Red Hat Satellite 6.8

Hammer Cheat Sheet

Hammer CLI Cheat Sheet for Red Hat Satellite

Last Updated: 2020-10-23

Red Hat Satellite 6.8 Hammer Cheat Sheet

Hammer CLI Cheat Sheet for Red Hat Satellite

Red Hat Satellite Documentation Team
satellite-doc-list@redhat.com

Legal Notice

Abstract

This document contains Hammer CLI commands for Red Hat Satellite.

Table of Contents

CHAPTER 1. INTRODUCTION

Hammer is a command-line tool provided with Red Hat Satellite 6. You can use Hammer to configure and manage a Red Hat Satellite Server by using either CLI commands or shell script automation. The following cheat sheet provides a condensed overview of essential Hammer commands.

For more information about Hammer, see the Red Hat Hammer CLI Guide.

CHAPTER 2. GENERAL INFORMATION

Table 2.1. General Information

Subcommand	Description and tasks
--help	Display hammer commands and options, append after a subcommand to get more information

org	The setting is organization-specific, append **--organization *org_name***, or set default organization with:
	`hammer defaults add \`
	`--param-name organization_id \`
	`--param-value org_ID`

loc	The setting is location-specific, append **--location *loc_name***, or set default location with:
	`hammer defaults add \`
	`--param-name location_id \`
	`--param-value loc_ID`

Note: This cheat sheet assumes saved credentials in **~/.hammer/cli_config.yml**. For more information, see Authentication in the Red Hat Hammer CLI Guide.

CHAPTER 3. ORGANIZATIONS, LOCATIONS, AND REPOSITORIES

Table 3.1. Organizations, Locations, and Repositories

Subcommand	Description and tasks

organization	Create an organization:
	`hammer organization create \`
	`--name org_name`
	List organizations:
	`hammer organization list`

location	See the options for organization

subscription
org
	Upload a subscription manifest:
	`hammer subscription upload \`
	`--file path`

repository-set
org
	Enable a repository:
	`hammer repository-set enable \`
	`--product prod_name \`
	`--basearch base_arch \`
	`--releasever rel_v \`
	`--name repo_name`

Subcommand	Description and tasks

repository
org
	Synchronize a repository:
	`hammer repository synchronize \`
	`--product prod_name \`
	`--name repo_name`
	Create a custom repository:
	`hammer repository create \`
	`--product prod_name \`
	`--content-type cont_type \`
	`--publish-via-http true \`
	`--url repo_url \`
	`--name repo_name`
	Upload content to a custom repository:
	`hammer repository upload-content \`
	`--product prod_name \`
	`--id repo_id \`
	`--path path_to_dir`

CHAPTER 4. CONTENT LIFE CYCLES

Table 4.1. Content Life Cycles

Subcommand	Description and tasks

lifecycle-environment org	Create a life cycle environment:
	`hammer lifecycle-environment create \`
	`--name env_name`
	`--description env_desc`
	`--prior prior_env_name`
	List life cycle environments:
	`hammer lifecycle-environment list`

Subcommand	Description and tasks

content-view
org
	Create a content view:
	`hammer content-view create \`
	`--name cv_n \`
	`--repository-ids repo_ID1,... \`
	`--description cv_description`
	Add repositories to a content view:
	`hammer content-view add-repository \`
	`--name cv_n \`
	`--repository-id repo_ID`
	Add Puppet modules to a content view:
	`hammer content-view puppet-module add \`
	`--content-view cv_n \`
	`--name module_name`
	Publishing a content view:
	`hammer content-view publish \`
	`--id cv_ID`
	Promoting a content view:
	`hammer content-view version promote \`
	`--content-view cv_n \`
	`--to-lifecycle-environment env_name`
	Incremental update of a content view:
	`hammer content-view version incremental-update \`
	`--content-view-version-id cv_ID \`
	`--packages pkg_n1,... \`
	`--lifecycle-environment-ids env_ID1,...`

CHAPTER 5. PROVISIONING ENVIRONMENTS

Table 5.1. Provisioning Environments

Subcommand	Description and tasks

domain	Create a domain:
	`hammer domain create \`
	`--name domain_name`

subnet
org
loc
	Add a subnet:
	`hammer subnet create \`
	`--name subnet_name \`
	`--organization-ids org_ID1,... \`
	`--location-ids loc_ID1,... \`
	`--domain-ids dom_ID1,... \`
	`--boot-mode boot_mode \`
	`--network network_address \`
	`--mask netmask --ipam ipam`

compute-resource
org
loc
	Create a compute resource:
	`hammer compute-resource create \`
	`--name cr_name \`
	`--organization-ids org_ID1,... \`
	`--location-ids loc_ID1,... \`
	`--provider provider_name`

medium	Add an installation medium:
	`hammer medium create \`
	`--name med_name \`
	`--path path_to_medium`

partition-table	Add a partition table:
	`hammer partition-table create \`
	`--name tab_name \`
	`--path path_to_file \`
	`--os-family os_family`

template	Add a provisioning template:
	`hammer template create \`
	`--name tmp_name \`
	`--file path_to_template`

Subcommand	Description and tasks

os	Add an operating system:
	`hammer os create \`
	`--name os_name \`
	`--version version_num`

CHAPTER 6. ACTIVATION KEYS

Table 6.1. Activation Keys

Subcommand	Description and tasks

activation-key
org
	Create an activation key:
	`hammer activation-key create \`
	`--name ak_name \`
	`--content-view cv_n \`
	`--lifecycle-environment lc_name`
	Add a subscription to the activation key:
	`hammer activation-key add-subscription \`
	`--id ak_ID \`
	`--subscription-id sub_ID`

CHAPTER 7. USERS AND PERMISSIONS

Table 7.1. Users and Permissions

Subcommand	Description and tasks

user
org
	Create a user:
	`hammer user create \`
	`--login user_name \`
	`--mail user_mail \`
	`--auth-source-id 1 \`
	`--organization-ids org_ID1,org_ID2,...`
	Add a role to a user:
	`hammer user add-role \`
	`--id user_id \`
	`--role role_name`

user-group	Create a user group:
	`hammer user-group create \`
	`--name ug_name`
	Add a role to a user group:
	`hammer user-group add-role \`
	`--id ug_id \`
	`--role role_name`

role	Create a role:
	`hammer role create \`
	`--name role_name`

filter	Create a filter and add it to a role:
	`hammer filter create \`
	`--role role_name \`
	`--permission-ids perm_ID1,perm_ID2,...`

CHAPTER 8. ERRATA

Table 8.1. Errata

Subcommand	Description and tasks

erratum	List errata:
	`hammer erratum list`
	Find erratum by CVE:
	`hammer erratum list --cve CVE`
	Inspect erratum:
	`hammer erratum info --id err_ID`

host	List errata applicable to a host:
	`hammer host errata list \`
	`--host host_name`
	Apply errata to a host:
	`hammer host errata apply \`
	`--host host_name \`
	`--errata-ids err_ID1,err_ID2,...`

CHAPTER 9. HOSTS

Table 9.1. Hosts

Subcommand	Description and tasks

hostgroup
org
loc
	Create a host group:
	`hammer hostgroup create \`
	`--name hg_name \`
	`--puppet-environment env_name \`
	`--architecture arch_name \`
	`--domain domain_name \`
	`--subnet subnet_name \`
	`--puppet-proxy proxy_name \`
	`--puppet-ca-proxy ca-proxy_name \`
	`--operatingsystem os_name \`
	`--partition-table table_name \`
	`--medium medium_name \`
	`--organization-ids org_ID1,... \`
	`--location-ids loc_ID1,...`
	Add an activation key to a host group:
	`hammer hostgroup set-parameter \`
	`--hostgroup "hg_name" \`
	`--name "kt_activation_keys" \`
	`--value key_name`

host
org
loc
	Create a host (inheriting parameters from a host group):
	`hammer host create \`
	`--name host_name \`
	`--hostgroup hg_name \`
	`--interface="primary=true, \`
	`mac=mac_addr, ip=ip_addr, \`
	`provision=true" \`
	`--organization-id org_ID \`
	`--location-id loc_ID \`
	`--ask-root-password yes`

job-template	Add a job template for remote execution:
	`hammer job-template create \`
	`--file path \`
	`--name template_name \`
	`--provider-type SSH \`
	`--job-category category_name`

Subcommand	Description and tasks
job-invocation	Invoke a remote job: hammer job-invocation create \ --job-template *template_name* \ --inputs key1=*value,...* \ --search-query *query* Monitor the remote job: hammer job-invocation output \ --id *job_id* --host *host_name*

CHAPTER 10. TASKS

Table 10.1. Tasks

Subcommand	Description and tasks
task	List all tasks: hammer task list Monitor progress of a running task: hammer task progress \ --id *task_ID*