

COVALENT
TECHNOLOGIES

Covalent SNMP Conductor User Guide

www.covalent.net

***Version 1.0.3
February 2001***

This product contains software developed by the Apache Software Foundation
(<http://www.apache.org>)

©2001 Covalent Technologies, Inc., 706 Mission Street, Second Floor, San Francisco, CA 94103.

All rights reserved. This product and documentation are protected by copyright and distributed under licenses restricting their use, copying distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Covalent Technologies and its licensors, if any.

THIS PUBLICATION IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS PUBLICATION COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED INTO NEW EDITIONS OF THE PUBLICATION. COVALENT TECHNOLOGIES, INC., MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS PUBLICATION AT ANY TIME.

Ver. 1.0.3, rev. 2-15-2001

The Apache Software License, Version 1.1

© 2000-2001 The Apache Software Foundation. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>)." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.
4. The names "Apache" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.
5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====

This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation. For more information on the Apache Software Foundation, please see <http://www.apache.org/>.

Portions of this software are based upon public domain software originally written at the National Center for Supercomputing Applications, University of Illinois, Urbana-Champaign.

ucd-snmp license <http://ucd-snmp.ucdavis.edu/COPYING.txt>

Copyright 1989, 1991, 1992 by Carnegie Mellon University.

Derivative Work

Copyright 1996, 1998, 1999, 2000 by The Regents of the University of California.

All Rights Reserved.

Permission to use, copy, modify and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appears in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of CMU and The Regents of the University of California not be used in advertising or publicity pertaining to distribution of the software without specific written permission.

CMU and the Regents of the University of California disclaim all warranties with regard to this software, including all implied warranties of merchantability and fitness. In no event shall CMU or the Regents of the University of California be liable for any special, indirect or consequential damages or any damages whatsoever resulting from the loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the user or performance of this software.

Table of Contents

Preface	vii
About this guide	vii
How this guide is organized	vii
Document conventions	viii
 <i>Section 1</i>	
Introduction	1
About Covalent SNMP Conductor 1.0.3	1
Key features	1
SNMP-based management	3
Important files and commands	5
 <i>Section 2</i>	
Installing and testing	6
Installation requirements	6
Security	7
Downloading and unpacking	8
Installing with the Covalent Apache binary (recommended)	8
Adding to an existing Apache server	16
Starting and stopping your server	19
Testing your installation	20
 <i>Section 3</i>	
Configuration directive reference	24
Apache configuration directives	25
SNMP agent configuration directives	27
 <i>Section 4</i>	
Command reference	32
Index of commands	32
Commands	33

Section 5

MIB modules	45
Application-specific MIB modules	46
Enterprise-specific MIB modules	47
Network management framework MIB modules	50

Section 6

Troubleshooting	52
------------------------------	-----------

Appendix A:

Manual installation	55
----------------------------------	-----------

Appendix B:

Configuration examples	60
-------------------------------------	-----------

Appendix C:

Default SNMP agent configuration	62
-----------------------------------------------	-----------

Preface

About this guide

This User Guide will help you install and configure the Covalent SNMP Conductor module. You can install and configure the module with the Apache binary provided in the Covalent SNMP Conductor package or with your existing Apache configuration.

After you complete the installation instructions to get Covalent SNMP Conductor up and running, you can consult the “Configuration directive reference”, “Command reference”, and “MIB modules” sections as needed.

How this guide is organized

The *Covalent SNMP Conductor User Guide* contains the following sections:

Section 1 “Introduction” provides an overview of the Covalent SNMP Conductor module and explains features and relevant terminology.

Section 2 “Installing and testing” provides detailed instructions for installing Covalent SNMP Conductor with or without the Apache binary included in the package.

Section 3 “Configuration directive reference” details SNMP configuration file directives.

Section 4 “Command reference” is based on UCD-SNMP man pages and describes the tools included in the Covalent SNMP Conductor package.

Section 5 “MIB modules” provides information about the MIB modules implemented in Covalent SNMP Conductor, and how you can use the modules to manage your Apache Web server.

Section 6 “Troubleshooting” is a reference for troubleshooting during installation and implementation of Covalent SNMP Conductor.

“Appendix A: Manual installation” explains how to perform a quick manual installation, configure the Apache and SNMP configuration files, and test your installation.

“*Appendix B: Configuration examples*” contains examples for using the Apache and Covalent SNMP Conductor configuration files.

“*Appendix C: Default SNMP agent configuration*” contains details about the default SNMP agent configuration.

Document conventions

This guide uses the following conventions:

<i>Convention</i>	<i>Use</i>
Bold	Covalent SNMP Conductor tools, options, and buttons; references to Web sites. <i>For example</i> , the Installation Program .
"Quotes"	Cross references to other sections of this guide. <i>For example</i> , refer to “Troubleshooting” on page 52.
Courier	File names, path names, file contents, and commands you execute. <i>For example</i> , <code>./setup</code> .
Courier Italic	Variables that you should replace with your system’s information. <i>For example</i> , in <code>http://yourserver.com</code> replace <code>yourserver.com</code> with your server’s (domain) name.



Section 1

Introduction

About Covalent SNMP Conductor 1.0.3

The SNMP (Simple Network Management Protocol) management framework is commonly used by Internet service providers (ISPs) and large organizations. SNMP is designed to manage complex network infrastructures and to help organizations ensure a specific quality of service.

The Covalent SNMP Conductor module is a plug-in for the Apache HTTP (Web) server. You can install and configure Covalent SNMP Conductor with the Apache binary provided in the Covalent SNMP Conductor package or with your existing Apache configuration.

Key features

- **Allows you to leverage your existing network management infrastructures.** For example, Covalent SNMP Conductor actively and automatically notifies your SNMP-based Network Operation Center of any failures which would jeopardize service. An SNMP-based management application can initiate regular polling of load values, enabling you to detect load anomalies early.
- **Provides support for the most advanced SNMP standard: SNMP version 3 (SNMPv3).** SNMPv3 is essential in ISP and e-commerce environments. It is the only SNMP version that allows for view-based and user-based access control, secure authentication, and encrypted communication. Please refer to “SNMP security features” on page 3 for more information.
- **Helps you counter a Denial of Service attack** in two ways: by providing a timely warning well in advance; and by providing you with the real-time configuration facilities needed to defuse the overload. SNMP is essential for those who want to provide quality Web service. Please refer to “SNMP-based management” on page 3 for more information.

- **Implements the open standard WWW-MIB and an the NETWORK-SERVICES-MIB.** The WWW-MIB module provides service information about HTTP server(s) in the system and HTTP protocol information, as well as more general information about document access, script errors, and Web service utilization. The NETWORK-SERVICES-MIB module provides administrative and connection information for networked applications.
- **Implements enterprise-specific MIB modules for the Apache Web server.** The Apache-specific MIB module allows real-time access to the most important Web server settings for live configuration changes.
- **Interoperates fully with all standards-compliant SNMP-based management applications.** MIB module definitions are in a standardized format that you can import into the management application.
- **Operates as a multi-protocol SNMP agent that supports the full range of SNMP protocols:** SNMPv1, SNMPv2c, and SNMPv3. Please refer to “SNMP security features” on page 3 for more information.
- **Is available on a wide range of UNIX and Linux platforms.**
- **Is compatible with other Apache modules** such as the Raven SSL module—essential to secure e-commerce.
- **Is standards-compliant** and adheres to the SNMP standards set forth by the IETF (Internet Engineering Task Force).

SNMP-based management

The **Simple Network Management Protocol (SNMP)** is a well-known Internet management framework and is an open industry standard. SNMP is widely deployed and available in almost every network device. The SNMP communication model is an asymmetric protocol, which an SNMP manager uses to retrieve management information from multiple agents.

An **SNMP agent** is simply a software process that responds to queries using SNMP to provide information about a network device.

The **Management Information Base (MIB)** is a base of managed objects accessed by network management protocols. A MIB is a set of parameters which an **SNMP management application** can query, or set in the SNMP agent, of a network device. Network devices include hardware (such as routers, bridges, and modems) and software (such as operating systems, network layers, and applications).

The management application queries the SNMP agent for the values of objects managed by the SNMP agent. The SNMP agent then provides the management application with a coherent view of the health of the network and its Web services.

SNMP security features

Initial versions of the SNMP framework are known to have simple security mechanisms.

SNMPv1 provides basic access to managed objects and has community-based security.

SNMPv2c includes additional data types and uses the network more efficiently by providing various bulk transports.

SNMPv3 extends the SNMP framework by addressing two areas: administration and security.

The SNMPv3 framework supports a modular architecture into which other security protocols can fit. This allows new security protocols to be defined without changing the protocol as new security technology becomes available.

Currently, administration and security are defined by the User-based Security model and the View-based Access Control model:

- **The User-based Security model (RFC 2574)** defines elements of procedures for providing SNMP message level security. The model provides security against threats of network management problems such as message notification, stream modification, masquerading, and disclosure.
- **The View-based Access Control model (VACM) (RFC 2575)** defines elements of procedures for accessing management information. The model restricts operations based on MIB views. (A MIB view provides partial access to the MIB.)

Important files and commands

Directory layout and important commands are as follows:

Covalent Apache configuration

```
apache/  
  bin/  
    httpsdctl  
  conf/  
    httpsd.conf  
  logs/  
    httpsd_error_log
```

Standard Apache configuration

```
apache/  
  bin/  
    apachectl  
  conf/  
    httpd.conf  
  logs/  
    httpd_error_log
```

After installation

```
raven/  
  module/  
    conductor1.0.3/  
      bin/  
        snmpwalk  
      conf/  
        snmpd.conf
```



Section 2

Installing and testing

Index to installing and testing

Installation requirements	6
Security	7
Downloading and unpacking	8
Installing with the Covalent Apache binary (recommended)	8
Adding to an existing Apache server	16
Testing your installation	20

Installation requirements

To install the Covalent SNMP Conductor 1.0.3 module, you need:

- 1 The Apache HTTP (Web) server.

Use either the Apache binary distribution included with Covalent SNMP Conductor,

or

Use Covalent SNMP Conductor with your existing Apache installation *if* your installation is able to link modules as dynamic shared objects (DSOs). See "About existing Apache installations" below.

- 2 The X Window System for the graphical user interfaces.

NOTE: If you do not have the X Window System, use the text-based user interface.

- 3 The gzip decompression utility.

About existing Apache installations

To verify whether your current installation supports DSOs, execute the following for a standard (non-Covalent) Apache installation:

```
apache/bin/httpd -l
```

Confirm that the output includes `mod_so.c`. If this module is not included in your current Apache installation, your server does not support DSOs. You must recompile the Apache server with the `--enable-module=so` flag added to the `./configure` command.

NOTE: If you have an Apache installation which is EAPI enabled and older than Apache 1.3.9, you may encounter problems linking our module to your existing Apache installation. You need to upgrade prior to installing Covalent SNMP Conductor.

Security

Prior to installing the Covalent SNMP Conductor module, check the documentation of your management application to verify which SNMP versions are supported.

Covalent SNMP Conductor supports SNMP versions 1 (SNMPv1), 2c (SNMPv2c), and 3 (SNMPv3).

SNMPv1 and SNMPv2c send information plaintext over the wire. If you install SNMPv1 and SNMPv2c, you should be aware that the community string protecting access to your information is set to 'public'.

If you want to limit access to your SNMP module, you should change the community string in the `snmpd.conf` file in the `raven/module/conductor1.0/conf` directory. This provides limited protection as communication is not encrypted in SNMPv1 and SNMPv2c.

Unless your Apache Web server sits outside your firewall, we recommend closing the default SNMP ports (161 and 162) on the firewall to prevent unauthorized access to your SNMP agents.

Downloading and unpacking

- 1 If you do not yet have the Covalent SNMP Conductor 1.0.3 distribution, purchase a full release at:
<http://www.covalent.net/products/snmp>
- 2 Download the distribution using the instructions you received when you completed your purchase.
- 3 To unpack the distribution, execute:

```
gunzip -c Conductor-1.*-1.3.*-platform.tar.gz | tar xf -
```

NOTE: Spaces and capitalization are significant.

After you unpack the distribution, the distribution files reside in the `Conductor-1.*` directory.

Installing with the Covalent Apache binary (recommended)

NOTE: Please refer to “Security” on page 7 prior to installing the Covalent SNMP Conductor module.

The Covalent SNMP Conductor package includes a complete, up-to-date Apache binary that contains all standard modules and support for dynamically loaded modules. Installing this Apache server is the easiest way to get Covalent SNMP Conductor up and running.

To install Covalent SNMP Conductor with the Covalent Apache binary:

- 1 Start the **Installation Program**.

If you are logged in as root, execute the following:

<i>For graphical mode</i>	<i>For text mode</i>
<code>cd Conductor-1.* ./setup</code>	<code>cd Conductor-1.* ./setup --textmode</code>

If you **are not** logged in as root, execute the following to log in as root and start the installation program:

<i>For graphical mode</i>	<i>For text mode</i>
<pre>xhost +servername su root cd Conductor-1.* ./setup</pre>	<pre>su root cd Conductor-1.* ./setup --textmode</pre>

NOTE: If you do not have root permissions, refer to “Troubleshooting” on page 52.

- 2 After you start the **Installation Program**, the **Covalent SNMP Conductor Welcome** screen displays.

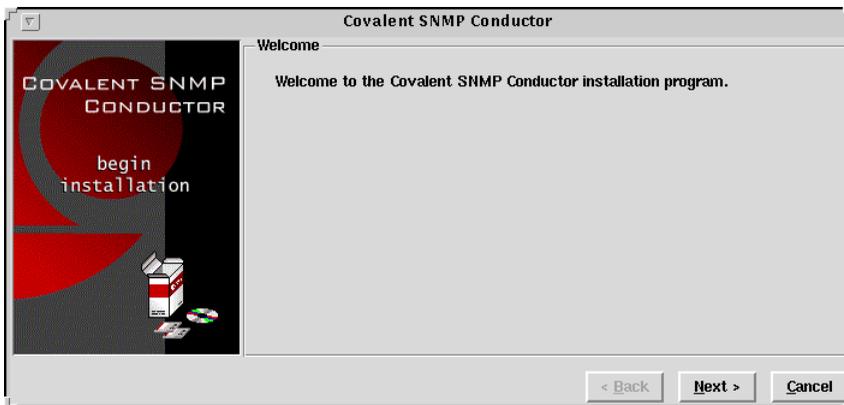


Figure 2-1 The graphical **Covalent SNMP Conductor Welcome** screen

Select the **Next** button to continue.

- 3 The **Installation Program** installs Covalent SNMP Conductor and Apache in the directories you choose during the interactive installation process. If you use the default settings, Covalent SNMP Conductor will be installed in `/usr/local/raven/module/conductor1.0` and Apache in `/usr/local/apache`.

To accept the default directory, select **Next**.

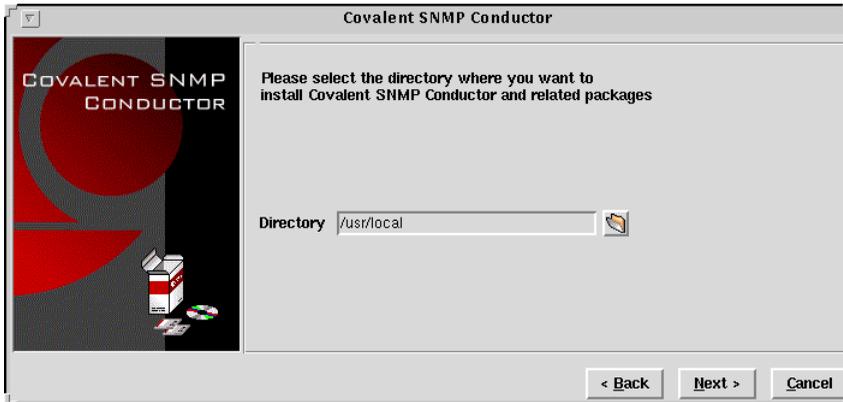


Figure 2-2 Defining the top level directory for Covalent SNMP Conductor

To define a different directory (new or existing), type the directory path in the **Directory** prompt box, or select the **file folder icon** and browse for the directory. When the correct directory displays in the prompt box, select **Next** to continue.

- 4 If you want to install the Apache binary included with Covalent SNMP Conductor, select **Next**.

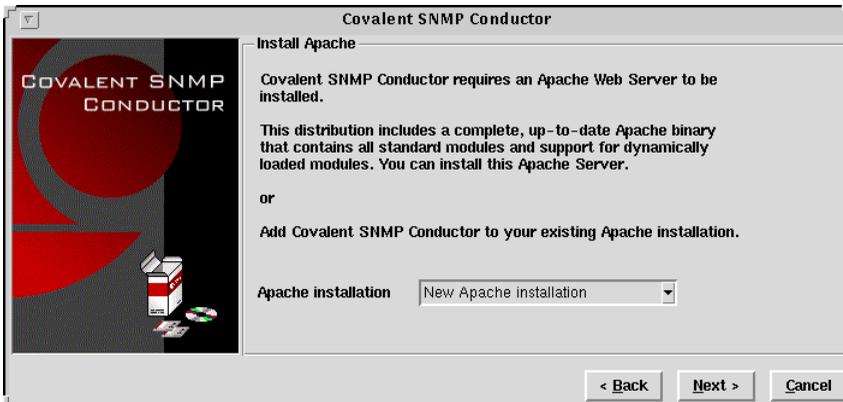


Figure 2-3 Selecting the appropriate Apache installation

If you want to link Covalent SNMP Conductor dynamically to your existing Apache installation, refer to “Adding to an existing Apache server” on page 16.

5 Define the directory where you want to install Apache.

The **Directory** prompt box defaults to the `apache` directory below the top level directory you defined in step 3.

For example, if you defined `/usr/local` as the top level directory, the prompt box defaults to `/usr/local/apache` as Figure 2-4 illustrates.

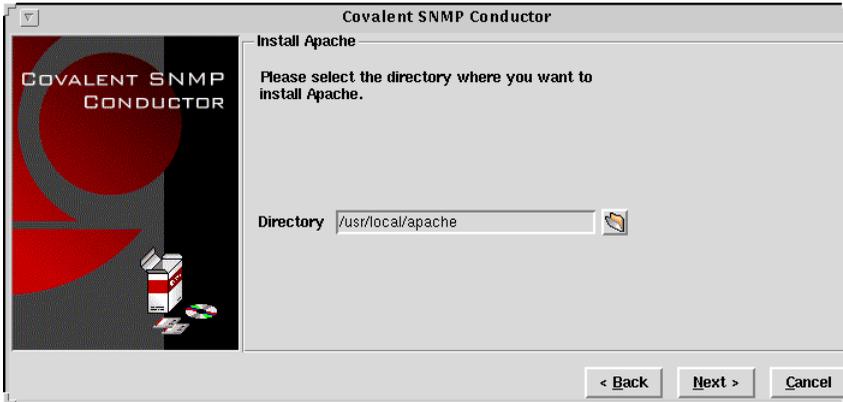


Figure 2-4 The Apache directory prompt box

To accept the default answer and continue, select **Next**.

6 If the **Installation Program** finds an existing Apache installation in the directory you define, a warning displays. If you do not receive this warning, proceed to the next step.

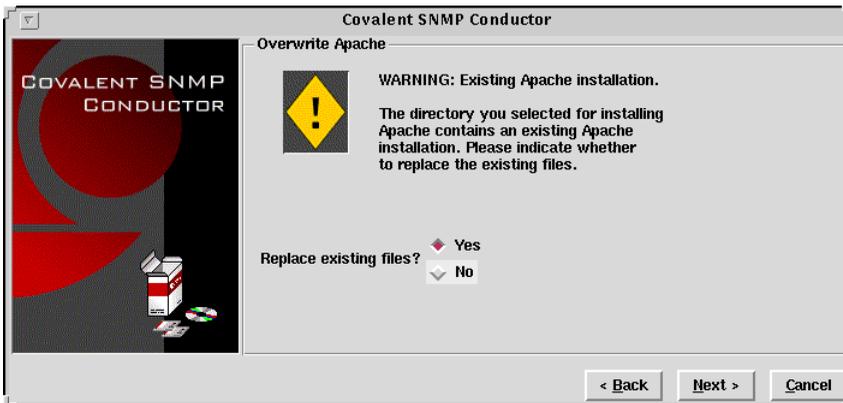


Figure 2-5 The overwrite Apache warning

To overwrite the existing Apache installation and install the Apache binary, select the **Yes** option. When you select **Yes**, the corresponding bullet becomes red. Select **Next** to continue.

7 Define the directory where you want to install Covalent SNMP Conductor.

The **Directory** prompt box defaults to the `raven` directory below the top level directory you defined in step 3.

For example, if you defined `/usr/local` as the top level directory, the prompt box defaults to `/usr/local/raven` as Figure 2-6 illustrates.



Figure 2-6 The Covalent SNMP Conductor directory prompt box

To accept the default answer and continue, select **Next**.

8 Select the SNMP version you want to install.

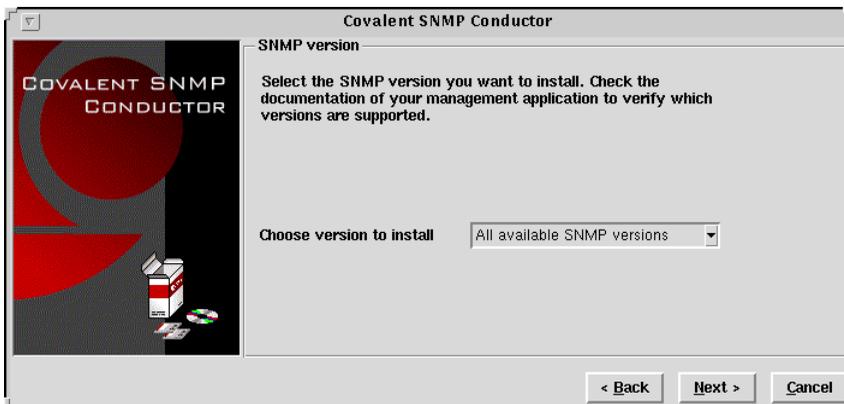


Figure 2-7 The SNMP version selection box

Check the documentation of your management application to verify which versions are supported. Covalent SNMP Conductor supports SNMP versions 1, 2c, and 3.

Please refer to “Security” on page 7 to verify the security implications of your selection.

To install all available versions of SNMP, as illustrated in Figure 2-7, select **Next**.

- 9 If you selected to install all available versions of SNMP or SNMPv1/v2c in step 8, you need to enter your network address and subnetmask to enable remote access to your Covalent SNMP Conductor module.

The network address is an IP number representing the network in which you are operating. The subnetmask indicates which part of the IP address denotes the network and subnet, and which part denotes the host ID.

For example, the network address 10.0.0.0 combined with the subnetmask 255.255.255.0 indicates that the network has IP addresses ranging from 10.0.0.1 to 1.0.0.255. The subnetmask should be entered in the installer in dotted numbers. For example, 255.255.255.0.

NOTE: If you don't enter your network address and subnet mask, your SNMP agent will only be accessible from the localhost.

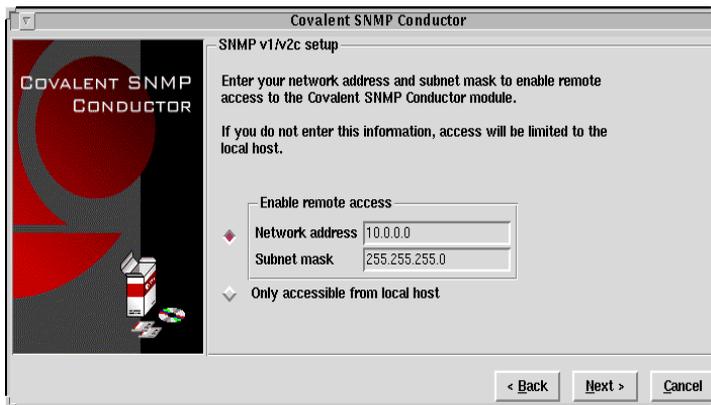


Figure 2-8 Enabling remote access

Enter your **Network address** and **Subnet mask**, then select **Next** to continue.

- 10 If you selected SNMPv3 in step 8, you must establish an SNMPv3 password. The password must be at least eight characters long.

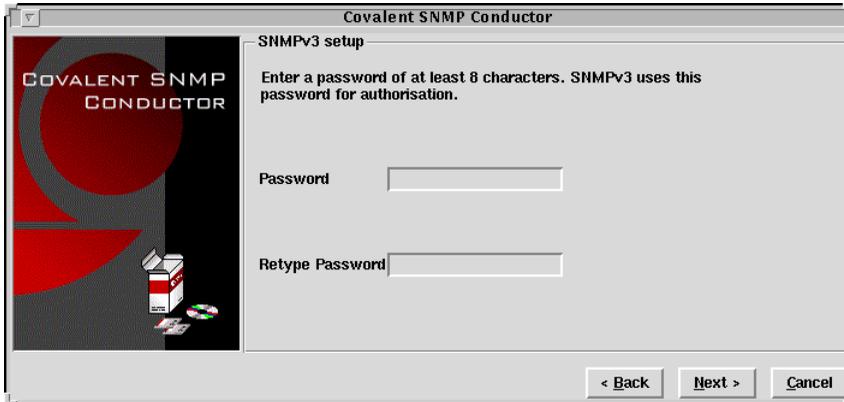


Figure 2-9 SNMPv3 password setup screen

Enter a password, retype the password, then select **Next** to continue.

- 11 The **Ready to install** screen displays.



Figure 2-10 The **Ready to install** screen

If the information is correct, select **Next** to continue.

If the information is incorrect, select **Back** until you can correct your installation parameters.

- 12 After you select **Next** on the **Ready to install** screen, progress bars display to indicate the progress of the file copy routines.

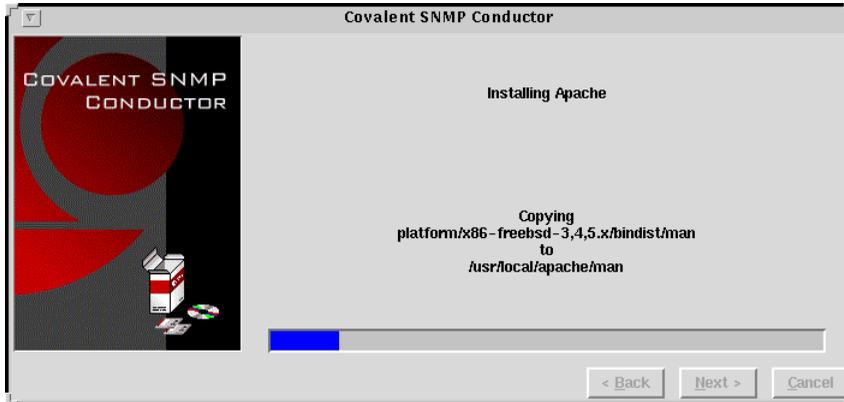


Figure 2-11 The installation in progress screen

If you are not logged in as root, warnings with specific instructions may display after the last progress bar. Retain a copy of the instructions and refer to “Troubleshooting” on page 52 for important information.

- 13 When the **Installation Program** finishes copying files, the **Congratulations!** screen displays indicating that you have successfully installed Covalent SNMP Conductor.

Select the **Finish** button.

Adding to an existing Apache server

You can link Covalent SNMP Conductor dynamically to your existing Apache installation as a dynamic shared object (DSO).

To determine whether your existing Apache server supports DSOs, you can run the `apache/bin/httpd -l` command. The result of this command should include `mod_so.c`. If this module is not included in your current Apache installation, you must recompile the Apache server with the `--enable-module=so` flag added to the `./configure` command.

To add Covalent SNMP Conductor to an existing Apache server:

Run `./setup` according to the installation procedures in the previous section, with the following exceptions:

- 1 When the **Install Apache** screen displays, you can choose to add Covalent SNMP Conductor to your existing Apache installation.

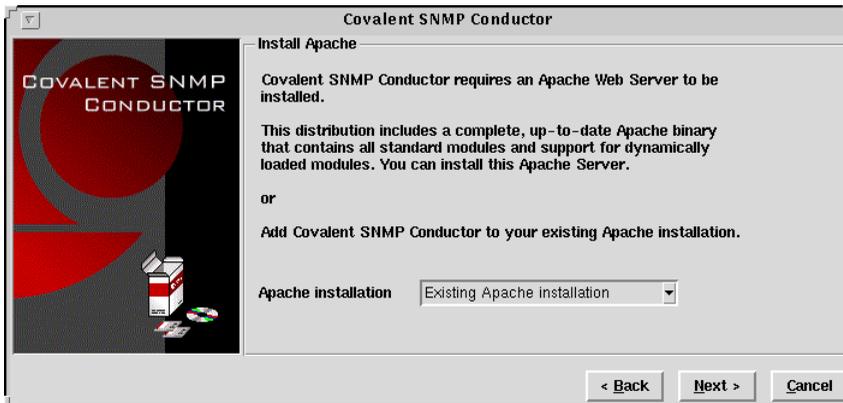


Figure 2-12 Selecting the appropriate Apache installation

Select **Existing Apache installation** from the Apache installation drop-down box, then select **Next** to continue.

- 2 When the screen illustrated in Figure 2-12 displays, establish an appropriate path to the Apache configuration file.

If you are adding Covalent SNMP Conductor to an existing Apache installation, you must enter the path to the Apache configuration file. The default path to this file is:

```
/usr/local/apache/conf/httpd.conf
```

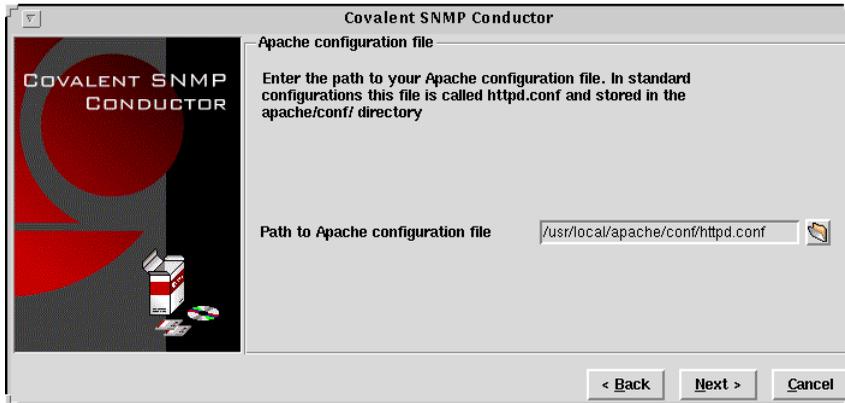


Figure 2-13 Establishing a path to the Apache configuration file

After you enter the path, select **Next** to continue.

- 3 When the screen illustrated in Figure 2-13 is displayed, enter the appropriate commands to start and stop your server.

Use the following commands:

For standard Apache configuration:

```
/path/to/apache/bin/  
apachectl start/stop
```

For Covalent Apache configuration:

```
/path/to/apache/bin/  
httpsdctl start/stop
```

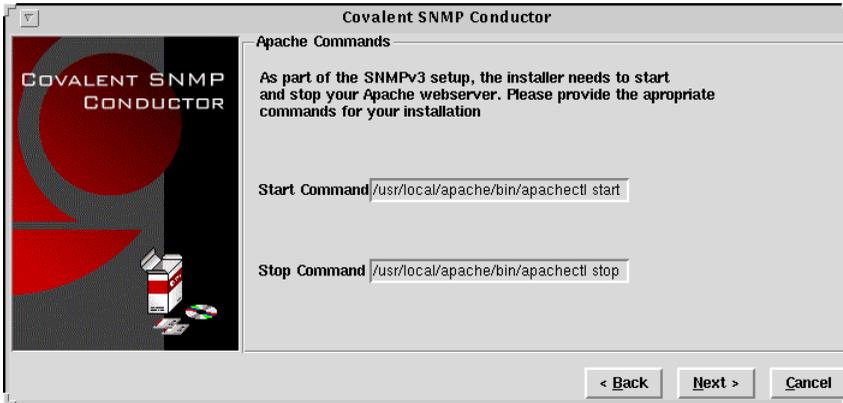


Figure 2-14 Establishing commands to start and stop your server

After you enter the appropriate commands, select **Next** to continue.

Starting and stopping your server

Starting your server

If you installed the Apache binary included with Covalent SNMP Conductor, to start your Apache Web server execute:

```
apache/bin/httpsdctl start
```

The output should be:

```
httpsdctl start: httpd started
```

NOTE: Refer to “Troubleshooting” on page 52 if you have problems starting your server.

Stopping your server

To stop your Apache Web server execute:

```
apache/bin/httpsdctl stop
```

Testing your installation

If the installation was successful, you are now ready to test the functionality of your Covalent SNMP Conductor.

Use this table to compare the standard Apache configuration with the Covalent Apache configuration:

For standard Apache configuration	For Covalent Apache configuration	Directory
<code>apachectl</code>	<code>httpsdctl</code>	<code>apache/bin/</code>
<code>httpd.conf</code>	<code>httpsd.conf</code>	<code>apache/conf/</code>
<code>httpd_error_log</code>	<code>httpsd_error_log</code>	<code>apache/logs/</code>

Testing if you installed as a root user

The following steps only apply if you installed Covalent SNMP Conductor as a root user. If you did *not* install as a root user, skip these steps and proceed to “Testing if you installed as a non-root user” on page 22.

- 1 Start your Apache Web server.
- 2 To test SNMPv1 enter the following, where *public* is your community string:

```
raven/module/conductor1.0/bin/snmpwalk -v 1 localhost public
```

The expected output for this command looks like this:

```
system.sysDescr.0 = FreeBSD localhost 4.1-RELEASE FreeBSD 4.1- Fri Jul
i386
system.sysObjectID.0 = OID:
enterprises.covalent.covalentGenericMIB.ctGenericOSoid.255
system.sysUpTime.0 = Timeticks: (2340) 0:00:23.40
system.sysContact.0 = Covalent SNMP Conductor <webmaster@localhost.com>
system.sysName.0 = localhost
system.sysLocation.0 = Apache Server with Covalent SNMP Conductor
...
```

- 3 To test SNMPv2c enter the following, where *public* is your community string:

```
raven/module/conductor1.0/bin/snmpwalk -v 2c localhost public
```

The expected output for this command is the same for testing SNMPv1.

- 4 To test SNMPv3 enter the following, where *password* is the password you established while installing Covalent SNMP Conductor:

```
raven/module/conductor1.0/bin/snmpwalk -v 3 -l authNoPriv -u  
covalent -a MD5 -A password localhost
```

NOTE: The default user after installing Covalent SNMP Conductor is called *covalent*.

The expected output for this command is the same for testing SNMPv1.

Testing if you installed as a non-root user

The following steps only apply if you did *not* install Covalent SNMP Conductor as a root user.

When you install Covalent SNMP Conductor as a non-root user, the module uses port 1610 for the SNMP agent. Therefore, when testing your installation you must include the `-p 1610` option after each command, as indicated in the following steps.

- 1 Start your Apache Web server.
- 2 To test SNMPv1 enter the following, where *public* is your community string:

```
raven/module/conductor1.0/bin/snmpwalk -p 1610 -v 1 localhost
public
```

The expected output for this command looks like this:

```
system.sysDescr.0 = FreeBSD localhost 4.1-RELEASE FreeBSD 4.1- Fri Jul
i386
system.sysObjectID.0 = OID:
enterprises.covalent.covalentGenericMIB.ctGenericOSoid.255
system.sysUpTime.0 = Timeticks: (2340) 0:00:23.40
system.sysContact.0 = Covalent SNMP Conductor <webmaster@localhost.com>
system.sysName.0 = localhost
system.sysLocation.0 = Apache Server with Covalent SNMP Conductor
...
```

- 3 To test SNMPv2c enter the following, where *public* is your community string:

```
raven/module/conductor1.0/bin/snmpwalk -p 1610 -v 2c localhost
public
```

The expected output for this command is the same for testing SNMPv1.

- 4 To test SNMPv3 enter the following, where *password* is the password you established while installing Covalent SNMP Conductor:

```
raven/module/conductor1.0/bin/snmpwalk -p 1610 -v 3 -l  
authNoPriv -u covalent -a MD5 -A password localhost
```

NOTE: The default user after installing Covalent SNMP Conductor is called *covalent*.

The expected output for this command is the same for testing SNMPv1.



Section 3

Configuration directive reference

This directive reference gives you specific information about two types of directives for the Covalent SNMP Conductor module:

- The “Apache configuration directives” section lists the directives that you use to configure your Apache Web server.
- The “SNMP agent configuration directives” section lists the SNMP agent configuration directives that you use to configure the Covalent SNMP Conductor configuration file.

Index of directives

Apache configuration directives

SNMPconf25
SNMPvar26

SNMP agent configuration directives

syslocation27
syscontact27
agentaddress28
rocommunity and rwcommunity28
rouser and rwuser29
com2sec29
group29
access30
view30
createUser31

Apache configuration directives

Covalent SNMP Conductor extends the Apache Web server with an SNMP agent. Therefore, two Apache configuration directives are available to provide the SNMP agent with configuration information: `SNMPconf` and `SNMPvar`.

SNMPconf

Name:	SNMPconf
Description:	Pointer to the directory where the SNMP agent configuration file resides
Syntax:	<code>SNMPconf</code> <i>directory</i>
Default:	<code>conf</code>
Context:	Global
Override:	Not applicable
Status:	Extension
Compatibility:	Covalent SNMP Conductor 1.0.3

This directive specifies the directory from which the SNMP agent will read its configuration file. This configuration file is used only by the SNMP agent.

SNMPvar

Name:	SNMPvar
Description:	Pointer to the directory where the SNMP agent stores its persistent information
Syntax:	SNMPvar <i>directory</i>
Default:	var
Context:	Global
Override:	Not applicable
Status:	Extension
Compatibility:	Covalent SNMP Conductor 1.0.3

This directive specifies the directory where the SNMP agent will store its persistent information. This persistent information is used to maintain SNMPv3 user information during a restart of the SNMP agent or between a stop and start of the SNMP agent. Under normal circumstances, you should not edit files in this directive.

SNMP agent configuration directives

The Covalent SNMP configuration file (`snmpd.conf`) defines how the Covalent SNMP Conductor agent operates and is stored in the `path/to/raven/module/conductor1.0/conf` directory.

NOTE: The Covalent `snmpd.conf` file and the information in this section are based on the UCD-SNMP package. Please refer to the license section in front of this manual for more information.

Covalent SNMP Conductor supports the View-based Access Control Model (VACM) as defined in RFC 2575, "View-based Access Control Model for the Simple Network Management Protocol (SNMP)", <http://www.ietf.org/rfc/rfc2575.txt>. To this end, it recognizes the keywords and directives described in the remainder of this section.

syslocation

Syntax:

```
syscontact STRING
```

Default:

```
syslocation Apache Server with Covalent SNMP Conductor
```

The `syslocation` directive sets the system location for the SNMP agent. This is the human-readable string that defines where the host physically resides. This information is reported by the `system` group in the mibII tree.

syscontact

Syntax:

```
syscontact STRING
```

Default:

```
syscontact Covalent SNMP Conductor <webmaster@localhost.com>
```

The `syscontact` directive sets the system contact for the SNMP agent. This is a human-readable string that specifies who is responsible for the system. This information is reported by the `system` group in the mibII tree.

agentaddress

The `agentaddress` directive places the agent list on the specified list of sockets instead of the default port (port 161). Multiple ports can be separated by commas. Transports can be specified by prepending the port number with the transport name (`udp` or `tcp`) followed by a colon. Finally, to bind to a particular interface, specify the address you want it to bind with.

For example:

Specifying `agentaddress 161,tcp:161,9161@localhost` will make the agent listen on `udp` port 161 for any address, `tcp` port 161 for any address, and `udp` port 9161 on only the interface associated with the `localhost` address.

NOTE: The `-T` flag changes the default transport mapping. In the example above, the default transport mapping is `udp`.

rocommunity and rwcommunity

Syntax:

```
rocommunity COMMUNITY [SOURCE] [OID]
rwcommunity COMMUNITY [SOURCE] [OID]
```

These create read-only and read-write communities that can be used to access the SNMP agent. They are a quick method of using the following `com2sec`, `group`, `access`, and `view` directive lines. They are not as efficient either, as groups aren't created so the tables are possibly larger. In other words: don't use these if you have complex situations to set up.

The format of the `SOURCE` token is described in the `com2sec` directive section below. The `OID` token restricts access for that community to everything below that given `OID`.

rouser and rwuser

Syntax:

```
rouser USER [noauth|auth|priv] [OID]
rwuser USER [noauth|auth|priv] [OID]
```

These directives create an SNMPv3 USM user in the VACM access configuration tables. Again, it is more efficient (and powerful) to use the combined `com2sec`, `group`, `access`, and `view` directives instead.

The minimum level of authentication and privacy the user must use is specified by the first token (which defaults to "auth"). The OID parameter restricts access for that user to everything below the given OID.

com2sec

Syntax:

```
com2sec NAME SOURCE COMMUNITY
```

This directive specifies the mapping from a source/community pair to a security name. `SOURCE` can be a hostname, a subnet, or the word "default". A subnet can be specified as `IP/MASK` or `IP/BITS`. The first source/community combination that matches the incoming packet is selected.

group

Syntax:

```
group NAME MODEL SECURITY
```

This directive defines the mapping from security.

access

Syntax:

```
NAME CONTEXT MODEL LEVEL PREFIX READ WRITE NOTIFY
```

The `access` directive maps from group/security model/security level to a view. `MODEL` is one of any, v1, v2c, or usm. `LEVEL` is one of noauth, auth, or priv. `PREFIX` specifies how `CONTEXT` should be matched against the context of the incoming pdu, either exact or prefix. `READ`, `WRITE` and `NOTIFY` specifies the view to be used for the corresponding access. For v1 or v2c access, `LEVEL` will be noauth, and `CONTEXT` will be empty.

view

Syntax:

```
NAME TYPE SUBTREE [MASK]
```

The `view` directive defines the named view. `TYPE` is either included or excluded. `MASK` is a list of hex octets, separated by '.' or ':'. The `MASK` defaults to ff if not specified.

The reason for the mask is that it allows you to control access to one row in a table, in a relatively simple way. As an example, as an ISP you might consider giving each customer access to his or her own interface:

```
view cust1 included interfaces.ifTable.ifEntry.ifIndex.1 ff.a0
view cust2 included interfaces.ifTable.ifEntry.ifIndex.2 ff.a0
```

(`interfaces.ifTable.ifEntry.ifIndex.1 == .1.3.6.1.2.1.2.2.1.1.1`, `ff.a0 == 11111111.10100000`. which nicely covers up and including the row index, but lets the user vary the field of the row)

createUser

Syntax:

```
createUser username (MD5|SHA) authpassphrase [DES] [priv-  
passphrase]
```

Default:

```
createUser covalent MD5 password DES
```

NOTE: The default username is `covalent`.

The reason is that the information is read from the file and then the line is removed (eliminating the storage of the master password for that user) and replaced with the key that is derived from it. This key is a localized key, so that if it is stolen it can not be used to access other agents. If the password is stolen, however, it can be.

MD5 and SHA are the authentication types to use, but you must have built the package with `openssl` installed in order to use SHA. The only privacy protocol currently supported is `DES`. If the privacy passphrase is not specified, it is assumed to be the same as the authentication passphrase. Note that the users created will be useless unless they are also added to the VACM access control tables described above.

WARNING: The minimum password length is eight characters.

SNMPv3 users can be created at runtime using the `snmpusm` command.



Section 4

Command reference

This command reference is based on the UCD-SNMP man pages. It gives descriptions of the UCD-SNMP tools included in the Covalent SNMP Conductor package. Please refer to the license section at the front of this manual.

NOTE: If you installed Covalent SNMP Conductor as a non-root user, the module uses port 1610. Therefore, when testing your installation you must include the `-p 1610` option after each command. For example, the command `snmpgetnext localhost public sysUpTime` would be `snmpgetnext -p 1610 localhost public sysUpTime` for non-root users.

Index of commands

<code>snmpwalk</code>	33
<code>snmpbulkwalk</code>	35
<code>snmptrapd</code>	37
<code>snmpgetnext</code>	39
<code>snmpusm</code>	40
<code>snmpset</code>	43
<code>snmpget</code>	44

Commands

snmpwalk

Name

`snmpwalk` — communicates with a network entity using SNMP GET Next Requests.

Syntax

```
snmpwalk [ common arguments ] [ objectID ]
```

Description

`snmpwalk` is an SNMP application that uses GET NEXT Requests to query for a tree of information about a network entity. The command line can include a variable to specify which portion of the object identifier space is searched.

All variables in the subtree below the given variable are queried and their values presented to the user. Each variable name is given in the format specified in `variables(5)`.

If the "objectID" argument is not present, `snmpwalk` will search MIB-2.

Example

To retrieve all the variables under `system`, use the command:

```
snmpwalk localhost public system
```

The expected output for this command looks like this:

```
system.sysDescr.0 = FreeBSD localhost 4.1-RELEASE FreeBSD 4.1- Fri Jul
i386
system.sysObjectID.0 = OID:
enterprises.covalent.covalentGenericMIB.ctGenericOSoid.255
system.sysUpTime.0 = Timeticks: (2340) 0:00:23.40
system.sysContact.0 = Covalent SNMP Conductor <webmaster@localhost.com>
system.sysName.0 = localhost
system.sysLocation.0 = Apache Server with Covalent SNMP Conductor
...
```

If the network entity has an error when processing the request packet, an error packet is returned and a message displayed to help identify the reason for the malformed request.

If the tree search causes attempts to search beyond the end of the MIB, the message is displayed: `End of MIB.`

snmpbulkwalk

Name

`snmpbulkwalk` — communicates with a network entity using SNMP BULK Requests.

Syntax

```
snmpbulkwalk [ common arguments ] [ objectID ]
```

Description

`snmpbulkwalk` is an SNMP application that uses BULK Requests to query for a tree of information about a network entity. The command line can include a variable to specify which portion of the object identifier space is searched.

All variables in the subtree below the given variable are queried as a single request and their values presented to the user. Each variable name is given in the format specified in `variables(5)`. If the `objectID` argument is not present, `snmpbulkwalk` searches MIB-2.

Example

To retrieve all the variables under `system`, use the command:

```
snmpbulkwalk -v 2c localhost public system
```

The expected output for this command looks like this:

```
system.sysDescr.0 = FreeBSD localhost 4.1-RELEASE FreeBSD 4.1- Fri
Jul i386
system.sysObjectID.0 = OID:
enterprises.covalent.covalentGenericMIB.ctGenericOSoid.255
system.sysUpTime.0 = Timeticks: (2340) 0:00:23.40
system.sysContact.0 = Covalent SNMP Conductor
<webmaster@localhost.com>
system.sysName.0 = localhost
system.sysLocation.0 = Apache Server with Covalent SNMP Conductor
...
```

If the network entity has an error when processing the request packet, an error packet is returned and a message displayed to help identify the reason for the malformed request.

If the tree search causes attempts to search beyond the end of the MIB, the message is displayed: `End of MIB.`

NOTE: As the name implies, `snmpbulkwalk` utilizes the SNMP `GET-snmpcmd(1)`, `variables(5)`.

snmptrapd

Name

`snmptrapd` — Receive and log SNMP trap messages.

Syntax

```
snmptrapd [ -a ] [ -D ] [ -d ] [ -e ] [ -f ] [ -H ] [ -h ]
[ -l[d0-7] ] [ -P ] [ -p port ] [ -q ] [ -S ] [ -s ]
```

Arguments

- a
Makes `snmptrapd` ignore Authentication Failure traps.
- D
Turn debugging output on.
- d
Causes the application to dump input and output packets.
- f
Don't fork away from the caller when using `syslog()`.
- h
Print a usage summary and exit.
- l [d0-7]
Specifies the syslog facility to use, `demon` or `local[0-7]`.
- P
Print the logged messages to `stdout`.
- p port
Specifies the port to run on, if the default 162 is not desired.
- q
Causes the application to use a quicker, less verbose output form.
- S
Requests short object identifiers on output: MIB-name:suffix, i.e., `system.sysDescr.0` is printed as `SNMPv2-MIB:sysDescr.0`
- s
Log the messages to `syslog(8)`. These syslog messages are sent with the level of `LOG_WARNING`, and to the `LOG_LOCAL0` facility (by default). The demon also forks away from its caller when using syslog facilities.

Description

`snmptrapd` is an SNMP application that receives and logs SNMP trap messages sent to the SNMP-TRAP port (162) on the local machine.

The log messages are of the form:

```
Sep 17 22:39:52 www.covalent.net snmptrapd: 128.2.13.41: Cold Start  
Trap (0) Uptime: 8 days, 0:35:46
```

`snmptrapd` must be run as root so UDP port 162 can be opened.

snmpgetnext

Name

`snmpgetnext` — communicates with a network entity using SNMP GET NEXT Requests.

Syntax

```
snmpgetnext [ common arguments ] objectID [ objectID ] ...
```

Description

`snmpget` is an SNMP application that uses the GET NEXT Request to query for information on a network entity. One or more object identifiers can be given as arguments on the command line. Each variable name is given in the format specified in `variables(5)`. For each one, the variable that is lexicographically "next" in the remote entity's MIB is returned.

Example

To retrieve the next variable of `sysUpTime`, use the command:

```
snmpgetnext localhost public sysUpTime
```

The expected output for this command looks like this:

```
system.sysUpTime.0 = Timeticks: (97947)
```

If the network entity has an error processing the request packet, an error message is displayed to help identify the reason for the malformed request.

snmpusm

Name

`snmpusm` — creates and maintains SNMPv3 users on a remote entity.

Syntax

```
snmpusm [ common arguments ] create username [ cloneFromUser ]
snmpusm [ common arguments ] delete username
snmpusm [ common arguments ] cloneFrom username cloneFromUser
snmpusm [ common arguments ] passwd -O old_passphrase -N
new_passphrase [ -a ] [ -o ] [ -x ]
```

Description

`snmpusm` is an SNMP application that can be used for simple maintenance on a SNMP agent's User-based Security Module (USM) table. Passwords of users configured on a running SNMP agent can be deleted, cloned, and changed.

The SNMPv3 USM specifications (see RFC 2574) dictate that users are created and maintained by adding and modifying rows to the `usmUser` MIB table. To create a new user, simply create the row using `snmpset`. User profiles contain private keys that are never transmitted over wire in clear text (regardless of whether administration requests are in encrypted or not).

The secret key for a user is initially set by cloning another user in the table, so that a user inherits the cloned user's secret key. A user can only be cloned once, after which the user must be deleted and re-created to be re-cloned. The authentication and privacy security types are also inherited during this cloning (e.g., MD5 vs. SHA1). To change the secret key for a user, both the user's old and new key must be identified. The `passwd` sub-command of the `snmpusm` command, therefore, requires both the new and the old password.

The SNMP agent comes with a few pre-configured template users that can be used to clone new users after setting the template user's pass-phrases in the `snmpd.conf` file. These users are called "templateMD5" and "templateSHA", and are configured to use MD5 and SHA, respectively, and DES encryption. After cloning from the appropriate template, immediately change the new user's password.

Examples

Assume for our examples that the following VACM and USM configurations lines are in the `snmpd.conf` file for a SNMP agent, which establishes another default user called "initial" with the passphrase "setup_password", so the initial setup of an agent can be performed:

```
# VACM configuration entries
group v3group any initial
view all included .1 80
access v3group "" any auth 0 all all all
# The new user's access:
group v3group any covalent
# USM configuration entries
userSetAuthPass initial * setup_password
userSetAuthPass templateMD5 * initial_MD5_pass
```

NOTE: Remove the "initial" user's setup after creating a real user to whom administrative privileges are granted (like the user `covalent` created in this example).

NOTE: Passwords (passphrases really) must be at least eight characters in length.

Example 1:

The following command uses the user "initial" to create a new user, here named `covalent`:

```
snmpusm -v 3 -u initial -n none -l authNoPriv -a
MD5 -A setup_password localhost create covalent templateMD5
```

`covalent` is cloned from `templateMD5` in the process, so `covalent` inherits that user's password.

Example 2:

After creating the user `covalent` with the same password as the "templateMD5" user, we need to change his passphrase.

```
snmpusm -v 3 -u covalent -n none -l authNoPriv -a MD5
-A initial_MD5_pass localhost passwd -O
initial_MD5_pass -N new_passphrase -a
```

This command changed the passphrase from "initial_MD5_pass" (inherited from the `templateMD5` user) to "new_passphrase".

Example 3:

If the commands in Examples 1 and 2 were successful, the following command should have properly performed an authenticated SNMPv3 GET request to the agent:

```
snmpget -v 3 -u covalent -n none -l authNoPriv -a  
MD5 -A new_passphrase localhost sysUpTime.0
```

Following this command, remove the vacm "group" `snmpd.conf` entry for the "initial" user. The valid user `covalent` can now be used for future (instead of initial) transactions.

snmpset

Name

`snmpset` — communicates with a network entity using SNMP SET Requests.

Syntax

```
snmpset [ common arguments ] objectID type value [ objectID type  
value ] ...
```

Description

`snmpset` is an SNMP application that uses the SET Request to set information on a network entity. One or more fully qualified object identifiers must be entered as arguments on the command line. A type and a value to set must accompany each object identifier. Each variable name is given in the format specified in variables(5).

The type is a single character, one of:

a	IPADDRESS
d	DECIMAL STRING
i	INTEGER
n	NULLOBJ
o	OBJID
s	STRING
t	TIMETICKS
x	HEX STRING

snmpget

Name

`snmpget` — communicates with a network entity using SNMP GET Requests.

Syntax

```
snmpget [ common arguments ] objectID [ objectID ] ...
```

Description

`snmpget` is an SNMP application that uses the GET Request to query for information on a network entity. One or more fully-qualified object identifiers can be given as arguments on the command line. Each variable name is given in the format specified in `variables(5)`.

Example

To retrieve the variable of `sysUpTime.0`, use the command:

```
snmpget localhost public sysUpTime.0
```

The expected output for this command looks like this:

```
system.sysUpTime.0 = Timeticks: (97947) 0:16:19.47
```

If the network entity has an error when processing the request packet, an error packet is returned and a message displayed to help identify the reason for the malformed request. If there were other variables in the request, the request is resent without the bad variable.



Section 5

MIB modules

This section provides a summary of the Management Information Base (MIB) modules implemented for the Covalent SNMP Conductor module, as well as some examples of their use. For detailed and authoritative descriptions, please refer to the MIB module definitions provided with Covalent SNMP Conductor.

Index of MIB modules

Application-specific MIB modules (IETF-defined)

WWW-MIB	46
NETWORK-SERVICES-MIB	46

Enterprise-specific MIB modules

COVALENT-GENERIC-MIB	47
COVALENT-WWW-RESP-NOTIFY-MIB	47
COVALENT-APACHE-CONFIG-MIB	47
COVALENT-APACHE-STATUS-MIB	50
COVALENT-APACHE-MODULES-MIB	50

Network management framework MIB modules (IETF-defined)

SNMPv2-MIB	50
SNMP-FRAMEWORK-MIB	50
SNMP-MPD-MIB	51
SNMP-USER-BASED-SM-MIB	51
SNMP-VIEW-BASED-ACM-MIB	51

Application-specific MIB modules

The following are Internet Engineering Task Force (IETF)-defined MIB modules.

WWW-MIB

This IETF-defined MIB module provides management information for WWW services.

This module is defined in RFC 2594, "Definitions of Managed Objects for WWW Services", <http://www.ietf.org/rfc/rfc2594.txt>.

NETWORK-SERVICES-MIB

This IETF-defined MIB module provides management information for networked applications.

This module is defined in RFC 2788, "Network Services Monitoring MIB", <http://www.ietf.org/rfc/rfc2788.txt>.

Enterprise-specific MIB modules

The following are enterprise-specific MIB modules.

COVALENT-GENERIC-MIB

This enterprise-specific MIB module conveys SNMP agent-specific definitions and generic traps.

COVALENT-WWW-RESP-NOTIFY-MIB

This enterprise-specific MIB module defines managed objects, enabling a notification/trap for WWW services based on the responseType that was returned to a client during a document access attempt.

COVALENT-APACHE-CONFIG-MIB

The `COVALENT-APACHE-CONFIG-MIB` contains managed objects specific to the Apache Web server. You can use these for runtime configuration changes.

This MIB module provides SNMP-based configuration for the following Apache directives:

- `ExtendedStatus`
- `MaxServers`
- `MinSpareServers`
- `MaxSpareServers`
- `MaxRequestsPerChild`

This set of runtime directives gives you the opportunity to "balance" the performance of your Apache Web server. Setting these values will require a compromise between performance, reliability and monitoring detail.

For example, you can "bump up" both `LogLevel` and `ExtendedStatus` to provide more detail on the operation of the server at the expense of performance. Or you can switch off these parameters altogether at the expense of losing detailed monitoring information. Depending on the situation, the operator is likely to change these values while investigating a problem.

Use the `snmpset` command (refer to “`snmpset`” on page 43) to set the directive values, using the syntax :

```
snmpset [ common arguments ] objectID type value
```

For example:

```
snmpset -v 1 -p 161 localhost public ctExtendedStatus.0 i 2
```

ExtendedStatus

The `ctExtendedStatus` object controls whether the server keeps track of extended status information for each request. This is only useful if the status module is enabled on the server. This setting applies to the entire server, and cannot be enabled or disabled on a virtual host-by-virtual host basis.

The following `snmpset` command would enable or disable the extended status information of Apache.

For example, to enable the extended status execute:

```
snmpset localhost public ctExtendedStatus.0 i enable
snmpset localhost public ctExtendedStatus.0 i 1
```

Or, to disable the extended status execute:

```
snmpset localhost public ctExtendedStatus.0 i disable
snmpset localhost public ctExtendedStatus.0 i 2
```

MaxServers

The `ctMaxServers` object limits the number of simultaneous requests can be supported; not more than this number of child server processes will be created.

Any connection attempts over the `MaxClients` limit will normally be queued, up to a number based on the `ListenBacklog` directive. Once a child process is freed at the end of a different request, the connection will then be serviced.

For example, to set (via SNMP) the maximum number of servers that handle requests to 50 execute:

```
snmpset localhost public ctMaxServers.0 i 50
```

MaxSpareServers

The `MaxSpareServers` object sets the maximum number of idle child server processes. An idle process is one which is not handling a request. If there are more than `MaxSpareServers` idle, the parent process will kill off the excess processes. By giving processes a finite lifetime, the number of processes is reduced when the server load reduces.

Tuning this parameter should only be necessary on very busy sites. Setting this parameter to a large number is almost always a bad idea.

For example, to set (via SNMP) the maximum number of spare servers that handle requests to 15, if you are logged in as root execute:

```
snmpset localhost public ctMaxSpareServers.0 i 15
```

MinSpareServers

The `MinSpareServers` object sets the minimum number of idle child server processes. An idle process is one which is not handling a request. If there are fewer than `MinSpareServers` idle, then the parent process creates new children at a maximum rate of one per second.

Tuning this parameter should only be necessary on very busy sites. Setting this parameter to a large number is almost always a bad idea.

For example, to set (via SNMP) the minimum number of spare servers to 10:

```
snmpset localhost public ctMinSpareServers.0 i 10
```

MaxRequestsPerChild

The `ctMaxRequestsPerChild` object limits the number of requests that an individual child server process will handle. After `MaxRequestsPerChild` requests, the child process will die.

If `MaxRequestsPerChild` is zero, then the process will never expire. Setting `MaxRequestsPerChild` to a non-zero limit controls the amount of memory the process can consume in the event of memory leakage.

For example, to set (via SNMP) the maximum number of requests handled per child process to 10:

```
snmpset localhost public ctMaxRequestsPerChild.0 i 10
```

COVALENT-APACHE-STATUS-MIB

This enterprise-specific MIB module defines managed objects to convey management information about the server status of the Apache Web server.

COVALENT-APACHE-MODULES-MIB

This enterprise-specific MIB module defines managed objects to convey management information about the current Apache Web server modules.

Network management framework MIB modules

The following are IETF-defined network management framework MIB modules.

SNMPv2-MIB

This IETF-defined MIB module is a standard required for SNMP agents. It provides managed objects for the system, SNMP statistics, and well-known notifications.

This module is defined in RFC 1907, "Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)", <http://www.ietf.org/rfc/rfc1907.txt>.

SNMP-FRAMEWORK-MIB

This IETF-defined MIB module is used for SNMPv3 only. It is the SNMP architecture MIB and provides generic managed objects and definitions.

This module is defined in RFC 2571, "An Architecture for Describing SNMP Management Frameworks", <http://www.ietf.org/rfc/rfc2571.txt>.

SNMP-MPD-MIB

This IETF-defined MIB module is used for SNMPv3 only. It contains management information and definitions for Message Processing and Dispatching.

This module is defined in RFC 2572, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", <http://www.ietf.org/rfc/rfc2572.txt>.

SNMP-USER-BASED-SM-MIB

This IETF-defined MIB module is used for SNMPv3 only. It contains management information and definitions for the User-based Security Model (USM) for SNMP.

This module is defined in RFC 2574, "User-based Security Model (USM) for Version 3 of the Simple Network Management Protocol (SNMPv3)", <http://www.ietf.org/rfc/rfc2574.txt>.

SNMP-VIEW-BASED-ACM-MIB

This IETF-defined MIB module is used for SNMPv3 only. It contains management information and definitions for the View-based Access Control Model (VACM) for SNMP.

This module is defined in RFC 2575, "View-based Access Control Model (VACM) for the Simple Network Management Protocol", <http://www.ietf.org/rfc/rfc2575.txt>.



Section 6

Troubleshooting

If you do not have root permissions:

Non-root users are generally not allowed to bind to port 80. Therefore, the default port for HTTP and HTTPS traffic is set to 8080 for non-root installations. This enables you to test your server without editing the `httpsd.conf` file. Test with:

```
http://www.yourserver.com:8080/
```

Be sure to edit `httpsd.conf` to use the standard port for HTTP (port 80) before your secure site goes live.

NOTE: The default SNMP port is set to 1610 for non-root installations.

If starting and stopping the server fails during installation:

During the installation of SNMPv3, the server needs to be started and stopped. This process will fail if there is an Apache server running on the port where the new Apache server will run.

- For root installations, the new Apache server will run on port 80.
- For non-root installations, the new Apache server will run on port 8080.

Before installing, you must stop all servers running on that port (80 for root or 8080 for non-root installations).

If you don't receive the expected response from the `snmpwalk` command:

- Check which tools you are using with the command `which snmpwalk`. If this is not pointing to `/path/to/raven/conductor1.0/bin/snmpwalk`, try executing the command with the full path name.
- Check to determine whether there is a daemon listening on port 161 with the command:

```
netstat |grep 161
```

If this does not provide a response, the SNMP agent did not start up. Try stopping and restarting the server.

- If there is another process using port 161, that must be stopped prior to restarting the server.

If you have problems starting your server:

- Execute:

```
./httpsdctl configtest
```

- Verify that the `SSLCertificateFile` and the `SSLCertificateKeyFile` directives in the `httpsd.conf` file point to your certificate and key files.
- Verify that the `ServerName` directive corresponds to the name of your server.
- Verify that the server is not already running by executing:

On BSD compatible systems:	<code>ps -aux grep http</code>
On System V release 4 compatible systems:	<code>ps -elf grep http</code>
On Linux systems:	<code>ps -ax grep http</code>

If this results in a list of processes, stop the running server before starting your new server.

- Check the `httpsd_error_log` file in the `/path/to/apache/logs` directory.
- Verify the syntax of your configuration file by executing:

```
./httpsdctl configtest
```

The server should respond with `syntax ok`.

NOTE: If necessary, refer to the table in step 3 of “Adding to an existing Apache server” on page 16.

If you are adding Covalent SNMP Conductor to an existing Apache installation:

The configuration must have `mod_so.c` enabled, otherwise you need to recompile Apache with the `--enable-module=mod_so` flag in the configuration.

If your existing Apache server is older than 1.3.9 with EAPI:

You must upgrade your server to use the Covalent SNMP Conductor module.

If you receive the following error:

```
bash-2.01$ ./snmpwalk -p 161 localhost public
ld.so.1:
/path/to/raven/module/conductor1.*/platform/sparc-sun-solaris-2.x/
snmpwalk: fatal: libz.so: open failed: No such file or directory
Killed
```

Set your `LD_LIBRARY_PATH` with the command:

```
export LD_LIBRARY_PATH=/usr/local/lib
```

If the above recommendations do not help:

Contact Covalent's online support and the Covalent SNMP Conductor online documentation and FAQ at:

<http://www.covalent.net/support/snmp>

Appendix A:

Manual installation

This section explains how to perform a quick manual installation, configure the Apache and SNMP configuration files, and test your installation.

NOTE: In order to use this product your Apache Web server needs to be compiled with `--enable-module=mod_so`.

Installing manually

To install Covalent SNMP Conductor manually, perform the following steps.

- 1 Unpack the tarball in the directory where your Apache directory is located. The default is `/usr/local`:

```
cd /usr/local/  
gunzip -c Conductor-1.*-1.3.*-platform.tar.gz | tar xf -
```

- 2 If you are using a standard (non-Covalent) Apache installation, in your `httpd.conf` file, add the directives that load the SNMP module as a shared library. Place each line above the line that loads or adds the SSL module:

```
LoadModule snmp_agt_module /path/to/raven/module/conductor1.0/  
libexec/libsnmp_agt.so
```

Each can be positioned just before the place where the SSL module is respectively loaded or added.

Configuring the configuration files

To configure your Apache configuration file:

- 1 In your `httpd.conf` file, add the extra directives that point to the location of the Covalent SNMP Conductor configuration file and the persistent snmp information:

```
SNMPconf /path/to/raven/module/conductor1.0/conf
SNMPvar  /path/to/raven/module/conductor1.0/var
```

- 2 In your `httpd.conf` file, the `maxClients` directive must be placed after the `LoadModule` and `AddModule` of the Covalent SNMP Conductor module.

Therefore, delete the `maxClients` configuration line from its current position, and add it as the last directive in the `httpd.conf` file.

To configure your SNMP configuration file:

Set up your `snmpd.conf` file to enable access to the managed objects in the MIB. The `httpd` directive `SNMPconf` points to the location of the `snmpd.conf` file. Refer to “Appendix B: Configuration examples” on page 60.

NOTE: Refer to the `snmpd.conf` file for more configuration information.

Starting and stopping your server

Starting the server

If you installed the Apache binary included with Covalent SNMP Conductor, to start your Apache Web server execute:

```
apache/bin/httpsdctl start
```

The output should be:

```
httpsdctl start: httpd started
```

NOTE: Refer to “Troubleshooting” on page 52 if you have problems starting your server.

Stopping the server

To stop your Apache Web server execute:

```
apache/bin/httpsdctl stop
```

Testing your installation

Testing if you installed as a root user

The following steps only apply if you installed Covalent SNMP Conductor as a root user. If you did *not* install as a root user, skip these steps and proceed to “Testing if you installed as a non-root user” on page 58.

- 1 To test SNMPv1 enter the following, where *public* is your community string:

```
bin/snmpwalk localhost public
```

- 2 To test SNMPv2c enter the following, where *public* is your community string:

```
bin/snmpwalk -v 2c localhost public
```

- 3 To test SNMPv3 enter the following, where *password* is the password you established while installing Covalent SNMP Conductor:

```
bin/snmpwalk -v 3 -l authNoPriv -u covalent -a MD5 -A password localhost
```

NOTE: The default user after installing Covalent SNMP Conductor is called *covalent*.

Testing if you installed as a non-root user

The following steps only apply if you did *not* install Covalent SNMP Conductor as a root user.

When you install Covalent SNMP Conductor as a non-root user, the module uses port 1610. Therefore, when testing your installation you must include the `-p 1610` option after each command, as indicated in the following steps.

- 1 To test SNMPv1 enter the following, where *public* is your community string:

```
bin/snmpwalk -p 1610 -v 1 localhost public
```

- 2 To test SNMPv2c enter the following, where *public* is your community string:

```
bin/snmpwalk -p 1610 -v 2c localhost public
```

- 3 To test SNMPv3 enter the following, where *password* is the password you established while installing Covalent SNMP Conductor:

```
bin/snmpwalk -p 1610 -v 3 -l authNoPriv -u covalent -a MD5 -A  
password localhost
```

NOTE: The default user after installing Covalent SNMP Conductor is called *covalent*.

Testing if you have configured the SNMP agent for SNMPv3

Perform these additional steps only if you have configured the SNMP agent for SNMPv3:

- 1 Stop your Apache Web server.
- 2 Delete the `createUser` lines from the `snmpd.conf` file.
- 3 Start your Apache Web server.

Appendix B:

Configuration examples

Configuring the SNMP agent

At the beginning of the `snmpd.conf` file:

```
# snmpaddress defines which local address the SNMP agent listens to.
snmpaddress <IP>:<PORT>
```

Configuring access control

First, map the community name into a security name:

```
#          sec.name  source      community
com2sec   username   localhost   community
com2sec   username   network/24  community
```

Second, map the security names into groups:

```
#          sec.model  sec.name
group MyRWGroup v1   username
group MyRWGroup v2c  username
group MyRWGroup usm  username
group MyOWGroup v1   username
group MyOWGroup v2c  username
group MyOWGroup usm  username
```

Third, create a view for us to let the groups have rights to:

```
#          inc/excl  subtree  mask
view all   included   .1       80
```

Finally, grant the groups access to a view with different permissions:

```
#          context sec.model  sec.level  match  read  write  notify
access   " "      any       noauth    exact  all   none   none

MyROGroup
access   " "      any       noauth    exact  all   all    none

MyRWGroup
```

Configuring user security (SNMPv3 only)

Create a user. (After the first start and stop these lines should be deleted.)

```
createUser local MD5 <PASSPHRASE> DES
createUser mynetwork MD5 <PASSPHRASE> DES
```

NOTE: Your <PASSPHRASE> need to be at least eight characters long!

Configuring system information

```
syslocation Right here, right now.
syscontact Me <me@somewhere.org>
```

-- End of the `snmpd.conf`

NOTE: Refer to the `snmpd.conf` file for more configuration information.

Appendix C:

Default SNMP agent configuration

The default SNMP agent configuration allows access for: SNMPv1 and SNMPv2c (which can be restricted by network address); SNMPv3 only; or SNMPv1, SNMPv2c, and SNMPv3. The **Installation Program** does not allow for fine-tuning of the SNMP agent configuration; however, it provides the following basic configuration that you can change to suit your needs.

Default SNMP agent configuration:

```
#####
#
# Conductor SNMP configuration file.
#
# This file is used to configure the Covalent SNMP Conductor
# agent.
#
# All lines beginning with a '#' are comments and are intended for you
# to read. All other lines are configuration commands for the agent.
#
# For more information, please refer to the snmpd.conf entry in the
# in the command reference section of the manual.
#
# Network setup:
# 'snmpaddress' defines to which localaddress the SNMP agent listens.
agentaddress 1610

#####
# View-based Access Control
#####
# This section explains on how to configure the View Based Access
# control.
#
# If you want SNMPv1 and/or SNMPv2c access, you need to map
# the community name into a security name.
# SYNTAX:
#com2sec <SECNAME> <NETWORK> <COMMUNITY>
# <SECNAME> is used as user within the SNMP agent.
# <NETWORK> is used to defined from which hosts or networks
#the SNMP agent can be accessed by using the associated
#community string.
# <COMMUNITY> is the community string as used in SNMPv1 and SNMPv2c
```

```

#for authentication.

com2sec local localhost public
com2sec mynetwork 10.0.0.0/255.255.255.0 public

####
# You need to map the <SECNAME>'s into one or muple groups
# and assicate a security model to it.
# SYNTAX:
#group <GROUPNAME> <SECMODEL> <SECNAME>
# <GROUPNAME> is the idenitfication of the group.
# <SECMODEL> is the security model used. The agent
# uses: 'v1', 'v2c', 'usm' or 'any'
# <SECNAME> is the name via which SNMPv1 and SNMPv2c
#have access or the 'username' of SNMPv3

group MyRWGroup v1 local
group MyRWGroup v2c local
group MyROGroup v1 mynetwork
group MyROGroup v2c mynetwork

####
# You need to create a view of the MIB that can either be a
# complete or partial view.
# SYNTAX:
#view <VIEWNAME> <TYPE> <SUBTREE> <MASK>
# <VIEWNAME> identification for a MIB view.
# <TYPE> is either 'included' or 'excluded' and perform
#the associated action on the <SUBTREE>
# <SUBTREE> the MIB subtree to which the view applies.
# <MASK> is on the places in the OID-string for the accessed objects
# that
#must be match to included in this view.

view all included .1 80

####
# Grant each group access and define its permissions.
# SYNTAX: access <GROUPNAME> <CONTEXT> <SECMODEL> <SECLEVEL> <MATCH>
# <READ> <WRITE> <NOTIFY>
# <GROUPNAME> the group identification as defined with 'group'.
# <CONTEXT> the context in which access is granted. Default ""
# <SECMODEL> the security model required to be used for access
# The agent uses: 'v1', 'v2c', 'usm' or 'any'
# <SECLEVEL> The security level used for access to the MIB. The agent
#uses: 'noauth', 'auth' or 'priv'
# <MATCH> specifies how CONTEXT should be matched against the

```

```

#context of the incoming pdu, either exact or prefix.
# <READ> specifies the view for corresponding access.
# <WRITE> specifies the view for corresponding access.
# <NOTIFY> specifies the view for corresponding access.
access MyROGroup "" any noauth exact all none none
access MyRWGroup "" any noauth exact all all none

#####
# User-based Security
#####
#####
# For SNMPv3 you need to create users. However, after starting and
# stopping you would like to delete the 'createuser' line. The SNMP
# agent will store this information encrypted, in a persistent
# configuration file.
#
# We first want to create or map the SNMPv3 user into a group and
# provide access with the above view-based access control.
# For instance, group MyRWGroup usm covalent

group MyRWGroup usm covalent

# For SNMPv3 (User-based Security) we need create a user.
# SYNTAX: CreateUser <USERNAME> <MD5/SHA> <MD5PASSPHRASE> [DES]
# <PASSPHRASE>
# <USERNAME> the SNMPv3 user name. Maps in to view-based access
#control on the <SECNAME>
# <MD5/SHA> type of authentication
# <MD5PASSPHRASE> the MD5 or SHA passphrase.
# <DES> the privacy protocol
# <DESPASSPHRASE> the privacy passphrase.
# Your passphrases must be at least 8 characters long!

# NOTE: If you have used the installer that was provided with Conductor
# You probably have already created the user 'covalent'.
# This was done if you were asked for an SNMPv3 password.
#####
# System information
#
# One can define some values to be returned for the system group.
# 'sysLocation' where the system is located.
# SYNTAX: syslocation <STRING>
# 'sysContact' for this system.
# SYNTAX: syscontact <STRING>

syscontact Covalent SNMP Conductor <webmaster@localhost.com>
syslocation Apache Server with Covalent SNMP Conductor

```