**Chapter 13: Alternate Management Interfaces**

Add a note hereThough the majority of users spend most of their time developing services and administering DataPower appliances from within the graphical WebGUI, there are a number of ways to access the device using nongraphical interfaces. The preferred method of interaction usually depends on the user’s past experience and current job description; many network administrators simply feel more comfortable using a Command Line Interface (CLI), whereas deployment engineers may want to script device interaction as a series of SOAP calls to the XML Management Interface, a secure listener for retrieving data or executing commands in an automated fashion. Retrieving a quick, interactive status update or running a series of batched, aliased commands using Secure Shell (SSH) may be easiest from the CLI, while the XML Management Interface lends itself to pushing configurations to large numbers of devices using HTTPS. These are both perfectly valid approaches, and this chapter focuses on the power of using these alternate management interfaces on the appliances. We discuss how to access these interfaces and how to employ the most common and helpful commands. Though these alternate methods of access provide nearly the complete functionality enabled by the WebGUI, this chapter focuses mainly on interactive monitoring and management. Less interactive methods for long-term logging and configuration management are discussed in [Chapter 14](http://www.books24x7.com/assetviewer.aspx?bkid=30903&destid=2478#2478), [“Logging and Monitoring,”](http://www.books24x7.com/assetviewer.aspx?bkid=30903&destid=2478#2478) and [Chapter 15](http://www.books24x7.com/assetviewer.aspx?bkid=30903&destid=2720#2720), [“Build and Deploy Techniques.”](http://www.books24x7.com/assetviewer.aspx?bkid=30903&destid=2720#2720) With that, let’s begin exploring our options.

**Add a note here****Command Line Interface**

Add a note hereThe CLI provides nearly all the same capabilities as the WebGUI interface, as well as some CLI-only features. You can create a Web Service Proxy, define a Processing Policy, and configure actions to act upon messages flowing through the device, all using text-based commands typed into a shell communicating to the device over SSH—but why would you want to? In the DataPower Services portion of this book, you learned how easy it is to drag and drop actions into a Processing Policy, configure a listener, and create a service using DataPower’s award-winning graphical interface; though entirely possible, there’s no need to investigate the syntax of each obscure CLI command to accomplish the same end goal. Instead, let’s focus on the strengths of the appliance’s CLI: quick access to extensive status information, easy debugging and modification of network configurations, manipulation/transfer of files to/from the appliance to various external servers, and configuration and firmware management.

**Add a note here****Accessing the CLI**

Add a note hereThere are three different ways to get to the device’s CLI. The first comes into play immediately upon receiving the appliance; in order to initialize the device after racking it in a datacenter, you must connect to it via serial cable and use the CLI in a terminal session to assign an Ethernet interface and enable the WebGUI. The other two connection methods, telnet and SSH, need to be enabled from either the serial interface or the WebGUI, as they are disabled by default. Telnet is included for backward compatibility and is not recommended due its unencrypted nature. To ensure an encrypted connection and verify the cryptographic authenticity of the server, SSH is the more secure choice.

Add a note hereFrom the initial login prompt, you need to enter configuration mode, a context that provides commands for device-level functionality. Type co, an abbreviation of configure terminal, at the login prompt to enter this mode. To enable SSH from the serial interface while in configuration mode, run the commands in Listing 13-1.

Add a note hereListing 13-1: Enabling SSH from the Command Line Interface

Add a note herexi50# co

Global configuration mode

xi50(config)# ssh 192.168.1.199 22

% Pending

SSH service listener enabled

xi50(config)# write mem

Overwrite previously saved configuration [y/n]? y

xi50(config)# exit

xi50# exit

Add a note hereRunning this command without a host or port specified attempts to bind SSH to the management Ethernet interface (mgmt). If no management interface has been defined, SSH will listen on all configured interfaces; be sure to define a specific address to isolate management traffic from application data traffic. In each case, SSH listens to the canonical port (22) unless otherwise specified. To turn off the SSH interface, no ssh is the appropriate command.

Add a note hereTo save these changes to the persisted file system and exit configuration mode, follow the command with a write mem and an exit command."Configuration Modes"

**Configuration Modes**

Add a note hereThe CLI is segmented into configuration modes that represent the context in which your commands are executed. You were introduced to the global configuration mode in Listing 13-1, and you’ll see more options (Action Configuration Mode, Flash Configuration Mode) as we delve deeper into the command line. In every case, you’ll enter the mode by typing its name or a unique abbreviation of that name (co, action, flash); you’ll exit and apply your changes by typing exit at the prompt.

**Add a note here****Navigating the CLI**

Add a note hereAfter SSH is enabled on an appliance, the CLI can be accessed remotely. You’ll need to connect to the port you defined previously using a client that supports SSH, such as PuTTY on Windows or the shell application ssh on most UNIX/Linux distributions. You will be prompted for a login and password, as shown in Listing 13-2."Login via SSH"

Add a note hereListing 13-2: Logging into an Appliance Through the CLI

Add a note hereUnauthorized access prohibited.

login: admin

Password: \*\*\*\*\*\*

Welcome to DataPower XI50 console configuration.

(C) 1999-2007 by DataPower Technology, Inc.

Version: XI50.3.6.1.4 build 154519 on 2008/03/14 13:10:30

Serial number: 2703063Z

xi50#

**Login via SSH**

Add a note hereMany traditional servers use password authentication for SSH login. However, DataPower requires a keyboard-interactive process to protect your credential information from potentially being sniffed or stolen during the cryptographic handshake. The device initiates a secure channel automatically, and then provides an encrypted login process. As a side effect of the initial connection, and depending on your client, you may see an extraneous login as: prompt; simply press Enter at this point. You are warned that Unauthorized access is prohibited and then presented with the prompt shown in Listing 13-2.

Add a note hereCLI permissions depend on either the access policies defined for your user group or on a combination of command groups and specifically provisioned users. See [Chapter 12](http://www.books24x7.com/assetviewer.aspx?bkid=30903&destid=2003#2003), [“Device Administration,”](http://www.books24x7.com/assetviewer.aspx?bkid=30903&destid=2003#2003) for more details. If granted access to the CLI with your username and password, you’ll find a prompt consisting of the device type (xa35, xs40, or xi50) followed by the pound sign (#). From this initial prompt, there are some initial, top-level commands that can be run, or you can enter the global configuration mode to make additional commands available. Enter global configuration mode by typing configure terminal (or co as a shortcut) at the CLI.

Add a note hereNotice the shortcut that just saved you sixteen keystrokes (configure terminal→co). CLI commands can always be abbreviated to the shortest unique string that identifies them; as you grow more comfortable with the range of commands available to you, try your hand at shortening them to save time."Global Configuration Mode"

**Global Configuration Mode**

Add a note hereAside from a few high-level commands that act on the appliance as a whole, most of your CLI time will be spent configuring or querying objects or status information on the device. These usually require you to be in a configuration mode. If your prompt does not contain some form of (config) and you’re having trouble with your commands not being recognized, enter global configuration mode by typing co at the prompt. The prompt changes to indicate you are in configuration mode. CLI commands can have various effects, depending on the context (configuration mode) in which they are executed. A full list of every available command in every configuration mode is available in the Reference Guide provided with the Resource CD that accompanies the DataPower appliances; this file can also be downloaded from the IBM support site. See [Chapter 2](http://www.books24x7.com/assetviewer.aspx?bkid=30903&destid=145#145), [“DataPower Quick Tour and Setup,”](http://www.books24x7.com/assetviewer.aspx?bkid=30903&destid=145#145) for more information. While we generally suggest that everyone RTFM (Read The Fantastic Manuals), this interface should generally feel familiar to anyone used to dealing with other datacenter hardware, such as routers and load balancers. This familiarity combined with powerful context-sensitive help should make CLI navigation relatively self-explanatory. When in doubt, let’s see how to get help.

**Add a note here****Help is Your Friend**

Add a note herePerhaps the most useful command available from the CLI is the help command. This provides context-based information about the commands available to you at any given point in CLI navigation. If you have just logged in, help displays the high-level commands that can be run. In configuration mode, help displays a long list of available commands to enter other modes, perform operations, and so on. When in a more specific mode such as Web Service Proxy (WSP) configuration mode, help shows you the available options for modifying the WSP in question. In addition, you can execute the help command on any of the commands available to you to get more information about parameters and syntax. Let’s take a look at a couple of examples in Listing 13-3.

Add a note hereListing 13-3: Using Context-Sensitive Help in the CLI

Add a note herexi50# help

clock Sets the date or time

configure terminal Enters Global Configuration Mode

disable Enters User Mode

disconnect Terminates a user session

echo Echoes text to the console

exec Executes a configuration script

exit Terminates the CLI connection

help Displays general or command-specific help

login Logs in as a specific user

ntp Identifies a network time server

ping Sends an ICMP echo request

show Displays statistical or configuration data

shutdown Shuts down or restarts the system

switch domain Moves to a different application domain

test tcp-connection Test TCP reachability

top Returns to initial login mode

traceroute Displays the route to a host

xi50# help show

show [what]

Displays configuration or status information relevant to the provided

argument. If no argument is provided, a list of available options is

displayed.

Add a note hereAn initial execution of help shows the commands available. By executing help again and referencing one of these possible commands, we get more information about that command’s syntax and purpose. In this case, the command we learned about is one of the most useful: the show command.

**Add a note here****“Show” Me the Money**

Add a note hereWe’ve learned how to get help; let’s talk about one additional command that is useful in just about every context: the show command. Show provides the current status or configuration of any object on the appliance. If there’s a status page for the object in the WebGUI, there’s a show command for it in the CLI. Similarly, if there’s a configuration page for an object, we can view that configuration with a show command. We’ll see a lot of examples in the following sections—this command is ubiquitous. Along with show commands, we’ll also discuss commands used for network configuration and troubleshooting, device information and health retrieval, object modification, and file system operations. Let’s dive right into some helpful command examples.

**Add a note here****Network Configuration Commands**

Add a note hereUpon receipt of an appliance, the device administrator used a serial cable and the CLI to assign an IP address and default gateway, and turn on the Web management interface. It’s likely that all further network configuration—including DNS (for hostname resolution), NTP (for time synchronization), and additional interface enablement—was done from the WebGUI. This is perfectly reasonable, but sometimes you need to perform these operations from the CLI as well. Let’s walk through some of the more common network commands.

Add a note hereWe’ll start from the beginning and see what’s already been done using [Chapter 2](http://www.books24x7.com/assetviewer.aspx?bkid=30903&destid=145#145). First, an IP address was assigned to at least one Ethernet interface. We can use the show interface command (shortened to show int) to display the current status of each interface, including configured IP addresses and transferred data, as shown in Listing 13-4.

Add a note hereListing 13-4: Displaying the Current Ethernet Interface Configuration

Add a note herexi50(config)# show int

interface IP Address RX (kb/pkts/errs) TX (kb/pkts/errs)

--------- ---------- ----------------- -----------------

eth1 10.32.40.80/32 0/0/0 0/0/0

eth2 172.18.206.210/23 0/0/0 0/0/0

eth4 192.168.1.199/24 2425/14535/0 14954/17314/0

eth0 0.0.0.0/0 0/0/0 0/0/0

Add a note hereFor more specific information about the configuration of one Ethernet interface in particular, we can expand our show int command to include the name of an interface. The management interface (mgt0) is listed as eth4 in the previous example; the names are synonymous when referencing this object. The more granular show command executed in Listing 13-5 provides us information about the MAC address, status, and interface mode.

Add a note hereListing 13-5: Retrieving More Information About an Ethernet Interface

Add a note herexi50(config)# show int mgt0

interface: mgt0

IP address: 192.168.1.199/24

MAC address: 00:0a:4b:80:06:54

status: ok

negotiate: auto

mode: 1000baseTx-FD

MTU: 1500

RX kbytes: 2433

RX packets: 14624

RX errors: 0

RX drops: 0

TX kbytes: 14960

TX packets: 17383

TX errors: 0

TX drops: 0

collisions: 0

Add a note hereThe next setup step is to enable the Web Management Interface. We can check to see which IP:port is assigned for the WebGUI and modify this if it needs to be changed. The show web-mgmt command displays the current settings; web-mgmt *ip-address port timeout* sets the new values, as shown in Listing 13-6.

Add a note hereListing 13-6: Viewing and Modifying the Web Management Interface Listener

Add a note herexi50(config)# show web-mgmt

web-mgmt [up] (modified)

--------

admin-state enabled

ip-address 0.0.0.0

port 9090

save-config-overwrite on

idle-timeout 600 seconds

acl web-mgmt [up]

xi50(config)# web-mgmt 192.168.1.199 9090 3600

Web management: successfully started

Add a note hereIn this example, the WebGUI was configured to listen on every enabled Ethernet interface. To restrict access to just the management interface, we set it to listen on a specific IP (192.168.1.199). We also raise the WebGUI idle timeout value from the default (600 seconds = 10 minutes) to 3600 seconds, or one hour. This enables DataPower users to pause during development without being timed out in the middle of a task. WebGUI timeouts should be set based on your internal security requirements (How long does it take your workstation to go to a locked screensaver?) and common sense. Turning off the timeout entirely by setting a value of 0 creates a potential security risk in the case of an unlocked workstation, and allows for orphaned browser sessions that consume device resources. For these reasons, removing the idle timeout is *not* recommended.

Add a note hereThe next piece of network configuration is the selection of DNS servers for hostname resolution. While this will likely be done using the WebGUI, we can view the configuration with our handy show command, as seen in Listing 13-7.

Add a note hereListing 13-7: Viewing Current DNS Settings

Add a note herexi50(config)# show dns

dns [up]

---

admin-state enabled

search-domain xyco.com

search-domain xycorp.com

name-server 172.20.135.156 53 53 3

name-server 172.21.135.156 53 53 3

Add a note hereThe DNS search domains have been defined to match our internal domain names, so a remote device referenced as server1 is looked up using server1.xyco.com and server1.xycorp.com. Having multiple DNS servers ensures that a single server failure is not the cause of network issues. The values after the IP addresses represent the UDP port, TCP port, and number of retries before failure.

Add a note hereNow that DNS has been configured, we can reference hostnames in any additional configuration objects. For example, we can set up our NTP server connectivity with a simple reference (ntp time.nist.gov) to a time server.

**Add a note here****Network Troubleshooting Commands**

Add a note hereNow that we’ve defined network connectivity options (Ethernet interfaces, management interfaces, DNS servers), we should test to make sure that the appliance can connect to other nodes in the network, including backend application servers, authentication/authorization servers, and any supporting Web servers. This can be done from the WebGUI as well, but there are often restrictions on having active Web interfaces in the Demilitarized Zone (DMZ). Let’s walk through some example commands that might be useful under these circumstances. Some of these will also be discussed in [Chapter 27](http://www.books24x7.com/assetviewer.aspx?bkid=30903&destid=4747#4747), [“Problem Determination Using Built-In Tools,”](http://www.books24x7.com/assetviewer.aspx?bkid=30903&destid=4747#4747) but they’re worth noting here as well.

Add a note hereIf there appears to be an issue in routing to the appropriate remote network, traceroute will display each hop to the targeted server. A failure to find the remote server indicates network inaccessibility, and the last successful gateway should give you an indication of where the problem lies. In Listing 13-8, we can see the full path to Google.

Add a note hereListing 13-8: Determining a Network Path to a Server

Add a note herexi50# traceroute www.google.com

Traceroute www.google.com (64.233.167.147)

1: 192.168.1.1 rtt= 0 ms

2: 76.176.224.1 rtt= 1 ms

3: 76.176.0.25 rtt= 0 ms

4: 76.176.0.13 rtt=11 ms

5: 4.79.36.9 rtt=52 ms

6: 4.68.113.69 rtt=43 ms

7: 4.68.101.34 rtt=59 ms

8: 4.79.208.18 rtt=58 ms

9: 72.14.232.53 rtt=57 ms

10: 64.233.175.42 rtt=61 ms

11: 64.233.167.147 rtt=50 ms

Add a note hereTo test general connectivity, you can ping a remote host or IP address, as shown in Listing 13-9. A failure to ping usually represents either host inaccessibility, network firewall misconfiguration, or simply that ICMP echo requests (the utility used by ping) have been disabled on the target server. Keep this last possibility in mind—do not assume that a failed ping means no addressability. Perform a similar ping test from another server/workstation in the same network with demonstrated connectivity. If that test is successful but DataPower is not, it’s likely a firewall/Access Control List (ACL) issue.

Add a note hereListing 13-9: Ensuring Network Connectivity to a Remote Server

Add a note herexi50(config)# ping 192.168.1.103

PING 192.168.1.103 (192.168.1.103) with 56 data bytes of data

64 bytes from 192.168.1.103: seq=0, ttl=128, rtt=0.0 ms

64 bytes from 192.168.1.103: seq=1, ttl=128, rtt=0.0 ms

64 bytes from 192.168.1.103: seq=2, ttl=128, rtt=0.0 ms

3 packets transmitted, 3 received, 0% loss, time 6001ms

xi50(config)#

Add a note hereAfter basic connectivity has been established, you may want to verify a listener on a specific TCP port. This IP:port combination could represent a backend application or an authentication server, such as the attempt to verify an LDAP installation shown in Listing 13-10. You may have performed this sort of test before using telnet from a server or workstation command shell.

Add a note hereListing 13-10: Testing for a Listener on a Specific TCP Port

Add a note herexi50# test tcp-connection 192.168.1.198 389

TCP connection successful

Add a note hereIf any of the preceding commands fail, check whether the appropriate ports are open on any internal firewalls, and whether the appliance’s IP addresses are whitelisted in any server-side ACL. If they all succeed and there is still a connectivity problem, check for SSL communication issues and/or application-layer access control.

**Add a note here****System Information Commands**

Add a note hereThere are a few times when it becomes necessary to retrieve system information from an appliance, including that unit’s hardware type, serial number, firmware version, and applicable license data. One of these moments is when upgrading the firmware; the other is when contacting IBM for DataPower support. In each of these cases, there are a series of CLI commands that become especially useful.

Add a note hereThe support contract for DataPower appliances is tied to the device’s serial number, so this will be one of the first things that Customer Support asks for when opening a support ticket with IBM. This can be retrieved with the show system command, as seen in Listing 13-11. The description tells you the model number (9002/3/4), and the Product ID field contains information about the hardware type (9003 in this example). These are both necessary pieces of information when selecting a firmware file for upgrading an appliance.

Add a note hereListing 13-11: Discovering an Appliance Serial Number

Add a note herexi50(config)# show system

description: DataPower XI50

serial number: 2703063Z

product id: 9003-XI50-03 [Rev 16]

OID: 1.3.6.1.4.1.1468.1.3

uptime: 0 days 01:57:23

contact: deviceadmin@xyco.com

name: John Smith

location: Peoria, IL

services: 72

login-message: Welcome to XMLGateway12

Add a note hereDetermining the current version of the firmware is often useful when checking for updates that may provide some new functionality. The show version command seen in Listing 13-12 displays the current firmware version and the versions of additional libraries used by the device. You will likely want these library versions to remain the same in the new firmware image. For instance, if the Tivoli Access Manager (TAM) version is listed as 6.0, be sure to download the firmware image that includes TAM6.

Add a note hereListing 13-12: Querying for Firmware and Library Versions

Add a note herexi50(config)# show version

Serial: 2703063Z

Version: XI50.3.6.1.4

Build: 154519

Build Date: 2008/03/14 13:10:30

Watchdog Build: XI50.3.6.1.4

Installed DPOS: XI50.3.6.1.4

Running DPOS: XI50.3.6.1.4

XML accelerator: xg3

Coproc library: 1.2

MQ: 6.0.0-0

ODBC: 5.2

TAM: 6.0

Tibco EMS: 4.4.0

WebSphere JMS: 1.2.3

Add a note hereTo see a full list of options for which the appliance is licensed, use the show license command as seen in Listing 13-13. If the feature is listed and marked as Enabled, this appliance can use that functionality; if the feature is not in the list, this appliance will not execute that functionality. Attempting to upgrade a firmware image containing unlicensed options will cause the update to fail. This has no negative side effects, but the firmware is not upgraded as desired.

Add a note hereListing 13-13: Gathering a List of Licensed Options

Add a note herexi50(config)# show license

Feature Enabled

------- -------

MQ Yes

TAM Yes

DataGlue Yes

JAXP-API Yes

PKCS7-SMIME Yes

SQL-ODBC Yes

Tibco-EMS Yes

WebSphere-JMS Yes

Add a note hereSome features are included with every firmware image; for example, DataGlue is licensed and installed on every XI50 appliance. These features do not require a firmware image that specifically includes them in the filename. For those options that *do* require special firmware (including MQ, TAM, Tibco, and ODBC), make sure you download the appropriate image.

Add a note hereFrom the device information returned by the three previous commands, we can determine that this appliance is an XI50 running on 9003 hardware and 3.6.1.4 firmware, and licensed for MQ, Tibco, ODBC, and TAM (Version 6). To upgrade this device to the next firmware release, the image filename should look like the following: xi3615.9003.mq6.odbc52.tam6.tibco.scrypt2.

**Add a note here****Appliance Load and Health Monitoring Commands**

Add a note hereMonitoring the current health of the DataPower appliance is generally handled by a network operations/monitoring team using standards such as SNMP and syslog. Setup of these interactions is thoroughly discussed in [Chapter 14](http://www.books24x7.com/assetviewer.aspx?bkid=30903&destid=2478#2478), [“Logging and Monitoring.”](http://www.books24x7.com/assetviewer.aspx?bkid=30903&destid=2478#2478) It is also sometimes helpful for an administrator to be able to submit a real-time request for hardware status.

Add a note hereFirst let’s focus on indicators of processing load. Fittingly enough, the best metric to use for overall device status is the show load command shown in Listing 13-14. This command, called System Usage in the WebGUI status menu, takes into account the current CPU, memory, and connection information to provide a single value (load %).

Add a note hereListing 13-14: Checking Load Provides a Good Indication of Device Usage

Add a note herexi50(config)# show load

interval: 1000 msec

load: 1 %

work list: 24

Add a note hereWhile system usage is the best overall indicator of device workload, many enterprises also desire measurements for common server characteristics, such as memory and CPU. Keep in mind that the CPU or memory usage may be temporarily high, even under low device load, due to optimization of message processing. Though they should not be used to trigger alerts or device health concerns, these statistics are available through the CLI as well. In Listing 13-15, the memory usage is shown as a percentage, along with specific byte counts. CPU usage percentage is shown at several intervals, demonstrating trends that may occur over a larger time period.

Add a note hereListing 13-15: Assessing the Device Memory and CPU Usage

Add a note herexi50(config)# show mem

Memory Usage: 12 %

Total Memory: 4150124 kbytes

Used Memory: 515859 kbytes

Free Memory: 3634265 kbytes

Requested Memory: 652800 kbytes

xi50(config)# show cpu

10 sec 1 min 10 min 1 hour 1 day

cpu usage (%): 3 3 15 15 15

Add a note hereWhile file system usage statistics have no impact on workload or current processing, they also represent commonly tracked metrics. If the file system becomes full, it could have an impact on on-device logging and other activities that use either the encrypted file store or the temporary file system loaded in memory. Using the CLI to check the file system space available, as shown in Listing 13-16, is an easy way to figure out how much of an impact file system operations (copying files on/off of the device, deleting files) have had on the free space. The encrypted space metrics refer to long-term directories such as local, store, and cert, while temporary space refers to directories that are cleared on device restart, such as temporary and logtemp.

Add a note hereListing 13-16: Checking for Free Space on the Device File System

Add a note herexi50(config)# show filesystem

Free Encrypted Space: 245 Mbytes

Total Encrypted Space: 475 Mbytes

Free Temporary Space: 234 Mbytes

Total Temporary Space: 242 Mbytes

Add a note hereDigging even deeper into actual hardware statistics, the show sensors and show sensors-fans commands provide information from internal environmental sensors. These include temperatures, fan speeds, voltages, power supply status, and intrusion detection sensor status. As seen in Listing 13-17, each CPU has its own temperature measurement and each fan has its own individual speed information.

Add a note hereListing 13-17: Monitoring the Internal Environmental and Hardware Sensors

Add a note herexi50(config)# show sensors

Ambient Temperature: 24 C

CPU 1 Temperature: 34 C

CPU 2 Temperature: 36 C

CPU 1 Fan speed: 7031 RPM

CPU 2 Fan speed: 6750 RPM

Chassis Fan 1 speed: 11250 RPM

Chassis Fan 2 speed: 9926 RPM

Chassis Fan 3 speed: 10546 RPM

Intrusion Detected: no

3.3 Voltage: 3.29 V

5 Voltage: 4.91 V

12 Voltage: 12.00 V

Power supply status: ok

Battery installation date: Mon Aug 6 09:54:20 2007

xi50(config)# show sensors-fans

Fan ID Speed

------ -----

chassis-1 7031

chassis-2 6750

chassis-3 11250

chassis-4 10546

chassis-5 10546

chassis-6 6750

chassis-7 7031

chassis-8 9926

Add a note hereNote that some of the monitoring and statistics commands require that you first enable the gathering of statistics on the device. This is particularly true of those commands that output values over time (1 second, 1 minute, 1 hour, etc.). Enter Global Configuration mode in the CLI (co) and then type statistics to enable statistics.

**Add a note here****Object Modification Commands**

Add a note hereAs we’ve mentioned, the command line interface is tailored very well to some tasks—namely, retrieving real-time status information, making quick changes to network and device configurations, and performing file system operations to move firmware, configurations, and files to and from the appliance. That said, the CLI can also be used to create and modify higher-level objects such as services, policies, and actions. Due to the large numbers of configuration fields and compatibility options, defining a service and all its supporting objects is generally not practical, but making small changes and creating simple objects is common, and all the commands are there if more is necessary. In this case, the Reference Guide and context-sensitive CLI help are the best places to look for the syntax and field defaults for each object; there are simply too many to cover here. However, we can give you some insight into the general methodology for making these modifications.

Add a note hereWhen you first log in to the CLI, you enter into an initial login mode that contains only a few available commands that act on the device as a whole. By switching into global configuration mode with the configure terminal command, you can open a whole new world of options—over two hundred of them! Try not to be overwhelmed; just be very specific about what you want to create/edit, and then navigate to that object’s configuration mode. For instance, let’s edit a Filter action that we’ve defined within a service’s Processing Policy.

Add a note hereIf you’ve followed our suggestions, there won’t be any services defined in the default domain. Those will be restricted to an application domain that has been created for the project or for the development group that’s working on that service. So, your first step to tracking down the action to edit is to move into the appropriate domain. The switch domain command takes you there. In Listing 13-18, we change our context from the default domain (no domain designation) into the HRServices domain, and the cursor changes to reflect our current position. You would have to execute the command switch domain default to get back into the default domain."CLI Permissions"

Add a note hereListing 13-18: Navigating to a Particular Domain with the Switch Domain Command

Add a note herexi50(config)# switch domain HRServices

xi50[HRServices](config)#

**CLI Permissions**

Add a note hereTo this point, we’ve been executing device-level commands in the default domain. These have required admin-level access to administrative commands. Now that we’re switching domains and focusing on configuration objects, you’ll have to make sure your user has CLI access to the domain; you may also need permission to additional commands. Talk to your device administrator about whether Role-Based Management is being enforced on the CLI (in which case you likely have all the permissions you’re going to get; use show rbm to see whether RBM for CLI is active), or whether additional permissions need to be provided to your user group or domain. See [Chapter 12](http://www.books24x7.com/assetviewer.aspx?bkid=30903&destid=2003#2003) for more information.

Add a note hereNow that we’re in a particular domain, let’s track down the action we want to modify. A simple show action command lists all the actions in the domain; depending on the extent of development, this can be a long list! Narrow it down by adding an action name to retrieve only the configuration information for that action. In Listing 13-19, we see all the configuration options that can be set for a Filter action—we warned you this was easier from the WebGUI.

Add a note hereListing 13-19: Reviewing the Configuration for a Particular Action

Add a note herexi50[HRServices](config)# show action HRService\_response\_Filter

action: HRService\_response\_Filter [up]

---------------------------------

admin-state enabled

type filter

input tempvar2

transform local:///HRService.xsl

named-inouts default

transactional off

soap-validation body

sql-source-type static

asynchronous off

results-mode first-available

retry-count 0

retry-interval 1000 msec

multiple-outputs off

iterator-type XPATH

timeout 0 msec

Add a note hereTo edit the action, we need to switch to the action configuration mode. Navigating to the configuration mode of any object is usually just a matter of combining its type and its name. If you want to create a new object, just use a new, unique name. In this case, action HRService\_response\_Filter gets us to where we want to be, as seen in the first line of Listing 13-20; the prompt changes to display both the domain (HRServices) and the current context (config action HRService\_response\_Filter). Now that we’re editing the action, we need to find the appropriate command to change a property. Using the help command to get additional information provides us with a long list of possible options; in fact, it provides every command for every type of processing action, not just the Filter Actions. However, at the end of the list are some commands that are common to all object configuration menus, as shown in Listing 13-20.

Add a note hereListing 13-20: Getting Help on Common Object Configuration Commands

Add a note herexi50[HRServices](config)# action HRService\_response\_Filter

Modify Processing Action configuration

xi50[HRServices](config action HRService\_response\_Filter)# help

\*\*\* *Full list of commands snipped here* \*\*\*

admin-state Administratively enables or disables the object.

cancel Exits the current mode (no changes applied)

disconnect Terminates a user session

echo Echoes text to the console

exit Applies properties and exits

help Displays general or command-specific help

ping Sends an ICMP echo request

reset Exits the current mode (default values applied)

show Displays statistical or configuration data

test tcp-connection Test TCP reachability

traceroute Displays the route to a host

xi50[HRServices](config action HRService\_response\_Filter)#

Add a note hereOne command that is common to all objects and often useful for taking quick action is the admin-state property. This enables or disables the object. Disabling an object (admin-state disabled) removes it from the possibility of execution without deleting or changing it in any other way. Enabling the object again (admin-state enabled) brings it back. As with all modification commands executed within configuration modes, this enable/disable change applies as soon as you exit the object configuration."Admin-state"

**Admin-state**

Add a note hereThe admin-state command sets an object to be enabled or disabled. While this is one method of removing an object (domain, policy, rule) from service, keep in mind that any objects dependent on a newly disabled object are impacted. If an action is disabled, all rules that reference the action are also down. This affects policies that reference the rule and services that reference those policies. Make sure this ripple effect is intended before changing the administrative state of any object.

Add a note hereOther common object commands tend to be meta-commands, in that they act on the other changes you have already made. The exit command applies the changes you have made to the object. If you change your mind, the cancel command exits the object configuration mode without applying the changes. The reset command also exits the current mode, but not before applying the *default* values to the object; this essentially sets the object back to factory settings.

Add a note hereBefore we can decide whether we want to save or discard our changes, we have to actually modify the object. We’ve been dealing with a Filter action that uses a stylesheet (local:///HRService.xsl) to perform its filtering duties. This is evident from the line that says transform local:///HRService.xsl in Listing 13-19. To change the execution to a new version of the stylesheet, just replace the line by referencing the transform parameter, as shown in Listing 13-21. The show command, executed within the action configuration mode, then provides a nice contextual display of the parameter values, including the new stylesheet name.

Add a note hereListing 13-21: Setting a New Transform, and then Checking Our Work

Add a note herexi50[HRServices](config action HRService\_response\_Filter)# transform

local:///hrservice2.xsl

xi50[HRServices](config action HRService\_response\_Filter)# show

admin-state enabled

type filter

input tempvar2

transform local:///hrservice2.xsl

*\*\*\* Lines snipped for brevity \*\*\**

Add a note hereAll changes that affect the configuration of objects on the appliance should be applied and then saved to the persistent configuration using the write mem command. This command can be executed only in the global configuration mode—it will not work from object configuration modes. Use the exit command to move out of the configuration contexts, saving changes as you progress. Save the configuration, and then exit the entire CLI when you’ve completed your work.

Add a note hereExiting the CLI is an important step—there is no predefined idle timeout as there is in the WebGUI. If your workstation is not locked, anyone can walk up and administer the device using your permissions.

**Add a note here****File and Configuration Management Commands**

Add a note hereAnother common use case of the CLI is the execution of commands that have an effect on the entire device at once. Backing up a configuration, upgrading the firmware, and rebooting the device are all simple. They also require some file management, to move files onto and off of the device. Let’s walk through the process of backing up specific domains, copying the backup file off of the appliance, copying a firmware image onto the device, and then upgrading the firmware by booting to the new image.

Add a note hereThe backup command combines domains into a single ZIP configuration file for moving off the device to a source control or backup/archive server. There are three commands shown in Listing 13-22. First, the backup command takes a filename, followed by a list of domains to include in the backup. Next, the dir command provides a simple directory listing; all domain exports are placed in the export: directory. Finally, the copy command copies the file from the export: directory to a remote configuration server.

Add a note hereListing 13-22: Exporting and Offloading a Backup File from the Appliance

Add a note herexi50(config)# backup AllHRServices.zip HRServices CoreServices

Backup to 'AllHRServices.zip' scheduled (may take a few minutes to

complete).

Backup is complete.

xi50(config)# dir export:

File Name Last Modified Size

--------- ------------- ----

CoreServices/ Tue Apr 8 00:33:53 2008 1024

HRServices/ Tue Apr 8 00:33:53 2008 1024

AllHRServices.zip Tue Apr 8 02:40:57 2008 1112948

233.1 MB available to export:

xi50(config)# copy export:///AllHRServices.zip

scp://jsmith@cfgserver.xyco.com/AllHRServices.zip

Password: \*\*\*\*\*\*\*

File copy success

Add a note hereThe copy command is powerful, as it can copy files on and off the appliance using several different protocols, including http, https, scp, sftp, and smtp. Local URLs are referred to using a *directory*:///*filename* syntax; notice the *three* slashes in every on-device directory URL. The syntax for remote URLs depends on the protocol. See the help copy command and the Reference Guide for the complete syntax descriptions. To see another example and segue into firmware management, let’s copy a new image to the device and then upgrade the firmware.

Add a note hereListing 13-23 uses the copy command to retrieve a file from a remote server and the dir command to display a directory listing for file verification; firmware upgrades should always be placed in the image: directory. The flash command then switches the context to flash configuration mode, where the device is booted using the new image. This boot image command produces a string of status indicators, after which the new firmware has been successfully installed and the device will reboot.

Add a note hereListing 13-23: Upgrading Using a New Firmware Image

Add a note herexi50(config)# copy https://cfgserver/xi3614.9003.mq6.tam6.scrypt2

image:///xi3614.9003.mq6.tam6.scrypt2

File copy success (49725285 bytes copied)

xi50(config)# dir image:

File Name Last Modified Size

--------- ------------- ----

xi3614.9003.mq6.tam6.scrypt2 Tue Apr 8 03:21:08 2008 49725285

185.5 MB available to image:

xi50(config)# flash

Flash configuration mode

xi50(config-flash)# boot image xi3614.9003.mq6.tam6.scrypt2

Add a note hereTo be assured of a successful firmware upgrade process, you should upload the firmware, reload the currently-running firmware as shown in Listing 13-24, and then boot to the new firmware image. Reloading the firmware clears out any unneeded temporary files from the file system.

Add a note hereThe appliance always reboots after a successful firmware upgrade. However, there are times when you may want to shut down or reboot the device manually. You can also reload the firmware on the device, which takes only a few seconds and doesn’t require a full reboot; it doesn’t even reboot the domains! These commands all allow for a timer countdown, in order to delay the actual shutdown/reboot. In Listing 13-24, each of the commands would execute ten seconds after responding affirmatively (y) to the prompt.

Add a note hereListing 13-24: Evaluating the Appliance Shutdown Options

Add a note herexi50# shutdown reboot 10

You have requested a system shutdown reboot. Do you want to continue?

[y/n]: n

xi50# shutdown halt 10

You have requested a system shutdown. Do you want to continue? [y/n]: n

xi50# shutdown reload 10

You have requested a system shutdown reload. Do you want to continue?

[y/n]: n

**Add a note here****Aliases**

Add a note hereExecuting individual CLI commands is useful, but putting them together into batches is even more powerful. To enable this type of batch processing, use the alias command, which creates aliases comprised of a series of individual commands. These aliases can represent commonly used tasks, such as those demonstrated earlier in this chapter. In Listing 13-25, we’ve created two aliases called disableHRService and enableHRService, which do exactly what their names imply. The alias command creates the sequence of commands, which are then executed using the alias name. In this case, the disableHRService alias switches to the HRServices domain, opens the WS-Proxy called HRService for modification, sets the service’s admin-state to disabled, exits to apply the changes, and then switches back to the default domain to wait for the next command. The full interaction with the CLI isn’t displayed by the alias execution, but the end result is that the Web Service Proxy is disabled.

Add a note hereListing 13-25: Defining and Executing a Command Alias

Add a note herexi50(config)# alias disableHRService "switch domain HRServices; wsgw

HRService; admin-state disabled; exit; switch domain default"

Alias update successful

xi50(config)# alias enableHRService "switch domain HRServices; wsgw

HRService; admin-state enabled; exit; switch domain default"

Alias update successful

xi50(config)# show alias

Alias Name Command

---------- -------

disableHRService switch domain HRServices; wsgw HRService;

admin-state disabled; exit; switch domain default

enableHRService switch domain HRServices; wsgw HRService;

admin-state enabled; exit; switch domain default

xi50(config)# disableHRService

Modify Web Service Proxy configuration[[1](http://www.books24x7.com/assetviewer.aspx?bookid=30903&chunkid=553296372&noteMenuToggle=0&hitSectionMenuToggle=0&leftMenuState=1" \l "ftn.ch13fn01)]

xi50(config)#

Add a note hereWe have now used the CLI from appliance initialization all the way to shutdown. In between, we used the CLI to set up DNS and management listeners, verify network connectivity, retrieve device information and health statistics, modify object configurations, move files to and from the file system, back up configurations, and upgrade the firmware. We’ve also combined these commands into a series of processing tasks using the alias command. These are the most common and useful CLI commands that are available and should be of great use to network personnel, device administrators, and configuration management experts.

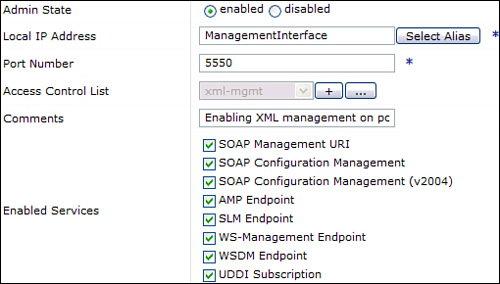
Add a note here[[1](http://www.books24x7.com/assetviewer.aspx?bookid=30903&chunkid=553296372&noteMenuToggle=0&hitSectionMenuToggle=0&leftMenuState=1" \l "ch13fn01)]This text is a side effect of the alias execution. Even though the rest of the commands and CLI prompts aren’t shown, rest assured that the modifications have taken place.

## XML Management Interface

Add a note hereThe XML Management Interface presents an interface to retrieve status information, modify configurations, and perform operations on the device by sending SOAP-formatted Web service calls to a user-defined address and port on the device. The same full range of configuration options and status gathering operations offered by the WebGUI are available through the XML Management Interface. This method of management will be referred to as SOAP management in the sections to follow. The XML Management interface is defined to listen on a device IP and port using SSL. This listener also supports other standards for automated device interaction, such as WSDM[[2](http://www.books24x7.com/assetviewer.aspx?bookid=30903&chunkid=928483369&noteMenuToggle=0&hitSectionMenuToggle=0&leftMenuState=1" \l "ftn.ch13fn02)], WS-Management, and AMP[[3](http://www.books24x7.com/assetviewer.aspx?bookid=30903&chunkid=928483369&noteMenuToggle=0&hitSectionMenuToggle=0&leftMenuState=1" \l "ftn.ch13fn03)], some of which are discussed in [Chapter 14](http://www.books24x7.com/assetviewer.aspx?bkid=30903&destid=2478#2478). For this discussion, we focus on just the SOAP management requests, which are sent to the XML management interface using HTTP Basic Authentication as a credential mechanism. Once received, these SOAP management calls will execute the desired operations on the appliance and will be subject to all the same permissions given to the user for WebGUI administration.

### Add a note hereEnabling the SOAP Management Interface

Add a note hereTo enable the SOAP listener, navigate in the WebGUI to Network→Management→XML Management Interface in the menu on the left. As shown in Figure 13-1, the service must be enabled and configured with an interface and port for the listener. Selecting the enabled services will allow for various requests to this interface. The SOAP-prefixed options are important for SOAP Management. For instance, the SOAP Configuration Management service listens on a URI of /service/mgmt/current and reflects the latest API. The option appended with (v2004) is available for legacy purposes and should not be used unless there are existing applications connecting to this interface using a legacy API. The SOAP Management URI option allows messages to be received on any URI. This is also for legacy purposes and should be enabled only if you have existing applications that connect to a URI other than those specifically defined. As always, administrative listeners (WebGUI, SSH, XML management) should be locked down to an Ethernet management interface for maximum security.

[](javascript:PopImage('IMG_309','http://images.books24x7.com/bookimages/id_30903/13fig01_alt.jpg','510','290'))  
Add a note hereFigure 13-1: Turning on the XML Management listener on port 5550.

### Add a note hereSubmitting SOAP Management Requests

Add a note hereBecause this is a Web service interface, valid requests must be POSTed to the SOAP management URL, using the appropriate URI, and contain an HTTP Basic Authentication header containing the user’s credentials. This can be accomplished using a simple Web services client, including cURL, SoapUI, XMLSpy, or any application platform with a Web service stack (.Net, J2EE, and so forth). The listener uses SSL by default; therefore, you may need a copy of the public certificate that will be sent during the SSL handshake. This file is named root-ca-cert.pem and is available for download from the DataPower Appliance Firmware site, or from the Resource CD that came with the appliance. Alternatively, a new SSL profile can be defined from the XML Management configuration page in the WebGUI.

Add a note hereA sample SOAP management request could be submitted using curl[[4](http://www.books24x7.com/assetviewer.aspx?bookid=30903&chunkid=928483369&noteMenuToggle=0&hitSectionMenuToggle=0&leftMenuState=1" \l "ftn.ch13fn04)] with the following syntax:

Add a note herecurl --data-binary @GetSystemUsage.xml -u jsmith:passwd

https://xmlgw01:5550/service/mgmt/current

### Add a note hereDefining the Management Service API

Add a note hereThe SOAP management Web service listening on the XML management interface is defined by a WSDL that is stored on the device. The WSDL and supporting schema documents are in the store: directory and can be downloaded from the File Management area in the WebGUI.

* Add a note herexml-mgmt.wsdl is the WSDL file that describes the services available from the XML Management interface.
* Add a note herexml-mgmt-ops.xsd is the schema file that lists the operations available for execution. Operations include get-file, get-log, do-backup, do-action, modify-config, and many more. See the Reference Guide for an explanation of each and this schema file for the operation syntax.
* Add a note herexml-mgmt.xsd is the schema file that defines the syntax for DataPower objects to be acted on. This includes high-level objects such as XMLFirewallService, low-level objects such as FilterAction, and even device-level objects such as Domain and TimeStatus.
* Add a note herexml-mgmt-base.xsd is the schema file that defines the primitive types used in xml-mgmt.xsd. These include text and numerical data types that map to XML Schema primitive types.

Add a note hereThese four files provide the basis for every SOAP configuration call that can be made to the appliance. For a full understanding of the message structure, download and review the files.

#### Request Message Structure

Add a note hereEvery request message has the same general structure, as seen in Listing 13-26. The SOAP body contains a request element in the dp: DataPower management namespace. That request element then contains the operation element, which in turn contains its own parameters. The only optional modification to this structure is the addition of a domain=*domainName* attribute on the dp:request element. This restricts the effects of the call to a single domain.

Add a note hereListing 13-26: Defining the SOAP Structure for SOAP Management Requests

Add a note here<?xml version="1.0" encoding="UTF-8"?>

<env:Envelope xmlns:env="http://www.w3.org/2001/12/soap-envelope">

<env:Body>

<dp:request xmlns:dp="http://www.datapower.com/schemas/management">

...

</dp:request>

</env:Body>

</env:Envelope>

#### Response Message Structure

Add a note hereEvery response message also has the same general structure, as seen in Listing 13-27. The SOAP body contains a response element in the DataPower management namespace. That response element then contains a timestamp element, which defines the moment at which the command was executed, followed by a response from the operation.

Add a note hereListing 13-27: Demonstrating the Response Structure from SOAP Management Requests

Add a note here<?xml version="1.0" encoding="UTF-8"?>

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Body>

<dp:response

xmlns:dp="http://www.datapower.com/schemas/management">

<dp:timestamp>2008-04-10T15:59:16-07:00</dp:timestamp>

...

</dp:response>

</env:Body>

</env:Envelope>

#### Common SOAP Management Operations

Add a note hereThough the XML management interface provides the full range of configuration options, some tasks are often better suited for other interfaces. For example, comparing configurations is much easier when viewing the graphical interface in the WebGUI. Similarly, copying files on and off the device is best suited to the CLI; performing the same operation using SOAP requires base64-encoding the file so that it can be embedded in the SOAP body. The Reference Guide and SOAP management schemas provide more information if you want to implement these use cases, but we’re going to focus on some more common use cases.

#### get-status

Add a note hereThe get-status operation is used for real-time retrieval of status information about an object configured on the appliance, statistics gathered by the device, or the health of the appliance itself. Sending the get-status operation with no domain limitations and no parameters retrieves every status object on the device. For a more manageable response message, define the class of status information desired, as shown in Listing 13-28. By requesting only the SystemUsage status, we get a single response with the current device load; this is equivalent to Status→System→System Usage from the WebGUI, or the show load command from the CLI.

Add a note hereListing 13-28: Requesting SystemUsage Status Using the get-status Operation

Add a note here<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Body>

<dp:request

xmlns:dp="http://www.datapower.com/schemas/management">

<dp:**get-status** class=**"SystemUsage"**/>

</dp:request>

</env:Body>

</env:Envelope>

<?xml version="1.0" encoding="UTF-8"?>

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Body>

<dp:response

xmlns:dp="http://www.datapower.com/schemas/management">

<dp:timestamp>2008-04-10T15:59:16-07:00</dp:timestamp>

<dp:status>

<SystemUsage>

<Interval>5000</Interval>

<Load>3</Load>

<WorkList>0</WorkList>

</SystemUsage>

</dp:status>

</dp:response>

</env:Body>

</env:Envelope>

Add a note hereA simple modification of the class retrieves other important device information. A full list of status reports that can be retrieved is enumerated in xml-mgmt.xsd as the StatusEnum type; some of the *most common* device-level status requests are as follows:

* Add a note here**ActiveUsers—** A list of users currently logged into the device
* Add a note here**DateTimeStatus—** The current time, according to the appliance clock
* Add a note here**Version—** The device serial number, currently running firmware version, and installed library (TAM, MQ, ODBC) versions
* Add a note here**MemoryStatus—** Returns the total, used, and free memory allocations
* Add a note here**FilesystemStatus—** The available free memory on the encrypted and temporary file systems
* Add a note here**EnvironmentalSensors—** A list of CPU temperatures and fan speeds
* Add a note here**CPUUsage—** A historical view of CPU percentages over time
* Add a note here**TCPTable—** A list of active ports and the services/listeners that are using them
* Add a note here**HTTPConnections—** A list of active HTTP connections and their current status
* Add a note here**DomainStatus—** A list of all domains, whether they have been modified, and whether their debug log or probe settings are enabled

Add a note hereIt is also helpful to have some insight into the services and objects that have been defined within a particular domain. By adding the domain attribute to the dp:request element, we can restrict the following types of status queries to a named domain:

* Add a note here**StylesheetStatus—** The stylesheets cached by each XML manager, including any compilation warnings or errors
* Add a note here**DocumentStatus—** The XML documents cached by each XML manager, including any compilation warnings or errors
* Add a note here**LoadBalancerStatus—** The current status of members in each configured load balancer object
* Add a note here**SLMSummaryStatus—** A list of Service Level Management statements and the number of matching requests received
* Add a note here**WSOperationMetrics—** Statistics gathered about services being executed in this domain

#### do-action

Add a note hereThe do-action operation is used for dynamic execution of remote commands on an appliance. These commands can act on the device itself (Shutdown), on a high-level object (RestartDomain), or on a low-level object (RefreshStylesheet). These operations can be selectively called within a particular domain (when appropriate) by adding the domain attribute. Within dp:request is a dp:action element containing the specific action being called. Additionally, the action taken may require more information about the object it is acting on or the parameters for execution. In Listing 13-29, the FlushStylesheetCache command takes a single argument, an XMLManager element containing the name of an XML Manager (LongTermCache) within the specified domain. Actions such as these can be automated for use in a build process; as stylesheets are modified, the cached version can be flushed from the XML management interface.

Add a note hereListing 13-29: Flushing a Designated Stylesheet Cache Within a Domain

Add a note here<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Body>

<dp:request

xmlns:dp="http://www.datapower.com/schemas/management"

domain=**"HRServices"**>

<dp:**do-action**>

<**FlushStylesheetCache**>

<XMLManager>**LongTermCache**</XMLManager>

</FlushStylesheetCache>

</dp:do-action>

</dp:request>

</env:Body>

</env:Envelope>

<?xml version="1.0" encoding="UTF-8"?>

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Body>

<dp:response

xmlns:dp="http://www.datapower.com/schemas/management">

<dp:timestamp>2008-04-10T17:18:48-07:00</dp:timestamp>

<dp:result>OK</dp:result>

</dp:response>

</env:Body>

</env:Envelope>

Add a note hereSuccessful do-action commands always return a dp:result node with the value OK. Failed requests either reference an authentication error, meaning something is wrong with the credentials being passed to DataPower using HTTP Basic Authentication, or an Internal Error, which likely means an error in the syntax of the request document.

Add a note hereThe syntax for each command is defined in the XML schema files we discussed earlier; these are the definitive sources for the exact formats required. All the do-action commands are enumerated in xml-mgmt.xsd as the ActionEnum type; some of the *most common* commands are as follows:

* Add a note here**FlushStylesheetCache—** Clears a domain’s stylesheet cache of all files
* Add a note here**FlushDocumentCache—** Clears a domain’s document cache of all documents
* Add a note here**RefreshStylesheet—** Recompiles a specific stylesheet (or group of stylesheets) without affecting the rest of the cache
* Add a note here**FlushAAACache—** Clears an access control policy’s cache to require upcoming requests to re-authenticate and re-authorize
* Add a note here**Shutdown—** Causes the appliance to reboot, reload, or halt
* Add a note here**SetTimeAndDate—** Sets the appliances time and date
* Add a note here**Ping—** Executes a ping command for network troubleshooting
* Add a note here**TCPConnectionTest—** Attempts to connect to a remote TCP port for network troubleshooting
* Add a note here**Keygen—** Generates a private key and self-signed certificate on the appliance
* Add a note here**SetLogLevel—** Sets the log level for a domain; useful for quickly modifying the level while troubleshooting
* Add a note here**SendLogEvent—** Creates a user-defined log event in a specified category at a given priority
* Add a note here**SaveConfig—** Saves the currently running configuration, including any changes made through the XML management interface, to the file system for persistence

#### get-config

Add a note hereThe get-config operation is used for the retrieval of current configuration parameters for a given object. These objects can be at a device level or within a domain (referenced with the domain attribute). Within dp:request is a dp:get-config element; this element can take class and name attributes to further narrow the selection to a smaller number of objects. In Listing 13-30, the request is not limited to a domain, as User Groups are a device-level object. We do, however, select by object type (UserGroup) and by the name of the object (sysadmin). Because names are unique among objects of the same type, we know that we will retrieve only one response, the full configuration of the sysadmin user group.

Add a note hereListing 13-30: Viewing a Specific User Group Configuration from a SOAP Management Call

Add a note here<soapenv:Envelope

xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">

<soapenv:Body>

<dp:request

xmlns:dp="http://www.datapower.com/schemas/management">

<dp:**get-config** class=**"UserGroup"** name=**"sysadmin"**/>

</dp:request>

</soapenv:Body>

</soapenv:Envelope

<?xml version="1.0" encoding="UTF-8"?>

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Body>

<dp:response xmlns:dp="http://www.datapower.com/schemas/management">

<dp:timestamp>2008-04-10T17:58:06-07:00</dp:timestamp>

<dp:config>

<UserGroup name="sysadmin">

<mAdminState>enabled</mAdminState>

<UserSummary>System Administrator</UserSummary>

<AccessPolicies>\*/\*/\*?Access=rwadx</AccessPolicies>

<AccessPolicies>\*/\*/file/store?Access=r</AccessPolicies>

<CommandGroup>configuration</CommandGroup>

<CommandGroup>crypto</CommandGroup>

</UserGroup>

</dp:config>

</dp:response>

</env:Body>

</env:Envelope>

Add a note hereAll the objects on which get-config can act are defined in xml-mgmt.xsd as the Config-Enum type.

#### set-config

Add a note hereBecause get-config retrieves static information (the defined configuration), it is often useful only when combined with modify-config or set-config. These two commands act similarly; each takes an object either retrieved from a get-config or built from scratch, and updates or creates the configuration on the appliance. Use set-config for a new object and use modify-config for an existing object. Multiple objects can also be combined into a single set-config request; each section of the request payload will create a new object. Let’s see how this works in a real situation. Consider the need to build out new development domains for users who are joining a project. We need a new domain, but we also need a new developer account and a user group that restricts that user to his new domain. We can create a domain, user, and user group in a single call to the appliance using the XML management interface, as shown in Listing 13-31.

Add a note hereListing 13-31: Creating a Domain, User, and User Group with a Single Set-Config Command

Add a note here<soapenv:Envelope

xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">

<soapenv:Body>

<dp:request

xmlns:dp="http://www.datapower.com/schemas/management">

<dp:**set-config**>

<**Domain** name="**JohnSmith**">

<UserSummary>Sandbox domain for John Smith</UserSummary>

<NeighborDomain class="Domain">default</NeighborDomain>

</Domain>

<**UserGroup** name="**JohnSmithGroup**">

<UserSummary>Developers in JohnSmith domain</UserSummary>

<AccessPolicies>\*/JohnSmith/\*?Access=rwadx</AccessPolicies>

</UserGroup>

<**User** name="**jsmith**">

<Password>newuser97</Password>

<GroupName>JohnSmithGroup</GroupName>

<AccessLevel>group-defined</AccessLevel>

<UserSummary>jsmith@xyco.com</UserSummary>

</User>

</dp:set-config>

</dp:request>

</soapenv:Body>

</soapenv:Envelope>

<?xml version="1.0" encoding="UTF-8"?>

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

<env:Body>

<dp:response

xmlns:dp="http://www.datapower.com/schemas/management">

<dp:timestamp>2008-04-10T18:30:57-07:00</dp:timestamp>

<dp:result>OK</dp:result>

<dp:result>OK</dp:result>

<dp:result>OK</dp:result>

</dp:response>

</env:Body>

</env:Envelope>

Add a note hereThe response from our set-config request contains three dp:result elements, one for each object that was created. This call has created a new domain called JohnSmith, a user group called JohnSmithGroup, with full permission to that domain and no others, and a user called jsmith, whose permissions are defined by the JohnSmithGroup. If we now send a do-action command with a SaveConfig action, the new configuration will be persisted.

Add a note hereWe have now used the XML management interface for retrieving status, performing device-level operations, viewing existing configurations, and creating new objects, all without ever logging into the WebGUI. The GUI is generally preferable for interactive development of complex services—you’d have to dig through a lot of XML schema definitions to define every object within a Web Service Proxy or Multi-Protocol Gateway to create them with a SOAP management call. However, for scripted or batch processing of commonly used commands or for creating multiple configurations that each require only a minor change, the SOAP management interface provides a valuable method of interaction.

Add a note hereAs the previous example demonstrated, the SOAP management interface presents an effective means for performing repetitive tasks with some degree of complexity. Such a request executed three different changes, much like the CLI alias command, but with only one call to the device. Because the interface employs an interface well-defined by a machine-readable schema, it is possible to create very powerful device configuration and management utilities that do not require access to either the CLI or the WebGUI. This means that some tasks can be completely automated by external hosts executing calls on a regular timetable.

Add a note hereSimilarly, once a set of requests for any given single device works, this same request can then be easily sent to other devices in the datacenter, thus keeping all devices synchronized with little effort. You can see that the SOAP management interface rocks!

Add a note here[[2](http://www.books24x7.com/assetviewer.aspx?bookid=30903&chunkid=928483369&noteMenuToggle=0&hitSectionMenuToggle=0&leftMenuState=1" \l "ch13fn02)]Web Services Distributed Management (WSDM) is an OASIS specification dealing with the automated monitoring and management of Web services. On DataPower, this functionality is used to interact with ITCAM for SOA, which will be discussed in later chapters.

Add a note here[[3](http://www.books24x7.com/assetviewer.aspx?bookid=30903&chunkid=928483369&noteMenuToggle=0&hitSectionMenuToggle=0&leftMenuState=1" \l "ch13fn03)]AMP is a proprietary management interface protocol used to communicate between appliances.

Add a note here[[4](http://www.books24x7.com/assetviewer.aspx?bookid=30903&chunkid=928483369&noteMenuToggle=0&hitSectionMenuToggle=0&leftMenuState=1" \l "ch13fn04)]cURL is a simple command-line application that can used to POST input messages to the appliance. It can be downloaded from [http://curl.haxx.se/.](http://curl.haxx.se/)

## Summary

Add a note hereThe DataPower Web interface gets the vast majority of attention from users, as its point-and-click configuration of complex functionality is unrivaled. This attention is well-deserved, but should not completely overshadow the power and flexibility inherent in alternate management interfaces. As the savvy network analyst, appliance administrator, and deployment engineer know, when used correctly, the CLI and XML management interface are potent instruments for efficient, stable, and repeatable configuration and administration.