OpenLane® SLM 5.5

Administrator's Guide

Document No. 7800-A2-GB32-00

January 2002

Warranty, Sales, Service, and Training Information

Contact your local sales representative, service representative, or distributor directly for any help needed. For additional information concerning warranty, sales, service, repair, installation, documentation, training, distributor locations, or Paradyne worldwide office locations, use one of the following methods:

- **Internet:** Visit the Paradyne World Wide Web site at **www.paradyne.com**. (Be sure to register your warranty at **www.paradyne.com/warranty**.)

- **Telephone:** Call our automated system to receive current information by fax or to speak with a company representative.

  — Within the U.S.A., call 1-800-870-2221

  — Outside the U.S.A., call 1-727-530-2340

Document Feedback

We welcome your comments and suggestions about this document. Please mail them to Technical Publications, Paradyne Corporation, 8545 126th Ave. N., Largo, FL 33773, or send e-mail to **userdoc@paradyne.com**. Include the number and title of this document in your correspondence. Please include your name and phone number if you are willing to provide additional clarification.

Trademarks

ACCULINK, COMSPHERE, FrameSaver, Hotwire, MVL, NextEDGE, OpenLane, and Performance Wizard are registered trademarks of Paradyne Corporation. ReachDSL and TruePut are trademarks of Paradyne Corporation. All other products and services mentioned herein are the trademarks, service marks, registered trademarks, or registered service marks of their respective owners.

Contents

About This Guide

1 Introduction to OpenLane SLM

2 Installing OpenLane on Solaris

3   Installing OpenLane on Windows

4   Configuring Distributed Components

5   Apache User Access Authentication

6 Adding Devices and Customer Profiles

7 OpenLane Maintenance

8 Building and Using a Firmware Library

A OLCmd Reference

B Using XML to Produce Reports

Index

About This Guide

Document Purpose and Intended Audience

This guide contains information about the OpenLane Service Level Management (SLM) system, Version 5.5. It is designed both as an introduction for novice users and a reference for experienced users of OpenLane.

Document Summary

Section	Description
Chapter 1, *Introduction to OpenLane SLM*	Describes the general purpose and features of OpenLane.
Chapter 2, *Installing OpenLane on Solaris*	Contains procedures for installing OpenLane on a Solaris platform.
Chapter 3, *Installing OpenLane on Windows*	Contains procedures for installing OpenLane on a Windows platform.
Chapter 4, *Configuring Distributed Components*	Explains how to run a Web server and a poller/reader server separate from the OpenLane management server. To configure a distributed Oracle database, see the *OpenLane SLM 5.5 Oracle Database Administration Instructions*.
Chapter 5, *Apache User Access Authentication*	Explains how to set up user access authentication through Apache.
Chapter 6, *Adding Devices and Customer Profiles*	Presents the basic tasks necessary to set up a customer on OpenLane.
Chapter 7, *OpenLane Maintenance*	Lists and describes the tasks necessary to maintain the OpenLane system.
Chapter 8, *Building and Using a Firmware Library*	Explains how to administer a library of device firmware.
Appendix A, *OLCmd Reference*	Describes the functions of the command line program OLCmd.
Index	Lists key terms, concepts, and sections in alphabetical order.

A master glossary of terms and abbreviations used in Paradyne documents is available online at **www.paradyne.com**. Select *Library → Technical Manuals → Technical Glossary.*

Product-Related Documents

Document Number	Document Title
7800-A2-GB30	*OpenLane SLM Reports Reference Guide*
7800-A2-GZ46	*OpenLane SLM 5.5 Oracle Database Administration Instructions*

Contact your sales or service representative to order additional product documentation.

Paradyne documents are also available online at **www.paradyne.com**. Select *Library → Technical Manuals.*

OpenLane's Online Help is available in the Technical Manuals area, as well as on Paradyne's documentation CD-ROM. The Online Help is the user's guide for OpenLane.

Introduction to OpenLane SLM

<div style="text-align: right; font-size: 4em;">1</div>

OpenLane SLM

Paradyne's OpenLane® Service Level Management (SLM) is an open, standards-based, highly distributable system offering robust scalability and flexibility. A Web browser-enabled user interface provides accessibility anytime, anywhere.

Paradyne's network management solution features support for diagnostics, real-time performance monitoring, historical reporting, and detailed health and status indicators for Paradyne's wide array of SNMP-managed narrow and broadband network access device families. Support is provided for Paradyne's FrameSaver® and Hotwire® products. OpenLane also supports Paradyne's 31xx, 7xxx, and 9xxx T1 and subrate access products.

Network Monitoring

OpenLane lets you proactively and reactively monitor, analyze, and troubleshoot your network with easy-to-use management tools, letting you discover and eliminate trouble spots before they affect service. With OpenLane you can verify line quality and the level of service provided, as well as monitor bandwidth utilization.

OpenLane comprises several features that provide extensive element, network, and service level management capabilities for network service providers, network managers, and their end customers.

Easy Installation

OpenLane is installed through an easy-to-use graphical interface. OpenLane's highly scalable architecture offers flexibility to grow your system as management scope increases. OpenLane can span customer network sizes from small customers with a few dozen devices and a few users to carrier-class installations with thousands of devices and many users.

OpenLane Features

- Easy-to-use Web browser-based user interface

- Support for the full array of Paradyne SNMP-manageable devices

- Support for several different levels of user access

- Easy-to-use graphical interface installs and configures each module

- E-commerce-class distributed architecture supports small-scale end-user networks up to carrier-class deployments

- Distributed architecture supports:

  — Optional customer-supplied Oracle 8i database

  — Distributed Web server

  — Distributed Poller/Reader (on Solaris platform)

  — LDAP (Lightweight Directory Access Protocol) directory service

- Integrates with HP OpenView

- Online Help system with full text search capability

- Device configuration through the Web interface

- Scheduled and on-demand firmware maintenance downloads

- Service level reporting, enabling quality of service (QoS) verification

- Historical data collection and reporting (FrameSaver only)

- On-demand polling

- Real-time device health and status, diagnostics, and performance monitoring

- Extensive Web-based diagnostics, including non-disruptive PVC loopback and end-to-end connectivity testing

Installing OpenLane on Solaris

2

About These Instructions

These instructions provide you with the necessary information to get your OpenLane Service Level Management software installed and running as quickly as possible. Installers should be familiar with Solaris and Solaris administration.

Planning the Installation

Before installing OpenLane:

■ Review the *Installation Requirements* on page 2-2.

■ Verify the JRE version on your system. Refer to *Checking the Java Runtime Environment Version* on page 2-3.

Installation Requirements

Review the following *Minimum Hardware Requirements* and *Software Requirements* to verify that you meet the prerequisites.

Minimum Hardware Requirements

Networks with Less Than 500 Devices*	Networks with 500–2000 Devices	Networks with 2000–5000 Devices
■ Sun Ultra 10 ■ 256 MB RAM	■ Sun Ultra 10 ■ 512 MB RAM	■ Sun Enterprise 450 Server ■ 512 MB RAM

■ 100 MB free disk space
■ Additional disk space for storage of performance data:
  – Cloudscape: 120 KB of disk space per PVC per day (60 KB per DLCI per day)
  – Oracle: 62 KB of disk space per PVC per day (31 KB per DLCI per day)		

* A device is an SNMP-manageable CSU, DSU, or DSL port.

Software Requirements

- Sun Solaris 2.6 or 2.8 operating system (Solaris 2.7 is not supported)

- Java Runtime Environment (JRE) Version 1.1.8_10

  The JRE is included on the OpenLane CD.

- Internet Explorer 5.5 (for Windows-based clients) or Netscape Communicator Version 4.76 or 4.78 (for Solaris-based clients)

- The following default TCP ports are used by the OpenLane server:

  — Port 80: Apache Web server 1.3.12 (for HTTP Requests. The port can be changed; see *Configuring Apache to Use a Different Port* on page 2-11)

  — Port 1521: Oracle Database (if used)

  — Port 2099: Remote Method Invocation (RMI for Application Communications)

  — Port 3890: LDAP (Lightweight Directory Access Protocol) Directory Services

  — Port 8007: Apache JServ (for Java Servlet Communications)

- *Optional:* Distributed architecture with customer-supplied Oracle 8i (Standard or Enterprise)

  OpenLane's internal Cloudscape database accommodates up to 80 PVCs (160 DLCIs). For larger networks, an Oracle database must be installed.

- *Optional:* HP OpenView Network Node Manager 6.1 or 6.2

To install OpenLane on a UNIX system (not Solaris), a graphical interface such as the X Window System is required. This is inherent in Solaris.

Checking the Java Runtime Environment Version

Before installing the OpenLane Service Level Management software, you need to verify that the Java Runtime Environment (JRE) has been installed correctly.

▶ **Procedure**

To verify the JRE installation:

1. Open a terminal window. Type `jre` and press Enter.

   The window should display the Java Runtime Environment Version number and a list of options. Verify that you have **Version 1.1.8_10**.

```
Java(tm) Runtime Loader Version 1.1.8_10          ◀━━━ Version Number
Usage: jre [-options] classname [arguments]
Options:
    -?, -help         print out this message
    -v, -verbose      turn on verbose mode
    -verbosegc        print a message when garbage collection occurs
    -noasyncgc        disable asynchronous garbage collection
    -noclassgc        disable class garbage collection
    -ss<number>       set the maximum native stack size for any thread
    -oss<number>      set the maximum Java stack size for any thread
    -ms<number>       set the initial Java heap size
    -mx<number>       set the maximum Java heap size
    -D<name>=<value>  set a system property
    -classpath <path> set class path to <path>
    -cp <path>        prepend <path> to base class path
    -verify           verify all classes when loaded
    -verifyremote     verify classes loaded from the network (default)
    -noverify         do not verify any classes
    -nojit            disable JIT compiler
```

2. If you receive a message that the JRE cannot be found or another version is found, verify that the path to the correct JRE is in your PATH environment variable. To display the PATH environment variable:

   Type `echo $PATH` and press Enter.

Upgrading from Earlier Software

If you are installing OpenLane on a system that is currently running:

■ **DCE Manager**, see *Uninstalling DCE Manager*, below.

■ **Performance Wizard 4.2**, see *Uninstalling Performance Wizard*, below.

■ **An earlier version of OpenLane 5**, the upgrade is handled automatically during the OpenLane installation. **Do not uninstall the earlier version.**

If you use an Oracle database, you must execute a migration script to update the schema. See the *OpenLane SLM 5.5 Oracle Database Administration Instructions*.

Uninstalling DCE Manager

To uninstall DCE Manager, follow this procedure before installing OpenLane:

▶ **Procedure**

1. Start the DCEinstall program.

   — If you downloaded the software from the World Wide Web, change to the directory where you expanded the file and run the installation program:

     cd <directory>

     ./DCEinstall

   — If you mounted the CD-ROM for Sun systems:

     cd /tmp/cdrom/DCE/Solaris2

     ./DCEinstall

2. The script detects the existing version and prompts you to uninstall it. Select **OK**.

3. When the software is uninstalled, the installation script prompts you to install DCE Manager. Select **No**.

Uninstalling Performance Wizard

▶ **Procedure**

1. Run UnInstallToOV:

   cd /opt/OV/pdn/PerfWiz

   ./UnInstallToOV

2. Delete the /opt/OV/pdn/ directory and all subordinate directories and files:

   rm -rf /opt/OV/pdn

The OpenLane Installation Program

To install OpenLane and the Apache Web server, execute the OpenLane **setup.sh** program. Follow the instructions and respond to the queries it presents. The OpenLane installation program:

■ Optionally installs and configures the Apache 1.3.12 Web server and the Apache JServ. (This is not required for upgrades from earlier versions of OpenLane 5.)

By default, the Apache software is installed with the following parameters:

Apache Parameter	Setting
Installation directory	/opt/apache
TCP Port for HTTP requests	80
User	nobody
Group	nobody

These options can be changed during the installation process. To modify the Apache Web server to use a port other than port 80 after installation, see *Configuring Apache to Use a Different Port* on page 2-11.

The default installation path for Apache software is:

/opt/apache

■ Installs the OpenLane components, including the Cloudscape Java SQL Database.

The default installation path for OpenLane software is:

/opt/pdn/OpenLane

■ Provides a startup command to control all Apache and OpenLane processes.

■ Includes Management Platform Integration Adapters for HP OpenView Network Node Manager for Sun Solaris.

The software-based platform adapters provide integration points for:

— Paradyne device-specific map icon creation

— Trap definition files

In addition, OpenLane is added to OpenView's menu structure.

NOTES:

— You do not have to install the Apache Web server in /opt/apache and OpenLane in /opt/pdn/OpenLane. However, it is *highly recommended* that you install Apache and OpenLane on the same mount point.

If you do not use the standard directory names, substitute the names you use for the directory references in this document.

— If you will be distributing the Web server, you must install OpenLane and Apache in the same directory path on the Web server and the OpenLane Management system.

Obtaining a License Key

An OpenLane license key is required to install OpenLane. You can request a temporary license key from the Paradyne Web site at **www.paradyne.com/pdn/key**. A 30-day temporary license key is provided by e-mail.

Installing from Paradyne's Web Site

▶ **Procedure**

1. Log on to the system as **root**.

2. Obtain a temporary license key and the OpenLane software from **www.paradyne.com/pdn/key**.

3. Save the setup.sh file in **/tmp** or any directory you choose.

4. Exit your Web browser if it is running. Change to the directory chosen in Step 3.

5. Change the file permissions:

   chmod 777 setup.sh

6. Run the setup.sh file:

   ./setup.sh

7. Follow the instructions on each screen with these exceptions:

   — If you are upgrading a version of OpenLane 5, do not remove the old version when prompted. Select **Continue**.

   — When prompted for the Apache User and Group information, use the default User and Group:

   User = nobody

   Group = nobody

Installing from CD-ROM

► **Procedure**

1. Log on to the system as `root`.

2. Exit your Web browser if it is running. Insert the OpenLane CD-ROM into the CD-ROM drive and mount it.

3. Change to the CD-ROM drive.

4. Change to the OpenLane directory and run the `setup.sh` file.

5. Follow the instructions on each screen with these exceptions:

   — If you are upgrading a version of OpenLane 5, do not remove the old version when prompted. Select **Continue**.

   — When prompted for the Apache User and Group information, use:

   `User = nobody`

   `Group = nobody`

Installing from FTP Files

Contact Technical Support to arrange this type of installation:

■ 1-800-795-8004

Installing Remotely Using X

You can install OpenLane from a remote host using the X Window System.

▶ **Procedure**

1. On the local host, execute the following commands:

   xhost + *remote_host_name*

   rlogin *remote_host_name* **-l** *username*

   username must have root privilege.

2. On the remote host, execute the following command or commands:

   — If using the Bourne shell (sh):

   DISPLAY=*local_host_name***:0.0**

   export DISPLAY

   — If using the Korn shell (ksh):

   export DISPLAY=*local_host_name***:0.0**

   — If using the C shell (csh):

   setenv DISPLAY *local_host_name***:0.0**

3. On the remote host, change the directory to the path where setup.sh resides:

   cd /tmp

4. Run setup.sh:

   ./setup.sh

Installing the Permanent License Key

▶ **Procedure**

1. Change the directory to /opt/pdn/OpenLane:

   cd /opt/pdn/OpenLane

2. Enter the command to install the permanent key:

   OLCmd key *permanent_key*

   Where *permanent_key* is the key e-mailed or faxed to you by Paradyne. For example:

   OLCmd key ALUKB-KAC9Y-H987N-CX5GK

Locating the License Key On an Installed System

To locate the license key on the system once OpenLane is installed, use one of these methods:

- Display the OpenLane About screen by clicking the **About OpenLane Service Level Management** link on the OpenLane Administration screen

- Display the `/opt/pdn/OpenLane/pdnKey.txt` file using a text editor

- Issue the command `OLCmd key` from a command line (see *Displaying the License Key Expiration Date*)

Displaying the License Key Expiration Date

You can view your license key's expiration date at any time using the **key** function of OLCmd.

▶ **Procedure**

1. Open a terminal window and change to the OpenLane directory.

2. Enter the command:

   `OLCmd key`

   The expiration date is displayed in day-month-year format. For example:

   `License expires on: 02-Jan-03`

   This license expires on January 2, 2003.

Migration to Current Release for Oracle

If you use an Oracle database for an earlier release of OpenLane, you must run a migration script to update the schema after OpenLane is installed. See the *OpenLane SLM 5.5 Oracle Database Administration Instructions*.

Starting OpenLane Web and Database Services

▶ **Procedure**

1. From a terminal window, change the directory to /opt/pdn/OpenLane:

   `cd /opt/pdn/OpenLane`

2. Type `OLControl start` and press Enter. The following services are started:

   — Apache Web Server

   — Cloudscape JDBMS (if used)

   — OpenLane Services

   — RMI Application Server

   — SCM Poller

   — OpenLDAP Directory Services

   — User History Poller

   — User History Reader

Stopping OpenLane Web and Database Services

▶ **Procedure**

1. From a terminal window, change the directory to /opt/pdn/OpenLane:

   `cd /opt/pdn/OpenLane`

2. Type `OLControl stop` and press Enter.

Configuring Apache to Use a Different Port

By default, the Apache Web server is installed and configured to use TCP port 80 (unless you changed it at installation time). You can configure the Apache Web server to use another port after installation, as long as that port is not in use by another service.

▶ **Procedure**

To change the TCP port:

1. From a terminal window, change the directory to /opt/pdn/OpenLane:

   `cd /opt/pdn/OpenLane`

2. To stop OpenLane, type **OLControl stop** and press Enter.

3. Change to the Apache configuration directory:

   `cd /opt/apache/conf`

4. Edit the file httpd.conf using a text editor, such as vi:

   `vi httpd.conf`

5. Find the entry for **PORT**. Change the value from 80 to the new port number.

6. Save the file after making changes.

7. Change to the OpenLane directory:

   `cd /opt/pdn/OpenLane`

8. To start OpenLane, type **OLControl start** and press Enter.

Installing OpenLane on Windows

3

About These Instructions

These instructions provide you with the necessary information to get your OpenLane Service Level Management software installed and running as quickly as possible. Installers should be familiar with Windows administration.

Planning the Installation

Before installing OpenLane:

- Review the *Installation Requirements* on page 3-2.

- Verify the JRE version on your system. Refer to *Checking the Java Runtime Environment Version* on page 3-3.

Installation Requirements

Review the *Minimum Hardware Requirements* and to verify that you meet the prerequisites.

Minimum Hardware Requirements

Networks with Less Than 500 Devices*	Network with 500–1000 Devices*
■ 500 MHz Processor** ■ 256 MB RAM	■ 700 MHz Processor** ■ 512 MB RAM

■ 100 MB free disk space
■ Additional disk space for storage of performance data:
  – Cloudscape: 120 KB of disk space per PVC per day (60 KB per DLCI per day)
  – Oracle: 62 KB of disk space per PVC per day (31 KB per DLCI per day)	

* A device is an SNMP-manageable CSU, DSU, or DSL port. The Solaris version of OpenLane is recommended for networks with more than 1000 devices.

** Dual processors are not supported.

Software Requirements

- Windows NT 4.0 with Service Pack 5 or higher

  – or –

  Windows 2000 Server with Service Pack 2 or higher

- Java Runtime Environment (JRE) Version 1.1.8_006

  The JRE is included on the OpenLane CD. It is also available at:
  http://www.paradyne.com/products/ol_support.html

- The following default TCP ports are used by the OpenLane server:

  — Port 80: Apache Web server 1.3.12 (for HTTP Requests. The port can be changed; it must be changed if you use Internet Information Server (IIS). See *Configuring Apache to Use a Different Port* on page 3-11)

  — Port 1521: Oracle Database (if used)

  — Port 2099: Remote Method Invocation (RMI for Application Communications)

  — Port 3890: LDAP (Lightweight Directory Access Protocol) Directory Services

  — Port 8007: Apache JServ (for Java Servlet Communications)

- *Optional:* Distributed architecture with customer-supplied Oracle 8i (Standard or Enterprise)

  OpenLane's internal Cloudscape database accommodates up to 80 PVCs (160 DLCIs). For larger networks, an Oracle database must be installed.

- *Optional:* HP OpenView Network Node Manager 6.1 or 6.2

Web Client Software Requirements

- Internet Explorer Version 5.5 (for Windows-based clients)

    – or –

- Netscape Communicator Version 4.76 or 4.78 (for Solaris-based clients)

Checking the Java Runtime Environment Version

Before installing the OpenLane Service Level Management software, you need to verify that the Java Runtime Environment (JRE) has been installed correctly.

▶ **Procedure**

To verify the JRE installation:

1. Open an MS-DOS command prompt window. Type **jre** and press Enter.

    The window should display the Java Runtime Environment Version number and a list of options. Verify that you have **Version 1.1.8_006**.

```
C:\>jre
Java(tm) Runtime Loader Version 1.1.8_006    ◄───── Version Number
Usage: jre [-options] classname [arguments]
Options:
    -?, -help           print out this message
    -v, -verbose        turn on verbose mode
    -verbosegc          print a message when garbage collection occurs
    -noasyncgc          disable asynchronous garbage collection
    -noclassgc          disable class garbage collection
    -ss<number>         set the maximum native stack size for any thread
    -oss<number>        set the maximum Java stack size for any thread
    -ms<number>         set the initial Java heap size
    -mx<number>         set the maximum Java heap size
    -D<name>=<value>    set a system property
    -classpath <path>   set class path to <path>
    -cp <path>          prepend <path> to base class path
    -verify             verify all classes when loaded
    -verifyremote       verify classes loaded from the network (default)
    -noverify           do not verify any classes
    -nojit              disable JIT compiler
```

2. If you receive a message that the JRE cannot be found or another version is found, completely uninstall and reinstall the JRE software.

Upgrading from Earlier Software

If you are installing OpenLane on a system that is currently running:

■ **DCE Manager**, see *Uninstalling DCE Manager*, below.

■ **Performance Wizard 4.2**, see *Uninstalling Performance Wizard*, below.

■ **An earlier version of OpenLane 5**, the upgrade is handled automatically during the OpenLane installation. **Do not uninstall the earlier version.** If you use an Oracle database, use the supplied migration script to update the schema. See the *OpenLane SLM 5.5 Oracle Database Administration Instructions*.

Uninstalling DCE Manager

▶ **Procedure**

1. Double-click on Control Panel, then double-click on Add/Remove Programs.

2. Select DCE Manager and click on Add/Remove.

Uninstalling Performance Wizard

▶ **Procedure**

1. Open an MS-DOS command prompt window and run \opt\OV\pdn\PerfWiz\UnInstallToOV.bat:

   cd \opt\OV\pdn\PerfWiz

   uninstalltoov

2. Delete the \opt\OV\pdn directory and all subordinate directories and files.

The OpenLane Installation Program

To install OpenLane and the Apache Web server, execute the OpenLane **setup.exe** program. Follow the instructions and respond to the queries it presents. When installation of OpenLane is complete, **restart Windows.**

The OpenLane installation program:

■ Optionally installs and configures the Apache 1.3.12 Web server and the Apache JServ. (This is not required for upgrades from earlier versions of OpenLane 5.)

Default installation path for Apache software:

    C:\Program Files\Apache Group\Apache

Recommended installation path for Apache software:

    C:\opt\apache

The Apache software is configured to use TCP port 80 for http requests. To modify the Apache Web server to use a port other than port 80, see *Configuring Apache to Use a Different Port* on page 3-11.

■ Installs the OpenLane components, including the Cloudscape Java SQL Database.

Default installation path for OpenLane software:

    C:\opt\pdn\OpenLane

■ Establishes OpenLane as a Windows service that starts automatically when Windows does.

■ Includes Management Platform Integration Adapters for:

— HP OpenView Network Node Manager for Microsoft Windows NT

— HP OpenView for Windows Workgroup Node Manager

The software-based platform adapters provide integration points for:

— Paradyne device-specific map icon creation

— Trap definition files

In addition, OpenLane is added to the menu structure of HP OpenView Network Node Manager.

NOTES:

— You do not have to install the Apache Web server at C:\opt\apache and OpenLane at C:\opt\pdn\OpenLane. However, it is *highly recommended* that you install Apache and OpenLane on the same drive.

  If you do not use the standard directory names, substitute the names you use for the directory references in this document.

— If you will be distributing the Web server, you must install OpenLane and Apache in the same directory path on the Web server and the OpenLane Management system. The path cannot exceed 18 characters.

Obtaining the License Key

An OpenLane license key is required to install OpenLane. You can request a temporary license key from the Paradyne Web site at `www.paradyne.com/pdn/key`.

Installing from Paradyne's Web Site

▶ **Procedure**

1. Log on to the Windows system as an administrator.

2. Obtain a license key and the OpenLane software from `www.paradyne.com/pdn/key`.

3. Save the setup.exe file in `C:\temp` or any directory you choose.

4. Exit your Web browser if it is running.

5. Run the `setup.exe` file.

6. Follow the instructions on each screen, with these exceptions:

   — If you are upgrading a version of OpenLane 5:

   Do not remove the old version of OpenLane when prompted. Select **Continue**.

   — If you are installing OpenLane 5 for the first time:

   Do not use the default Apache installation directory. Specify instead: `C:\opt\apache`

   Do not restart the computer if prompted to do so during the Apache portion of the installation.

Installing from CD-ROM

▶ **Procedure**

1. Log on to the Windows system as an administrator.

2. Exit your Web browser if it is running.

3. Insert the OpenLane CD-ROM into the CD-ROM drive.

4. Change to the OpenLane directory on the CD-ROM drive and run the `setup.exe` file.

5. Follow the instructions on each screen, with these exceptions:

   — Do not use the default Apache installation directory. Specify instead:
   `C:\opt\apache`

   — Do not restart the computer when prompted during the Apache portion of the installation.

Installing from FTP Files

Contact Technical Support to arrange this type of installation:

- 1-800-795-8004

Installing the Permanent License Key

▶ **Procedure**

1. Open an MS-DOS command prompt window and change to the OpenLane directory:

   `cd \opt\pdn\OpenLane`

2. Enter the command to install the permanent key:

   `OLCmd key` *permanent_key*

   Where *permanent_key* is the key e-mailed or faxed to you by Paradyne. For example:

   `OLCmd key ALUKB-KAC9Y-H987N-CX5GK`

Locating the License Key On an Installed System

To locate the license key on the system once OpenLane is installed:

- Display the OpenLane About screen by clicking the **About OpenLane Service Level Management** link on the OpenLane Administration screen

- Display the `C:\opt\pdn\OpenLane\pdnKey.txt` file using a text editor

- Issue the command `OLCmd key` from a command line (see *Displaying the License Key Expiration Date* on page 3-8)

Displaying the License Key Expiration Date

You can view your license key's expiration date at any time using the **key** function of OLCmd.

▶ **Procedure**

1. Open an MS-DOS Prompt and change to the OpenLane directory.

2. Enter the command:

   OLCmd key

   The expiration date is displayed in day-month-year format. For example:

   License expires on: 02-Jan-03

   This license expires on January 2, 2003.

Migration to Current Release for Oracle

If you use an Oracle database for an earlier release of OpenLane, you must run a migration script to update the schema after OpenLane is installed. See the *OpenLane SLM 5.5 Oracle Database Administration Instructions*.

Starting OpenLane Web and Database Services

OpenLane runs as a Windows service. It starts automatically when Windows does, whether or not a user has logged in to Windows.

If you have shut down the OpenLane Monitor service manually (see *Stopping OpenLane Web and Database Services* on page 3-10), you can restart it by restarting Windows or by following this procedure.

▶ **Procedure**

To restart OpenLane:

1. Invoke the Services management screen. It can be found:

   — In the Control Panel folder under Windows NT

   — In the Administrative Tools start menu under Windows 2000

2. Issue a **Start** command for the OpenLane Monitor service.
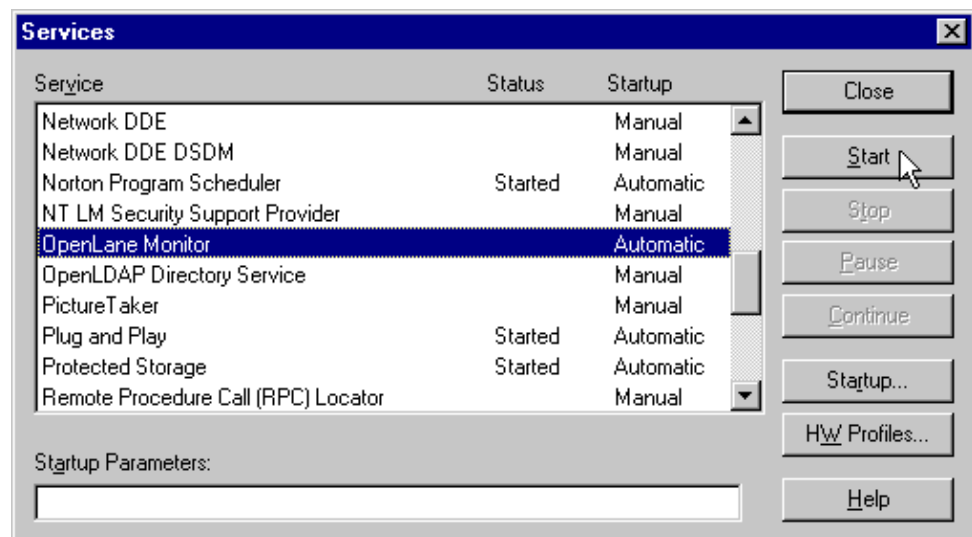
The example shown is from Windows NT.

Stopping OpenLane Web and Database Services

Use the Services management screen to shut down OpenLane. Since the OpenLDAP Directory Service is dependent on the OpenLane Monitor Service, OpenLDAP should be shut down rather than OpenLane Monitor.

▶ **Procedure**

1. Invoke the Services management screen. It can be found:

   — In the Control Panel folder under Windows NT

   — In the Administrative Tools start menu under Windows 2000

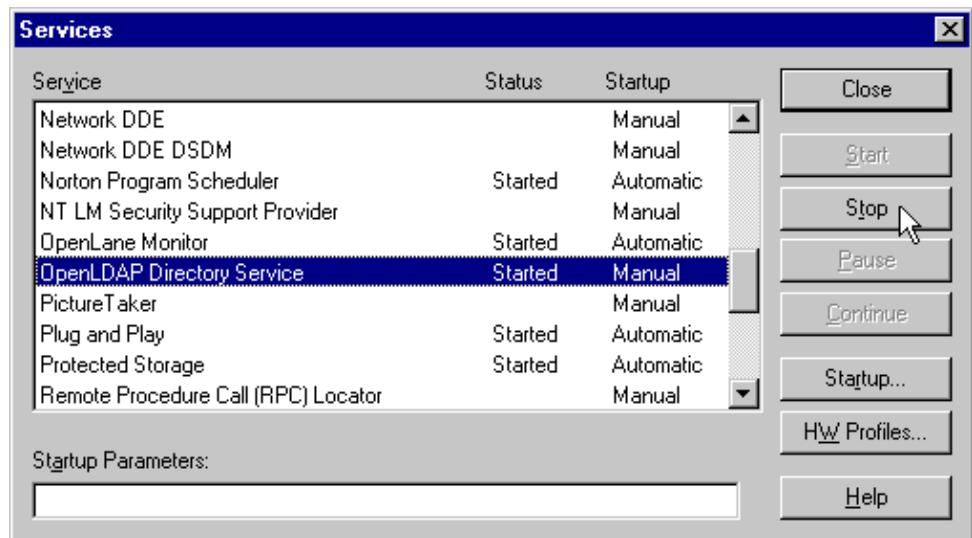2. Issue a **Stop** command for the OpenLDAP Directory Service.

The example shown is from Windows NT.

Configuring Apache to Use a Different Port

By default the Apache Web server is installed and configured to use TCP port 80. You can choose to configure the Apache Web server to use any port you wish as long as that port is not in use by another service.

Internet Information Server (IIS), the Web server installed with Windows 2000 Server, also uses Port 80. To use OpenLane on a system with IIS, you must either disable IIS or change the port used by Apache.

When you use a port other than 80, the port must be specified in URLs directed to the Apache server. For example, if you use Port 8080 on the host olserver.com, the URL for OpenLane is:

```
http://olserver.com:8080/openlane/
```

▶ **Procedure**

To change the TCP port for Apache:

1. If OpenLane is running, stop it. (See *Stopping OpenLane Web and Database Services* on page 3-10.)

2. Change the directory to C:\opt\apache\conf:

    ```
    cd \opt\apache\conf
    ```

3. Edit the file httpd.conf:

    ```
    edit httpd.conf
    ```

4. Find the entry for **PORT**. Change the value from 80 to the new port number.

5. Save the file after making changes.

6. Change the directory to C:\opt\pdn\OpenLane:

    ```
    cd \opt\pdn\OpenLane
    ```

7. Start OpenLane. See *Starting OpenLane Web and Database Services* on page 3-9.

Configuring Distributed Components

4

OpenLane Distributed Components

OpenLane provides a highly distributable framework. Distribution of OpenLane functions can be used to share workload over several servers, and to provide a more secure configuration. This chapter describes:

■ Distributing the Web server

■ Distributing the Poller/Reader

To configure a distributed Oracle database, see the *OpenLane SLM 5.5 Oracle Database Administration Instructions*.

Setting Up Web Server Distribution

With OpenLane, Web Server functionality can be distributed to another server to offload the task of serving Web pages or provide a more secure configuration. The Web Server and the OpenLane Management Server can be Solaris or Windows. The following procedure uses this diagram to illustrate distributing the Web server.

Internet

Web Server
172.20.1.10

OpenLane Server
172.20.1.20

00-16651a
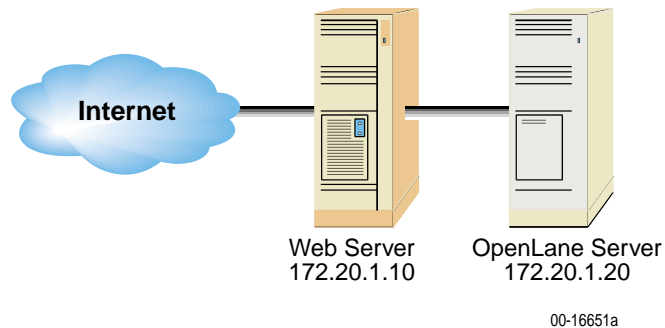
▶ **Procedure**

1. Install OpenLane on the OpenLane Management server and the Web server. Install OpenLane using the same directory names on both servers. The directory path cannot exceed 18 characters.

   Installation Path examples:

   — **Solaris:** If the OpenLane server and/or the Web server use a Solaris server, then install:

   OpenLane in **`/opt/pdn/OpenLane`**

   Apache in **`/opt/apache`** (or your choice of directory. Procedures in this document show the directory as /opt/apache and must be modified for the path you use.)

   — **Windows:** If the Web server uses a Windows server, then install:

   OpenLane in **`C:\opt\pdn\OpenLane`**

   Apache in **`C:\opt\apache`** (or your choice of directory. Procedures in this document show the directory as c:\opt\apache and must be modified for the path you use.)

2. If it is running, stop the OpenLane server on both machines.

3. On the OpenLane server (172.20.1.20), edit the jserv.properties file:

— **Solaris:**

Edit the `/opt/apache/conf/jserv.properties` file and add the address of the Web server to the `security.allowedAddresses` directive:

`security.allowedAddresses=127.0.0.1,172.20.1.20,172.20.1.10`

Set the bind address to the IP address of the OpenLane server:

`bindaddress=172.20.1.20`

Edit the `/opt/apache/conf/jserv.conf` file and remove the comment symbol (#) from `ApJServDefaultHost`. Set the `ApJServDefaultHost` to the IP address of the OpenLane server:

`ApJServDefaultHost 172.20.1.20`

— **Windows:**

Edit the `c:\opt\apache\conf\mod_jserv.properties` file and add the address of the Web server to the `security.allowedAddresses` directive and set the bind address to the IP address of the OpenLane server:

`security.allowedAddresses=127.0.0.1,172.20.1.20,172.20.1.10`

`bindaddress=172.20.1.20`

Edit the `c:\opt\apache\conf\mod_jserv.conf` file and remove the comment symbol (#) from `ApJServDefaultHost`. Set the `ApJServDefaultHost` to the address of the OpenLane server:

`ApJServDefaultHost 172.20.1.20`

4. Restart the OpenLane server.

5. On the Web server (172.20.1.10):

— **Solaris:** Edit `OLControlConfig`. Disable everything but the Web server (disable by changing the value to 0).

— **Windows:** Edit `OLControlConfig.bat`. Disable all functions but the Web server (by changing the value for other functions from 1 to 0). For example:

```
    .
    .
    .
set DATABASE=0
set WEBSERVER=1
set MGMTSVCS=0
set UHPOLLER=0
set UHREADER=0
set SNMPPOLLER=0
set SCHEDRPTS=0
set SCMPOLLER=0
set DIRSVCS=0
    .
    .
    .
```

6. On the Web server (172.20.1.10):

— **Solaris:** Edit the `/opt/apache/conf/jserv.conf` file. Modify these directives:

`ApJServManual On`

`ApJServDefaultHost 172.20.1.20`

`ApJServMount /OLServlets ajpv12://172.20.1.20/OLServlets`

`ApJServMount /OLServlets2 ajpv12://172.20.1.20/OLServlets2`

— **Windows:** Edit the `c:\opt\apache\conf\mod_jserv.conf` file. Modify these directives:

`ApJServManual On`

`ApJServDefaultHost 172.20.1.20`

`ApJServMount /OLServlets ajpv12://172.20.1.20/OLServlets`

`ApJServMount /OLServlets2 ajpv12://172.20.1.20/OLServlets2`

7. Start the Web server.

In this example, you can now access the OpenLane Management System by entering:

`http://172.20.1.10/OpenLane`

Operational Notes

An httpd server will be running on both the Web server and the OpenLane Management server. This will allow you to access the OpenLane Management server if the Web server should fail.

If Apache authentication is enabled, it must be enabled on both systems. Apache authentication is specific to individual machines. To have the same set of users access both machines, you need an identical **.htpasswd** file on both systems. You may choose to allow different sets of users to access each machine by keeping a different .htpasswd file on each machine.

Setting Up the Distributed Poller and Reader

OpenLane allows you to offload the Poller and Reader functions from the main Solaris OpenLane server to another Solaris hardware platform. The following procedures use this diagram to illustrate the distributed poller and reader.

OpenLane
Distributed Poller &
Reader
172.20.1.20
(poller1)

OpenLane
Server
172.20.1.10

00-16757-01
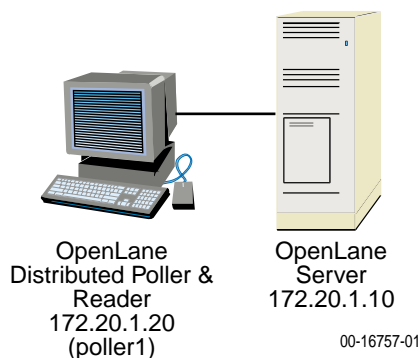
Distributed Poller/Reader — Management Server Setup

▶ **Procedure**

Perform the following steps on the OpenLane Management Server (172.20.1.10 in the example diagram).

1. Stop OpenLane:

   ```
   cd /opt/pdn/OpenLane
   OLControl stop
   ```

2. Edit **/etc/dfs/dfstab** and add the following line at the end:

   ```
   share -F nfs -o rw /opt/pdn/OpenLane/uhpoller
   ```

3. Stop and start the NFS server to put the sharing into effect:

   ```
   /etc/init.d/nfs.server stop
   /etc/init.d/nfs.server start
   ```

4. Verify that the directory is being shared:

   ```
   showmount -e
   ```

5. Create a directory to mirror the one that will be created automatically on the distributed poller/reader (named poller1 in this example):

   ```
   cd /opt/pdn/OpenLane/uhpoller
   mkdir poller_poller1
   ```

6. Set up the permissions:

   ```
   chmod -R 777 /opt/pdn/OpenLane/uhpoller
   ```

7. Restart OpenLane:

   ```
   cd /opt/pdn/OpenLane
   OLControl start
   ```

8. Define the location of the database. Use the appropriate command for your database type:

   ```
   OLCmd dbconfig '-m 172.20.1.10'          (Cloudscape)
   OLCmd dbconfig '-m 172.20.1.10:1521'     (Oracle)
   ```

9. Define the location of the log server:

   ```
   OLCmd logsvc 172.20.1.10
   ```

Distributed Poller/Reader — Poller/Reader Setup

▶ **Procedure**

Perform the following steps on the prospective Distributed Poller/Reader platform (172.20.1.20 in the example diagram).

1. Install OpenLane on the Distributed Poller platform. **Do not install the Apache Web server or traps for HP OpenView.**

2. Edit **/opt/pdn/OpenLane/OLControlConfig** to turn off all services except Management Services, User History Poller, and User History Reader (1 = on; 0 = off). Change **DOMAIN** and **READER_DOMAIN** from **default** to the new domain name (poller1 in this example).

   Example of the OLControlConfig file:

   ```
                     •
                     •
                     •
   DOMAIN=default              ← Change to poller1
   READER_DOMAIN=default       ← Change to poller1
   PARAMS=
   RMI=2099
   LDAP=3890
   DATABASE=1                  ← Change to 0
   WEBSERVER=1                 ← Change to 0
   MGMTSVCS=1
   UHPOLLER=1
   UHREADER=1
   SNMPPOLLER=0
   SCMPOLLER=1                 ← Change to 0
   SCHEDRPTS=0
   DIRSVCS=1                   ← Change to 0
                     •
                     •
                     •
   ```

3. Edit **/opt/pdn/OpenLane/database/config/Preferences.dat** file to define the OpenLane Management Server as the Java naming provider. Change:

   java.naming.provider.url=ldap://localhost:3890

   to:

   java.naming.provider.url=ldap://172.20.1.10:3890

4. Start OpenLane:

   cd /opt/pdn/OpenLane
   OLControl start

   Ignore any Java errors.

5. Stop OpenLane:

   `OLControl stop`

   If OpenLane does not shut down within two minutes, proceed with the next step regardlessly.

6. Reboot the server.

7. Mount the directory:

   ```
   mount -F nfs -o rw
   172.20.1.10:/opt/pdn/OpenLane/uhpoller/poller_poller1
   /opt/pdn/OpenLane/uhpoller/poller_poller1
   ```

   This is a temporary mount and will not be reestablished after a reboot. To set up a permanent mount, add the following line to the `/etc/vfstab` file:

   ```
   172.20.1.10:/opt/pdn/OpenLane/uhpoller/poller_poller1 -
   /opt/pdn/OpenLane/uhpoller/poller_poller1 nfs - yes rw,bg
   ```

8. Verify the mount:

   `mount`

9. Start OpenLane:

   ```
   cd /opt/pdn/OpenLane
   OLControl start
   ```

10. When you add devices to OpenLane's database, specify the new location of the poller/reader. Modify existing devices to use the new poller/reader (poller1 in this example):

Apache User Access Authentication

5

User Access Authentication

User access authentication is used to specify who has access to the Web server. When authentication is enabled on the Web server, it affects all users of the system.

There are many mechanisms that can be used to set user access. HyperText Transfer Protocol (HTTP) Challenge/Response is presented here. You can choose another authentication mechanism or a combination of methods, as long as the authentication mechanism provides the REMOTE_USER environment variable to CGI programs. Methods and protocols include:

- HTTP Challenge/Response

- SSL (Secure Sockets Layer)

- Secure HTTP with a well-configured firewall

- VPN (Virtual Private Network)

- A router

The following are requirements of Apache user access authentication under OpenLane:

- The Customer ID and username must match.

- Authentication is either enabled or disabled for all customer IDs. If authentication is enabled, every customer ID must have an associated user with a username.

- There can only be one username per Customer ID for authentication.

When authentication is enabled, the first OpenLane Login page displays a **Log In** button instead of an **Enter your Customer ID** prompt. Clicking on the button invokes a window for entering User Name and Password.

Configuring HTTP Challenge/Response

Follow this procedure to configure HTTP Challenge/Response for OpenLane access authentication with the Apache Web server.

▶ **Procedure**

1. If OpenLane is running, stop it.

2. Find the directory where Apache is installed. This is the directory where the file named apache (on Solaris) or apache.exe (on Windows) resides.

3. In a command line window, change to the Apache directory with the cd command. For example:

   cd /opt/apache (Solaris)

   cd \opt\apache (Windows)

4. Under the /opt/apache/conf (or C:\opt\apache\conf) directory, open the file called **httpd.conf**. If the Apache configuration files were installed with OpenLane, then the text referred to below is commented out in the httpd.conf file.

   Uncomment the text and make sure the **AuthUserFile** parameter (ordinarily /opt/pdn/OpenLane/database/config/ in Solaris and C:\opt\pdn\OpenLane\database\config\ in Windows) is correct. If the Apache configuration files were installed during an OpenLane installation, AuthUserFile is already set.

   ```
   <Location "/OLServlets/*">
   order allow,deny
   allow from all
   AuthType Basic
   AuthName "OpenLane"
   AuthUserFile /opt/pdn/OpenLane/database/config/.htpasswd
   require valid-user
   </Location>
   ```

   In this case, forward slashes should be used in the directory name for both Solaris and Windows.

   If the commented text is not found, add the text as shown to the bottom of the file, specifying the location of the **.htpasswd** file where usernames and passwords are stored. The directory name must be an absolute path (full directory path and filename).

5. Change to the **/opt/apache/bin** (Solaris) or **c:\opt\apache\bin** (Windows) directory.

6. Run **htpasswd**. The first time you use the htpasswd command, use the **-c** parameter to create the password file:

   Solaris: **htpasswd -c /opt/pdn/OpenLane/database/config/.htpasswd** *username*

   Windows: **htpasswd.exe -c c:\opt\pdn\OpenLane\database\config\.htpasswd** *username*

Where *username* is the name of the user being added. For example:

```
htpasswd -c
  /opt/pdn/OpenLane/database/config/.htpasswd Admin
```

Add additional usernames by running the htpasswd program without the `-c` option.

If additional help is needed, run the .htpasswd program without specifying any command line parameters to display additional help.

7. Change the directory to `/opt/pdn/OpenLane` (Solaris) or `C:\opt\pdn\OpenLane` (Windows).

8. Run the `confauth` function of OLCmd from a Console window (Solaris) or MS-DOS prompt (Windows). The confauth function configures OpenLane to run with an authenticated system. For help and a list of command line parameters, run confauth without any parameters.

To configure OpenLane to run with an authenticated system, type:

```
OLCmd confauth on
```

To configure OpenLane to run without an authenticated system, type:

```
OLCmd confauth off
```

NOTE:

If any problems are encountered, refer to the file `/opt/apache/logs/error.log` (Solaris) or `c:\opt\apache\logs\error.log` (Windows) for error information related to the authentication configuration.

Removing a User's Access

To prevent a user's access to the OpenLane system, edit the htpasswd file. Delete the line containing the name of the user to be deleted.

Adding Devices and Customer Profiles

6

Accessing the OpenLane Web Applications

▶ **Procedure**

1. Using your Web browser, enter the URL of the OpenLane system.

    Example: **http://172.24.9.1/OpenLane/**

    ### NOTE:

    You can use the IP address or the host machine name in the URL to access the OpenLane system. OpenLane must be running on the host machine.

2. When the OpenLane login screen appears, enter **Admin** in the Customer Id field to access the system as an administrator.

You are now ready to create additional Customer Profiles and add manageable devices. See the OpenLane online Help for more information.

Adding a Device

▶ **Procedure**

To add a device to the OpenLane System:

1. If you have not already done so, log in to the OpenLane system with the customer ID of:

   Admin

2. On the OpenLane Administration screen, select **Network Diagnostics**.

3. Select the **New Device** button from the bottom frame on the screen.

4. The following New Device screen appears:

```
OpenLane®  Add New Device                              | Help |

        IP Address: [                ]          [     Save     ]

   Read Community: [public          ]          [ Save and Sync ]

  Write Community: [******          ]     Device Sync Options:

        FTP Login: [                ]     ☑ Discover circuits and endpoints

     FTP Password: [                ]     ☐ Resync already discovered endpoints

   Device Group ID: [                ]    ☐ Overwrite circuit names


For DSLAMs Only:
  Enable SCM Polling: ☐


For FrameSavers Only:
       Enable Polling: ☐

      Polling Domain: [default       ]

      Polling Interval: [480] minutes

  Save Report Data For: [100] calendar days

    Save Raw Data For: [7  ] calendar days

      Head End Device: ☐
```

New Device Field Names	Field Description*
IP Address	IP Address of the device being added.
Read Community	Read Community Name must match the Read Community Name set in the device. Default = public.
Write Community	Write Community Name must match the Write Community Name set in the device. Default = public.
FTP Login	FTP Login must match FTP Login in this device. Used by RMON Data Collection and Firmware Download.
FTP Password	FTP Password must match FTP Password in this device. Used by RMON Data Collection and Firmware Download.
Device Group ID	Device Group ID is an administrative name used only in OpenLane to group together certain devices. For example, devices could be grouped by a region name. The OpenLane Network Navigator allows you to then search for devices by ID.
Enable SCM Polling	Enables polling for Hotwire GranDSLAM devices. Used to speed the display of cross-connect information. Default = Disabled.
Enable Polling	Enables RMON Data Collection. Default = Disabled.
Polling Domain	User history polling domain. Should be left as "default."
Polling Interval	Number of minutes between each RMON Data Collection. Default = 480 minutes.
Save Report Data For	Number of calendar days to keep hourly RMON data for this device in the SQL database. Used by the dbage utility. Default = 100 calendar days.
Save Raw Data For	Number of calendar days to keep quarterly-hourly RMON data for this device in the SQL database. Used by the dbage utility. Default = 7 calendar days.
Head End Device	Specifies that this is the head end FrameSaver device in a link.
Save	Adds the new device to the OpenLane Directory Services database.
Save and Sync	Adds the new device to the OpenLane Directory Services database and performs a device synchronization to ensure that the OpenLane directory matches the current configuration of the device. The behavior of the device synchronization can be changed using the following options: **Device Sync Options:** ❏ Discover circuits and endpoints ❏ Resync already discovered endpoints ❏ Overwrite circuit names

* Default values for fields can altered by modifying the file Preferences.dat. See the OpenLane online Help for information.

5. Modify fields as desired. Make sure **Discovered circuits and endpoints** is checked and click on **Save and Sync**.

6. Use the Device Browser or DSLAM Device Display to verify that the device is added properly and endpoints and circuits are discovered. See the OpenLane online Help for more information.

Adding a Customer Profile

▶ **Procedure**

To create a new Customer Profile for Web-based report access:

1. Log in to the OpenLane system with the Customer ID of:

   Admin

2. On the OpenLane Administration screen, select **Customer Profiles**.

3. Under **Select a customer profile**, enter a new Customer ID.

4. Under **Select an action to perform on the customer profile**, select Create a new customer profile.

5. The following form appears:

6. Fill in the customer information in the fields provided. No fields are required, but Customer Name and Account # will not appear on reports if they are left blank here.

7. Under **2. Select a system access level**, specify one of the following:

— **Scheduled Reports Only** – The customer can view the customer's own scheduled FrameSaver SLV reports.

— **Reports Only** – The customer can view the customer's own scheduled FrameSaver SLV reports and DSL inventory reports, and run on-demand reports.

— **Realtime** – The customer can can generate Performance graphs (real-time and historical), generate on-demand reports, view scheduled reports, view Health and Status, view configurations, run nondisruptive tests, and use the Device Browser in read-only mode. Apache user access authentication must be active for read-only restrictions to be enforced on a user with Realtime access. See Chapter 5, *Apache User Access Authentication*.

— **Administrative** – The customer is a system administrator. The Administrative access level gives the user read/write access to all features, including scheduling reports and running disruptive tests, so is not ordinarily suitable for a customer.

— **Network** – The customer can access the Network Navigator and view log files.

— **All Customer Reports** – The customer can view on-demand and scheduled reports for all customer IDs.

8. Click on the **Save** button to save the customer profile.

9. Assign devices to the customer using the **Modify Assigned Devices** link of the Customer Profile Administration screen.

OpenLane® Modify Assigned Devices | **Help** |

Customer Id: Exact_Corporation

Use this page to add or remove devices from the list of authorized devices for this customer. All devices must be added to the OpenLane database before being assigned to a customer. Note that a device may be assigned to more than one customer if desired. A user who logs in as this customer will only be able to view devices that are specifically assigned to this customer profile.

1. Select a device (press Search to search all devices in database):

Enter IP address: [   ]

Or, select from list: [-- Already Assigned Devices -- ▼] (Reload page to update list)

Or, Check here ☐ to use all devices in the database

2. Enter an optional customer visible name for this device:

If left blank, the system name of the device will be used.

Customer device name: [   ]

Or, select from list: [-- Already Assigned Names -- ▼] (Reload page to update list)

3. Select an action to perform:

Assign device
Assigns the device selected above to the authorized device list for this customer. Users who login as this customer will be able to view reports on this device.

OpenLane Maintenance

7

The following tasks are the responsibility of the OpenLane administrator:

Task	Importance	Ideal Frequency
Back up user-modified files. See *Backing Up OpenLane Files* on page 7-2.	Recommended	When you add or modify: ■ Customer profiles or devices ■ Passwords ■ Firmware ■ Report filters and exceptions ■ Web pages
Back up the SQL database. See *Backing Up OpenLane Files* on page 7-2.	Recommended	Daily
Run the **dbage** function of OLCmd. See *Deleting Older Records from the Database* on page 7-5.	Critical	Daily
Run the **processreports** function of OLCmd. See *Generating Scheduled Reports* on page 7-6.	Required for scheduled reports users	Daily
Monitor log files (see *Monitoring Log Files* on page 7-9) and clean them periodically (see *Archiving and Deleting Log Files* on page 7-10): ■ OpenLane ■ Apache ■ Oracle (if used)	Recommended	Ongoing
Monitor disk space usage.	Critical	Ongoing
Upload device configurations as they are changed. See *Uploading Configurations* on page 7-11.	Recommended	Ongoing
Maintain current firmware for all devices. See Chapter 8, *Building and Using a Firmware Library*.	Recommended	Ongoing

Backing Up OpenLane Files

Directories that cannot be restored by reinstalling OpenLane should be backed up periodically for safekeeping. These may include directories listed in Table 7-1.

Table 7-1.   Directories That May Contain User-Modified Files

Directory	Contents
/opt/pdn/OpenLane/data (Solaris) C:\opt\pdn\OpenLane\data (Windows)	Customized default template files, customized default preferences
/opt/pdn/OpenLane/database (Solaris) C:\opt\pdn\OpenLane\database (Windows)	Directory services (LDAP) database (includes device information, report schedules, report filters), Cloudscape SQL database, firmware library, configurations
/opt/apache/.htpasswd (Solaris) C:\opt\apache\.htpasswd (Windows)	Password file
/opt/pdn/OpenLane/web/cgi-bin (Solaris) C:\opt\pdn\OpenLane\web\cgi-bin (Windows)	Customer profiles, reports, report schedules, filters
/opt/pdn/OpenLane/web/htdocs (Solaris) C:\opt\pdn\OpenLane\web\htdocs (Windows)	Customized Web pages, files changed for user authentication
/opt/pdn/OpenLane/web/templates (Solaris) C:\opt\pdn\OpenLane\web\templates (Windows)	Files changed for user authentication
/opt/apache/conf (Solaris) C:\opt\apache\conf (Windows)	Files changed for user authentication and Web server customization
/opt/apache/htdocs (Solaris) C:\opt\apache\htdocs (Windows)	Customized Web pages

Directories shown here are the recommended installation directories for OpenLane and Apache. Replace them with the actual directory names if you have not used the recommended names.

Oracle Database

If you are using Oracle for the performance statistics database, see the vendor's documentation for information about backup and recovery of the database.

Backing Up OpenLane Under Solaris

▶ **Procedure**

To back up files under Solaris:

1. Create a text file (called backup.txt in this example) listing the directories to be backed up. For example:

    `/opt/pdn/OpenLane/data`

    `/opt/pdn/OpenLane/database`

    `/opt/pdn/OpenLane/web/cgi-bin`

    `/opt/pdn/OpenLane/web/htdocs`

2. Verify that you have sufficient space in the backup directory (called /backup_dir in this example) to accommodate a copy of the files.

3. From a command prompt, execute the tar (tape archive) command:

    `tar cvf /backup_dir/backup.tar -l backup.txt`

4. From a command prompt, execute the gzip command:

    `gzip /backup_dir/backup.tar`

    The file **backup.tar.gz** contains a compressed copy of the OpenLane directories.

Restoring OpenLane Under Solaris

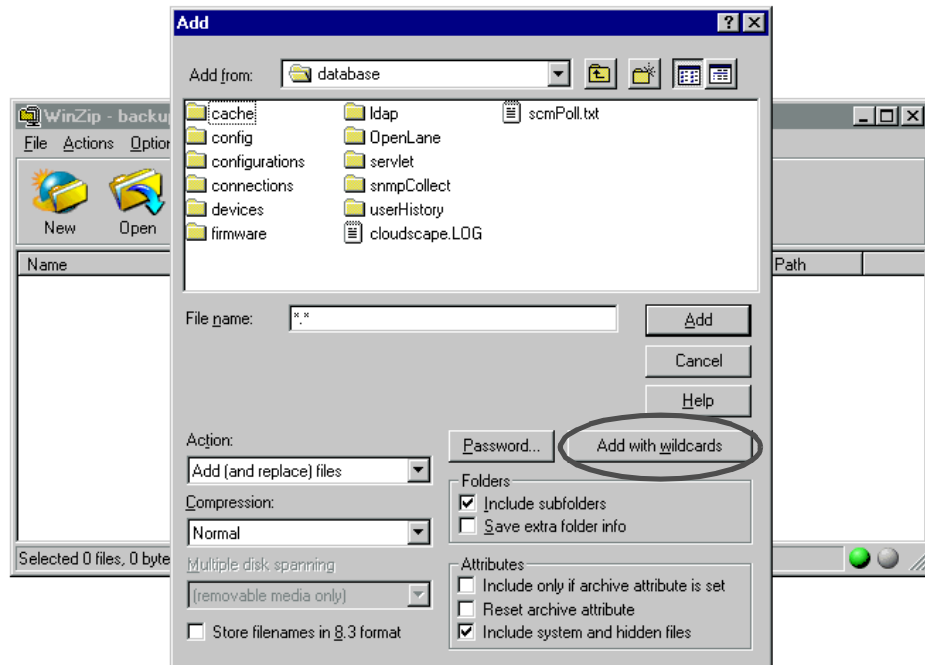▶ **Procedure**

To restore OpenLane under Solaris:

1. Reinstall OpenLane.

2. Execute the commands:

    `gzip -d /backup_dir/backup.tar.gz`

    `tar xvf backup.tar`

Backing Up OpenLane Under Windows

Windows provides no tool for compressing files, but add-on archiving software is readily available. Use it to create a backup of the directories listed in Table 7-1, Directories That May Contain User-Modified Files, that contain altered files.

Using WinZip, for example, you might begin an archive file by selecting all files and subfolders in `\opt\pdn\OpenLane\database`:

Restoring OpenLane Under Windows

▶ **Procedure**

To restore OpenLane under Windows:

1. Reinstall OpenLane.

2. Restore the files backed up with the archiving utility.

Deleting Older Records from the Database

The **dbage** function of OLCmd has four modes:

- The **-d** parameter causes records to be deleted according to individual device definitions (the Save Report Data For and Save Raw Data For values set when each device was added)

- The **-a** parameter causes records to be deleted according to the system-wide preferences set in the preferences.dat file

- The **-rpt** and **-raw** parameters cause report and raw data records to be deleted according to values specified when the dbage function is executed

- The **-dd** parameter deletes data for connections which no longer exist

When the dbage function is run periodically, it should be run twice: once to delete records for nonexistent connections with the -dd (dead data) option, and once to force the deletion of outdated records.

For a complete description of the dbage command, see Appendix A, *OLCmd Reference*.

Frequency

How often dbage needs to be run depends in part on the disk space allotted to the database. In a system collecting data for 15-minute and 24-hour intervals, the database grows per day for each PVC approximately:

Tiny Database (Up to 25 devices and 75 PVCs)	Small Database (Up to 250 devices and 750 PVCs)	Medium Database (Up to 750 devices and 2250 PVCs)	Large Database (Up to 5000 devices and 15,000 PVCs)
76 KB	62 KB	62 KB	61 KB

However, regardless of available space, the database will run more efficiently the more its size is limited.

The running of dbage can be automated using **cron** (under Solaris) or **at** (under Windows). See *Automating the Generation of Scheduled Reports* on page 7-6 for examples.

Example

To delete all dead data records and then delete report and raw data records according to device definitions, enter the commands:

```
OLCmd dbage -n OracleStatsDB -dd
OLCmd dbage -n OraclStatsDB -d
```

Generating Scheduled Reports

To generate scheduled reports, run the **processreports** function of OLCmd.

The processreports application can generate reports based on a date range. This allows, for example, several days' worth of reports to be generated at once. processreports also allows one customer, a list of customers, or all customers to be specified for report generation. For more information, see Appendix A, *OLCmd Reference*.

Automating the Generation of Scheduled Reports

To automate running scheduled reports (and other periodic processes), use the scheduler for the platform where OpenLane was installed. This is typically **cron** for Solaris platforms and **at** for Windows NT. The scheduler runs the ProcessReports command line program at the times or intervals specified by the scheduler.

Using Cron Under Solaris to Automate Reports

▶ **Procedure**

1. Using a text editor, create a shell script in the /opt/pdn/OpenLane directory that contains the commands you would ordinarily use to generate reports. For example, to create reports for all customers, add the following commands:

   ```
   cd /opt/pdn/OpenLane
   ```

   ```
   .OLCmd processreports
   ```

2. Save the script. In this example, the script is called MyReports.sh.

3. Make the file executable:

   ```
   chmod 777 MyReports.sh
   ```

4. Open the crontab for edit:

   ```
   crontab -e
   ```

5. Add a line to the end of the crontab file for the shell script. Each line of the crontab consists of six fields:
   — **Minute** (0–59)
   — **Hour** (0–23)
   — **Day of Month** (1–31)
   — **Month** (1–12)
   — **Day of Week** (0–6, where 0 is Sunday and 6 is Saturday)
   — Command to be executed

   Lists, ranges, and asterisks (meaning all legal values) are allowed.

   For example, to execute MyReports.sh every weekday (1-5) at midnight and noon (0,12), a correct crontab entry is:

   ```
   0 0,12 * * 1-5 /opt/pdn/OpenLane/MyReports.sh
   ```

Using AT under Windows to Automate Reports

▶ **Procedure**

1. Using a text editor, create a batch script in the c:\opt\pdn\OpenLane directory that contains the commands you would ordinarily use to generate reports. For example, to create reports for all customers, add the following commands:

   ```
   c:
   ```

   ```
   cd \opt\pdn\OpenLane
   ```

   ```
   .OLCmd processreports
   ```

2. Save the script. In this example, the script is called Myreports.bat.

3. Open an MS-DOS prompt window. Type the **at** command required to schedule your batch script. The **at** command has the following fields:

   — **\\computername** – A remote host on which the command will be executed (optional).

   — **ID** – An identification number for the scheduled command (optional).

   — **/delete** – Cancels a scheduled command. If id is omitted, all the scheduled commands on the computer are canceled.

   — **/yes** – Used with cancel all jobs command when no further confirmation is desired.

   — **time** – Specifies the time when command is to run.

   — **/interactive** – Allows the job to interact with the desktop of the user who is logged on at the time the job runs.

   — **/every:date[,...]** – Runs the command on each specified day(s) of the week or month. If date is omitted, the current day of the month is assumed.

   — **/next:date[,...]** – Runs the specified command on the next occurrence of the day (for example, next Thursday). If date is omitted, the current day of the month is assumed.

   — **Command** – Is the Windows NT command, or batch program to be run.

   For example, to execute MyReports.sh every weekday at midnight and noon on the same computer as the AT command is executed, the correct commands are:

   ```
   at 00:00 /every:Monday,Tuesday,Wednesday,Thursday,Friday
   c:\opt\pdn\OpenLane\Myreports.bat
   ```

   ```
   at 12:00 /every:Monday,Tuesday,Wednesday,Thursday,Friday
   c:\opt\pdn\OpenLane\Myreports.bat
   ```

Rebuilding Oracle Database Indexes

As records are added to and deleted from the database, indexes may become fragmented and index trees may increase in height, slowing access to the database.

The **ol_rbldidx.sql** script rebuilds indexes to maintain performance.

Indexes should be rebuilt weekly, or, if less frequently, as often as practicable.

▶ **Procedure**

To rebuild the indexes of the database named OpenLane every Sunday morning starting at 12:15 am on a Solaris system:

1. Move **/opt/pdn/OpenLane/data/sql/scripts/Oracle/ Maintenance/ol_rbldidx.sql** to the $ORACLE_HOME directory.

2. Create a shell script:

```
ORACLE_HOME=/opt/oracle;
export ORACLE_HOME;
ORACLE_SID=OLDB;
export ORACLE_SID;
/opt/oracle/product/8.1.6/bin/sqlplus
 my_username/my_password @ol_rbldidx
```

   For this example, presume the script is saved as:
   /OL_Scripts/rebuild_ix

3. Create or a modify a crontab file with the entry:

```
15 0 * * sunday "/OL_Scripts/rebuild_ix"
```

   For this example, presume the script is saved as:
   /OL_Scripts/my_cronfile

4. Submit the crontab file to cron:

```
crontab /OL_Scripts/my_cronfile
```

Monitoring Log Files

The following log files can be viewed from the View Log Files screen, or by running the **viewlog** function of OLCmd.

- Default Polling Domain
- Additional Polling Domains (if any exist)
- Firmware/Configuration/Feature Maintenance
- Database Reader
- Poller Controller
- On-Demand SLV Reports
- Scheduled SLV Reports
- Database Aging (dbage)
- Database Roll-up (dbroll)
- Alarm Configuration
- Device Sync
- Configuration/Health and Status
- Device Tests
- Exception Periods
- Customer Profile Administration
- System Access (logins and logouts)
- PVC Maintenance Periods

Other OpenLane log files may be created in **/opt/pdn/OpenLane/logs** (Solaris) or **\opt\pdn\OpenLane\logs** (Windows), and can be displayed using any text editor.

Apache Log Files

Apache log files can be found at:

    Solaris: **/opt/apache/logs/access.log**
               /opt/apache/logs/error.log

    Windows: **c:\opt\apache\logs\access.log**
               c:\opt\apache\logs\error.log

SQL Database Log File

If you use a Cloudscape database, the following log file is created:

    Solaris: **/opt/pdn/OpenLane/database/cloudscape.LOG**

    Windows: **c:\opt\pdn\OpenLane\database\cloudscape.LOG**

For information about log files created for an Oracle database, see the *OpenLane SLM 5.5 Oracle Database Administration Instructions*.

Legacy Logger Log Files

These files are created if **pdn.uhpoller.legacyLogger** is enabled (set to true) in **preferences.dat**:

Solaris: `/opt/pdn/OpenLane/logs/UHReader.log`
`/opt/pdn/OpenLane/logs/UHPoller_default.log`

Windows: `c:\opt\pdn\OpenLane\logs\UHReader.log`
`c:\opt\pdn\OpenLane\logs\UHPoller_default.log`

Archiving and Deleting Log Files

The **clean** function of OLCmd renames log files and deletes log files older than a specified number of days. Periodic use of the command keeps your log files at a manageable size.

For example, the command:

```
OLCmd clean logs .log 14
```

(executed in the OpenLane home directory) performs two functions. It:

- Renames any file in the OpenLane logs directory that has a file extension of **.log** to a file extension of .log_*yyyy_mm_dd*.arc (where *yyyy_mm_dd* is the current date)

- Deletes any file in the OpenLane logs directory that has a file extension of **.arc** and was created more than 14 days ago

For more information about the clean function, see Appendix A, *OLCmd Reference*. The running of the clean function can be automated using **cron** (under Solaris) or **at** (under Windows). See *Automating the Generation of Scheduled Reports* on page 7-6 for examples.

Uploading Configurations

You can maintain a library of configuration files so that in the event a device must be replaced, its configuration can be loaded into the new hardware quickly and without discrepancy. Use the Schedule Configuration Upload screen to schedule or immediately perform configuration uploads from devices in the network to OpenLane's configuration database.

After you log in as an administrator, the OpenLane Administration Main Menu appears. Select Firmware/Configuration/Feature Maintenance. The Firmware/Configuration/Feature Maintenance screen appears. Click on **Schedule configuration uploads**.

▶ **Procedure**

To schedule or perform configuration uploads:

1. Select a device or group of devices by model, name, or IP address.

2. Click on **Click here to search devices and refresh the lower frame**. Devices selected according to your criterion are listed in the lower frame.

3. If there are devices listed in the lower frame that you do not want to be part of the upload operation, click once in the Upload check box for that device to omit it.

4. Specify when the upload should occur. Optionally, specify that the upload will be repeated periodically.

5. Optionally, specify whether the upload operation is to be stopped if it does not finish within a certain time period.

6. Click on **Click here to schedule the tasks**.

See the online Help for detailed information.

Configuration files are stored in the configurations directory:

    Solaris: `/opt/pdn/OpenLane/database/configurations`

    Windows: `\opt\pdn\openlane\database\configurations`

Under the configurations directory, subdirectories are named for the IP addresses of the devices whose configurations you have saved.

Like all OpenLane files, configurations should periodically be backed up for safekeeping. See *Backing Up OpenLane Files* on page 7-2.

Building and Using a Firmware Library

8

Building a Firmware Library

OpenLane allows you to maintain a firmware library from which you can download firmware to devices in your network.

Before the Download Feature can schedule or activate firmware download to a device, you must retrieve the firmware files from the Paradyne Web site.

▶ **Procedure**

1. Go to Paradyne's Technical Support site to obtain access to firmware files: **www.paradyne.com/tech_support**.

2. Download the appropriate firmware file or files.

   If the firmware file is a .exe file, it is a self-extracting zip file. Unarchive it on a Windows platform and, if it will reside on a Solaris platform, use FTP to move it.

3. Move the files to the subdirectory under **/opt/pdn/OpenLane/database/firmware** (Solaris) or **C:\opt\pdn\OpenLane\database\firmware** (Windows) for the appropriate model.

   — **.OCD** files are for FrameSaver products

   — **.IMG** and **.FPI** files are for DSL products

4. Map the firmware version to the firmware filename by editing the firmware.dat file in the selected model's directory. The syntax is:

   version=filename

   FrameSaver Map Example:

   01.02.01=R010201.ocd

   DSL Map Example:

   04.00.30=pc8314.3001.img

New Device Models Not Supported by OpenLane

You can create your own subdirectory and definition files for Paradyne devices not yet supported by OpenLane. The device model number will appear in the drop-down Device Model selection list for scheduling downloads.

▶ **Procedure**

1. Create a directory under `/opt/pdn/OpenLane/database/firmware` (Solaris) or `c:\opt\pdn\OpenLane\database\firmware` (Windows) for the device model. Start the directory name with **fw** and follow with the device model number. For example:

   `mkdir fw9880`

2. Retrieve the firmware as described in *Building a Firmware Library* on page 8-1.

3. Copy any firmware.dat file into the directory you just created in Step 1 and modify the file. The syntax is:

   `version=filename`

   FrameSaver Map Example:

   `01.02.01=R010201.ocd`

   DSL Map Example:

   `04.00.30=pc8314.3001.img`

4. Edit the `/opt/pdn/OpenLane/database/firmware/models.dat` (Solaris) or `c:\opt\pdn\OpenLane\database\firmware\models.dat` (Windows) file to map the device model number to the new directory name. The syntax is:

   `model=directory`

   Example:

   `9880=fw9880`

   You can determine the correct device model number from the OpenLane Network Navigator screen.

5. Edit the `/opt/pdn/OpenLane/database/firmware/mtypes.dat` (Solaris) or `c:\opt\pdn\OpenLane\database\firmware\mtypes.dat` (Windows) file to map the device model number to the device type. The syntax is:

   `model=devicetype`

where devicetype is one of:

      `dbm`   – Dial Backup Module
      `dslaan` – GranDSLAM Access Node (also used for older DSL)
      `dslasn` – Service Node
      `dsldsn` – 5620/6310
      `dslman` – TDM Access Node
      `dslmcc` – Management Communications Controller (MCC)
      `dslmsn` – TDM Service Node
      `dslscm` – Shelf Concentration Module (SCM)
      `dslssn` – SNMP-Managed Service Node
      `fs`   – FrameSaver
      `fsflex` – FrameSaver FLEX

Example:
`9880=fs`

Downloading and Activating Firmware

Use the Schedule Downloads and Firmware Activations screen to administer firmware for selected devices.

▶ **Procedure**

1. Select **Firmware/Configuration/Feature Maintenance** from the OpenLane Home Page. The Firmware/Configuration/Feature Maintenance screen appears. Select **Schedule firmware downloads/activations**. The Schedule Downloads and Firmware Activations screen appears.

2. Select one or more devices of the same model or enter a device's IP address.

3. Click on **Click here to search devices and refresh the lower frame**. The lower frame displays the selected devices. Verify that the first column titled Selected includes a checkmark for each device requiring a firmware update. If desired, select a **Download Firmware** version from the drop-down list for the device.

4. Select **Make the firmware active (this causes a device reset)** if you would like to activate the firmware immediately. The Alternate Firmware becomes the Active Firmware, followed by a device reset. If do not want to activate the new firmware yet, uncheck this option.

Prior to activating firmware, decide which options should be performed:

— RMON data collection (for SLV Reports)

— Device Sync

Both options are recommended.

5. Select when the downloads/activations should be scheduled.

— **Begin the tasks immediately**. The Alternate Firmware becomes the Active Firmware, followed by a Device Reset.

— **Begin the tasks using the following start time**. This schedules firmware activation anytime in the next seven days.

For either option (Begin the tasks immediately or Scheduling tasks), an amount of time can be specified for the maintenance window. If the firmware download process has not started within the specified time set in the maintenance window, the firmware downloads/activations must be rescheduled.

The download server will initiate up to ten FTPs to process ten tasks concurrently. Any task in process will continue until complete, regardless of the maintenance window scheduled start time (the end time is not specified).

6. Click on **Click here to schedule the tasks**. A verification page displays all choices from the previous screen for review and confirmation. All devices selected for firmware download will have a checkmark in the first column of the lower frame.

Do not use the browser's Back button. Instead, use the Back button at the bottom of the activation page or the **Click here to go back and reselect** option to modify any values. To approve the displayed data, select **Apply**. You are returned to the OpenLane main menu.

OLCmd Reference

A

OLCmd (OLCmd.bat in Windows) is the command line interface for OpenLane. OLCmd commands include:

- **OLCmd addcust** on page A-3 – Adds customers to the OpenLane LDAP database.

- **OLCmd assigndev** on page A-5 – Assigns devices to customers.

- **OLCmd chgrawfileext** on page A-6 – Renames raw data files.

- **OLCmd clean** on page A-7 – Creates and deletes archived log files.

- **OLCmd confauth** on page A-8 – Turns user access authentication on and off.

- **OLCmd dbage** on page A-9 – Deletes outdated user history.

- **OLCmd dbconfig** on page A-10 – Configures the naming/directory service description of the user history database.

- **OLCmd dbcopy** on page A-12 – Copies data from one database instance to another.

- **OLCmd dbip** on page A-13 – Translates IP addresses from a string to an integer and vice versa.

- **OLCmd dbroll** on page A-14 – Rolls up raw history data into hourly data.

- **OLCmd delcust** on page A-15 – Deletes a customer or a list of customers from the database.

- **OLCmd deldev** on page A-16 – Deletes a device from the LDAP database.

- **OLCmd dumpldap** on page A-17 – Creates an LDIF file from the LDAP database.

- **OLCmd importldif** on page A-18 – Adds data to the LDAP database from an LDIF file.

- **OLCmd key** on page A-19 – Sets or tests the OpenLane license key.

- **OLCmd ldapmodify** on page A-20 – Adds or modifies entries in the LDAP database.

- **OLCmd ldapsearch** on page A-21 – Searches the LDAP database.

- **OLCmd logsvc** on page A-23 – Configures the logger for distributed logging.

- **OLCmd processreports** on page A-24 – Generates scheduled reports.

- **OLCmd snmpget** on page A-25 – Gets an SNMP object.

- **OLCmd snmptest** on page A-26 – Tests SNMP connection.

- **OLCmd sync** on page A-27 – Synchronizes a device.

- **OLCmd uhread** on page A-29 – Starts and stops UHReader.

- **OLCmd unassigndev** on page A-30 – Removes a device from a customer record.

- **OLCmd viewlog** on page A-31 – Displays OpenLane log files.

- **OLCmd xmlreport** on page A-33 – Generates OpenLane reports in XML format.

OLCmd must be executed in the OpenLane home directory. For example (in Windows):

```
cd \opt\pdn\openlane
olcmd confauth on
```

In Windows, OLCmd can be invoked by typing `OLCmd, olcmd, OLCmd.bat`, or `olcmd.bat`. In Solaris, only `OLCmd` is valid.

OLCmd Helps

All OLCmd commands accept the **-h** parameter in lieu of other parameters. This causes usage notes for the command to be displayed. For example, the following command causes the available parameters of the viewlog function to be displayed:

```
olcmd viewlog -h
```

OLCmd addcust

The **addcust** command adds customers to the LDAP database.

Format:

`OLCmd addcust -a` *access_level* [[`-c` *customer_id*] | [`-l` *customer_list*]]
[`-o` *customer_details*] [`-t` *template*]

Where:

`-a` *access_level*	Is the access level of the customer or customers to be added. It may be one of:

- **admin** – The customer has full administrator rights.
- **realtime** – The customer may view on-demand and scheduled reports, generate performance graphs, view device configuration, run nondisruptive tests, and use the Device Browser in read-only mode.
- **reports** – The customer may view on-demand and scheduled reports.
- **schedreports** – The customer may view scheduled reports on devices.
- **network** – The customer may use the Network Navigator and view log files.
- **allcustreports** – The customer may view all on-demand and scheduled reports for all customers.

`-c` *customer_id*	Is the identifier of a customer to be added to the LDAP database. Customer ID can contain no spaces.
`-l` *customer_list*	Is a text file containing a list of customer names separated by carriage returns.
`-o` *customer_details*	Is a text file containing customer information. The file must consist of eight lines, which are taken to be:

— Customer Name
— Account Number
— Contact Name
— Phone
— Fax
— E-Mail Address
— Street Address
— Comments

Any line may be blank.

`-t` *template*	Is the name of an existing customer record to be used as a template for the customer being added.

Examples:

The following command adds a single customer to the LDAP database with a Realtime access level:

```
OLCmd addcust -a realtime -c Exact_Corporation
```

The file customer_list.txt contains several customer IDs on separate lines:

```
Rent-a-Phone.com
Rent-a-Phone_GmbH
Rent-a-Phone_USA
```

The following command adds the list of customers to the database with the same characteristics as those of the customer Exact_Corporation:

```
OLCmd addcust -a realtime -l customer_list.txt
 -t Exact_Corporation
```

The file customer_data.txt contains details about the customer Exact_Corp:

```
Exact Corporation
261800-1023-9
Priscilla Eiss
800-727-2396
727-530-2000
priseiss@exact.com
123 Punctilious Rd., Exacton, NY
Office closed Tuesdays
```

The following command adds the customer Exact_Corporation to the database using the details in customer_data.txt:

```
OLCmd addcust -a realtime -c Exact_Corporation
 -o customer_data.txt
```

OLCmd assigndev

The **assigndev** command assigns devices to a customer ID.

Format:

`OLCmd assigndev -c` *customer_name* [`-a` *device_address*] | [`-l` *device_list*]

Where:

`-c` *customer_name*	Is the customer name to be acted upon.
`-a` *device_address*	Is the IP address of a device to be assigned.
`-l` *device_list*	Is a text file containing a list of device IP addresses separated by carriage returns. The group of devices is assigned to the customer ID.

Example:

The following command adds a single device to customer Exact_Corporation:

`OLCmd assigndev -c Exact_Corporation -a 137.16.255.5`

The following command assigns a group of devices, which are listed in the file exact_devices.txt:

`OLCmd assigndev -c Exact_Corporation -l exact_devices.txt`

OLCmd chgrawfileext

The **chgrawfileext** command changes the file extensions of raw data files on seleted polling domains.

Format:

```
OLCmd chgrawfileext [-d polling_domain ] [-o current_extension ]
[-e new_extension ]
```

Where:

-d *polling_domain*	Is the polling domain where file extensions are to be changed. If -d is not specified, all polling domains are affected.
-o *current_extension*	Is the current file extension of files to be renamed. Valid extensions are bad, ftp, and raw.
-e *new_extension*	Is the new file extension of files to be renamed. Valid extensions are bad, ftp, and raw.

Example:

The following command renames all .raw files on all polling domains to .ftp.

```
OLCmd chgrawfileext -o raw -e ftp
```

The following command renames .bad files on the polling domain named dallas to .raw.

```
OLCmd chgrawfileext -d dallas -o bad -e raw
```

OLCmd clean

The **clean** command creates archive copies of specified log files and deletes archived files older than a specified age.

Format:

OLCmd clean *directory extension days* [*new-extension*]

Where:

directory	Is the directory where the log files reside.
extension	Is the suffix of the log files to be archived.
days	Is the number of days old existing archive files must be for the clean command to delete them.
new-extension	Is an optional name to be added to the suffix of archived files. It is also used to identify files to be deleted. If new-extension is not specified, an extension of .arc is presumed.

Example:

The following command moves files in the \opt\pdn\openlane\logs directory named *.log to *.log_<date>.arc (for example, database.log_2001_05_13.arc) and deletes any *.arc files older than 7 days.

OLCmd clean logs .log 7

The following command moves files in the \opt\apache\logs directory named *.log to *.log_<date>admin (for example, jserv.log_2001_05_13admin) and deletes any *.log_<date>admin files older than 21 days.

OLCmd clean \opt\apache\logs .log 21 admin

OLCmd confauth

The **confauth** command configures OpenLane for use in a system with Apache user access authentication.

Format:

OLCmd confauth on | off

Where:

on	Specifies that Apache user access authentication is in use.
off	Specifies that Apache user access authentication is not in use.

Example:

To configure OpenLane for use in a system with Apache user access authentication:

OLCmd confauth on

OLCmd dbage

The **dbage** command removes older records from the performance statistics database.

Format:

```
OLCmd dbage [-n instance] [[ -d | -a ] | [[ -raw days ] [ -rpt days ]] | -dd]
```

Where:

-n *instance*	Specifies the database name. If -n is omitted, the default database is used.
-d	Removes records from the database by device according to the parameters selected when each device was added.
-a	Removes records from the database using the number of days in the system-wide preferences (preferences.dat).
-raw *days*	Optionally specifies the age in days of the oldest Raw (quarter-hourly user history) data record to be retained. This overrides the Save Raw Data For value entered when each device was added to the database. If 0 is specified, all Raw data records in the database are deleted.
-rpt *days*	Optionally specifies the age in days of the oldest Report (hourly user history) data record to be retained. This overrides the Save Report Data For value entered when each device was added to the database. If 0 is specified, all Report data records in the database are deleted.
-dd	Optionally deletes data for connections which no longer exist. The -dd parameter may not be specified with the -d, -a, -raw, or -rpt parameters.
	A connection is determined to be nonexistent when an interface appears in the user history database but not in the LDAP database. If the -dd option is never used, user history records for nonexistent connections are eventually deleted like other records, according to age.

Example:

To delete Report data records older than 90 days and Raw data records older than 7 days, enter the command:

```
OLCmd dbage -n OracleStatsDB -raw 7 -rpt 90
```

To delete all records for nonexistent connections, then delete Report and Raw data records according to individual device definitions, enter the commands:

```
OLCmd dbage -n OracleStatsDB -dd
OLCmd dbage -n OracleStatsDB -d
```

OLCmd dbconfig

Information that OpenLane uses to connect to a database is stored in a naming/directory service. The **dbconfig** command creates or modifies the naming/directory service database entry that points to the performance statistics database.

Format:

Solaris:
```
OLCmd dbconfig '[[-d c|o]|[-a instance]]|[-l]|[-h]|
[[-c new_instance]][-m host][-n dbname][-u user][-p password]
[Advanced Parameters]'
```

Windows:
```
OLCmd dbconfig "[[-d c|o]|[-a instance]]|[-l]|[-h]|
[[-c new_instance]][-m host][-n dbname][-u user][-p password]
[Advanced Parameters]"
```

Parameters may be set to null by specifying null. Example: **-u null**.

Where:

-d c|o	Sets the default database to Cloudscape (c) or Oracle (o).
-a *instance*	Alters the options specified on a specific database instance. If -a is omitted, the default database is assumed.
-c *new-instance*	Creates a new database instance. If the -c parameter is not followed by a -d parameter, the database type is derived from the default instance.
-m *host*	Sets the host machine name or IP address (and port if needed) where the database is located. The default is localhost:1521.
-n *dbname*	Specifies the database name (SID in Oracle). The default is OpenLane for Cloudscape or OLDB for Oracle.
-u *user*	Specifies the database user. The default is empty for Cloudscape or pdyn for Oracle.
-p *password*	Specifies the database user password. The default is empty for Cloudscape or pdyn for Oracle.
-l	Lists all the database instances in the naming/directory service.
-h	Prints usage notes similar to those shown here.

Advanced Parameters:

-url	Specifies the database connection URL in raw form. It may contain $host$, which gets replaced by the host, and $dbname$, which gets replaced by the database name when the connection is made.
-dr *driver*	Specifies the name of the JDBC driver from the database vendor. It must be in the classpath for connections to work.
-err *number*	Specifies the vendor-specific JDBC error number to ignore for duplicate keys. If not set or correct, duplicate key warnings are logged.
-sm *statmap*	Specifies the filename of file that maps internal statistic names to the database schema.
-rs *rawsql*	Specifies the filename of file that contains raw SQL for the database schema/vendor.
-cm *calcmap*	Specifies the filename of file that defines and maps calculations to the database schema.
-om *oidmap*	Specifies the filename of file that maps MIB OIDs to the database schema.
-fc *feedclass*	Specifies the feed class name for feeding user history statistics to a database.
-ac *ageclass*	Specifies the age class name for aging user history statistics in a database.

Example:

To change the default database to Oracle and set the database parameters, enter the command:

```
OLCmd dbconfig '-d o -m mycomputer.mycompany.com:1521 -n mydb
-u myself -p mypassword'
```

OLCmd dbcopy

The **dbcopy** command copies all the data in one database instance defined in the naming/directory service to another database instance. It can be used, for example, to convert an outgrown Cloudscape database to an Oracle database.

Format:

`OLCmd dbcopy` [*from_instance* | `d`] [*to_instance* | `d`] [*start_time end_time*]

Where:

from_instance	Specifies the database instance in the naming/directory service to copy data from.
to_instance	Specifies the database instance in the naming/directory service to copy data to.
`d`	Specifies the default database instance. It can be placed in the source or target position.
start_time	Optionally specifies the start time of statistics to be copied in the form YYYYMMDDhhmmss. Example: 20011101235959 denotes one second before midnight (23:59:59) on November 30, 2001.
end_time	Optionally specifies the end time of statistics to be copied in the form YYYYMMDDhhmmss. Example: 20020101000000 denotes midnight (00:00:00) on January 1, 2002.

Example:

To copy the default database instance to an Oracle database (created with the name OracleStatsDB using the dbconfig command), enter the command:

`OLCmd dbcopy d OracleStatsDB`

OLCmd dbip

IP addresses are maintained in the statistics database as single decimal integers. For example, the IP address 135.26.10.37 is maintained in the database as −2028336603. The **dbip** command converts a single integer into a dotted decimal address (four decimal integers separated by periods), or a dotted decimal address into a single integer.

Format:

```
OLCmd dbip address
```

Where:

address	May be a positive or negative decimal integer, or four decimal integers from 0–255 separated by periods. A negative number is denoted by a leading hyphen.

Example:

The following are valid uses of the dbip command (shown with responses):

```
OLCmd dbip -2023301243
-2023301243 = 135.102.223.133
OLCmd dbip 174195980
174195980 = 10.98.5.12
OLCmd dbip 200.23.132.10
200.23.132.10 = 937982966
```

OLCmd dbroll

User history statistics are stored by 15-minute intervals. To speed up the processing of SLV reports, and to reduce the size of the database, you can roll up statistics into hourly intervals by using the **dbroll** function of OLCmd.

Format:

OLCmd dbroll [**-d** *database_instance*] [**-s** *start_time*] [**-e** *end_time*]
[**-i** *ip_address*] [**-r**] [**-h**]

Where:

-d *database_instance*	Optionally specifies the user history database to roll up. If no database is specified, the default database is used.
-s *start_time*	Optionally specifies the start time of statistics to be rolled up, in the form YYYYMMDDhhmmss. Example: 20011012235959 denotes one second before midnight (23:59:59) on October 12, 2001. If not specified, roll-up begins at the beginning of the user history database.
-e *end_time*	Optionally specifies the end time of statistics to be rolled up, in the form YYYYMMDDhhmmss. Example: 20020101000000 denotes midnight (00:00:00) on January 1, 2002. If not specified, roll-up ends at the end of the user history database.
-i *ip_address*	Optionally specifies the IP address of a device whose statistics are to be rolled up. If not specified, statistics for all devices are rolled up.
-r	If specified, statistics that are successfully rolled up are deleted from the database.
-h	Prints usage notes similar to those shown here.

Example:

To roll up and delete 15-minute buckets created before January 2001 for all devices, enter the command:

OLCmd dbroll '-e 20010101000000 -r'

OLCmd delcust

The **delcust** command allows you to delete a customer or a list of customers from the database.

Format:

OLCmd delcust [**-c** *customer*] | [**-l** *customer-list*]

Where:

-c *customer*	Is the customer ID of a single customer to be deleted from the database.
-l *customer-list*	Is the pathname of a file containing customer IDs separated by carriage returns. All the customer IDs in the file are deleted from the database.

Example:

The file ex-customers.txt contains customer IDs on separate lines:

```
Rent-a-Phone.com
Rent-a-Phone_GmbH
Rent-a-Phone_USA
```

The following command deletes from the database all the customer IDs in the ex-customer.txt file:

```
OLCmd delcust -l ex-customers.txt
```

OLCmd deldev

The **deldev** command allows you to delete a device and all associated circuits from the database.

Format:

```
OLCmd deldev ip_address
```

Where:

    ip_address   Is the IP address of the device to be deleted from the database.

Example:

The following command deletes the device whose IP address is 135.26.10.37 and all its circuits.

```
OLCmd deldev 135.26.10.37
```

OLCmd dumpldap

The **dumpldap** command creates an LDAP Data Interchange Format (LDIF) file from the LDAP database. The file may be used for export to a different directory server system.

The file created is written to standard output.

The dumpldap command also uses the environmental variable *LDAP*, located in the file OLControlConfig (Solaris) or OLControlConfig.bat (Windows), to identify the socket used to connect to the LDAP database.

Format:

```
OLCmd dumpldap host
```

Where:

> *host*   Is the name or IP address of the OpenLane management server.

Example:

The following command dumps the LDAP database on olhost.exactcorp.com to an LDIF file.

```
OLCmd dumpldap olhost.exactcorp.com > backups/060501.ldif
```

OLCmd importldif

The **importldif** command adds entries to an LDAP database based on an LDAP Data Interchange Format (LDIF) file.

The importldif command also uses the environmental variable *LDAP*, located in the file OLControlConfig (Solaris) or OLControlConfig.bat (Windows), to identify the socket used to connect to the LDAP database.

Format:

OLCmd importldif [**-h** *host*] *ldif_file*

Where:

> *host*　　　　　　Is the name or IP address of the OpenLane management server. This parameter is optional.
>
> *ldif_file*　　　　Is the pathname of the LDIF file to be imported. This parameter is required.

Example:

The following command imports data from an LDIF file to the LDAP database on olhost.exactcorp.com.

OLCmd importldif -h olhost.exactcorp.com openldap/new.ldif

OLCmd key

The **key** command sets the OpenLane license key, or tests the current key to determine its expiration date.

Format:

`OLCmd key` [*license_key*]

Where:

> *license_key*   Is the new license key to be applied to OpenLane. If the key is omitted, the current key is tested and its expiration date is displayed.

Example:

The following command displays the expiration date of the OpenLane license key:

`OLCmd key`

The following command sets the OpenLane license key to WZBCK-AAM6E-U898Q-MX5HK:

`OLCmd key WZBCK-AAM6E-U898Q-MX5HK`

OLCmd ldapmodify

The **ldapmodify** command adds or modifes entries in the LDAP database. It accepts input from the STDIN stream.

Format:

```
OLCmd ldapmodify [-D bindDN] [-w password] [-h host] [-p port] [-F]
[-a | -r] [-b] [-c] [-v]
```

Where:

-D *bindDN*	Is the Distinguished Name (DN) used to bind to the directory.
-w *password*	Is the database password.
-h *host*	Is the name or IP address of the OpenLane management server.
-p *port*	Is the port used to connect to the LDAP database.
-F	Specifies that all changes will be forced, even when input lines begin with **replica:**.
-a	Specifies that data is to be added. If neither **-a** nor **-r** is specified, **-r** is presumed.
-r	Specifies data is to be replaced. If neither **-a** nor **-r** is specified, **-r** is presumed.
-b	Specifies that any value starting with a forward slash (/) is the pathname of a binary file.
-c	Specifies that the command should continue execution even after an error.
-v	Specifies that debugging information should be returned.

Example:

The file new.ldif contains data for new database entries, in LDAP Data Interchange Format (LDIF). The following command adds the entries to the LDAP database:

```
OLCmd ldapmodify -D c=US -w admin -a -c < data/new.ldif
```

OLCmd ldapsearch

The **ldapsearch** command retrieves data from the LDAP database based on search criteria.

Format:

OLCmd ldapsearch [**-n**] [**-D** *bindDN*] [**-w** *password*] [**-h** *host*] [**-p** *port*] [**-b** *searchbase*] [**-F** *separator*] [**-s** *sort_attrib*] [**-l** *time_limit*] [**-z** *size_limit*] [**-u**] [**-t**] [**-B**] [**-A**] [**-L**] [**-v**] [**-d** *debug_level*]

Where:

-n	Specifies that the command will return information on what would be changed if executed, but does not modify entries.
-D *bindDN*	Is the Distinguished Name (DN) used to bind to the directory.
-w *password*	Is the database password.
-h *host*	Is the name or IP address of the OpenLane management server.
-p *port*	Is the port used to connect to the LDAP database.
-b *searchbase*	Is an attribute value denoting the starting point of the search.
-F *separator*	Is the character used as a separator between attribute names and values. The default is an equal sign (=). This parameter is ignored if **-L** is specified.
-s *sort_attrib*	Is the attribute used to sort the entries returned.
-l *time_limit*	Is the amount of time which may elapse before the search is abandoned.
-z *size_limit*	Is the size of the output (in bytes) which may be produced before the search is abandoned.
-u	Specifies that the output should include the human-readable form of the Distinguished Name.
-t	Specifies that retrieved values should be written to temporary files.
-B	Specifies that non-ASCII values should not be suppressed.
-A	Specifies that only attribute names (not values) should be retrieved.
-L	Specifies that search results should be displayed in LDAP Data Interchange Format (LDIF).

-v	Specifies that debugging information should be returned.
-d *debug_level*	Specifies the level of detail of debugging information that should be returned in the event of an error. Possible values are 1, 2, 3, and 4, with 1 returning the least information and 4 the most.

Example:

The following command exports the entire database to an LDIF file:

```
OLCmd ldapsearch -D c=US -w admin -b c=US -L objectclass=* >
data/alldata.ldif
```

OLCmd logsvc

The **logsvc** command sets the IP address of the logger when distributed logging is used.

Format:

```
OLCmd logsvc ip_address
```

Where:

> *ip_address*   Is the address of the host that will service logging requests.

Example:

The following command sets the IP address of the logger:

```
OLCmd logsvc 135.26.10.37
```

OLCmd processreports

The **processreports** command generates scheduled reports for specified customers and dates.

Format:

```
OLCmd processreports [-date date] [-enddate enddate]
[-customer customer_id] [-customerlist list_file]
```

Where:

-date *date*	Is the optional starting date of the reports. If -date is not specified, the current date is used.
-enddate *enddate*	Is the optional end date of the reports, creating a date range between -date and -enddate. If -enddate is not specified, there is no date range.
-customer *customer_id*	Is the optional customer for whom reports are to be generated. If neither -customer nor -customerlist is specified, scheduled reports for all customers are generated.
-customerlist *list_file*	Is the optional list of customers for whom reports are to be generated. *list_file* is the pathname of a text file with one customer ID per line. Both -customer and -customerlist may be specified. If neither -customer nor -customerlist is specified, scheduled reports for all customers are generated.

Example:

The following command generates scheduled reports for all customers:

```
OLCmd processreports
```

The following command generates scheduled reports for customer Exact_Corp using 13 May 2001 as a start date:

```
OLCmd processreports -date 05/13/2001 -customer Exact_Corp
```

The following command generates scheduled reports for a list of customers in a file called la_west.txt:

```
OLCmd processreports -customerlist la_west.txt
```

OLCmd snmpget

The **snmpget** command performs an SNMP get on a specified MIB object.

Format:

```
OLCmd snmpget [community_name] ip_address object_id
```

Where:

community_name	Is the community name for read access. If omitted, **public** is used.
ip_address	Is the address of the device to be accessed.
object_id	Is the object to be displayed.

Example:

The following command performs an SNMP get and displays the sysDescr MIB object:

```
OLCmd snmpget 135.26.10.37 1.3.6.1.2.1.1.1.0
```

OLCmd snmptest

The **snmptest** command tests SNMP connectivity using a specified community name. If successful, snmptest displays sysDescr, sysName, sysContact, and sysLocation.

Format:

OLCmd snmptest *ip_address* [*community_name*]

Where:

ip_address	Is the address of the device to be accessed.
community_name	Is the community name to be used. If omitted, **public** is used.

Example:

The following command tests SNMP connectivity using a community name of nmswrite:

OLCmd snmptest 135.26.10.37 nmswrite

OLCmd sync

The **sync** command synchronizes a specified device and the database. If the device is not in the database, it is added.

Format:

OLCmd sync [[**-v**] [**-c** *community-string*] [**-w** *community-string*]
[[**-i** *ftp_user*] [**-p** *ftp_passwd*]] [**-g** *groupid*] [**-s**] [**-f**] [**-d** *domain*]
[**-I** *minutes*] [**-k** *days*] [**-r** *days*] [**-h**] [**-A**] [**-S**] [**-E**] [**-O**]
[*ip_address* | *hostname*]]

Where:

ip_address	Is the address of the device to be synchronized.
hostname	Is the host name of the device to be synchronized.
-v	Specifies that debugging information should be returned.
-c *community-string*	Sets the name of the read community string for the device.
-w *community-string*	Sets the name of the write community string for the device.
-i *ftp_user*	Specifies the user ID for the FTP session used to collect performance statistics.
-p *ftp_passwd*	Specifies the password for an FTP session initiated by ftp_user.
-g *groupid*	Specifies the group ID for the device.
-s	Enables SCM polling (for DSLAM devices only).
-f	Enables user history polling (for FrameSaver devices only).
-d *domain*	Specifies the domain where the poller/reader resides.
-I *minutes*	Specifies the number of minutes between polls. This should be less than 1440 (24 hours) to ensure that no data is lost.
-k *days*	Specifies the number of days' worth of hourly report data that should be maintained for the device.
-r *days*	Specifies the number of days' worth of quarter-hourly report data that should be maintained for the device.
-h	Specifies that the device is a head-end FrameSaver.
-A	Causes the parameters to be saved without synchronizing the device.
-S	Requests that circuits and endpoints be discovered in the course of synchronizing the device.

-E	Requests that endpoints already discovered be resynchronized.
-O	Specifies that circuit names may be overwritten.

Example:

The following command synchronizes device 172.20.4.7:

OLCmd sync 172.20.4.7

OLCmd uhread

The **uhread** command starts and stops the User History Reader (UHReader) from a command line. It can be used to process user history records automatically during periods of low network activity.

Format:

```
OLCmd uhread -c command [-n poller-name ]
```

Where:

-c *command*	Specifies the action to be taken. *command* may be one of:

- **singleRead** – Reads all collected user history poll files, then halts
- **run** – Runs UHReader until the shutdown command is issued
- **shutdown** – Shuts down UHReader if it is running
- **status** – Shows the status of UHReader

-n *poller-name*	Optionally specifies the poller domain to read from when the singleRead or run command is issued. If no poller domain is specified, all poller domains are read.

Example:

The following command causes UHReader to read all collected user history poll files, then stop:

```
OLCmd uhread -c singleRead
```

OLCmd unassigndev

The **unassigndev** command removes devices from a customer record.

Format:

```
OLCmd unassigndev -c customer_name [-a device_address] | [-l device_list]
```

Where:

-c *customer_name*	Is the customer name to be acted upon.
-a *device_address*	Is the IP address of a device to be unassigned.
-l *device_list*	Is a text file containing a list of device IP addresses separated by carriage returns. The group of devices is unassigned.

Example:

The following command removes a single device from customer Exact_Corporation:

```
OLCmd unassigndev -c Exact_Corporation -a 137.16.255.5
```

The following command assigns a group of devices, which are listed in the file exact_devices.txt:

```
OLCmd unassigndev -c Exact_Corporation -l exact_devices.txt
```

OLCmd viewlog

The **viewlog** command displays selected records from a log file.

Format:

```
OLCmd viewlog [-l] [-f log_type] [-sd mm/dd/yyyy] [-ed mm/dd/yyyy]
 [-st hh:mm:ss] [-et hh:mm:ss] [-u userid] [-ip ip_address] [log_name]
```

Where:

-l	Requests a list of available log files which may be specified as *log_name*. If -l is specified, other parameters are ignored.
-f *log_type*	Is the optional severity of log messages to be displayed. *log_type* may be one of:

- **fi** – Fatal and internal messages are displayed.
- **fic** – Fatal, internal, and critical messages are displayed.
- **ficw** – Fatal, internal, critical, and warning messages are displayed.
- **full** – All messages are displayed.

If -f is not specified, all messages are displayed.

-sd *mm/dd/yyyy*	Is the optional starting date for log records to be displayed. If omitted, records from the beginning of the file are displayed.
-ed *mm/dd/yyyy*	Is the optional ending date for log records to be displayed. If omitted, records up to the end of the file are displayed.
-st *hh:mm:ss*	Is the optional starting time for log records to be displayed. If omitted, records from the beginning of each day are displayed.
-et *hh:mm:ss*	Is the optional ending time for log records to be displayed. If omitted, records up to the end of each day are displayed.
-u *userid*	Specifies a userid. Log records containing the userid are displayed.

-ip *ip_address*   Specifies an IP address. Log records containing the IP address are displayed.

log_name   Is the name of the log file to be displayed. It is required unless the -l option is used. *log_name* may be one of:

- **AlarmConfigServer** – Alarm Configuration
- **batch_slv_rpt** – Scheduled SLV Reports
- **Configuration** – Configuration Health and Status
- **dbage** – Database Aging
- **dbroll** – Database Roll-up
- **demand_slv_rpt** – On Demand SLV Reports
- **DeviceSync** – Device Synchronization
- **DownloadServer** – Firmware/Configuration/Feature Maintenance
- **UHLister** – Poller Controller
- **UHPoller_default** – Default Polling Domain
- **UHReader** – Database Reader

Example:

The following command displays fatal and internal messages collected in the UHReader log file during one week:

```
OLCmd viewlog -f fi -sd 05/07/2000 -ed 05/13/2000 UHReader
```

OLCmd xmlreport

The **xmlreport** command generates OpenLane reports based on eXtensible Markup Language (XML) report requests, and optionally transforms the output based on an eXtensible Stylesheet Language Transformations (XSLT) file. For more information, see Appendix B, *Using XML to Produce Reports*.

Format:

OLCmd xmlreport *xml_request_file* [*xml_output_file*] [*xsl_transform_file*] [*transformed_out_file*]

Where:

xml_request_file	Is the full path to the file containing the report request in XML.
xml_output_file	Is the full path of file where the XML output should reside. If not specified, output goes to Standard Out.
xsl_transform_file	Is the full path of an XSLT file. This is optional.
transformed_out_file	Is the full path of the output file from the XSL transformation. If not specified, output goes to a file named <xml_output_file>.xml.

Example:

The following command generates a report. The report request is in the file named request1234.xml, and the report is written to a file named report_10122001.html.

```
OLCmd \opt\pdn\openlane\cust_requests\request1234.xml
 \opt\pdn\openlane\xmlreports\report_10122001.html
```

Using XML to Produce Reports

B

XML Reports

OpenLane's eXtensible Markup Language (XML) reporting capability allows reports to be requested using XML. The format of the report to be returned is defined in the XML request itself. Reports can be generated in XML, HyperText Markup Language (HTML), or Comma-Separated Variable (CSV) formats.

Additionally, if the report output is specified to be XML, an eXtensible Stylesheet Language (XSL) transformation may be applied to that report to reformat the output into different XML, or into a totally different format.

Note that some web browsers (such as Internet Explorer 5) can interpret XML in the correct hierarchical format. Some browsers just show the XML source.

There are two ways to produce XML reports:

■ Command Line interface

■ Servlet interface

Command Line Interface (OLCmd xmlreport)

The OLCmd command xmlreport accepts a report request in XML and generates a report in a specified format. The output is sent to a file. XML output can optionally be transformed into different formats by use of an XSL Transformation file.

For more information, see *OLCmd xmlreport* in Appendix A, *OLCmd Reference*.

Servlet Interface (SlvXmlServlet)

The XML report servlet can be found at:

`/opt/pdn/OpenLane/OLServlets/pdn.webcgi.SlvXmlServlet`

The SlvXmlServlet servlet accepts a report request in XML and generates a report in a specified format. The output is sent to the browser window. XML output can optionally be transformed into different formats by an XSL Transformation.

There is a sample Web page to demonstrate the use of the servlet at `/opt/pdn/OpenLane/samples/xml/reports/xmlentry.html`. You can run the sample page from the URL `http://your.domain.name/OpenLane/xmlentry.html` when OpenLane is running.

The easiest way to try the sample Web page is to open an XML report request file, select and copy the text, and paste it into the Web page. Sample XML report requests can be found in the following directories:

- `/opt/pdn/OpenLane/samples/xml/reports/DeviceReports/Requests`

- `/opt/pdn/OpenLane/samples/xml/reports/SlvReports/Requests`

Writing an Interface to the Servlet

The servlet expects a HyperText Transfer Protocol (HTTP) post with up to three parameters:

- **XmlReportRequest.** The first and only mandatory parameter is XmlReportRequest, which contains the report request in XML format.

- **OutputClassOrMime.** The post may also contain a parameter called OutputClassOrMime, which tells a Web browser how to interpret the reply for viewing. This usually matches the output type specified in the report request, but does not have to. For example, you may want to view an HTML/JavaScript report as ASCII by using text/plain as the Multipurpose Internet Mail Extension (MIME) type. The normal reply types for the supported report formats are shown in the following table:

Output	Default MIME Type
DEFAULT (not specified) and errors	text/plain
XML	text/xml
HTML/JavaScript	text/html
CSV	text/plain

- **XslRequest.** When the requested report output type is XML, the post optionally may also contain a parameter called XslRequest, which contains an XSL Transformation to apply.

XML and XSL Examples

The XML format of the report requests and the report outputs are documented through examples. The primary report request example, with comments showing XML tags, attributes, and possible values, is at
/opt/pdn/OpenLane/samples/xml/reports/CommentedExampleRequest.xml.

For more specific inventory report requests (applicable for all device types) see the files in
/opt/pdn/OpenLane/samples/xml/reports/DeviceReports/Requests/.
Examples of the outputs for those reports are in
/opt/pdn/OpenLane/samples/xml/reports/DeviceReports/Reports.

For more specific SLV/SLM report requests (only for FrameSaver devices) see the files in
/opt/pdn/OpenLane/samples/xml/reports/SlvReports/Requests/.
Examples of the outputs for those reports are in
/opt/pdn/OpenLane/samples/xml/reports/SlvReports/Reports.

There is also an XSL transformation example at
/opt/pdn/OpenLane/samples/xml/reports/noRequestTag.xsl.
Normally, when the report output is in XML, the report request is returned as part of the report. This can hinder the import of the XML into a row-based or table-based tool such as Microsoft Access. The XSL transformation in the example removes all non-row data such as the request and messages from the output to facilitate that kind of import.

Additional Resources

For information about XML and for the XML and XSL specifications, see:

■   **www.w3.org/XML**

■   **www.w3.org/Style/XSL**

For information about XML editors and other software, see:

■   **www.xmlsoftware.com**

Index