTML parsing (common formats for RESTful responses).

Programming for a RESTful API involves using an HTTP library for your programming language of choice in order to deliver an HTTP request to the

PDU, and acquire HTTP responses which are usually already parsed into their header and body components. Again, there are plenty of online resources to help overcome the learning curve.

Why Use the RESTful API?

The RESTful API is provided as an alternative to the common practice of scripting to a Telnet interface or even SNMP. First, Telnet is intended to be a human interface. From the machine perspective, there's a lot of wordy noise to parse out to acquire just the facts needed. The RCM RESTful API eliminates all that noise. SNMP is a machine interface, but can be rather complex to develop interfaces with. Since a RESTful API is basic HTTP, it can be much easier to develop quick, custom automation scripts with.

HTTP Verbs

The RCM Software RESTful API makes use of GET, POST, PATCH, and DELETE behaviors. There is no need for PUT, and HEAD is not supported. However, the embedded web server supports GET, POST, and HEAD request methods. Since HEAD is not used, there's no conflict there. However, to provide support for PATCH and DELETE, the API uses a common technique of creating a pseudo-verb through a form parameter.

Whenever a URI needs GET or POST, the standard HTTP request method can be specified. However, when PATCH or DELETE are needed, the URI must be submitted with POST, and the request must include a form parameter named rest_method along with a string value of "PATCH" or "DELETE" as needed. For example, while the following would be ideal:

PATCH https://192.168.0.10/outlet/1/switch? new_value=on&session_id=abcxyz



it is necessary to accomplish that message this way:

POST https://192.168.0.10/outlet/1/switch? new_value=on&session_id=abcxyz&rest_method=PATCH

This form-parameter technique is quite common in RESTful APIs since many web servers (and browsers) do not support all HTTP verbs. The world wide web got started with GET, POST, and HEAD, and thus we are left to create a work around for those verbs not widely supported yet.

Responses

Following the principles of focus and simplicity for the API, it is assumed the automated test environment is one in which the power architecture (how many power sources, circuits, outlets, etc.) of the PDU is a known entity which will remain stable over the life of the script. Therefore, the API does not support discovery of the PDU power architecture. That is, it does not support requesting collections which could be parsed to navigate or "walk" available resources. The API is focused on delivering and changing state of specific attributes of resources already known to the remote script.

The result of this simplicity is that all responses to RESTful API requests are available as an html/text media type only, and have only a single value. In effect, each response is a simple string of a single value.

While these limitations might been seen as restrictive from the traditional RESTful API perspective, it helps to focus the limited resources of an embedded controller on the most value-added capabilities for the majority of applications.

If a request fails, or was malformed to start with, a return value of some type is always provided. Errors are identified as negative numbers.

RCM REST Error Codes

- -1 General error. A low-level internal error happened where no specifics were being tracked as to the cause.
- -12 the new_value value is out of range for the attribute
- -200 the resource ID in the URL is not valid (too big or too small)
- -217 the string for the switch state new_value is not valid
- -298 the resource attribute in the URL is not valid
- -317 session_id passed by the request is not recognized. Start a new session to get a new id.



Authentication API

MPDe resources are not public. A conversation over the RESTful API must be initialized with an authentication which results in the creation of a session. Each subsequent request must be authenticated with a session ID.

A session ID is obtained with a POST to /session with parameters account=&password=. A 32-character string is returned. That string has to be submitted as part of all other requests in a parameter named session_id=.

Reminder: for maximum security on any open network, the PDU should have HTTPS enabled to encrypt the authentication dialog which occurs between the automation script and the PDU web server.

API Resources

The following sections identify the available RESTful resources and the API requirements for each. Remember that all requests require authentication. See the section above for details.

/session

A session must be created before any other PDU resource can be requested. Then, the session id must be passed to all subsequent requests, and when the session's use is no longer needed, it should be destroyed.

The RESTful API session space is shared with the regular web user space. If a script makes use of a specific user login, then a human (or another script) subsequently uses the same login credentials, the originally running script will be blocked because its session will have been destroyed (overwritten by the second session). Each script and each human should have unique user login credentials.

POST /session

URI Parameters:

• none

Form Parameters

- account (required) the user account name
- password (required) the password for the account (be sure this is URL encoded)

Response

• (session_id) a 32-character string which must be saved to send along with future requests

DELETE /session

URI Parameters:

• none

Form Parameters

- session_id (required) the previsouly acquired ID
- rest_method (required) must be "DELETE" in upper case

Response

• a string of 32 zeros

/outlet

There will always be one or more outlets. The resource GET/outlets is not supported, the script must know which outlet of the PDU to request. Each attribute of the outlet is requested directly as identified in the sections below.

GET /outlet/:id/label

URI Parameters:

• :id — the PDU's numeric instance of the Outlet (values start at 1, a label cannot be used as an :id)

Form Parameters:

• session_id — the string returned from a POST /session

Response

• a string of arbitrary words labeling the outlet instance

GET /outlet/:id/switch

URI Parameters:

• :id — the PDU's numeric instance of the Outlet (values start at 1, a label cannot be used as an :id).

Form Parameters:

• session_id — the string returned from a POST /session

Response

- a value of "on" or "off" would be normal
- a value of "unknown" means the outlet status cannot be determined (this may require a power cycle, and/or an All On / All Off cycle of the outlets to clear).
- a value of "stuck on" or "stuck off" means the actual outlet status is out of sync with what the RCM controller expects it to be. This may indicate a hardware failure, or may require an All On / All Off cycle of the outlets to clear.

GET /outlet/:id/rated_amps

URI Parameters:

• :id — the PDU's numeric instance of the Outlet (values start at 1, a label cannot be used as an :id)

Form Parameters:

• session_id — the string returned from a POST /session

Response

• a numeric string (i.e. "20") indicating the maximum amps rating of the outlet



GET /outlet/:id/rated_volts

URI Parameters:

• :id — the PDU's numeric instance of the Outlet (values start at 1, a label cannot be used as an :id)

Form Parameters:

• session_id — the string returned from a POST /session

Response

• a numeric string (i.e. "125") indicating the maximum volts rating of the outlet

GET /outlet/:id/startup_state

URI Parameters:

• :id — the PDU's numeric instance of the Outlet (values start at 1, a label cannot be used as an :id)

Form Parameters:

• session_id — the string returned from a POST /session

Response

• "on" or "off" or "last known" corresponding to the Startup State options

GET /outlet/:id/on_delay

URI Parameters:

• :id — the PDU's numeric instance of the Outlet (values start at 1, a label cannot be used as an :id)

Form Parameters:

• session_id — the string returned from a POST /session

Response

• a numeric string (i.e. "1.50" or "2") representing seconds



GET /outlet/:id/off_delay

URI Parameters:

• :id — the PDU's numeric instance of the Outlet (values start at 1, a label cannot be used as an :id)

Form Parameters:

• session_id — the string returned from a POST /session

Response

• a numeric string (i.e. "1.50" or "2") representing seconds

GET /outlet/:id/cycle_delay

URI Parameters:

• :id — the PDU's numeric instance of the Outlet (values start at 1, a label cannot be used as an :id)

Form Parameters:

• session_id — the string returned from a POST /session

Response

• a numeric string (i.e. "1.50" or "2") representing seconds

PATCH /outlet/:id/switch

Actually sent as a POST, but identified as a PATCH through a form parameter.

URI Parameters:

• :id — the PDU's numeric instance of the Outlet (values start at 1, a label cannot be used as an :id)

Form Parameters:

- session_id the string returned from a POST /session
- rest_method must be "PATCH" in upper case
- new_value a value of "on" turns the outlet on (apply power to it), a value of "off" turns the outlet off (remove power from it), and a value of "cycle" turns the outlet off, pauses for cycle_delay time, then turns the outlet back on. Regardless of whether the first state is on or off, the cycle command is always off-on (it never cycles on-off).

Response

- if new_value is "on" the result should also be "on"
- if new_value is "off" the result should also be "off"
- if new_value is "cycle" the result should be "on"


PATCH /outlet/:id/startup_state

Actually sent as a POST, but identified as a PATCH through a form parameter.

URI Parameters:

• :id — the PDU's numeric instance of the Outlet (values start at 1, a label cannot be used as an :id)

Form Parameters:

- session_id the string returned from a POST /session
- rest_method must be "PATCH" in upper case
- new_value a value of "on" or "off" or "last known" corresponding to the Startup State options

Response

• the response should be the same string as what was submitted

PATCH /outlet/:id/on_delay

Actually sent as a POST, but identified as a PATCH through a form parameter. URI

Parameters:

• :id — the PDU's numeric instance of the Outlet (values start at 1, a label cannot be used as an :id)

Form Parameters:

- session_id the string returned from a POST /session
- rest_method must be "PATCH" in upper case
- new_value a numeric string such as 0.5, 2.0, or 1

Response

• the response should be the same string as what was submitted

PATCH /outlet/:id/off_delay

Actually sent as a POST, but identified as a PATCH through a form parameter. URI

Parameters:

• :id — the PDU's numeric instance of the Outlet (values start at 1, a label cannot be used as an :id)

Form Parameters:

- session_id the string returned from a POST /session
- rest_method must be "PATCH" in upper case
- new_value a numeric string such as 0.5, 2.0, or 1

Response

• the response should be the same string as what was submitted



PATCH /outlet/:id/cycle_delay

Actually sent as a POST, but identified as a PATCH through a form parameter. URI

Parameters:

• :id — the PDU's numeric instance of the Outlet (values start at 1, a label cannot be used as an :id)

Form Parameters:

- session_id the string returned from a POST /session
- rest_method must be "PATCH" in upper case
- new_value a numeric string such as 0.5, 2.0, or 1

Response

• the response should be the same string as what was submitted

/system

System identification can be useful for automated logging. The script will already know which IP address is being connected to, but the System label and location can provide addition arbitrary user description of the PDU.

GET /system/1/label

URI Parameters:

• note that in place of :id, this resource is always called with a fixed 1 for the ID

Form Parameters:

• session_id — the string returned from a POST /session

Response

• a string of arbitrary words labeling the PDU system

GET /system/1/location

URI Parameters:

• note that in place of :id, this resource is always called with a fixed 1 for the ID

Form Parameters:

• session_id — the string returned from a POST /session

Response

• a string of arbitrary words intended to represent the PDU's location



MPD Series RCM Software Settings Files Reference



Settings Files Reference

This chapter includes an overview of the settings files format, syntax, and default content, and how the file might be used to save time and effort managing multiple PDUs.

Know What You're Doing

Read the alert at the right!

The safest way to change settings on an MPD RCM PDU is to use the built-in web interface or command line. However, it is understood that making changes to multiple PDUs this way can be time consuming.

Making settings available in user-editable files is an advanced capability. It is intended for people with experience in managing server-like configuration files, and who are practiced at learning and recognizing the format, syntax, and requirements of such files.

Not every suitable scenario for or consequence of editing files directly are explained in this guide.

Accessing Settings Files

Settings files are accessed through FTP. Using the IP address of the PDU as the FTP server identification, login with the root user account. This will expose the file system of the PDU. It is highly recommended that files be copied to a computer or tablet for editing. Only after all changes are completed, should the files be uploaded, and then the PDU restarted.

EDITING SETTINGS FILES DIRECTLY CAN RESULT IN A MISCONFIGURED PDU WHICH DOES NOT OPERATE AS EXPECTED.

It is highly recommended that if you plan to edit files directly, that you have a nonmission-critical PDU available to experiment, practice, and learn with. Only after you're comfortable (and perhaps your boss is comfortable) that you know what you're doing should you work on the files of PDUs in service.

Marway Power Solutions has attempted to make the MPD RCM software selfhealing from the affects of malformed or missing settings files. However, we cannot guarantee that every conceivable malformed variation within a file or combination of files will not result in a PDU which must be returned to the factory to restore it to complete operation.

Be careful. Test your changes on a non-mission-critical PDU before deploying.

AFTER UPLOADING SETTINGS FILES THE PDU MUST BE RESTARTED FOR THE CHANGES TO TAKE EFFECT.

If the PDU is not restarted, any other user making changes via the web or CLI could overwrite the changes just uploaded.



File Format and Syntax

Settings files are ASCII text with "Windows-style" line endings (\r\n). Most files are not whitespace sensitive, but powerSettings.txt is. The format of each file is detailed in the sections below.

Name/Value Pairs

All files include name/value pairs where a setting name is followed by an = and the setting value. The whitespace surrounding the = is not significant. Most files have whitespace before = in order to neatly align the setting values, but this spacing is not meaningful. If the file is manually edited, and the whitespace altered, it will revert back to the aligned spacing whenever the RCM software updates the file. Examples of these pairs (not form the same file):

loginName = root
ipv4Address = 192.168.1.10

Objects and IDs

Some files include named objects with IDs in the format of object~N where object is a label such as outlet, circuit, user (and others), and where N is a simple incremental integer (1, 2, 3, etc.), The whitespace around the ~ is not significant, but will be removed whenever the RCM Software updates the file. Examples of objects and IDs:

outlet~3 user~root A block like this will be used to show a whole file (if it can fit on the page).

A block like this will be used to show a fragment of a file (that is, the whole file is not being shown, only a portion of it).



Object Groups of Settings

Files using objects will have settings for that object grouped together under the object. The grouped settings will be indented. In most files (except factorySettings.txt and powerSettings.txt), the indentation will be two spaces. These spaces aren't meaningful, but will be restored automatically when the file is updated by the RCM Software. An example of an object and its settings:

outlet~8	
label	= Frequency Generator A
switchState	= on
startupState	= last known
onDelay	= 0.10
offDelay	= 0.00
cycleDelay	= 2.00

File Identity and Version

The first two lines of a settings file are critical to identifying the contents of the file. The first line identifies the file type, and the second line identifies the format version of the file (not the version of the RCM Software).

Being a part of the text file, theses values are technically editable (but they should never be edited), therefore it is critical to ensure they remain correct.



systemSettings.txt

Refer to "System Settings" on page 40 for details about the settings.

This is a simple file consisting of only the file identity a couple of settings. The entire file is shown at the right. The formatting of the file consists of basic name/ value pairs. There are no objects.

networkSettings.txt

Refer to "Network Settings" on page 41 for details about the settings.

This file consists of only the file identity with basic name/value pairs. There are no objects. The order of the protocols and the spacing between them are not significant, but the RCM Software will restore its default formatting whenever the file is updated by the web or CLI.

userSettings.txt

Refer to "User Settings" on page 44 for details about the settings.

This file consists of a file identity and multiple user objects. Each user has a sequential object ID starting with 1, and is followed by several settings (not all are shown in the sample at the right).

There is one required user whose loginName must be root. The root user must be the first user in the file and have an ID of 1.

```
identity = RCM_SYS
version = 1.0.0
// NOTE: user added comments are not preserved in this file.
// See the User Guide for details about format and options.
label = Test Bay 2 PDU
location = R&D Building 112
```





After the root user, the order of the users is not significant, though their IDs must increment by one (1, 2, 3, etc.).

The order of the settings within the user object and the spacing between user objects is not significant, but the RCM Software will restore its default formatting whenever the file is updated by the web or CLI.

If you need but do not know the root user password, the correct way to reset and update the root user password is explained in "Change root User Password" on page 8.

powerSettings.txt

Refer to "Outlet Settings" on page 49 for details about the settings.

This file consists of a file identity and multiple nested objects. Each level of nested objects has its own series of sequential object IDs starting with 1. For example, a unit may have three Circuits numbered 1, 2, and 3. There may also be Outlets with IDs of 1, 2, and 3. The incremental numbering starts over with each change of object type. Each object is followed by several settings (not all are shown in the sample at the right).

The MPD series RCM Software is adapted from a more complex line of products which support many features per power object (Sources, Inputs, Lines, etc.). These objects will be listed in the powerSettings file, but will rarely have any attributes for an MPD series product where the remote control emphasis is on outlet switching. To generate a factory default file, delete the powerSettings.txt file, and restart the RCM Software. A new file will be created.

This file IS whitespace sensitive—it uses meaningful tab indentation. Make sure your text editor does not replace tabs with spaces.

```
(a fragment of userSettings.txt)
identity = RCM USER
version = 1.0.0
// NOTE: user added comments are not preserved in this file.
// See the User Guide for details about format and options.
user~1
  loginName
                           = root
  loginPswd
                           = c33e6f892497cc56e23d8159834a0db299e27c8b
  pswdSalt
                           = 15bGJp1QpJmm
  pswdHint
                           = p8
  // (big piece cut out of this example)
user~2
  loginName
                           = lester
                           = c122bc8666ec22658f35cb0f0b438d85d56d8ade
  loginPswd
```



In the example listing, the > characters are not a part of the text file, but rather an illustrative way to show how many tab characters have been inserted before the start of the text on that line.

As discussed previously, the spacing around the = symbols is not significant. In this file, only the number of tabs before the start of the text is significant.

The order of the objects is significant (source, input, line, circuit, outlet). The order of the individual settings underneath each object is not significant. However, this does not mean that something like switchState can be anywhere. It must be under, and correctly indented for, the outlet object to which the setting belongs.

While it is generally safe to edit the values of the settings, it is not advisable to alter the structure or order of the lines within file.

factorySettings.txt

This file is not editable, not deletable, and not replaceable. It may be copied for reference and archiving. It is formatted similarly to powerSettings.txt.

```
(a fragment of powerSettings.txt)
identity = RCM POWR
version = 1.0.0
// NOTE: user added comments are not preserved in this file.
// See the User Guide for details about format and options.
// This file's format IS sensitive to the leading whitespace tabs.
// Be sure your editor inserts tabs, and does not substitute spaces.
source~1
   label
>
                                 = Source 1
>
   input~1
   input~2
>
>
   line~1
       circuit~1
>>
           label
                                  = Circuit A
>>>
           outlet~1
>>>
               label
>>>>
                                  = Outlet 1
>>>>
               switchState
                                  = on
                                  = last known
>>>>
               startupState
>>>>
               onDelav
                                  = 0.20
               offDelay
                                  = 0.00
>>>>
>>>>
               cycleDelay
                                  = 5.00
>>>
           outlet~2
               label
>>>>
                                  = Outlet 2
>>>>
               switchState
                                  = on
>>>>
               startupState
                                  = last known
                                  = 0.20
>>>>
               onDelav
               offDelay
                                  = 0.00
>>>>
>>>>
               cycleDelay
                                 = 5.00
```



MPD Series RCM Software Firmware Updates



Firmware Updates

Should there be a need to update the firmware in the MPD RCM PDU, there's a very simple way to do that. A single file named image.bin is the main application for the PDU. The file will be enclosed in a folder identifying its version and accompanied by release notes, but the upload file will always be named simply image.bin.

Use an FTP client to upload the file to the on-board FLASH0 volume of the PDU. (What you see with your FTP program may look different to what's shown on the right). Wait approximately 2 minutes after the upload to ensure the file has been successfully copied and installed, then restart the PDU using the web interface or command line.

After the file has uploaded, it is moved to another location, so it won't remain visible in the FTP file listing.

FIRMWARE UPDATE SUPPORT

All firmware updates which have been made for security or bug patches are available at no cost from Marway's web site. Look for firmware updates at:

http://www.marway.com/pdu/mpd_rcm_software

There may or may not be any which apply to your system. Other conditions may apply. Check your warranty, and possibly your purchase agreement if your PDU was custom designed.





RCM Software[™] for MPD Series[™] Switched Outlet PDUs

Software Version:1.0.xOwner's Guide P/N:501014-000

© 2013, Marway Power Systems, Inc. All rights reserved.



Marway Power Solutions 1721 S. Grand Ave., Santa Ana, CA 92705 800-462-7929 • marway@marway.com