



FortiManager™

Version 4.0

XML API Technical Note

FortiManager XML API Technical Note

Version 4.0

18 March 2009

02-400-93588-20090318

© Copyright 2009 Fortinet, Inc. All rights reserved. No part of this publication including text, examples, diagrams or illustrations may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical or otherwise, for any purpose, without prior written permission of Fortinet, Inc.

Trademarks

Dynamic Threat Prevention System (DTPS), APSecure, FortiASIC, FortiBIOS, FortiBridge, FortiClient, FortiGate®, FortiGate Unified Threat Management System, FortiGuard®, FortiGuard-Antispam, FortiGuard-Antivirus, FortiGuard-Intrusion, FortiGuard-Web, FortiLog, FortiAnalyzer, FortiManager, Fortinet®, FortiOS, FortiPartner, FortiProtect, FortiReporter, FortiResponse, FortiShield, FortiVoIP, and FortiWiFi are trademarks of Fortinet, Inc. in the United States and/or other countries. The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Regulatory compliance

FCC Class A Part 15 CSA/CUS



CAUTION: Risk of Explosion if Battery is replaced by an Incorrect Type.
Dispose of Used Batteries According to the Instructions.

Contents

Introduction	5
Registering your Fortinet product.....	5
Customer service and technical support.....	5
Fortinet documentation	5
Fortinet Tools and Documentation CD	6
Fortinet Knowledge Center	6
Comments on Fortinet technical documentation	6
Conventions	6
IP addresses.....	6
CLI constraints.....	6
Notes, Tips and Cautions	6
Typographical conventions.....	7
Using the FortiManager API	9
Connecting to FortiManager Web Services.....	9
Enabling Web Services.....	9
Obtaining the WSDL file	9
Getting information from the FortiManager unit.....	10
getAdomList.....	10
getDeviceList	11
getDevices	12
getConfig	13
getConfigRevisionHistory	14
getGroupList	16
Working with scripts.....	17
getScript.....	17
deleteScript.....	18
createScript.....	18
runScript	19
installConfig	20
Index.....	21

Introduction

Welcome and thank you for selecting Fortinet products for your network protection.

FortiManager System version 4.0 includes a Web Services interface to facilitate integration with provisioning systems. The FortiManager API is XML-based. This manual describes how to use the FortiManager API to obtain information from the FortiManager unit, run scripts to modify device configurations, and install the modified configurations on the managed devices.

This chapter contains the following topics:

- [Registering your Fortinet product](#)
- [Customer service and technical support](#)
- [Fortinet documentation](#)
- [Conventions](#)

Registering your Fortinet product

Before you begin, take a moment to register your Fortinet product at the Fortinet Technical Support web site, <https://support.fortinet.com>.

Many Fortinet customer services, such as firmware updates, technical support, and FortiGuard Antivirus and other FortiGuard services, require product registration.

For more information, see the Fortinet Knowledge Center article [Registration Frequently Asked Questions](#).

Customer service and technical support

Fortinet Technical Support provides services designed to make sure that your Fortinet products install quickly, configure easily, and operate reliably in your network.

To learn about the technical support services that Fortinet provides, visit the Fortinet Technical Support web site at <https://support.fortinet.com>.

You can dramatically improve the time that it takes to resolve your technical support ticket by providing your configuration file, a network diagram, and other specific information. For a list of required information, see the Fortinet Knowledge Center article [What does Fortinet Technical Support require in order to best assist the customer?](#)

Fortinet documentation

The Fortinet Technical Documentation web site, <http://docs.fortinet.com>, provides the most up-to-date versions of Fortinet publications, as well as additional technical documentation such as technical notes.

In addition to the Fortinet Technical Documentation web site, you can find Fortinet technical documentation on the Fortinet Tools and Documentation CD, and on the Fortinet Knowledge Center.

Fortinet Tools and Documentation CD

Many Fortinet publications are available on the Fortinet Tools and Documentation CD shipped with your Fortinet product. The documents on this CD are current at shipping time. For current versions of Fortinet documentation, visit the Fortinet Technical Documentation web site, <http://docs.fortinet.com>.

Fortinet Knowledge Center

The Fortinet Knowledge Center provides additional Fortinet technical documentation, such as troubleshooting and how-to-articles, examples, FAQs, technical notes, a glossary, and more. Visit the Fortinet Knowledge Center at <http://kc.fortinet.com>.

Comments on Fortinet technical documentation

Please send information about any errors or omissions in this or any Fortinet technical document to techdoc@fortinet.com.

Conventions

Fortinet technical documentation uses the conventions described below.

IP addresses

To avoid publication of public IP addresses that belong to Fortinet or any other organization, the IP addresses used in Fortinet technical documentation are fictional and follow the documentation guidelines specific to Fortinet. The addresses used are from the private IP address ranges defined in RFC 1918: Address Allocation for Private Internets, available at <http://ietf.org/rfc/rfc1918.txt?number-1918>.

CLI constraints

CLI constraints, such as `<address_ipv4>`, indicate which data types or string patterns are acceptable input for a given parameter or variable value. CLI constraint conventions are described in the CLI Reference document for each product.

Notes, Tips and Cautions

Fortinet technical documentation uses the following guidance and styles for notes, tips and cautions.



Tip: Highlights useful additional information, often tailored to your workplace activity.



Note: Also presents useful information, but usually focused on an alternative, optional method, such as a shortcut, to perform a step.



Caution: Warns you about commands or procedures that could have unexpected or undesirable results including loss of data or damage to equipment.

Typographical conventions

Fortinet documentation uses the following typographical conventions:

Table 1: Typographical conventions in Fortinet technical documentation

Convention	Example
Button, menu, text box, field, or check box label	From <i>Minimum log level</i> , select <i>Notification</i> .
CLI input	<pre>config system dns set primary <address_ipv4> end</pre>
CLI output	<pre>FGT-602803030703 # get system settings comments : (null) opmode : nat</pre>
Emphasis	HTTP connections are <i>not</i> secure and can be intercepted by a third party.
File content	<pre><HTML><HEAD><TITLE>Firewall Authentication</TITLE></HEAD> <BODY><H4>You must authenticate to use this service.</H4></pre>
Hyperlink	Visit the Fortinet Technical Support web site, https://support.fortinet.com .
Keyboard entry	Type a name for the remote VPN peer or client, such as <code>Central_Office_1</code> .
Navigation	Go to <code>VPN > IPSEC > Auto Key (IKE)</code> .
Publication	For details, see the FortiGate Administration Guide .

Using the FortiManager API

The FortiManager API enables you to configure managed FortiGate devices through a Web Services interface. You can obtain information, create and run FortiOS CLI scripts on the FortiManager database, and then install the changes on FortiGate units.

The following topics are included in this section:

- [Connecting to FortiManager Web Services](#)
- [Getting information from the FortiManager unit](#)
- [Working with scripts](#)

Connecting to FortiManager Web Services

To start working with Web Services on your FortiManager unit, you need to enable Web Services and obtain the Web Services Description Language (WSDL) file that defines the XML requests you can make and the responses that FortiManager can provide.

Enabling Web Services

You must enable Web Services on the network interfaces to which Web Services clients will connect.

To enable Web Services on an interface - web-based manager

- 1 Go to *System Settings > Network > Interface*.
- 2 Select the *Edit* icon for the interface that you want to use.
- 3 In the *Administrative Access* section, select *Web Service*.
- 4 Select *OK*.

To enable Web Services on an interface - CLI

- 1 Enter the following CLI commands:

```
config fmsystem interface
  edit <port>
    set allowaccess webservice
  end
```

where <port> is the network interface that you want to use for Web Services.

The `allowaccess` command should also include the other types of administrative access that you want to permit. For example, to allow HTTPS, SSH, and Web Services, enter the command `set allowaccess https ssh webservice`.

The FortiManager unit handles Web Services requests on port 8080.

Obtaining the WSDL file

You can download the WSDL file directly from the URL `https://<ip_address>:8080/`.

You can also use the web-based manager to download the WSDL file. Go to *System Settings > Advanced > Advanced Settings* and select the *Download WSDL file* icon.

Getting information from the FortiManager unit

To work with your managed devices, you need to obtain information from the FortiManager unit, such as:

- the list of ADOMs
- the list of managed devices
- information about individual devices
- the current configuration of devices, according to the database
- the revision history of devices

getAdomList

Use this request to get a list of the ADOMs defined on the FortiManager unit. Only an administrator with the Super_User profile can run this command.

Table 2: getAdomList request fields

<password>	Administrator password
<userID>	Administrator ID, e.g. admin

Example request:

```
<r20:getAdomList>
  <servicePass>
    <password></password>
    <userID>admin</userID>
  </servicePass>
</r20:getAdomList>
```

The response is a series of <return> tags, each containing information about an ADOM.

Table 3: getAdomList response fields

<description>	Optional description
<name>	ADOM name
<oid>	Object identifier for the ADOM

Example response:

```
<ns3:getAdomListResponse>
  <return>
    <oid>3</oid>
    <name>root</name>
    <description/>
  </return>
  <return>
    information about another ADOM
  </return>
</ns3:getAdomListResponse>
```

getDeviceList

Use this request to get summary information about the managed devices, optionally limited to a particular ADOM.

Table 4: getDeviceList request fields

<password>	Administrator password
<userID>	Administrator ID, e.g. admin
<adom>	Enter an ADOM name to list only the devices in that ADOM. If <adom> is empty or does not match an ADOM name, the response lists all devices.

Example request:

```
<r20:getDeviceList>
  <servicePass>
    <password></password>
    <userID>admin</userID>
  </servicePass>
  <adom></adom>
</r20:getDeviceList>
```

The response is a series of <return> tags, each containing information about a device.

Table 5: getDeviceList response fields

<firmware>	One of: FortiGate, FortiCarrier, FortiAnalyzer, FortiMail, or FortiSwitch.
<firmwareVersion>	Major software version number, 4, for example.
<buildNum>	Software build number, 65, for example.
<description>	Device description from FortiManager database.
<hostname>	Device host name.
<IPSContract>	FortiGuard IPS definitions version and last update time, 2.00461(2008-12-08 11:23), for example.
<antiVirusContract>	AV contract and expiry date, 8.00631(2009-02-15 14:27), for example.
<platform>	Platform name for device, FortiGate-1000A, for example.
<sn>	Serial number of device.
<ip>	IP address of device network interface from which response was received.

Example response:

```
<ns3:getDeviceListResponse>
  <return>
    <firmware>FortiGate</firmware>
    <firmwareVersion>4</firmwareVersion>
    <buildNum>68</buildNum>
    <description/>
    <hostname>FG30002801030058</hostname>
    <IPSContract>2.00542(2008-09-04 23:08)</IPSContract>
    <antiVirusContract>8.00631(2008-01-15 14:27)
      </antiVirusContract>
    <platform>FortiGate-3000</platform>
    <sn>FG30002801030058</sn>
    <ip>172.20.120.134</ip>
  </return>
  <return>
    another device
```

```

</return>
</ns3:getDeviceListResponse>

```

getDevices

Use this request to get information about specific managed devices, identified by serial number or device ID. You can obtain device ID values by using the `execute dmserver showdev` CLI command.

If you want information about the device's configuration, see [“getConfig” on page 13](#).

Table 6: getDevices request fields

<password>	Administrator password
<userID>	Administrator ID, e.g. admin
<serialNumbers>	Serial number of the device. This is the secondary identifier. You can enter multiple <serialNumbers> fields.
<devlds>	Device ID. This is the primary device identifier. You can omit this field and use <serialNumber> instead. You can enter multiple <Devlds> fields.

Example request:

```

<r20:getDevices>
  <servicePass>
    <password></password>
    <userID>admin</userID>
  </servicePass>
  <serialNumbers>FGT1002801021024</serialNumbers>
  <serialNumbers>FGT50B3G06500085</serialNumbers>
</r20:getDevices>

```

The response is a series of <return> tags, each containing information about a device.

Table 7: getDevices response fields

<firmware>	One of: FortiGate, FortiCarrier, FortiAnalyzer, FortiMail, or FortiSwitch.
<firmwareVersion>	Major software version number, 4, for example.
<buildNum>	Software build number, 65, for example.
<description>	Device description from FortiManager database.
<hostname>	Device host name.
<IPSCContract>	FortiGuard IPS definitions version and last update time, 2.00461(2008-12-08 11:23), for example.
<antiVirusContract>	AV contract and expiry date, 8.00631(2009-02-15 14:27), for example.
<platform>	Platform name for device, FortiGate-1000A, for example.
<sn>	Serial number of device.
<ip>	IP address of device network interface from which response was received.

Example response:

```

<ns3:getDevicesResponse>
  <return>
    <firmware>FortiGate</firmware>
    <firmwareVersion>3.00</firmwareVersion>
    <buildNum>650</buildNum>
    <description/>

```

```

    <hostname>Dev3</hostname>
    <IPSContract>2.442 (2007-11-08 11:23)</IPSContract>
    <antiVirusContract>8.368 (2007-11-15
      13:59)</antiVirusContract>
    <platform>FortiGate-100</platform>
    <sn>FGT1002801021024</sn>
    <ip>172.20.120.126</ip>
  </return>
</return>
  <firmware>FortiGate</firmware>
  <firmwareVersion>4</firmwareVersion>
  <buildNum>59</buildNum>
  <description/>
  <hostname>FGT50B3G06500085</hostname>
  <IPSContract/>
  <antiVirusContract/>
  <platform>Fortigate-50B</platform>
  <sn>FGT50B3G06500085</sn>
  <ip>172.20.120.127</ip>
</return>
</ns3:getDevicesResponse>

```

getConfig

Use this request to retrieve a particular revision of the device configuration from the device database.

Table 8: getConfig request fields

<password>	Administrator password
<userID>	Administrator ID, e.g. admin
<serialNumber>	Serial number of the device. This device identifier is secondary to <devId>.
<devId>	The Device ID. This is the primary device identifier. You can omit this field and use <serialNumber> instead.
<checkinUser>	The userID of the administrator who saved this revision.
<revisionNumber>	The revision that you want to view. Use a negative number to retrieve the latest revision.

Example request:

```

<r20:getConfig>
  <servicePass>
    <password></password>
    <userID>admin</userID>
  </servicePass>
  <serialNumber>FGT1002801021024</serialNumber>
  <!-- <devId></devId> -->
  <revisionNumber>3</revisionNumber>
</r20:getConfig>

```

The response is a <return> field containing the device configuration and other information.

Table 9: getConfigRevisionHistory response fields

<branchPoint>	The firmware build number, except for some special branch builds.
<checkinDate>	Date and time (UTC) when this revision was installed to the device.
<checkinUser>	The userID of the administrator who installed this revision.
<content>	The device configuration file contents.
<oid>	The devID for the device.
<osVersion>	Version of device operating system, 4, for example for FortiOS 4.0.
<platform>	Platform name for the device, FortiGate-1000A, for example.
<revisionNum>	Configuration revision ID.
<serialNumber>	The serial number of the device.

Example response

```

<ns3:getConfigResponse>
  <return>
    <branchPoint>650</branchPoint>
    <checkinDate>2009-02-02T19:37:39Z</checkinDate>
    <checkinUser>admin</checkinUser>
    <content>
      ... configuration file content ...
    </content>
    <message/>
    <oid>109</oid>
    <osVersion>3.00</osVersion>
    <platform>FortiGate-100</platform>
    <revisionNum>3</revisionNum>
    <serialNumber>FGT1002801021024</serialNumber>
  </return>
</ns3:getConfigResponse>

```

getConfigRevisionHistory

Use this request to retrieve multiple revisions of the device configuration from the device database. You can retrieve based on revision numbers or check-in times.

Table 10: getConfigRevisionHistory request fields

<password>	Administrator password
<userID>	Administrator ID, e.g. admin
<serialNumber>	Serial number of the device. This device identifier is secondary to <devId>.
<devId>	The Device ID. This is the primary device identifier. You can omit this field and use <serialNumber> instead.
<checkinUser>	Optionally, specify the userID of the administrator who saved this revision.
<minCheckinDate>	Optionally, specify the earliest revision check-in time to retrieve. Use with <maxCheckinDate>.
<maxCheckinDate>	Optionally, specify the latest revision check-in time to retrieve. Use with <minCheckinDate>.
<minRevisionNumber>	Optionally, specify the first revision to retrieve. Use with <maxRevisionNumber>.
<maxRevisionNumber>	Optionally, specify the last revision to retrieve. Use with <minRevisionNumber>.

Example request:

```
<r20:getConfigRevisionHistory>
  <servicePass>
    <password></password>
    <userID>admin</userID>
  </servicePass>
  <serialNumber>FGT1002801021024</serialNumber>
  <!-- <devId>?</devId> -->
  <!-- <checkinUser></checkinUser> -->
  <!-- <minCheckinDate></minCheckinDate> -->
  <!-- <maxCheckinDate></maxCheckinDate> -->
  <!-- <minRevisionNumber>1</minRevisionNumber> -->
  <!-- <maxRevisionNumber>?</maxRevisionNumber> -->
</r20:getConfigRevisionHistory>
```

The response is a series of <return> fields containing the device configuration and other information.

Table 11: getConfigRevisionHistory response fields

<branchPoint>	The firmware build number, except for some special branch builds.
<checkinDate>	Date and time (UTC) when this revision was installed to the device.
<checkinUser>	The userID of the administrator who installed this revision.
<content>	The device configuration file contents.
<oid>	The devID for the device.
<osVersion>	Version of device operating system, 4, for example for FortOS 4.0.
<platform>	Platform name for the device, FortiGate-1000A, for example.
<revisionNum>	Configuration revision ID.
<serialNumber>	The serial number of the device.

Example response:

```
<ns3:getConfigRevisionHistoryResponse>
  <return>
    <branchPoint>650</branchPoint>
    <checkinDate>2009-02-02T19:37:39Z</checkinDate>
    <checkinUser>admin</checkinUser>
    <content>
      ... configuration file content - newest retrieved revision ...
    </content>
    <message/>
    <oid>109</oid>
    <osVersion>3.00</osVersion>
    <platform>FortiGate-100</platform>
    <revisionNum>3</revisionNum>
    <serialNumber>FGT1002801021024</serialNumber>
  </return>
  <return>
    ... information about preceding revision ...
  </return>
```

getGroupList

Use this request to obtain a list of device groups, optionally limited to a particular ADOM.

Table 12: getGroupList request fields

<password>	Administrator password
<userID>	Administrator ID, e.g. admin
<adom>	Enter an ADOM name to list only the device groups in that ADOM. If <adom> is empty or does not match an ADOM name, the response lists all device groups.

Example request:

```
<r20:getGroupList>
  <servicePass>
    <password></password>
    <userID>admin</userID>
  </servicePass>
  <adom></adom>
</r20:getGroupList>
```

The response is a series of <return> fields, each listing one group, in ascending order of object identifier (OID). Both built-in groups, like All FortiGate, and user-defined groups are listed.

Table 13: getGroupList response fields

<name>	Group name
<oid>	Object identifier for the group

Example response:

```
<ns3:getGroupListResponse>
  <return>
    <oid>118</oid>
    <name>Group1</name>
  </return>
  <return>
    ... another group ...
  </return>
```

Working with scripts

You can upload scripts to the FortiManager unit and execute them on the FortiManager database or on a managed device. If your scripts make configuration changes to the database, you can install the changes onto the affected devices.

getScript

Use this command to retrieve a script from the FortiManager unit. This is a way to verify the contents of the script. Also, you could modify the script and use the createScript request to update the script on the FortiManager unit.

Table 14: getScript request fields

<password>	Administrator password
<userID>	Administrator ID, e.g. admin
<name>	The script name.

Example request:

```
<r20:getScript>
  <servicePass>
    <password></password>
    <userID>admin</userID>
  </servicePass>
  <name>script1</name>
</r20:getScript>
```

The response is a return tag that includes the script content and information about the script.

Table 15: getScript response fields

<content>	Script content
<description>	Administrator password
<name>	Script name
<oid>	Object ID for script

Example response

```
<ns3:getScriptResponse>
  <return>
    <content>
      ... script ...
    </content>
    <description>Generated by XML API</description>
    <name>script1</name>
    <oid>14</oid>
  </return>
</ns3:getScriptResponse>
```

deleteScript

Use this request to delete a script from the FortiManager unit.

Table 16: deleteScript request fields

<password>	Administrator password
<userID>	Administrator ID, e.g. admin
<name>	The name of the script to delete.

Example request:

```
<r20:deleteScript>
  <servicePass>
    <password></password>
    <userID>admin</userID>
  </servicePass>
  <name>xml_script1</name>
</r20:deleteScript>
```

If the script is found and deleted, the response is empty. If the script could not be found, Web Services returns an error message.

Response example: script was deleted

```
<ns3:deleteScriptResponse/>
```

Example response: script not found

```
<faultcode>SOAP-ENV:Client</faultcode>
<faultstring>script xml_script1 is not found</faultstring>
<detail>
  <error xmlns="http://localhost/">script xml_script1 is not
    found</error>
</detail>
```

createScript

Use this request to upload a script to the FortiManager unit.

Table 17: createScript request fields

<password>	Administrator password
<userID>	Administrator ID, e.g. admin
<name>	Enter the script name.
<description>	Optional brief description
<content>	The script.
<overwrite>	1 to overwrite an existing script of that name, otherwise, 0.

Example request:

```
<r20:createScript>
  <servicePass>
    <password></password>
    <userID>admin</userID>
  </servicePass>
  <name>xml_script1</name>
  <description>Generated by XML API</description>
  <content>
config firewall address
```

```

edit "33"
  set subnet 33.33.33.33 255.255.255.0
end
  </content>
  <overwrite>1</overwrite>
</r20:createScript>

```

The response is a `<return>` value of 0 if successful, 1 if not. If `<overwrite>` was 0, `createScript` can fail because there is already a script of that name on the FortiManager unit.

Example response: script created

```

<ns3:createScriptResponse>
  <return>0</return>
</ns3:createScriptResponse>

```

runScript

Use this request to run a script. You can run a script

- on the global database
- on the device database
- on the managed device

Table 18: runScript request fields

<password>	Administrator password
<userID>	Administrator ID, e.g. admin
<name>	The name of the script to run.
<devId>	Provide the Device ID when you run a script on the device or device database. You can also omit the <code><devId></code> field and use <code><serialNumber></code> to identify the unit. Set <code><devId></code> to -1 when you run the script on the global database.
<serialNumber>	Serial number of the device. This device identifier is secondary to <code><devId></code> .
<runOnDB>	1 — run on global or device database, depending on <code><devId></code> . 0 — run on the device. Specify <code><devId></code> or <code><serialNumber></code> .

Example request:

```

<r20:runScript>
  <servicePass>
    <password></password>
    <userID>admin</userID>
  </servicePass>
  <name>xml_script1</name>
  <devId>109</devId>
  <serialNumber>FGT1002801021024</serialNumber>
  <runOnDB>1</runOnDB>
</r20:runScript>

```

If the script runs successfully, the response is empty. Otherwise, Web Services returns an error message.

Example response - script ran successfully:

```

<ns3:runScriptResponse/>

```

installConfig

When you have made configuration changes on the global or device database with your scripts, use this request to install the changes to the devices.

Table 19: installConfig request fields

<password>	Administrator password
<userID>	Administrator ID, e.g. admin
<devId>	The Device ID. This is the primary device identifier. You can omit this field and use <serialNumber> instead.
<serialNumber>	Serial number of the device. This device identifier is secondary to <devId>.

Example request:

```
<r20:installConfig>
  <servicePass>
    <password></password>
    <userID>admin</userID>
  </servicePass>
  <devId>109</devId>
  <serialNumber>FGT1002801021024</serialNumber>
</r20:installConfig>
```

If the installation is successful, the response is empty. Otherwise, Web Services returns an error message.

Example response - updated configuration installed successfully:

```
<ns3:installConfigResponse/>
```

Example response - updated configuration could not be installed:

```
<faultcode>SOAP-ENV:Client</faultcode>
<faultstring>Run install+save on deviceId 109 failed</faultstring>
<detail>
  <error xmlns="http://localhost/">Run install+save on deviceId
    109 failed</error>
</detail>
```

Index

C

- comments, documentation, 6
- createScript, 18
- customer service, 5

D

- deleteScript, 18
- deleting scripts, 18
- device IDs
 - obtaining, 12
- documentation
 - commenting on, 6
 - Fortinet, 5

E

- executing scripts, 19

F

- FortiGate documentation
 - commenting on, 6
- Fortinet customer service, 5
- Fortinet documentation, 5
- Fortinet Knowledge Center, 6

G

- getAdomList, 10
- getConfig, 13
- getConfigRevisionHistory, 14
- getDeviceList, 11
- getDevices, 12
- getGroupList, 16
- getScript, 17

I

- installConfig, 20

- installing configuration changes, 20
- introduction
 - Fortinet documentation, 5

L

- listing
 - ADOMs, 10
 - device configuration, 13
 - device configuration history, 14
 - device groups, 16
 - device information, 12
 - devices, 11

R

- retrieving scripts, 17
- running scripts, 19
- runScript, 19

S

- scripts
 - deleting, 18
 - executing, 19
 - listing, 17
 - retrieving, 17
 - uploading, 18

T

- technical support, 5

U

- uploading scripts, 18

W

- Web Services
 - enabling, 9
- WSDL file
 - obtaining, 9

FORTINET®

www.fortinet.com

FORTINET®

www.fortinet.com