CHAPTER 1

# Introduction

This section gives a brief overview of the features available in the Cisco Enhanced Device Interface (Cisco E-DI) XML Programmatic Interface (PI).

This chapter includes the following details:



## What Is a Programmatic Interface?

In a network management context, a PI is an interface for management applications to control and monitor networking devices. It is crafted specifically to meet the needs of network management applications. This is in contrast to a command line interface (CLI), which is intended and crafted for human operators.

## What Is Cisco E-DI PI?

Cisco E-DI provides an XML PI based on the NETCONF configuration protocol (draft-ietf-netconf-prot-07.txt). Cisco E-DI acts as the NETCONF agent on behalf of a managed device. NETCONF specifies the protocol operations. Currently the device data model used in an operation is not specified by the protocol. Cisco E-DI defines the device data model based on the device command structure. The data models are described in XML schema definition (XSD) files. These data models are generated at development time, and are specific to a device type and OS combination. The XSD files are available for applications using XML PI.

Some users have management applications using the XML-based data model and script-based tools using CLI. In order to accommodate these users who want to continue making use of the CLI rather than the XML command model, the PI operations allow the transport of CLI commands and responses. See Chapter 2, “XML Programmatic Interface” for details about how to use the XML data mode, and Chapter 3, “CLI Mode” for details about how to use the CLI commands in NETCONF operations.

**Note** Refer to the NETCONF Configuration Protocol draft-ietf-netconf-prot-07 document which is included on the Cisco E-DI Documentation CD-ROM. This document defines the NETCONF configuration protocol which provides mechanisms to install, manipulate, and delete the configuration of network

devices.It uses an Extensible Markup Language (XML) based data encoding for the configuration data as well as the protocol messages.The NETCONF protocol operations are realized on top of a simple Remote Procedure Call (RPC) layer.

The Cisco E-DI XML PI uses Telnet and SSH version1 or version 2 for transport.

Cisco E-DI PI is used for the following device level tasks:

- Configuring the device (also available through the CLI)
- Executing **EXEC** commands on the device (also available through the CLI)

The Cisco E-DI XML PI supports:

- Sessions Establishment
- Session Release
- Protocol operations/services.
- RPC messages based communication model for message exchange
- RPC message pipelining
- Various data stores during configuration e.g. running, startup
- Capabilities such as writable-running capability, candidate-configuration capability
- Sub-tree filtering
- The following protocol operations/services:
  - <get-config>
  - <edit-config>
  - <copy-config>
  - <delete-config
  - <lock>
  - <unlock>
  - <get>
  - <close-session>
  - <kill-session>
  - <commit>
  - <discard-changes>
- Versioning of the data model

**Note** Management operations and data models will be backward compatible. See Appendix A, “Level of Support for NETCONF Protocol Operations and Features”.

# Example Use Case

This simple use case details how to add a user to a running configuration. It identifies the protocol operations that are available, and explains how to use them. The remaining chapters of this guide will go into details of individual protocol operations and how to use them with E-DI XML PI.

Applications can use the NETCONF primitives to build more complex management scenarios.

1. The application establishes a NETCONF session with Cisco E-DI for the device to be managed—Cisco E-DI provides various ways of establishing a NETCONF session. See Appendix B, "NETCONF Client GUI" for more details.
2. Get the running configuration using a filter on the username—Applications use the standard **<get-config>** operation. In the filter, to express the command for the username, application refers to the device specific XSD. Alternatively, application can use CLI commands.
3. Make sure that the user does not already exist—This is done in the application's code.
4. Add a username to the candidate configuration—Application uses the **<edit-config>** operation with the candidate as the target data store.
5. Validate the candidate configuration—Application uses the **<validate>** operation.
6. Get a lock on the running configuration—Application uses the **<lock>** operation.
7. Commit the configuration change—Application uses the **<commit>** operation.
8. Check the running configuration with a filter on the username—Applications use the standard operation. In the filter, to express the command for the username, application refers to the device specific XSD. Alternatively, application can use CLI commands.
9. Make sure that the username is now returned—This check is done by the application in its own code.
10. Release the lock on the running configuration—Application uses the **<unlock>** operation.
11. Close the session—Application uses the **<close-session>** operation.

The Cisco E-DI XML PI includes the following supplementary material to help you understand and use the XML PI. These files are available on the Cisco E-DI Documentation CD-ROM:

- A simple GUI client application—Appendix B, "NETCONF Client GUI" explains how to use this GUI tool.
- The application is shipped as jar files.
- A set of JAVA API classes—These classes help the applications to connect to Cisco E-DI and conduct NETCONF sessions.
- A simple java program which illustrates the usage of API classes.
- A set of javadoc documentation for the API classes.

# Using Cisco E-DI With CatOS Devices

Cisco Catalyst OS (CatOS) devices do not have a startup configuration. For consistency with Cisco IOS devices, Cisco E-DI allows the notion of a startup configuration for CatOS devices, and shows the running configuration as the startup configuration. Since both configurations are essentially the same, the following variations in the behavior of NETCONF operations occur:

- Locking the running configuration or startup configuration locks both running and startup configurations. Unlocking unlocks both running and startup configurations.
- Copying a running configuration into a startup configuration has no effect.

## Using a Filter With CatOS Devices

In CLI mode, <get-config> and <get> permits filter specification using the <text-filter-spec> element:

- On Cisco IOS devices, this can take any values which are allowed on the device following the **show running-config** command.
- On CatOS devices, this can take any values allowed on the device following the **show config** command.

**Note** For details of the Command Line Interface (CLI) mode provided by Cisco E-DI XML PI, see Chapter 3, "CLI Mode"