

Bright Cluster Manager 7.2

Developer Manual

Revision: 08dc3f7

Date: Mon Dec 10 2018



©2015 Bright Computing, Inc. All Rights Reserved. This manual or parts thereof may not be reproduced in any form unless permitted by contract or by written permission of Bright Computing, Inc.

Trademarks

Linux is a registered trademark of Linus Torvalds. PathScale is a registered trademark of Cray, Inc. Red Hat and all Red Hat-based trademarks are trademarks or registered trademarks of Red Hat, Inc. SUSE is a registered trademark of Novell, Inc. PGI is a registered trademark of NVIDIA Corporation. FLEXlm is a registered trademark of Flexera Software, Inc. ScaleMP is a registered trademark of ScaleMP, Inc. All other trademarks are the property of their respective owners.

Rights and Restrictions

All statements, specifications, recommendations, and technical information contained herein are current or planned as of the date of publication of this document. They are reliable as of the time of this writing and are presented without warranty of any kind, expressed or implied. Bright Computing, Inc. shall not be liable for technical or editorial errors or omissions which may occur in this document. Bright Computing, Inc. shall not be liable for any damages resulting from the use of this document.

Limitation of Liability and Damages Pertaining to Bright Computing, Inc.

The Bright Cluster Manager product principally consists of free software that is licensed by the Linux authors free of charge. Bright Computing, Inc. shall have no liability nor will Bright Computing, Inc. provide any warranty for the Bright Cluster Manager to the extent that is permitted by law. Unless confirmed in writing, the Linux authors and/or third parties provide the program as is without any warranty, either expressed or implied, including, but not limited to, marketability or suitability for a specific purpose. The user of the Bright Cluster Manager product shall accept the full risk for the quality or performance of the product. Should the product malfunction, the costs for repair, service, or correction will be borne by the user of the Bright Cluster Manager product. No copyright owner or third party who has modified or distributed the program as permitted in this license shall be held liable for damages, including general or specific damages, damages caused by side effects or consequential damages, resulting from the use of the program or the un-usability of the program (including, but not limited to, loss of data, incorrect processing of data, losses that must be borne by you or others, or the inability of the program to work together with any other program), even if a copyright owner or third party had been advised about the possibility of such damages unless such copyright owner or third party has signed a writing to the contrary.

Table of Contents

Table of Contents	i
0.1 About This Manual	xv
0.2 About The Manuals In General	xv
0.3 Getting Administrator-Level Support	xvi
0.4 Getting Developer-Level Support	xvi
1 Bright Cluster Manager Python API	1
1.1 Installation	1
1.1.1 Windows Clients	1
1.1.2 Linux Clients	2
1.2 Examples	2
1.2.1 First Program	2
1.3 Methods And Properties	3
1.3.1 Viewing All Properties And Methods	3
1.3.2 Property Lists	4
1.3.3 Creating New Objects	4
1.3.4 List Of Objects	5
1.3.5 Useful Methods	7
1.3.6 Useful Example Program	7
2 Metric Collections	11
2.1 Metric Collections Added Using <code>cmsh</code>	11
2.2 Metric Collections Initialization	11
2.3 Metric Collections Output During Regular Use	12
2.4 Metric Collections Error Handling	13
2.5 Metric Collections Consolidator Syntax	13
2.6 Metric Collections Environment Variables	13
2.7 Metric Collections Examples	16
2.8 Metric Collections On iDataPlex And Similar Units	16
3 Bright Cluster Manager JSON API	19
3.1 Services	19
3.1.1 <code>auth</code>	19
3.1.2 <code>ceph</code>	19
3.1.3 <code>cert</code>	19
3.1.4 <code>cloud</code>	19
3.1.5 <code>device</code>	19
3.1.6 <code>gui</code>	19
3.1.7 <code>hadoop</code>	19
3.1.8 <code>job</code>	19
3.1.9 <code>keyvalue</code>	19

3.1.10	lustre	19
3.1.11	main	19
3.1.12	mon	19
3.1.13	net	19
3.1.14	openstack	19
3.1.15	part	19
3.1.16	proc	19
3.1.17	prov	19
3.1.18	puppet	19
3.1.19	serv	19
3.1.20	session	19
3.1.21	test	19
3.1.22	ticket	19
3.1.23	user	19
3.2	Entities	19
3.2.1	BackupRole	19
3.2.2	BadEntityManagers	20
3.2.3	BasicResource	20
3.2.4	BillingHistory	20
3.2.5	BootRole	20
3.2.6	BurnConfig	20
3.2.7	BurnStatus	20
3.2.8	BurnTestStatus	20
3.2.9	Category	20
3.2.10	Ceph	20
3.2.11	CephMonitorRole	20
3.2.12	CephOSDAssociation	20
3.2.13	CephOSDPool	20
3.2.14	CephOSDRole	20
3.2.15	CephState	20
3.2.16	Certificate	20
3.2.17	CertificateRequest	20
3.2.18	Chassis	20
3.2.19	CloudDirectorRole	20
3.2.20	CloudGatewayRole	20
3.2.21	CloudImage	20
3.2.22	CloudJobDescription	20
3.2.23	CloudJobSubmissionStatus	20
3.2.24	CloudNode	20
3.2.25	CloudPrivateCloud	20
3.2.26	CloudProvider	20
3.2.27	CloudRegion	20
3.2.28	CloudSettings	20
3.2.29	CloudStaticIP	20
3.2.30	CloudStorageAction	20
3.2.31	CloudStorageNodeState	20

3.2.32	CloudType	20
3.2.33	CloudVirtualNetworkInterface	20
3.2.34	ClusterSetup	20
3.2.35	CMDaemonBackgroundTask	20
3.2.36	CMDaemonFailover	20
3.2.37	CMDaemonFailoverGroup	20
3.2.38	CMDaemonFailoverGroupStatus	20
3.2.39	CMDaemonFailoverPeer	20
3.2.40	CMDaemonFailoverStatus	20
3.2.41	CMDaemonStatus	20
3.2.42	CMService	20
3.2.43	CondorClientRole	20
3.2.44	CondorJob	20
3.2.45	CondorJobQueue	20
3.2.46	CondorJobQueueStat	20
3.2.47	CondorServerRole	20
3.2.48	ConfigSum	20
3.2.49	ConfigurationOverlay	21
3.2.50	ConsolidatorConf	21
3.2.51	DatabaseRole	21
3.2.52	DellClustat	21
3.2.53	DellClustatGroup	21
3.2.54	DellClustatNode	21
3.2.55	DellDiskGroupInfo	21
3.2.56	DellPhysicalDiskDriveInfo	21
3.2.57	DellRAIDControllerInfo	21
3.2.58	DellSettings	21
3.2.59	DellSettingsFirmware	21
3.2.60	DellSettingsNicDevice	21
3.2.61	DellStorageInfo	21
3.2.62	DellVirtualDiskInfo	21
3.2.63	Device	21
3.2.64	DevStatus	21
3.2.65	DiskAssertion	21
3.2.66	DiskDevice	21
3.2.67	DiskInfo	21
3.2.68	DiskPartition	21
3.2.69	DiskRaid	21
3.2.70	DiskSetup	21
3.2.71	DiskVolume	21
3.2.72	DiskVolumeGroup	21
3.2.73	DrainResult	21
3.2.74	EBSattachAction	21
3.2.75	EBSdetachAction	21
3.2.76	EC2AMI	21
3.2.77	EC2AvailabilityZone	21

3.2.78	EC2EBSStorage	21
3.2.79	EC2EphemeralStorage	21
3.2.80	EC2PrivateCloud	21
3.2.81	EC2Provider	21
3.2.82	EC2Region	21
3.2.83	EC2RegionAMI	21
3.2.84	EC2Settings	21
3.2.85	EC2StaticIP	21
3.2.86	EC2Storage	21
3.2.87	EC2Type	21
3.2.88	EC2VirtualNetworkInterface	21
3.2.89	EntityManagersMD5	21
3.2.90	EthernetSwitch	21
3.2.91	FailoverRole	21
3.2.92	FakeData	21
3.2.93	FSExport	21
3.2.94	FSMount	21
3.2.95	FSPart	21
3.2.96	FSPartAssociation	22
3.2.97	FSPartBasicAssociation	22
3.2.98	FSPartProviderAssociation	22
3.2.99	GenericDevice	22
3.2.100	GenericResource	22
3.2.101	GPUInfo	22
3.2.102	GPUSettings	22
3.2.103	GpuUnit	22
3.2.104	GPUUnitInfo	22
3.2.105	GridEngineJob	22
3.2.106	GridEngineJobQueue	22
3.2.107	GridEngineJobQueueStat	22
3.2.108	GridEngineParallelEnvironment	22
3.2.109	Group	22
3.2.110	GuiCephOsdPoolInfo	22
3.2.111	GuiCephOverview	22
3.2.112	GuiCephPgsInfo	22
3.2.113	GuiClusterOverview	22
3.2.114	GuiCompleteOpenStackOverview	22
3.2.115	GuiDiskUsage	22
3.2.116	GuiGpuUnitOverview	22
3.2.117	GuiHadoopHDFSDetailHBase	22
3.2.118	GuiHadoopHDFSDetailHDFS	22
3.2.119	GuiHadoopHDFSDetailMapreduce	22
3.2.120	GuiHadoopHDFSDetailSpark	22
3.2.121	GuiHadoopHDFSDetailYarn	22
3.2.122	GuiHadoopHDFSDetailZooKeeper	22
3.2.123	GuiHadoopHDFSOverview	22

3.2.124 GuiJob	22
3.2.125 GuiNetSwitchStatus	22
3.2.126 GuiNetworkInterface	22
3.2.127 GuiNodeOverview	22
3.2.128 GuiNodeStatus	22
3.2.129 GuiOpenStackOverview	22
3.2.130 GuiOpenStackProjectOverview	22
3.2.131 GuiOpenStackTenantOverview	22
3.2.132 GuiPDUBank	22
3.2.133 GuiPDUOutlet	22
3.2.134 GuiWorkload	22
3.2.135 HadoopBaseConfiguration	22
3.2.136 HadoopDataNodeHDFSConfiguration	22
3.2.137 HadoopDataNodeRole	22
3.2.138 HadoopHBaseClientHDFSConfiguration	22
3.2.139 HadoopHBaseClientRole	22
3.2.140 HadoopHBaseServerHDFSConfiguration	22
3.2.141 HadoopHBaseServerRole	22
3.2.142 HadoopHDFS	22
3.2.143 HadoopHiveHDFSConfiguration	23
3.2.144 HadoopHiveRole	23
3.2.145 HadoopJob	23
3.2.146 HadoopJobQueue	23
3.2.147 HadoopJobQueueStat	23
3.2.148 HadoopJobTrackerHDFSConfiguration	23
3.2.149 HadoopJobTrackerRole	23
3.2.150 HadoopJournalHDFSConfiguration	23
3.2.151 HadoopJournalRole	23
3.2.152 HadoopKMServerHDFSConfiguration	23
3.2.153 HadoopKMServerRole	23
3.2.154 HadoopNameNodeHDFSConfiguration	23
3.2.155 HadoopNameNodeRole	23
3.2.156 HadoopNFSGatewayHDFSConfiguration	23
3.2.157 HadoopNFSGatewayRole	23
3.2.158 HadoopSecondaryNameNodeHDFSConfiguration	23
3.2.159 HadoopSecondaryNameNodeRole	23
3.2.160 HadoopSparkMasterHDFSConfiguration	23
3.2.161 HadoopSparkMasterRole	23
3.2.162 HadoopSparkWorkerHDFSConfiguration	23
3.2.163 HadoopSparkWorkerRole	23
3.2.164 HadoopSparkYARNHDFSConfiguration	23
3.2.165 HadoopSparkYARNRole	23
3.2.166 HadoopSqoopHDFSConfiguration	23
3.2.167 HadoopSqoopRole	23
3.2.168 HadoopTaskTrackerHDFSConfiguration	23
3.2.169 HadoopTaskTrackerRole	23

3.2.170 HadoopYARNClientHDFSConfiguration	23
3.2.171 HadoopYARNClientRole	23
3.2.172 HadoopYARNServerHDFSConfiguration	23
3.2.173 HadoopYARNServerRole	23
3.2.174 HadoopZooKeeperHDFSConfiguration	23
3.2.175 HadoopZooKeeperRole	23
3.2.176 HAProxyBackendInformation	23
3.2.177 HAProxyEntry	23
3.2.178 HAProxyEntryBind	23
3.2.179 HAProxyFrontendInformation	23
3.2.180 HAProxyNodeInformation	23
3.2.181 HAProxyRole	23
3.2.182 HAProxyServer	23
3.2.183 HAProxySharedSettings	23
3.2.184 HealthCheck	23
3.2.185 HeatMapData	23
3.2.186 IBSSwitch	23
3.2.187 IniConfigFileCustomizationEntry	23
3.2.188 IPCPerm	23
3.2.189 IPResource	23
3.2.190 Job	24
3.2.191 JobQueue	24
3.2.192 JobQueuePlaceholder	24
3.2.193 JobQueueStat	24
3.2.194 KernelModule	24
3.2.195 KeyValuePair	24
3.2.196 LicenseInfo	24
3.2.197 LicenseManagerRole	24
3.2.198 LoginRole	24
3.2.199 LSFBaseJob	24
3.2.200 LSFBaseJobQueue	24
3.2.201 LSFBaseJobQueueStat	24
3.2.202 LSFClientRole	24
3.2.203 LSFJob	24
3.2.204 LSFJobQueue	24
3.2.205 LSFJobQueueStat	24
3.2.206 LSFServerRole	24
3.2.207 LustreAlert	24
3.2.208 LustreClientMount	24
3.2.209 LustreFileSystem	24
3.2.210 LustreFileSystemTarget	24
3.2.211 LustreLog	24
3.2.212 LustreOverview	24
3.2.213 LustreServer	24
3.2.214 LustreServerProfile	24
3.2.215 LustreSettings	24

3.2.216 LustreTargetMap	24
3.2.217 LustreUser	24
3.2.218 LustreVolume	24
3.2.219 LustreVolumeNode	24
3.2.220 MasterNode	24
3.2.221 MasterRole	24
3.2.222 MemcachedRole	24
3.2.223 MemoryInfo	24
3.2.224 Metric	24
3.2.225 MetricPrmId	24
3.2.226 MICHostRole	24
3.2.227 MICInfo	24
3.2.228 MICNode	24
3.2.229 MICNodeCategory	24
3.2.230 MICOverlay	24
3.2.231 MICSettings	24
3.2.232 MonConf	24
3.2.233 MonGlobalConf	24
3.2.234 MonHealthConf	24
3.2.235 MonitoringRole	24
3.2.236 MonMetricConf	24
3.2.237 MsgQueue	25
3.2.238 MyrinetSwitch	25
3.2.239 Network	25
3.2.240 NetworkAliasInterface	25
3.2.241 NetworkBmcInterface	25
3.2.242 NetworkBondInterface	25
3.2.243 NetworkBridgeInterface	25
3.2.244 NetworkInterface	25
3.2.245 NetworkNetMapInterface	25
3.2.246 NetworkPhysicalInterface	25
3.2.247 NetworkTunnelInterface	25
3.2.248 NetworkVLANInterface	25
3.2.249 NewNode	25
3.2.250 NFSexportAction	25
3.2.251 NFSmountAction	25
3.2.252 NFSunexportAction	25
3.2.253 NFSunmountAction	25
3.2.254 Node	25
3.2.255 NodeCategory	25
3.2.256 NodeGroup	25
3.2.257 OpenLavaClientRole	25
3.2.258 OpenLavaJob	25
3.2.259 OpenLavaJobQueue	25
3.2.260 OpenLavaJobQueueStat	25
3.2.261 OpenLavaServerRole	25

3.2.262 OpenStack	25
3.2.263 OpenStackApiAgent	25
3.2.264 OpenStackApiDomain	25
3.2.265 OpenStackApiEndpoint	25
3.2.266 OpenStackApiEntity	25
3.2.267 OpenStackApiFlavor	25
3.2.268 OpenStackApiFloatingIP	25
3.2.269 OpenStackApiGroup	25
3.2.270 OpenStackApiHostAggregate	25
3.2.271 OpenStackApiHypervisor	25
3.2.272 OpenStackApiImage	25
3.2.273 OpenStackApiNetwork	25
3.2.274 OpenStackApiPort	25
3.2.275 OpenStackApiProject	25
3.2.276 OpenStackApiRole	25
3.2.277 OpenStackApiRoleAssignment	25
3.2.278 OpenStackApiRouter	25
3.2.279 OpenStackApiSecurityGroup	25
3.2.280 OpenStackApiServer	25
3.2.281 OpenStackApiService	25
3.2.282 OpenStackApiSubnet	25
3.2.283 OpenStackApiUser	25
3.2.284 OpenStackApiVolume	26
3.2.285 OpenStackApiVolumeSnapshot	26
3.2.286 OpenStackApiVolumeType	26
3.2.287 OpenStackBareMetalApiRole	26
3.2.288 OpenStackBareMetalConductorRole	26
3.2.289 OpenStackBareMetalDiscoverdDNSMasqRole	26
3.2.290 OpenStackBareMetalDiscoverdRole	26
3.2.291 OpenStackBlockStorage	26
3.2.292 OpenStackComputeApiEC2Role	26
3.2.293 OpenStackComputeApiMetadataRole	26
3.2.294 OpenStackComputeApiRole	26
3.2.295 OpenStackComputeConductorRole	26
3.2.296 OpenStackComputeRole	26
3.2.297 OpenStackComputeSchedulerRole	26
3.2.298 OpenStackComputeVNCProxyRole	26
3.2.299 OpenStackConfigFileCustomization	26
3.2.300 OpenStackDashboardRole	26
3.2.301 OpenStackDataProcessingApiRole	26
3.2.302 OpenStackDBaaSRole	26
3.2.303 OpenStackDefaultUserRole	26
3.2.304 OpenStackIdentityApiRole	26
3.2.305 OpenStackImageApiRole	26
3.2.306 OpenStackImageBackend	26
3.2.307 OpenStackImageBackendCeph	26

3.2.308 OpenStackImageBackendFS	26
3.2.309 OpenStackImageRegistryRole	26
3.2.310 OpenStackMessageQueueServerRole	26
3.2.311 OpenStackNetworkApiRole	26
3.2.312 OpenStackNetworkRole	26
3.2.313 OpenStackNodeRole	26
3.2.314 OpenStackNovaImageBackend	26
3.2.315 OpenStackNovaImageBackendCeph	26
3.2.316 OpenStackNovaImageBackendCow	26
3.2.317 OpenStackObjectAccountRole	26
3.2.318 OpenStackObjectApiRole	26
3.2.319 OpenStackObjectContainerRole	26
3.2.320 OpenStackObjectStoreRole	26
3.2.321 OpenStackOrchestrationApiRole	26
3.2.322 OpenStackOrchestrationRole	26
3.2.323 OpenStackSettings	26
3.2.324 OpenStackSettingsAdvanced	26
3.2.325 OpenStackSettingsCMDaemonInteractions	26
3.2.326 OpenStackSettingsCompute	26
3.2.327 OpenStackSettingsCredentials	26
3.2.328 OpenStackSettingsDatabase	26
3.2.329 OpenStackSettingsLogging	26
3.2.330 OpenStackSettingsNetworking	26
3.2.331 OpenStackSettingsPorts	27
3.2.332 OpenStackSettingsQuota	27
3.2.333 OpenStackSettingsUserPortal	27
3.2.334 OpenStackSettingsUsers	27
3.2.335 OpenStackStorage	27
3.2.336 OpenStackTelemetryAgentCentralRole	27
3.2.337 OpenStackTelemetryAgentComputeRole	27
3.2.338 OpenStackTelemetryAgentIpmiRole	27
3.2.339 OpenStackTelemetryAgentNotificationRole	27
3.2.340 OpenStackTelemetryAlarmEvaluatorRole	27
3.2.341 OpenStackTelemetryAlarmNotifierRole	27
3.2.342 OpenStackTelemetryApiRole	27
3.2.343 OpenStackTelemetryCollectorRole	27
3.2.344 OpenStackUserRole	27
3.2.345 OpenStackUserSettings	27
3.2.346 OpenStackVolumeApiRole	27
3.2.347 OpenStackVolumeBackend	27
3.2.348 OpenStackVolumeBackendCeph	27
3.2.349 OpenStackVolumeBackendNFS	27
3.2.350 OpenStackVolumeBackupBackend	27
3.2.351 OpenStackVolumeBackupBackendCeph	27
3.2.352 OpenStackVolumeBackupRole	27
3.2.353 OpenStackVolumeRole	27

3.2.354 OpenStackVolumeSchedulerRole	27
3.2.355 OsapiPortIP	27
3.2.356 OsapiSecurityGroupRule	27
3.2.357 OsapiSubnetAllocationPool	27
3.2.358 OSService	27
3.2.359 OSServiceArray	27
3.2.360 OSServiceConfig	27
3.2.361 Partition	27
3.2.362 PBSJob	27
3.2.363 PBSJobQueue	27
3.2.364 PBSJobQueueStat	27
3.2.365 PbsProClientRole	27
3.2.366 PbsProJob	27
3.2.367 PbsProJobQueue	27
3.2.368 PbsProJobQueueStat	27
3.2.369 PbsProServerRole	27
3.2.370 PDUPort	27
3.2.371 PhysicalNode	27
3.2.372 PowerDistributionUnit	27
3.2.373 PowerStatus	27
3.2.374 Process	27
3.2.375 Processor	27
3.2.376 Profile	27
3.2.377 ProgramRunnerInput	27
3.2.378 ProgramRunnerKill	28
3.2.379 ProgramRunnerOutput	28
3.2.380 ProgramRunnerStatus	28
3.2.381 ProvisioningNodeStatus	28
3.2.382 ProvisioningProcessorJob	28
3.2.383 ProvisioningRequestStatus	28
3.2.384 ProvisioningRole	28
3.2.385 ProvisioningStatus	28
3.2.386 Puppet	28
3.2.387 PuppetApplyResult	28
3.2.388 PuppetApplySession	28
3.2.389 PuppetClass	28
3.2.390 PuppetRole	28
3.2.391 PuppetRunInfo	28
3.2.392 Rack	28
3.2.393 RackSensor	28
3.2.394 RadosGatewayRole	28
3.2.395 RateElem	28
3.2.396 ReadMonDataId	28
3.2.397 ReadMonDataOutput	28
3.2.398 RemoteMonConf	28
3.2.399 RemoteMonMetricConf	28

3.2.400 RemoteNodeInstallerInteraction	28
3.2.401 RemoteSetupExecution	28
3.2.402 RemoteThreshold	28
3.2.403 ResourcePool	28
3.2.404 ResourcePoolStatus	28
3.2.405 Role	28
3.2.406 RunJobAction	28
3.2.407 S3DataDownload	28
3.2.408 S3DataUpload	28
3.2.409 S3ResultsDownload	28
3.2.410 S3ResultsUpload	28
3.2.411 S3Transfer	28
3.2.412 Semaphore	28
3.2.413 Sensor	28
3.2.414 Session	28
3.2.415 SGEClientRole	28
3.2.416 SGEJob	28
3.2.417 SGEJobQueue	28
3.2.418 SGEJobQueueStat	28
3.2.419 SGEParallelEnvironment	28
3.2.420 SGEServerRole	28
3.2.421 SharedMemory	28
3.2.422 SlaveMonotonicElem	28
3.2.423 SlaveMonSnapshot	28
3.2.424 SlaveNode	28
3.2.425 SlaveRateElem	29
3.2.426 SlurmClientRole	29
3.2.427 SlurmJob	29
3.2.428 SlurmJobQueue	29
3.2.429 SlurmJobQueueStat	29
3.2.430 SlurmServerRole	29
3.2.431 SoftwareImage	29
3.2.432 SoftwareImageProxy	29
3.2.433 StartStorageNodeAction	29
3.2.434 StateElem	29
3.2.435 StaticRoute	29
3.2.436 StatisticData	29
3.2.437 StopStorageNodeAction	29
3.2.438 StorageNodePolicy	29
3.2.439 StorageRole	29
3.2.440 StringListObject	29
3.2.441 SubnetManagerRole	29
3.2.442 Switch	29
3.2.443 SwitchPort	29
3.2.444 SysInfoCollector	29
3.2.445 SysMonotonicWithId	29

3.2.446 SysRateWithId	29
3.2.447 ThreshAction	29
3.2.448 ThreshActionConf	29
3.2.449 Threshold	29
3.2.450 Ticket	29
3.2.451 TorqueClientRole	29
3.2.452 TorqueJob	29
3.2.453 TorqueJobQueue	29
3.2.454 TorqueJobQueueStat	29
3.2.455 TorqueServerRole	29
3.2.456 UCSAdaptorEthCompQueueProfile	29
3.2.457 UCSAdaptorEthGenProfile	29
3.2.458 UCSAdaptorEthInterruptProfile	29
3.2.459 UCSAdaptorEthOffloadProfile	29
3.2.460 UCSAdaptorEthRecvQueueProfile	29
3.2.461 UCSAdaptorEthUSNICProfile	29
3.2.462 UCSAdaptorEthWorkQueueProfile	29
3.2.463 UCSAdaptorExtEthIf	29
3.2.464 UCSAdaptorExtIpV6RssHashProfile	29
3.2.465 UCSAdaptorFcCdbWorkQueueProfile	29
3.2.466 UCSAdaptorFcErrorRecoveryProfile	29
3.2.467 UCSAdaptorFcGenProfile	29
3.2.468 UCSAdaptorFcInterruptProfile	29
3.2.469 UCSAdaptorFcPortFLogiProfile	29
3.2.470 UCSAdaptorFcPortPLogiProfile	29
3.2.471 UCSAdaptorFcPortProfile	29
3.2.472 UCSAdaptorFcRecvQueueProfile	30
3.2.473 UCSAdaptorFcWorkQueueProfile	30
3.2.474 UCSAdaptorHostEthIf	30
3.2.475 UCSAdaptorHostFcIf	30
3.2.476 UCSAdaptorIpV4RssHashProfile	30
3.2.477 UCSAdaptorIpV6RssHashProfile	30
3.2.478 UCSAdaptorPortProfiles	30
3.2.479 UCSAdaptorRssProfile	30
3.2.480 UCSBase	30
3.2.481 UCSBiosBootDev	30
3.2.482 UCSBiosBootDevGrp	30
3.2.483 UCSBiosSettings	30
3.2.484 UCSBiosVfAdjacentCacheLinePrefetch	30
3.2.485 UCSBiosVfAltitude	30
3.2.486 UCSBiosVfASPMsSupport	30
3.2.487 UCSBiosVfConsoleRedirection	30
3.2.488 UCSBiosVfCoreMultiProcessing	30
3.2.489 UCSBiosVfCPUEnergyPerformance	30
3.2.490 UCSBiosVfCPUFrequencyFloor	30
3.2.491 UCSBiosVfCPUPerformance	30

3.2.492 UCSBiosVfCPUPowerManagement	30
3.2.493 UCSBiosVfDCUPrefetch	30
3.2.494 UCSBiosVfDemandScrub	30
3.2.495 UCSBiosVfDirectCacheAccess	30
3.2.496 UCSBiosVfDRAMClockThrottling	30
3.2.497 UCSBiosVfDramRefreshRate	30
3.2.498 UCSBiosVfEnhancedIntelSpeedStepTech	30
3.2.499 UCSBiosVfExecuteDisableBit	30
3.2.500 UCSBiosVfFRB2Enable	30
3.2.501 UCSBiosVfHardwarePrefetch	30
3.2.502 UCSBiosVfIntelHyperThreadingTech	30
3.2.503 UCSBiosVfIntelTurboBoostTech	30
3.2.504 UCSBiosVfIntelVirtualizationTechnology	30
3.2.505 UCSBiosVfIntelVTForDirectedIO	30
3.2.506 UCSBiosVfLegacyUSBSupport	30
3.2.507 UCSBiosVfLOMPortOptionROM	30
3.2.508 UCSBiosVfLvDIMMSupport	30
3.2.509 UCSBiosVfMemoryInterleave	30
3.2.510 UCSBiosVfMemoryMappedIOAbove4GB	30
3.2.511 UCSBiosVfNUMAOptimized	30
3.2.512 UCSBiosVfOnboardStorage	30
3.2.513 UCSBiosVfOnboardStorageSWStack	30
3.2.514 UCSBiosVfOSBootWatchdogTimer	30
3.2.515 UCSBiosVfOSBootWatchdogTimerPolicy	30
3.2.516 UCSBiosVfOSBootWatchdogTimerTimeout	30
3.2.517 UCSBiosVfPatrolScrub	30
3.2.518 UCSBiosVfPCIOptionROMs	30
3.2.519 UCSBiosVfPCISlotOptionROMEnable	31
3.2.520 UCSBiosVfProcessorC1E	31
3.2.521 UCSBiosVfProcessorC6Report	31
3.2.522 UCSBiosVfPStateCoordType	31
3.2.523 UCSBiosVfQPIConfig	31
3.2.524 UCSBiosVfSelectMemoryRASConfiguration	31
3.2.525 UCSBiosVfTPMSupport	31
3.2.526 UCSBiosVfUCSMBootOrderRuleControl	31
3.2.527 UCSBiosVfUSBEmulation	31
3.2.528 UCSBiosVfUSBPortsConfig	31
3.2.529 UCSBiosVfVgaPriority	31
3.2.530 UCSCommNtpProvider	31
3.2.531 UCSCommSyslog	31
3.2.532 UCSCommSyslogClient	31
3.2.533 UCSEquipmentIndicatorLed	31
3.2.534 UCSEquipmentLocatorLed	31
3.2.535 UCSFaultInst	31
3.2.536 UCSFirmwareRunning	31
3.2.537 UCSInfo	31

3.2.538 UCSLogs	31
3.2.539 UCSLsbootDef	31
3.2.540 UCSLsbootEfi	31
3.2.541 UCSLsbootLan	31
3.2.542 UCSLsbootStorage	31
3.2.543 UCSLsbootVirtualMedia	31
3.2.544 UCSStatus	31
3.2.545 UGECgroupsSettings	31
3.2.546 UGEClientRole	31
3.2.547 UGEJob	31
3.2.548 UGEJobQueue	31
3.2.549 UGEJobQueueStat	31
3.2.550 UGEParallelEnvironment	31
3.2.551 UGEServerRole	31
3.2.552 User	31
3.2.553 Validation	31
3.2.554 VersionInfo	31
3.2.555 VirtualNode	31
3.2.556 VirtualNodeSettings	31
3.2.557 VirtualSMPNode	31
3.2.558 VScaleMPSettings	31
3.2.559 VsmpSettings	31
3.2.560 WillChange	31
3.2.561 XeonPhiSettings	31
3.3 JSON Examples	31

Preface

Welcome to the *Developer Manual* for Bright Cluster Manager 7.2.

0.1 About This Manual

This manual is aimed at helping developers who would like to program the Bright Cluster Manager in order to enhance or alter its functionality. It is not intended for end users who simply wish to submit jobs that run programs to workload managers, which is discussed in the *User Manual*. The developer is expected to be reasonably familiar with the parts of the *Administrator Manual* that is to be dealt with—primarily CMDaemon, of which `cmsh` and `cmgui` are the front ends.

This manual discusses the Python API to CMDaemon, and also covers how to program for metric collections.

0.2 About The Manuals In General

Regularly updated versions of the Bright Cluster Manager 7.2 manuals are available on updated clusters by default at `/cm/shared/docs/cm`. The latest updates are always online at <http://support.brightcomputing.com/manuals>.

- The *Administrator Manual* describes the general management of the cluster.
- The *Installation Manual* describes installation procedures for a basic cluster.
- The *User Manual* describes the user environment and how to submit jobs for the end user.
- The *Cloudbursting Manual* describes how to deploy the cloud capabilities of the cluster.
- The *Developer Manual* has useful information for developers who would like to program with Bright Cluster Manager.
- The *OpenStack Deployment Manual* describes how to deploy OpenStack with Bright Cluster Manager.
- The *Hadoop Deployment Manual* describes how to deploy Hadoop with Bright Cluster Manager.
- The *UCS Deployment Manual* describes how to deploy the Cisco UCS server with Bright Cluster Manager.

If the manuals are downloaded and kept in one local directory, then in most pdf viewers, clicking on a cross-reference in one manual that refers to a section in another manual opens and displays that section in the second manual. Navigating back and forth between documents is usually possible with keystrokes or mouse clicks.

For example: `<Alt>-<Backarrow>` in Acrobat Reader, or clicking on the bottom leftmost navigation button of `xpdf`, both navigate back to the previous document.

The manuals constantly evolve to keep up with the development of the Bright Cluster Manager environment and the addition of new hardware and/or applications. The manuals also regularly incorporate customer feedback. Administrator and user input is greatly valued at Bright Computing. So any comments, suggestions or corrections will be very gratefully accepted at manuals@brightcomputing.com.

0.3 Getting Administrator-Level Support

If the reseller from whom Bright Cluster Manager was bought offers direct support, then the reseller should be contacted.

Otherwise the primary means of support is via the website <https://support.brightcomputing.com>. This allows the administrator to submit a support request via a web form, and opens up a trouble ticket. It is a good idea to try to use a clear subject header, since that is used as part of a reference tag as the ticket progresses. Also helpful is a good description of the issue. The followup communication for this ticket goes via standard e-mail. Section 11.2 of the *Administrator Manual* has more details on working with support.

0.4 Getting Developer-Level Support

Developer support is given free, within reason. For more extensive support, Bright Computing can be contacted in order to arrange a support contract.

1

Bright Cluster Manager Python API

This chapter introduces the Python API of Bright Cluster Manager. For a head node `bright72`, the API reference documentation for all available objects is available in a default cluster via browser access to the URL:

```
https://bright72/userportal/downloads/python
```

The preceding access is via the User Portal (section 10.9 of the *Administrator Manual*).

The documentation is also available directly on the head node itself at:

```
file:///cm/local/docs/cmd/python/index.html
```

1.1 Installation

The Python cluster manager bindings are pre-installed on the head node.

1.1.1 Windows Clients

For windows clients, Python version 2.5.X is needed. Newer versions of Python do not work with the API.

For Windows a redistributable package is supplied in the `pythoncm-dist` RPM package installed on the cluster. The file at

```
/cm/shared/apps/pythoncm/dist/windows-pythoncm.7.2.r15673.zip
```

—the exact version number may differ—is copied to the Windows PC and unzipped.

A Windows shell (`cmd.exe`) is opened to the directory where the Python bindings are. The `headnodeinfo.py` example supplied with the unzipped files has a line that has the following format:

```
cluster = clustermanager.addCluster(<parameters>);
```

where `<parameters>` is either:

```
'<URL>', '<PEMAuth1>', '<PEMAuth2>'
```

or

```
'<URL>', '<PFXauth>', '"', '<password>'
```

The `<parameters>` entry is edited as follows:

- the correct hostname is set for the `<URL>` entry. By default it is set to `https://localhost:8081`
- If PEM key files are to be used for client authentication,

- `<PEMauth1>` is set to path of `cert.pem`
- `<PEMauth2>` is set to the path of `cert.key`
- If a PFX file is used for client authentication,
 - `<PFXauth>` is set to path of `admin.pfx`
 - `<password>` is set to the password

To verify everything is working, it can be run as follows:

```
c:\python25\python headnodeinfo.py
```

1.1.2 Linux Clients

For Linux clients, a redistributable source package is supplied in the `pythoncm-dist` package installed on the cluster. The file at `/cm/shared/apps/pythoncm/dist/pythoncm-7.2-r18836-src.tar.bz2`—the exact version number may differ—is copied and untarred to any directory.

The `build.sh` script is then run to compile the source. About 4GB of memory is usually needed for compilation, and additional packages may be required for compilation to succeed. A list of packages needed to build Python cluster manager bindings can be found in the `README` file included with the package.

The `headnodeinfo.py` example supplied with the untarred files is edited as for in the earlier windows client example, for the `clustermanager.addCluster` line.

The path to the remote cluster manager library is added:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:remotecm
```

To verify everything is working, the following can be run:

```
python ./headnodeinfo.py
```

1.2 Examples

A set of examples can be found in `/cm/local/examples/cmd/python/` on the head node of the cluster.

1.2.1 First Program

A Python script is told to use the cluster manager bindings by importing `pythoncm` at the start of the script:

```
import pythoncm
```

If not working on the cluster, the administrator needs to set the path where the shared libraries can be found (`pythoncm.so` in Linux, or `python.pyd` in windows). This is done by adding the following to the start of the script:

```
import sys
sys.path.append(".") # path to pythoncm.so/python.pyd
```

Python cluster manager bindings allow for simultaneous connections to several clusters. For this reason the first thing to do is to create a `ClusterManager` object:

```
clustermanager = pythoncm.ClusterManager()
```

A connection to a cluster can now be made. There are two possible ways of connecting.

The first is using the certificate and private key file that `cmsh` uses by default when it authenticates from the head node.

```
cluster = clustermanager.addCluster('https://mycluster:8081', \
'/root/.cm/admin.pem', '/root/.cm/admin.key');
```

The second way uses the password protected `admin.pfx` file, which is generated with the `cmd -c` command. A Python script could ask for the password and store it in a variable for increased security.

```
cluster = clustermanager.addCluster('https://mycluster:8081', \
'/root/.cm/cmgui/admin.pfx', '', '<password>');
```

Having defined the cluster, a connection can now be made to it:

```
isconnected = cluster.connect()
if !isconnected:
    print "Unable to connect"
    print cluster.getLastError()
    exit(1)
```

If a connection cannot be made, the function `cluster.connect()` returns false. The function `cluster.getLastError()` shows details about the problem. The two most likely problems are due to a wrong password setting or a firewall settings issue.

Similar to `cmgui` and `cmsh`, the cluster object contains a local cache of all objects. This cache will be filled automatically when the connection is established. All changes to properties will be done on these local copies and will be lost after the Python scripts exits, unless a `commit` operation is done.

The most common operation is finding specific objects in the cluster.

```
active = cluster.find('active')
if active == None:
    print "Unable to find active head node"
    exit(1)
else:
    print "Hostname of the active head node is %s" % active.hostname
```

If creating an automated script that runs at certain times, then it is highly recommended to check if objects can be found. During a failover, for instance, there will be a period over a few minutes in which the active head node will not be set.

It is good practice to disconnect from the cluster at the end of the script.

```
cluster.disconnect()
```

When connecting to a cluster with a failover setup, it is the shared IP address that should be connected to, and not the fixed IP address of either of the head nodes.

1.3 Methods And Properties

1.3.1 Viewing All Properties And Methods

All properties visible in `cmsh` and `cmgui` are also accessible from Python cluster manager bindings. The easiest way to get an overview of the methods and properties of an object is to define the following function:

```
import re
def dump(obj):
    print "--- DUMP ---"
    for attr in dir(obj):
        p = re.compile('^__.*__$')
        if not p.match(attr):
            print "%s = %s" % (attr, getattr(obj, attr))
```

An overview of all properties and methods for the active head node can be obtained with:

```
active = cluster.find('active')
dump(active)
```

1.3.2 Property Lists

Most properties are straightforward and their names are almost identical to the `cmsh` equivalent.

For instance:

```
node.mac = '00:00:00:00:00:00'
category.softwareimage = cluster.find('testimage')
```

Properties that contain lists, like `node.roles`, `node.interfaces`, `category.fsmounts` and several others, are trickier to deal with. While iterating over a list property is simple enough:

```
for role in node.roles:
    print role.name
```

because of an implementation restriction, adding a new role requires that a local copy of the roles list be made:

```
roles = node.roles
provisioningrole = pythoncm.ProvisioningRole() # Create a new pro\
                                              visioning role object
roles.append(provisioningrole)
node.roles = roles # This will update the internal\
                  roles list with the local copy
```

1.3.3 Creating New Objects

Creating a new node can be done with:

```
node = pythoncm.Node()
```

This is valid command, but fairly useless because a node has to be a `MasterNode`, `PhysicalNode` or `VirtualSMPNode`. So to create a normal compute or login node, the object is created as follows:

```
node = pythoncm.PhysicalNode()
```

The first thing to do after creating a new object is to add it to a cluster.

```
cluster.add(node)
```

It is impossible to add one node to more than one cluster.

After the node has been added its properties can be set. In `cmsh` and `cmgui` this is semi-automated, but in Python cluster manager bindings it has to be done by hand.

```
node.hostname = 'node001'
node.partition = cluster.find('base')
node.category = cluster.find('default')
```

Similar to the `node` object, a `NetworkInterface` object has several subtypes: `NetworkPhysicalInterface`, `NetworkVLANInterface`, `NetworkAliasInterface`, `NetworkBondInterface`, and `NetworkIPMIInterface`.

```
interface = pythoncm.NetworkPhysicalInterface()
interface.name = 'eth0'
interface.ip = '10.141.0.1'
interface.network = cluster.find('internalnet')
node.interfaces = [interface]
node.provisioningInterface = interface
```

Having set the properties of the new node, it can now be committed.

```
cr = node.commit()
```

If a commit fails for some reason, the reason can be found:

```
if not cr.result:
    print "Commit of %s failed:" % node.resolveName()
    for j in range(cr.count):
        print cr.getValidation(j).msg
```

1.3.4 List Of Objects

In the following lists of objects:

- Objects marked with (*) cannot be used
- Trees marked with (+) denote inheritance

Roles

```
Role (*)
+ BackupRole
+ BootRole
+ DatabaseRole
+ EthernetSwitch
+ LoginRole
+ LSFClientRole
+ LSFServerRole
+ MasterRole
+ PbsProClientRole
+ PbsProServerRole
+ ProvisioningRole
+ SGEClientRole
+ SGEServerRole
+ SlurmClientRole
+ SlurmServerRole
+ SubnetManagerRole
+ TorqueClientRole
+ TorqueServerRole
```

Devices

```
Device (*)
+ Chassis
+ GpuUnit
+ GenericDevice
+ PowerDistributionUnit
+ Switch (*)
  + EthernetSwitch
  + IBSwitch
  + MyrinetSwitch
Node (*)
+ FSExport
+ FSMount
+ MasterNode
+ SlaveNode (*)
  + PhysicalNode
  + VirtualSMPNode
```

Network Interfaces

```
NetworkInterface (*)
+ NetworkAliasInterface
+ NetworkBondInterface
```

- + NetworkIpmiInterface
- + NetworkPhysicalInterface
- + NetworkVLANInterface

Information Objects

- ClusterSetup
- GuiClusterOverview
- GuiCephOverview
- GuiHadoopHDFSOverview
- GuaOpenStackOverview
- GuiOpenStackTenantOverview
- GuiGpuUnitOverview
- GuiNodeOverview
- GuiNodeStatus
- LicenseInfo
- SysInfoCollector
- VersionInfo

Monitoring Configuration Objects

- MonConf
- ConsolidatorConf
- MonHealthConf
- HealthCheck
- MonMetricConf
- ThreshActionConf
- ThreshAction
- Threshold

LDAP Objects

- User
- Group

Category Objects

- Category
- FSExport
- FSMount

Miscellaneous Objects

- SoftwareImage

- KernelModule

- Network

- NodeGroup

- Partition
- + BurnConfig
- Rack

1.3.5 Useful Methods

For The Cluster Object:

Name	Description
<code>find(<name>)</code>	Find the object with a given name, <i><name></i>
<code>find(<name>, <type>)</code>	Because it is possible to give a category and node the same name, sometimes the type <i><type></i> of the object needs to be specified too
<code>getAll(<type>)</code>	Get a list of all objects of a given type: e.g. device, category
<code>activeMaster()</code>	Get the active master object
<code>passiveMaster()</code>	Get the active master object
<code>overview()</code>	Get all the data shown in the <code>cmgui</code> cluster overview
<code>add(<object>)</code>	Add a newly created object <i><object></i> to the cluster. Only after an object is added can it be used
<code>pexec(<nodes>, <command>)</code>	Execute a command <i><command></i> on one or more nodes

For Any Object:

Name	Description
<code>commit()</code>	Save changes to the cluster
<code>refresh()</code>	Undo all changes and restore the object to its last saved state
<code>remove()</code>	Remove an object from the cluster
<code>clone()</code>	Make an identical copy. The newly created object is not added to a cluster yet

For Any Device:

Name	Description
<code>close()</code>	Close a device
<code>open()</code>	Open a device
<code>powerOn()</code>	Power on a device
<code>powerOff()</code>	Power off a device
<code>powerReset()</code>	Power reset a device
<code>latestMonitoringData()</code>	Return a list of the most recent monitoring data

For Any Node:

Name	Description
<code>overview()</code>	Get the data displayed in the <code>cmgui</code> node overview tab
<code>sysinfo()</code>	Get the data displayed in the <code>cmgui</code> node system information tab
<code>pexec(<command>)</code>	Execute a command

1.3.6 Useful Example Program

In the directory `/cm/local/examples/cmd/python` are some example programs using the python API.

One of these is `printall.py`. It displays values for objects in an easily viewed way. With `all` as the argument, it displays resource objects defined in a list in the program. The objects are 'Partition', 'MasterNode', 'SlaveNode', 'Category', 'SoftwareImage', 'Network', 'NodeGroup'. The output is displayed something like (some output elided):

Example

```
[root@bright72 ~]# cd /cm/local/examples/cmd/python
[root@bright72 python]# ./printall all
Partition base
+- revision .....
| name ..... base
| clusterName ..... Bright 7.2 Cluster
...
| burnConfigs
| +- revision .....
| | name ..... default
| | description ..... Standard burn test.
| | configuration ..... < 2780 bytes >
| +- revision .....
| | name ..... long-hpl
...
| provisioningInterface ..... None
| fsmounts ..... < none >
| fsexports
| +- revision .....
| | name ..... /cm/shared@internalnet
| | path ..... /cm/shared
| | hosts ..... !17179869185!
...
Category default
+- revision .....
| name ..... default
| softwareImage ..... default-image
| defaultGateway ..... 10.141.255.253
| nameServers ..... < none >
...
```

The values of a particular resource-level object, such as `softwareimage`, can be viewed by specifying it as the argument:

Example

```
[root@bright72 python]# ./printall.py softwareimage
softwareimage default-image
+- revision .....
| name ..... default-image
| path ..... /cm/images/default-image
| originalImage ..... 0
| kernelVersion ..... 2.6.32-431.11.2.el6.x86_64
| kernelParameters ..... rdblacklist=nouveau
| creationTime ..... 1398679806
| modules
| +- revision .....
| | name ..... xen-netfront
...
```

```
| +- revision .....  
| | name ..... hpilo  
| | parameters .....  
| enableSOL ..... False  
| SOLPort ..... ttyS1  
| SOLSpeed ..... 115200  
| SOLFlowControl ..... True  
| notes .....  
| fspart ..... 98784247812  
| bootfspart ..... 98784247813  
...  
[root@bright72 python]#
```


2

Metric Collections

This chapter gives details on metric collections.

Section 10.4.4 of the *Administrator Manual* introduces metric collections, and describes how to add a metric collections script with `cmgui`.

This chapter covers how to add a metric collections script with `cmsh`. It also describes the output specification of a metric collections script, along with example outputs, so that a metric collections script can be made by the administrator.

2.1 Metric Collections Added Using `cmsh`

A metric collections script, `responsiveness`, is added in the `monitoring metrics` mode just like any other metric.

Example

```
[bright72]% monitoring metrics
[bright72->monitoring->metrics]% add responsiveness
[...[responsiveness]]% set command /cm/local/apps/cmd/scripts/metrics/s\
ample_responsiveness
[...*[responsiveness*]]% set classofmetric prototype; commit
```

For `classofmetric`, the value `prototype` is the class used to distinguish metric collections from normal metrics.

2.2 Metric Collections Initialization

When a metric collections script is added to `CMDaemon` for the first time, `CMDaemon` implicitly runs it with the `--initialize` flag. The output is used to define the collections table header structure. The structure is composed of the component metrics in the collections script, and the resulting structure is placed in the `CMDaemon` monitoring database. After the initialization step, data values can be added to the collections table during regular use of the script.

The displayed output of a metric collections script when using the `--initialize` flag is a list of available metrics and their parameter values. The format of each line in the list is:

```
metric <name[:parameter]> <unit> <class> "<description>" <cumulative> <min> <max>
```

where the items in the line are:

- `metric`: A bare word.
- `<name[:parameter]>`: The name of the metric, with for certain metrics a parameter value. For example, the metric `AlertLevel` can have the parameter `sum` assigned to it with the ":" character.

- `<unit>`: The unit of measurement that the metric uses.
- `<class>`: Any of: misc cpu disk memory network environmental operatingsystem internal workload cluster.
- `<description>`: This can contain spaces, but should be enclosed with quotes.
- `<cumulative>`: Either `yes` or `no`. This indicates whether the metric increases monotonically (e.g., bytes received) or not (e.g., temperature).
- `<min>` and `<max>`: The minimum and maximum numeric values of this metric are determined dynamically based on the values so far.

Example

```
[root@myheadnode metrics]# ./sample_responsiveness --initialize
metric util_sda % internal "Percentage of CPU time during which I/O
requests were issued to device sda" no 0 100
metric await_sda ms internal "The average time (in milliseconds) for
I/O requests issued to device sda to be served" no 0 500
```

2.3 Metric Collections Output During Regular Use

The output of a metric collection script without a flag is a list of outputs from the available metrics. The format of each line in the list is:

```
metric <name[:parameter]> <value> [infomessage]
```

where the parameters to the `metric` bare word are:

- `<name[:parameter]>`: The name of the metric, with optional parameter for some metrics.
- `<value>`: The numeric value of the measurement.
- `[infomessage]`: An optional infomessage.

Example

```
[root@myheadnode metrics]# ./sample_responsiveness
metric await_sda 0.00
metric util_sda 0.00
[root@myheadnode metrics]#
```

If the output has more metrics than that suggested by when the `--initialize` flag is used, then the extra sampled data is discarded. If the output has fewer metrics, then the metrics are set to NaN (not a number) for the sample.

A metric or health check inside a metric collection appears as a check when viewing metrics or healthcheck lists. Attempting to remove such a check specifically using `cmsh` or `cmgui` only succeeds until the node is updated or rebooted. It is the metric collection itself that should have the check removed from within it, in order to remove the check from the list of checks permanently.

Setting a node that is UP to a CLOSED state, and then bringing it out of that state with the `open` command (section 5.5.4 of the *Administrator Manual*) also has CMDaemon run the metric collections script with the `--initialize` flag. This is useful for allowing CMDaemon to re-check what metrics in the collections can be sampled, and then re-configure them.

2.4 Metric Collections Error Handling

If the exit code of the script is 0, CMDaemon assumes that there is no error. So, with the `--initialize` flag active, despite no numeric value output, the script does not exit with an error.

If the exit code of the script is non-zero, the output of the script is assumed to be a diagnostic message and is passed to the head node. This shows up as an event in `cmsh` or `cmgui`.

For example, the `sample_ipmi` script uses the `ipmi-sensors` binary internally. Calling the binary directly returns an error code if the device has no IPMI configured. However, the `sample_ipmi` script in this case simply returns 0, and no output. The rationale here being that the administrator is aware of this situation and would not expect data from that IPMI anyway, let alone an error.

2.5 Metric Collections Consolidator Syntax

Metric collections can have a consolidator format defined per metric. The consolidator definition must be placed as an output in the line immediately preceding the corresponding metric initialization output line. The consolidator definition line can take the following forms:

```
consolidators default
consolidators none
consolidators CONSOLIDATORNAME FORMAT SPECIFICATION
```

The meanings of the texts after `consolidators` are as follows:

- `default`: The metrics follow the default consolidator names and interval values (section 10.7.4, page 409 of the *Administrator Manual*). That is, consolidator names take the value of `Hour`, `Day`, `Week`, while the interval values are the corresponding durations in seconds.
- `consolidators none`: No consolidation is done, only raw data values are collected for the metrics.
- `CONSOLIDATORNAME FORMAT SPECIFICATION`: This has the form:
`<name: interval [:kind [:tablelength]]> . . .`
 - `name`: the consolidator name. A special feature here is that it can also define a new consolidator if the name does not already exist. Multiple consolidators can be defined in each consolidator definition line, with `name` separated from any preceding definition on the same line by a space.
 - `interval`: the duration in seconds, between consolidation, for the consolidator.
 - `kind`: an optional value of `min`, `max`, or `average`. By default it is `average`.
 - `tablelength`: an optional value for the length of the table, if `kind` has been specified. By default it is 1000.

2.6 Metric Collections Environment Variables

The following environment variables are available for a metric collection script, as well as for custom scripts, running from CMDaemon:

On all devices:

`CMD_HOSTNAME`: name of the device. For example:

```
CMD_HOSTNAME=myheadnode
```

Only on non-node devices:

CMD_IP: IP address of the device. For example:

```
CMD_IP=192.168.1.33
```

Only on node devices:

Because these devices generally have multiple interfaces, the single environment variable CMD_IP is often not enough to express these. Multiple interfaces are therefore represented by these environment variables:

- CMD_INTERFACES: list of names of the interfaces attached to the node. For example:

```
CMD_INTERFACES=eth0 eth1 ipmi0 BOOTIF
```

- CMD_INTERFACE_<interface>_IP: IP address of the interface with the name <interface>. For example:

```
CMD_INTERFACE_eth0_IP=10.141.255.254
CMD_INTERFACE_eth1_IP=0.0.0.0
```

- CMD_INTERFACE_<interface>_TYPE: type of interface with the name <interface>. For example:

```
CMD_INTERFACE_eth1_TYPE=NetworkPhysicalInterface
CMD_INTERFACE_ipmi0_TYPE=NetworkBmcInterface
```

Possible values are:

- NetworkBmcInterface
- NetworkPhysicalInterface
- NetworkVLANInterface
- NetworkAliasInterface
- NetworkBondInterface
- NetworkBridgeInterface
- NetworkTunnelInterface
- NetworkNetMapInterface

- CMD_BMCUSERNAME: username for the BMC device at this node (if available).
- CMD_BMCPASSWORD: password for the BMC device at this node (if available).

To parse the above information to get the BMC IP address of the node for which this script samples, one could use (in Perl):

```
my $ip;
my $interfaces = $ENV{"CMD_INTERFACES"};
foreach my $interface ( split( " ", $interfaces ) ) {
    if( $ENV{"CMD_INTERFACE_" . $interface . "_TYPE"} eq
"NetworkBmcInterface" ) {
        $ip = $ENV{"CMD_INTERFACE_" . $interface . "_IP"};
        last;
    }
}
# $ip holds the bmc ip
```


A list of environment variables available under the CMDaemon environment can be found by running a script under CMDaemon and exporting the environment variables to a file for viewing. For example, the `/cm/local/apps/cmd/scripts/healthchecks/testhealthcheck` script can be modified and run to sample on the head node, with the added line: `set>/tmp/environment`. The resulting file `/tmp/environment` that is generated as part of the healthcheck run then includes the `CMD_*` environment variables.

Example

```
CMD_BMCPASSWORD
CMD_BMCUSERNAME
CMD_CLUSTERNAME
CMD_CMDSTARTEDTIME
CMD_DEVICE_TYPE
CMD_EXPORTS
CMD_FSEXPRT__SLASH_cm_SLASH_shared_AT_internalnet_ALLOWWRITE
CMD_FSEXPRT__SLASH_cm_SLASH_shared_AT_internalnet_HOSTS
CMD_FSEXPRT__SLASH_cm_SLASH_shared_AT_internalnet_PATH
CMD_FSEXPRT__SLASH_home_AT_internalnet_ALLOWWRITE
CMD_FSEXPRT__SLASH_home_AT_internalnet_HOSTS
CMD_FSEXPRT__SLASH_home_AT_internalnet_PATH
CMD_FSEXPRT__SLASH_var_SLASH_spool_SLASH_burn_AT_internalnet_ALLOWWRITE
CMD_FSEXPRT__SLASH_var_SLASH_spool_SLASH_burn_AT_internalnet_HOSTS
CMD_FSEXPRT__SLASH_var_SLASH_spool_SLASH_burn_AT_internalnet_PATH
CMD_HOSTNAME
CMD_INTERFACES
CMD_INTERFACE_eth0_IP
CMD_INTERFACE_eth0_MTU
CMD_INTERFACE_eth0_SPEED
CMD_INTERFACE_eth0_STARTIF
CMD_INTERFACE_eth0_TYPE
CMD_INTERFACE_eth1_IP
CMD_INTERFACE_eth1_MTU
CMD_INTERFACE_eth1_SPEED
CMD_INTERFACE_eth1_STARTIF
CMD_INTERFACE_eth1_TYPE
CMD_IP
CMD_MAC
CMD_METRICNAME
CMD_METRICPARAM
CMD_MOUNTS
CMD_NODEGROUPS
CMD_PARTITION
CMD_PORT
CMD_PROTOCOL
CMD_ROLES
CMD_SCRIPTTIMEOUT
CMD_STATUS
CMD_STATUS_CLOSED
CMD_STATUS_HEALTHCHECK_FAILED
CMD_STATUS_HEALTHCHECK_UNKNOWN
CMD_STATUS_MESSAGE
CMD_STATUS_RESTART_REQUIRED
CMD_STATUS_STATEFLAPPING
CMD_STATUS_USERMESSAGE
CMD_SYSINFO_SYSTEM_MANUFACTURER
```

```
CMD_SYSINFO_SYSTEM_NAME
CMD_USERDEFINED1
CMD_USERDEFINED2
```

2.7 Metric Collections Examples

Bright Cluster Manager has several scripts in the `/cm/local/apps/cmd/scripts/metrics` directory. Among them are the metric collections scripts `testmetriccollection` and `sample_responsiveness`. A glance through them while reading this chapter may be helpful.

2.8 Metric Collections On iDataPlex And Similar Units

IBM's iDataPlex is a specially engineered dual node rack unit. When the term iDataPlex is used in the following text in this section, it also implies any other dual node units that show similar behavior.

This section gives details on configuring an iDataPlex if IPMI metrics retrieval seems to skip most IPMI values from one of the nodes in the unit.

When carrying out metrics collections on an iDataPlex unit, Bright Cluster Manager should work without any issues. However, it may be that due to the special paired node design of an iDataPlex unit, most IPMI metrics of one member of the pair are undetectable by the `sample_ipmi` script sampling on that particular node. The missing IPMI metrics can instead be retrieved from the second member in the pair (along with the IPMI metrics of the second member).

The output may thus look something like:

Example

```
[root@master01 ~]# cmsh
[master01]% device latestmetricdata node181 | grep Domain
Metric                               Value
-----
Domain_A_FP_Temp                     23
Domain_A_Temp1                       39
Domain_A_Temp2                       37
Domain_Avg_Power                     140
Domain_B_FP_Temp                     24
Domain_B_Temp1                       40
Domain_B_Temp2                       37
[master01]% device latestmetricdata node182 | grep Domain
Metric                               Value
-----
Domain_A_FP_Temp                     no data
Domain_A_Temp1                       no data
Domain_A_Temp2                       no data
Domain_Avg_Power                     170
Domain_B_FP_Temp                     no data
Domain_B_Temp1                       no data
Domain_B_Temp2                       no data
[master01]%
```

Because there are usually many iDataplex units in the rack, the metrics retrieval response of each node pair in a unit should be checked for this behavior.

The issue can be dealt with by Bright Cluster Manager by modifying the configuration file for the `sample_ipmi` script in `/cm/local/apps/cmd/scripts/metrics/configfiles/sample_ipmi.conf`. Two parameters that can be configured there are `chassisContainsLeadNode` and `chassisContainsLeadNodeRegex`.

- Setting `chassisContainsLeadNode` to `on` forces the `sample_ipmi` script to treat the unit as an iDataPlex unit.

In particular:

- `auto` (recommended) means the unit is checked by the IPMI metric sample collection script for whether it behaves like an iDataPlex unit.
 - `on` means the unit is treated as an iDataPlex node pair, with one node being a lead node that has all the IPMI metrics.
 - `off` means the unit is treated as a non-iDataPlex node pair, with each node having normal behavior when retrieving IPMI metrics. This setting may need to be used in case the default value of `auto` ever falsely detects a node as part of an iDataPlex pair.
- The value of `chassisContainsLeadNodeRegex` can be set to a regular expression pattern that matches the system information pattern for the name, as obtained by `CMDaemon` for an iDataPlex unit (or similar clone unit). The pattern that it is matched against is the output of:

```
cmsh -c 'device ; sysinfo master | grep "^System Name"'
```

If the pattern matches, then the IPMI sample collection script assumes the unit behaves like an iDataPlex dual node pair. The missing IPMI data values are then looked for on the lead node.

The value of `chassisContainsLeadNodeRegex` is set to `iDataPlex` by default.

3

Bright Cluster Manager JSON API

This chapter gives an alphabetical list of the JSON API services and entities available for Bright Cluster Manager. The API reference documentation for all available services and entities is available on the head node at:

`/cm/local/docs/cmd/json/index.html`.

Some examples of JSON use are given in section 3.3

3.1 Services

- 3.1.1 **auth**
- 3.1.2 **ceph**
- 3.1.3 **cert**
- 3.1.4 **cloud**
- 3.1.5 **device**
- 3.1.6 **gui**
- 3.1.7 **hadoop**
- 3.1.8 **job**
- 3.1.9 **keyvalue**
- 3.1.10 **lustre**
- 3.1.11 **main**
- 3.1.12 **mon**
- 3.1.13 **net**
- 3.1.14 **openstack**
- 3.1.15 **part**
- 3.1.16 **proc**
- 3.1.17 **prov**
- 3.1.18 **puppet**
- 3.1.19 **serv**
- 3.1.20 **session**
- 3.1.21 **test**
- 3.1.22 **ticket**
- 3.1.23 **user**

3.2 Entities

- 3.2.1 **BackupRole**

- 3.2.2 **BadEntityManagers**
- 3.2.3 **BasicResource**
- 3.2.4 **BillingHistory**
- 3.2.5 **BootRole**
- 3.2.6 **BurnConfig**
- 3.2.7 **BurnStatus**
- 3.2.8 **BurnTestStatus**
- 3.2.9 **Category**
- 3.2.10 **Ceph**
- 3.2.11 **CephMonitorRole**
- 3.2.12 **CephOSDAssociation**
- 3.2.13 **CephOSDPool**
- 3.2.14 **CephOSDRole**
- 3.2.15 **CephState**
- 3.2.16 **Certificate**
- 3.2.17 **CertificateRequest**
- 3.2.18 **Chassis**
- 3.2.19 **CloudDirectorRole**
- 3.2.20 **CloudGatewayRole**
- 3.2.21 **CloudImage**
- 3.2.22 **CloudJobDescription**
- 3.2.23 **CloudJobSubmissionStatus**
- 3.2.24 **CloudNode**
- 3.2.25 **CloudPrivateCloud**
- 3.2.26 **CloudProvider**
- 3.2.27 **CloudRegion**
- 3.2.28 **CloudSettings**
- 3.2.29 **CloudStaticIP**
- 3.2.30 **CloudStorageAction**
- 3.2.31 **CloudStorageNodeState**
- 3.2.32 **CloudType**
- 3.2.33 **CloudVirtualNetworkInterface**
- 3.2.34 **ClusterSetup**
- 3.2.35 **CMDaemonBackgroundTask**
- 3.2.36 **CMDaemonFailover**
- 3.2.37 **CMDaemonFailoverGroup**
- 3.2.38 **CMDaemonFailoverGroupStatus**
- 3.2.39 **CMDaemonFailoverPeer**
- 3.2.40 **CMDaemonFailoverStatus**
- 3.2.41 **CMDaemonStatus**
- 3.2.42 **CMService**
- 3.2.43 **CondorClientRole**
- 3.2.44 **CondorJob**
- 3.2.45 **CondorJobQueue**
- 3.2.46 **CondorJobQueueStat**
- 3.2.47 **CondorServerRole**
- 3.2.48 **ConfigSum**

- 3.2.49 ConfigurationOverlay
- 3.2.50 ConsolidatorConf
- 3.2.51 DatabaseRole
- 3.2.52 DellClustat
- 3.2.53 DellClustatGroup
- 3.2.54 DellClustatNode
- 3.2.55 DellDiskGroupInfo
- 3.2.56 DellPhysicalDiskDriveInfo
- 3.2.57 DellRAIDControllerInfo
- 3.2.58 DellSettings
- 3.2.59 DellSettingsFirmware
- 3.2.60 DellSettingsNicDevice
- 3.2.61 DellStorageInfo
- 3.2.62 DellVirtualDiskInfo
- 3.2.63 Device
- 3.2.64 DevStatus
- 3.2.65 DiskAssertion
- 3.2.66 DiskDevice
- 3.2.67 DiskInfo
- 3.2.68 DiskPartition
- 3.2.69 DiskRaid
- 3.2.70 DiskSetup
- 3.2.71 DiskVolume
- 3.2.72 DiskVolumeGroup
- 3.2.73 DrainResult
- 3.2.74 EBSattachAction
- 3.2.75 EBSdetachAction
- 3.2.76 EC2AMI
- 3.2.77 EC2AvailabilityZone
- 3.2.78 EC2EBSStorage
- 3.2.79 EC2EphemeralStorage
- 3.2.80 EC2PrivateCloud
- 3.2.81 EC2Provider
- 3.2.82 EC2Region
- 3.2.83 EC2RegionAMI
- 3.2.84 EC2Settings
- 3.2.85 EC2StaticIP
- 3.2.86 EC2Storage
- 3.2.87 EC2Type
- 3.2.88 EC2VirtualNetworkInterface
- 3.2.89 EntityManagersMD5
- 3.2.90 EthernetSwitch
- 3.2.91 FailoverRole
- 3.2.92 FakeData
- 3.2.93 FSExport
- 3.2.94 FSMount
- 3.2.95 FSPart

- 3.2.96 FSPartAssociation
- 3.2.97 FSPartBasicAssociation
- 3.2.98 FSPartProviderAssociation
- 3.2.99 GenericDevice
- 3.2.100 GenericResource
- 3.2.101 GPUInfo
- 3.2.102 GPUSettings
- 3.2.103 GpuUnit
- 3.2.104 GPUUnitInfo
- 3.2.105 GridEngineJob
- 3.2.106 GridEngineJobQueue
- 3.2.107 GridEngineJobQueueStat
- 3.2.108 GridEngineParallelEnvironment
- 3.2.109 Group
- 3.2.110 GuiCephOsdPoolInfo
- 3.2.111 GuiCephOverview
- 3.2.112 GuiCephPgslInfo
- 3.2.113 GuiClusterOverview
- 3.2.114 GuiCompleteOpenStackOverview
- 3.2.115 GuiDiskUsage
- 3.2.116 GuiGpuUnitOverview
- 3.2.117 GuiHadoopHDFSDetailHBase
- 3.2.118 GuiHadoopHDFSDetailHDFS
- 3.2.119 GuiHadoopHDFSDetailMapreduce
- 3.2.120 GuiHadoopHDFSDetailSpark
- 3.2.121 GuiHadoopHDFSDetailYarn
- 3.2.122 GuiHadoopHDFSDetailZooKeeper
- 3.2.123 GuiHadoopHDFSOverview
- 3.2.124 GuiJob
- 3.2.125 GuiNetSwitchStatus
- 3.2.126 GuiNetworkInterface
- 3.2.127 GuiNodeOverview
- 3.2.128 GuiNodeStatus
- 3.2.129 GuiOpenStackOverview
- 3.2.130 GuiOpenStackProjectOverview
- 3.2.131 GuiOpenStackTenantOverview
- 3.2.132 GuiPDUBank
- 3.2.133 GuiPDUOutlet
- 3.2.134 GuiWorkload
- 3.2.135 HadoopBaseConfiguration
- 3.2.136 HadoopDataNodeHDFSConfiguration
- 3.2.137 HadoopDataNodeRole
- 3.2.138 HadoopHBaseClientHDFSConfiguration
- 3.2.139 HadoopHBaseClientRole
- 3.2.140 HadoopHBaseServerHDFSConfiguration
- 3.2.141 HadoopHBaseServerRole
- 3.2.142 HadoopHDFS

- 3.2.143 HadoopHiveHDFSConfiguration
- 3.2.144 HadoopHiveRole
- 3.2.145 HadoopJob
- 3.2.146 HadoopJobQueue
- 3.2.147 HadoopJobQueueStat
- 3.2.148 HadoopJobTrackerHDFSConfiguration
- 3.2.149 HadoopJobTrackerRole
- 3.2.150 HadoopJournalHDFSConfiguration
- 3.2.151 HadoopJournalRole
- 3.2.152 HadoopKMServerHDFSConfiguration
- 3.2.153 HadoopKMServerRole
- 3.2.154 HadoopNameNodeHDFSConfiguration
- 3.2.155 HadoopNameNodeRole
- 3.2.156 HadoopNFSGatewayHDFSConfiguration
- 3.2.157 HadoopNFSGatewayRole
- 3.2.158 HadoopSecondaryNameNodeHDFSConfiguration
- 3.2.159 HadoopSecondaryNameNodeRole
- 3.2.160 HadoopSparkMasterHDFSConfiguration
- 3.2.161 HadoopSparkMasterRole
- 3.2.162 HadoopSparkWorkerHDFSConfiguration
- 3.2.163 HadoopSparkWorkerRole
- 3.2.164 HadoopSparkYARNHDFSConfiguration
- 3.2.165 HadoopSparkYARNRole
- 3.2.166 HadoopSqoopHDFSConfiguration
- 3.2.167 HadoopSqoopRole
- 3.2.168 HadoopTaskTrackerHDFSConfiguration
- 3.2.169 HadoopTaskTrackerRole
- 3.2.170 HadoopYARNClientHDFSConfiguration
- 3.2.171 HadoopYARNClientRole
- 3.2.172 HadoopYARNServerHDFSConfiguration
- 3.2.173 HadoopYARNServerRole
- 3.2.174 HadoopZooKeeperHDFSConfiguration
- 3.2.175 HadoopZooKeeperRole
- 3.2.176 HAProxyBackendInformation
- 3.2.177 HAProxyEntry
- 3.2.178 HAProxyEntryBind
- 3.2.179 HAProxyFrontendInformation
- 3.2.180 HAProxyNodeInformation
- 3.2.181 HAProxyRole
- 3.2.182 HAProxyServer
- 3.2.183 HAProxySharedSettings
- 3.2.184 HealthCheck
- 3.2.185 HeatMapData
- 3.2.186 IBSwitch
- 3.2.187 IniConfigFileCustomizationEntry
- 3.2.188 IPCPerm
- 3.2.189 IPResource

3.2.190 Job
3.2.191 JobQueue
3.2.192 JobQueuePlaceholder
3.2.193 JobQueueStat
3.2.194 KernelModule
3.2.195 KeyValuePair
3.2.196 LicenseInfo
3.2.197 LicenseManagerRole
3.2.198 LoginRole
3.2.199 LSFBaseJob
3.2.200 LSFBaseJobQueue
3.2.201 LSFBaseJobQueueStat
3.2.202 LSFClientRole
3.2.203 LSFJob
3.2.204 LSFJobQueue
3.2.205 LSFJobQueueStat
3.2.206 LSFServerRole
3.2.207 LustreAlert
3.2.208 LustreClientMount
3.2.209 LustreFileSystem
3.2.210 LustreFileSystemTarget
3.2.211 LustreLog
3.2.212 LustreOverview
3.2.213 LustreServer
3.2.214 LustreServerProfile
3.2.215 LustreSettings
3.2.216 LustreTargetMap
3.2.217 LustreUser
3.2.218 LustreVolume
3.2.219 LustreVolumeNode
3.2.220 MasterNode
3.2.221 MasterRole
3.2.222 MemcachedRole
3.2.223 MemoryInfo
3.2.224 Metric
3.2.225 MetricPrmId
3.2.226 MICHostRole
3.2.227 MICInfo
3.2.228 MICNode
3.2.229 MICNodeCategory
3.2.230 MICOverlay
3.2.231 MICSettings
3.2.232 MonConf
3.2.233 MonGlobalConf
3.2.234 MonHealthConf
3.2.235 MonitoringRole
3.2.236 MonMetricConf

- 3.2.237 **MsgQueue**
- 3.2.238 **MyrinetSwitch**
- 3.2.239 **Network**
- 3.2.240 **NetworkAliasInterface**
- 3.2.241 **NetworkBmcInterface**
- 3.2.242 **NetworkBondInterface**
- 3.2.243 **NetworkBridgeInterface**
- 3.2.244 **NetworkInterface**
- 3.2.245 **NetworkNetMapInterface**
- 3.2.246 **NetworkPhysicalInterface**
- 3.2.247 **NetworkTunnelInterface**
- 3.2.248 **NetworkVLANInterface**
- 3.2.249 **NewNode**
- 3.2.250 **NFSexportAction**
- 3.2.251 **NFSmountAction**
- 3.2.252 **NFSunexportAction**
- 3.2.253 **NFSunmountAction**
- 3.2.254 **Node**
- 3.2.255 **NodeCategory**
- 3.2.256 **NodeGroup**
- 3.2.257 **OpenLavaClientRole**
- 3.2.258 **OpenLavaJob**
- 3.2.259 **OpenLavaJobQueue**
- 3.2.260 **OpenLavaJobQueueStat**
- 3.2.261 **OpenLavaServerRole**
- 3.2.262 **OpenStack**
- 3.2.263 **OpenStackApiAgent**
- 3.2.264 **OpenStackApiDomain**
- 3.2.265 **OpenStackApiEndpoint**
- 3.2.266 **OpenStackApiEntity**
- 3.2.267 **OpenStackApiFlavor**
- 3.2.268 **OpenStackApiFloatingIP**
- 3.2.269 **OpenStackApiGroup**
- 3.2.270 **OpenStackApiHostAggregate**
- 3.2.271 **OpenStackApiHypervisor**
- 3.2.272 **OpenStackApiImage**
- 3.2.273 **OpenStackApiNetwork**
- 3.2.274 **OpenStackApiPort**
- 3.2.275 **OpenStackApiProject**
- 3.2.276 **OpenStackApiRole**
- 3.2.277 **OpenStackApiRoleAssignment**
- 3.2.278 **OpenStackApiRouter**
- 3.2.279 **OpenStackApiSecurityGroup**
- 3.2.280 **OpenStackApiServer**
- 3.2.281 **OpenStackApiService**
- 3.2.282 **OpenStackApiSubnet**
- 3.2.283 **OpenStackApiUser**

3.2.284 **OpenStackApiVolume**
3.2.285 **OpenStackApiVolumeSnapshot**
3.2.286 **OpenStackApiVolumeType**
3.2.287 **OpenStackBareMetalApiRole**
3.2.288 **OpenStackBareMetalConductorRole**
3.2.289 **OpenStackBareMetalDiscoverdDNsmasqRole**
3.2.290 **OpenStackBareMetalDiscoverdRole**
3.2.291 **OpenStackBlockStorage**
3.2.292 **OpenStackComputeApiEC2Role**
3.2.293 **OpenStackComputeApiMetadataRole**
3.2.294 **OpenStackComputeApiRole**
3.2.295 **OpenStackComputeConductorRole**
3.2.296 **OpenStackComputeRole**
3.2.297 **OpenStackComputeSchedulerRole**
3.2.298 **OpenStackComputeVNCProxyRole**
3.2.299 **OpenStackConfigFileCustomization**
3.2.300 **OpenStackDashboardRole**
3.2.301 **OpenStackDataProcessingApiRole**
3.2.302 **OpenStackDBaaSRole**
3.2.303 **OpenStackDefaultUserRole**
3.2.304 **OpenStackIdentityApiRole**
3.2.305 **OpenStackImageApiRole**
3.2.306 **OpenStackImageBackend**
3.2.307 **OpenStackImageBackendCeph**
3.2.308 **OpenStackImageBackendFS**
3.2.309 **OpenStackImageRegistryRole**
3.2.310 **OpenStackMessageQueueServerRole**
3.2.311 **OpenStackNetworkApiRole**
3.2.312 **OpenStackNetworkRole**
3.2.313 **OpenStackNodeRole**
3.2.314 **OpenStackNovalImageBackend**
3.2.315 **OpenStackNovalImageBackendCeph**
3.2.316 **OpenStackNovalImageBackendCow**
3.2.317 **OpenStackObjectAccountRole**
3.2.318 **OpenStackObjectApiRole**
3.2.319 **OpenStackObjectContainerRole**
3.2.320 **OpenStackObjectStoreRole**
3.2.321 **OpenStackOrchestrationApiRole**
3.2.322 **OpenStackOrchestrationRole**
3.2.323 **OpenStackSettings**
3.2.324 **OpenStackSettingsAdvanced**
3.2.325 **OpenStackSettingsCMDaemonInteractions**
3.2.326 **OpenStackSettingsCompute**
3.2.327 **OpenStackSettingsCredentials**
3.2.328 **OpenStackSettingsDatabase**
3.2.329 **OpenStackSettingsLogging**
3.2.330 **OpenStackSettingsNetworking**

- 3.2.331 **OpenStackSettingsPorts**
- 3.2.332 **OpenStackSettingsQuota**
- 3.2.333 **OpenStackSettingsUserPortal**
- 3.2.334 **OpenStackSettingsUsers**
- 3.2.335 **OpenStackStorage**
- 3.2.336 **OpenStackTelemetryAgentCentralRole**
- 3.2.337 **OpenStackTelemetryAgentComputeRole**
- 3.2.338 **OpenStackTelemetryAgentIpmiRole**
- 3.2.339 **OpenStackTelemetryAgentNotificationRole**
- 3.2.340 **OpenStackTelemetryAlarmEvaluatorRole**
- 3.2.341 **OpenStackTelemetryAlarmNotifierRole**
- 3.2.342 **OpenStackTelemetryApiRole**
- 3.2.343 **OpenStackTelemetryCollectorRole**
- 3.2.344 **OpenStackUserRole**
- 3.2.345 **OpenStackUserSettings**
- 3.2.346 **OpenStackVolumeApiRole**
- 3.2.347 **OpenStackVolumeBackend**
- 3.2.348 **OpenStackVolumeBackendCeph**
- 3.2.349 **OpenStackVolumeBackendNFS**
- 3.2.350 **OpenStackVolumeBackupBackend**
- 3.2.351 **OpenStackVolumeBackupBackendCeph**
- 3.2.352 **OpenStackVolumeBackupRole**
- 3.2.353 **OpenStackVolumeRole**
- 3.2.354 **OpenStackVolumeSchedulerRole**
- 3.2.355 **OsapiPortIP**
- 3.2.356 **OsapiSecurityGroupRule**
- 3.2.357 **OsapiSubnetAllocationPool**
- 3.2.358 **OSService**
- 3.2.359 **OSServiceArray**
- 3.2.360 **OSServiceConfig**
- 3.2.361 **Partition**
- 3.2.362 **PBSJob**
- 3.2.363 **PBSJobQueue**
- 3.2.364 **PBSJobQueueStat**
- 3.2.365 **PbsProClientRole**
- 3.2.366 **PbsProJob**
- 3.2.367 **PbsProJobQueue**
- 3.2.368 **PbsProJobQueueStat**
- 3.2.369 **PbsProServerRole**
- 3.2.370 **PDUPort**
- 3.2.371 **PhysicalNode**
- 3.2.372 **PowerDistributionUnit**
- 3.2.373 **PowerStatus**
- 3.2.374 **Process**
- 3.2.375 **Processor**
- 3.2.376 **Profile**
- 3.2.377 **ProgramRunnerInput**

3.2.378 ProgramRunnerKill
3.2.379 ProgramRunnerOutput
3.2.380 ProgramRunnerStatus
3.2.381 ProvisioningNodeStatus
3.2.382 ProvisioningProcessorJob
3.2.383 ProvisioningRequestStatus
3.2.384 ProvisioningRole
3.2.385 ProvisioningStatus
3.2.386 Puppet
3.2.387 PuppetApplyResult
3.2.388 PuppetApplySession
3.2.389 PuppetClass
3.2.390 PuppetRole
3.2.391 PuppetRunInfo
3.2.392 Rack
3.2.393 RackSensor
3.2.394 RadosGatewayRole
3.2.395 RateElem
3.2.396 ReadMonDataId
3.2.397 ReadMonDataOutput
3.2.398 RemoteMonConf
3.2.399 RemoteMonMetricConf
3.2.400 RemoteNodeInstallerInteraction
3.2.401 RemoteSetupExecution
3.2.402 RemoteThreshold
3.2.403 ResourcePool
3.2.404 ResourcePoolStatus
3.2.405 Role
3.2.406 RunJobAction
3.2.407 S3DataDownload
3.2.408 S3DataUpload
3.2.409 S3ResultsDownload
3.2.410 S3ResultsUpload
3.2.411 S3Transfer
3.2.412 Semaphore
3.2.413 Sensor
3.2.414 Session
3.2.415 SGEClientRole
3.2.416 SGEJob
3.2.417 SGEJobQueue
3.2.418 SGEJobQueueStat
3.2.419 SGEParallelEnvironment
3.2.420 SGEServerRole
3.2.421 SharedMemory
3.2.422 SlaveMonotonicElem
3.2.423 SlaveMonSnapshot
3.2.424 SlaveNode

3.2.425 SlaveRateElem
3.2.426 SlurmClientRole
3.2.427 SlurmJob
3.2.428 SlurmJobQueue
3.2.429 SlurmJobQueueStat
3.2.430 SlurmServerRole
3.2.431 SoftwareImage
3.2.432 SoftwareImageProxy
3.2.433 StartStorageNodeAction
3.2.434 StateElem
3.2.435 StaticRoute
3.2.436 StatisticData
3.2.437 StopStorageNodeAction
3.2.438 StorageNodePolicy
3.2.439 StorageRole
3.2.440 StringListObject
3.2.441 SubnetManagerRole
3.2.442 Switch
3.2.443 SwitchPort
3.2.444 SysInfoCollector
3.2.445 SysMonotonicWithId
3.2.446 SysRateWithId
3.2.447 ThreshAction
3.2.448 ThreshActionConf
3.2.449 Threshold
3.2.450 Ticket
3.2.451 TorqueClientRole
3.2.452 TorqueJob
3.2.453 TorqueJobQueue
3.2.454 TorqueJobQueueStat
3.2.455 TorqueServerRole
3.2.456 UCSAdaptorEthCompQueueProfile
3.2.457 UCSAdaptorEthGenProfile
3.2.458 UCSAdaptorEthInterruptProfile
3.2.459 UCSAdaptorEthOffloadProfile
3.2.460 UCSAdaptorEthRecvQueueProfile
3.2.461 UCSAdaptorEthUSNICProfile
3.2.462 UCSAdaptorEthWorkQueueProfile
3.2.463 UCSAdaptorExtEthIf
3.2.464 UCSAdaptorExtIpv6RssHashProfile
3.2.465 UCSAdaptorFcCdbWorkQueueProfile
3.2.466 UCSAdaptorFcErrorRecoveryProfile
3.2.467 UCSAdaptorFcGenProfile
3.2.468 UCSAdaptorFcInterruptProfile
3.2.469 UCSAdaptorFcPortFLogiProfile
3.2.470 UCSAdaptorFcPortPLogiProfile
3.2.471 UCSAdaptorFcPortProfile

3.2.472 UCSAdaptorFcRecvQueueProfile
3.2.473 UCSAdaptorFcWorkQueueProfile
3.2.474 UCSAdaptorHostEthIf
3.2.475 UCSAdaptorHostFclIf
3.2.476 UCSAdaptorIpV4RssHashProfile
3.2.477 UCSAdaptorIpV6RssHashProfile
3.2.478 UCSAdaptorPortProfiles
3.2.479 UCSAdaptorRssProfile
3.2.480 UCSBase
3.2.481 UCSBiosBootDev
3.2.482 UCSBiosBootDevGrp
3.2.483 UCSBiosSettings
3.2.484 UCSBiosVfAdjacentCacheLinePrefetch
3.2.485 UCSBiosVfAltitude
3.2.486 UCSBiosVfASPMSupport
3.2.487 UCSBiosVfConsoleRedirection
3.2.488 UCSBiosVfCoreMultiProcessing
3.2.489 UCSBiosVfCPUEnergyPerformance
3.2.490 UCSBiosVfCPUFrequencyFloor
3.2.491 UCSBiosVfCPUPerformance
3.2.492 UCSBiosVfCPUPowerManagement
3.2.493 UCSBiosVfDCUPrefetch
3.2.494 UCSBiosVfDemandScrub
3.2.495 UCSBiosVfDirectCacheAccess
3.2.496 UCSBiosVfDRAMClockThrottling
3.2.497 UCSBiosVfDramRefreshRate
3.2.498 UCSBiosVfEnhancedIntelSpeedStepTech
3.2.499 UCSBiosVfExecuteDisableBit
3.2.500 UCSBiosVfFRB2Enable
3.2.501 UCSBiosVfHardwarePrefetch
3.2.502 UCSBiosVfIntelHyperThreadingTech
3.2.503 UCSBiosVfIntelTurboBoostTech
3.2.504 UCSBiosVfIntelVirtualizationTechnology
3.2.505 UCSBiosVfIntelVTFForDirectedIO
3.2.506 UCSBiosVfLegacyUSBSupport
3.2.507 UCSBiosVfLOMPortOptionROM
3.2.508 UCSBiosVfLvDIMMSupport
3.2.509 UCSBiosVfMemoryInterleave
3.2.510 UCSBiosVfMemoryMappedIOAbove4GB
3.2.511 UCSBiosVfNUMAOptimized
3.2.512 UCSBiosVfOnboardStorage
3.2.513 UCSBiosVfOnboardStorageSWStack
3.2.514 UCSBiosVfOSBootWatchdogTimer
3.2.515 UCSBiosVfOSBootWatchdogTimerPolicy
3.2.516 UCSBiosVfOSBootWatchdogTimerTimeout
3.2.517 UCSBiosVfPatrolScrub
3.2.518 UCSBiosVfPCIOptionROMs

3.2.519 UCSBiosVfPCISlotOptionROMEnable
3.2.520 UCSBiosVfProcessorC1E
3.2.521 UCSBiosVfProcessorC6Report
3.2.522 UCSBiosVfPStateCoordType
3.2.523 UCSBiosVfQPIConfig
3.2.524 UCSBiosVfSelectMemoryRASConfiguration
3.2.525 UCSBiosVfTPMSupport
3.2.526 UCSBiosVfUCSMBootOrderRuleControl
3.2.527 UCSBiosVfUSBEmulation
3.2.528 UCSBiosVfUSBPortsConfig
3.2.529 UCSBiosVfVgaPriority
3.2.530 UCSCommNtpProvider
3.2.531 UCSCommSyslog
3.2.532 UCSCommSyslogClient
3.2.533 UCSEquipmentIndicatorLed
3.2.534 UCSEquipmentLocatorLed
3.2.535 UCSFaultInst
3.2.536 UCSFirmwareRunning
3.2.537 UCSInfo
3.2.538 UCSLogs
3.2.539 UCSLsbootDef
3.2.540 UCSLsbootEfi
3.2.541 UCSLsbootLan
3.2.542 UCSLsbootStorage
3.2.543 UCSLsbootVirtualMedia
3.2.544 UCSStatus
3.2.545 UGECgroupsSettings
3.2.546 UGEClientRole
3.2.547 UGEJob
3.2.548 UGEJobQueue
3.2.549 UGEJobQueueStat
3.2.550 UGEParallelEnvironment
3.2.551 UGEServerRole
3.2.552 User
3.2.553 Validation
3.2.554 VersionInfo
3.2.555 VirtualNode
3.2.556 VirtualNodeSettings
3.2.557 VirtualSMPNode
3.2.558 VScaleMPSettings
3.2.559 VsmpSettings
3.2.560 WillChange
3.2.561 XeonPhiSettings

3.3 JSON Examples

complete.sh

```
#!/bin/bash

URL=https://localhost:2081/json/
user=koen
pass=koen

echo "==== login ====="
curl -c curl.cookie.txt -i -k -X POST -d '{"service":"login", \
"username":"koen", "password":"' $pass'"}' $URL; echo
echo "==== master ====="
curl --cookie curl.cookie.txt -i -k -X POST -d '{"service":"cm\
device", "call":"getNode", "arg":"master"}' $URL; echo
echo "==== logout ====="
curl --cookie curl.cookie.txt -i -k -X POST -d '{"service":"lo\
gout"}' $URL; echo
echo "==== denied ====="
curl --cookie curl.cookie.txt -i -k -X POST -d '{"service":"cm\
device", "call":"getNode", "arg":"master"}' $URL; echo
rm -f curl.cookie.txt

echo "==== cert ====="
curl --cert $HOME/.cm/admin.pem --key $HOME/.cm/admin.key -i -k \
-X POST -d '{"service":"cmdevice", "call":"getNode", "arg":"master\
"}' $URL; echo
```

curl.sh

```
#!/bin/bash

source url
if [ -z "$1" ]; then
    pass=koen
else
    pass=$1
fi
read -p "pass: " -s -a $pass

curl -c curl.cookie.txt -i -k -X POST -d '{"service":"login", \
"username":"koen", "password":"' $pass'"}' $URL

# curl --cookie curl.cookie.txt -i -k -X POST -d '{"service":"\
cmsession", "call":"getLastEvents", "args":[0,256]}' $URL

curl --cookie curl.cookie.txt -i -k -X POST -d '{"service":"cm\
main", "call":"getProfile"}' $URL
curl --cookie curl.cookie.txt -i -k -X POST -d '{"service":"cm\
main", "call":"getSubjectName"}' $URL
```

devices.sh

```
#!/bin/bash
source url

if [ "$1" == "gzip" ]; then
```

```

    wget --load-cookies cookie.txt --header='Accept-Encoding: gzip\
' --no-check-certificate --server-response -qO- $URL --post-dat\
a='{"service":"cmdevice","call":"getDevices"}'
else
    wget --load-cookies cookie.txt --no-check-certificate --server\
-response -qO- $URL --post-data='{"service":"cmdevice","call":"g\
etDevices"}'
fi

```

loadone.sh

```

#!/bin/bash
source url

# not perfect but gets the job done
function jsonval {
    temp=`echo $json | sed 's/\\\\\\\\/\\/g' | sed 's/[{}]/g' | awk\
-v k="text" '{n=split($0,a,","); for (i=1; i<=n; i++) print a[i\
]}' | sed 's/\"\\:\\"/\\/g' | sed 's/[\\,]/ /g' | sed 's/\"/\"/g' | g\
rep -w $prop`
    r=$(echo ${temp##*|} | tr ']' ' ' | tr ' ' '\n' | cut -d: -f2 \
| sort -n)
    echo $(echo $r | cut -d' ' -f 1)
}

prop='uniqueKey'

node=master
json=`wget --load-cookies cookie.txt --no-check-certificate --se\
rver-response -qO- $URL --post-data='{"service":"cmdevice","call\
":"getDevice","arg1":"' $node' }'`
nkey=$(jsonval)
if [ -z $nkey ]; then
    echo $json
    exit 1
fi
echo "$node.uniqueKey = $nkey"

json=`wget --load-cookies cookie.txt --no-check-certificate --se\
rver-response -qO- $URL --post-data='{"service":"cmmon","call": "\
getMetric","arg1":"loadOne"}'`
mkey=$(jsonval)
echo "loadone.uniqueKey = $mkey"

now=$(date +%s)
day=$((now-86400))

# echo -----
# wget --load-cookies cookie.txt --no-check-certificate --server\
-response -qO- $URL \
# --post-data='{"service":"cmmon","call":"readDataByIntervalNu\
m",
# "readMonDataIdArray":[{"devId":"' $nkey', "metric\
Id":"' $mkey',

```

```

#                                     "begTime": '$day', "endTi\
me": '$now' }},
#                                     "intervalNum": 0}'
#
# echo
echo -----
wget --load-cookies cookie.txt --no-check-certificate --server-r\
esponse -qO- $URL \
  --post-data='{"service": "cmmon", "call": "readDataByIntervalNum",
               "args": [{"baseType": "ReadMonDataId", "uniqueKey"\
: 0, "modified": false, "toBeRemoved": false, "childType": "",
                        "devId": '$nkey', "metricId": '$mkey',
                        "begTime": '$day', "endTime": '$now' }], 0}'

# echo
# echo -----
# data='{"service": "cmmon", "call": "readDataByIntervalNum",
#       "args": [{"baseType": "ReadMonDataId", "uniqueKe\
y": 0, "modified": false, "toBeRemoved": false, "childType": "",
                "devId": '$nkey', "metricId": '$mkey',
                "begTime": '$day', "endTime": '$now' }], \
0}'
# rm loadone.txt.gz
# echo $data > loadone.txt
# gzip -n loadone.txt
# len=$(wc -c loadone.txt.gz | cut -d" " -f1)
# wget --load-cookies cookie.txt --no-check-certificate --header\
"Content-Length: $len" --header 'Content-Encoding: gzip' --serv\
er-response -O- $URL \
#   --post-file=loadone.txt.gz

```

login.sh

```

#!/bin/bash
source url
user=$USER
pass=$user
wget --keep-session-cookies --save-cookies cookie.txt --no-check\
-certificate --server-response -qO- $URL \
  --post-data='{"service": "login", "username": "'$user'", "passwor\
d": "'$pass'"}'
echo

```

logout.sh

```

#!/bin/bash
source url
wget --load-cookies cookie.txt --no-check-certificate --server-r\
esponse -qO- $URL --post-data='{"service": "logout"}'
rm cookie.txt
echo

```

node001.sh

```

#!/bin/bash
source url

```

```
if [ -z "$1" ]; then
    node=node001
else
    node=$1
fi

wget --load-cookies cookie.txt --no-check-certificate --server-r\
esponse -qO- $URL --post-data='{ "service": "cmdevice", "call": "get\
Device", "arg1": "' $node' "}'
```