



## **Cisco UCS Director API Integration and Customization Guide, Release 6.5**

**First Published:** 2017-07-11

### **Americas Headquarters**

Cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, CA 95134-1706  
USA  
<http://www.cisco.com>  
Tel: 408 526-4000  
800 553-NETS (6387)  
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <http://www.cisco.com/go/trademarks>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

© 2016 Cisco Systems, Inc. All rights reserved.



## CONTENTS

---

### Preface

#### Preface v

Audience v

Conventions v

Related Documentation vii

Documentation Feedback vii

Obtaining Documentation and Submitting a Service Request vii

---

### CHAPTER 1

#### New and Changed Information for this Release 1

New and Changed Information for this Release 1

---

### CHAPTER 2

#### Overview 3

Audience 3

Cisco UCS Director SDK 3

---

### CHAPTER 3

#### The REST API 5

Cisco UCS Director REST API 5

When to Use Cisco UCS Director REST API 5

Using Cisco UCS Director REST API 5

---

### CHAPTER 4

#### Open Automation 7

Cisco UCS Director Open Automation 7

When to Use Cisco UCS Director Open Automation 7

Using Cisco UCS Director Open Automation 8

---

### CHAPTER 5

#### Custom Tasks 9

Custom Tasks and Script Bundles 9

When to Use Custom Tasks and Script Bundles 9

Using Custom Tasks and Script Bundles 11

---

**CHAPTER 6**

**Script Modules 13**

Script Modules 13

When to Use Script Modules 13

Using Script Modules 13

---

**CHAPTER 7**

**The PowerShell API 15**

Cisco UCS Director PowerShell API 15

When to Use Cisco UCS Director PowerShell API 15

Using Cisco UCS Director PowerShell API 15



## Preface

---

- [Audience, page v](#)
- [Conventions, page v](#)
- [Related Documentation, page vii](#)
- [Documentation Feedback, page vii](#)
- [Obtaining Documentation and Submitting a Service Request, page vii](#)

## Audience

This guide is intended primarily for data center administrators who use Cisco UCS Director and who have responsibilities and expertise in one or more of the following:

- Server administration
- Storage administration
- Network administration
- Network security
- Virtualization and virtual machines

## Conventions

| Text Type       | Indication   |
|-----------------|--|
| GUI elements    | GUI elements such as tab titles, area names, and field labels appear in <b>this font</b> .<br>Main titles such as window, dialog box, and wizard titles appear in <b>this font</b> . |
| Document titles | Document titles appear in <i>this font</i> .   |
| TUI elements    | In a Text-based User Interface, text the system displays appears in <i>this font</i> .   |

| Text Type     | Indication  |
|---------------|---|
| System output | Terminal sessions and information that the system displays appear in <i>this font</i> .                                     |
| CLI commands  | CLI command keywords appear in <b>this font</b> .<br>Variables in a CLI command appear in <i>this font</i> .                |
| [ ]           | Elements in square brackets are optional.   |
| {x   y   z}   | Required alternative keywords are grouped in braces and separated by vertical bars.   |
| [x   y   z]   | Optional alternative keywords are grouped in brackets and separated by vertical bars.                                       |
| string        | A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks. |
| <>            | Nonprinting characters such as passwords are in angle brackets.   |
| [ ]           | Default responses to system prompts are in square brackets.   |
| !, #          | An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.                   |

**Note**

Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the document.

**Caution**

Means *reader be careful*. In this situation, you might perform an action that could result in equipment damage or loss of data.

**Tip**

Means *the following information will help you solve a problem*. The tips information might not be troubleshooting or even an action, but could be useful information, similar to a Timesaver.

**Timesaver**

Means *the described action saves time*. You can save time by performing the action described in the paragraph.

**Warning****IMPORTANT SAFETY INSTRUCTIONS**

This warning symbol means danger. You are in a situation that could cause bodily injury. Before you work on any equipment, be aware of the hazards involved with electrical circuitry and be familiar with standard practices for preventing accidents. Use the statement number provided at the end of each warning to locate its translation in the translated safety warnings that accompanied this device.

SAVE THESE INSTRUCTIONS

## Related Documentation

**Cisco UCS Director Documentation Roadmap**

For a complete list of Cisco UCS Director documentation, see the *Cisco UCS Director Documentation Roadmap* available at the following URL: [http://www.cisco.com/en/US/docs/unified\\_computing/ucs/ucs-director/doc-roadmap/b\\_UCSDirectorDocRoadmap.html](http://www.cisco.com/en/US/docs/unified_computing/ucs/ucs-director/doc-roadmap/b_UCSDirectorDocRoadmap.html).

**Cisco UCS Documentation Roadmaps**

For a complete list of all B-Series documentation, see the *Cisco UCS B-Series Servers Documentation Roadmap* available at the following URL: <http://www.cisco.com/go/unifiedcomputing/b-series-doc>.

For a complete list of all C-Series documentation, see the *Cisco UCS C-Series Servers Documentation Roadmap* available at the following URL: <http://www.cisco.com/go/unifiedcomputing/c-series-doc>.

**Note**

The *Cisco UCS B-Series Servers Documentation Roadmap* includes links to documentation for Cisco UCS Manager and Cisco UCS Central. The *Cisco UCS C-Series Servers Documentation Roadmap* includes links to documentation for Cisco Integrated Management Controller.

## Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to [ucs-director-docfeedback@cisco.com](mailto:ucs-director-docfeedback@cisco.com). We appreciate your feedback.

## Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, using the Cisco Bug Search Tool (BST), submitting a service request, and gathering additional information, see [What's New in Cisco Product Documentation](#).

To receive new and revised Cisco technical content directly to your desktop, you can subscribe to the [What's New in Cisco Product Documentation RSS feed](#). RSS feeds are a free service.







# New and Changed Information for this Release

---

This chapter contains the following sections:

- [New and Changed Information for this Release, page 1](#)

## New and Changed Information for this Release

No significant changes were made to this guide for the current release.





## CHAPTER 2

# Overview

---

This chapter contains the following sections:

- [Audience, page 3](#)
- [Cisco UCS Director SDK, page 3](#)

## Audience

This document is intended for technical professionals interested in using the Cisco UCS Director Software Development Kit (SDK) and related technologies. Such users might be:

- System and cloud administrators who use Cisco UCS Director and want to extend their ability to manage cloud resources using Cisco UCS Director.
- Third-party developers who want to build connectors to enable their devices to be managed by Cisco UCS Director.
- Anyone else who wants to compare Cisco UCS Director SDK or API technologies and determine which would be most appropriate for their application.

## Cisco UCS Director SDK

The Cisco UCS Director SDK is a collection of technologies that enable you to extend the capabilities of Cisco UCS Director, access Cisco UCS Director data, and invoke Cisco UCS Director's automation and orchestration operations from any application. The Cisco UCS Director SDK includes the REST APIs and Open Automation. Scripting technologies include the Cisco UCS Director PowerShell API, custom tasks bundled in Cisco UCS Director script modules, and the ability to write your own custom tasks using Cloupiascript, a server-side JavaScript implementation.

With Cisco UCS Director SDK technologies, you can:

- Access Cisco UCS Director programmatically—Use the Cisco UCS Director REST API to invoke workflows and obtain reports.
- Customize Cisco UCS Director—Create custom workflow tasks. Customize Cisco UCS Director by deploying your own jar files and script libraries in script modules. Use custom tasks from script bundles.

- Extend Cisco UCS Director—Use Cisco UCS Director Open Automation to build connectors that support additional devices and systems. Use the Cisco UCS Director PowerShell API to connect to Microsoft System Center Virtual Machine Manager (SCVMM) and other PowerShell enabled devices.

This document describes each of these enabling technologies and their uses and tells you where to look for information about getting started. For each of the SDK technologies, this document describes the following:

- The basic concepts required to begin using the programmatic features of Cisco UCS Director.
- How and when to use which SDK components.
- What skills are required to use an interface.
- What computing resources are required to use an interface.



## The REST API

---

This chapter contains the following sections:

- [Cisco UCS Director REST API, page 5](#)

### Cisco UCS Director REST API

Cisco UCS Director offers a REST API that enables applications to consume or manipulate the data stored in Cisco UCS Director. Cisco UCS Director REST API (hereafter, simply "the REST API") is a lightweight framework that requires little overhead for an application to use.

Applications use HTTP and HTTPS requests from the REST API to perform Create/Read/Update/Delete (CRUD) operations on Cisco UCS Director resources.

### When to Use Cisco UCS Director REST API

Cisco UCS Director REST API is a language-independent interface that can be used by any program or script capable of making HTTP or HTTPS requests. Use the REST API when you want to invoke operations on Cisco UCS Director from a separate program or process.

Applications can use the REST API to do the following:

- Retrieve Cisco UCS Director reports on physical and virtual devices, networks, appliances, groups and users, policies, resource accounting, funds, and other monitored entities within your Cisco UCS domains.
- Invoke Cisco UCS Director Orchestrator workflow and task operations.
- Invoke additional operations specific to Cisco UCS Director.

### Using Cisco UCS Director REST API

Because a REST API client interacts with Cisco UCS Director using standard HTTP or HTTPS requests and responses, the REST API responses are compatible with any web browser. Many programming languages have libraries devoted to creating and sending HTTP requests and handling HTTP responses.

Most of the REST API calls send and return data in the request or the response, respectively. These data payloads may be formatted in one of two ways depending on the REST API call. Some of the REST API calls use a JavaScript Object Notation (JSON) payload, while others use an XML payload. You probably have to use both for any reasonably complex application.

A JSON-based REST API call is a plain HTTP request and response with a JSON payload. JSON is a lightweight text-based open standard designed for human-readable data interchange. JSON represents simple data structures and associative arrays. Your application directly invokes the JSON-based API without using any specialized REST API libraries, and parses JSON data using any means native to your application.

An XML-based REST API call is also an HTTP request and response, but requires that your application include the Cisco UCS Director REST API library jars and that your application use the API-defined JavaBeans (also called "plain old Java objects, or POJOs) in the REST payload. The API jars enable you to marshal and unmarshal the payload of the XML message to and from a Java object respectively.

Authorization in the REST API is enforced by requiring that only registered users of Cisco UCS Director are able to make API requests. When a user is created and registered on Cisco UCS Director, the user is assigned a unique REST API access key. This key must be provided in the HTTP or HTTPS request header when making an API call. The user then has access to the same data and operations via the REST API that it has in Cisco UCS Director.

For system requirements and information about how to set up a development environment, install libraries, and begin using the REST API, see the [Cisco UCS Director REST API Getting Started Guide](#).

For annotated examples of using the REST API, see the [Cisco UCS Director REST API Cookbook](#).

For a complete list of every Cisco UCS Director REST API function, see the [Cisco UCS Director REST API URI Reference](#).



## Open Automation

---

This chapter contains the following sections:

- [Cisco UCS Director Open Automation, page 7](#)

### Cisco UCS Director Open Automation

Cisco UCS Director provides the Cisco UCS Director Open Automation module (hereafter, simply "Open Automation") to enable you to enhance the functionality of the Cisco UCS Director appliance.

Using Open Automation, you can add a module to Cisco UCS Director. The module adds a new capability to Cisco UCS Director, such as the ability to control a new device type or to generate a new type of report.

For more information about existing functionality in Cisco UCS Director, see the [Cisco UCS Director Administration Guide](#), the [Cisco UCS Director Application Container Guide](#), the [Cisco UCS Director Orchestration Guide](#), and the numerous other guides that document functionality available using the Cisco UCS Director application.

For system requirements and information about how to set up a development environment, install libraries, and begin using the Open Automation SDK, see the [Cisco UCS Director Open Automation Getting Started Guide](#).

For annotated examples of using the Open Automation SDK, see the [Cisco UCS Director Open Automation Cookbook](#).

For a reference to the Cisco UCS Director Open Automation API, see the [Cisco UCS Director API Javadoc](#).

### When to Use Cisco UCS Director Open Automation

Open Automation is a Java SDK and framework for extending the functionality of Cisco UCS Director. Use Open Automation to enhance Cisco UCS Director in the following ways:

- Develop your own or third-party components with Cisco UCS Director.
- Design a custom menu for displaying your device or component.
- Inventory your devices.
- Provide the ability to test connections between your device and Cisco UCS Director.

- Develop tasks that can be used in workflows.
- Expose your tasks in the form of a REST API.
- Develop and schedule repeatable tasks.
- Develop your own Cisco UCS Director reports and report actions.
- Develop new Cisco UCS Director Cloudsense reports.
- Track changes made to the system through your module.
- Customize your dashboard display by providing your own dashboard (stack) builder.
- Develop your own items that can be displayed in your dropdown boxes.
- Provide support for new account types.

## Using Cisco UCS Director Open Automation

A *module* is the topmost logical entry point into Cisco UCS Director. To add or extend any functionality, you must develop and deploy a module on Cisco UCS Director. A module developed using Open Automation behaves just as do any Cisco UCS Director built-in features or modules.

To develop and deploy a module, you:

- Create a module by extending the `AbstractCloupiModule` class in the Open Automation Java API.
- Add forms, reports, workflow tasks, or other functionality to the module.
- If necessary, create new custom tasks by implementing task interfaces in the Java API.
- Package the module using a build tool such as Apache ANT or Apache Maven. An example ANT build file is provided with Open Automation to help you get started with module packaging.
- Deploy the package on Cisco UCS Director.

For system requirements and information about how to set up a development environment, install libraries, and begin using Cisco UCS Director Open Automation, see the [Cisco UCS Director Open Automation Getting Started Guide](#).

For annotated examples of using Open Automation, see the [Cisco UCS Director Open Automation Cookbook](#).

For a reference to Cisco UCS Director Open Automation classes and methods, see the [Cisco UCS Director Open Automation Javadoc](#).





## Custom Tasks

---

This chapter contains the following sections:

- [Custom Tasks and Script Bundles](#), page 9

### Custom Tasks and Script Bundles

Custom tasks enable developers to perform customized operations on Cisco UCS Director resources. You write custom tasks using Cloupiascript, a language similar to JavaScript. You then use your custom tasks like any other task, including them in workflows to orchestrate work on your systems.

Script bundles are collections of custom tasks that are included with each new Cisco UCS Director release for a variety of specific applications. You can download a script bundle and import the custom tasks into Cisco UCS Director.

In both cases the goal is the same: to expand the range of tasks available for use in orchestration workflows in Cisco UCS Director.

### When to Use Custom Tasks and Script Bundles

Custom tasks and script bundles supplement the task library that comes with Cisco UCS Director. Use them when there is no existing task that implements the functionality you need.

#### When to Use Custom Tasks

Use custom tasks when you have a single operation or unit of work that will recur in your workflows, and there is no existing task that performs the operation.

#### When to Use Script Bundles

Script bundles are provided with every major release. It is worthwhile to investigate new script bundles to see if any of the new or enhanced tasks implement workflow functionality required by your organization.

Following is a list of scripts provided in the script bundles that can be downloaded with the Cisco UCS Director image:

| Script Number | Script File                                      |
|---------------|--|
| 1             | Add-UCSM-UUIDPool.wfdx                           |
| 2             | AddPortgroupToNetworkPolicy.wfdx                 |
| 3             | AddVMToInventory.wfdx                            |
| 4             | AssignCatalogsToAGroup.wfdx                      |
| 5             | Change-Cluster-In-Computing-Policy.wfdx          |
| 6             | ConfigureVMAnnotation.wfdx                       |
| 7             | Create-UCSM-WWNN-WWPN-Pool.wfdx                  |
| 8             | CreateNetappAccount.wfdx                         |
| 9             | CreateUCSAccount.wfdx                            |
| 10            | CreateUCSMNetworkPolicy.wfdx                     |
| 11            | CreateUCSMServiceProfileTemplate.wfdx            |
| 12            | CreateUCSMStoragePolicy.wfdx                     |
| 13            | CreateVMWareDatastoreClusterStoragePolicy.wfdx   |
| 14            | DeleteVMWareSystemPolicy.wfdx                    |
| 15            | Free-VLANFrom-VLANPool.wfdx                      |
| 16            | Get Vmware HostNode Identity.wfdx                |
| 17            | Get vCenter account details.wfdx                 |
| 18            | Get-NetApp-Volume.wfdx                           |
| 19            | Get-WWN-Of-Vmware-iSCSI-and-FC-Adapter-Type.wfdx |
| 20            | Get-Workflow-Output-Value.wfdx                   |
| 21            | GetAPICAccountDetails.wfdx                       |
| 22            | GetAllVMDetailsInAnArray.wfdx                    |
| 23            | GetFailedServiceRequestDetails.wfdx              |
| 24            | GetGroupAdminUserFromUserGroup.wfdx              |
| 25            | GetHypervVMSnapshotDetails.wfdx                  |
| 26            | GetInputDetailsBySRId.wfdx                       |
| 27            | GetNetAppInterfaceIdentity.wfdx                  |
| 28            | GetUCSServerUUID.wfdx                            |
| 29            | GetVMwareVMsByUserId.wfdx                        |
| 30            | GetVNICAdapterOfProvisionedVM.wfdx               |
| 31            | GetWWNofBootDevicesWorkflow.wfdx                 |

| Script Number | Script File                         |
|---------------|-------------------------------------|
| 32            | Modify VM Property Attributes.wfdx  |
| 33            | ModifyUserPasswordAndAPIKey.wfdx    |
| 34            | MultiTaskExample.wfdx               |
| 35            | RemoveEsxiCluster.wfdx              |
| 36            | ScheduleWorkflowServiceRequest.wfdx |
| 37            | ServiceContainerGatewayVLANs.wfdx   |
| 38            | VM-Naming.wfdx                      |

## Using Custom Tasks and Script Bundles

As with any other task, custom tasks and tasks included in script bundles are executed as part of a workflow.

### How to Use Custom Tasks

You create custom tasks using a wizard in Cisco UCS Director. To create a custom task, you must supply:

- A name for the custom task.
- Inputs and outputs for the custom task.
- (Optional) Input Controllers, written in Cloupiascript.
- Cloupiascript instructions to do the work of the custom task.

For system requirements and information about how to set up a development environment, install libraries, and begin using CloupiaScript to write custom tasks, see the [Cisco UCS Director Custom Task Getting Started Guide](#).

For examples of creating custom tasks, see the [Cisco UCS Director Custom Task Cookbook](#).

For a reference to Cisco UCS Director Custom Task classes and methods, see the [Cisco UCS Director Javadoc](#).

### How to Use Script Bundles

To use the custom tasks in a script bundle, you simply download the script bundle and import custom tasks from the bundle.

To download a script bundle for the current release of Cisco UCS Director, go to [Cisco DevNet](#).

For instructions on importing custom tasks from a script bundle, see *Importing Workflows* in the [Cisco UCS Director Orchestration Guide](#).





## Script Modules

---

This chapter contains the following sections:

- [Script Modules, page 13](#)

### Script Modules

Script modules are used to integrate third-party jars and custom libraries with Cisco UCS Director to perform specific operations. Some script module operations are already defined in Cisco UCS Director, including:

- Creating advanced controls to collect user input in workflows, including tabular reports and lists of values (LOVs).
- Context workflow mapping, which enables you to attach workflows to custom actions in a report in Cisco UCS Director.

For more examples of script module use, see the [Cisco UCS Director Orchestration Guide](#).

### When to Use Script Modules

Use a script module when you want to create custom functionality that can be used in custom workflow tasks or added to Cisco UCS Director. Script modules can be exported and re-used in different instances of Cisco UCS Director.

### Using Script Modules

You create script modules in Cisco UCS Director. You can then call the script module's libraries and methods from custom tasks.

You can import and export script modules along with workflows, custom tasks, and activities in Cisco UCS Director using the **Import** and **Export** actions respectively in the **Policies > Orchestration > Workflows** tab.

See the [Cisco UCS Director Orchestration Guide](#) for how to create a script module and add libraries and jar files to the script module.

See the [Cisco UCS Director Custom Task Getting Started Guide](#) for information about using script modules in custom tasks.



---

**Note**

Despite similar names, script modules and script bundles are very different things. Script bundles are packaged collections of workflow tasks that are released with Cisco UCS Director. Script modules are a mechanism for you to add custom functionality to the Cisco UCS Director UI.

---



## The PowerShell API

---

This chapter contains the following sections:

- [Cisco UCS Director PowerShell API, page 15](#)

### Cisco UCS Director PowerShell API

Cisco UCS Director offers JSON-based REST APIs that enable you to submit workflow requests, examine workflow inputs and output schemas, and fetch reports. You can integrate Cisco UCS Director APIs with the Cisco UCS Director PowerShell Console for improved automation of datacenter management.

Cisco UCS Director PowerShell Console provides cmdlet wrappers for the JSON-based APIs. Each cmdlet performs a single operation. The cmdlets are executed in a Microsoft Windows server. Depending on the data returned by the JSON-based APIs, the cmdlets automatically interpret the data and convert them to Windows PowerShell objects. You can chain multiple cmdlets together. To view a list of available cmdlets, see the Cmdlet List in the [Cisco UCS Director PowerShell API Getting Started Guide](#). For more information about REST APIs, see the [Cisco UCS Director REST API Getting Started Guide](#).

### When to Use Cisco UCS Director PowerShell API

Use the Cisco UCS Director PowerShell API to execute the JSON API's of Cisco UCS Director that are provided as cmdlets.

### Using Cisco UCS Director PowerShell API

For an introduction to the Cisco UCS Director PowerShell API, see the [Cisco UCS Director PowerShell API Getting Started Guide](#).

