



Automation Scripts

- [Operational simplicity using automation scripts](#) , on page 1

Operational simplicity using automation scripts

An automation script is a software program that

- automates configuration and operational tasks on network devices,
- interacts with network operating systems through standard interfaces such as NETCONF, SNMP, and SSH, and
- can be executed either externally or directly on the network device.
- On-box automation scripts: Scripts that reside and execute directly on the router, eliminating the need for external controllers.
- Off-box automation scripts: Scripts that run on external controllers and interact with network devices through APIs over the network.

Table 1: Feature History Table

Feature Name	Release Information	Description
Operational Simplicity Using Automation Scripts	Release 25.1.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100])(select variants only*) *This feature is now supported on Cisco 8712-MOD-M routers.

Feature Name	Release Information	Description
Operational Simplicity Using Automation Scripts	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*) *This feature is now supported on: • 8212-48FH-M • 8711-32FH-M • 88-LC1-12TH24FH-E • 88-LC1-36EH+A8:B12 • 88-LC1-52Y8H-EM
Operational Simplicity Using Automation Scripts	Release 7.3.2	This feature lets you host and execute your automation scripts directly on a router running IOS XR software, instead of managing them on external controllers. The scripts available on-box can now leverage Python libraries, access the underlying router information to execute CLI commands, and monitor router configurations continuously. This results in setting up a seamless automation workflow by improving connectivity, access to resources, and speed of script execution. The following categories of on-box scripts are used to achieve operational simplicity: • Config scripts—Implement custom configuration rules, and notify the user to take action when the configuration conditions are not met. • Exec scripts—Automate operational tasks and network troubleshooting. • Process scripts—Monitor the system continuously using daemons. • EEM scripts—Respond to a predefined set of events.

Types of automation scripts

There are four types of on-box automation scripts that you can leverage to automate your network operations:

- Configuration (Config) scripts
- Execution (Exec) scripts
- Process scripts
- EEM scripts

This table provides the scope and benefit of on-box scripts.

Table 2: On-Box Automation Scripts

	Config Scripts	Exec Scripts	Process Scripts	EEM Scripts
What is the scope of the script?	Enforce contextual and conditional changes to configurations, validate configurations before committing the changes to detect and notify potential errors. If configuration does not comply with the rules that are defined in the script, an action can be invoked. For example, generate a warning, syslog message, or halt a commit operation.	Run operational commands or RPCs, process the output, generate syslogs, configure system, perform system action commands such as system reload, process restarts, and collect logs for further evaluation.	Daemonize to continuously run as an agent on the router to execute additional checks outside traditional ZTP. Daemonized scripts are similar to exec scripts but run continuously. The script executes operational commands on the router and analyzes the output.	Run operational commands or RPCs, generate, and determine the next steps like logging the root cause or changing device configuration. Event policies can upload the output of event scripts to an on-box or off-box location for further analysis.
How to invoke the script?	All config scripts are processed automatically when commit command is executed on the router.	Exec script is invoked manually via CLI command or RPC.	Process script is activated via configuration CLI command.	Event scripts are invoked by defined event policies in response to a system event and allow for immediate action to take effect.
What are the main benefits of using the script?	Simplifies complex configurations and averts potential errors before a configuration is committed. Ensures that the network configuration complies with rules and policies that are defined in the script.	Collects operational information, and decreases the time that is involved in troubleshooting issues. Provides flexibility in changing the input parameters for every script run. This fosters dynamic automation of operational information.	Runs scripts as a daemon to continuously perform tasks that are not transient.	Automates log collection upon detecting error conditions that are defined by event policies. Uploads the output of event scripts to an on-box or off-box location for further analysis.

