

Oracle® Solaris Cluster 参考手册

ORACLE®

文件号码 E51743
2014 年 7 月, E51743-01

版权所有 © 2000, 2014, Oracle 和/或其附属公司。保留所有权利。

本软件和相关文档是根据许可证协议提供的，该许可证协议中规定了关于使用和公开本软件和相关文档的各种限制，并受知识产权法的保护。除非在许可证协议中明确许可或适用法律明确授权，否则不得以任何形式、任何方式使用、拷贝、复制、翻译、广播、修改、授权、传播、分发、展示、执行、发布或显示本软件和相关文档的任何部分。除非法律要求实现互操作，否则严禁对本软件进行逆向工程设计、反汇编或反编译。

此文档所含信息可能随时被修改，恕不另行通知，我们不保证该信息没有错误。如果贵方发现任何问题，请书面通知我们。

如果将本软件或相关文档交付给美国政府，或者交付给以美国政府名义获得许可证的任何机构，必须符合以下规定：

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

本软件或硬件是为了在各种信息管理应用领域内的一般使用而开发的。它不应被应用于任何存在危险或潜在危险的应用领域，也不是为此而开发的，其中包括可能会产生人身伤害的应用领域。如果在危险应用领域内使用本软件或硬件，贵方应负责采取所有适当的防范措施，包括备份、冗余和其它确保安全使用本软件或硬件的措施。对于因在危险应用领域内使用本软件或硬件所造成的一切损失或损害，Oracle Corporation 及其附属公司概不负责。

Oracle 和 Java 是 Oracle 和/或其附属公司的注册商标。其他名称可能是各自所有者的商标。

Intel 和 Intel Xeon 是 Intel Corporation 的商标或注册商标。所有 SPARC 商标均是 SPARC International, Inc 的商标或注册商标，并应按照许可证的规定使用。AMD、Opteron、AMD 徽标以及 AMD Opteron 徽标是 Advanced Micro Devices 的商标或注册商标。UNIX 是 The Open Group 的注册商标。

本软件或硬件以及文档可能提供了访问第三方内容、产品和服务的方式或有关这些内容、产品和服务的信息。对于第三方内容、产品和服务，Oracle Corporation 及其附属公司明确表示不承担任何种类的担保，亦不对其承担任何责任。对于因访问或使用第三方内容、产品或服务所造成的任何损失、成本或损害，Oracle Corporation 及其附属公司概不负责。

目录

前言	11
I 简介	15
Intro	17
II OSC4 1	23
libscho.st.so.1	25
III OSC4 1cl	29
claccess	31
cldev	37
cldevice	51
cldevicegroup	65
cldg	81
clinterconnect	97
clintr	107
clmib	117
clnas	125
clnasdevice	137
clnode	149
clps	167
clpstring	175
clq	183
clquorum	195
clreslogicalhostname	207
clresource	225
clresourcegroup	253
clresourcetype	277
clressharedaddress	289
clrg	307
clrs	331
clrslh	359
clrssa	377
clrt	395

clsetup	407
clsnmphost	409
clsnmpmib	417
clsnmpuser	425
clta	433
cltelemetryattribute	447
cluster	461
clzc	491
clzonecluster	517
IV OSC4 1ha	543
rt_callbacks	545
scdsbuilder	553
scdsconfig	555
scdscreate	559
scha_check_app_user	563
scha_cluster_get	567
scha_cmds	573
scha_control	579
scha_resource_get	585
scha_resource_setstatus	593
scha_resourcegroup_get	597
scha_resourcetype_get	601
V OSC4 1m	605
ccradm	607
cl_eventd	613
cl_pnmd	615
dcs_config	617
halockrun	621
hatimerun	623
pmfadm	625
pmfd	631
rpc.pmfad	633
sc_zonesd	635
scconf	637
scconf_dg_rawdisk	655
scconf_dg_svm	659
scconf_quorum_dev_quorum_server	661
scconf_quorum_dev_scsi	665
scconf_transp_adap_bge	669
scconf_transp_adap_e1000g	671

sconf_transp_jct_etherswitch	673
sconf_transp_jct_ibswitch	675
scdidadm	677
scdpm	685
sceventmib	689
scgdevs	695
scinstall	697
scnas	717
scnasdir	721
scprivipadm	725
scprivipd	729
scrgadm	731
scsetup	741
scshutdown	743
scstat	745
scswitch	751
sctelemetry	763
scversions	767
VI OSC4 3ha	769
scds_calls	771
scds_close	777
scds_error_string	779
scds_error_string_i18n	781
scds_failover_rg	783
scds_fm_action	785
scds_fm_net_connect	789
scds_fm_net_disconnect	793
scds_fm_print_probes	795
scds_fm_sleep	797
scds_fm_tcp_connect	799
scds_fm_tcp_disconnect	801
scds_fm_tcp_read	803
scds_fm_tcp_write	805
scds_free_ext_property	807
scds_free_net_list	809
scds_free_netaddr_list	811
scds_free_port_list	813
scds_get_current_method_name	815
scds_get_ext_property	817
scds_get_fullname	821

scds_get_fullname_nodeid	823
scds_get_netaddr_list	825
scds_get_port_list	827
scds_get_resource_group_name	829
scds_get_resource_name	831
scds_get_resource_type_name	833
scds_get_rg_hostnames	835
scds_get_rg_hostnames_zone	837
scds_get_rs_hostnames	839
scds_get_zone_name	841
scds_hasp_check	843
scds_initialize	847
scds_is_zone_cluster	849
scds_pmf_get_status	851
scds_pmf_restart_fm	853
scds_pmf_signal	855
scds_pmf_start	857
scds_pmf_start_env	859
scds_pmf_stop	861
scds_pmf_stop_monitoring	863
scds_print_net_list	865
scds_print_netaddr_list	867
scds_print_port_list	869
scds_property_functions	871
scds_restart_resource	879
scds_restart_rg	881
scds_simple_net_probe	883
scds_simple_probe	885
scds_svc_wait	887
scds_syslog	891
scds_syslog_debug	893
scds_timerun	895
scha_calls	897
scha_cluster_close	903
scha_cluster_get	909
scha_cluster_get_zone	915
scha_cluster_getlogfacility	921
scha_cluster_getnodename	923
scha_cluster_getzone	925
scha_cluster_open	927

	scha_cluster_open_zone	933
	scha_control	939
	scha_control_zone	945
	scha_get_fullname	951
	scha_resource_close	953
	scha_resource_get	963
	scha_resource_get_zone	973
	scha_resource_open	983
	scha_resource_open_zone	993
	scha_resource_setstatus	1003
	scha_resource_setstatus_zone	1005
	scha_resourcegroup_close	1007
	scha_resourcegroup_get	1013
	scha_resourcegroup_get_zone	1019
	scha_resourcegroup_open	1025
	scha_resourcegroup_open_zone	1031
	scha_resourcetype_close	1037
	scha_resourcetype_get	1041
	scha_resourcetype_get_zone	1045
	scha_resourcetype_open	1049
	scha_resourcetype_open_zone	1053
	scha_strerror	1057
	scha_strerror_i18n	1059
VII	OSC4 4	1061
	clusters	1063
	commandlog	1065
	rt_reg	1067
	scdpmd.conf	1075
	serialports	1077
VIII	OSC4 5	1079
	crs_framework	1081
	derby	1085
	property_attributes	1087
	Proxy_SMF_failover	1091
	Proxy_SMF_multimaster	1095
	Proxy_SMF_scalable	1099
	r_properties	1103
	rac_framework	1127
	rg_properties	1129
	rt_properties	1143

scalable_service	1153
ScalDeviceGroup	1155
ScalMountPoint	1161
SCTelemetry	1169
SUNW.crs_framework	1171
SUNW.derby	1175
SUNW.Event	1177
SUNW.gds	1183
SUNW.HAStoragePlus	1193
SUNW.Proxy_SMF_failover	1201
SUNW.Proxy_SMF_multimaster	1205
SUNW.Proxy_SMF_scalable	1209
SUNW.rac_framework	1213
SUNWct.ScalDeviceGroup	1215
SUNW.ScalMountPoint	1221
SUNW.SCTelemetry	1229
SUNW.vucmm_framework	1231
SUNW.vucmm_svm	1233
vucmm_framework	1237
vucmm_svm	1239
IX OSC4 5cl	1243
clconfiguration	1245
X OSC4 7	1271
clprivnet	1273
did	1275
XI OSC4 7p	1277
sctransp_dlpi	1279
索引	1281

前言

《Oracle Solaris Cluster 参考手册》提供了有关 Oracle Solaris Cluster 软件中的命令、函数和其他公共接口的参考信息。本书面向具有丰富的 Oracle 软硬件知识的有经验的系统管理员。请不要将本书用作规划指南或售前指南。本书中的信息假定读者具备 Oracle Solaris 操作系统知识以及与 Oracle Solaris Cluster 软件一起使用的卷管理器软件的专业知识。

初学者和那些熟悉 Oracle Solaris 操作系统的用户均可以使用联机手册页获取有关基于 SPARC 的系统或基于 x86 的系统及其功能的信息。

手册页可用来简明地回答“该命令的功能是什么？”问题。通常，手册页包含参考手册。它们并不作为教程使用。

注 - Oracle Solaris Cluster 软件在以下两种平台上运行：SPARC 和 x86。除非在特定章节、注释、列表项、图形、表格或示例中另有说明，否则本书中的信息对于两种平台均适用。

概述

以下内容包含对各手册页部分及其所引用的信息的简要说明：

- 第 1 部分按字母顺序介绍了适用于操作系统的各种命令。
- 第 1CL 部分按字母顺序介绍了用于维护和管理 Oracle Solaris Cluster 的各种命令。
- 第 1HA 部分按字母顺序介绍了各种 Oracle Solaris Cluster 高可用性 (high availability, HA) 命令。
- 第 1M 部分按字母顺序介绍了主要用于系统维护和管理各种命令。
- 第 3HA 部分按字母顺序介绍了 Oracle Solaris Cluster HA 和数据服务功能。
- 第 4 部分概述了各种文件的格式。还提供了文件格式的 C 结构声明（如果适用）。
- 第 5 部分包含其他 Oracle Solaris Cluster 文档，例如资源类型的描述。
- 第 5CL 部分介绍了 Oracle Solaris Cluster 标准、环境和宏。
- 第 7 部分介绍了 Oracle Solaris Cluster 设备和网络接口。
- 第 7P 部分介绍了 Oracle Solaris Cluster 协议。

以下是手册页的通用格式。每个手册的手册页部分通常遵循该顺序，但只包括需要的标题。例如，如果未报告任何已知问题，则不包括“已知问题”部分。有关更多信息以及每

个部分的详细信息，请参见 [intro](#) 页面；有关手册页的一般信息，请参见 [Unresolved link to "man1"](#)。

名称	本部分提供了记录的命令或函数的名称，后跟它们功能的简要说明。								
用法概要	<p>本部分显示了命令或函数的语法。如果命令或文件不存在于标准路径中，则显示它的完整路径名称。除非要求使用不同的参数顺序，否则选项和参数均按字母顺序排列，首先是单个字母的参数，接下来是带有参数的选项。</p> <p>本部分使用以下特定字符：</p> <table><tr><td>[]</td><td>方括号。以这些括号括起来的选项或参数是可选的。如果省略括号，则必须指定参数。</td></tr><tr><td>...</td><td>省略号。可以为前一个参数提供多个值，或者可以多次指定前一个参数，例如“ filename ...”。</td></tr><tr><td> </td><td>分隔符。一次只能指定一个由该字符分隔的参数。</td></tr><tr><td>{ }</td><td>花括号。以花括号括起来的选项和/或参数是相互依赖的。花括号内的所有字符必须视为一个整体。</td></tr></table>	[]	方括号。以这些括号括起来的选项或参数是可选的。如果省略括号，则必须指定参数。	...	省略号。可以为前一个参数提供多个值，或者可以多次指定前一个参数，例如“ filename ...”。		分隔符。一次只能指定一个由该字符分隔的参数。	{ }	花括号。以花括号括起来的选项和/或参数是相互依赖的。花括号内的所有字符必须视为一个整体。
[]	方括号。以这些括号括起来的选项或参数是可选的。如果省略括号，则必须指定参数。								
...	省略号。可以为前一个参数提供多个值，或者可以多次指定前一个参数，例如“ filename ...”。								
	分隔符。一次只能指定一个由该字符分隔的参数。								
{ }	花括号。以花括号括起来的选项和/或参数是相互依赖的。花括号内的所有字符必须视为一个整体。								
协议	本部分仅在第 3R 子部分出现，用于指示协议描述文件。								
描述	本部分定义了服务的功能和行为。因此它简明地介绍了命令执行哪些操作。“描述”不讨论“选项”或引用“示例”。在“用法”下介绍了交互式命令、子命令、请求、宏和函数。								
IOCTL	本部分仅在第 7 部分中的页面上显示。只有为 Unresolved link to "ioctl2" 系统调用提供相应参数的设备类才称为 <code>ioctl</code> 并生成其自己的标题。特定设备的 <code>ioctl</code> 调用按字母顺序列出（位于该特定设备的手册页上）。 <code>ioctl</code> 调用可用于特定设备类。所有这些调用均以 <code>io</code> 结尾，例如 Unresolved link to "mtio71" 。								
选项	本部分列出了各个命令选项，并且包括每个选项所具有的功能的简明摘要。逐个列出各个选项，并以它们在“用法概要”部分显示的顺序排列。在选项下讨论各个选项可能的参数，还提供默认值（如果适用）。								
操作数	本部分列出了命令操作数，并介绍它们对命令操作的影响。								
输出	本部分介绍了命令所生成的输出（标准输出、标准错误或输出文件）。								

返回值	如果手册页记录返回值的函数，则本部分列出这些值并介绍返回这些值应满足的条件。如果函数只能返回常量值（例如 0 或 -1），则将在标记的段落中列出这些值。否则，会有单个段落介绍每个函数的返回值。声明为 void 的函数不返回值，因此不会在“返回值”中讨论这些函数。
错误	对于故障，大多数函数将指出它们出现故障的原因的错误代码置于全局变量 <code>errno</code> 中。本部分按字母顺序列出了函数可以生成的所有错误代码，并介绍了导致每个错误的条件。如果多个条件可以导致同一错误，则在错误代码下以单独的段落介绍每个条件。
用法	本部分列出了需要详细说明了特定规则、功能和命令。此处列出的子部分用于说明内置功能： 命令 修饰符 变量 表达式 输入语法
示例	本部分提供了用法的示例，或者如何使用命令或函数的示例。尽可能显示包括命令行条目和计算机响应的完整示例。只要给出了示例，就会显示 <code>example%</code> 提示，如果用户必须为 <code>root</code> 角色，则提示显示为 <code>example#</code> 。示例后面跟有说明、变量替换规则或返回值。大部分示例说明了“用法概要”、“描述”、“选项”和“用法”部分的概念。
环境变量	本部分列出了命令或函数影响的所有环境变量，其后附加了关于影响的简要说明。
退出状态	本部分列出了命令返回到调用程序或 shell 中的值以及导致返回这些值的条件。通常，成功完成会返回值零，而各种错误条件会返回除零之外的其他值。
文件	本部分列出了手册页引用的所有文件名称、相关文件以及命令创建或所需的文件。每个文件名称后面都具有描述性摘要或说明。
属性	本部分通过定义属性类型及其相应的值列出了命令、实用程序和设备驱动程序的特征。有关更多信息，请参见 Unresolved link to "attributes5" 。
另请参见	本部分列出了对其他手册页、内部文档和外部出版物的引用。
诊断	本部分列出了诊断消息，其中包含导致错误的条件的简要说明。
警告	本部分列出了有关特定条件的警告，这些条件可能会严重影响您的工作条件。“警告”不是诊断列表。

附注	本部分列出了不属于页面任何部分的其他信息。“附注”包括用户特别关注的要点。此处不包含关键信息。
已知问题	本部分介绍了已知问题，并尽可能给出解决方法。

简介

名称

Intro, intro — Oracle Solaris Cluster 维护命令的简介

此部分描述 Oracle Solaris Cluster 的面向对象的命令集。虽然原始的 Oracle Solaris Cluster 命令集仍可用，但是使用面向对象的命令可以更直观地进行群集配置。此外，未来的新功能可能在原始命令集中不可用。

面向对象的命令集使用通用前缀 `cl`。原始命令集使用前缀 `sc`。`sc` 和 `cl` 命令都位于 `/usr/cluster/bin` 中。

此命令集中的许多命令都有长短两种格式。例如，[clresource\(1CL\) \[225\]](#) 和 [clrs\(1CL\) \[331\]](#) 是相同的。

每个面向对象的命令设计用于管理单个类型的群集对象。命令名称指示它管理的对象类型。例如，`clresource` 命令管理 Oracle Solaris Cluster 数据服务资源。在一个命令中，子命令定义允许对特定群集对象执行的操作。

面向对象的命令集中命令的一般格式如下所示：

```
cmdname [subcommand] [option...] [operand ...]
```

和面向对象的命令一起使用的选项也有长短两种格式。可以通过使用单个破折号 (-) 后跟单个字符来指定选项的简短格式。可以通过使用两个破折号 (--) 后跟一个选项词来指定选项的长格式。例如，`-p` 是属性选项的简短格式。`--property` 为长格式。

某些选项接受选项参数，而其他选项不接受。如果选项接受选项参数，则需要该选项参数。`-?` 选项不需要任何参数。但是，`--property` 选项需要一个选项参数，该选项参数标识对其执行操作的属性。

可以对单个破折号 (-) 后面不带参数的选项的简短格式进行分组。例如，`-eM`。必须通过逗号、制表符或空格字符分隔选项后面紧跟选项参数的组。使用制表符或空格时，将选项参数用引号引起 (`-o xxx,z.yy` 或 `-o "xxx z yy"`)。

要指定具有长选项名称的选项参数，请使用 `--input=configurationfile` 格式或 `--input configurationfile` 格式。

此命令集中的所有命令均接受 `-?` 或 `--help` 选项。如果仅提供这些选项而不提供子命令，将会显示命令的摘要帮助。如果提供子命令，将只显示该子命令的帮助。

某些命令与配置文件结合使用。有关此文件所需格式的信息，请参见 [clconfiguration\(5CL\) \[1245\]](#) 手册页。

此命令集中的许多子命令都接受 `+` 作为操作数来表示所有适用对象。

命令列表

此部分以字母顺序描述可用于 Oracle Solaris Cluster 产品的面向对象的命令。

[claccess\(1CL\) \[31\]](#)

管理 Oracle Solaris Cluster 访问策略以添加节点

[cldevice\(1CL\) \[51\]](#)、[cldev\(1CL\) \[37\]](#)

管理 Oracle Solaris Cluster 设备

[cldevicegroup\(1CL\) \[65\]](#)、[cldg\(1CL\) \[81\]](#)

管理 Oracle Solaris Cluster 设备组

[clinterconnect\(1CL\) \[97\]](#)、[clintr\(1CL\) \[107\]](#)

管理 Oracle Solaris Cluster 互连

[clnasdevice\(1CL\) \[137\]](#)、[clnas\(1CL\) \[125\]](#)

为 Oracle Solaris Cluster 管理对 NAS 设备的访问权限

[clnode\(1CL\) \[149\]](#)

管理 Oracle Solaris Cluster 节点

[clpstring\(1CL\) \[175\]](#)

管理 Oracle Solaris Cluster 专用字符串

[clquorum\(1CL\) \[195\]](#)、[clq\(1CL\) \[183\]](#)

管理 Oracle Solaris Cluster 法定

[clreslogicalhostname\(1CL\) \[207\]](#)、[clrslh\(1CL\) \[359\]](#)

管理 Oracle Solaris Cluster 逻辑主机名的资源

[clresource\(1CL\) \[225\]](#)、[clrs\(1CL\) \[331\]](#)

管理 Oracle Solaris Cluster 数据服务的资源

[clresourcegroup\(1CL\) \[253\]](#)、[clrg\(1CL\) \[307\]](#)

管理 Oracle Solaris Cluster 数据服务的资源组

[clresourcetype\(1CL\) \[277\]](#)、[clrt\(1CL\) \[395\]](#)

管理 Oracle Solaris Cluster 数据服务的资源类型

[clressharedaddress\(1CL\) \[289\]](#)、[clrssa\(1CL\) \[377\]](#)

管理 Oracle Solaris Cluster 共享地址的资源

[clsetup\(1CL\) \[407\]](#)

以交互方式配置 Oracle Solaris Cluster

[clsnmphost\(1CL\) \[409\]](#)

管理 Oracle Solaris Cluster SNMP 主机

[clsnmpmib\(1CL\) \[417\]](#)、[clmib\(1CL\) \[117\]](#)
管理单个 Oracle Solaris Cluster SNMP MIB

[clsnmpuser\(1CL\) \[425\]](#)
管理 Oracle Solaris Cluster SNMP 用户

[cltelemetryattribute\(1CL\) \[447\]](#)
配置系统资源监视

[cluster\(1CL\) \[461\]](#)
管理 Sun Cluster 的全局配置和全局状态

[clzonecluster\(1CL\) \[517\]](#)、[clzc\(1CL\) \[491\]](#)
管理 Oracle Solaris Cluster 的区域群集

将原始的 Oracle Solaris Cluster 命令映射到面向对象的命令

由于较新的命令集面向对象，因此不存在与原始命令集的清晰的一对一映射。以下列表提供来自原始集的一些通用 Oracle Solaris Cluster 命令及其面向对象集中的对等项。

<code>scstat</code>	<code>cluster status</code> 还可以使用可用于许多面向对象的命令的 <code>status</code> 子命令。
<code>scinstall</code>	使用 <code>cluster create</code> 可从 XML 配置文件创建群集。 要以交互方式创建群集，请使用 <code>scinstall</code> 。
<code>scrgadm</code>	<ul style="list-style-type: none">▪ <code>clresource</code>▪ <code>clresourcetype</code>▪ <code>clresourcegroup</code> 当您使用这些特定资源类型时， <code>clressharedaddress</code> 和 <code>clreslogicalhostname</code> 将提供额外的便利。
<code>scswitch</code>	<ul style="list-style-type: none">▪ <code>clresource</code>▪ <code>clresourcetype</code>▪ <code>clresourcegroup</code>▪ <code>clreslogicalhostname</code>▪ <code>clressharedaddress</code>▪ <code>clnode evacuate</code> (从节点移除所有资源组和设备组)
<code>scconf</code>	<ul style="list-style-type: none">▪ <code>cldevicegroup</code>

-
- clinterconnect
 - clquorum
 - clnode
 - claccess

使用 `cluster show` 而非 `scconf -p`。

<code>sccheck</code>	<code>cluster check</code>
<code>scdidadm</code>	<code>cldevice</code>
<code>scgdevs</code>	<code>cldevice populate</code>
<code>scdpm</code>	<code>cldevice</code>
<code>scnas, scnasdir</code>	<code>clnasdevice</code>
<code>scsetup</code>	<code>clsetup</code>

如果面向对象的 Oracle Solaris Cluster 命令对于所有指定的操作数均成功执行，命令将返回零 (CL_NOERR)。如果某个操作数出错，命令会继续处理操作数列表中的下一个操作数。返回的退出代码始终反映最先发生的错误。

这些退出代码在整个命令集中是共享的。

0 CL_NOERR

没有错误

您发出的命令已成功完成。

1 CL_ENOMEM

交换空间不足

某个群集节点耗尽了交换内存或其他操作系统资源。

3 CL_EINVAL

参数无效

您键入的命令不正确，或者使用 `-i` 选项提供的群集配置信息语法不正确。

5 CL_ERECONF

正在重新配置群集

正在重新配置群集。

6 CL_EACCESS

权限被拒绝

指定的对象不可访问。要发出命令，可能需要拥有超级用户或 RBAC 访问权限。有关更多信息，请参见 [Unresolved link to " su1M"](#) 和 [Unresolved link to " rbac5"](#) 手册页。

9 CL_ESTATE

对象处于错误状况

您尝试修改一个在特定时间无法修改，或任何时候均无法修改的属性、资源组或其他对象。

10 CL_EMETHOD

资源方法失败

资源方法失败。方法因以下某个原因而失败：

- 尝试创建资源或修改资源属性时，`validate` 方法失败。
- 尝试启用、禁用或删除资源时，`validate` 以外的其他方法失败。

15 CL_EPROP

属性无效

您使用 `-p`、`-y` 或 `-x` 选项指定的属性或值不存在或不被允许。

18 CL_EINTERNAL

遇到内部错误

内部错误表示软件缺陷或其他缺陷。

35 CL_EIO

I/O 错误

发生了物理输入/输出错误。

36 CL_ENOENT

没有这样的对象

由于以下某个原因，找不到您所指定的对象：

- 该对象不存在。
- 您尝试使用 `-o` 选项创建的配置文件路径中的某个目录不存在。
- 您尝试使用 `-i` 选项访问的配置文件包含错误。

37 CL_EOP

不允许操作

您尝试对不受支持的配置执行操作，或者您所执行的操作不受支持。

38 CL_EBUSY

对象忙

您尝试将一根电缆从上一个群集互连路径移到一个活动群集节点。或者，您尝试将一个节点从尚未删除引用的群集配置中移除。

39 CL_EEXIST

对象已存在

您所指定的设备、设备组、群集互连组件、节点、群集、资源、资源类型、资源组或专用字符串已存在。

41 CL_ETYPE

类型无效

您使用 -t 或 -p 选项指定的类型不存在。

50 CL_ECLMODE

节点处于群集模式

您尝试对以群集模式引导的节点执行某项操作。但是，您只能对以非群集模式引导的节点执行此操作。

51 CL_ENOTCLMODE

节点不处于群集模式

您尝试对以非群集模式引导的节点执行某项操作。但是，您只能对以群集模式引导的节点执行此操作。

[Unresolved link to " getopt1"](#)

有关下列属性的说明，请参见 [Unresolved link to " attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

OSC4 1

名称

libschost.so.1 — 共享对象提供逻辑主机名而不是物理主机名

libschost.so.1

libschost.so.1 共享对象提供一种机制，通过该机制可为启动的进程及其后代进程选择性配置物理主机名。

在 Oracle Solaris Cluster 环境中，应用程序可能会在故障转移或切换后尝试访问同一主机名。因此，故障转移或切换失败，因为在故障转移或切换之后物理主机名进行了更改。在这种情况下，应用程序数据服务可以使用 libschoost.so.1 共享对象为应用程序提供逻辑主机名，而不是物理主机名。

要启用 libschoost.so.1，您需要设置 SC_LHOSTNAME 环境变量及以下两种环境变量：

```
LD_PRELOAD_32=$LD_PRELOAD_32:/usr/cluster/lib/libschoost.so.1
LD_PRELOAD_64=$LD_PRELOAD_64:/usr/cluster/lib/64/libschoost.so.1
```

通过设置 LD_PRELOAD_32 和 LD_PRELOAD_64 环境变量，您可确保 libschoost.so.1 共享对象可与 32 位和 64 位应用程序协同工作。

运行时链接程序访问 32 位对象的默认受信目录 /usr/lib/secure 和 64 位对象的默认受信目录 /usr/lib/secure/64。如果您的安全应用程序使用 libschoost.so.1 共享对象，您需要确保从受信目录中访问 libschoost.so.1 共享对象。

要执行此操作，请为 32 位应用程序创建从 /usr/cluster/lib/libschoost.so.1 到 /usr/lib/secure/libschoost.so.1 的符号链接，或为 64 位应用程序创建从 /usr/cluster/lib/64/libschoost.so.1 到 /usr/lib/secure/64/libschoost.so.1 的符号链接。

在创建这些符号链接之后，LD_PRELOAD_32 和 LD_PRELOAD_64 环境变量将使用受信目录中的 libschoost.so.1 共享对象。

您还可以使用 crle 命令指定其他受信目录，或更改安全应用程序的默认受信目录。请参见 [Unresolved link to " crle1" 手册页](#)。

预装入之后，libschoost.so.1 共享对象读取以下环境变量并将其作为主机名返回。

SC_LHOSTNAME=*hostname*

SC_LHOSTNAME 指定逻辑主机名。指定的主机名可用于所有启动的进程和后代进程。

hostname 值最多可为 MAXHOSTNAMELEN 个字符长度。MAXHOSTNAMELEN 常量在 netdb.h 头文件中定义为 256 个字符。

例 1 于 C 中在运行时为主机名配置逻辑主机名

以下示例中的 C 代码将为主机名配置逻辑主机名。该示例包括调用 `scds_get_rs_hostnames()` Oracle Solaris Cluster 函数，还包括引用 `scds_handle_t` 和 `scds_net_resource_list_t` Oracle Solaris Cluster 数据结构。

`scds_get_rs_hostnames()` 函数提供资源所使用的主机名列表。代码将该列表中的第一个主机名值分配给 `SC_LHOSTNAME` 环境变量。

在执行以下代码后启动的任何应用程序均将获得一个逻辑主机名，而不是物理主机名。

```
/* 13 bytes to hold "SC_LHOSTNAME=" string */
#define HOSTLENGTH (MAXHOSTNAMELEN + 13)

/* 14 bytes to hold "LD_PRELOAD_XX=" string */
#define PATHLENGTH (MAXPATHLEN + 14)

char lhostname[HOSTLENGTH], ld_32[PATHLENGTH], \
    ld_64[PATHLENGTH];

scds_get_rs_hostnames(scds_handle, &snrlp);
if (snrlp != NULL && snrlp->num_netresources != 0) {
    snprintf(lhostname, HOSTLENGTH, "SC_LHOSTNAME=%s", \
        snrlp->netresources[0].hostnames[0]);
    putenv(lhostname);
}

/* Setting LD_PRELOAD_32 environment variable */
if (getenv("LD_PRELOAD_32") == NULL)
    snprintf(ld_32, PATHLENGTH, "LD_PRELOAD_32="
        "/usr/cluster/lib/libschost.so.1");
else
    snprintf(ld_32, PATHLENGTH, "LD_PRELOAD_32=%s:"
        "/usr/cluster/lib/libschost.so.1", \
        getenv("LD_PRELOAD_32"));

putenv(ld_32);

/* Setting LD_PRELOAD_64 environment variable */
if (getenv("LD_PRELOAD_64") == NULL)
    snprintf(ld_64, PATHLENGTH, "LD_PRELOAD_64="
        "/usr/cluster/lib/64/libschost.so.1");
else
    snprintf(ld_64, PATHLENGTH,
        "LD_PRELOAD_64=%s:/usr/cluster/lib/"
        "64/libschost.so.1", getenv("LD_PRELOAD_64"));

putenv(ld_64);
```

例 2 使用 Shell 命令在运行时为主机名配置逻辑主机名

以下示例中的 shell 命令显示了应用程序数据服务如何使用 `gethostnames` 命令为主机名配置逻辑主机名。`gethostnames` 命令采用以下参数：

- `-R resource-name`
- `-G resourcegroup-name`
- `-T resourcetype-name`

`gethostnames` 命令返回与该资源关联的所有逻辑主机名，并以分号 (;) 分隔。命令将该列表中的第一个主机名值分配给 `SC_LHOSTNAME` 环境变量。

```
phys-schost-1$ LD_PRELOAD_32=$LD_PRELOAD_32:/usr/cluster/lib/libschost.so.1
phys-schost-1$ LD_PRELOAD_64=$LD_PRELOAD_64:/usr/cluster/lib/64/libschost.so.1
phys-schost-1$ SC_LHOSTNAME=`/usr/cluster/lib/scdsbuilder/src/scripts/gethostnames \
-R nfs-r -G nfs-rg -T SUNW.nfs:3.1 |cut -f1 -d", "`
phys-schost-1$ export LD_PRELOAD_32 LD_PRELOAD_64 SC_LHOSTNAME
```

例 3 使用 Shell 命令为安全应用程序配置逻辑主机名

以下示例中的 shell 命令将配置逻辑主机名。在执行以下 shell 命令后启动的任何安全应用程序均将获得 `SC_LHOSTNAME` 环境变量值（即，逻辑主机名），而不是物理主机名。

```
phys-schost-1$ cd /usr/lib/secure
phys-schost-1$ ln -s /usr/cluster/lib/libschost.so.1 .
phys-schost-1$ cd /usr/lib/secure/64
phys-schost-1$ ln -s /usr/cluster/lib/64/libschost.so.1 .
phys-schost-1$ LD_PRELOAD_32=$LD_PRELOAD_32:/usr/lib/secure/libschost.so.1
phys-schost-1$ LD_PRELOAD_64=$LD_PRELOAD_64:/usr/lib/secure/64/libschost.so.1
phys-schost-1$ SC_LHOSTNAME=test
phys-schost-1$ export LD_PRELOAD_32 LD_PRELOAD_64 SC_LHOSTNAME
```

`/usr/cluster/lib/libschost.so.1`

32 位应用程序的默认共享对象位置

`/usr/cluster/lib/64/libschost.so.1`

64 位应用程序的默认共享对象位置

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

Unresolved link to " crle1"、Unresolved link to " cut1"、Unresolved link to " hostname1"、Unresolved link to " ld1"、Unresolved link to " ld.so.11"、Unresolved link to " proc1"、Unresolved link to " uname1"、Unresolved link to " exec2"、Unresolved link to " sysinfo2"、Unresolved link to " uname2"、Unresolved link to " gethostname3C"、Unresolved link to " putenv3C"、Unresolved link to " snprintf3C"、Unresolved link to " system3C"、Unresolved link to " proc4"

会继承逻辑主机名。

通过调用以下命令或函数获取主机名的用户程序可以获得逻辑主机名，而不是物理主机名：

- hostname 命令
- uname 命令
- uname () 函数
- sysinfo () 函数
- gethostname () 函数

通过调用其他命令或函数获取主机名的用户程序无法获得逻辑主机名。

OSC4 1cl

名称

claccess — 管理节点的 Oracle Solaris Cluster 访问策略

```
/usr/cluster/bin/claccess -V
/usr/cluster/bin/claccess [subcommand] -?
/usr/cluster/bin/claccess subcommand [options] -v [hostname[,...]]
/usr/cluster/bin/claccess allow -h hostname[,...]
/usr/cluster/bin/claccess allow-all
/usr/cluster/bin/claccess deny -h hostname[,...]
/usr/cluster/bin/claccess deny-all
/usr/cluster/bin/claccess list
/usr/cluster/bin/claccess set -p protocol=authprotocol
/usr/cluster/bin/claccess show
```

claccess 命令控制尝试访问群集配置的计算机的网络访问策略。claccess 命令没有简短格式。

群集保留可以访问群集配置的计算机的列表。群集还存储验证协议的名称，这些节点使用该验证协议访问群集配置。

当计算机尝试访问群集配置时，例如，当它要求添加到群集配置时（请参见 [clnode\(1CL\) \[149\]](#)），群集将检查该列表以确定节点是否具有访问权限。如果节点具有权限，则该节点将通过验证，且可以访问群集配置。

可以使用 claccess 命令执行以下任务：

- 允许任何新的计算机将自身添加到群集配置中以及将自身从群集配置中删除
- 阻止任何节点将自身添加到群集配置中以及将自身从群集配置中删除
- 控制用于检查的验证类型

仅可以在全局区域中使用此命令。

claccess 命令的一般格式如下所示：

```
claccess [subcommand] [options]
```

仅当 options 指定 -? 选项或 -v 选项时，才可以省略 subcommand。

此命令的每个选项都有长格式和简短格式两种格式。在本手册页“选项”部分给出了两种格式的选项，以及对该选项的描述。

子命令

支持以下子命令：

allow

允许一台或多台指定计算机访问群集配置。

超级用户以外的用户需要具有 `solaris.cluster.modify` 基于角色的访问控制 (Role-Based Access Control, RBAC) 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#)。

另请参见 `deny` 和 `allow-all` 子命令的描述。

allow-all

允许所有计算机添加自身以访问群集配置。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#)。

另请参见 `deny-all` 和 `allow` 子命令的描述。

deny

阻止一台或多台指定计算机访问群集配置。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#)。

另请参见 `allow` 和 `deny-all` 子命令的描述。

deny-all

阻止所有计算机访问群集配置。

群集首次配置后，默认设置为对任何节点都没有访问权限。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#)。

另请参见 `allow-all` 和 `deny` 子命令的描述。

list

显示对群集配置拥有访问授权的计算机的名称。如果还要查看验证协议，请使用 `show` 子命令。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#)。

set

将验证协议设置为您使用 `-p` 选项指定的值。默认情况下，系统使用 `sys` 作为验证协议。请参见“选项”中的 `-p` 选项。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#)。

show

显示对群集配置拥有访问权限的计算机的名称。还显示验证协议。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#)。

支持以下选项：

-?

--help

显示帮助信息。使用该选项时，不会执行任何其他处理。

指定此选项时，子命令可有可无。如果仅指定此选项而不使用子命令，将会显示此命令的子命令列表。如果使用子命令且指定此选项，将会显示该子命令的用法选项。

-h *hostname*

--host=*hostname*

-- host *hostname*

指定为其授予或拒绝访问权限的节点的名称。

-p *protocol=authentication-protocol*

--authprotocol=*authentication-protocol*

--authprotocol *authentication-protocol*

指定用于检查计算机对群集配置是否具有访问权限的验证协议。

支持的协议为 `des` 和 `sys` (或 `unix`)。默认的验证类型为提供最少的安全验证的 `sys`。有关添加和删除节点的更多信息，请参见[Unresolved link to "Oracle Solaris Cluster 系统管理指南 中的第 8 章 管理群集节点"](#)。有关这些验证类型的更多信息，请参见[Unresolved link to "Managing Kerberos and Other Authentication Services in Oracle Solaris 11.2 中的第 10 章 Configuring Network Services Authentication"](#)。

-V

--version

显示该命令的版本。

指定此选项时，不要将其与子命令、操作数或其他选项结合使用。子命令、操作数或其他选项将被忽略。-v 选项只显示命令的版本。不会执行任何其他处理。

-v

--verbose

将详细信息显示到标准输出 (stdout)。

如果命令对于所有指定的操作数均成功执行，则返回零 (CL_NOERR)。如果某个操作数出错，命令会继续处理操作数列表中的下一个操作数。返回的退出代码始终反映最先发生的错误。

可返回下列退出代码：

0 CL_NOERR

没有错误
您发出的命令已成功完成。

1 CL_ENOMEM

交换空间不足
某个群集节点耗尽了交换内存或其他操作系统资源。

3 CL_EINVAL

参数无效
您键入的命令不正确，或者使用 `-i` 选项提供的群集配置信息语法不正确。

6 CL_EACCESS

权限被拒绝
指定的对象不可访问。要发出命令，可能需要拥有超级用户或 RBAC 访问权限。有关更多信息，请参见 [Unresolved link to " su1M"](#) 和 [Unresolved link to " rbac5"](#) 手册页。

18 CL_EINTERNAL

遇到内部错误
内部错误表示软件缺陷或其他缺陷。

39 CL_EEXIST

对象已存在
您所指定的设备、设备组、群集互连组件、节点、群集、资源、资源类型、资源组或专用字符串已存在。

例 4 允许新的主机访问

以下 `claccess` 命令允许新的主机访问群集配置。

```
# claccess allow -h phys-schost-1
```

例 5 设置验证类型

以下 `claccess` 命令将当前验证类型设置为 `des`。

```
# claccess set -p protocol=des
```

例 6 拒绝所有主机的访问

以下 `claccess` 命令拒绝所有主机访问群集配置。

```
# claccess deny-all
```

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

[Intro\(1CL\) \[17\]](#)、[clnode\(1CL\) \[149\]](#)、[cluster\(1CL\) \[461\]](#)

超级用户可以运行此命令的所有格式。

任何用户都可以运行带有以下子命令和选项的此命令：

- `-?` 选项
- `-v` 选项

要运行带有其他子命令的此命令，超级用户以外的用户需要具有 RBAC 授权。请参见下表。

子命令	RBAC 授权
<code>allow</code>	<code>solaris.cluster.modify</code>
<code>allow-all</code>	<code>solaris.cluster.modify</code>
<code>deny</code>	<code>solaris.cluster.modify</code>
<code>deny-all</code>	<code>solaris.cluster.modify</code>
<code>list</code>	<code>solaris.cluster.read</code>
<code>set</code>	<code>solaris.cluster.modify</code>
<code>show</code>	<code>solaris.cluster.read</code>

名称

cldevice, cldev — 管理 Oracle Solaris Cluster 设备

```
/usr/cluster/bin/cldevice -V
/usr/cluster/bin/cldevice [Subcommand] -?
/usr/cluster/bin/cldevice subcommand [options] -v [+ | device ...]
/usr/cluster/bin/cldevice check [-n node[,...]] [+]
/usr/cluster/bin/cldevice clear [-n node[,...]] [+]
/usr/cluster/bin/cldevice combine -t replication-type -g
    replication-device-group -d destination-device device
/usr/cluster/bin/cldevice export [-o {- | configfile}] [-n node[,...]]
    [+ | device...]
/usr/cluster/bin/cldevice list [-n node[,...]] [+ | device ...]
/usr/cluster/bin/cldevice monitor [-i {- | clconfigfile}] [-n
    node[,...]] {+ | disk-device ...}
/usr/cluster/bin/cldevice populate
/usr/cluster/bin/cldevice refresh [-n node[,...]] [+]
/usr/cluster/bin/cldevice rename -d destination-device device
/usr/cluster/bin/cldevice repair [-n node[,...]]
    {+ | device ...}
/usr/cluster/bin/cldevice replicate -t replication-type [-S
    source-node] -D destination-node [+]
/usr/cluster/bin/cldevice set
    -p default_fencing={global | pathcount | scsi3 | nofencing | nofencing-noscrub}
    [-n node[,...]] device ...
/usr/cluster/bin/cldevice show [-n node[,...]] [+ | device ...]
/usr/cluster/bin/cldevice status [-s state] [-n node[,...]] [+ |
    [disk-device ]]
/usr/cluster/bin/cldevice unmonitor [-i {- | clconfigfile}]
    [-n node[,...]] {+ | disk-device ...}
```

cldevice 命令管理 Oracle Solaris Cluster 环境中的设备。使用此命令可管理 Oracle Solaris Cluster 设备标识符 (device identifier, DID) 伪设备驱动程序, 以及监视磁盘设备路径。

- DID 驱动程序为设备提供唯一设备 ID, 即使有多个设备路径可用也是如此。有关更多信息, 请参见 [did\(7\) \[1275\]](#) 手册页。

-
- 磁盘路径是群集节点和物理磁盘或 LUN 存储设备之间的连接。磁盘路径包括 Oracle Solaris 内核驱动程序栈、主机总线适配器和任何介入电缆、交换机或网络连接。

cldev 命令是 cldevice 命令的简短格式。您可以使用此命令的任一种格式。

除 list 和 show 子命令以外，必须从联机并处于群集模式的群集节点运行 cldevice 命令。

此命令的一般格式如下所示：

```
cldevice [subcommand] [options] [operands]
```

仅当 options 指定 -? 选项或 -v 选项时，才可以省略 subcommand。

此命令的每个选项都有长格式和简短格式两种格式。在本手册页的“选项”部分给出了每个选项的两种格式，以及对该选项的描述。

有关更多信息，请参见 [Intro\(1CL\) \[17\]](#) 手册页。

仅可以在全局区域中使用此命令。

子命令

支持以下子命令：

check

执行一致性检查，将设备的内核表示与物理设备相比较。一致性检查失败时，会显示错误消息。此过程将继续，直到检查完所有设备。

默认情况下，该子命令只影响当前节点。使用 -n 选项为连接到其他节点的设备执行检查操作。

超级用户以外的用户需要具有 solaris.cluster.read RBAC 授权才能使用该子命令。

clear

删除所有已从当前节点断开的底层设备的 DID 引用。

默认情况下，该子命令只影响当前节点。使用 -n 选项可指定将对其执行清除操作的其他群集节点。

超级用户以外的用户需要具有 solaris.cluster.modify RBAC 授权才能使用该子命令。

combine

组合指定的设备与指定的目标设备。

combine 子命令用于组合源设备的路径与目标设备的路径。该组合的路径会导致产生与目标设备的 DID 实例编号相同的单个 DID 实例编号。使用此子命令可组合与使用 SRDF 复制的 EMC LUN 对应的 DID 实例。

可以使用 combine 子命令手动配置 DID 设备以执行基于存储的复制。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

export

导出群集设备的配置信息。

如果使用 `-o` 选项指定一个文件名，则会将配置信息写入该新文件。如果未提供 `-o` 选项，则会将配置信息写入标准输出。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

list

显示所有设备路径。

如果不提供操作数，或者如果提供加号 (+) 操作数，报告将包含所有设备。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

monitor

打开对指定磁盘路径的监视。

`monitor` 子命令只用于磁盘设备。磁带或其他设备不受该子命令影响。

可以使用 `monitor` 子命令调整磁盘路径监视守护进程 `scdpmd`。有关配置文件的更多信息，请参见 [scdpmd.conf\(4\) \[1075\]](#) 手册页。

默认情况下，该子命令打开对所有节点的路径的监视。

使用 `-i` 选项可指定群集配置文件，从该文件中可设置磁盘路径的监视属性。`-i` 选项启动对在指定文件中标记为已监视的那些磁盘路径的磁盘路径监视。不会对其他磁盘路径进行任何更改。有关群集配置文件的更多信息，请参见 [clconfiguration\(5CL\) \[1245\]](#) 手册页。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

populate

填充全局设备名称空间。

全局设备名称空间挂载在 `/global` 目录下。名称空间由指向物理设备的一组逻辑链接组成。由于 `/dev/global` 目录对群集的每个节点都可见，因此每个物理设备在整个群集中均可见。该可见性意味着，可从群集中的任何节点访问添加到全局设备名称空间的任何磁盘、磁带或 CD-ROM。

`populate` 子命令使管理员能够将新的全局设备附加到全局设备名称空间，而不需要系统重新引导。这些设备可以是磁带驱动器、CD-ROM 驱动器或磁盘驱动器。

在运行 `populate` 子命令之前，必须执行 [Unresolved link to "devfsadm1M"](#) 命令。或者，可以执行重新配置重新引导来重新生成全局设备名称空间，并附加新的全局设备。有关重新配置重新引导的更多信息，请参见 [Unresolved link to "boot1M"](#) 手册页。

必须从作为当前群集成员的节点运行 `populate` 子命令。

`populate` 子命令在远程节点上以异步方式执行其工作。因此，在您发出命令的节点上完成命令并不表示命令已在所有群集节点上完成操作。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

`refresh`

在群集节点上更新基于当前设备树的设备配置信息。该命令对 `rdsk` 和 `rmt` 设备树执行完全搜索。对于之前未识别的每个设备标识符，该命令将分配一个新的 DID 实例编号。此外，还会为每个新识别的设备添加新路径。

默认情况下，该子命令只影响当前节点。结合使用 `-n` 选项和 `refresh` 子命令可指定要对其执行刷新操作的群集节点。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

`rename`

将指定的设备移动到新的 DID 实例编号。

该命令删除与源设备的 DID 实例编号对应的 DID 设备路径，并重新创建具有指定的目标 DID 实例编号的设备路径。可以使用该子命令来恢复意外更改的 DID 实例编号。

在连接到共享存储的所有群集节点上运行 `rename` 子命令后，请运行 `devfsadm` 和 `cldevice populate` 命令来使用配置更改更新全局设备名称空间。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

`repair`

在指定设备上执行修复过程。

默认情况下，该子命令只影响当前节点。使用 `-n` 选项可指定将对其执行修复操作的群集节点。

如果不提供操作数，或者如果提供加号 (+) 操作数，该命令将更新与当前节点连接的所有设备上的配置信息。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

`replicate`

配置要用于基于存储的复制的 DID 设备。

注 - `replicate` 子命令不支持组合 DID 实例与 EMC SRDF。使用 `cldevice combine` 可组合 DID 实例与 SRDF。

`replicate` 子命令用于组合源节点上的每个 DID 实例编号与目标节点上相应的 DID 实例编号。每对复制的设备会合并到单个逻辑 DID 设备中。

默认情况下，当前节点就是源节点。使用 `-s` 选项指定其他源节点。
超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

set

修改指定设备的属性。
使用 `-p` 选项可指定要修改的属性。
超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

show

显示所有指定设备路径的配置报告。
该报告显示设备的路径，以及路径是否受监视。
默认情况下，该子命令显示所有设备的配置信息。
超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

status

显示所有指定磁盘设备路径的状态。
默认情况下，该子命令显示所有节点的所有磁盘路径的状态。
`status` 子命令只用于磁盘设备。报告不包含磁带或其他设备。
超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

unmonitor

关闭对作为命令操作数指定的磁盘路径的监视。
默认情况下，该子命令关闭对所有节点的所有路径的监视。
`unmonitor` 子命令只用于磁盘设备。磁带或其他设备不受该子命令影响。
使用 `-i` 选项可指定从中关闭对磁盘路径的监视的群集配置文件。将会关闭对在指定文件中标记为未监视的那些磁盘路径的磁盘路径监视。不会对其他磁盘路径进行任何更改。有关更多信息，请参见 [clconfiguration\(5CL\) \[1245\]](#) 手册页。
超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

支持以下选项：

`-?`

`--help`

显示帮助信息。
此选项可以单独使用，也可以与子命令结合使用。

-
- 如果单独使用该选项，则显示可用子命令的列表。
 - 如果将该选项与子命令结合使用，将会显示该子命令的用法选项。

使用该选项时，不会执行任何其他处理。

`--D destination-node`
`-destinationnode=destination-node`
`-destinationnode destination-node`

指定要在其上复制设备的目标节点。您可以通过节点名称或节点 ID 来指定一个节点。

-D 选项仅与 `replicate` 子命令一起使用时才有效。

`-d destination-device`
`--device=destination-device`
`--device destination-device`

指定目标设备（用于基于存储的复制）的 DID 实例编号。

请仅将 DID 实例编号和 -d 选项结合使用。请勿使用 DID 名称的其他格式或完整的 UNIX 路径名称来指定目标设备。

-d 选项仅与 `rename` 和 `combine` 子命令一起使用时才有效。

`-g replication-device-group`

指定复制设备组。该选项只能与 `combine` 子命令结合使用。

`-i {- | clconfigfile}`
`--input={- | clconfigfile}`
`--input {- | clconfigfile}`

指定用于监视或取消监视磁盘路径的配置信息。此信息必须符合 [clconfiguration\(5CL\) \[1245\]](#) 手册页中定义的格式。此信息可包含在文件中或者通过标准输入提供。要指定标准输入，请指定减号 (-) 而不是文件名。

-i 选项仅与 `monitor` 和 `unmonitor` 子命令一起使用时才有效。

在命令中指定的选项将覆盖在配置文件中设置的任何选项。如果群集配置文件中缺少配置参数，则必须在命令行中指定这些参数。

`-n node[,...]`
`--node=node[,...]`
`--node node[,...]`

指定子命令只包含使用 -n 选项指定的节点的磁盘路径。您可以通过节点名称或节点 ID 来指定一个节点。

`-o {- | configfile}`
`--output={- | configfile}`
`--output {- | configfile}`

使用 [clconfiguration\(5CL\) \[1245\]](#) 手册页中定义的格式写入磁盘路径配置信息。可以将此信息写入文件或标准输出中。

-o 选项仅与 export 子命令一起使用时才有效。

如果将文件名作为参数提供给该选项，此命令将创建新的文件，配置会显示在该文件中。如果已经存在同名的文件，此命令将退出，并显示一个错误。不会对现有文件进行任何更改。

如果将减号 (-) 作为参数提供给该选项，命令会将配置信息显示到标准输出。命令的所有其他标准输出将受到抑制。

```
-p default_fencing={global | pathcount| scsi3 | nofencing | nofencing-noscrub}
--property=default_fencing={global|pathcount|scsi3|nofencing|nofencing-noscrub}
--property default_fencing={global|pathcount|scsi3|nofencing|nofencing-noscrub}
```

指定要修改的属性。

将该选项与 set 子命令一起使用可修改以下属性：

default_fencing

覆盖指定设备的全局默认隔离算法。无法在配置为法定设备的设备上更改默认隔离算法。

可以将设备的默认隔离算法设置为下列值之一：

global

使用全局默认隔离设置。有关设置全局隔离默认值的信息，请参见 [cluster\(1CL\) \[461\]](#) 手册页。

nofencing

检查并删除任何持久组保留 (Persistent Group Reservation, PGR) 密钥后，请关闭一个或多个指定设备的隔离。



注意 - 如果您使用的是不支持 SCSI 的磁盘（如串行高级技术附件 (Serial Advanced Technology Attachment, SATA) 磁盘），请关闭隔离。

nofencing-noscrub

关闭一个或多个指定设备的隔离，而不首先检查或删除 PGR 密钥。



注意 - 如果您使用的是不支持 SCSI 的磁盘（如串行高级技术附件 (Serial Advanced Technology Attachment, SATA) 磁盘），请关闭隔离。

pathcount

根据连接到共享设备的 DID 路径的数目来确定隔离协议。

- 对于使用少于 3 个 DID 路径的设备，该命令会设置 SCSI-2 协议。
- 对于使用不少于 3 个 DID 路径的设备，该命令将设置 SCSI-3 协议。

scsi3

设置 SCSI-3 协议。如果设备不支持 SCSI-3 协议，则隔离协议设置将保持不变。

`-S source-node`
`--sourcename=Source-node`
`--sourcename source-node`

指定将其中的设备复制到目标节点的源节点。您可以通过节点名称或节点 ID 来指定一个节点。

`-S` 选项仅与 `replicate` 子命令一起使用时才有效。

`-s state[,...]`
`--state=State[,...]`
`--state state[,...]`

显示处于指定状态的磁盘路径的状态信息。

`-s` 选项仅与 `status` 子命令一起使用时才有效。提供 `-s` 选项时，状态输出限制为处于指定 `state` 的磁盘路径。以下为 `state` 的可能值：

- fail
- ok
- unknown
- unmonitored

`-t`

指定复制设备类型。该选项可与 `replicate` 和 `combine` 子命令结合使用。

`-V`
`--version`

显示该命令的版本。

指定此选项时，不要将其与子命令、操作数或其他选项结合使用。子命令、操作数或其他选项将被忽略。`-V` 选项只显示命令的版本。不会执行任何其他操作。

`-v`
`--verbose`

将详细信息显示到标准输出。

可以在任何格式的此命令中指定该选项。

支持以下操作数：

device

指定设备的名称。设备可以为但不限于磁盘、磁带和 CD-ROM。

如果子命令接受多个设备，则可以使用加号 (+) 指定所有设备。

`cldevice` 命令的所有子命令（`repair` 子命令除外）接受设备路径作为操作数。`repair` 子命令只接受设备名称作为操作数。`device` 名称可以为完整的全局路径名称、设备名称或 DID 实例编号。设备名称的这些格式的示例分别为 `/dev/did/dsk/d3`、`d3` 和 `3`。有关更多信息，请参见 [did\(7\) \[1275\]](#) 手册页。

设备名称还可以是完整的 UNIX 路径名称，如 `/dev/rdisk/c0t0d0s0`。

指定的设备可以具有多个将设备连接到节点的路径。如果未使用 `-n` 选项，则选择所有节点到指定设备的所有路径。

`monitor`、`unmonitor` 和 `status` 子命令只接受磁盘设备作为操作数。

[Intro\(1CL\) \[17\]](#) 手册页列出了该命令集中所有命令的一整套退出状态代码。

如果命令对于所有指定的操作数均成功执行，则返回零 (`CL_NOERR`)。如果某个操作数出错，命令会继续处理操作数列表中的下一个操作数。返回的退出代码始终反映最先发生的错误。

此命令返回下列退出状态代码：

```
0 CL_NOERR
    没有错误

1 CL_ENOMEM
    交换空间不足

3 CL_EINVAL
    参数无效

6 CL_EACCESS
    权限被拒绝

9 CL_ESTATE
    对象处于错误状况

15CL_EPROP
    属性无效

35 CL_EIO
    I/O 错误

36 CL_ENOENT
    没有这样的对象

37 CL_EOP
    不允许操作
```

例 7 监视群集中的所有磁盘路径

以下示例显示如何启用对群集基础结构中的所有磁盘路径的监视。

```
# cldevice monitor +
```

例 8 监视单个磁盘路径

以下示例显示如何在磁盘 `/dev/did/dsk/d3` 的路径有效的所有节点上启用对该路径的监视。

```
# cldevice monitor /dev/did/dsk/d3
```

例 9 监视单个节点上的磁盘路径

以下示例显示如何在节点 `phys-schost-2` 上启用对磁盘 `/dev/did/dsk/d4` 和 `/dev/did/dsk/d5` 的路径的监视。

第一个示例使用 `-n` 选项来限制对连接到节点 `phys-schost-2` 的磁盘路径的监视，然后进一步限制对指定设备 `d4` 和 `d5` 的监视。

```
# cldevice monitor -n phys-schost-2 d4 d5
```

第二个示例通过 `node:device` 名称 `phys-schost-2:d4` 和 `phys-schost-2:d5` 来指定要监视的磁盘路径。

```
# cldevice monitor phys-schost-2:d4 phys-schost-2:d5
```

例 10 显示所有磁盘路径及其状态

以下示例表明如何显示群集中的所有磁盘路径及其状态。

```
# cldevice status
Device Instance      Node                Status
-----
/dev/did/rdisk/d1    phys-schost-2      Unmonitored
/dev/did/rdisk/d2    phys-schost-2      Unmonitored
/dev/did/rdisk/d3    phys-schost-1      Ok
                    phys-schost-2      Ok
/dev/did/rdisk/d4    phys-schost-1      Ok
                    phys-schost-2      Ok
/dev/did/rdisk/d5    phys-schost-1      Unmonitored
```

例 11 显示状态为 `fail` 的所有磁盘路径

以下示例说明如何显示节点 `phys-schost-2` 上受监视且状态为 `fail` 的所有磁盘路径。

```
# cldevice status -s fail -n phys-schost-1
Device Instance      Node                Status
-----
/dev/did/rdisk/d3    phys-schost-1      Fail
/dev/did/rdisk/d4    phys-schost-1      Fail
```

例 12 显示单个节点上所有磁盘路径的状态

以下示例表明如何显示在节点 `phys-schost-2` 上联机的所有磁盘路径的路径及状态。

```
# cldevice status -n phys-schost-1
Device Instance      Node                Status
-----
/dev/did/rdisk/d3    phys-schost-1      Ok
/dev/did/rdisk/d4    phys-schost-1      Ok
/dev/did/rdisk/d5    phys-schost-1      Unmonitored
```

例 13 将新的设备添加到设备配置数据库

以下示例显示如何使用从中发出命令的节点 `phys-schost-2` 的当前设备配置来更新 CCR 数据库。此命令不更新与群集中任何其他节点连接的设备的数据库。

```
phys-schost-2# cldevice refresh
```

例 14 组合单个 DID 下的设备

以下示例显示如何组合一个设备的路径与另一个设备的路径。该组合的路径会导致产生与目标设备的 DID 实例编号相同的单个 DID 实例编号。

```
# cldevice combine -t srdf -g devgrp1 -d 20 30
```

例 15 列出设备实例的设备路径

以下示例显示如何列出与 DID 驱动程序的实例 3 对应的所有设备的路径。

```
# cldevice list 3
d3
```

例 16 列出群集中的所有设备路径

以下示例显示如何列出与任何群集节点连接的所有设备的所有设备路径。

```
# cldevice list -v
DID Device          Full Device Path
-----
d1                  phys-schost-1:/dev/rdisk/c0t0d0
d2                  phys-schost-1:/dev/rdisk/c0t1d0
d3                  phys-schost-1:/dev/rdisk/c1t8d0
d3                  phys-schost-2:/dev/rdisk/c1t8d0
d4                  phys-schost-1:/dev/rdisk/c1t9d0
d4                  phys-schost-2:/dev/rdisk/c1t9d0
d5                  phys-schost-1:/dev/rdisk/c1t10d0
d5                  phys-schost-2:/dev/rdisk/c1t10d0
```

```
d6          phys-schost-1:/dev/rdisk/c1t11d0
d6          phys-schost-2:/dev/rdisk/c1t11d0
d7          phys-schost-2:/dev/rdisk/c0t0d0
d8          phys-schost-2:/dev/rdisk/c0t1d0
```

例 17 显示有关设备的配置信息

以下示例说明如何显示设备 `c4t8d0` 的相关配置信息。

```
# cldevice show /dev/rdsk/c4t8d0
```

```
=== DID Device Instances ===
```

```
DID Device Name:          /dev/did/rdsk/d3
Full Device Path:        phys-schost1:/dev/rdsk/c4t8d0
Full Device Path:        phys-schost2:/dev/rdsk/c4t8d0
Replication:             none
default_fencing:         nofencing
```

例 18 设置单个设备的 SCSI 协议

以下示例为设备 11（由实例编号指定）设置 SCSI-3 协议。该设备不是配置的法定设备。

```
# cldevice set -p default_fencing=scsi3 11
```

例 19 关闭设备的隔离，而不首先检查 PGR 密钥

以下示例关闭设备上磁盘 `/dev/did/dsk/d5` 的隔离。此命令关闭设备的隔离，而不首先检查和删除任何持久组保留 (Persistent Group Reservation, PGR) 密钥。

```
# cldevice set -p default_fencing=no fencing-noscrub d5
```

如果您使用的是不支持 SCSI 的磁盘（如串行高级技术附件 (Serial Advanced Technology Attachment, SATA) 磁盘），请关闭 SCSI 隔离。

例 20 为双节点群集 `phys-schost` 中的所有设备关闭隔离

以下示例关闭名为 `phys-schost` 的双节点群集中所有磁盘的隔离。

```
# cluster set -p global_fencing=no fencing
# cldevice set -p default_fencing=global -n phys-schost-1,phys-schost-2 d5
```

有关 `cluster` 命令和 `global_fencing` 属性的更多信息，请参见 [cluster\(1CL\) \[461\]](#) 手册页。

如果您使用的是不支持 SCSI 的磁盘（如串行高级技术附件 (Serial Advanced Technology Attachment, SATA) 磁盘），请关闭 SCSI 隔离。

例 21 通过使用设备名称执行修复过程

以下示例显示如何对与设备 `/dev/dsk/clt4d0` 关联的设备标识符执行修复过程。此设备已替换为新的设备标识符现在与其关联的新设备。在数据库中，`repair` 子命令记录现在实例编号对应于新的设备标识符。

```
# cldevice repair clt4d0
```

例 22 通过使用实例编号执行修复过程

以下示例显示如何提供备用方法对设备标识符执行修复过程。该示例指定与替换设备的设备路径关联的实例编号。替换设备的实例编号为 2。

```
# cldevice repair 2
```

例 23 填充全局设备名称空间

以下示例说明在添加新的全局设备或将 DID 设备移动到新的实例编号之后如何填充全局设备名称空间。

```
# devfsadm
# cldevice populate
```

例 24 移动 DID 设备

以下示例将源实例上的 DID 实例 15 移动到新的 DID 实例 10，然后使用配置更改更新全局设备名称空间。

```
# cldevice rename 15:10
# devfsadm
# cldevice populate
```

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

[Intro\(1CL\) \[17\]](#)、[cluster\(1CL\) \[461\]](#)、[Unresolved link to "boot1M"](#)、[Unresolved link to "devfsadm1M"](#)、[clconfiguration\(5CL\) \[1245\]](#)、[Unresolved link to "rbac5"](#)、[did\(7\) \[1275\]](#)

超级用户可以运行此命令的所有格式。

任何用户都可以运行带有以下选项的此命令：

- -? (帮助) 选项
- -V (版本) 选项

要运行带有其他子命令的此命令，超级用户以外的用户需要具有 RBAC 授权。请参见下表。

子命令	RBAC 授权
check	solaris.cluster.read
clear	solaris.cluster.modify
combine	solaris.cluster.modify
export	solaris.cluster.read
list	solaris.cluster.read
monitor	solaris.cluster.modify
populate	solaris.cluster.modify
refresh	solaris.cluster.modify
rename	solaris.cluster.modify
repair	solaris.cluster.modify
replicate	solaris.cluster.modify
set	solaris.cluster.modify
show	solaris.cluster.read
status	solaris.cluster.read
unmonitor	solaris.cluster.modify

磁盘路径状态更改是通过使用 `syslogd` 命令记录的。

每次进行物理连接时，每个多端口的磁带驱动器或 CD-ROM 驱动器在名称空间中显示一次。

名称

cldevice, cldev — 管理 Oracle Solaris Cluster 设备

```
/usr/cluster/bin/cldevice -V
/usr/cluster/bin/cldevice [Subcommand] -?
/usr/cluster/bin/cldevice subcommand [options] -v [+ | device ...]
/usr/cluster/bin/cldevice check [-n node[,...]] [+]
/usr/cluster/bin/cldevice clear [-n node[,...]] [+]
/usr/cluster/bin/cldevice combine -t replication-type -g
    replication-device-group -d destination-device device
/usr/cluster/bin/cldevice export [-o {- | configfile}] [-n node[,...]]
    [+ | device...]
/usr/cluster/bin/cldevice list [-n node[,...]] [+ | device ...]
/usr/cluster/bin/cldevice monitor [-i {- | clconfigfile}] [-n
    node[,...]] {+ | disk-device ...}
/usr/cluster/bin/cldevice populate
/usr/cluster/bin/cldevice refresh [-n node[,...]] [+]
/usr/cluster/bin/cldevice rename -d destination-device device
/usr/cluster/bin/cldevice repair [-n node[,...]]
    {+ | device ...}
/usr/cluster/bin/cldevice replicate -t replication-type [-S
    source-node] -D destination-node [+]
/usr/cluster/bin/cldevice set
    -p default_fencing={global | pathcount | scsi3 | nofencing | nofencing-noscrub}
    [-n node[,...]] device ...
/usr/cluster/bin/cldevice show [-n node[,...]] [+ | device ...]
/usr/cluster/bin/cldevice status [-s state] [-n node[,...]] [+ |
    disk-device ]]
/usr/cluster/bin/cldevice unmonitor [-i {- | clconfigfile}]
    [-n node[,...]] {+ | disk-device ...}
```

cldevice 命令管理 Oracle Solaris Cluster 环境中的设备。使用此命令可管理 Oracle Solaris Cluster 设备标识符 (device identifier, DID) 伪设备驱动程序, 以及监视磁盘设备路径。

- DID 驱动程序为设备提供唯一设备 ID, 即使有多个设备路径可用也是如此。有关更多信息, 请参见 [did\(7\) \[1275\]](#) 手册页。

-
- 磁盘路径是群集节点和物理磁盘或 LUN 存储设备之间的连接。磁盘路径包括 Oracle Solaris 内核驱动程序栈、主机总线适配器和任何介入电缆、交换机或网络连接。

cldev 命令是 cldevice 命令的简短格式。您可以使用此命令的任一种格式。

除 list 和 show 子命令以外，必须从联机并处于群集模式的群集节点运行 cldevice 命令。

此命令的一般格式如下所示：

```
cldevice [subcommand] [options] [operands]
```

仅当 options 指定 -? 选项或 -v 选项时，才可以省略 subcommand。

此命令的每个选项都有长格式和简短格式两种格式。在本手册页的“选项”部分给出了每个选项的两种格式，以及对该选项的描述。

有关更多信息，请参见 [Intro\(1CL\) \[17\]](#) 手册页。

仅可以在全局区域中使用此命令。

子命令

支持以下子命令：

check

执行一致性检查，将设备的内核表示与物理设备相比较。一致性检查失败时，会显示错误消息。此过程将继续，直到检查完所有设备。

默认情况下，该子命令只影响当前节点。使用 -n 选项为连接到其他节点的设备执行检查操作。

超级用户以外的用户需要具有 solaris.cluster.read RBAC 授权才能使用该子命令。

clear

删除所有已从当前节点断开的底层设备的 DID 引用。

默认情况下，该子命令只影响当前节点。使用 -n 选项可指定将对其执行清除操作的其他群集节点。

超级用户以外的用户需要具有 solaris.cluster.modify RBAC 授权才能使用该子命令。

combine

组合指定的设备与指定的目标设备。

combine 子命令用于组合源设备的路径与目标设备的路径。该组合的路径会导致产生与目标设备的 DID 实例编号相同的单个 DID 实例编号。使用此子命令可组合与使用 SRDF 复制的 EMC LUN 对应的 DID 实例。

可以使用 combine 子命令手动配置 DID 设备以执行基于存储的复制。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

export

导出群集设备的配置信息。

如果使用 `-o` 选项指定一个文件名，则会将配置信息写入该新文件。如果未提供 `-o` 选项，则会将配置信息写入标准输出。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

list

显示所有设备路径。

如果不提供操作数，或者如果提供加号 (+) 操作数，报告将包含所有设备。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

monitor

打开对指定磁盘路径的监视。

`monitor` 子命令只用于磁盘设备。磁带或其他设备不受该子命令影响。

可以使用 `monitor` 子命令调整磁盘路径监视守护进程 `scdpmd`。有关配置文件的更多信息，请参见 [scdpmd.conf\(4\) \[1075\]](#) 手册页。

默认情况下，该子命令打开对所有节点的路径的监视。

使用 `-i` 选项可指定群集配置文件，从该文件中可设置磁盘路径的监视属性。`-i` 选项启动对在指定文件中标记为已监视的那些磁盘路径的磁盘路径监视。不会对其他磁盘路径进行任何更改。有关群集配置文件的更多信息，请参见 [clconfiguration\(5CL\) \[1245\]](#) 手册页。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

populate

填充全局设备名称空间。

全局设备名称空间挂载在 `/global` 目录下。名称空间由指向物理设备的一组逻辑链接组成。由于 `/dev/global` 目录对群集的每个节点都可见，因此每个物理设备在整个群集中均可见。该可见性意味着，可从群集中的任何节点访问添加到全局设备名称空间的任何磁盘、磁带或 CD-ROM。

`populate` 子命令使管理员能够将新的全局设备附加到全局设备名称空间，而不需要系统重新引导。这些设备可以是磁带驱动器、CD-ROM 驱动器或磁盘驱动器。

在运行 `populate` 子命令之前，必须执行 [Unresolved link to "devfsadm1M"](#) 命令。或者，可以执行重新配置重新引导来重新生成全局设备名称空间，并附加新的全局设备。有关重新配置重新引导的更多信息，请参见 [Unresolved link to "boot1M"](#) 手册页。

必须从作为当前群集成员的节点运行 `populate` 子命令。

`populate` 子命令在远程节点上以异步方式执行其工作。因此，在您发出命令的节点上完成命令并不表示命令已在所有群集节点上完成操作。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

`refresh`

在群集节点上更新基于当前设备树的设备配置信息。该命令对 `rdsk` 和 `rmt` 设备树执行完全搜索。对于之前未识别的每个设备标识符，该命令将分配一个新的 DID 实例编号。此外，还会为每个新识别的设备添加新路径。

默认情况下，该子命令只影响当前节点。结合使用 `-n` 选项和 `refresh` 子命令可指定要对其执行刷新操作的群集节点。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

`rename`

将指定的设备移动到新的 DID 实例编号。

该命令删除与源设备的 DID 实例编号对应的 DID 设备路径，并重新创建具有指定的目标 DID 实例编号的设备路径。可以使用该子命令来恢复意外更改的 DID 实例编号。

在连接到共享存储的所有群集节点上运行 `rename` 子命令后，请运行 `devfsadm` 和 `cldevice populate` 命令来使用配置更改更新全局设备名称空间。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

`repair`

在指定设备上执行修复过程。

默认情况下，该子命令只影响当前节点。使用 `-n` 选项可指定将对其执行修复操作的群集节点。

如果不提供操作数，或者如果提供加号 (+) 操作数，该命令将更新与当前节点连接的所有设备上的配置信息。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

`replicate`

配置要用于基于存储的复制的 DID 设备。

注 - `replicate` 子命令不支持组合 DID 实例与 EMC SRDF。使用 `cldevice combine` 可组合 DID 实例与 SRDF。

`replicate` 子命令用于组合源节点上的每个 DID 实例编号与目标节点上相应的 DID 实例编号。每对复制的设备会合并到单个逻辑 DID 设备中。

默认情况下，当前节点就是源节点。使用 `-s` 选项指定其他源节点。
超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

set

修改指定设备的属性。
使用 `-p` 选项可指定要修改的属性。
超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

show

显示所有指定设备路径的配置报告。
该报告显示设备的路径，以及路径是否受监视。
默认情况下，该子命令显示所有设备的配置信息。
超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

status

显示所有指定磁盘设备路径的状态。
默认情况下，该子命令显示所有节点的所有磁盘路径的状态。
`status` 子命令只用于磁盘设备。报告不包含磁带或其他设备。
超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

unmonitor

关闭对作为命令操作数指定的磁盘路径的监视。
默认情况下，该子命令关闭对所有节点的所有路径的监视。
`unmonitor` 子命令只用于磁盘设备。磁带或其他设备不受该子命令影响。
使用 `-i` 选项可指定从中关闭对磁盘路径的监视的群集配置文件。将会关闭对在指定文件中标记为未监视的那些磁盘路径的磁盘路径监视。不会对其他磁盘路径进行任何更改。有关更多信息，请参见 [clconfiguration\(5CL\) \[1245\]](#) 手册页。
超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

支持以下选项：

`-?`

`--help`

显示帮助信息。
此选项可以单独使用，也可以与子命令结合使用。

- 如果单独使用该选项，则显示可用子命令的列表。
- 如果将该选项与子命令结合使用，将会显示该子命令的用法选项。

使用该选项时，不会执行任何其他处理。

`--D destination-node`
`-destinationnode=destination-node`
`-destinationnode destination-node`

指定要在其上复制设备的目标节点。您可以通过节点名称或节点 ID 来指定一个节点。

-D 选项仅与 `replicate` 子命令一起使用时才有效。

`-d destination-device`
`--device=destination-device`
`--device destination-device`

指定目标设备（用于基于存储的复制）的 DID 实例编号。

请仅将 DID 实例编号和 -d 选项结合使用。请勿使用 DID 名称的其他格式或完整的 UNIX 路径名称来指定目标设备。

-d 选项仅与 `rename` 和 `combine` 子命令一起使用时才有效。

`-g replication-device-group`

指定复制设备组。该选项只能与 `combine` 子命令结合使用。

`-i {- | clconfigfile}`
`--input={- | clconfigfile}`
`--input {- | clconfigfile}`

指定用于监视或取消监视磁盘路径的配置信息。此信息必须符合 [clconfiguration\(5CL\) \[1245\]](#) 手册页中定义的格式。此信息可包含在文件中或者通过标准输入提供。要指定标准输入，请指定减号 (-) 而不是文件名。

-i 选项仅与 `monitor` 和 `unmonitor` 子命令一起使用时才有效。

在命令中指定的选项将覆盖在配置文件中设置的任何选项。如果群集配置文件中缺少配置参数，则必须在命令行中指定这些参数。

`-n node[,...]`
`--node=node[,...]`
`--node node[,...]`

指定子命令只包含使用 -n 选项指定的节点的磁盘路径。您可以通过节点名称或节点 ID 来指定一个节点。

`-o {- | configfile}`
`--output={- | configfile}`
`--output {- | configfile}`

使用 [clconfiguration\(5CL\) \[1245\]](#) 手册页中定义的格式写入磁盘路径配置信息。可以将此信息写入文件或标准输出中。

-o 选项仅与 export 子命令一起使用时才有效。

如果将文件名作为参数提供给该选项，此命令将创建新的文件，配置会显示在该文件中。如果已经存在同名的文件，此命令将退出，并显示一个错误。不会对现有文件进行任何更改。

如果将减号 (-) 作为参数提供给该选项，命令会将配置信息显示到标准输出。命令的所有其他标准输出将受到抑制。

```
-p default_fencing={global | pathcount| scsi3 | nofencing | nofencing-noscrub}
--property=default_fencing={global|pathcount|scsi3|nofencing|nofencing-noscrub}
--property default_fencing={global|pathcount|scsi3|nofencing|nofencing-noscrub}
```

指定要修改的属性。

将该选项与 set 子命令一起使用可修改以下属性：

default_fencing

覆盖指定设备的全局默认隔离算法。无法在配置为法定设备的设备上更改默认隔离算法。

可以将设备的默认隔离算法设置为下列值之一：

global

使用全局默认隔离设置。有关设置全局隔离默认值的信息，请参见 [cluster\(1CL\) \[461\]](#) 手册页。

nofencing

检查并删除任何持久组保留 (Persistent Group Reservation, PGR) 密钥后，请关闭一个或多个指定设备的隔离。



注意 - 如果您使用的是不支持 SCSI 的磁盘（如串行高级技术附件 (Serial Advanced Technology Attachment, SATA) 磁盘），请关闭隔离。

nofencing-noscrub

关闭一个或多个指定设备的隔离，而不首先检查或删除 PGR 密钥。



注意 - 如果您使用的是不支持 SCSI 的磁盘（如串行高级技术附件 (Serial Advanced Technology Attachment, SATA) 磁盘），请关闭隔离。

pathcount

根据连接到共享设备的 DID 路径的数目来确定隔离协议。

- 对于使用少于 3 个 DID 路径的设备，该命令会设置 SCSI-2 协议。
- 对于使用不少于 3 个 DID 路径的设备，该命令将设置 SCSI-3 协议。

scsi3

设置 SCSI-3 协议。如果设备不支持 SCSI-3 协议，则隔离协议设置将保持不变。

`-S source-node`
`--sourcename=Source-node`
`--sourcename source-node`

指定将其中的设备复制到目标节点的源节点。您可以通过节点名称或节点 ID 来指定一个节点。

`-S` 选项仅与 `replicate` 子命令一起使用时才有效。

`-s state[,...]`
`--state=State[,...]`
`--state state[,...]`

显示处于指定状态的磁盘路径的状态信息。

`-s` 选项仅与 `status` 子命令一起使用时才有效。提供 `-s` 选项时，状态输出限制为处于指定 `state` 的磁盘路径。以下为 `state` 的可能值：

- fail
- ok
- unknown
- unmonitored

`-t`

指定复制设备类型。该选项可与 `replicate` 和 `combine` 子命令结合使用。

`-V`
`--version`

显示该命令的版本。

指定此选项时，不要将其与子命令、操作数或其他选项结合使用。子命令、操作数或其他选项将被忽略。`-V` 选项只显示命令的版本。不会执行任何其他操作。

`-v`
`--verbose`

将详细信息显示到标准输出。

可以在任何格式的此命令中指定该选项。

支持以下操作数：

device

指定设备的名称。设备可以为但不限于磁盘、磁带和 CD-ROM。

如果子命令接受多个设备，则可以使用加号 (+) 指定所有设备。

`cldevice` 命令的所有子命令（`repair` 子命令除外）接受设备路径作为操作数。`repair` 子命令只接受设备名称作为操作数。`device` 名称可以为完整的全局路径名称、设备名称或 DID 实例编号。设备名称的这些格式的示例分别为 `/dev/did/dsk/d3`、`d3` 和 `3`。有关更多信息，请参见 [did\(7\) \[1275\]](#) 手册页。

设备名称还可以是完整的 UNIX 路径名称，如 `/dev/rdisk/c0t0d0s0`。

指定的设备可以具有多个将设备连接到节点的路径。如果未使用 `-n` 选项，则选择所有节点到指定设备的所有路径。

`monitor`、`unmonitor` 和 `status` 子命令只接受磁盘设备作为操作数。

[Intro\(1CL\) \[17\]](#) 手册页列出了该命令集中所有命令的一整套退出状态代码。

如果命令对于所有指定的操作数均成功执行，则返回零 (`CL_NOERR`)。如果某个操作数出错，命令会继续处理操作数列表中的下一个操作数。返回的退出代码始终反映最先发生的错误。

此命令返回下列退出状态代码：

```
0 CL_NOERR
    没有错误

1 CL_ENOMEM
    交换空间不足

3 CL_EINVAL
    参数无效

6 CL_EACCESS
    权限被拒绝

9 CL_ESTATE
    对象处于错误状况

15CL_EPROP
    属性无效

35 CL_EIO
    I/O 错误

36 CL_ENOENT
    没有这样的对象

37 CL_EOP
    不允许操作
```

例 25 监视群集中的所有磁盘路径

以下示例显示如何启用对群集基础结构中的所有磁盘路径的监视。

```
# cldevice monitor +
```

例 26 监视单个磁盘路径

以下示例显示如何在磁盘 `/dev/did/dsk/d3` 的路径有效的所有节点上启用对该路径的监视。

```
# cldevice monitor /dev/did/dsk/d3
```

例 27 监视单个节点上的磁盘路径

以下示例显示如何在节点 `phys-schost-2` 上启用对磁盘 `/dev/did/dsk/d4` 和 `/dev/did/dsk/d5` 的路径的监视。

第一个示例使用 `-n` 选项来限制对连接到节点 `phys-schost-2` 的磁盘路径的监视，然后进一步限制对指定设备 `d4` 和 `d5` 的监视。

```
# cldevice monitor -n phys-schost-2 d4 d5
```

第二个示例通过 `node:device` 名称 `phys-schost-2:d4` 和 `phys-schost-2:d5` 来指定要监视的磁盘路径。

```
# cldevice monitor phys-schost-2:d4 phys-schost-2:d5
```

例 28 显示所有磁盘路径及其状态

以下示例表明如何显示群集中的所有磁盘路径及其状态。

```
# cldevice status
Device Instance      Node                Status
-----
/dev/did/rdisk/d1    phys-schost-2      Unmonitored
/dev/did/rdisk/d2    phys-schost-2      Unmonitored
/dev/did/rdisk/d3    phys-schost-1      Ok
                    phys-schost-2      Ok
/dev/did/rdisk/d4    phys-schost-1      Ok
                    phys-schost-2      Ok
/dev/did/rdisk/d5    phys-schost-1      Unmonitored
```

例 29 显示状态为 `fail` 的所有磁盘路径

以下示例说明如何显示节点 `phys-schost-2` 上受监视且状态为 `fail` 的所有磁盘路径。

```
# cldevice status -s fail -n phys-schost-1
Device Instance      Node                Status
-----
/dev/did/rdisk/d3    phys-schost-1      Fail
/dev/did/rdisk/d4    phys-schost-1      Fail
```

例 30 显示单个节点上所有磁盘路径的状态

以下示例表明如何显示在节点 `phys-schost-2` 上联机的所有磁盘路径的路径及状态。

```
# cldevice status -n phys-schost-1
Device Instance      Node                Status
-----
/dev/did/rdisk/d3    phys-schost-1      Ok
/dev/did/rdisk/d4    phys-schost-1      Ok
/dev/did/rdisk/d5    phys-schost-1      Unmonitored
```

例 31 将新的设备添加到设备配置数据库

以下示例显示如何使用从中发出命令的节点 `phys-schost-2` 的当前设备配置来更新 CCR 数据库。此命令不更新与群集中任何其他节点连接的设备的数据库。

```
phys-schost-2# cldevice refresh
```

例 32 组合单个 DID 下的设备

以下示例显示如何组合一个设备的路径与另一个设备的路径。该组合的路径会导致产生与目标设备的 DID 实例编号相同的单个 DID 实例编号。

```
# cldevice combine -t srdf -g devgrp1 -d 20 30
```

例 33 列出设备实例的设备路径

以下示例显示如何列出与 DID 驱动程序的实例 3 对应的所有设备的路径。

```
# cldevice list 3
d3
```

例 34 列出群集中的所有设备路径

以下示例显示如何列出与任何群集节点连接的所有设备的所有设备路径。

```
# cldevice list -v
DID Device          Full Device Path
-----
d1                  phys-schost-1:/dev/rdisk/c0t0d0
d2                  phys-schost-1:/dev/rdisk/c0t1d0
d3                  phys-schost-1:/dev/rdisk/c1t8d0
d3                  phys-schost-2:/dev/rdisk/c1t8d0
d4                  phys-schost-1:/dev/rdisk/c1t9d0
d4                  phys-schost-2:/dev/rdisk/c1t9d0
d5                  phys-schost-1:/dev/rdisk/c1t10d0
d5                  phys-schost-2:/dev/rdisk/c1t10d0
```

```
d6          phys-schost-1:/dev/rdisk/c1t11d0
d6          phys-schost-2:/dev/rdisk/c1t11d0
d7          phys-schost-2:/dev/rdisk/c0t0d0
d8          phys-schost-2:/dev/rdisk/c0t1d0
```

例 35 显示有关设备的配置信息

以下示例说明如何显示设备 `c4t8d0` 的相关配置信息。

```
# cldevice show /dev/rdisk/c4t8d0
```

```
=== DID Device Instances ===
```

```
DID Device Name:          /dev/did/rdisk/d3
  Full Device Path:       phys-schost1:/dev/rdisk/c4t8d0
  Full Device Path:       phys-schost2:/dev/rdisk/c4t8d0
  Replication:            none
  default_fencing:        nofencing
```

例 36 设置单个设备的 SCSI 协议

以下示例为设备 11（由实例编号指定）设置 SCSI-3 协议。该设备不是配置的法定设备。

```
# cldevice set -p default_fencing=scsi3 11
```

例 37 关闭设备的隔离，而不首先检查 PGR 密钥

以下示例关闭设备上磁盘 `/dev/did/dsk/d5` 的隔离。此命令关闭设备的隔离，而不首先检查和删除任何持久组保留 (Persistent Group Reservation, PGR) 密钥。

```
# cldevice set -p default_fencing=no fencing-noscrub d5
```

如果您使用的是不支持 SCSI 的磁盘（如串行高级技术附件 (Serial Advanced Technology Attachment, SATA) 磁盘），请关闭 SCSI 隔离。

例 38 为双节点群集 `phys-schost` 中的所有设备关闭隔离

以下示例关闭名为 `phys-schost` 的双节点群集中所有磁盘的隔离。

```
# cluster set -p global_fencing=no fencing
# cldevice set -p default_fencing=global -n phys-schost-1,phys-schost-2 d5
```

有关 `cluster` 命令和 `global_fencing` 属性的更多信息，请参见 [cluster\(1CL\) \[461\]](#) 手册页。

如果您使用的是不支持 SCSI 的磁盘（如串行高级技术附件 (Serial Advanced Technology Attachment, SATA) 磁盘），请关闭 SCSI 隔离。

例 39 通过使用设备名称执行修复过程

以下示例显示如何对与设备 `/dev/dsk/clt4d0` 关联的设备标识符执行修复过程。此设备已替换为新的设备标识符现在与其关联的新设备。在数据库中，`repair` 子命令记录现在实例编号对应于新的设备标识符。

```
# cldevice repair clt4d0
```

例 40 通过使用实例编号执行修复过程

以下示例显示如何提供备用方法对设备标识符执行修复过程。该示例指定与替换设备的设备路径关联的实例编号。替换设备的实例编号为 2。

```
# cldevice repair 2
```

例 41 填充全局设备名称空间

以下示例说明在添加新的全局设备或将 DID 设备移动到新的实例编号之后如何填充全局设备名称空间。

```
# devfsadm
# cldevice populate
```

例 42 移动 DID 设备

以下示例将源实例上的 DID 实例 15 移动到新的 DID 实例 10，然后使用配置更改更新全局设备名称空间。

```
# cldevice rename 15:10
# devfsadm
# cldevice populate
```

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

[Intro\(1CL\) \[17\]](#)、[cluster\(1CL\) \[461\]](#)、[Unresolved link to "boot1M"](#)、[Unresolved link to "devfsadm1M"](#)、[clconfiguration\(5CL\) \[1245\]](#)、[Unresolved link to "rbac5"](#)、[did\(7\) \[1275\]](#)

超级用户可以运行此命令的所有格式。

任何用户都可以运行带有以下选项的此命令：

- -? (帮助) 选项
- -V (版本) 选项

要运行带有其他子命令的此命令，超级用户以外的用户需要具有 RBAC 授权。请参见下表。

子命令	RBAC 授权
check	solaris.cluster.read
clear	solaris.cluster.modify
combine	solaris.cluster.modify
export	solaris.cluster.read
list	solaris.cluster.read
monitor	solaris.cluster.modify
populate	solaris.cluster.modify
refresh	solaris.cluster.modify
rename	solaris.cluster.modify
repair	solaris.cluster.modify
replicate	solaris.cluster.modify
set	solaris.cluster.modify
show	solaris.cluster.read
status	solaris.cluster.read
unmonitor	solaris.cluster.modify

磁盘路径状态更改是通过使用 `syslogd` 命令记录的。

每次进行物理连接时，每个多端口的磁带驱动器或 CD-ROM 驱动器在名称空间中显示一次。

名称

cldevicegroup, cldg — 管理 Oracle Solaris Cluster 设备组

```
/usr/cluster/bin/cldevicegroup -V

/usr/cluster/bin/cldevicegroup [subcommand] -?

/usr/cluster/bin/cldevicegroup subcommand [options] -v
    [devicegroup ...]

/usr/cluster/bin/cldevicegroup add-device -d device
    [,...] devicegroup

/usr/cluster/bin/cldevicegroup add-node -n node[,...] [-t
    devicegroup-type[,...]] {+ | devicegroup ...}

/usr/cluster/bin/cldevicegroup create -n node[,...] -t
    devicegroup-type [-d device[,...]] [-p name=value]
    devicegroup ...

/usr/cluster/bin/cldevicegroup create -i {- | clconfigfile} [-d
    device[,...]] [-n node[,...]] [-p name=value] [-t
    devicegroup-type[,...]] {+ | devicegroup ...}

/usr/cluster/bin/cldevicegroup delete [-t devicegroup-type[,...]]
    {+ | devicegroup ...}

/usr/cluster/bin/cldevicegroup disable [-t devicegroup-type[,...]]
    {+ | devicegroup ...}

/usr/cluster/bin/cldevicegroup enable [-t devicegroup-type[,...]]
    {+ | devicegroup ...}

/usr/cluster/bin/cldevicegroup export [-n node[,...]] [-o
    {- | clconfigfile}] [-t devicegroup-type[,...]]
    {+ | devicegroup ...}

/usr/cluster/bin/cldevicegroup list [-n node[,...]] [-t
    devicegroup-type[,...]] {+ | devicegroup ...}

/usr/cluster/bin/cldevicegroup offline [-t devicegroup-type[,...]]
    {+ | devicegroup ...}

/usr/cluster/bin/cldevicegroup online [-e] [-n node] [-t
    devicegroup-type[,...]] {+ | devicegroup ...}

/usr/cluster/bin/cldevicegroup remove-device -d device
    [,...] devicegroup

/usr/cluster/bin/cldevicegroup remove-node -n node[,...]
    [-t devicegroup-type[,...]] {+ | devicegroup ...}

/usr/cluster/bin/cldevicegroup set -p name=value [-p name=value]...
```

```
[-d device[,...]] [-n node[,...]] [-t devicegroup-type[,...]]
{+ | devicegroup ...}

/usr/cluster/bin/cldevicegroup show [-n node[,...]] [-t
devicegroup-type[,...]] [+ | devicegroup ...]

/usr/cluster/bin/cldevicegroup status [-n node[,...]] [-t
devicegroup-type[,...]] [+ | devicegroup ...]

/usr/cluster/bin/cldevicegroup switch -n node [-t
devicegroup-type[,...]] [+ | devicegroup ...]

/usr/cluster/bin/cldevicegroup sync [-t devicegroup-type[,...]]
{+ | devicegroup ...}
```

cldevicegroup 命令管理 Oracle Solaris Cluster 设备组。cldg 命令是 cldevicegroup 命令的简短格式。这两个命令是相同的。您可以使用此命令的任一种格式。

此命令的一般格式如下所示：

```
cldevicegroup [subcommand] [options] [operands]
```

仅当 *options* 指定 *-?* 选项或 *-v* 选项时，才可以省略 *subcommand*。

此命令的每个选项都有长格式和简短格式两种格式。在本手册页的“选项”部分给出了每个选项的两种格式，以及对该选项的描述。

除了 *list*、*show* 和 *status* 以外，大多数子命令都需要至少一个操作数。许多子命令接受使用加号 (+) 作为操作数来表示 *all* 适用对象。有关详细信息，请参阅该手册页的“用法概要”和其他部分。

除以下子命令之外，每个子命令都可以用于所有设备组类型：

- *add-device* 和 *remove-device* 子命令只对 *rawdisk* 类型有效。
- *add-node*、*create*、*delete* 和 *remove-node* 子命令仅对 *rawdisk* 类型有效。

仅可以在全局区域中使用此命令。

子命令

支持以下子命令：

add-device

向现有原始磁盘设备组中添加新的成员磁盘设备。

对于现有的 *rawdisk* 类型的设备组，您只能使用 *add-device* 子命令。有关设备组类型的更多信息，请参见 *-t* 选项的描述。

超级用户以外的用户需要具有 *solaris.cluster.modify* RBAC 授权才能使用该子命令。

有关如何从原始磁盘设备组中删除磁盘设备的信息，请参见 `remove-device` 子命令的描述。

`add-node`

向现有设备组中添加新节点。

该子命令仅支持 `rawdisk` 设备组类型。无法使用 Oracle Solaris Cluster 命令向 `svm` 或 `sds` 设备组添加节点。而应改用 Solaris Volume Manager 命令向 Solaris Volume Manager 磁盘集添加节点。磁盘集会自动在 Oracle Solaris Cluster 软件中注册为 `svm` 或 `sds` 设备组。有关设备组类型的更多信息，请参见 `-t` 选项的描述。

如果某个设备组的 `preferenced` 属性设置为 `true`，则无法对该设备组使用该子命令。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

有关如何从设备组中删除节点的信息，请参见 `remove-node` 子命令的描述。

`create`

创建新设备组。

该子命令仅支持 `rawdisk` 设备组类型。无法使用 Oracle Solaris Cluster 命令创建 `svm` 或 `sds` 设备组。而应改用 Solaris Volume Manager 命令创建 Solaris Volume Manager 磁盘集。磁盘集会自动在 Oracle Solaris Cluster 软件中注册为 `svm` 或 `sds` 设备组。有关设备组类型的更多信息，请参见 `-t` 选项的描述。

如果使用 `-i` 选项指定配置文件，则可以提供加号 (+) 作为操作数。使用此操作数时，该命令会创建配置文件中指定的、且尚不存在的所有设备组。

对于 `rawdisk` 类型的设备组，结合使用 `-d` 选项和 `create` 子命令可以为此类设备组指定一个或多个设备。指定设备时，每个命令调用使用一个 `-d` 选项。无法使用一个命令调用创建多个原始磁盘设备组，除非使用 `-i` 选项。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

有关如何删除设备组的信息，请参见 `delete` 子命令的描述。

`delete`

删除设备组。

该子命令仅支持 `rawdisk` 设备组类型。

无法使用 Oracle Solaris Cluster 命令删除 `svm` 或 `sds` 设备组。要删除 `svm` 或 `sds` 设备组，请使用 Solaris Volume Manager 命令删除底层 Solaris Volume Manager 磁盘集。

删除设备组之前，设备组必须处于脱机状态。

如果指定 + 操作数，则只会影响将 `autogen` 属性设置为 `false` 的设备组。为了对在引导时系统自动创建的设备组（已将 `autogen` 属性设置为 `true`）应用命令，您必须明确指定每个设备组。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

有关如何创建设备组的信息，请参见 `create` 子命令的描述。

`disable`

禁用脱机设备组。

重新引导后，设备组的禁用状态不变。

要禁用联机设备组，必须首先使用 `offline` 子命令将设备组脱机。

如果某个设备组当前处于联机状态，则 `disable` 操作将失败，且不会禁用指定的设备组。

无法使用 `switch` 或 `online` 子命令使禁用的设备组联机。必须首先使用 `enable` 子命令清除该设备组的禁用状态。

如果指定 + 操作数，则只会影响将 `autogen` 属性设置为 `false` 的设备组。为了对在引导时系统自动创建的设备组（已将 `autogen` 属性设置为 `true`）应用命令，您必须明确指定每个设备组。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

有关如何启用设备组的信息，请参见 `enable` 子命令的描述。

`enable`

启用设备组。

重新引导后，设备组的禁用状态不变。

要使禁用的设备组联机，必须首先使用 `enable` 子命令清除设备组的禁用状态。

如果指定 + 操作数，则只会影响将 `autogen` 属性设置为 `false` 的设备组。为了对在引导时系统自动创建的设备组（已将 `autogen` 属性设置为 `true`）应用命令，您必须明确指定每个设备组。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

有关如何禁用设备组的信息，请参见 `disable` 子命令的描述。

`export`

导出设备组的配置信息。

如果使用 `-o` 选项指定一个文件名，则会将配置信息写入该新文件。如果未提供 `-o` 选项，则会将输出写入标准输出。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

`list`

显示设备组列表。

默认情况下，该子命令会列出 `autogen` 属性设置为 `false` 的群集中的所有设备组。要显示群集中的所有设备组，还需要指定 `-v` 选项。

如果指定 + 操作数，则只会影响将 `autogen` 属性设置为 `false` 的设备组。为了对在引导时系统自动创建的设备组（已将 `autogen` 属性设置为 `true`）应用命令，您必须明确指定每个设备组。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

offline

使设备组脱机。

如果设备组处于联机状态，必须先运行 `offline` 子命令将其脱机，然后再运行 `disable` 子命令。

要启动脱机设备组，可以执行以下任一操作：

- 发出显式 `online` 子命令或 `switch` 子命令。
- 访问该设备组中的某个设备。
- 挂载一个依赖于该设备组的文件系统。

如果指定 + 操作数，则只会影响将 `autogen` 属性设置为 `false` 的设备组。为了对在引导时系统自动创建的设备组（已将 `autogen` 属性设置为 `true`）应用命令，您必须明确指定每个设备组。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。

有关如何使设备组联机的信息，请参见 `online` 子命令的描述。

online

使设备组在预先指定的节点上联机。

如果设备组处于禁用状态，则必须使用以下方式之一将其启用，然后才能使它联机：

- 结合使用 `-e` 选项和 `online` 子命令。
- 运行 `enable` 子命令，然后再运行 `online` 子命令。

如果指定 + 操作数，则只会影响将 `autogen` 属性设置为 `false` 的设备组。为了对在引导时系统自动创建的设备组（已将 `autogen` 属性设置为 `true`）应用命令，您必须明确指定每个设备组。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。

有关如何使设备组脱机的信息，请参见 `offline` 子命令的描述。

remove-device

从原始磁盘设备组中删除成员磁盘设备。

`remove-device` 子命令只对 `rawdisk` 类型的设备组有效。该子命令对 `svm` 或 `sds` 设备组类型无效。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

有关如何向原始磁盘设备组中添加磁盘设备的信息，请参见 `add-device` 子命令的描述。

`remove-node`

从现有设备组中删除节点。

该子命令仅支持 `rawdisk` 设备组类型。无法使用 Oracle Solaris Cluster 命令从 `svm` 或 `sds` 设备组中删除节点。而应改用 Solaris Volume Manager 命令从 Solaris Volume Manager 磁盘集中删除节点。磁盘集会自动在 Oracle Solaris Cluster 软件中注册为 `svm` 或 `sds` 设备组。有关设备组类型的更多信息，请参见 `-t` 选项的描述。

如果设备组的 `preferenced` 属性设置为 `true`，则无法对其使用 `remove-node` 子命令。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

有关如何向设备组中添加节点的信息，请参见 `add-node` 子命令的描述。

`set`

修改与设备组关联的属性。

对于 `rawdisk` 类型的设备组，结合使用 `-d` 选项和 `set` 子命令可以为指定的设备组指定一个新的成员磁盘设备列表。

如果指定 `+` 操作数，则只会影响将 `autogen` 属性设置为 `false` 的设备组。为了对在引导时系统自动创建的设备组（已将 `autogen` 属性设置为 `true`）应用命令，您必须明确指定每个设备组。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

`show`

生成设备组配置报告。

默认情况下，该子命令会报告 `autogen` 属性设置为 `false` 的群集中的所有设备组。要显示群集中的所有设备组，还需要指定 `-v` 选项。

如果指定 `+` 操作数，则只会影响将 `autogen` 属性设置为 `false` 的设备组。为了对在引导时系统自动创建的设备组（已将 `autogen` 属性设置为 `true`）应用命令，您必须明确指定每个设备组。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

status

生成设备组状态报告。

默认情况下，该子命令会报告 autogen 属性设置为 false 的群集中的所有设备组。要显示群集中的所有设备组，还需要指定 -v 选项。

如果指定 + 操作数，则只会影响将 autogen 属性设置为 false 的设备组。为了对在引导时系统自动创建的设备组（已将 autogen 属性设置为 true）应用命令，您必须明确指定每个设备组。

超级用户以外的用户需要具有 solaris.cluster.read RBAC 授权才能使用该子命令。

switch

将设备组从 Oracle Solaris Cluster 配置中的一个主节点转移到另一个节点。

如果指定 + 操作数，则只会影响将 autogen 属性设置为 false 的设备组。为了对在引导时系统自动创建的设备组（已将 autogen 属性设置为 true）应用命令，您必须明确指定每个设备组。

超级用户以外的用户需要具有 solaris.cluster.modify RBAC 授权才能使用该子命令。

sync

使用群集软件同步设备组信息。

每当更改任何卷属性（例如所有者、组或访问权限）时，请使用此子命令。

还可以使用 sync 子命令将设备组配置更改为复制的或非复制的配置。

创建 Solaris Volume Manager 磁盘集之后，如果其中包含为复制而配置的磁盘，则必须为相应的 svm 或 sds 设备组运行 sync 子命令。Solaris Volume Manager 磁盘集会自动在 Oracle Solaris Cluster 软件中注册为 svm 或 sds 设备组，但在此期间不会同步复制信息。

对于新创建的 rawdisk 设备组类型，无需手动同步磁盘的复制信息。向 Oracle Solaris Cluster 软件注册原始磁盘设备组时，软件会自动搜索磁盘的任何复制信息。

如果指定 + 操作数，则只会影响将 autogen 属性设置为 false 的设备组。为了对在引导时系统自动创建的设备组（已将 autogen 属性设置为 true）应用命令，您必须明确指定每个设备组。

超级用户以外的用户需要具有 solaris.cluster.admin RBAC 授权才能使用该子命令。

支持以下选项：

-?

--help

显示帮助信息。

该选项可以单独使用，也可以和子命令结合使用。

- 如果单独使用该选项，则显示可用子命令的列表。
- 如果将该选项与子命令结合使用，将会显示该子命令的用法选项。

使用该选项时，不会执行任何其他处理。

`-d device[...]`
`--device=device[...]`
`--device device[...]`

指定要成为指定原始磁盘设备组的成员的磁盘设备的列表。

对于 `rawdisk` 类型的设备组，`-d` 选项仅在与 `create` 和 `set` 子命令一起使用才有效。必须始终提供整个节点列表。无法使用此选项向成员磁盘列表中添加、或从中删除单个磁盘。

只能以 DID 全局设备名称指定磁盘，例如 `d3`。有关更多信息，请参见 [did\(7\) \[1275\]](#) 手册页。

`-e`
`--enable`

启用设备组。此选项只有和 `online` 子命令结合使用时才有效。

如果指定的设备组已经启用，则 `-e` 选项将被忽略，命令会继续使该设备组联机。

`-i {- | clconfigfile}`
`--input={- | clconfigfile}`
`--input {- | clconfigfile}`

指定用于创建设备组的配置信息。此信息必须符合 [clconfiguration\(5CL\) \[1245\]](#) 手册页中定义的格式。此信息可包含在文件中或者通过标准输入提供。要指定标准输入，请提供减号 (-) 而不是文件名。

`-i` 选项只影响完全限定设备组列表中包含的设备组。

在命令中指定的选项将覆盖在配置文件中设置的任何选项。如果群集配置文件中缺少配置参数，则必须在命令行中指定这些参数。

`-n node[...]`
`--node=node[...]`
`--node node[...]`

指定一个节点或一系列节点。

默认情况下，节点列表的顺序表示节点尝试作为主节点接管设备组的优先顺序。例外情况是，对于纯本地磁盘组，由于不受 Oracle Solaris Cluster 的控制，因此主节点和辅助节点的概念不适用。

如果设备组的 `preferenced` 属性设置为 `false`，则将忽略节点列表的顺序。第一个访问该设备组中设备的节点会自动成为该设备组的主节点。有关设置设备组节点列表 `preferenced` 属性的更多信息，请参见 `-p` 选项。

无法使用 `-n` 选项指定 `svm` 或 `sds` 设备组的节点列表。必须使用 Solaris Volume Manager 命令或实用程序指定底层磁盘集节点列表。

create 和 set 子命令使用 -n 选项，以仅为 rawdisk 类型的设备组指定潜在主节点列表。必须指定设备组的整个节点列表。无法使用 -n 选项向节点列表中添加、或从中删除单个节点。

switch 子命令使用 -n 选项将单个节点指定为新的设备组主节点。

export、list、show 和 status 子命令使用 -n 选项将那些没有在指定节点上联机的设备组从输出中排除。

主节点和辅助节点的概念不适用于 localonly 磁盘组，这些磁盘组不受 Oracle Solaris Cluster 的控制。

-o {- | *clconfigfile*}
--output={- | *clconfigfile*}
--output {- | *clconfigfile*}

以 [clconfiguration\(5CL\) \[1245\]](#) 手册页所描述的格式显示设备组配置。可以将此信息写入文件或标准输出中。

如果将文件名作为参数提供给该选项，此命令将创建新的文件，配置会显示在该文件中。如果已经存在同名的文件，此命令将退出，并显示一个错误。不会对现有文件进行任何更改。

如果将减号 (-) 作为参数提供给该选项，命令会将配置信息显示到标准输出。命令的所有其他标准输出将受到抑制。

-o 选项仅与 export 子命令一起使用时才有效。

-p *name= value*
--property=*name =value*
--property *name=value*

设置设备组属性的值。

-p 选项仅在与 create 和 set 子命令一起使用时才有效。允许 -pname=*value* 的多个实例。

支持以下属性：

autogen

autogen 属性的值可以为 true 或 false。对于手动创建的设备组，默认值为 false。对于系统创建的设备组，默认值为 true。

autogen 属性是 list、show 和 status 子命令的指示符。这些子命令不会列出 autogen 属性设置为 true 的设备，除非使用 -v 选项。

该属性只对 rawdisk 类型的设备组有效。有关设备组类型的更多信息，请参见 -t 选项。

failback

failback 属性的值可以为 true 或 false。缺省值是 /。

failback 指定当设备组主节点脱离群集并在随后返回时系统的行为。

当设备组主节点脱离群集时，设备组会故障转移到辅助节点。当发生故障的节点重新加入群集时，设备组可继续由辅助节点控制，也可以故障恢复到原始主节点。

- 如果 `failback` 属性设置为 `true`，则设备组将变为由原始主节点控制。
- 如果 `failback` 属性设置为 `false`，则设备组继续由辅助节点控制。

默认情况下，`failback` 属性在创建设备组时处于禁用状态。`failback` 属性在 `set` 操作期间不会被更改。

`localonly`

`localonly` 属性的值可以为 `true` 或 `false`。缺省值是 `/`。

`localonly` 属性仅对 `rawdisk` 类型的磁盘组有效。

如果希望磁盘组仅由特定节点管理，则需要为磁盘组配置属性设置 `localonly=true`。纯本地磁盘组不受 Oracle Solaris Cluster 软件的控制。在纯本地磁盘组的节点列表中只能指定一个节点。如果将某个磁盘组的 `localonly` 属性设置为 `true`，则该磁盘组的节点列表必须只包含一个节点。

`numsecondaries`

`numsecondaries` 属性必须有一个大于 0 但小于节点列表中节点总数的整数值。默认值为 1。

此属性设置可用于动态更改所需的设备组辅助节点数。如果当前主节点发生故障，设备组的辅助节点可以作为主节点进行接管。

可以使用 `numsecondaries` 属性在保持指定可用性级别的情况下更改设备组的辅助节点数。如果从设备组的辅助节点列表中删除了某个节点，则该节点将无法再作为主节点进行接管。

`numsecondaries` 属性仅适用于设备组中当前处于群集模式的节点。节点还必须能够与设备组的 `preferenced` 属性一起使用。如果设备组的 `preferenced` 属性设置为 `true`，则首先会从辅助节点列表中删除优先级最低的节点。如果设备组中没有任何节点标记为首选的，群集会随机选择节点进行删除。

当设备组的实际辅助节点数低于所需的级别时，会将从辅助节点列表中删除的符合条件的节点逐个添加回列表。每个节点必须满足以下所有条件，才有资格添加回辅助节点列表：

- 节点当前位于群集中。
- 节点属于设备组。
- 节点当前不是主节点或辅助节点。

转换从设备组中优先级最高的节点开始。多个节点将按照优先顺序进行转换，直到符合所需的辅助节点数。

如果某个节点加入了群集，并且比设备组中现有辅助节点的优先级要高，则优先级较低的节点将从辅助节点列表中删除。新添加的节点会替代删除的节点。只有当群集中存在的实际辅助节点多于所需级别时，才会发生此类替代。

有关设置设备组节点列表 `preferenced` 属性的更多信息，请参见 `preferenced` 属性。

`preferenced`

`preferenced` 属性的值可以为 `true` 或 `false`。缺省值为 `true`。

在创建设备组的过程中，如果 `preferenced` 属性设置为 `true`，则节点列表的顺序也表示节点的优先顺序。节点的优先顺序决定每个节点尝试作为主节点接管设备组的顺序。

在创建设备组的过程中，如果该属性设置为 `false`，则第一个访问设备组中设备的节点会自动成为主节点。指定的节点列表中的节点顺序没有意义。将此属性设置回 `true` 而不重新指定节点列表，不会重新激活节点顺序。

在 `set` 操作期间，节点的优先顺序不会更改，除非同时指定 `preferenced=true` 属性，并使用 `-n` 选项为设备组提供具有优先顺序的整个节点列表。

```
-t devicegroup-type[...]  
--type=devicegroup-type[...]  
--type devicegroup-type[...]
```

指定一个设备组类型或设备组类型列表。

对于 `create` 子命令，只能指定一个设备组类型。随后会创建使用此选项指定的类型的设备组。

对于所有接受 `-t` 选项的其他子命令，此选项将您提供给命令的设备组列表限定为仅包含指定类型的设备组。

并非所有子命令和选项对所有设备组类型都有效。例如，`create` 子命令仅对 `rawdisk` 设备组类型有效，而对 `svm` 或 `sds` 设备组类型无效。

`-t` 选项支持以下设备组类型：

`rawdisk`

指定一个原始磁盘设备组。

原始磁盘是不属于卷管理器卷或元设备的一部分的磁盘。原始磁盘设备组支持在设备组中定义一组磁盘。默认情况下，系统在引导时，会为配置中的每一个设备 ID 伪驱动程序 (DID) 设备创建一个原始磁盘设备组。根据约定，原始磁盘设备组的名称是在初始化时分配的。这些名称是由 DID 设备名称派生而来的。对于每一个添加到原始磁盘设备组的节点，`cldevicegroup` 命令可以验证设备组中的每个设备是否已通过端口物理连接到该节点。

`create` 子命令创建原始磁盘设备组，并向该设备组添加多个磁盘设备。创建新的原始磁盘设备组之前，必须从引导时为设备创建的设备组中删除要添加到新设备组的各个设备。然后可以创建一个新的包含这些设备的原始磁盘设备组。可以使用 `-d` 选项指定这些设备的列表，同时使用 `-n` 选项指定潜在主节点优先列表。

要在单个指定的节点上管理一个设备组，请使用 `-p` 选项为该设备组配置属性设置 `localonly=true`。创建纯本地设备组时，只能在节点列表中指定一个节点。

`delete` 子命令从群集设备组配置中删除设备组名称。

`set` 子命令对原始磁盘设备组进行以下更改：

- 更改潜在主节点的优先顺序
- 指定一个新的节点列表
- 启用或禁用故障恢复
- 设置所需的辅助节点数
- 向设备组中添加更多全局设备

如果某个原始磁盘设备名称已在原始磁盘设备组中注册，则无法同时在 Solaris Volume Manager 设备组中注册该原始磁盘设备名称。

sds

指定最初使用 Solstice DiskSuite™ 软件创建的设备组。除多属主磁盘集以外，该设备组类型等同于 Solaris Volume Manager 设备组类型 `svm`。有关更多信息，请参见 `svm` 设备组类型的描述。

svm

指定 Solaris Volume Manager 设备组。

Solaris Volume Manager 设备组是由以下几部分定义的：

- 名称
- 节点，可以通过它访问设备组
- 磁盘集中设备的全局列表
- 一组属性，控制潜在主节点优先选择和故障恢复行为等操作

Solaris Volume Manager 有多宿主磁盘集或共享磁盘集概念。共享磁盘集由两个或多个主机和磁盘驱动器组成。这些磁盘驱动器可供所有主机访问，并且在所有主机中具有相同的设备名称。该相同设备命名的要求是通过使用原始磁盘设备构成磁盘集来实现的。设备 ID 伪驱动程序 (DID) 允许多宿主磁盘在群集中拥有一致的名称。只有已经配置为磁盘集一部分的主机才能配置到 Solaris Volume Manager 设备组节点列表中。向共享磁盘集添加驱动器时，驱动器不得属于任何其他共享磁盘集。

Solaris Volume Manager `metaset` 命令创建磁盘集，并自动向 Oracle Solaris Cluster 软件将该磁盘集注册为 Solaris Volume Manager 设备组。创建设备组之后，必须使用 `cldevicegroup` 命令的 `set` 子命令设置节点优先列表以及 `preferenced`、`failback` 和 `numsecondaries` 属性。

只能为每个设备组分配一个 Solaris Volume Manager 磁盘集。设备组的名称必须始终与磁盘集的名称相匹配。

无法使用 `add-node` 或 `remove-node` 子命令在 Solaris Volume Manager 设备组中添加或删除节点。可以使用 Solaris Volume Manager `metaset` 命令在底层 Solaris Volume Manager 磁盘集中添加或删除节点。

无法使用 `delete` 子命令从群集配置中删除 Solaris Volume Manager 设备组。而应改用 Solaris Volume Manager `metaset` 命令删除底层 Solaris Volume Manager 磁盘集。

只有 `export`、`list`、`show`、`status` 和 `sync` 子命令可用于 Solaris Volume Manager 多所有者磁盘集。必须使用 Solaris Volume Manager 命令或实用程序才能创建和删除 Solaris Volume Manager 设备组的底层磁盘集。

`-v`
`--version`

显示该命令的版本。

指定此选项时，不要将其与子命令、操作数或其他选项结合使用。子命令、操作数或其他选项将被忽略。`-v` 选项只显示命令的版本。不会执行任何其他操作。

`-v`
`--verbose`

将详细消息显示到标准输出。

可以将此选项与命令的任何格式一起使用。

支持以下操作数：

devicegroup 指定一个设备组。
`cldevicegroup` 命令只接受 Oracle Solaris Cluster 设备组名称作为操作数。对于大多数接受多个设备组名称的命令格式，可以使用加号 (+) 指定所有可能的设备组。

注 - + 操作数只包括手动创建的设备组，忽略所有自动创建的、`autogen` 属性设置为 `true` 的设备组。Oracle Solaris Cluster 软件在每次系统引导时会自动创建此类设备组。要对这些“隐藏”设备组应用命令，必须明确指定每个设备组。

[Intro\(1CL\) \[17\]](#) 手册页列出了该命令集中所有命令的一整套退出状态代码。

如果命令对于所有指定的操作数均成功执行，则返回零 (`CL_NOERR`)。如果某个操作数出错，命令会继续处理操作数列表中的下一个操作数。返回的退出代码始终反映最先发生的错误。

此命令返回下列退出状态代码：

0 `CL_NOERR`
没有错误

1 `CL_ENOMEM`
交换空间不足

3 CL_EINVAL
参数无效

6 CL_EACCESS
权限被拒绝

35 CL_EIO
I/O 错误

36 CL_ENOENT
没有这样的对象

39 CL_EEXIST
对象已存在

例 43 修改设备组

以下示例显示了如何将设备组 devgrp1 的 preference 属性设置为 true，并将 numsecondaries 属性设置为 2。该命令还指定所需的节点列表 phys-schost-1,phys-schost-2,phys-schost-3。

```
# cldevicegroup set -p preferenced=true -p numsecondaries=2 \  
-n phys-schost-1,phys-schost-2,phys-schost-3 devgrp1
```

例 44 修改原始磁盘设备组

以下示例显示了如何修改现有的原始磁盘设备组 rawdevgrp1。该命令在新成员设备列表中指定设备 d3 和 d4，并将 localonly 属性设置为 true。节点 phys-schost-1 是允许用于纯本地原始磁盘设备组的唯一主节点。

```
# cldevicegroup set -d d3,d4 \  
-p localonly=true -n phys-schost-1 rawdevgrp1
```

例 45 重置设备组的 numsecondaries 属性

以下示例显示了如何通过不指定任何值给设备组 devgrp1 的 numsecondaries 属性，将其重置为相应的系统默认值。

```
# cldevicegroup set -p numsecondaries= devgrp1
```

例 46 切换设备组

以下示例说明如何将设备组 devgrp1 切换到新的主节点 phys-schost-2。

```
# cldevicegroup switch -n phys-schost-2 devgrp1
```

例 47 禁用设备组

以下示例显示了如何禁用设备组 devgrp1 。

```
# cldevicegroup disable devgrp1
```

例 48 使设备组脱机

以下示例说明如何先将设备组 devgrp1 脱机再将其禁用。

```
# cldevicegroup offline devgrp1
# cldevicegroup disable devgrp1
```

例 49 使设备组在其主节点上联机

以下示例说明如何使设备组 devgrp1 在其默认主节点上联机。该命令首先启用设备组。

```
# cldevicegroup online -e devgrp1
```

例 50 使设备组在指定节点上联机

以下示例说明如何使设备组 devgrp1 在作为其新主节点的 phys-schost-2 上联机。

```
# cldevicegroup switch -n phys-schost-2 devgrp1
```

例 51 向设备组添加新节点

以下示例显示了如何向设备组 devgrp1 添加一个新节点 phys-schost--3。此设备组不属于设备组类型 svm。

```
# cldevicegroup add-node -n phys-schost-3 devgrp1
```

例 52 删除设备组

以下示例说明如何从 Oracle Solaris Cluster 配置中删除设备组 devgrp1。此设备组不属于设备组类型 svm。

```
# cldevicegroup delete devgrp1
```

例 53 将复制信息与设备组配置进行同步

以下示例显示了如何让 Oracle Solaris Cluster 软件知道设备组 devgrp1 中的磁盘所使用的复制配置。

```
# cldevicegroup sync devgrp1
```

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

[Intro\(1CL\) \[17\]](#)、[cldevice\(1CL\) \[51\]](#)、[cluster\(1CL\) \[461\]](#)、[Unresolved link to "metaset1M"](#)、[clconfiguration\(5CL\) \[1245\]](#)、[Unresolved link to "rbac5"](#)、[did\(7\) \[1275\]](#)

[Unresolved link to "Oracle Solaris Cluster 系统管理指南"](#)

超级用户可以运行此命令的所有格式。

此外，任何用户都可以运行带有以下选项的此命令：

- -? (帮助) 选项
- -V (版本) 选项

要运行带有其他子命令的此命令，超级用户以外的用户需要具有 RBAC 授权。请参见下表。

子命令	RBAC 授权
add-device	solaris.cluster.modify
add-node	solaris.cluster.modify
create	solaris.cluster.modify
delete	solaris.cluster.modify
disable	solaris.cluster.modify
enable	solaris.cluster.modify
export	solaris.cluster.read
list	solaris.cluster.read
offline	solaris.cluster.admin
online	solaris.cluster.admin
remove-device	solaris.cluster.modify
remove-node	solaris.cluster.modify
set	solaris.cluster.modify
show	solaris.cluster.read
status	solaris.cluster.read
switch	solaris.cluster.modify
sync	solaris.cluster.admin

名称

cldevicegroup, cldg — 管理 Oracle Solaris Cluster 设备组

```
/usr/cluster/bin/cldevicegroup -V

/usr/cluster/bin/cldevicegroup [subcommand] -?

/usr/cluster/bin/cldevicegroup subcommand [options] -v
    [devicegroup ...]

/usr/cluster/bin/cldevicegroup add-device -d device
    [,...] devicegroup

/usr/cluster/bin/cldevicegroup add-node -n node[,...] [-t
    devicegroup-type[,...]] {+ | devicegroup ...}

/usr/cluster/bin/cldevicegroup create -n node[,...] -t
    devicegroup-type [-d device[,...]] [-p name=value]
    devicegroup ...

/usr/cluster/bin/cldevicegroup create -i {- | clconfigfile} [-d
    device[,...]] [-n node[,...]] [-p name=value] [-t
    devicegroup-type[,...]] {+ | devicegroup ...}

/usr/cluster/bin/cldevicegroup delete [-t devicegroup-type[,...]]
    {+ | devicegroup ...}

/usr/cluster/bin/cldevicegroup disable [-t devicegroup-type[,...]]
    {+ | devicegroup ...}

/usr/cluster/bin/cldevicegroup enable [-t devicegroup-type[,...]]
    {+ | devicegroup ...}

/usr/cluster/bin/cldevicegroup export [-n node[,...]] [-o
    {- | clconfigfile}] [-t devicegroup-type[,...]]
    {+ | devicegroup ...}

/usr/cluster/bin/cldevicegroup list [-n node[,...]] [-t
    devicegroup-type[,...]] {+ | devicegroup ...}

/usr/cluster/bin/cldevicegroup offline [-t devicegroup-type[,...]]
    {+ | devicegroup ...}

/usr/cluster/bin/cldevicegroup online [-e] [-n node] [-t
    devicegroup-type[,...]] {+ | devicegroup ...}

/usr/cluster/bin/cldevicegroup remove-device -d device
    [,...] devicegroup

/usr/cluster/bin/cldevicegroup remove-node -n node[,...]
    [-t devicegroup-type[,...]] {+ | devicegroup ...}

/usr/cluster/bin/cldevicegroup set -p name=value [-p name=value]...
```

```
[-d device[,...]] [-n node[,...]] [-t devicegroup-type[,...]]
{+ | devicegroup ...}

/usr/cluster/bin/cldevicegroup show [-n node[,...]] [-t
devicegroup-type[,...]] [+ | devicegroup ...]

/usr/cluster/bin/cldevicegroup status [-n node[,...]] [-t
devicegroup-type[,...]] [+ | devicegroup ...]

/usr/cluster/bin/cldevicegroup switch -n node [-t
devicegroup-type[,...]] [+ | devicegroup ...]

/usr/cluster/bin/cldevicegroup sync [-t devicegroup-type[,...]]
{+ | devicegroup ...}
```

cldevicegroup 命令管理 Oracle Solaris Cluster 设备组。cldg 命令是 cldevicegroup 命令的简短格式。这两个命令是相同的。您可以使用此命令的任一种格式。

此命令的一般格式如下所示：

```
cldevicegroup [subcommand] [options] [operands]
```

仅当 *options* 指定 -? 选项或 -v 选项时，才可以省略 *subcommand*。

此命令的每个选项都有长格式和简短格式两种格式。在本手册页的“选项”部分给出了每个选项的两种格式，以及对该选项的描述。

除了 list、show 和 status 以外，大多数子命令都需要至少一个操作数。许多子命令接受使用加号 (+) 作为操作数来表示 all 适用对象。有关详细信息，请参阅该手册页的“用法概要”和其他部分。

除以下子命令之外，每个子命令都可以用于所有设备组类型：

- add-device 和 remove-device 子命令只对 rawdisk 类型有效。
- add-node、create、delete 和 remove-node 子命令仅对 rawdisk 类型有效。

仅可以在全局区域中使用此命令。

子命令

支持以下子命令：

add-device

向现有原始磁盘设备组中添加新的成员磁盘设备。

对于现有的 rawdisk 类型的设备组，您只能使用 add-device 子命令。有关设备组类型的更多信息，请参见 -t 选项的描述。

超级用户以外的用户需要具有 solaris.cluster.modify RBAC 授权才能使用该子命令。

有关如何从原始磁盘设备组中删除磁盘设备的信息，请参见 `remove-device` 子命令的描述。

`add-node`

向现有设备组中添加新节点。

该子命令仅支持 `rawdisk` 设备组类型。无法使用 Oracle Solaris Cluster 命令向 `svm` 或 `sds` 设备组添加节点。而应改用 Solaris Volume Manager 命令向 Solaris Volume Manager 磁盘集添加节点。磁盘集会自动在 Oracle Solaris Cluster 软件中注册为 `svm` 或 `sds` 设备组。有关设备组类型的更多信息，请参见 `-t` 选项的描述。

如果某个设备组的 `preferenced` 属性设置为 `true`，则无法对该设备组使用该子命令。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

有关如何从设备组中删除节点的信息，请参见 `remove-node` 子命令的描述。

`create`

创建新设备组。

该子命令仅支持 `rawdisk` 设备组类型。无法使用 Oracle Solaris Cluster 命令创建 `svm` 或 `sds` 设备组。而应改用 Solaris Volume Manager 命令创建 Solaris Volume Manager 磁盘集。磁盘集会自动在 Oracle Solaris Cluster 软件中注册为 `svm` 或 `sds` 设备组。有关设备组类型的更多信息，请参见 `-t` 选项的描述。

如果使用 `-i` 选项指定配置文件，则可以提供加号 (+) 作为操作数。使用此操作数时，该命令会创建配置文件中指定的、且尚不存在的所有设备组。

对于 `rawdisk` 类型的设备组，结合使用 `-d` 选项和 `create` 子命令可以为此类设备组指定一个或多个设备。指定设备时，每个命令调用使用一个 `-d` 选项。无法使用一个命令调用创建多个原始磁盘设备组，除非使用 `-i` 选项。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

有关如何删除设备组的信息，请参见 `delete` 子命令的描述。

`delete`

删除设备组。

该子命令仅支持 `rawdisk` 设备组类型。

无法使用 Oracle Solaris Cluster 命令删除 `svm` 或 `sds` 设备组。要删除 `svm` 或 `sds` 设备组，请使用 Solaris Volume Manager 命令删除底层 Solaris Volume Manager 磁盘集。

删除设备组之前，设备组必须处于脱机状态。

如果指定 + 操作数，则只会影响将 `autogen` 属性设置为 `false` 的设备组。为了对在引导时系统自动创建的设备组（已将 `autogen` 属性设置为 `true`）应用命令，您必须明确指定每个设备组。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

有关如何创建设备组的信息，请参见 `create` 子命令的描述。

`disable`

禁用脱机设备组。

重新引导后，设备组的禁用状态不变。

要禁用联机设备组，必须首先使用 `offline` 子命令将设备组脱机。

如果某个设备组当前处于联机状态，则 `disable` 操作将失败，且不会禁用指定的设备组。

无法使用 `switch` 或 `online` 子命令使禁用的设备组联机。必须首先使用 `enable` 子命令清除该设备组的禁用状态。

如果指定 + 操作数，则只会影响将 `autogen` 属性设置为 `false` 的设备组。为了对在引导时系统自动创建的设备组（已将 `autogen` 属性设置为 `true`）应用命令，您必须明确指定每个设备组。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

有关如何启用设备组的信息，请参见 `enable` 子命令的描述。

`enable`

启用设备组。

重新引导后，设备组的禁用状态不变。

要使禁用的设备组联机，必须首先使用 `enable` 子命令清除设备组的禁用状态。

如果指定 + 操作数，则只会影响将 `autogen` 属性设置为 `false` 的设备组。为了对在引导时系统自动创建的设备组（已将 `autogen` 属性设置为 `true`）应用命令，您必须明确指定每个设备组。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

有关如何禁用设备组的信息，请参见 `disable` 子命令的描述。

`export`

导出设备组的配置信息。

如果使用 `-o` 选项指定一个文件名，则会将配置信息写入该新文件。如果未提供 `-o` 选项，则会将输出写入标准输出。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

`list`

显示设备组列表。

默认情况下，该子命令会列出 `autogen` 属性设置为 `false` 的群集中的所有设备组。要显示群集中的所有设备组，还需要指定 `-v` 选项。

如果指定 `+` 操作数，则只会影响将 `autogen` 属性设置为 `false` 的设备组。为了对在引导时系统自动创建的设备组（已将 `autogen` 属性设置为 `true`）应用命令，您必须明确指定每个设备组。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

offline

使设备组脱机。

如果设备组处于联机状态，必须先运行 `offline` 子命令将其脱机，然后再运行 `disable` 子命令。

要启动脱机设备组，可以执行以下任一操作：

- 发出显式 `online` 子命令或 `switch` 子命令。
- 访问该设备组中的某个设备。
- 挂载一个依赖于该设备组的文件系统。

如果指定 `+` 操作数，则只会影响将 `autogen` 属性设置为 `false` 的设备组。为了对在引导时系统自动创建的设备组（已将 `autogen` 属性设置为 `true`）应用命令，您必须明确指定每个设备组。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。

有关如何使设备组联机的信息，请参见 `online` 子命令的描述。

online

使设备组在预先指定的节点上联机。

如果设备组处于禁用状态，则必须使用以下方式之一将其启用，然后才能使它联机：

- 结合使用 `-e` 选项和 `online` 子命令。
- 运行 `enable` 子命令，然后再运行 `online` 子命令。

如果指定 `+` 操作数，则只会影响将 `autogen` 属性设置为 `false` 的设备组。为了对在引导时系统自动创建的设备组（已将 `autogen` 属性设置为 `true`）应用命令，您必须明确指定每个设备组。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。

有关如何使设备组脱机的信息，请参见 `offline` 子命令的描述。

remove-device

从原始磁盘设备组中删除成员磁盘设备。

`remove-device` 子命令只对 `rawdisk` 类型的设备组有效。该子命令对 `svm` 或 `sds` 设备组类型无效。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

有关如何向原始磁盘设备组中添加磁盘设备的信息，请参见 `add-device` 子命令的描述。

`remove-node`

从现有设备组中删除节点。

该子命令仅支持 `rawdisk` 设备组类型。无法使用 Oracle Solaris Cluster 命令从 `svm` 或 `sds` 设备组中删除节点。而应改用 Solaris Volume Manager 命令从 Solaris Volume Manager 磁盘集中删除节点。磁盘集会自动在 Oracle Solaris Cluster 软件中注册为 `svm` 或 `sds` 设备组。有关设备组类型的更多信息，请参见 `-t` 选项的描述。

如果设备组的 `preferenced` 属性设置为 `true`，则无法对其使用 `remove-node` 子命令。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

有关如何向设备组中添加节点的信息，请参见 `add-node` 子命令的描述。

`set`

修改与设备组关联的属性。

对于 `rawdisk` 类型的设备组，结合使用 `-d` 选项和 `set` 子命令可以为指定的设备组指定一个新的成员磁盘设备列表。

如果指定 `+` 操作数，则只会影响将 `autogen` 属性设置为 `false` 的设备组。为了对在引导时系统自动创建的设备组（已将 `autogen` 属性设置为 `true`）应用命令，您必须明确指定每个设备组。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

`show`

生成设备组配置报告。

默认情况下，该子命令会报告 `autogen` 属性设置为 `false` 的群集中的所有设备组。要显示群集中的所有设备组，还需要指定 `-v` 选项。

如果指定 `+` 操作数，则只会影响将 `autogen` 属性设置为 `false` 的设备组。为了对在引导时系统自动创建的设备组（已将 `autogen` 属性设置为 `true`）应用命令，您必须明确指定每个设备组。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

status

生成设备组状态报告。

默认情况下，该子命令会报告 autogen 属性设置为 false 的群集中的所有设备组。要显示群集中的所有设备组，还需要指定 -v 选项。

如果指定 + 操作数，则只会影响将 autogen 属性设置为 false 的设备组。为了对在引导时系统自动创建的设备组（已将 autogen 属性设置为 true）应用命令，您必须明确指定每个设备组。

超级用户以外的用户需要具有 solaris.cluster.read RBAC 授权才能使用该子命令。

switch

将设备组从 Oracle Solaris Cluster 配置中的一个主节点转移到另一个节点。

如果指定 + 操作数，则只会影响将 autogen 属性设置为 false 的设备组。为了对在引导时系统自动创建的设备组（已将 autogen 属性设置为 true）应用命令，您必须明确指定每个设备组。

超级用户以外的用户需要具有 solaris.cluster.modify RBAC 授权才能使用该子命令。

sync

使用群集软件同步设备组信息。

每当更改任何卷属性（例如所有者、组或访问权限）时，请使用此子命令。

还可以使用 sync 子命令将设备组配置更改为复制的或非复制的配置。

创建 Solaris Volume Manager 磁盘集之后，如果其中包含为复制而配置的磁盘，则必须为相应的 svm 或 sds 设备组运行 sync 子命令。Solaris Volume Manager 磁盘集会自动在 Oracle Solaris Cluster 软件中注册为 svm 或 sds 设备组，但在此期间不会同步复制信息。

对于新创建的 rawdisk 设备组类型，无需手动同步磁盘的复制信息。向 Oracle Solaris Cluster 软件注册原始磁盘设备组时，软件会自动搜索磁盘的任何复制信息。

如果指定 + 操作数，则只会影响将 autogen 属性设置为 false 的设备组。为了对在引导时系统自动创建的设备组（已将 autogen 属性设置为 true）应用命令，您必须明确指定每个设备组。

超级用户以外的用户需要具有 solaris.cluster.admin RBAC 授权才能使用该子命令。

支持以下选项：

-?

--help

显示帮助信息。

该选项可以单独使用，也可以和子命令结合使用。

- 如果单独使用该选项，则显示可用子命令的列表。
- 如果将该选项与子命令结合使用，将会显示该子命令的用法选项。

使用该选项时，不会执行任何其他处理。

`-d device[...]`
`--device=device[...]`
`--device device[...]`

指定要成为指定原始磁盘设备组的成员的磁盘设备的列表。

对于 `rawdisk` 类型的设备组，`-d` 选项仅在与 `create` 和 `set` 子命令一起使用才有效。必须始终提供整个节点列表。无法使用此选项向成员磁盘列表中添加、或从中删除单个磁盘。

只能以 DID 全局设备名称指定磁盘，例如 `d3`。有关更多信息，请参见 [did\(7\) \[1275\]](#) 手册页。

`-e`
`--enable`

启用设备组。此选项只有和 `online` 子命令结合使用时才有效。

如果指定的设备组已经启用，则 `-e` 选项将被忽略，命令会继续使该设备组联机。

`-i {- | clconfigfile}`
`--input={- | clconfigfile}`
`--input {- | clconfigfile}`

指定用于创建设备组的配置信息。此信息必须符合 [clconfiguration\(5CL\) \[1245\]](#) 手册页中定义的格式。此信息可包含在文件中或者通过标准输入提供。要指定标准输入，请提供减号 (-) 而不是文件名。

`-i` 选项只影响完全限定设备组列表中包含的设备组。

在命令中指定的选项将覆盖在配置文件中设置的任何选项。如果群集配置文件中缺少配置参数，则必须在命令行中指定这些参数。

`-n node[...]`
`--node=node[...]`
`--node node[...]`

指定一个节点或一系列节点。

默认情况下，节点列表的顺序表示节点尝试作为主节点接管设备组的优先顺序。例外情况是，对于纯本地磁盘组，由于不受 Oracle Solaris Cluster 的控制，因此主节点和辅助节点的概念不适用。

如果设备组的 `preferenced` 属性设置为 `false`，则将忽略节点列表的顺序。第一个访问该设备组中设备的节点会自动成为该设备组的主节点。有关设置设备组节点列表 `preferenced` 属性的更多信息，请参见 `-p` 选项。

无法使用 `-n` 选项指定 `svm` 或 `sds` 设备组的节点列表。必须使用 Solaris Volume Manager 命令或实用程序指定底层磁盘集节点列表。

create 和 set 子命令使用 -n 选项，以仅为 rawdisk 类型的设备组指定潜在主节点列表。必须指定设备组的整个节点列表。无法使用 -n 选项向节点列表中添加、或从中删除单个节点。

switch 子命令使用 -n 选项将单个节点指定为新的设备组主节点。

export、list、show 和 status 子命令使用 -n 选项将那些没有在指定节点上联机的设备组从输出中排除。

主节点和辅助节点的概念不适用于 localonly 磁盘组，这些磁盘组不受 Oracle Solaris Cluster 的控制。

-o {- | *clconfigfile*}
--output={- | *clconfigfile*}
--output {- | *clconfigfile*}

以 [clconfiguration\(5CL\) \[1245\]](#) 手册页所描述的格式显示设备组配置。可以将此信息写入文件或标准输出中。

如果将文件名作为参数提供给该选项，此命令将创建新的文件，配置会显示在该文件中。如果已经存在同名的文件，此命令将退出，并显示一个错误。不会对现有文件进行任何更改。

如果将减号 (-) 作为参数提供给该选项，命令会将配置信息显示到标准输出。命令的所有其他标准输出将受到抑制。

-o 选项仅与 export 子命令一起使用时才有效。

-p *name= value*
--property=*name =value*
--property *name=value*

设置设备组属性的值。

-p 选项仅在与 create 和 set 子命令一起使用时才有效。允许 -pname=*value* 的多个实例。

支持以下属性：

autogen

autogen 属性的值可以为 true 或 false。对于手动创建的设备组，默认值为 false。对于系统创建的设备组，默认值为 true。

autogen 属性是 list、show 和 status 子命令的指示符。这些子命令不会列出 autogen 属性设置为 true 的设备，除非使用 -v 选项。

该属性只对 rawdisk 类型的设备组有效。有关设备组类型的更多信息，请参见 -t 选项。

failback

failback 属性的值可以为 true 或 false。缺省值是 /。

failback 指定当设备组主节点脱离群集并在随后返回时系统的行为。

当设备组主节点脱离群集时，设备组会故障转移到辅助节点。当发生故障的节点重新加入群集时，设备组可继续由辅助节点控制，也可以故障恢复到原始主节点。

- 如果 `failback` 属性设置为 `true`，则设备组将变为由原始主节点控制。
- 如果 `failback` 属性设置为 `false`，则设备组继续由辅助节点控制。

默认情况下，`failback` 属性在创建设备组时处于禁用状态。`failback` 属性在 `set` 操作期间不会被更改。

`localonly`

`localonly` 属性的值可以为 `true` 或 `false`。缺省值是 `/`。

`localonly` 属性仅对 `rawdisk` 类型的磁盘组有效。

如果希望磁盘组仅由特定节点管理，则需要为磁盘组配置属性设置 `localonly=true`。纯本地磁盘组不受 Oracle Solaris Cluster 软件的控制。在纯本地磁盘组的节点列表中只能指定一个节点。如果将某个磁盘组的 `localonly` 属性设置为 `true`，则该磁盘组的节点列表必须只包含一个节点。

`numsecondaries`

`numsecondaries` 属性必须有一个大于 0 但小于节点列表中节点总数的整数值。默认值为 1。

此属性设置可用于动态更改所需的设备组辅助节点数。如果当前主节点发生故障，设备组的辅助节点可以作为主节点进行接管。

可以使用 `numsecondaries` 属性在保持指定可用性级别的情况下更改设备组的辅助节点数。如果从设备组的辅助节点列表中删除了某个节点，则该节点将无法再作为主节点进行接管。

`numsecondaries` 属性仅适用于设备组中当前处于群集模式的节点。节点还必须能够与设备组的 `preferenced` 属性一起使用。如果设备组的 `preferenced` 属性设置为 `true`，则首先会从辅助节点列表中删除优先级最低的节点。如果设备组中没有任何节点标记为首选的，群集会随机选择节点进行删除。

当设备组的实际辅助节点数低于所需的级别时，会将从辅助节点列表中删除的符合条件的节点逐个添加回列表。每个节点必须满足以下所有条件，才有资格添加回辅助节点列表：

- 节点当前位于群集中。
- 节点属于设备组。
- 节点当前不是主节点或辅助节点。

转换从设备组中优先级最高的节点开始。多个节点将按照优先顺序进行转换，直到符合所需的辅助节点数。

如果某个节点加入了群集，并且比设备组中现有辅助节点的优先级要高，则优先级较低的节点将从辅助节点列表中删除。新添加的节点会替代删除的节点。只有当群集中存在的实际辅助节点多于所需级别时，才会发生此类替代。

有关设置设备组节点列表 `preferenced` 属性的更多信息，请参见 `preferenced` 属性。

`preferenced`

`preferenced` 属性的值可以为 `true` 或 `false`。缺省值为 `true`。

在创建设备组的过程中，如果 `preferenced` 属性设置为 `true`，则节点列表的顺序也表示节点的优先顺序。节点的优先顺序决定每个节点尝试作为主节点接管设备组的顺序。

在创建设备组的过程中，如果该属性设置为 `false`，则第一个访问设备组中设备的节点会自动成为主节点。指定的节点列表中的节点顺序没有意义。将此属性设置回 `true` 而不重新指定节点列表，不会重新激活节点顺序。

在 `set` 操作期间，节点的优先顺序不会更改，除非同时指定 `preferenced=true` 属性，并使用 `-n` 选项为设备组提供具有优先顺序的整个节点列表。

```
-t devicegroup-type[...]  
--type=devicegroup-type[...]  
--type devicegroup-type[...]
```

指定一个设备组类型或设备组类型列表。

对于 `create` 子命令，只能指定一个设备组类型。随后会创建使用此选项指定的类型的设备组。

对于所有接受 `-t` 选项的其他子命令，此选项将您提供给命令的设备组列表限定为仅包含指定类型的设备组。

并非所有子命令和选项对所有设备组类型都有效。例如，`create` 子命令仅对 `rawdisk` 设备组类型有效，而对 `svm` 或 `sds` 设备组类型无效。

`-t` 选项支持以下设备组类型：

`rawdisk`

指定一个原始磁盘设备组。

原始磁盘是不属于卷管理器卷或元设备的一部分的磁盘。原始磁盘设备组支持在设备组中定义一组磁盘。默认情况下，系统在引导时，会为配置中的每一个设备 ID 伪驱动程序 (DID) 设备创建一个原始磁盘设备组。根据约定，原始磁盘设备组的名称是在初始化时分配的。这些名称是由 DID 设备名称派生而来的。对于每一个添加到原始磁盘设备组的节点，`cldevicegroup` 命令可以验证设备组中的每个设备是否已通过端口物理连接到该节点。

`create` 子命令创建原始磁盘设备组，并向该设备组添加多个磁盘设备。创建新的原始磁盘设备组之前，必须从引导时为设备创建的设备组中删除要添加到新设备组的各个设备。然后可以创建一个新的包含这些设备的原始磁盘设备组。可以使用 `-d` 选项指定这些设备的列表，同时使用 `-n` 选项指定潜在主节点优先列表。

要在单个指定的节点上管理一个设备组，请使用 `-p` 选项为该设备组配置属性设置 `localonly=true`。创建纯本地设备组时，只能在节点列表中指定一个节点。

`delete` 子命令从群集设备组配置中删除设备组名称。

`set` 子命令对原始磁盘设备组进行以下更改：

- 更改潜在主节点的优先顺序
- 指定一个新的节点列表
- 启用或禁用故障恢复
- 设置所需的辅助节点数
- 向设备组中添加更多全局设备

如果某个原始磁盘设备名称已在原始磁盘设备组中注册，则无法同时在 Solaris Volume Manager 设备组中注册该原始磁盘设备名称。

`sds`

指定最初使用 Solstice DiskSuite™ 软件创建的设备组。除多属主磁盘集以外，该设备组类型等同于 Solaris Volume Manager 设备组类型 `svm`。有关更多信息，请参见 `svm` 设备组类型的描述。

`svm`

指定 Solaris Volume Manager 设备组。

Solaris Volume Manager 设备组是由以下几部分定义的：

- 名称
- 节点，可以通过它访问设备组
- 磁盘集中设备的全局列表
- 一组属性，控制潜在主节点优先选择和故障恢复行为等操作

Solaris Volume Manager 有多宿主磁盘集或共享磁盘集概念。共享磁盘集由两个或多个主机和磁盘驱动器组成。这些磁盘驱动器可供所有主机访问，并且在所有主机中具有相同的设备名称。该相同设备命名的要求是通过使用原始磁盘设备构成磁盘集来实现的。设备 ID 伪驱动程序 (DID) 允许多宿主磁盘在群集中拥有一致的名称。只有已经配置为磁盘集一部分的主机才能配置到 Solaris Volume Manager 设备组节点列表中。向共享磁盘集添加驱动器时，驱动器不得属于任何其他共享磁盘集。

Solaris Volume Manager `metaset` 命令创建磁盘集，并自动向 Oracle Solaris Cluster 软件将该磁盘集注册为 Solaris Volume Manager 设备组。创建设备组之后，必须使用 `cldevicegroup` 命令的 `set` 子命令设置节点优先列表以及 `preferenced`、`failback` 和 `numsecondaries` 属性。

只能为每个设备组分配一个 Solaris Volume Manager 磁盘集。设备组的名称必须始终与磁盘集的名称相匹配。

无法使用 `add-node` 或 `remove-node` 子命令在 Solaris Volume Manager 设备组中添加或删除节点。可以使用 Solaris Volume Manager `metaset` 命令在底层 Solaris Volume Manager 磁盘集中添加或删除节点。

无法使用 `delete` 子命令从群集配置中删除 Solaris Volume Manager 设备组。而应改用 Solaris Volume Manager `metaset` 命令删除底层 Solaris Volume Manager 磁盘集。

只有 `export`、`list`、`show`、`status` 和 `sync` 子命令可用于 Solaris Volume Manager 多所有者磁盘集。必须使用 Solaris Volume Manager 命令或实用程序才能创建和删除 Solaris Volume Manager 设备组的底层磁盘集。

`-v`
`--version`

显示该命令的版本。

指定此选项时，不要将其与子命令、操作数或其他选项结合使用。子命令、操作数或其他选项将被忽略。`-v` 选项只显示命令的版本。不会执行任何其他操作。

`-v`
`--verbose`

将详细消息显示到标准输出。

可以将此选项与命令的任何格式一起使用。

支持以下操作数：

devicegroup 指定一个设备组。
`cldevicegroup` 命令只接受 Oracle Solaris Cluster 设备组名称作为操作数。对于大多数接受多个设备组名称的命令格式，可以使用加号 (+) 指定所有可能的设备组。

注 - + 操作数只包括手动创建的设备组，忽略所有自动创建的、`autogen` 属性设置为 `true` 的设备组。Oracle Solaris Cluster 软件在每次系统引导时会自动创建此类设备组。要对这些“隐藏”设备组应用命令，必须明确指定每个设备组。

[Intro\(1CL\) \[17\]](#) 手册页列出了该命令集中所有命令的一整套退出状态代码。

如果命令对于所有指定的操作数均成功执行，则返回零 (`CL_NOERR`)。如果某个操作数出错，命令会继续处理操作数列表中的下一个操作数。返回的退出代码始终反映最先发生的错误。

此命令返回下列退出状态代码：

0 `CL_NOERR`
 没有错误

1 `CL_ENOMEM`
 交换空间不足

3 CL_EINVAL
参数无效

6 CL_EACCESS
权限被拒绝

35 CL_EIO
I/O 错误

36 CL_ENOENT
没有这样的对象

39 CL_EEXIST
对象已存在

例 54 修改设备组

以下示例显示了如何将设备组 devgrp1 的 preference 属性设置为 true，并将 numsecondaries 属性设置为 2。该命令还指定所需的节点列表 phys-schost-1,phys-schost-2,phys-schost-3。

```
# cldevicegroup set -p preferenced=true -p numsecondaries=2 \  
-n phys-schost-1,phys-schost-2,phys-schost-3 devgrp1
```

例 55 修改原始磁盘设备组

以下示例显示了如何修改现有的原始磁盘设备组 rawdevgrp1。该命令在新成员设备列表中指定设备 d3 和 d4，并将 localonly 属性设置为 true。节点 phys-schost-1 是允许用于纯本地原始磁盘设备组的唯一主节点。

```
# cldevicegroup set -d d3,d4 \  
-p localonly=true -n phys-schost-1 rawdevgrp1
```

例 56 重置设备组的 numsecondaries 属性

以下示例显示了如何通过不指定任何值给设备组 devgrp1 的 numsecondaries 属性，将其重置为相应的系统默认值。

```
# cldevicegroup set -p numsecondaries= devgrp1
```

例 57 切换设备组

以下示例说明如何将设备组 devgrp1 切换到新的主节点 phys-schost-2。

```
# cldevicegroup switch -n phys-schost-2 devgrp1
```

例 58 禁用设备组

以下示例显示了如何禁用设备组 devgrp1 。

```
# cldevicegroup disable devgrp1
```

例 59 使设备组脱机

以下示例说明如何先将设备组 devgrp1 脱机再将其禁用。

```
# cldevicegroup offline devgrp1
# cldevicegroup disable devgrp1
```

例 60 使设备组在其主节点上联机

以下示例说明如何使设备组 devgrp1 在其默认主节点上联机。该命令首先启用设备组。

```
# cldevicegroup online -e devgrp1
```

例 61 使设备组在指定节点上联机

以下示例说明如何使设备组 devgrp1 在作为其新主节点的 phys-schost-2 上联机。

```
# cldevicegroup switch -n phys-schost-2 devgrp1
```

例 62 向设备组添加新节点

以下示例显示了如何向设备组 devgrp1 添加一个新节点 phys-schost--3。此设备组不属于设备组类型 svm。

```
# cldevicegroup add-node -n phys-schost-3 devgrp1
```

例 63 删除设备组

以下示例说明如何从 Oracle Solaris Cluster 配置中删除设备组 devgrp1。此设备组不属于设备组类型 svm。

```
# cldevicegroup delete devgrp1
```

例 64 将复制信息与设备组配置进行同步

以下示例显示了如何让 Oracle Solaris Cluster 软件知道设备组 devgrp1 中的磁盘所使用的复制配置。

```
# cldevicegroup sync devgrp1
```

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

[Intro\(1CL\) \[17\]](#)、[cldevice\(1CL\) \[51\]](#)、[cluster\(1CL\) \[461\]](#)、[Unresolved link to "metaset1M"](#)、[clconfiguration\(5CL\) \[1245\]](#)、[Unresolved link to "rbac5"](#)、[did\(7\) \[1275\]](#)

[Unresolved link to "Oracle Solaris Cluster 系统管理指南"](#)

超级用户可以运行此命令的所有格式。

此外，任何用户都可以运行带有以下选项的此命令：

- -? (帮助) 选项
- -V (版本) 选项

要运行带有其他子命令的此命令，超级用户以外的用户需要具有 RBAC 授权。请参见下表。

子命令	RBAC 授权
add-device	solaris.cluster.modify
add-node	solaris.cluster.modify
create	solaris.cluster.modify
delete	solaris.cluster.modify
disable	solaris.cluster.modify
enable	solaris.cluster.modify
export	solaris.cluster.read
list	solaris.cluster.read
offline	solaris.cluster.admin
online	solaris.cluster.admin
remove-device	solaris.cluster.modify
remove-node	solaris.cluster.modify
set	solaris.cluster.modify
show	solaris.cluster.read
status	solaris.cluster.read
switch	solaris.cluster.modify
sync	solaris.cluster.admin

名称

clinterconnect, clintr — 管理 Oracle Solaris Cluster 互连

```
/usr/cluster/bin/clinterconnect -V

/usr/cluster/bin/clinterconnect [Subcommand] -?

/usr/cluster/bin/clinterconnect subcommand [options] -v
    [endpoint[,endpoint] ...]

/usr/cluster/bin/clinterconnect add [-d] endpoint[,endpoint] ...

/usr/cluster/bin/clinterconnect add -i {- | clconfigfile} [-d]
    [-n node[,...]] {+ | endpoint[,endpoint] ...}

/usr/cluster/bin/clinterconnect disable [-n node[,...]]
    {+ | endpoint[,endpoint] ...}

/usr/cluster/bin/clinterconnect enable [-n node[,...]]
    {+ | endpoint[,endpoint] ...}

/usr/cluster/bin/clinterconnect export [-o {- | configfile}] [-n
    node[,...]] [+ | endpoint[,endpoint] ...]

/usr/cluster/bin/clinterconnect remove [-l] endpoint[,endpoint] ...

/usr/cluster/bin/clinterconnect show [-n node[,...]]
    [+ | endpoint[,endpoint] ...]

/usr/cluster/bin/clinterconnect status [-n node[,...]]
    [+ | endpoint[,endpoint] ...]
```

clinterconnect 命令用于管理群集互连配置以及显示配置和状态信息。clintr 命令是 clinterconnect 命令的简短格式。clinterconnect 命令和 clintr 命令是相同的。您可以使用此命令的任一种格式。

群集互连由以电缆连接的两个端点构成。端点可以是节点上的适配器或交换机，也称为结点。电缆可以连接一个适配器和一个交换机，在某些拓扑中还能连接两个适配器。群集拓扑管理器使用可用的电缆在节点之间构建端到端互连路径。提供给此命令的群集互连组件的名称应正确反映实际物理配置。如果无法做到这一点，将会阻止系统构建端到端群集互连路径。缺乏群集互连功能会导致群集节点之间无法相互通信、节点发生紧急情况及类似情况。

必须从已联机且处于群集模式的群集节点运行 clinterconnect 命令。

此命令的一般格式如下所示：

```
clinterconnect [subcommand] [options] [operands]
```

仅当 *options* 指定 *-?* 选项或 *-v* 选项时，才可以省略 *subcommand*。

此命令的每个选项都有长格式和简短格式两种格式。在本手册页的“选项”部分给出了每个选项的两种格式，以及对该选项的描述。

可以在非全局区域中使用该命令的某些格式。有关有效使用该命令的更多信息，请参见各个子命令的描述。为便于管理，请在全局区域中使用此命令。

子命令

支持以下子命令：

add

添加作为命令操作数指定的新群集互连组件。

仅在全局区域中使用此子命令。

操作数的语法确定是要添加电缆、交换机还是适配器。有关更多信息，请参阅此手册页的“操作数”部分。

使用 *add* 子命令可配置适配器和其他节点上的适配器或互连交换机之间的互连电缆。组成电缆的适配器或交换机端点不需要已存在。也可以使用此子命令向配置添加适配器或交换机。

在将适配器或交换机添加到配置时，该命令还会启用所添加的适配器或交换机。在添加电缆时，该命令还会启用电缆的每个端点，如果这些端点尚未启用。

在双节点群集中，如果所添加的电缆在每个端点都有一个适配器，则还会创建一个虚拟交换机。

使用 *-d* 选项可在禁用状态下添加端点。

如果使用 *-i* 选项指定配置文件，则可以指定加号 (+) 作为操作数。在使用此操作数时，该命令会创建配置文件中指定的、且群集中尚不存在的所有互连组件。

超级用户以外的用户需要具有 *solaris.cluster.modify RBAC* 授权才能使用该子命令。

有关删除互连组件的信息，请参见 *remove* 命令的描述。

disable

禁用作为命令操作数指定的互连组件。

仅在全局区域中使用此子命令。

操作数的语法确定是要禁用电缆、交换机还是适配器。有关更多信息，请参阅此手册页的“操作数”部分。

如果尝试禁用连接到已启用的电缆的适配器或交换机，该操作会导致发生错误。在尝试禁用所连接的适配器之前，必须首先禁用电缆。

在禁用电缆时，该命令还会禁用与电缆关联的每个端点（可能是适配器也可能是交换机端口）。如果所有交换机端口都处于禁用状态，该命令还会禁用交换机。

如果尝试禁用的电缆或端点是活动群集节点的最后一条群集互连路径，该操作会导致错误。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

有关启用互连组件的信息，请参见 `enable` 子命令的描述。

`enable`

启用作为命令操作数指定的互连组件。

仅在全局区域中使用此子命令。

操作数的语法确定是要启用电缆、交换机还是适配器。有关更多信息，请参阅此手册页的“操作数”部分。

在启用电缆时，该命令还会启用与电缆关联的每个端点（可能是适配器也可能是交换机端口）。

有关禁用互连组件的信息，请参见 `disable` 子命令的描述。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

`export`

导出群集互连的配置信息。

仅在全局区域中使用此子命令。

如果使用 `-o` 选项提供一个文件名，则会将配置信息写入该新文件。如果未使用 `-o` 选项，则会将输出写入标准输出。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

`remove`

删除作为命令操作数指定的群集互连组件。

仅在全局区域中使用此子命令。

操作数的语法确定是要删除电缆、交换机还是适配器。有关更多信息，请参阅此手册页的“操作数”部分。

在删除电缆时以下行为适用：

- 必须首先禁用电缆，然后才可以删除电缆。
- 如果尝试删除处于启用状态的电缆，删除操作会导致错误。
- 如果删除已禁用的电缆，则还会删除电缆的端点，以下情况除外：
 - 交换机正被其他电缆使用。
 - 您同时指定了 `-l` 选项。

在删除适配器或交换机端点时以下行为适用：

- 如果删除未与电缆关联的端点，则会删除指定的端点。
- 如果尝试删除与电缆关联的端点，删除操作会导致错误。无论电缆处于启用还是禁用状态，都是如此。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

有关添加互连组件的信息，请参见 `add` 子命令的描述。

show

显示作为命令操作数指定的互连组件的配置。

您可以在全局区域或区域群集中使用该子命令。

配置信息包括组件是处于启用还是禁用状态。默认情况下，会显示所有互连组件的配置。

`show` 子命令接受加号 (+) 作为操作数来指定所有组件。可以使用 `-z` 选项查看您指定的独占 IP 区域群集的专用网络配置信息。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

status

显示互连路径的状态。默认情况下，报告会显示系统中所有互连路径的状态。可以使用 `-z` 选项显示您指定的独占 IP 区域群集的专用网络配置信息的状态。

您可以在全局区域或独占 IP 区域群集中使用该子命令。

下面是互连路径的可能状况。

`faulted` 互连路径遇到阻止其正常工作的错误。

`Path online` 互连路径处于联机状态且正在提供服务。

`waiting` 互连路径正在向 `Path online` 状态过渡。

要确定互连组件是处于启用还是禁用状态，请使用 `show` 子命令。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

支持以下选项：

`-?`

`--help`

显示帮助信息。使用该选项时，不会执行任何其他处理。

该选项可以单独使用，也可以和子命令结合使用。

- 如果单独指定此选项，将会显示可用子命令的列表。
- 如果使用子命令且指定此选项，将会显示该子命令的用法选项。

`-d`

指定在禁用状态下添加端点。

`-i {- | clconfigfile}`

`--input={- | clconfigfile-}`

`--input {- | clconfigfile-}`

指定用来添加或修改电缆的配置信息。此信息必须符合 [clconfiguration\(5CL\) \[1245\]](#) 手册页中定义的格式。此信息可包含在文件中或者通过标准输入提供。要指定标准输入，请提供减号 (-) 而不是文件名。

您在命令中指定的选项将覆盖群集配置文件中设置的任何选项。如果群集配置文件中缺少必需元素，则必须在命令行上指定这些元素。

可以随此选项使用减号 (-) 参数，以指定配置将作为标准输入提供。

`-l`

`--limited`

指定电缆删除操作仅删除电缆而不删除其任何端点。

`-l` 选项仅与 `remove` 子命令一起使用时才有效。如果没有为 `remove` 子命令指定此选项，该命令将删除指定的电缆以及任何关联的适配器。此外，如果电缆删除操作删除与交换机的最后一个连接，该命令还会从配置中删除交换机。

`-n node[...]`

`--node=node[...]`

`--node node[...]`

指定一个节点或一系列节点。使用此选项可将操作限制到连接到指定节点的适配器和电缆。

您可以通过节点名称或节点 ID 来指定一个节点。

`-o {- | clconfigfile}`

`--output={- | clconfigfile}`

`--output {- | clconfigfile}`

以 [clconfiguration\(5CL\) \[1245\]](#) 手册页所描述的格式显示互连配置。

只有 `export` 子命令接受 `-o` 选项。

如果将文件名作为参数提供给该选项，此命令将创建新的文件，配置会显示在该文件中。如果已经存在同名的文件，此命令将退出，并显示一个错误。不会对现有文件进行任何更改。

如果将减号 (-) 作为参数提供给该选项，命令会将配置信息显示到标准输出。命令的所有其他标准输出将受到抑制。

`-v`

`--version`

显示该命令的版本。

指定此选项时，不要将其与子命令、操作数或其他选项结合使用。子命令、操作数或其他选项将被忽略。`-v` 选项只显示命令的版本。不会执行任何其他操作。

-v

--verbose

将详细消息显示到标准输出。默认情况下，show 和 status 命令显示详细输出。
可以将此选项与命令的任何格式一起使用。

-z {zoneclustername}

--zoneclustername={zoneclustername }

--zoneclustername { zoneclustername}

指定您要执行操作的群集。

show 和 status 子命令支持此选项。

如果指定此选项，还必须指定以下列表中的参数之一：

zoneclustername 指定使用此选项的命令仅对名为 *zoneclustername* 的区域群集起作用。

global 指定使用此选项的命令仅对全局群集起作用。

此命令接受互连端点或以逗号分隔的端点对作为操作数。端点可以是适配器或交换机。
以逗号分隔的端点对表示电缆。

对于接受多个互连组件的命令格式，可以使用加号 (+) 参数指定所有可能的组件。

支持以下操作数：

node:adapter

指定适配器端点。

适配器端点具有一个节点名称和一个适配器名称。适配器名称由互连名称及紧跟其后的物理单元编号构成，例如 net0。承载适配器的节点不需要在群集中处于活动状态，这些操作也可成功执行。

以下类型的适配器可配置为群集传输适配器：

Ethernet 可以将 Ethernet 适配器连接到另一 Ethernet 适配器或 Ethernet 交换机。

InfiniBand 只能将 InfiniBand 适配器连接到 InfiniBand 交换机。

默认情况下，适配器被配置为使用 *dlpi* 传输类型。

要指定标记 VLAN 适配器，请使用通过物理设备名称和 VLAN 实例号得出的标记 VLAN 适配器名称。VLAN 实例号等于 VLAN ID 与 1000 相乘再加上原始物理单元号。例如，物理设备 net2 上的 VLAN ID 11 转化为带标记的 VLAN 适配器名称 net11002。

switch[@port]

指定交换机端点。

每个互连交换机名称在群集的名称空间内必须唯一。可以使用字母、数字或字母数字组合。交换机名称的第一个字符必须是字母。

如果没有为交换机端点提供 *port* 组件，该命令会使用默认端口名称。默认端口名称等于电缆另一端所连接节点的节点 ID。

可以将以下类型的交换机配置为群集传输交换机：

Ethernet 使用 Ethernet 交换机与 Ethernet 适配器。

InfiniBand 使用 InfiniBand 交换机与 InfiniBand 适配器。

默认情况下，交换机被配置为使用 *switch* 类型。

node:adapter, node:adapter

node:adapter,switch [@port]

指定电缆。

电缆是以逗号分隔的适配器或交换机端点对。端点的顺序无关紧要。使用电缆操作数添加完整的群集互连。因为在添加电缆时 *clinterconnect* 命令会自动创建两个端点，所以无需分别创建适配器或交换机端点。

[Intro\(1CL\) \[17\]](#) 手册页列出了该命令集中所有命令的一整套退出状态代码。

如果命令对于所有指定的操作数均成功执行，则返回零 (CL_NOERR)。如果某个操作数出错，命令会继续处理操作数列表中的下一个操作数。返回的退出代码始终反映最先发生的错误。

此命令返回下列退出状态代码：

0 CL_NOERR

没有错误

1 CL_ENOMEM

交换空间不足

3 CL_EINVAL

参数无效

6 CL_EACCESS

权限被拒绝

35 CL_EIO

I/O 错误

36 CL_ENOENT

没有这样的对象

37CL_EOP
不允许操作

38 CL_EBUSY
对象忙

39 CL_EEXIST
对象已存在

例 65 创建直连群集互连电缆

以下示例显示了如何添加电缆连接节点 `phys-schost-1` 上的适配器 `net0` 与节点 `phys-schost-2` 上的适配器 `net0` 之间的端口。

```
# clinterconnect add phys-schost-1:net0,phys-schost-2:net0
```

例 66 创建交换机和适配器之间的电缆

以下示例显示了如何在节点 `phys-schost-1` 上的适配器 `net0` 和交换机 `ether_switch` 之间添加电缆。

```
# clinterconnect add phys-schost-1:net0,ether_switch
```

例 67 禁用电缆

以下示例显示了如何禁用节点 `phys-schost-1` 上的适配器 `net0` 和交换机 `ether_switch` 之间连接的电缆。

```
# clinterconnect disable phys-schost-1:net0,ether_switch
```

例 68 删除群集互连电缆

以下示例显示了如何删除节点 `phys-schost-1` 上的适配器 `net0` 和交换机 `ether_switch` 之间连接的电缆。

```
# clinterconnect remove phys-schost-1:net0,ether_switch
```

例 69 创建标记 VLAN 适配器和交换机之间的电缆

以下示例显示了如何在节点 `phys-schost-1` 上的标记 VLAN 适配器 `net73002` 和支持 VLAN 的交换机 `switch1` 之间添加电缆。适配器的物理名称是 `net2`，VLAN ID 是 73。

```
# clinterconnect add phys-schost-1:net73002,switch1
```

例 70 启用交换机

以下示例显示了如何启用交换机端点 `switch1`。

```
# clinterconnect enable switch1
```

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving（发展中）

[Intro\(1CL\) \[17\]](#)、[cluster\(1CL\) \[461\]](#)、[clconfiguration\(5CL\) \[1245\]](#)、[Unresolved link to "rbac5"](#)

[Unresolved link to "Oracle Solaris Cluster 4.2 Hardware Administration Manual "](#)

[Unresolved link to "Oracle Solaris Cluster 软件安装指南 "](#)

超级用户可以运行此命令的所有格式。

任何用户都可以运行带有以下选项的此命令。

- `-?`（帮助）选项
- `-V`（版本）选项

要运行带有其他子命令的此命令，超级用户以外的用户需要具有 RBAC 授权。请参见下表。

子命令	RBAC 授权
<code>add</code>	<code>solaris.cluster.modify</code>
<code>disable</code>	<code>solaris.cluster.modify</code>
<code>enable</code>	<code>solaris.cluster.modify</code>
<code>export</code>	<code>solaris.cluster.read</code>
<code>remove</code>	<code>solaris.cluster.modify</code>
<code>show</code>	<code>solaris.cluster.read</code>
<code>status</code>	<code>solaris.cluster.read</code>

名称

clinterconnect, clintr — 管理 Oracle Solaris Cluster 互连

```
/usr/cluster/bin/clinterconnect -V

/usr/cluster/bin/clinterconnect [Subcommand] -?

/usr/cluster/bin/clinterconnect subcommand [options] -v
    [endpoint[,endpoint] ...]

/usr/cluster/bin/clinterconnect add [-d] endpoint[,endpoint] ...

/usr/cluster/bin/clinterconnect add -i {- | clconfigfile} [-d]
    [-n node[,...]] {+ | endpoint[,endpoint] ...}

/usr/cluster/bin/clinterconnect disable [-n node[,...]]
    {+ | endpoint[,endpoint] ...}

/usr/cluster/bin/clinterconnect enable [-n node[,...]]
    {+ | endpoint[,endpoint] ...}

/usr/cluster/bin/clinterconnect export [-o {- | configfile}] [-n
    node[,...]] [+ | endpoint[,endpoint] ...]

/usr/cluster/bin/clinterconnect remove [-l] endpoint[,endpoint] ...

/usr/cluster/bin/clinterconnect show [-n node[,...]]
    [+ | endpoint[,endpoint] ...]

/usr/cluster/bin/clinterconnect status [-n node[,...]]
    [+ | endpoint[,endpoint] ...]
```

clinterconnect 命令用于管理群集互连配置以及显示配置和状态信息。clintr 命令是 clinterconnect 命令的简短格式。clinterconnect 命令和 clintr 命令是相同的。您可以使用此命令的任一种格式。

群集互连由以电缆连接的两个端点构成。端点可以是节点上的适配器或交换机，也称为结点。电缆可以连接一个适配器和一个交换机，在某些拓扑中还能连接两个适配器。群集拓扑管理器使用可用的电缆在节点之间构建端到端互连路径。提供给此命令的群集互连组件的名称应正确反映实际物理配置。如果无法做到这一点，将会阻止系统构建端到端群集互连路径。缺乏群集互连功能会导致群集节点之间无法相互通信、节点发生紧急情况及类似情况。

必须从已联机且处于群集模式的群集节点运行 clinterconnect 命令。

此命令的一般格式如下所示：

```
clinterconnect [subcommand] [options] [operands]
```

仅当 *options* 指定 *-?* 选项或 *-v* 选项时，才可以省略 *subcommand*。

此命令的每个选项都有长格式和简短格式两种格式。在本手册页的“选项”部分给出了每个选项的两种格式，以及对该选项的描述。

可以在非全局区域中使用该命令的某些格式。有关有效使用该命令的更多信息，请参见各个子命令的描述。为便于管理，请在全局区域中使用此命令。

子命令

支持以下子命令：

add

添加作为命令操作数指定的新群集互连组件。

仅在全局区域中使用此子命令。

操作数的语法确定是要添加电缆、交换机还是适配器。有关更多信息，请参阅此手册页的“操作数”部分。

使用 *add* 子命令可配置适配器和节点上的适配器或互连交换机之间的互连电缆。组成电缆的适配器或交换机端点不需要已存在。也可以使用此子命令向配置添加适配器或交换机。

在将适配器或交换机添加到配置时，该命令还会启用所添加的适配器或交换机。在添加电缆时，该命令还会启用电缆的每个端点，如果这些端点尚未启用。

在双节点群集中，如果所添加的电缆在每个端点都有一个适配器，则还会创建一个虚拟交换机。

使用 *-d* 选项可在禁用状态下添加端点。

如果使用 *-i* 选项指定配置文件，则可以指定加号 (+) 作为操作数。在使用此操作数时，该命令会创建配置文件中指定的、且群集中尚不存在的所有互连组件。

超级用户以外的用户需要具有 *solaris.cluster.modify RBAC* 授权才能使用该子命令。

有关删除互连组件的信息，请参见 *remove* 命令的描述。

disable

禁用作为命令操作数指定的互连组件。

仅在全局区域中使用此子命令。

操作数的语法确定是要禁用电缆、交换机还是适配器。有关更多信息，请参阅此手册页的“操作数”部分。

如果尝试禁用连接到已启用的电缆的适配器或交换机，该操作会导致发生错误。在尝试禁用所连接的适配器之前，必须首先禁用电缆。

在禁用电缆时，该命令还会禁用与电缆关联的每个端点（可能是适配器也可能是交换机端口）。如果所有交换机端口都处于禁用状态，该命令还会禁用交换机。

如果尝试禁用的电缆或端点是活动群集节点的最后一条群集互连路径，该操作会导致错误。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

有关启用互连组件的信息，请参见 `enable` 子命令的描述。

`enable`

启用作为命令操作数指定的互连组件。

仅在全局区域中使用此子命令。

操作数的语法确定是要启用电缆、交换机还是适配器。有关更多信息，请参阅此手册页的“操作数”部分。

在启用电缆时，该命令还会启用与电缆关联的每个端点（可能是适配器也可能是交换机端口）。

有关禁用互连组件的信息，请参见 `disable` 子命令的描述。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

`export`

导出群集互连的配置信息。

仅在全局区域中使用此子命令。

如果使用 `-o` 选项提供一个文件名，则会将配置信息写入该新文件。如果未使用 `-o` 选项，则会将输出写入标准输出。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

`remove`

删除作为命令操作数指定的群集互连组件。

仅在全局区域中使用此子命令。

操作数的语法确定是要删除电缆、交换机还是适配器。有关更多信息，请参阅此手册页的“操作数”部分。

在删除电缆时以下行为适用：

- 必须首先禁用电缆，然后才可以删除电缆。
- 如果尝试删除处于启用状态的电缆，删除操作会导致错误。
- 如果删除已禁用的电缆，则还会删除电缆的端点，以下情况除外：
 - 交换机正被其他电缆使用。
 - 您同时指定了 `-l` 选项。

在删除适配器或交换机端点时以下行为适用：

- 如果删除未与电缆关联的端点，则会删除指定的端点。
- 如果尝试删除与电缆关联的端点，删除操作会导致错误。无论电缆处于启用还是禁用状态，都是如此。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

有关添加互连组件的信息，请参见 `add` 子命令的描述。

show

显示作为命令操作数指定的互连组件的配置。

您可以在全局区域或区域群集中使用该子命令。

配置信息包括组件是处于启用还是禁用状态。默认情况下，会显示所有互连组件的配置。

`show` 子命令接受加号 (+) 作为操作数来指定所有组件。可以使用 `-z` 选项查看您指定的独占 IP 区域群集的专用网络配置信息。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

status

显示互连路径的状态。默认情况下，报告会显示系统中所有互连路径的状态。可以使用 `-z` 选项显示您指定的独占 IP 区域群集的专用网络配置信息的状态。

您可以在全局区域或独占 IP 区域群集中使用该子命令。

下面是互连路径的可能状况。

`faulted` 互连路径遇到阻止其正常工作的错误。

`Path online` 互连路径处于联机状态且正在提供服务。

`waiting` 互连路径正在向 `Path online` 状态过渡。

要确定互连组件是处于启用还是禁用状态，请使用 `show` 子命令。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

支持以下选项：

`-?`

`--help`

显示帮助信息。使用该选项时，不会执行任何其他处理。

该选项可以单独使用，也可以和子命令结合使用。

- 如果单独指定此选项，将会显示可用子命令的列表。
- 如果使用子命令且指定此选项，将会显示该子命令的用法选项。

`-d`

指定在禁用状态下添加端点。

`-i {- | clconfigfile}`

`--input={- | clconfigfile-}`

`--input {- | clconfigfile-}`

指定用来添加或修改电缆的配置信息。此信息必须符合 [clconfiguration\(5CL\) \[1245\]](#) 手册页中定义的格式。此信息可包含在文件中或者通过标准输入提供。要指定标准输入，请提供减号 (-) 而不是文件名。

您在命令中指定的选项将覆盖群集配置文件中设置的任何选项。如果群集配置文件中缺少必需元素，则必须在命令行上指定这些元素。

可以随此选项使用减号 (-) 参数，以指定配置将作为标准输入提供。

`-l`

`--limited`

指定电缆删除操作仅删除电缆而不删除其任何端点。

`-l` 选项仅与 `remove` 子命令一起使用时才有效。如果没有为 `remove` 子命令指定此选项，该命令将删除指定的电缆以及任何关联的适配器。此外，如果电缆删除操作删除与交换机的最后一个连接，该命令还会从配置中删除交换机。

`-n node[...]`

`--node=node[...]`

`--node node[...]`

指定一个节点或一系列节点。使用此选项可将操作限制到连接到指定节点的适配器和电缆。

您可以通过节点名称或节点 ID 来指定一个节点。

`-o {- | clconfigfile}`

`--output={- | clconfigfile}`

`--output {- | clconfigfile}`

以 [clconfiguration\(5CL\) \[1245\]](#) 手册页所描述的格式显示互连配置。

只有 `export` 子命令接受 `-o` 选项。

如果将文件名作为参数提供给该选项，此命令将创建新的文件，配置会显示在该文件中。如果已经存在同名的文件，此命令将退出，并显示一个错误。不会对现有文件进行任何更改。

如果将减号 (-) 作为参数提供给该选项，命令会将配置信息显示到标准输出。命令的所有其他标准输出将受到抑制。

`-v`

`--version`

显示该命令的版本。

指定此选项时，不要将其与子命令、操作数或其他选项结合使用。子命令、操作数或其他选项将被忽略。`-v` 选项只显示命令的版本。不会执行任何其他操作。

-v

--verbose

将详细消息显示到标准输出。默认情况下，show 和 status 命令显示详细输出。
可以将此选项与命令的任何格式一起使用。

-z {zoneclustername}

--zoneclustername={zoneclustername }

--zoneclustername { zoneclustername}

指定您要执行操作的群集。

show 和 status 子命令支持此选项。

如果指定此选项，还必须指定以下列表中的参数之一：

zoneclustername 指定使用此选项的命令仅对名为 *zoneclustername* 的区域群集起作用。

global 指定使用此选项的命令仅对全局群集起作用。

此命令接受互连端点或以逗号分隔的端点对作为操作数。端点可以是适配器或交换机。
以逗号分隔的端点对表示电缆。

对于接受多个互连组件的命令格式，可以使用加号 (+) 参数指定所有可能的组件。

支持以下操作数：

node:adapter

指定适配器端点。

适配器端点具有一个节点名称和一个适配器名称。适配器名称由互连名称及紧跟其后的物理单元编号构成，例如 net0。承载适配器的节点不需要在群集中处于活动状态，这些操作也可成功执行。

以下类型的适配器可配置为群集传输适配器：

Ethernet 可以将 Ethernet 适配器连接到另一 Ethernet 适配器或 Ethernet 交换机。

InfiniBand 只能将 InfiniBand 适配器连接到 InfiniBand 交换机。

默认情况下，适配器被配置为使用 *dlpi* 传输类型。

要指定标记 VLAN 适配器，请使用通过物理设备名称和 VLAN 实例号得出的标记 VLAN 适配器名称。VLAN 实例号等于 VLAN ID 与 1000 相乘再加上原始物理单元号。例如，物理设备 net2 上的 VLAN ID 11 转化为带标记的 VLAN 适配器名称 net11002。

switch[@port]

指定交换机端点。

每个互连交换机名称在群集的名称空间内必须唯一。可以使用字母、数字或字母数字组合。交换机名称的第一个字符必须是字母。

如果没有为交换机端点提供 *port* 组件，该命令会使用默认端口名称。默认端口名称等于电缆另一端所连接节点的节点 ID。

可以将以下类型的交换机配置为群集传输交换机：

Ethernet 使用 Ethernet 交换机与 Ethernet 适配器。

InfiniBand 使用 InfiniBand 交换机与 InfiniBand 适配器。

默认情况下，交换机被配置为使用 *switch* 类型。

```
node:adapter, node:adapter
```

```
node:adapter,switch [@port]
```

指定电缆。

电缆是以逗号分隔的适配器或交换机端点对。端点的顺序无关紧要。使用电缆操作数添加完整的群集互连。因为在添加电缆时 *clinterconnect* 命令会自动创建两个端点，所以无需分别创建适配器或交换机端点。

[Intro\(1CL\) \[17\]](#) 手册页列出了该命令集中所有命令的一整套退出状态代码。

如果命令对于所有指定的操作数均成功执行，则返回零 (CL_NOERR)。如果某个操作数出错，命令会继续处理操作数列表中的下一个操作数。返回的退出代码始终反映最先发生的错误。

此命令返回下列退出状态代码：

0 CL_NOERR

没有错误

1 CL_ENOMEM

交换空间不足

3 CL_EINVAL

参数无效

6 CL_EACCESS

权限被拒绝

35 CL_EIO

I/O 错误

36 CL_ENOENT

没有这样的对象

37CL_EOP
不允许操作

38 CL_EBUSY
对象忙

39 CL_EEXIST
对象已存在

例 71 创建直连群集互连电缆

以下示例显示了如何添加电缆连接节点 `phys-schost-1` 上的适配器 `net0` 与节点 `phys-schost-2` 上的适配器 `net0` 之间的端口。

```
# clinterconnect add phys-schost-1:net0,phys-schost-2:net0
```

例 72 创建交换机和适配器之间的电缆

以下示例显示了如何在节点 `phys-schost-1` 上的适配器 `net0` 和交换机 `ether_switch` 之间添加电缆。

```
# clinterconnect add phys-schost-1:net0,ether_switch
```

例 73 禁用电缆

以下示例显示了如何禁用节点 `phys-schost-1` 上的适配器 `net0` 和交换机 `ether_switch` 之间连接的电缆。

```
# clinterconnect disable phys-schost-1:net0,ether_switch
```

例 74 删除群集互连电缆

以下示例显示了如何删除节点 `phys-schost-1` 上的适配器 `net0` 和交换机 `ether_switch` 之间连接的电缆。

```
# clinterconnect remove phys-schost-1:net0,ether_switch
```

例 75 创建标记 VLAN 适配器和交换机之间的电缆

以下示例显示了如何在节点 `phys-schost-1` 上的标记 VLAN 适配器 `net73002` 和支持 VLAN 的交换机 `switch1` 之间添加电缆。适配器的物理名称是 `net2`，VLAN ID 是 73。

```
# clinterconnect add phys-schost-1:net73002,switch1
```

例 76 启用交换机

以下示例显示了如何启用交换机端点 `switch1`。

```
# clinterconnect enable switch1
```

有关以下属性的描述，请参见 [Unresolved link to " attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

[Intro\(1CL\) \[17\]](#)、[cluster\(1CL\) \[461\]](#)、[clconfiguration\(5CL\) \[1245\]](#)、[Unresolved link to " rbac5"](#)

[Unresolved link to " Oracle Solaris Cluster 4.2 Hardware Administration Manual "](#)

[Unresolved link to " Oracle Solaris Cluster 软件安装指南 "](#)

超级用户可以运行此命令的所有格式。

任何用户都可以运行带有以下选项的此命令。

- `-?` (帮助) 选项
- `-V` (版本) 选项

要运行带有其他子命令的此命令，超级用户以外的用户需要具有 RBAC 授权。请参见下表。

子命令	RBAC 授权
<code>add</code>	<code>solaris.cluster.modify</code>
<code>disable</code>	<code>solaris.cluster.modify</code>
<code>enable</code>	<code>solaris.cluster.modify</code>
<code>export</code>	<code>solaris.cluster.read</code>
<code>remove</code>	<code>solaris.cluster.modify</code>
<code>show</code>	<code>solaris.cluster.read</code>
<code>status</code>	<code>solaris.cluster.read</code>

名称

clsnmpmib, clmib — 管理 Oracle Solaris Cluster SNMP MIB

```
/usr/cluster/bin/clsnmpmib -V
/usr/cluster/bin/clsnmpmibsubcommand [] -?
/usr/cluster/bin/clsnmpmib [subcommand] [options] -v [mib]
/usr/cluster/bin/clsnmpmib disable [-n node[,...]] {+ | mib ...}
/usr/cluster/bin/clsnmpmib enable [-n node[,...]] {+ | mib ...}
/usr/cluster/bin/clsnmpmib export [-n node[,...]]
    [-o {- | clconfigfile}] {+ | mib ...}
/usr/cluster/bin/clsnmpmib list [-n node[,...]] {+ | mib ...}
/usr/cluster/bin/clsnmpmib set [-p name=value] [...]
    [-n node[,...]] {+ | mib ...}
/usr/cluster/bin/clsnmpmib show [-n node[,...]] {+ | mib ...}
```

clsnmpmib 命令可管理当前节点上的现有 Oracle Solaris Cluster 简单网络管理协议 (Simple Network Management Protocol, SNMP) 管理信息库 (Management Information Base, MIB)。要创建可以管理 MIB 的 SNMP 主机，请参见 [clsnmphost\(1CL\) \[409\]](#) 手册页。要定义可以使用 SNMP 版本 3 (SNMP Version 3, SNMPv3) 协议访问 MIB 的 SNMPv3 用户，请参见 [clsnmpuser\(1CL\) \[425\]](#) 手册页。

此命令的一般格式如下所示：

```
clsnmpmib [subcommand] [options] [operands]
```

仅当 *options* 指定选项 *-?* 或 *-v* 时，才可以省略 *subcommand*。

此命令的每个选项都有长格式和简短格式两种格式。在“选项”部分给出了每个选项的两种格式，以及对该选项的描述。

有关更多信息，请参见 [Intro\(1CL\) \[17\]](#) 手册页。

Oracle Solaris Cluster MIB

Oracle Solaris Cluster 当前支持一个 MIB，即事件 MIB。Oracle Solaris Cluster SNMP 事件 MIB 可实时向 SNMP 管理器通知群集事件。一经启用，Oracle Solaris Cluster 事件 MIB 就会自动向 clsnmphost 命令定义的所有主机发送陷阱通知。Oracle Solaris Cluster 事件 MIB 在端口 11162 上发送陷阱通知。可在端口 11161 上查看 SNMP 树。

可以使用 `clsnmpmib set` 子命令指定 `min_severity` 或 `log_number` 值。由于群集会生成大量事件通知，因此只有严重程度为 `min_severity` 或更高的事件才作为陷阱通知发送。默认情况下，`min_severity` 值设置为 `NOTICE`。`log_number` 值指定在停止使用较旧的条目之前要在 MIB 表中记录的事件的数量。MIB 可维护一个只读表，其中包含最近发生的已作为陷阱发送的事件。事件数受 `log_number` 值限制。系统重新引导后此信息将不再存在。

仅可以在全局区域中使用此命令。

支持以下子命令：

`disable`

在指定节点禁用一个或多个群集 MIB。

仅可以在全局区域中运行该子命令。

如果没有指定 `-n` 选项，则仅禁用当前节点上的 MIB。禁用 MIB 时，无法访问 MIB 表，并且 MIB 不会发送任何陷阱通知。

超级用户以外的用户需要具有 `solaris.cluster.modify` 基于角色的访问控制 (Role-Based Access Control, RBAC) 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

`enable`

在指定节点启用一个或多个群集 MIB。

仅可以在全局区域中运行该子命令。

如果没有指定 `-n` 选项，则仅启用当前节点上的 MIB。要限制启用的 MIB，请使用 `mib` 操作数。

启用 MIB 时，也就启用了它的所有功能。但是，要使所有 MIB 功能完全发挥作用，可能有必要进行一些进一步的配置。例如，如果未配置任何主机，MIB 将无法发送陷阱通知。有关配置 SNMP 主机的信息，请参见 [clsnmphost\(1CL\) \[409\]](#) 手册页。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

`export`

导出群集 MIB 配置信息。

仅可以在全局区域中运行该子命令。

使用 `-n` 选项可指定一个或多个要从中导出 MIB 配置信息的节点。如果在不带 `-n` 选项的情况下使用 `export`，则该子命令仅从当前节点导出 MIB 配置信息。默认情况下，该子命令导出当前节点上所有 MIB 的配置信息。要进一步完善输出，请指定您需要其配置信息的一个或多个 MIB 的名称。

有关 `export` 子命令的输出格式的更多信息，请参见 [clconfiguration\(5CL\) \[1245\]](#) 手册页。默认情况下，所有输出都会发送到标准输出。使用 `-o` 选项后跟文件名可将输出重定向到该文件。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

list

显示指定节点上的群集 MIB 列表。

仅可以在全局区域中运行该子命令。

使用 `-n` 选项可指定要列出的群集 MIB 的节点。如果在不带 `-n` 选项的情况使用 `list` 子命令，则该子命令仅列出当前节点上的 MIB。要限制列出的 MIB，请指定要列出的一个或多个 MIB 的名称。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

set

更改指定节点上的一个或多个 MIB 使用的 SNMP 协议 `min_severity` 或 `log_number` 设置。

仅可以在全局区域中运行该子命令。

默认情况下，该子命令会更改节点上的所有 MIB。如果没有指定节点，则仅修改当前节点上 MIB 的 SNMP 属性。必须通过使用 `-p` 选项指定 SNMP 属性。

所有 MIB 都使用以下默认属性值：`protocol:SNMPv2`，`min_severity:NOTICE`，`log_number:100`。除非使用 `mib` 操作数指定 MIB 名称，否则 `set` 子命令会更改所有 MIB 的协议、`min_severity` 或 `log_number` 设置。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

show

显示指定节点上 MIB 的信息。

仅可以在全局区域中运行该子命令。

`show` 子命令显示 MIB 的名称、其 SNMP 协议版本、`min_severity` 值或 `log_number` 值。默认情况下，该子命令会显示节点上所有 MIB 的信息。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

支持以下选项：

`-?`

`--help`

显示帮助信息。

指定此选项时，子命令可有可无。

- 如果在不带子命令的情况下使用此选项，将会显示可用子命令的列表。
- 如果将该选项与子命令结合使用，将会显示该子命令的用法选项。

使用该选项时，不会执行任何其他处理。

`-n node[...]`

`--node[s] node[...]`

指定一个节点或一系列节点。可以使用节点名称或节点 ID 指定每个节点。`clsnmpmib` 命令的所有格式都接受此选项。可以使用 `-n` 选项指定要在其上执行操作的节点。如果不使用 `-n` 选项，则该命令会采用当前节点。

`-o {- | clconfigfile}`

`--output {- | clconfigfile}`

指定要写入群集 MIB 配置信息的位置。此位置可以是文件，也可以是标准输出。要指定标准输出，请指定减号 (-) 而不是文件名。如果指定标准输出，将会抑制该命令的其他所有标准输出。如果没有指定 `-o` 选项，输出将发送到标准输出。此选项仅可以与 `export` 子命令一同指定。

配置信息是采用 [clconfiguration\(5CL\) \[1245\]](#) 手册页中定义的格式写入的。

`-p name= value`

`--property=name =value`

`--property name=value`

`version` 指定 MIB 将使用的 SNMP 协议的版本。Oracle Solaris Cluster 支持 SNMPv2 和 SNMPv3 协议版本。

`min_severity` 指定最低严重程度值。只有其值等于或大于 `min_severity` 值的事件才会记录在 MIB 表中，对于这些事件，将向配置的主机发送陷阱。

`log_number` 指定在停止使用较旧的条目之前要在 MIB 表中记录的事件的数量。

允许使用多个 `-p name= value` 实例。

可以使用该选项设置以下属性：

`version`

指定 MIB 将使用的 SNMP 协议的版本。您可以按如下方式指定 `value`：

- `version=SNMPv2`
- `version=snmpv2`
- `version=2`
- `version=SNMPv3`
- `version=snmpv3`
- `version=3`

`min_severity`

指定要用于 MIB 的最低严重性级别值。您可以按如下方式指定值：

- `min_severity=NOTICE`

-
- min_severity=WARNING
 - min_severity=ERROR
 - min_severity=CRITICAL
 - min_severity=FATAL

允许使用大写或小写的值。

log_number

指定在停止使用较旧的条目之前要在 MIB 表中记录的事件的数量。默认值为 100。值范围为 100 到 500。您可以按如下方式指定value：

- log_number=*number*

-V

--version

显示该命令的版本。

不要将该选项与子命令、操作数或其他选项一起指定，因为子命令、操作数或其他选项会被忽略。-V 选项只显示命令的版本。不会执行任何其他操作。

-v

--verbose

将详细信息显示到标准输出。

可以将该选项与命令的任何格式一同指定，尽管一些子命令可能不会生成扩展输出。例如，指定详细选项时 export 子命令不会生成扩展输出。

支持以下操作数：

mib 指定要应用子命令的一个或多个 MIB 的名称。如果您没有指定此操作数，该子命令将使用默认的加号 (+)，这意味着所有 MIB。如果使用 *mib* 操作数，请在空格分隔列表中所有其他命令行选项之后指定 MIB。

+

所有群集 MIB。

如果命令对于所有指定的操作数均成功执行，则返回零 (CL_NOERR)。如果某个操作数出错，命令会继续处理操作数列表中的下一个操作数。返回的退出代码始终反映最先发生的错误。

此命令返回下列退出状态代码：

0 CL_NOERR

没有错误

您发出的命令已成功完成。

1 CL_ENOMEM

交换空间不足

某个群集节点耗尽了交换内存或其他操作系统资源。

3 CL_EINVAL

参数无效

您键入的命令不正确，或者使用 -i 选项提供的群集配置信息语法不正确。

6 CL_EACCESS

权限被拒绝

指定的对象不可访问。要发出命令，可能需要拥有超级用户或 RBAC 访问权限。有关更多信息，请参见 [Unresolved link to " su1M"](#) 和 [Unresolved link to " rbac5"](#) 手册页。

18 CL_EINTERNAL

遇到内部错误

内部错误表示软件缺陷或其他缺陷。

35 CL_EIO

I/O 错误

发生了物理输入/输出错误。

36 CL_ENOENT

没有这样的对象

由于以下某个原因，找不到您所指定的对象：

- 该对象不存在。
- 您尝试使用 -o 选项创建的配置文件路径中的某个目录不存在。
- 您尝试使用 -i 选项访问的配置文件包含错误。

例 77 列出 MIB

以下命令可列出群集节点上的所有 MIB。

```
# clsnmpmib list
Event
```

例 78 启用 MIB

以下命令可启用当前节点上的事件 MIB。

```
# clsnmpmib enable event
```

群集 MIB 的名称不区分大小写。

例 79 更改协议

以下命令可将 phys-cluster-2 群集节点上 Event (事件) MIB 的协议更改为 SNMPv3。

```
# clsnmpmib set -n phys-cluster-2 -p version=SNMPv3 Event
```

如果使用 -n 选项，也可以使用节点 ID 代替节点名称。

例 80 显示配置

以下命令可显示群集节点 phys-cluster-1 和 phys-cluster-2 上的配置信息。

```
# clsnmpmib show -n phys-cluster-1,phys-cluster-2
```

```
--- SNMP MIB Configuration on myhost ---
```

```
SNMP MIB Name:          phys-cluster-1
State:                  Event
Enabled:                yes
Protocol:               SNMPv3
min_severity:          1
log_number:             100
SNMP MIB Name:          phys-cluster-2
State:                  Event
Enabled:                yes
Protocol:               SNMPv3
min_severity:          3
log_number:             250
```

例 81 更改 Min Severity 值

以下命令可将 phys-cluster-2 群集节点上事件 MIB 的 min_severity 更改为 WARNING。

```
# clsnmpmib set -n phys-cluster-2 -p min_severity=WARNING Event
```

如果使用 -n 选项，也可以使用节点 ID 代替节点名称。

例 82 更改 Log_Number 值

以下命令可将 phys-cluster-2 群集节点上事件 MIB 的 log_number 更改为 250。

```
# clsnmpmib set -n phys-cluster-2 -p log_number=250 Event
```

如果使用 -n 选项，也可以使用节点 ID 代替节点名称。

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

/usr/cluster/lib/mib/sun-cluster-event-mib.mib

Oracle Solaris Cluster SNMP 事件 MIB 定义文件

[clsnmphost\(1CL\) \[409\]](#)、[clsnmpuser\(1CL\) \[425\]](#)、[Intro\(1CL\) \[17\]](#)、[cluster\(1CL\) \[461\]](#)、[sceventmib\(1M\) link to "attributes5"](#)、[Unresolved link to "rbac5"](#)、[clconfiguration\(5CL\) \[1245\]](#)

超级用户可以运行此命令的所有格式。

所有用户都可以运行带有 `-?` (帮助) 或 `-v` (版本) 选项的此命令。

要运行带有其他子命令的 `clsnmpmib` 命令，超级用户以外的用户需要具有 RBAC 授权。请参见下表。

子命令	RBAC 授权
disable	solaris.cluster.modify
enable	solaris.cluster.modify
export	solaris.cluster.read
list	solaris.cluster.read
set	solaris.cluster.modify
show	solaris.cluster.read

名称

clnasdevice, clnas — 为 Oracle Solaris Cluster 管理对 NAS 设备的访问权限

```
/usr/cluster/bin/clnasdevice -V

/usr/cluster/bin/clnasdevice [subcommand] -?

/usr/cluster/bin/clnasdevice subcommand [options] -v [nasdevice[...]]

/usr/cluster/bin/clnasdevice add -t type {-p name=value
[,...] | -u userid} [-f passwdfile] [-Z {zoneclustername | global}]
nasdevice

/usr/cluster/bin/clnasdevice add -i {- | clconfigfile}[-t type]
[-p name=value | -u userid] {-f passwdfile} [-Z {zoneclustername | global}]
{nasdevice

/usr/cluster/bin/clnasdevice add-dir -d directory[,...] [-Z
{zoneclustername | global}] nasdevice

/usr/cluster/bin/clnasdevice add-dir -i {- | clconfigfile} [-d all |
directory [,...]] [-f passwordfile] [-Z {zoneclustername | global}]
{nasdevice

/usr/cluster/bin/clnasdevice export [-o {- | clconfigfile}] [-t
type[,...]] [-d all | directory[,...]] [+ | nasdevice[...]]

/usr/cluster/bin/clnasdevice find-dir [-d {all |directory[,...]}
[-t type[,...]] [-Z {zoneclustername[,...] | global | all}]
[+ | nasdevice[...]]

/usr/cluster/bin/clnasdevice list [-t type[,...]] [-Z
{zoneclustername[,...] | global | all}] [+ | nasdevice[...]]

/usr/cluster/bin/clnasdevice remove [-t type[,...]] [-Z
{zoneclustername | global}] [-F ] [+ | nasdevice[...]]

/usr/cluster/bin/clnasdevice remove-dir -d all | directory[,...]
[-Z {zoneclustername | global}] nasdevice

/usr/cluster/bin/clnasdevice set {-p name=value[,...] | -u
userid} [-f passwdfile] [-Z {zoneclustername | global}]
nasdevice

/usr/cluster/bin/clnasdevice show [-d {all | directory[,...]} [-t
type[,...]] [-Z {zoneclustername[,...] | global | all}]
[+ | nasdevice[...]]
```

clnasdevice 命令用于管理 NAS 设备及其目录或项目的 Oracle Solaris Cluster 配置信息。

`clnas` 命令是 `clnasdevice` 命令的简短格式。`clnas` 和 `clnasdevice` 命令是相同的。您可以使用此命令的任一种格式。

此命令的一般格式如下所示：

```
clnasdevice [subcommand] [options] [operands]
```

仅当 *options* 指定 `-?` 选项或 `-v` 选项时，才可以省略 *subcommand*。

此命令的每个选项都有长格式和简短格式两种格式。在本手册页“选项”部分给出了两种格式的选项，以及对该选项的描述。

在群集中使用 `clnasdevice` 命令配置 NAS 设备之前，您的 NAS 设备必须符合以下条件：

- NAS 设备必须已设置且正常工作。
- NAS 设备必须已引导且正在运行。
- NAS 设备的目录必须已创建且可用于群集节点。
- 如果 NAS 设备将用作法定设备，则必须创建该法定设备的 LUN。有关配置 NAS 法定设备的信息，请参见 `clquorum` 手册页。

在将设备配置到群集之前，您可能还需要执行其他任务，具体取决于 NAS 设备供应商。有关这些任务的详细信息，请参见“选项”部分的 `-t` 选项。有关设置 NAS 设备和导出目录的过程，请参阅特定 NAS 设备的文档。

在 NAS 设备完全可操作且准备好为群集提供存储之后，请使用 `clnasdevice` 命令管理群集中的 NAS 设备配置信息。否则，群集将无法检测到 NAS 设备及其导出目录。因此，群集将无法保护这些目录中信息的完整性。

使用 `clnasdevice` 命令执行以下管理任务：

- 创建 NAS 设备配置
- 更新特定于 NAS 类型的属性
- 从群集配置中删除 NAS 设备的目录
- 从群集配置中删除 NAS 设备

只能在活动群集节点上运行 `clnasdevice` 命令。无论您在哪个节点上运行该命令，运行结果始终相同。

在区域群集中，可以将 `clnasdevice` 命令与所有子命令（`export` 除外）一起使用。也可以将 `-z` 选项与所有子命令（`export` 除外）一起使用来指定要将操作限制到的特定区域群集的名称。

您可以从全局群集节点访问所有区域群集信息，但无法从特定区域群集访问其他区域群集信息。如果不将操作限制到特定区域群集，则所用的子命令仅会在当前群集中运行。

支持以下子命令：

add

将 NAS 设备添加到 Oracle Solaris Cluster 配置中。

使用 `-t` 选项可指定 NAS 设备的供应商。有关详细信息，请参见“选项”部分的 `-t` 选项描述。

可能还需要设置其他属性，具体取决于您的 NAS 设备类型。“选项”部分的 `-t` 选项描述中也介绍了这些必需属性。

超级用户以外的用户需要具有 `solaris.cluster.modify` 基于角色的访问控制 (role-based access control, RBAC) 授权才能使用该命令。请参见 [Unresolved link to "rbac5"](#)。

另请参见 `remove` 子命令的描述。

add-dir

将已配置的 NAS 设备的指定目录或项目添加到群集配置中。在使用该子命令之前，必须已在设备上创建这些目录或项目并使其可用于群集。有关创建目录或项目的信息，请参见您的 NAS 设备的文档。

可以使用下列其中一种方法添加 NAS 设备目录或项目：

- 使用 `clnasdevice add` 命令在群集中配置 NAS 设备。然后使用 `clnasdevice add-dir` 命令在群集中配置该设备的目录或项目。
- 使用 `clnasdevice add-dir -i configurationfile` 格式的命令可在单个步骤中同时添加设备并配置设备目录或项目。要使用此方法添加目录或项目，请使用 `-f` 选项提供密码文件。有关该选项的详细信息，请参见“选项”部分。有关更多信息，请参见 [clconfiguration\(5CL\) \[1245\]](#) 手册页。

只要在 NAS 设备上创建新目录或项目并使其可用于群集节点，就需要使用此 `add-dir` 子命令将目录或项目添加到群集配置。可以使用 `find-dir` 子命令列出可使用 `add-dir` 子命令添加到群集的可用目录或项目。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该命令。请参见 [Unresolved link to "rbac5"](#)。

另请参见 `remove-dir` 子命令的描述。

export

导出群集 NAS 设备的配置信息。如果使用 `-o` 选项指定一个文件，则会将配置信息写入该文件。如果不使用 `-o` 选项，则会将输出写入标准输出 (stdout)。

`export` 子命令不会修改群集配置信息。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该命令。请参见 [Unresolved link to "rbac5"](#)。

find-dir

显示群集可能会使用的 NAS 设备上设置的 `sun_uss` 项目。尚未使用 `add-dir` 子命令将这些项目添加到群集配置中。输出中列出的这些项目可以在您使用 `add-dir` 子命令时充当 `-d` 选项的候选项。

要显示特定类型的 NAS 设备，请指定 `-t` 选项。

要显示 `sun_uss` 项目和这些项目内的文件系统，请指定 `-v` 选项。

要显示特定的 `sun_uss` NAS 设备项目，请指定 `-d` 选项。

要显示特定 `sun_uss` NAS 设备项目和这些项目内的文件系统，请指定 `-v` 和 `-d` 选项。

`find-dir` 子命令不会修改群集配置信息。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该命令。请参见 [Unresolved link to "rbac5"](#)。

list

显示群集中配置的 NAS 设备。

要显示群集中配置的设备的目录及设备类型，请使用详细选项 `-v`。

要显示特定类型的 NAS 设备，请使用 `-t` 选项。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该命令。请参见 [Unresolved link to "rbac5"](#)。

remove

从 Oracle Solaris Cluster 配置中删除指定的一个或多个 NAS 设备。

如果不指定强制选项 `-F`，则必须已通过使用 `remove-dir` 子命令从配置中删除 NAS 设备目录。

如果指定强制选项 `-F`，该命令将从群集配置中删除 NAS 设备及其目录。请参见 `-OPTIONS` 中的 `F`。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该命令。请参见 [Unresolved link to "rbac5"](#)。

另请参见 `add` 子命令的描述。

remove-dir

从 Oracle Solaris Cluster 配置中删除指定的 NAS 目录或项目。

`remove-dir` 子命令用于删除 `-d` 选项指定的导出目录或项目。在使用 `-d all` 时，该子命令可删除指定 NAS 设备的所有目录或项目。

只要从 NAS 设备中删除目录或项目，都需要使用此 `remove-dir` 子命令从群集配置中删除目录或项目。群集配置中的 NAS 目录或项目必须与从 NAS 设备导出的现有目录或项目相匹配。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该命令。请参见 [Unresolved link to "rbac5"](#)。

另请参见 `add-dir` 子命令的描述。

set

设置特定 NAS 设备的指定属性。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该命令。请参见 [Unresolved link to "rbac5"](#)。

show

在未提供任何选项时，显示以下信息：

- Oracle Solaris Cluster 中配置的所有当前 NAS 设备的列表
- 每个 NAS 设备的可用目录
- 与每个 NAS 设备关联的所有属性

要显示特定类型的 NAS 设备，请指定 `-t` 选项。要显示有关特定设备的信息，请传送 NAS 设备的主机名用作该命令的操作对象。

要显示指定项目中包含的文件系统，请将 `-d` 和 `-v` 选项与 `show` 子命令一起使用。可以使用 `all` 关键字显示 NAS 设备的所有项目，或者仅显示单个项目。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该命令。请参见 [Unresolved link to "rbac5"](#)。

支持以下选项：

`-?`

`--help`

显示帮助信息。使用此选项时，不会对任何其他选项执行任何其他处理。

指定此选项时，子命令可有可无。

如果仅指定此选项而不使用子命令，将会显示此命令的子命令列表。

如果使用子命令且指定此选项，将会显示该子命令的用法选项。

`-d directory[,...]`

`--directory=directory-[,...]`

`--directory directory-[,...]`

`-d project[,...]`

`--directory=project-[,...]`

`--directory project-[,...]`

指定 `sun_uss` NAS 设备的项目。对于 `sun_uss` NAS 设备，必须在创建文件系统之前在 NAS 设备上创建项目。项目名称不得以 `/` 开头。文件系统必须在项目内部创建。`project` 是一个 `sun_uss` NAS 设备术语，您可以在一个项目内创建所需数量的文件系统。仅可将此选项与 `add-dir`、`remove-dir`、`export` 和 `show` 子命令一起使用。

此选项接受特殊关键字 `all`。如果使用 `-d all` 选项，则会指定位于指定 NAS 设备上的所有目录。

- 使用 `remove-dir` 子命令，可删除指定设备上的所有目录。
- 使用 `export` 子命令，可将指定设备上的所有目录的配置信息显示到指定输出。
- 使用 `add-dir` 子命令和 `-i configfile` 选项，可添加配置文件中列出的指定 NAS 设备上的所有目录。

-
- 如果针对 sun_uss NAS 设备将 show 和 find-dir 子命令与 -v 选项一起使用，将显示 -d 选项中的指定项目内包含的文件系统。可以使用 all 关键字显示 NAS 设备的所有项目，或者仅显示单个项目。

-F
--force

强制删除指定的 NAS 设备。

该强制选项仅可与 remove 子命令一起使用。将此强制选项与 remove 子命令一起使用时，可从 Oracle Solaris Cluster 配置中删除 NAS 设备及其已配置目录。

-f *passwd-file*
--passwdfile=*passwd-file*
--passwdfile *passwd-file*

指定包含登录到 NAS 设备时要使用的密码的密码文件。

为安全起见，无法在命令行选项中指定密码。为了确保密码安全，请将其置于文本文件中并通过使用 -f 选项指定该文件。如果未指定密码的输入文件，此命令会提示您提供密码。

将输入文件的权限设置为可由 root 用户读取并禁止组访问或全局访问。

将 clnasdevice add 与 -i 选项一起使用时，如果您的 *clconfigfile* 不包含密码，则必须使用 -f *passwdfile* 选项。

在输入文件中，请遵守以下限制：

- 在单独一行中指定密码。不要跨多行键入密码。
- 忽略前导空格和制表符。
- 注释以未加引号的 # 符号开头。可在下一新行中继续输入注释。
解析器会忽略所有注释。
- 如果使用输入文件提供设备用户密码，则密码中不能包含 # 符号。

-i *clconfigfile*
--input={- | *clconfigfile*}
--input {- | *clconfigfile*}

指定用于创建或修改 NAS 设备的配置信息。此信息必须符合 [clconfiguration\(5CL\) \[1245\]](#) 手册页中定义的格式。此信息可包含在文件中或者通过标准输入 (stdin) 提供。要指定标准输入，请指定 - 而不是文件名。

如果在命令行和 *clconfigfile* 文件中指定了相同的属性，则优先使用在命令行上设置的属性。

将 clnasdevice add 与 -i 选项一起使用时，必须使用 -f *passwdfile* 选项。

```
-o {- | clconfigfile}
--output={- | clconfigfile}
--output {- | clconfigfile}
```

使用 [clconfiguration\(5CL\) \[1245\]](#) 手册页中定义的格式写入 NAS 设备配置信息。可以将此信息写入到文件或标准输出 (stdout)。要指定标准输出，请指定 - 而不是文件名。

```
-p name=value [,...]  
--property=name=value [,...]  
--property name value [,...]
```

指定特定于 NAS 设备类型的属性。

在使用 add 子命令将新的 NAS 设备添加到群集配置中时，必须指定此选项。在使用 set 子命令修改 NAS 设备的属性时，也必须指定此选项。有关更多信息，请参见 -t 选项的描述。

```
-t nas-device-type  
--type=nas-device-type  
--type nas-device-type
```

指定 NAS 设备类型。将 NAS 设备添加到 Oracle Solaris Cluster 配置中时，必须指定此选项。NAS 设备类型通过供应商名称进行标识。例如，OracleZFS Storage Appliance NAS 设备类型为 sun_uss。

不同类型的 NAS 设备具有不同的属性，有时不具有任何属性。

sun_uss	指定 OracleZFS Storage Appliance NAS 设备。 -p <i>userid=osc_agent</i> [-f <i>passwd-file</i>] 或 -u <i>userid</i> [-f <i>passwdfile</i>] <i>userid</i> 必须为 <i>osc_agent</i> 。在使用 sun_uss 之前，必须下载客户机代码并在所有群集节点上加以安装。此 <i>osc_agent</i> <i>userid</i> 是通过在设备上运行某个工作流创建的。在使用以 <i>userid</i> 作为输入的 <i>clnasdevice</i> 子命令之前，必须已在设备上创建该 <i>userid</i> 。 <i>userid</i> 和 <i>password</i> 属性为必需属性。 -p " <i>nodeIPs{node}</i> "= <i>[IP]</i> 此属性可为每个节点指定一个 IP。如果要使用群集节点名称 IP 以外的 IP 来访问 NAS 设备，可以使用 <i>nodeIPsnode</i> 属性指定该 IP。此属性是可选的。如果不指定 IP，系统会使用群集节点名称的 IP。这些 IP 必须与 NAS 设备上项目的 NFS Access Mode 中指定的 IP 相匹配。 如果不指定属性值（例如，-p " <i>nodeIPs{node}</i> "=），则会从群集配置中删除指定节点的 IP，并且系统会使用群集节点名称的 IP。
---------	---

在添加 sun_uss NAS 设备及其项目之前，必须执行必要的设置。设置任务包括下载客户机代码并在群集节点上加以安装。运行 Configure for Oracle Solaris Cluster NFS 工作流以在设备上创建 `userid osc_agent` 及其密码。创建其 Share Mode 为无或只读（支持读写模式，但不建议使用）的项目。NFS Access Mode 必须使用“网络”概念并授予对群集节点 IP 的读写访问权。

必须先执行以下任务，然后再将 NAS 设备及其导出目录添加到群集配置：

- 设置 NAS 设备。
- 设置目录并使其可用于群集节点。
- 确定在设备上执行管理任务要使用的用户 ID 及密码。

NAS 设备还必须已启动且正在运行。有关更多信息，请参见 [Unresolved link to "Oracle Solaris Cluster With Network-Attached Storage Device Manual"](#)。

`-u userid`

`--userid=userid`

`--userid userid`

指定用于登录到 NAS 设备的用户 ID。

群集需要知道用户 ID 以登录到设备并执行管理职责。

或者，可以使用 `-p` 选项指定用户 ID。有关详细信息，请参见 `-p`。

此选项只能与 `add` 和 `set` 子命令一起使用。

`-V`

`--version`

显示该命令的版本。

指定此选项时，不要将其与子命令、操作数或其他选项结合使用。子命令、操作数或其他选项将被忽略。将显示该命令的版本。不会执行任何其他处理。

`-v`

`--verbose`

将详细信息显示到标准输出 (stdout)。

`-Z {zoneclustername | global | all}`

`--zoneclustername={zoneclustername | global | all}`

指定已注册 *nas-device-type* 且要对其执行操作的群集。

除 `export` 命令以外的所有子命令均支持此选项。

如果指定此选项，还必须指定以下参数之一：

zoneclustername 指定使用此选项的命令仅在名为 *zoneclustername* 的区域群集中的所有指定 *nas-device-type* 上进行操作。

global	指定使用此选项的命令仅在全局群集中的所有指定 <i>nas-device-type</i> 上进行操作。
all	<p>如果在全局群集中使用此参数，则会指定使用此参数的命令在所有群集（包括全局群集和所有区域群集）中的所有指定 <i>nas-device-type</i> 上进行操作。</p> <p>如果在某区域群集中使用此参数，则会指定使用此参数的命令仅在该区域群集中的所有指定 <i>nas-device-type</i> 上进行操作。</p>

支持以下操作数：

nasdevice

NAS 设备的名称。NAS 设备名称是 NAS 设备通过网络进行通信所使用的主机名。群集需要 NAS 设备的 NAS 主机名来与 NAS 设备通信。如果子命令接受多个 NAS 设备，则可以使用加号 (+) 指定所有 NAS 设备。对于 `add` 和 `add-dir` 子命令，加号操作对象指示指定配置文件中的所有 NAS 设备。

如果命令对于所有指定的操作数均成功执行，则返回零 (CL_NOERR)。如果某个操作数出错，命令会继续处理操作数列表中的下一个操作数。返回的退出代码始终反映最先发生的错误。

可能会返回以下退出值：

0 CL_NOERR

没有错误
您发出的命令已成功完成。

1 CL_ENOMEM

交换空间不足
某个群集节点耗尽了交换内存或其他操作系统资源。

3 CL_EINVAL

参数无效
您键入的命令不正确，或者使用 `-i` 选项提供的群集配置信息语法不正确。

6 CL_EACCESS

权限被拒绝
指定的对象不可访问。要发出命令，可能需要拥有超级用户或 RBAC 访问权限。有关更多信息，请参见 [Unresolved link to "su1M"](#) 和 [Unresolved link to "rbac5"](#) 手册页。

15 CL_EPROP

属性无效

您使用 -p、-y 或 -x 选项指定的属性或值不存在或不被允许。

18 CL_EINTERNAL

遇到内部错误

内部错误表示软件缺陷或其他缺陷。

36 CL_ENOENT

没有这样的对象

由于以下某个原因，找不到您所指定的对象：

- 该对象不存在。
- 您尝试使用 -o 选项创建的配置文件路径中的某个目录不存在。
- 您尝试使用 -i 选项访问的配置文件包含错误。

39 CL_EEXIST

对象已存在

您所指定的设备、设备组、群集互连组件、节点、群集、资源、资源类型、资源组或专用字符串已存在。

41 CL_ETYPE

类型无效

您使用 -t 或 -p 选项指定的类型不存在。

例 83 将 NAS 设备从 OracleZFS Storage Appliance 添加到群集或区域群集

以下 `clnasdevice` 命令可将 `sun_uss` 类型的 OracleZFS Storage Appliance `uss7110-01` 添加到配置中。

```
# clnasdevice add -t sun_uss -p userid=osc_agent -f passwd-file uss7110-01
```

例 84 向群集添加 NAS 项目

以下 `clnasdevice` 命令可将两个项目添加到已配置的 NAS 设备 `uss7110-01` 中。

```
# clnasdevice add-dir -d pool-0/local/nassa-p1,pool-0/local/nassa-p2 uss7110-01
```

例 85 从群集或区域群集中删除 NAS 设备

以下 `clnasdevice` 命令可从群集 `sun` 配置中删除 NAS 设备 `uss7110-01` 及其所有剩余项目（如果有）。

```
# clnasdevice remove -F uss7110-01
```

以下 `clnasdevice` 命令可从区域群集 `zC` 配置中删除 NAS 设备 `uss7110-01` 及其所有剩余项目（如果有）。

```
# clnasdevice remove -Z ZC -F uss7110-01
```

以下示例显示了如何更新 nodeIPs 属性。

```
# clnasdevice set -p "nodeIPs{cluster-1}"=10.155.55.145 \  
-p "nodeIPs{cluster-2}"=10.155.55.146 uss7110-01
```

以下示例从群集配置中删除当前 IP 设置，以便系统使用群集节点名称的 IP。

```
# clnasdevice set -p "nodeIPs{cluster-1}"= -p "nodeIPs{cluster-2}"= uss7110-01
```

例 86 显示尚未添加到群集的 NAS 设备项目

以下 clnasdevice 命令可显示尚未添加到群集的 NAS 设备项目名称。

```
# clnasdevice find-dir uss7110-01  
Nas Device:            uss7110-01  
Type:                 sun_uss  
Unconfigured Project: pool-0/local/nassa-p2  
Unconfigured Project: pool-0/local/nassa-p1
```

例 87 显示群集或区域群集中配置的 NAS 设备

以下 clnasdevice 命令可显示群集中配置的所有 NAS 设备的名称。要查看设备及其目录的列表，请使用详细选项或 show 子命令。

```
# clnasdevice list  
uss7110-01
```

以下 clnasdevice 命令可显示区域群集 ZC 中配置的所有 NAS 设备的名称。要查看设备及其目录的列表，请使用详细选项或 show 子命令。

```
# clnasdevice list -Z ZC  
ZC:uss7110-01
```

以下 clnasdevice 命令可显示区域群集 ZC 中配置的所有 NAS 设备的名称。要查看设备及其目录的列表，请使用详细选项或 show 子命令。

```
# clnasdevice list -Z all  
global:uss7110-01  
ZC:uss7110-01
```

例 88 显示 NAS 设备及其项目

以下 clnasdevice 命令可显示群集中配置的所有 NAS 设备的名称以及项目文件系统。

```
# clnasdevice show -v -d all uss7110-01  
Nas Device:            uss7110-01  
Type:                 sun_uss  
Project:               pool-0/local/nassa-p1
```

```

File System: /export/nassa-p1/nassa-p1-fs1
File System: /export/nassa-p1/nassa-p1-fs2
File System: /export/nassa-p1/nassa-p1-fs3
Project: pool-0/local/nassa-p2
File System: /export/nassa-p2/nassa-p2-fs1
File System: /export/nassa-p2/nassa-p2-fs2

```

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

[Intro\(1CL\) \[17\]](#)、[cluster\(1CL\) \[461\]](#)

[Unresolved link to "Oracle Solaris Cluster With Network-Attached Storage Device Manual"](#)

超级用户可以运行此命令的所有格式。

任何用户都可以运行带有以下子命令和选项的此命令：

- -? 选项
- -V 选项

要运行带有其他子命令的此命令，超级用户以外的用户需要具有 RBAC 授权。请参见下表。

子命令	RBAC 授权
add	solaris.cluster.modify
add-dir	solaris.cluster.modify
export	solaris.cluster.read
find-dir	solaris.cluster.read
list	solaris.cluster.read
set	solaris.cluster.modify
remove	solaris.cluster.modify
remove-dir	solaris.cluster.modify
show	solaris.cluster.read

名称

clnasdevice, clnas — 为 Oracle Solaris Cluster 管理对 NAS 设备的访问权限

```
/usr/cluster/bin/clnasdevice -V

/usr/cluster/bin/clnasdevice [subcommand] -?

/usr/cluster/bin/clnasdevice subcommand [options] -v [nasdevice[...]]

/usr/cluster/bin/clnasdevice add -t type {-p name=value
[,...] | -u userid} [-f passwdfile] [-Z {zoneclustername | global}]
nasdevice

/usr/cluster/bin/clnasdevice add -i {- | clconfigfile}[-t type]
[-p name=value | -u userid] {-f passwdfile} [-Z {zoneclustername | global}]
{nasdevice

/usr/cluster/bin/clnasdevice add-dir -d directory[,...] [-Z
{zoneclustername | global}] nasdevice

/usr/cluster/bin/clnasdevice add-dir -i {- | clconfigfile} [-d all |
directory [,...]] [-f passwordfile] [-Z {zoneclustername | global}]
{nasdevice

/usr/cluster/bin/clnasdevice export [-o {- | clconfigfile}] [-t
type[,...]] [-d all | directory[,...]] [+ | nasdevice[...]]

/usr/cluster/bin/clnasdevice find-dir [-d {all |directory[,...]}
[-t type[,...]] [-Z {zoneclustername[,...] | global | all}]
[+ | nasdevice[...]]

/usr/cluster/bin/clnasdevice list [-t type[,...]] [-Z
{zoneclustername[,...] | global | all}] [+ | nasdevice[...]]

/usr/cluster/bin/clnasdevice remove [-t type[,...]] [-Z
{zoneclustername | global}] [-F ] [+ | nasdevice[...]]

/usr/cluster/bin/clnasdevice remove-dir -d all | directory[,...]
[-Z {zoneclustername | global}] nasdevice

/usr/cluster/bin/clnasdevice set {-p name=value[,...] | -u
userid} [-f passwdfile] [-Z {zoneclustername | global}]
nasdevice

/usr/cluster/bin/clnasdevice show [-d {all | directory[,...]} [-t
type[,...]] [-Z {zoneclustername[,...] | global | all}]
[+ | nasdevice[...]]
```

clnasdevice 命令用于管理 NAS 设备及其目录或项目的 Oracle Solaris Cluster 配置信息。

`clnas` 命令是 `clnasdevice` 命令的简短格式。`clnas` 和 `clnasdevice` 命令是相同的。您可以使用此命令的任一种格式。

此命令的一般格式如下所示：

```
clnasdevice [subcommand] [options] [operands]
```

仅当 *options* 指定 `-?` 选项或 `-v` 选项时，才可以省略 *subcommand*。

此命令的每个选项都有长格式和简短格式两种格式。在本手册页“选项”部分给出了两种格式的选项，以及对该选项的描述。

在群集中使用 `clnasdevice` 命令配置 NAS 设备之前，您的 NAS 设备必须符合以下条件：

- NAS 设备必须已设置且正常工作。
- NAS 设备必须已引导且正在运行。
- NAS 设备的目录必须已创建且可用于群集节点。
- 如果 NAS 设备将用作法定设备，则必须创建该法定设备的 LUN。有关配置 NAS 法定设备的信息，请参见 `clquorum` 手册页。

在将设备配置到群集之前，您可能还需要执行其他任务，具体取决于 NAS 设备供应商。有关这些任务的详细信息，请参见“选项”部分的 `-t` 选项。有关设置 NAS 设备和导出目录的过程，请参阅特定 NAS 设备的文档。

在 NAS 设备完全可操作且准备好为群集提供存储之后，请使用 `clnasdevice` 命令管理群集中的 NAS 设备配置信息。否则，群集将无法检测到 NAS 设备及其导出目录。因此，群集将无法保护这些目录中信息的完整性。

使用 `clnasdevice` 命令执行以下管理任务：

- 创建 NAS 设备配置
- 更新特定于 NAS 类型的属性
- 从群集配置中删除 NAS 设备的目录
- 从群集配置中删除 NAS 设备

只能在活动群集节点上运行 `clnasdevice` 命令。无论您在哪个节点上运行该命令，运行结果始终相同。

在区域群集中，可以将 `clnasdevice` 命令与所有子命令（`export` 除外）一起使用。也可以将 `-z` 选项与所有子命令（`export` 除外）一起使用来指定要将操作限制到的特定区域群集的名称。

您可以从全局群集节点访问所有区域群集信息，但无法从特定区域群集访问其他区域群集信息。如果不将操作限制到特定区域群集，则所用的子命令仅会在当前群集中运行。

支持以下子命令：

add

将 NAS 设备添加到 Oracle Solaris Cluster 配置中。

使用 `-t` 选项可指定 NAS 设备的供应商。有关详细信息，请参见“选项”部分的 `-t` 选项描述。

可能还需要设置其他属性，具体取决于您的 NAS 设备类型。“选项”部分的 `-t` 选项描述中也介绍了这些必需属性。

超级用户以外的用户需要具有 `solaris.cluster.modify` 基于角色的访问控制 (role-based access control, RBAC) 授权才能使用该命令。请参见 [Unresolved link to "rbac5"](#)。

另请参见 `remove` 子命令的描述。

add-dir

将已配置的 NAS 设备的指定目录或项目添加到群集配置中。在使用该子命令之前，必须已在设备上创建这些目录或项目并使其可用于群集。有关创建目录或项目的信息，请参见您的 NAS 设备的文档。

可以使用下列其中一种方法添加 NAS 设备目录或项目：

- 使用 `clnasdevice add` 命令在群集中配置 NAS 设备。然后使用 `clnasdevice add-dir` 命令在群集中配置该设备的目录或项目。
- 使用 `clnasdevice add-dir -i configurationfile` 格式的命令可在单个步骤中同时添加设备并配置设备目录或项目。要使用此方法添加目录或项目，请使用 `-f` 选项提供密码文件。有关该选项的详细信息，请参见“选项”部分。有关更多信息，请参见 [clconfiguration\(5CL\) \[1245\]](#) 手册页。

只要在 NAS 设备上创建新目录或项目并使其可用于群集节点，就需要使用此 `add-dir` 子命令将目录或项目添加到群集配置。可以使用 `find-dir` 子命令列出可使用 `add-dir` 子命令添加到群集的可用目录或项目。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该命令。请参见 [Unresolved link to "rbac5"](#)。

另请参见 `remove-dir` 子命令的描述。

export

导出群集 NAS 设备的配置信息。如果使用 `-o` 选项指定一个文件，则会将配置信息写入该文件。如果不使用 `-o` 选项，则会将输出写入标准输出 (stdout)。

`export` 子命令不会修改群集配置信息。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该命令。请参见 [Unresolved link to "rbac5"](#)。

find-dir

显示群集可能会使用的 NAS 设备上设置的 `sun_uss` 项目。尚未使用 `add-dir` 子命令将这些项目添加到群集配置中。输出中列出的这些项目可以在您使用 `add-dir` 子命令时充当 `-d` 选项的候选项。

要显示特定类型的 NAS 设备，请指定 `-t` 选项。

要显示 `sun_uss` 项目和这些项目内的文件系统，请指定 `-v` 选项。

要显示特定的 `sun_uss` NAS 设备项目，请指定 `-d` 选项。

要显示特定 `sun_uss` NAS 设备项目和这些项目内的文件系统，请指定 `-v` 和 `-d` 选项。

`find-dir` 子命令不会修改群集配置信息。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该命令。请参见 [Unresolved link to "rbac5"](#)。

list

显示群集中配置的 NAS 设备。

要显示群集中配置的设备的目录及设备类型，请使用详细选项 `-v`。

要显示特定类型的 NAS 设备，请使用 `-t` 选项。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该命令。请参见 [Unresolved link to "rbac5"](#)。

remove

从 Oracle Solaris Cluster 配置中删除指定的一个或多个 NAS 设备。

如果不指定强制选项 `-F`，则必须已通过使用 `remove-dir` 子命令从配置中删除 NAS 设备目录。

如果指定强制选项 `-F`，该命令将从群集配置中删除 NAS 设备及其目录。请参见 `-OPTIONS` 中的 `F`。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该命令。请参见 [Unresolved link to "rbac5"](#)。

另请参见 `add` 子命令的描述。

remove-dir

从 Oracle Solaris Cluster 配置中删除指定的 NAS 目录或项目。

`remove-dir` 子命令用于删除 `-d` 选项指定的导出目录或项目。在使用 `-d all` 时，该子命令可删除指定 NAS 设备的所有目录或项目。

只要从 NAS 设备中删除目录或项目，都需要使用此 `remove-dir` 子命令从群集配置中删除目录或项目。群集配置中的 NAS 目录或项目必须与从 NAS 设备导出的现有目录或项目相匹配。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该命令。请参见 [Unresolved link to "rbac5"](#)。

另请参见 `add-dir` 子命令的描述。

set

设置特定 NAS 设备的指定属性。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该命令。请参见 [Unresolved link to "rbac5"](#)。

show

在未提供任何选项时，显示以下信息：

- Oracle Solaris Cluster 中配置的所有当前 NAS 设备的列表
- 每个 NAS 设备的可用目录
- 与每个 NAS 设备关联的所有属性

要显示特定类型的 NAS 设备，请指定 `-t` 选项。要显示有关特定设备的信息，请传送 NAS 设备的主机名用作该命令的操作对象。

要显示指定项目中包含的文件系统，请将 `-d` 和 `-v` 选项与 `show` 子命令一起使用。可以使用 `all` 关键字显示 NAS 设备的所有项目，或者仅显示单个项目。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该命令。请参见 [Unresolved link to "rbac5"](#)。

支持以下选项：

`-?`

`--help`

显示帮助信息。使用此选项时，不会对任何其他选项执行任何其他处理。

指定此选项时，子命令可有可无。

如果仅指定此选项而不使用子命令，将会显示此命令的子命令列表。

如果使用子命令且指定此选项，将会显示该子命令的用法选项。

`-d directory[,...]`

`--directory=directory-[,...]`

`--directory directory-[,...]`

`-d project[,...]`

`--directory=project-[,...]`

`--directory project-[,...]`

指定 `sun_uss` NAS 设备的项目。对于 `sun_uss` NAS 设备，必须在创建文件系统之前在 NAS 设备上创建项目。项目名称不得以 `/` 开头。文件系统必须在项目内部创建。`project` 是一个 `sun_uss` NAS 设备术语，您可以在一个项目内创建所需数量的文件系统。仅可将此选项与 `add-dir`、`remove-dir`、`export` 和 `show` 子命令一起使用。

此选项接受特殊关键字 `all`。如果使用 `-d all` 选项，则会指定位于指定 NAS 设备上的所有目录。

- 使用 `remove-dir` 子命令，可删除指定设备上的所有目录。
- 使用 `export` 子命令，可将指定设备上的所有目录的配置信息显示到指定输出。
- 使用 `add-dir` 子命令和 `-i configfile` 选项，可添加配置文件中列出的指定 NAS 设备上的所有目录。

-
- 如果针对 sun_uss NAS 设备将 show 和 find-dir 子命令与 -v 选项一起使用，将显示 -d 选项中的指定项目内包含的文件系统。可以使用 all 关键字显示 NAS 设备的所有项目，或者仅显示单个项目。

-F
--force

强制删除指定的 NAS 设备。

该强制选项仅可与 remove 子命令一起使用。将此强制选项与 remove 子命令一起使用时，可从 Oracle Solaris Cluster 配置中删除 NAS 设备及其已配置目录。

-f *passwd-file*
--passwdfile=*passwd-file*
--passwdfile *passwd-file*

指定包含登录到 NAS 设备时要使用的密码的密码文件。

为安全起见，无法在命令行选项中指定密码。为了确保密码安全，请将其置于文本文件中并通过使用 -f 选项指定该文件。如果未指定密码的输入文件，此命令会提示您提供密码。

将输入文件的权限设置为可由 root 用户读取并禁止组访问或全局访问。

将 `clnasdevice add` 与 -i 选项一起使用时，如果您的 *clconfigfile* 不包含密码，则必须使用 -f *passwdfile* 选项。

在输入文件中，请遵守以下限制：

- 在单独一行中指定密码。不要跨多行键入密码。
- 忽略前导空格和制表符。
- 注释以未加引号的 # 符号开头。可在下一新行中继续输入注释。
解析器会忽略所有注释。
- 如果使用输入文件提供设备用户密码，则密码中不能包含 # 符号。

-i *clconfigfile*
--input={- | *clconfigfile*}
--input {- | *clconfigfile*}

指定用于创建或修改 NAS 设备的配置信息。此信息必须符合 [clconfiguration\(5CL\) \[1245\]](#) 手册页中定义的格式。此信息可包含在文件中或者通过标准输入 (stdin) 提供。要指定标准输入，请指定 - 而不是文件名。

如果在命令行和 *clconfigfile* 文件中指定了相同的属性，则优先使用在命令行上设置的属性。

将 `clnasdevice add` 与 -i 选项一起使用时，必须使用 -f *passwdfile* 选项。

```
-o {- | clconfigfile}
--output={- | clconfigfile}
--output {- | clconfigfile}
```

使用 [clconfiguration\(5CL\) \[1245\]](#) 手册页中定义的格式写入 NAS 设备配置信息。可以将此信息写入到文件或标准输出 (stdout)。要指定标准输出，请指定 - 而不是文件名。

```
-p name=value [,...]  
--property=name=value [,...]  
--property name value [,...]
```

指定特定于 NAS 设备类型的属性。

在使用 add 子命令将新的 NAS 设备添加到群集配置中时，必须指定此选项。在使用 set 子命令修改 NAS 设备的属性时，也必须指定此选项。有关更多信息，请参见 -t 选项的描述。

```
-t nas-device-type  
--type=nas-device-type  
--type nas-device-type
```

指定 NAS 设备类型。将 NAS 设备添加到 Oracle Solaris Cluster 配置中时，必须指定此选项。NAS 设备类型通过供应商名称进行标识。例如，OracleZFS Storage Appliance NAS 设备类型为 sun_uss。

不同类型的 NAS 设备具有不同的属性，有时不具有任何属性。

sun_uss	指定 OracleZFS Storage Appliance NAS 设备。 -p <i>userid=osc_agent</i> [-f <i>passwd-file</i>] 或 -u <i>userid</i> [-f <i>passwdfile</i>] <i>userid</i> 必须为 <i>osc_agent</i> 。在使用 sun_uss 之前，必须下载客户机代码并在所有群集节点上加以安装。此 <i>osc_agent</i> <i>userid</i> 是通过在设备上运行某个工作流创建的。在使用以 <i>userid</i> 作为输入的 <i>clnasdevice</i> 子命令之前，必须已在设备上创建该 <i>userid</i> 。 <i>userid</i> 和 <i>password</i> 属性为必需属性。 -p " <i>nodeIPs{node}</i> "= <i>[IP]</i> 此属性可为每个节点指定一个 IP。如果要使用群集节点名称 IP 以外的 IP 来访问 NAS 设备，可以使用 <i>nodeIPsnode</i> 属性指定该 IP。此属性是可选的。如果不指定 IP，系统会使用群集节点名称的 IP。这些 IP 必须与 NAS 设备上项目的 NFS Access Mode 中指定的 IP 相匹配。 如果不指定属性值（例如，-p " <i>nodeIPs{node}</i> "=），则会从群集配置中删除指定节点的 IP，并且系统会使用群集节点名称的 IP。
---------	---

在添加 sun_uss NAS 设备及其项目之前，必须执行必要的设置。设置任务包括下载客户机代码并在群集节点上加以安装。运行 Configure for Oracle Solaris Cluster NFS 工作流以在设备上创建 `userid osc_agent` 及其密码。创建其 Share Mode 为无或只读（支持读写模式，但不建议使用）的项目。NFS Access Mode 必须使用“网络”概念并授予对群集节点 IP 的读写访问权。

必须先执行以下任务，然后再将 NAS 设备及其导出目录添加到群集配置：

- 设置 NAS 设备。
- 设置目录并使其可用于群集节点。
- 确定在设备上执行管理任务要使用的用户 ID 及密码。

NAS 设备还必须已启动且正在运行。有关更多信息，请参见 [Unresolved link to "Oracle Solaris Cluster With Network-Attached Storage Device Manual"](#)。

`-u userid`

`--userid=userid`

`--userid userid`

指定用于登录到 NAS 设备的用户 ID。

群集需要知道用户 ID 以登录到设备并执行管理职责。

或者，可以使用 `-p` 选项指定用户 ID。有关详细信息，请参见 `-p`。

此选项只能与 `add` 和 `set` 子命令一起使用。

`-V`

`--version`

显示该命令的版本。

指定此选项时，不要将其与子命令、操作数或其他选项结合使用。子命令、操作数或其他选项将被忽略。将显示该命令的版本。不会执行任何其他处理。

`-v`

`--verbose`

将详细信息显示到标准输出 (stdout)。

`-Z {zoneclustername | global | all}`

`--zoneclustername={zoneclustername | global | all}`

指定已注册 *nas-device-type* 且要对其执行操作的群集。

除 `export` 命令以外的所有子命令均支持此选项。

如果指定此选项，还必须指定以下参数之一：

zoneclustername 指定使用此选项的命令仅在名为 *zoneclustername* 的区域群集中的所有指定 *nas-device-type* 上进行操作。

global	指定使用此选项的命令仅在全局群集中的所有指定 <i>nas-device-type</i> 上进行操作。
all	<p>如果在全局群集中使用此参数，则会指定使用此参数的命令在所有群集（包括全局群集和所有区域群集）中的所有指定 <i>nas-device-type</i> 上进行操作。</p> <p>如果在某区域群集中使用此参数，则会指定使用此参数的命令仅在该区域群集中的所有指定 <i>nas-device-type</i> 上进行操作。</p>

支持以下操作数：

nasdevice

NAS 设备的名称。NAS 设备名称是 NAS 设备通过网络进行通信所使用的主机名。群集需要 NAS 设备的 NAS 主机名来与 NAS 设备通信。如果子命令接受多个 NAS 设备，则可以使用加号 (+) 指定所有 NAS 设备。对于 `add` 和 `add-dir` 子命令，加号操作对象指示指定配置文件中的所有 NAS 设备。

如果命令对于所有指定的操作数均成功执行，则返回零 (CL_NOERR)。如果某个操作数出错，命令会继续处理操作数列表中的下一个操作数。返回的退出代码始终反映最先发生的错误。

可能会返回以下退出值：

0 CL_NOERR

没有错误
您发出的命令已成功完成。

1 CL_ENOMEM

交换空间不足
某个群集节点耗尽了交换内存或其他操作系统资源。

3 CL_EINVAL

参数无效
您键入的命令不正确，或者使用 `-i` 选项提供的群集配置信息语法不正确。

6 CL_EACCESS

权限被拒绝
指定的对象不可访问。要发出命令，可能需要拥有超级用户或 RBAC 访问权限。有关更多信息，请参见 [Unresolved link to "su1M"](#) 和 [Unresolved link to "rbac5"](#) 手册页。

15 CL_EPROP

属性无效

您使用 -p、-y 或 -x 选项指定的属性或值不存在或不被允许。

18 CL_EINTERNAL

遇到内部错误

内部错误表示软件缺陷或其他缺陷。

36 CL_ENOENT

没有这样的对象

由于以下某个原因，找不到您所指定的对象：

- 该对象不存在。
- 您尝试使用 -o 选项创建的配置文件路径中的某个目录不存在。
- 您尝试使用 -i 选项访问的配置文件包含错误。

39 CL_EEXIST

对象已存在

您所指定的设备、设备组、群集互连组件、节点、群集、资源、资源类型、资源组或专用字符串已存在。

41 CL_ETYPE

类型无效

您使用 -t 或 -p 选项指定的类型不存在。

例 89 将 NAS 设备从 OracleZFS Storage Appliance 添加到群集或区域群集

以下 `clnasdevice` 命令可将 `sun_uss` 类型的 OracleZFS Storage Appliance `uss7110-01` 添加到配置中。

```
# clnasdevice add -t sun_uss -p userid=osc_agent -f passwd-file uss7110-01
```

例 90 向群集添加 NAS 项目

以下 `clnasdevice` 命令可将两个项目添加到已配置的 NAS 设备 `uss7110-01` 中。

```
# clnasdevice add-dir -d pool-0/local/nassa-p1,pool-0/local/nassa-p2 uss7110-01
```

例 91 从群集或区域群集中删除 NAS 设备

以下 `clnasdevice` 命令可从群集 `sun` 配置中删除 NAS 设备 `uss7110-01` 及其所有剩余项目（如果有）。

```
# clnasdevice remove -F uss7110-01
```

以下 `clnasdevice` 命令可从区域群集 `zC` 配置中删除 NAS 设备 `uss7110-01` 及其所有剩余项目（如果有）。

```
# clnasdevice remove -Z ZC -F uss7110-01
```

以下示例显示了如何更新 nodeIPs 属性。

```
# clnasdevice set -p "nodeIPs{cluster-1}"=10.155.55.145 \  
-p "nodeIPs{cluster-2}"=10.155.55.146 uss7110-01
```

以下示例从群集配置中删除当前 IP 设置，以便系统使用群集节点名称的 IP。

```
# clnasdevice set -p "nodeIPs{cluster-1}"= -p "nodeIPs{cluster-2}"= uss7110-01
```

例 92 显示尚未添加到群集的 NAS 设备项目

以下 clnasdevice 命令可显示尚未添加到群集的 NAS 设备项目名称。

```
# clnasdevice find-dir uss7110-01  
Nas Device:            uss7110-01  
Type:                 sun_uss  
Unconfigured Project: pool-0/local/nassa-p2  
Unconfigured Project: pool-0/local/nassa-p1
```

例 93 显示群集或区域群集中配置的 NAS 设备

以下 clnasdevice 命令可显示群集中配置的所有 NAS 设备的名称。要查看设备及其目录的列表，请使用详细选项或 show 子命令。

```
# clnasdevice list  
uss7110-01
```

以下 clnasdevice 命令可显示区域群集 ZC 中配置的所有 NAS 设备的名称。要查看设备及其目录的列表，请使用详细选项或 show 子命令。

```
# clnasdevice list -Z ZC  
ZC:uss7110-01
```

以下 clnasdevice 命令可显示区域群集 ZC 中配置的所有 NAS 设备的名称。要查看设备及其目录的列表，请使用详细选项或 show 子命令。

```
# clnasdevice list -Z all  
global:uss7110-01  
ZC:uss7110-01
```

例 94 显示 NAS 设备及其项目

以下 clnasdevice 命令可显示群集中配置的所有 NAS 设备的名称以及项目文件系统。

```
# clnasdevice show -v -d all uss7110-01  
Nas Device:            uss7110-01  
Type:                 sun_uss  
Project:               pool-0/local/nassa-p1
```

```

File System: /export/nassa-p1/nassa-p1-fs1
File System: /export/nassa-p1/nassa-p1-fs2
File System: /export/nassa-p1/nassa-p1-fs3
Project: pool-0/local/nassa-p2
File System: /export/nassa-p2/nassa-p2-fs1
File System: /export/nassa-p2/nassa-p2-fs2

```

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

[Intro\(1CL\) \[17\]](#)、[cluster\(1CL\) \[461\]](#)

[Unresolved link to "Oracle Solaris Cluster With Network-Attached Storage Device Manual"](#)

超级用户可以运行此命令的所有格式。

任何用户都可以运行带有以下子命令和选项的此命令：

- -? 选项
- -V 选项

要运行带有其他子命令的此命令，超级用户以外的用户需要具有 RBAC 授权。请参见下表。

子命令	RBAC 授权
add	solaris.cluster.modify
add-dir	solaris.cluster.modify
export	solaris.cluster.read
find-dir	solaris.cluster.read
list	solaris.cluster.read
set	solaris.cluster.modify
remove	solaris.cluster.modify
remove-dir	solaris.cluster.modify
show	solaris.cluster.read

名称

clnode — 管理 Oracle Solaris Cluster 节点

```
/usr/cluster/bin/clnode -V
/usr/cluster/bin/clnode [subcommand] -?
/usr/cluster/bin/clnode subcommand [options] -v [node ...]
/usr/cluster/bin/clnode add -n sponsornode[-i {- | clconfigfile}] -c clustername] [-G globaldevfs][-e endpoint,endpoint] node
/usr/cluster/bin/clnode create-loadlimit -p limitname=value[-p softlimit=value] [-p hardlimit=value] {+ | node[:zone] ...}
/usr/cluster/bin/clnode clear [-F] node...
/usr/cluster/bin/clnode delete-loadlimit -p limitname=value {+ | node[:zone] ...}
/usr/cluster/bin/clnode evacuate [-T seconds] {+ | node ...}
/usr/cluster/bin/clnode export [-o {- | clconfigfile}][+ | node ...]
/usr/cluster/bin/clnode list [-Z {zoneclustername | global | all}] {+ | node ...}
/usr/cluster/bin/clnode rename -n newnodename[node]
/usr/cluster/bin/clnode remove [-n sponsornode][-G globaldevfs] [-F] [node]
/usr/cluster/bin/clnode set [-p name=value] [...] {+ | node ...}
/usr/cluster/bin/clnode set-loadlimit -p limitname=value[-p softlimit=value] [-p hardlimit=value] {+ | node[:zone] ...}
/usr/cluster/bin/clnode show [-p name[,...]][-Z {zoneclustername | global | all}][+ | node ...]
/usr/cluster/bin/clnode show-rev [node]
/usr/cluster/bin/clnode status [-m][-Z {zoneclustername | global | all}][+ | node ...]
```

该命令执行以下操作：

- 向群集添加节点
- 从群集中删除节点
- 尝试切换所有资源组和设备组

-
- 修改节点的属性
 - 管理节点上的负载限制
 - 报告或导出一个或多个节点的状态和配置

`clnode` 命令的大多数子命令在群集模式中运行。您可以从群集的任何节点中运行大多数这些子命令。但是，`add` 和 `remove` 子命令例外。您必须在非群集模式下运行这些子命令。

当运行 `add` 和 `remove` 子命令时，您必须在要添加或删除的节点上运行它们。`clnode add` 命令还初始化节点本身，以便加入群集。`clnode remove` 命令还在已删除的节点上执行清除操作。

仅当 `options` 是 `-?` 选项或 `-v` 选项时，才可以省略 `subcommand`。

每个选项都有长和短两种格式。在 选项 中给出每个选项的两种格式以及该选项的描述。

`clnode` 命令没有短格式。

可以在区域群集中使用该命令的某些格式。有关在群集中有效使用该命令的更多信息，请参见各个子命令的描述。

子命令

支持以下子命令：

`add`

配置并向群集添加一个节点。

仅可以在全局区域中运行该子命令。只能在全局群集中使用该子命令。

您必须在非群集模式下运行该子命令。

要配置和添加节点，您必须使用 `-n sponsornode` 选项。该选项将现有的活动节点指定为监护人节点。当您在群集中配置节点时，始终需要监护人节点。

如果您未指定 `-c clustername`，该子命令将使用添加的第一个节点的名称作为新的群集名称。

操作数 `node` 是可选的。但是，如果您指定一个操作数，它必须是运行子命令的节点的主机名。

注 - 运行 `pkg install` 命令以安装 Oracle Solaris Cluster 软件。然后，运行 `scinstall` 实用程序以创建新群集或向现有群集添加节点。有关说明，请参见[Unresolved link to "Oracle Solaris Cluster 软件安装指南"](#)。

超级用户以外的用户需要具有 `solaris.cluster.modify` 基于角色的访问控制 (Role-Based Access Control, RBAC) 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

clear

运行 `remove` 子命令之后，清理或清除有关群集节点的任何其余信息。

仅可以在全局区域中运行该子命令。只能在全局群集中使用该子命令。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

create-loadlimit

在节点上添加负载限制。

您可以在全局区域中或区域群集中使用该子命令。

请参见 选项 中的 `-p` 选项。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

delete-loadlimit

在节点上删除现有负载限制。

您可以在全局区域中或区域群集中使用该子命令。

请参见 选项 中的 `-p` 选项。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

evacuate

尝试将所有资源组和设备组从指定节点切换到一组新的主节点。

您可以在全局区域或在区域群集节点中使用该子命令。

系统尝试基于每个组的配置首选项选择新的主节点。所有清除的资源组未必由同一主节点重新控制。如果一个或多个资源组或设备组无法从指定节点清除，则该子命令失败。如果该子命令失败，它将发出错误消息并以非零退出代码退出。如果该子命令无法将设备组的主所有权更改为其他节点，则原始节点将保留该设备组的主所有权。如果 RGM 无法在新的主节点上启动清除的资源组，则清除的资源组可能会最终脱机。

您可以将 `-t` 选项与该子命令一起使用，以指定用于阻止资源组切换回的秒数。如果未指定值，将使用默认值 60 秒。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

export

将节点配置信息导出到文件或标准输出 (`stdout`)。

仅可以在全局区域中运行该子命令。只能在全局群集中使用该子命令。

如果指定 `-o` 选项和文件名称，则会将配置信息写入该文件。

如果未提供 `-o` 选项和文件名称，则会将输出写入到标准输出。

该子命令不会修改群集配置数据。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

list

显示群集中配置的节点的名称。

如果使用该子命令指定 `-z` 选项，它将列出您指定的特定群集中的节点的名称，如下所示：

- 所有全局群集节点和区域群集节点
- 仅所有全局群集节点
- 仅指定名称的区域群集节点

您可以在全局群集中或区域群集中使用该子命令。

如果未指定 `node` 操作数，或指定加号操作数 (+)，该子命令将显示所有节点成员。

您必须在群集模式中运行该子命令。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令选项。请参见 [Unresolved link to "rbac5"](#) 手册页。

remove

从群集删除一个节点。

仅可以在全局区域中运行该子命令。只能在全局群集中使用该子命令。

您必须在非群集模式下运行该子命令。

要从群集中删除节点，请遵守以下准则。如果您未遵守这些准则，删除的节点可能会危害群集中的法定设备。

- 除非您还指定了 `-F` 选项，否则取消配置要从任何法定设备中删除的节点。
- 确保要删除的节点不是活动的群集成员。
- 除非至少配置了一个共享法定设备，否则不要从三节点群集中删除节点。

该子命令尝试从群集配置数据库中删除一部分对该节点的引用。如果指定 `-F` 选项，该子命令将尝试从群集配置数据库中删除所有对该节点的引用。

注 - 必须运行 `scinstall -r` 命令才能从节点中删除群集软件。有关更多信息，请参见 [Unresolved link to "Oracle Solaris Cluster 软件安装指南"](#)。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

rename

将节点重命名为新的节点名称。

仅可以在全局区域中运行该子命令。您必须在非群集模式中运行该子命令。

注 - 您必须在更改 Oracle Solaris 主机名的同一节点上运行该命令。

要将节点重命名为 *newnodename*，您必须使用 `-nnewnodename` 选项。必须从 *oldnodename* 中重命名当前活动的 Oracle Solaris 节点。群集中的所有节点必须处于非群集模式，该命令才可以成功运行。

操作数是可选的，它必须是运行子命令的节点的主机名。

注 - 在可以重命名节点之前，您必须首先运行 Oracle Solaris 主机名更改过程，才能在群集中重命名群集节点。有关说明，请参见 [Unresolved link to "Managing System Information, Processes, and Performance in Oracle Solaris 11.2 中的 How to Change a System's Identity"](#)。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

set

修改与您所指定的节点相关联的属性。

仅可以在全局区域中运行该子命令。只能在全局群集中使用该子命令。

请参见 选项 中的 `-p` 选项。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

set-loadlimit

在节点上修改现有负载限制。

您可以在全局区域中或区域群集中使用该子命令。

请参见 选项 中的 `-p` 选项。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

show

显示指定的一个或多个节点的配置或有关其属性的信息。

如果使用该子命令指定 `-z` 选项，它将显示您指定的一个或多个特定群集中的一个或多个节点的配置或属性信息，如下所示：

- 所有全局群集节点和区域群集节点
- 仅所有全局群集节点
- 仅指定名称的区域群集节点

仅可以在全局区域中运行该子命令。您可以在全局群集中或区域群集中使用该子命令。

如果未指定操作数，或指定加号 (+)，该子命令将显示所有群集节点的信息。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

show - rev

显示节点上安装的 Solaris Cluster 软件包的名称和发行版本信息。

只能在全局群集中使用该子命令。

您可以在非群集模式和群集模式中运行该子命令。如果在非群集模式中运行该命令，您只能指定运行该命令的节点的名称并获取有关该节点的信息。如果在群集模式中运行该命令，您可以指定群集中的任何节点并获取有关这些节点的信息。

您将该子命令与 -v 一起使用时，该子命令将显示软件包的名称、版本以及已应用于这些软件包的修补程序。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

status

显示您指定的一个或多个节点的状态，或者 Internet 协议网络多路径 (Internet Protocol (IP) network multipathing, IPMP) 组。

您可以在全局群集中或区域群集中使用该子命令。

如果未指定操作数，或指定加号 (+)，该子命令将显示所有群集节点的状态。节点的状态可以为 Online 或 Offline。

如果您在该子命令中指定 -m 选项，它将只显示 Oracle Solaris IPMP 组。

如果您在该子命令中指定详细选项 -v，它将显示群集节点和 Oracle Solaris IPMP 组的状态。

如果使用该子命令指定 -z 选项，它将显示您指定的一个或多个特定群集中的一个或多个节点的状态信息，如下所示：

- 所有全局群集节点和区域群集节点
- 仅所有全局群集节点
- 仅指定名称的区域群集节点

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

注 - 该部分显示了每个选项的短和长两种格式。

支持以下选项：

-?

--help

显示帮助信息。

指定此选项时，*subcommand* 可有可无。

如果没有指定 *subcommand*，将显示所有可用子命令的列表。

如果指定了 *subcommand*，将显示该子命令的用法。

如果指定此选项和其他选项，将会忽略其他选项。

`-c clustername`
`--clustername=clustername`
`--clustername clustername`

指定要添加节点的群集的名称。

仅将该选项与 `add` 子命令一起使用。

如果您指定该选项，则指定的 `clustername` 必须与现有群集的名称匹配。否则会出现错误。

`-e endpoint,endpoint`
`--endpoint=endpoint,endpoint`
`--endpoint endpoint,endpoint`

指定传输连接。

仅将该选项与 `add` 子命令一起使用。您可以指定该选项来建立群集传输拓扑。您可以通过配置连接适配器和交换机的电缆建立拓扑。您可以将适配器或交换机指定为端点。要指示电缆，您可以指定以逗号分隔的端点对。电缆可以建立从当前节点的群集传输适配器到以下任一位置的连接：

- 群集传输交换机的端口，也称为传输结点。
- 群集中已包含的其他节点的适配器。

如果您未指定 `-e` 选项，`add` 子命令将尝试配置默认电缆。但是，如果您在 `clnode` 命令的一个实例内配置多个传输适配器或交换机，则 `clnode` 无法构造默认电缆。默认情况下，将配置从单独配置的传输适配器到单独配置的或默认的传输交换机的电缆。

每当指定 `-e` 选项时，您必须始终指定以逗号分隔的两个端点。每对端点定义一个电缆。使用以下方法之一指定各个端点：

- 适配器端点：
`node:adapter`
- 交换机端点：
`switch[@ port]`

要指定标记 VLAN 适配器，请使用通过物理设备名称和 VLAN 实例号得出的标记 VLAN 适配器名称。VLAN 实例号等于 VLAN ID 与 1000 相乘再加上原始物理单元号。例如，物理设备 `net2` 上的 VLAN ID 11 转化为带标记的 VLAN 适配器名称 `net11002`。

如果您没有为交换机端点指定端口组件，则将分配一个默认端口。

`-F`
`--force`

强行删除或清除指定的节点，而不验证全局挂载是否保留在该节点上。

仅将该选项与 `clear` 或 `remove` 子命令一起使用。

```
-G {lofi | special | mount-point}
--globaldevfs={lofi | special | mount-point}
--globaldevfs {lofi | special | mount-point}
```

为全局设备挂载点指定 lofi 设备、原始 *special* 磁盘设备或专用文件系统。

仅将该选项与 add 或 remove 子命令一起使用。

每个群集节点必须具有全局挂载到 /global/.devices/node@nodeID 上的本地文件系统，然后该节点才可以成功作为群集成员参与。但是，在运行 clnode 命令之前，节点 ID 未知。默认情况下，clnode add 命令寻找挂载到 /globaldevices 或指定给 -G 选项的挂载点上的空文件系统。如果提供此类文件系统，则 clnode add 命令会对 /etc/vfstab 文件进行必要的更改。您指定的文件系统将在 /globaldevices 重新挂载。clnode 命令无法找到节点 ID 挂载时，该命令会尝试将条目添加到 vfstab 文件。请参见 [Unresolved link to " vfstab4" 手册页](#)。

如果未挂载 /global/.devices/node@nodeID 并且未提供空 /globaldevices 文件系统，则该命令失败。

如果指定 -G lofi，将创建 /.globaldevices 文件。lofi 设备与该文件相关联，并在 lofi 设备上创建全局设备文件系统。不会将任何 /global/.devices/node@nodeID 条目添加到 /etc/vfstab 文件。有关 lofi 设备的更多信息，请参见 [Unresolved link to " lofi7D" 手册页](#)。

如果指定原始 *special* 磁盘设备名称但未挂载 /global/.devices/node@nodeID，将使用 newfs 命令在设备上创建文件系统。不能为已挂载的文件系统提供设备名称。

根据准则，专用文件系统大小必须至少为 512 兆字节。如果该分区或文件系统不可用或不够大，您可能需要重新安装 Oracle Solaris OS。

对于在 lofi 设备上创建的名称空间，在根文件系统中需要 100 兆字节的空闲空间。

将该选项与 remove 子命令一起使用，以指定用于恢复以前的 /global/.devices 挂载点的新挂载点名称。

当与 remove 子命令一起使用时，如果在专用分区上挂载全局设备名称空间，该选项将指定用于恢复以前的 /global/.devices 挂载点的新挂载点名称。如果未指定 -G 选项且全局设备名称空间挂载在专用分区上，则默认将挂载点重命名为 /globaldevices。

```
-i {- | clconfigfile}
--input={- | clconfigfile}
--input {- | clconfigfile}
```

从文件或标准输入 (stdin) 中读取节点配置信息。[clconfiguration\(5CL\) \[1245\]](#) 手册页中描述了该配置信息的格式。

如果您使用该选项指定文件名称，则该选项将读取文件中的节点配置信息。如果您使用该选项指定 -，将会从标准输入 (stdin) 中读取配置信息。

```
-m
```

指定 IPMP 组。与 status 子命令一起使用，以仅显示 IPMP 组的状态。

```
-n newnodename
--newnodename=newnodename
--newnodename newnodename
```

指定新的节点名称。

该选项只可以与 `rename` 子命令一起使用。

您可以为当前节点指定新的节点名称。您使用 `rename` 子命令将节点重命名为 *newnodename* 时，必须已将当前节点主机名更改为 *newnodename*。

```
-n sponsornode
--sponsornode=sponsornode
--sponsornode sponsornode
```

指定监护人节点的名称。

您可以为 *sponsornode* 指定名称或节点标识符。使用 `add` 子命令向群集添加节点时，监护人节点是添加到群集中的第一个活动节点。从这一点来说，该节点将保留此群集的 *sponsornode*。使用 `remove` 子命令删除节点时，您可以指定除要删除的节点之外的任何活动节点作为监护人节点。

默认情况下，每次您使用子命令指定 *sponsornode* 时，*sponsornode* 所属的群集都是受该子命令影响的群集。

```
-o {- | clconfigfile}
--output={- | clconfigfile}
--output {- | clconfigfile}
```

将节点配置信息写入到文件或标准输出 (stdout)。[clconfiguration\(5CL\) \[1245\]](#) 手册页中描述了该配置信息的格式。

如果您使用该选项指定文件名称，该选项将创建新的文件。然后将配置信息置于该文件中。如果您使用该选项指定 `-`，则会将配置信息发送到标准输出 (stdout)。命令的所有其他标准输出将受到抑制。

仅可以将此选项与 `export` 子命令一起使用。

```
-p name
--property=name
--property name
```

指定您想要使用 `show` 子命令显示其相关信息的节点属性。

有关可以使用 `set` 子命令添加或修改的属性的信息，请参见 `-p name= value` 选项的描述。

您可以使用该选项指定以下属性：

`privatehostname`

专用主机名用于通过专用群集互连使用 IP 访问给定的节点。默认情况下，向群集添加节点时，该选项会使用专用主机名 `clusternode nodeid-priv`。

`reboot_on_path_failure`

该属性可以设置的值为 `enabled` 和 `disabled`。

`-p name=value`
`--property=name=value`
`--property name=value`

指定您想要使用 `set` 子命令添加或修改的节点属性。

允许使用多个 `-p name= value` 实例。

有关可以使用 `show` 子命令显示其相关信息的属性的信息，请参见 `-p name` 选项的描述。

您可以使用该选项修改以下属性：

`defaultpsetmin`

设置默认处理器集资源中可用的最小 CPU 数。

默认值是 1，最小值也是 1。最大值为设置该属性的计算机上的 CPU 数。

`globalzoneshares`

设置分配给全局区域的份额数。

您可以指定介于 1 和 65535（含这两个数）之间的值。要了解该上限，请参见 [Unresolved link to "prctl1"](#) 手册页中有关 `zone.cpu-shares` 属性的信息。`globalzoneshares` 的默认值为 1。

`hardlimit`

定义节点上资源组负载的强制性上界。节点上的总负载绝不允许超过硬限制。

`hardlimit` 属性为无符号整数。`softlimit` 属性为无符号整数。`hardlimit` 属性的默认值为 `null`。Null 或空值表示相应的 `limitname` 在节点上没有限制。如果指定非空值，它不得超过 10,000,000。

`limitname`

`limitname` 属性为字符串。名称与由 `hardlimit` 和 `softlimit` 属性分别指定的两个值（硬负载限制和软负载限制）相关联。

有关如何为每个 `limitname` 属性分配负载因子的信息，请参见 [clresourcegroup\(1CL\) \[253\]](#) 手册页。您还可以使用 `clresourcegroup` 命令确定优先级和抢占模式。有关如何在所有节点中分配资源组负载的信息，请参见 [cluster\(1CL\) \[461\]](#) 手册页。

`privatehostname`

用于通过专用群集传输使用 IP 访问给定的节点。默认情况下，向群集添加节点时，该选项会使用专用主机名 `clusternodeid -priv`。

在修改专用主机名之前，您必须在所有节点上禁用使用该专用主机名的所有资源或应用程序。请参见 [Unresolved link to "Oracle Solaris Cluster 系统管理指南中的如何更改节点专用主机名"](#) 中标题为“更改专用主机名”的示例。

请勿将专用主机名存储在 `hosts` 数据库或任何命名服务数据库中。请参见 [Unresolved link to "hosts4"](#) 手册页。特定的 `nsswitch` 命令在所有主机名中执行专用主机名查找。请参见 [Unresolved link to "nsswitch.conf4"](#) 手册页。

如果未指定 *value*，该选项将会使用默认专用主机名 `clusternode nodeid-priv`。

`reboot_on_path_failure`

如果满足以下条件，则启用节点在所有受监视的共享磁盘路径发生故障时自动重新引导的功能：

- 节点上所有受监视的共享磁盘路径均发生故障。
- 至少有一个磁盘可从群集中的其他节点进行访问。scdpm 守护进程使用专用互连检查是否可以从群集中的其他节点访问磁盘。如果专用互连处于禁用状态，scdpm 守护进程将无法从其他节点获取磁盘状态。

您只可以使用 `set` 子命令修改该属性。您可以将该属性设置为 `enabled` 或 `disabled`。

重新引导节点会将该节点管理的所有资源组和设备组在另一个节点上重新启动。

当节点自动重新引导后，如果该节点上所有受监视的共享磁盘路径仍不可访问，该节点不会再次自动重新引导。但是，当节点重新引导后，如果任何受监视的共享磁盘路径可用而随后所有受监视的共享磁盘路径又出现故障，则该节点会再次自动重新引导。

如果启用 `reboot_on_path_failure` 属性，在判断是否需要重新引导节点时不会考虑本地磁盘路径的状态。仅受监视的共享磁盘会受影响。

如果将该属性设置为 `disabled` 并且节点上的所有受监视的共享磁盘路径发生故障，则该节点不会重新引导。

`softlimit`

定义节点上资源组负载的建议性上界。例如，群集容量不足以分配负载时，节点上的总负载可以超过软限制。超过软负载限制时，将在显示群集状态的命令或工具中标记条件。

`softlimit` 属性为无符号整数。`softlimit` 属性的默认值为 0。软限制的值 0 表示没有强加任何软限制；状态命令中不存在任何已超过软限制警告。`softlimit` 属性的最大值为 10,000,000。用于特定负载限制的 `softlimit` 属性必须小于或等于 `hardlimit` 值。

`-T seconds`

`--time=seconds`

`--time seconds`

指定在从节点中清除资源组之后阻止资源组切换回该节点的秒数。

仅可以将此选项与 `evacuate` 子命令一起使用。您必须将 `seconds` 指定为介于 0 和 65535 之间的整数值。如果未指定值，将使用默认值 60 秒。

在清除完成之后的 60 秒或指定的秒数内，防止资源组在清除的节点上进行故障转移或自动联机。

但是，如果您使用 `switch` 或 `online` 子命令使资源组联机，或清除的节点重新引导，则清除计时器将立即过期并再次允许自动故障转移。

`-v`
`--verbose`

在标准输出 (stdout) 中显示详细信息。

`-V`
`--version`

显示该命令的版本。

如果是与其他选项、子命令或操作数一起指定该选项，将忽略所有这些内容。仅显示该命令的版本。不会发生任何其他处理。

`-z {zoneclustername | global | all}`
`--zonecluster={zoneclustername | global | all}`
`--zonecluster {zoneclustername | global | all}`

指定您想要在其中查找某些节点信息的群集。

如果指定此选项，还必须指定以下列表中的参数之一：

`zoneclustername` 指定仅显示有关名为 `zoneclustername` 的区域群集节点的信息。

`global` 指定仅显示有关全局群集节点的信息。

`all` 指定显示有关所有全局群集节点和区域群集节点的信息。

支持以下操作数：

`node` 要管理的节点的名称。
使用 `add` 子命令时，您需为 `node` 指定主机名。使用其他子命令时，您需要为 `node` 指定节点名称或节点标识符。

`+` 群集中的所有节点。

[Intro\(1CL\) \[17\]](#) 手册页列出了该命令集中所有命令的一整套退出状态代码。

如果命令对于所有指定的操作数均成功执行，则返回零 (CL_NOERR)。如果某个操作数出错，命令会继续处理操作数列表中的下一个操作数。返回的退出代码始终反映最先发生的错误。

此命令返回下列退出状态代码：

0 CL_NOERR

没有错误

您发出的命令已成功完成。

1 CL_ENOMEM

交换空间不足

某个群集节点耗尽了交换内存或其他操作系统资源。

3 CL_EINVAL

参数无效

您键入的命令不正确，或者使用 `-i` 选项提供的群集配置信息语法不正确。

6 CL_EACCESS

权限被拒绝

指定的对象不可访问。要发出命令，可能需要拥有超级用户或 RBAC 访问权限。有关更多信息，请参见 [Unresolved link to "su1M"](#) 和 [Unresolved link to "rbac5"](#) 手册页。

15 CL_EPROP

属性无效

您使用 `-p`、`-y` 或 `-x` 选项指定的属性或值不存在或不被允许。

35 CL_EIO

I/O 错误

发生了物理输入/输出错误。

36 CL_ENOENT

没有这样的对象

由于以下某个原因，找不到您所指定的对象：

- 该对象不存在。
- 您尝试使用 `-o` 选项创建的配置文件路径中的某个目录不存在。
- 您尝试使用 `-i` 选项访问的配置文件包含错误。

37 CL_EOP

不允许操作

您尝试对不受支持的配置执行操作，或者您所执行的操作不受支持。

例 95 向群集添加节点

以下命令将配置运行该命令的节点并将其添加到现有群集中。默认情况下，该示例使用 `/globaldevices` 作为全局设备挂载点。默认情况下，该示例还使用 `clusternode1-priv` 作为专用主机名。

该命令将群集命名为 `cluster-1` 并指定监护人节点为 `phys-schost-1`。此命令还指定将适配器 `net1` 连接到传输交换机 `switch1`。最后，此命令指定将适配器 `net2` 连接到传输交换机 `switch2`。

```
# clnode add -c cluster-1 -n phys-schost-1 \
```

```
-e phys-schost-2:net1,switch1 -e phys-schost-2:net2,switch2
```

例 96 从群集中删除节点

以下命令可从群集中删除节点。该命令可删除运行该命令的节点。该节点处于非群集模式中。

```
# clnode remove
```

例 97 更改与节点关联的专用主机名

以下命令可将节点 `phys-schost-1` 的专用主机名更改为默认设置。

```
# clnode set -p privatehost=phys-schost-1
```

例 98 更改所有节点的专用主机名设置

以下命令可将所有节点的专用主机名设置更改为默认值。在这种情况下，您必须在等号 (=) 和加号 (+) 之间插入一个空格，以表示 + 是加号操作数。

```
# clnode set -p privatehost= +
```

例 99 在全局群集节点和区域群集节点上设置负载限制

以下命令可修改全局群集中所有节点上的现有负载限制。该示例定义了三个负载限制 (`mem_load`、`disk_load` 和 `cpu_load`) 并为每个负载限制设置软限制和硬限制。`mem_load` 负载限制的软限制为 11，`disk_load` 没有软限制，`cpu_load` 没有硬限制。示例中的 + 操作数可修改所有节点上的负载限制。

```
# clnode set-loadlimit -p limitname=mem_load -p softlimit=11 -p hardlimit=20 +
```

```
# clnode set-loadlimit -p limitname=disk_load -p hardlimit=20 +
```

```
# clnode set-loadlimit -p limitname=cpu_load -p softlimit=8 node1:zone1 node2:zone2
```

在全局区域中，以下命令可修改区域群集节点上的负载限制。该示例为区域群集节点定义了具有硬限制的负载限制。

```
# clnode set-loadlimit -  
Z zoneclustername  
-p limitname=zc_disk_load -p  
hardlimit=15  
zc-node1
```

例 100 显示群集中所有节点的状态

以下命令可显示群集中所有节点的状态。

```
# clnode status
```

```
=== Cluster Nodes ===
```

```
--- Node Status ---
```

Node Name	Status
phys-schost-1	Online
phys-schost-2	Online

例 101 显示群集中所有节点的详细状态

以下命令可显示群集中所有节点的详细状态。

```
# clnode status -v
```

```
=== Cluster Nodes ===
```

```
--- Node Status ---
```

Node Name	Status
phys-schost-1	Online
phys-schost-2	Online

```
--- Node IPMP Group Status ---
```

Node Name	Group Name	Status	Adapter	Status
phys-schost-1	sc_ipmp0	Online	net0	Online
phys-schost-2	sc_ipmp0	Online	net0	Online

```
--- Load Limit Status ---
```

Node Name	Load Limit Name	Soft Limit/Hard Limit	Load	Status
phys-schost-1	mem_load	30/50	23	OK
	disk_load	10/15	14	Softlimit Exceeded
	cpu_load	2/unlimited	1	OK
phys-schost-2	disk_load	90/97	11	OK
	cpu_load	unlimited/unlimited	0	OK

例 102 显示所有节点的负载限制状态

以下命令可显示群集中所有节点的负载限制状态。

```
# clnode status -l
```

```
--- Load Limit Status ---
```

Node Name	Load Limit Name	Soft Limit/Hard Limit	Load	Status
phys-schost-1	mem_load	30/50	23	OK
	disk_load	10/15	14	Softlimit Exceeded
	cpu_load	2/unlimited	1	OK
phys-schost-2	disk_load	90/97	11	OK
	cpu_load	unlimited/unlimited	0	OK

例 103 显示群集中所有全局群集节点和区域群集节点的状态

以下命令可显示群集中所有全局群集节点和区域群集节点的状态。

```
# clnode status -Z all

=== Cluster Nodes ===

--- Node Status ---

Node Name                               Status
-----
global:phys-schost-1                    Online
global:phys-schost-2                    Online
global:phys-schost-4                    Online
global:phys-schost-3                    Online

=== Zone Cluster Nodes ===

--- Node Status ---

Node Name                               Status
-----
cz2:phys-schost-1                       Online
cz2:phys-schost-3                       Offline
```

例 104 显示群集中所有节点的配置信息

以下命令可显示群集中所有节点的配置信息。

```
# clnode show

=== Cluster Nodes ===

Node Name:                               phys-schost-1
Node ID:                                  1
Enabled:                                  yes
privatehostname:                          clusternode1-priv
reboot_on_path_failure:                   disabled
globalzonestates:                         1
defaultpsetmin:                           1
quorum_vote:                              1
quorum_defaultvote:                       1
```

```

quorum_resv_key:                0x4487349A00000001
Transport Adapter List:         net2, net3

Node Name:                      phys-schost-2
Node ID:                        2
Enabled:                        yes
privatehostname:               clusternode2-priv
reboot_on_path_failure:        disabled
globalzoneshares:              1
defaultpsetmin:                1
quorum_vote:                   1
quorum_defaultvote:            1
quorum_resv_key:                0x4487349A00000002
Transport Adapter List:         net2, net3

```

例 105 显示群集中特定节点的配置信息

以下命令可显示群集中 phys-schost-1 的配置信息。

```

# clnode show phys-schost-1
=== Cluster Nodes ===

Node Name:                      phys-schost-1
Node ID:                        1
Enabled:                        yes
privatehostname:               clusternode1-priv
reboot_on_path_failure:        disabled
globalzoneshares:              1
defaultpsetmin:                1
quorum_vote:                   1
quorum_defaultvote:            1
quorum_resv_key:                0x4487349A00000001
Transport Adapter List:         net2, net3

```

有关下列属性的说明，请参见 [Unresolved link to " attributes5"](#) :

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

[Unresolved link to "](#)

[prctl1"](#)、[claccess\(1CL\) \[31\]](#)、[clresourcegroup\(1CL\) \[253\]](#)、[cluster\(1CL\) \[461\]](#)、[Intro\(1CL\) \[17\]](#)、[Unreso](#)
[link to " newfs1M"](#)、[Unresolved link to " su1M"](#)、[Unresolved link to "](#)
[hosts4"](#)、[scinstall\(1M\) \[697\]](#)、[Unresolved link to " nsswitch.conf4"](#)、[Unresolved](#)
[link to " vfstab4"](#)、[Unresolved link to " attributes5"](#)、[Unresolved link to "](#)
[rbac5"](#)、[clconfiguration\(5CL\) \[1245\]](#)、[Unresolved link to " lofi7D"](#)

请参见[Unresolved link to " Oracle Solaris Cluster 系统管理指南 中的管理群集概述"](#)中介绍如何更改专用主机名的示例。

超级用户可以运行此命令的所有格式。

所有用户都可以运行带有 -? (帮助) 或 -v (版本) 选项的此命令。

要运行带有子命令的 `clnode` 命令，超级用户以外的用户需要具有 RBAC 授权。请参见下表。

子命令	RBAC 授权
<code>add</code>	<code>solaris.cluster.modify</code>
<code>clear</code>	<code>solaris.cluster.modify</code>
<code>create-loadlimit</code>	<code>solaris.cluster.modify</code>
<code>delete-loadlimit</code>	<code>solaris.cluster.modify</code>
<code>evacuate</code>	<code>solaris.cluster.admin</code>
<code>export</code>	<code>solaris.cluster.read</code>
<code>list</code>	<code>solaris.cluster.read</code>
<code>remove</code>	<code>solaris.cluster.modify</code>
<code>rename</code>	<code>solaris.cluster.modify</code>
<code>set</code>	<code>solaris.cluster.modify</code>
<code>set-loadlimit</code>	<code>solaris.cluster.modify</code>
<code>show</code>	<code>solaris.cluster.read</code>
<code>show-rev</code>	<code>solaris.cluster.read</code>
<code>status</code>	<code>solaris.cluster.read</code>

名称

clpstring, clps — 管理 Oracle Solaris Cluster 专用字符串

```
/usr/cluster/bin/clpstring -V

/usr/cluster/bin/clpstring subcommand -?

/usr/cluster/bin/clpstring subcommand [options] -v [pstring-name[...]]

/usr/cluster/bin/clpstring create -b object-instance [-f
    stringvalue-file] [-t object-type] [-Z {zoneclustername | global}]
    pstring-name

/usr/cluster/bin/clpstring delete [-F] [-Z {zoneclustername[,...] |
    global | all}] + | pstring-name ...

/usr/cluster/bin/clpstring list [-b object-instance[,...]] [-t
    type[,...]] [-Z {zoneclustername[,...] | global | all}] [+ |
    pstring-name[...]]

/usr/cluster/bin/clpstring set [-f stringvalue-file] [-Z
    {zoneclustername | global}] pstring-name

/usr/cluster/bin/clpstring show [-b object-instance[,...]] [-t
    type[,...]] [-Z {zoneclustername[,...] | global | all}] [+ |
    pstring-name[...]]
```

clpstring 命令管理 Oracle Solaris Cluster 专用字符串。专用字符串使用唯一名称进行标识，其具有的编码值只能使用 `scha_cluster_get(1HA)` [567] 命令获取。

群集对象（例如资源）使用专用字符串来安全地存储和检索专用值。一个典型的应用可能是代理使用的内部密码。

clps 命令是 clpstring 命令的简短格式。clpstring 命令和 clps 命令是相同的。您可以使用此命令的任一种格式。

此命令的一般格式如下所示：

```
clpstring [subcommand] [options] [operands]
```

仅当 *options* 指定 -? 选项或 -v 选项时，才可以省略 *subcommand*。

此命令的每个选项都有长格式和简短格式两种格式。在本手册页的 选项 部分给出了每个选项的两种格式，以及对该选项的描述。

使用 clpstring 命令执行以下管理任务：

- 创建要由可能存在也可能尚不存在的群集对象实例使用的专用字符串

-
- 更新专用字符串的值
 - 从群集配置中删除专用字符串
 - 显示专用字符串的指定信息

只能在活动群集节点上运行 `clpstring` 命令。无论您在哪个节点上运行该命令，运行结果始终相同。

`clpstring` 命令的所有子命令都可以在全局区域和区域群集中运行。在全局区域中运行该命令时，可以使用 `-z` 选项指定要将操作限制到的特定区域群集的名称。

您可以从全局群集节点访问所有区域群集信息，但无法从特定区域群集访问其他区域群集信息。如果不将操作限制到特定区域群集，则所用的子命令仅会在当前群集中运行。

子命令

支持以下子命令：

create

创建要由 Oracle Solaris Cluster 对象实例使用的专用字符串。

使用 `-b` 选项可指定要使用此专用字符串的群集对象实例。在为实例创建专用字符串时，对象实例不必存在于群集配置中。使用 `-t` 选项可指示群集对象实例的类型。默认对象类型为 `resource`。

使用 `-f` 选项可指定包含专用字符串值的文件。如果未指定 `-f`，该命令将会提示您提供专用字符串值。可以在“选项”部分中找到详细信息。

超级用户以外的用户需要具有 `solaris.cluster.modify` 基于角色的访问控制 (role-based access control, RBAC) 授权才能使用 `create` 子命令。有关更多信息，请参见 [Unresolved link to "rbac5"](#) 手册页。

另请参见 `delete` 子命令的描述。

delete

从 Oracle Solaris Cluster 配置中删除指定的专用字符串。

如果不指定强制选项 `-F`，则必须已删除了为其创建专用字符串的群集对象实例。如果指定了 `-F` 选项，该命令将强制删除专用字符串，即使关联的对象实例仍存在于群集配置中并在使用该专用字符串也是如此。有关更多信息，请参见“选项”中的 `-F`。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用 `delete` 子命令。

另请参见 `create` 子命令的描述。

list

显示在群集中创建的所有专用字符串的名称，但不显示它们的值。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

set

设置指定专用字符串的值。可以使用 `-f` 选项指定专用字符串值的源。如果未指定 `-f`，该命令将会提示您提供相应的值。有关专用字符串值的信息，请参见“选项”部分中的 `-f` 选项描述。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

show

显示专用字符串的指定信息，但不显示它们的值。指定信息包括专用字符串名称、其关联的对象实例以及实例的对象类型。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

支持以下选项：

`-?`

`--help`

显示帮助信息。使用该选项时，不会执行任何其他处理。

指定此选项时，子命令可有可无。

如果仅指定此选项而不使用子命令，将会显示此命令的子命令列表。

如果使用子命令且指定此选项，将会显示该子命令的用法选项。

`-b object-instance`

`--object-instance=object-instance`

`--object-instance object-instance`

指定使用或将要使用专用字符串的对象实例的名称。当前仅支持对象类型为 `resource` 的对象实例。

`-F`

`--force`

强制删除指定的专用字符串。此选项仅可以与 `delete` 子命令一同指定。

将此选项与 `delete` 子命令一起使用时，将删除指定的专用字符串，即使使用专用字符串的对象实例仍存在于群集中也是如此。通常应先从群集中删除对象实例，然后再删除其专用字符串。

`-f stringvalue-file`

`--stringvalue-file=stringvalue-file`

`--stringvalue-file stringvalue-file`

指定包含专用字符串值的文件。文件名必须是可从运行该命令的节点访问的完整路径。

为安全起见，无法在命令行选项中指定专用字符串值。为了确保值安全，请将其置于纯文本文件中并通过使用 `-f` 选项指定该文件的完整路径。将字符串值文件的所

所有者设置为 root，并将文件的权限设置为可由 root 读取并禁止任何组访问和全局访问。要进一步提高安全性，可以在运行设置专用字符串值的命令之后删除该文件。

如果不指定 -f 选项，该命令将会提示您输入专用字符串值两次，以确认输入内容相同。该命令将从控制终端读取相应的值，同时禁用回显。

可以指定 -f - (-f 后跟一个空格和短划线) 以直接从标准输入只读取一次专用字符串值。专用字符串值在键入时会回显在屏幕上，如果该命令用于编写脚本，专用字符串值会显示在脚本中；因此以该方式设置专用字符串值时应小心。

专用字符串值输入具有以下要求：

- 字符串长度不得超过 257 个字符。
- 字符串中不能包含 NULL 字符。

-t *object-type*

--object-type=*object-type*

--object-type *object-type*

指定对象实例的类型。默认类型为 resource，该类型是当前唯一可使用专用字符串的对象类型，因此不需要 -t 选项。

-V

--version

显示该命令的版本。

指定此选项时，不要将其与子命令、操作数或其他选项结合使用。子命令、操作数或其他选项将被忽略。-V 选项只显示命令的版本。不会执行任何其他操作。

-v

--verbose

将详细消息显示到标准输出。

可以在任何格式的此命令中指定该选项。

-Z {*zoneclustername* | global | all}

--zoneclustername={*zoneclustername* | global | all}

指定要在其中创建专用字符串的群集或专用字符串所位于的群集。

所有子命令都支持该选项。

如果指定此选项，还必须指定以下列表中的参数之一：

zoneclustername 指定使用此选项的命令仅对名为 *zoneclustername* 的区域群集中的所有指定专用字符串起作用。

global 指定使用此选项的命令仅对全局群集中的所有指定专用字符串起作用。

all 如果在全局群集中使用此参数，则会指定使用此参数的命令对所有群集（包括全局群集和所有区域群集）中的所有指定资源起作用。

仅支持以下操作数：

pstring-name 指定专用字符串的名称。创建专用字符串时，指定的名称在群集中必须是唯一的。如果子命令接受多个专用字符串，则可以使用加号 (+) 替代 *pstring-name* 来指定所有专用字符串。

如果命令对于所有指定的操作数均成功执行，则返回零 (CL_NOERR)。如果某个操作数出错，命令会继续处理操作数列表中的下一个操作数。返回的退出代码始终反映最先发生的错误。

可返回下列退出代码：

0 CL_NOERR

没有错误
您发出的命令已成功完成。

1 CL_ENOMEM

交换空间不足
某个群集节点耗尽了交换内存或其他操作系统资源。

3 CL_EINVAL

参数无效
您键入的命令不正确，或者使用 `-i` 选项提供的群集配置信息语法不正确。

6 CL_EACCESS

权限被拒绝
指定的对象不可访问。要发出命令，可能需要拥有超级用户或 RBAC 访问权限。有关更多信息，请参见 [Unresolved link to " su1M"](#) 和 [Unresolved link to " rbac5"](#) 手册页。

18 CL_EINTERNAL

遇到内部错误
内部错误表示软件缺陷或其他缺陷。

36 CL_ENOENT

没有这样的对象
由于以下某个原因，找不到您所指定的对象：

- 该对象不存在。
- 您尝试使用 `-o` 选项创建的配置文件路径中的某个目录不存在。
- 您尝试使用 `-i` 选项访问的配置文件包含错误。

37 CL_EOP

不允许操作

您尝试对不受支持的配置执行操作，或者您所执行的操作不受支持。

39 CL_EEXIST

对象已存在

您所指定的设备、设备组、群集互连组件、节点、群集、资源、资源类型、资源组或专用字符串已存在。

41 CL_ETYPE

类型无效

您使用 -t 或 -p 选项指定的类型不存在。

这些退出值与 [scha_calls\(3HA\) \[897\]](#) 手册页中描述的返回码兼容。

例 106 为全局群集或区域群集中的资源创建专用字符串

以下命令为全局群集中的资源实例创建专用字符串。

```
# clpstring create -b resource1 -t resource -v pstring1
Enter string value:
Enter string value again:
Private string "pstring1" is created for the global cluster.
```

以下命令在全局区域中运行并为名为 zc1 的区域群集创建专用字符串。专用字符串的值是在 /pvalue.file 文件中指定的。

```
# clpstring create -Z zc1 -b resource2 -f /pvalue.file pstring2
```

例 107 从全局群集或区域群集配置中删除专用字符串

以下命令将从群集配置中删除所有专用字符串，无论对象实例是否仍存在于群集中。

```
# clpstring delete -F +
```

以下命令从名为 zc1 的区域群集中删除指定的专用字符串。

```
# clpstring delete -Z zc1 pstring1
```

例 108 显示在群集中创建的专用字符串的指定信息

以下命令显示群集中的专用字符串。

```
# clpstring show
=== Private Strings ===
```

```

Pstring Name:                pstring1
  Object Instance:           resource1
  Object Type:                resource

Pstring Name:                pstring2
  Object Instance:           object2
  Object Type:                resource

```

例 109 列出全局群集和区域群集中的专用字符串

以下命令显示全局群集和所有区域群集中的专用字符串名称。

```

# clpstring list -Z all
global:pstring1
global:pstring2
zc1:pstring1
zc1:pstring2
zc2:pstring

```

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

[Intro\(1CL\) \[17\]](#)、[cluster\(1CL\) \[461\]](#)、[scha_calls\(3HA\) \[897\]](#)、[Unresolved link to "attributes5"](#)、[Unresolved link to "rbac5"](#)

超级用户可以运行此命令的所有格式。

任何用户都可以运行带有以下选项的此命令：

- -? 选项
- -v 选项

要运行带有子命令的此命令，超级用户以外的用户需要具有 RBAC 授权。请参见下表。

子命令	RBAC 授权
create	solaris.cluster.modify
delete	solaris.cluster.modify
list	solaris.cluster.read

子命令	RBAC 授权
set	solaris.cluster.modify
show	solaris.cluster.read

名称

clpstring, clps — 管理 Oracle Solaris Cluster 专用字符串

```
/usr/cluster/bin/clpstring -V

/usr/cluster/bin/clpstring subcommand -?

/usr/cluster/bin/clpstring subcommand [options] -v [pstring-name[...]]

/usr/cluster/bin/clpstring create -b object-instance [-f
    stringvalue-file] [-t object-type] [-Z {zoneclustername | global}]
    pstring-name

/usr/cluster/bin/clpstring delete [-F] [-Z {zoneclustername[,...] |
    global | all}] + | pstring-name ...

/usr/cluster/bin/clpstring list [-b object-instance[,...]] [-t
    type[,...]] [-Z {zoneclustername[,...] | global | all}] [+ |
    pstring-name[...]]

/usr/cluster/bin/clpstring set [-f stringvalue-file] [-Z
    {zoneclustername | global}] pstring-name

/usr/cluster/bin/clpstring show [-b object-instance[,...]] [-t
    type[,...]] [-Z {zoneclustername[,...] | global | all}] [+ |
    pstring-name[...]]
```

clpstring 命令管理 Oracle Solaris Cluster 专用字符串。专用字符串使用唯一名称进行标识，其具有的编码值只能使用 `scha_cluster_get(1HA)` [567] 命令获取。

群集对象（例如资源）使用专用字符串来安全地存储和检索专用值。一个典型的应用可能是代理使用的内部密码。

clps 命令是 clpstring 命令的简短格式。clpstring 命令和 clps 命令是相同的。您可以使用此命令的任一种格式。

此命令的一般格式如下所示：

```
clpstring [subcommand] [options] [operands]
```

仅当 *options* 指定 -? 选项或 -v 选项时，才可以省略 *subcommand*。

此命令的每个选项都有长格式和简短格式两种格式。在本手册页的 选项 部分给出了每个选项的两种格式，以及对该选项的描述。

使用 clpstring 命令执行以下管理任务：

- 创建要由可能存在也可能尚不存在的群集对象实例使用的专用字符串

- 更新专用字符串的值
- 从群集配置中删除专用字符串
- 显示专用字符串的指定信息

只能在活动群集节点上运行 `clpstring` 命令。无论您在哪个节点上运行该命令，运行结果始终相同。

`clpstring` 命令的所有子命令都可以在全局区域和区域群集中运行。在全局区域中运行该命令时，可以使用 `-z` 选项指定要将操作限制到的特定区域群集的名称。

您可以从全局群集节点访问所有区域群集信息，但无法从特定区域群集访问其他区域群集信息。如果不将操作限制到特定区域群集，则所用的子命令仅会在当前群集中运行。

子命令

支持以下子命令：

create

创建要由 Oracle Solaris Cluster 对象实例使用的专用字符串。

使用 `-b` 选项可指定要使用此专用字符串的群集对象实例。在为实例创建专用字符串时，对象实例不必存在于群集配置中。使用 `-t` 选项可指示群集对象实例的类型。默认对象类型为 `resource`。

使用 `-f` 选项可指定包含专用字符串值的文件。如果未指定 `-f`，该命令将会提示您提供专用字符串值。可以在“选项”部分中找到详细信息。

超级用户以外的用户需要具有 `solaris.cluster.modify` 基于角色的访问控制 (role-based access control, RBAC) 授权才能使用 `create` 子命令。有关更多信息，请参见 [Unresolved link to "rbac5"](#) 手册页。

另请参见 `delete` 子命令的描述。

delete

从 Oracle Solaris Cluster 配置中删除指定的专用字符串。

如果不指定强制选项 `-F`，则必须已删除了为其创建专用字符串的群集对象实例。如果指定了 `-F` 选项，该命令将强制删除专用字符串，即使关联的对象实例仍存在于群集配置中并在使用该专用字符串也是如此。有关更多信息，请参见“选项”中的 `-F`。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用 `delete` 子命令。

另请参见 `create` 子命令的描述。

list

显示在群集中创建的所有专用字符串的名称，但不显示它们的值。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

set

设置指定专用字符串的值。可以使用 `-f` 选项指定专用字符串值的源。如果未指定 `-f`，该命令将会提示您提供相应的值。有关专用字符串值的信息，请参见“选项”部分中的 `-f` 选项描述。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

show

显示专用字符串的指定信息，但不显示它们的值。指定信息包括专用字符串名称、其关联的对象实例以及实例的对象类型。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

支持以下选项：

`-?`

`--help`

显示帮助信息。使用该选项时，不会执行任何其他处理。

指定此选项时，子命令可有可无。

如果仅指定此选项而不使用子命令，将会显示此命令的子命令列表。

如果使用子命令且指定此选项，将会显示该子命令的用法选项。

`-b object-instance`

`--object-instance=object-instance`

`--object-instance object-instance`

指定使用或将要使用专用字符串的对象实例的名称。当前仅支持对象类型为 `resource` 的对象实例。

`-F`

`--force`

强制删除指定的专用字符串。此选项仅可以与 `delete` 子命令一同指定。

将此选项与 `delete` 子命令一起使用时，将删除指定的专用字符串，即使使用专用字符串的对象实例仍存在于群集中也是如此。通常应先从群集中删除对象实例，然后再删除其专用字符串。

`-f stringvalue-file`

`--stringvalue-file=stringvalue-file`

`--stringvalue-file stringvalue-file`

指定包含专用字符串值的文件。文件名必须是可从运行该命令的节点访问的完整路径。

为安全起见，无法在命令行选项中指定专用字符串值。为了确保值安全，请将其置于纯文本文件中并通过使用 `-f` 选项指定该文件的完整路径。将字符串值文件的所

所有者设置为 root，并将文件的权限设置为可由 root 读取并禁止任何组访问和全局访问。要进一步提高安全性，可以在运行设置专用字符串值的命令之后删除该文件。

如果不指定 -f 选项，该命令将会提示您输入专用字符串值两次，以确认输入内容相同。该命令将从控制终端读取相应的值，同时禁用回显。

可以指定 -f - (-f 后跟一个空格和短划线) 以直接从标准输入只读取一次专用字符串值。专用字符串值在键入时会回显在屏幕上，如果该命令用于编写脚本，专用字符串值会显示在脚本中；因此以该方式设置专用字符串值时应小心。

专用字符串值输入具有以下要求：

- 字符串长度不得超过 257 个字符。
- 字符串中不能包含 NULL 字符。

-t *object-type*

--object-type=*object-type*

--object-type *object-type*

指定对象实例的类型。默认类型为 resource，该类型是当前唯一可使用专用字符串的对象类型，因此不需要 -t 选项。

-V

--version

显示该命令的版本。

指定此选项时，不要将其与子命令、操作数或其他选项结合使用。子命令、操作数或其他选项将被忽略。-V 选项只显示命令的版本。不会执行任何其他操作。

-v

--verbose

将详细消息显示到标准输出。

可以在任何格式的此命令中指定该选项。

-Z {*zoneclustername* | global | all}

--zoneclustername={*zoneclustername* | global | all}

指定要在其中创建专用字符串的群集或专用字符串所位于的群集。

所有子命令都支持该选项。

如果指定此选项，还必须指定以下列表中的参数之一：

zoneclustername 指定使用此选项的命令仅对名为 *zoneclustername* 的区域群集中的所有指定专用字符串起作用。

global 指定使用此选项的命令仅对全局群集中的所有指定专用字符串起作用。

all 如果在全局群集中使用此参数，则会指定使用此参数的命令对所有群集（包括全局群集和所有区域群集）中的所有指定资源起作用。

仅支持以下操作数：

pstring-name 指定专用字符串的名称。创建专用字符串时，指定的名称在群集中必须是唯一的。如果子命令接受多个专用字符串，则可以使用加号 (+) 替代 *pstring-name* 来指定所有专用字符串。

如果命令对于所有指定的操作数均成功执行，则返回零 (CL_NOERR)。如果某个操作数出错，命令会继续处理操作数列表中的下一个操作数。返回的退出代码始终反映最先发生的错误。

可返回下列退出代码：

0 CL_NOERR

没有错误
您发出的命令已成功完成。

1 CL_ENOMEM

交换空间不足
某个群集节点耗尽了交换内存或其他操作系统资源。

3 CL_EINVAL

参数无效
您键入的命令不正确，或者使用 `-i` 选项提供的群集配置信息语法不正确。

6 CL_EACCESS

权限被拒绝
指定的对象不可访问。要发出命令，可能需要拥有超级用户或 RBAC 访问权限。有关更多信息，请参见 [Unresolved link to " su1M"](#) 和 [Unresolved link to " rbac5"](#) 手册页。

18 CL_EINTERNAL

遇到内部错误
内部错误表示软件缺陷或其他缺陷。

36 CL_ENOENT

没有这样的对象
由于以下某个原因，找不到您所指定的对象：

- 该对象不存在。
- 您尝试使用 `-o` 选项创建的配置文件路径中的某个目录不存在。
- 您尝试使用 `-i` 选项访问的配置文件包含错误。

37 CL_EOP

不允许操作

您尝试对不受支持的配置执行操作，或者您所执行的操作不受支持。

39 CL_EEXIST

对象已存在

您所指定的设备、设备组、群集互连组件、节点、群集、资源、资源类型、资源组或专用字符串已存在。

41 CL_ETYPE

类型无效

您使用 -t 或 -p 选项指定的类型不存在。

这些退出值与 [scha_calls\(3HA\) \[897\]](#) 手册页中描述的返回码兼容。

例 110 为全局群集或区域群集中的资源创建专用字符串

以下命令为全局群集中的资源实例创建专用字符串。

```
# clpstring create -b resource1 -t resource -v pstring1
Enter string value:
Enter string value again:
Private string "pstring1" is created for the global cluster.
```

以下命令在全局区域中运行并为名为 zc1 的区域群集创建专用字符串。专用字符串的值是在 /pvalue.file 文件中指定的。

```
# clpstring create -Z zc1 -b resource2 -f /pvalue.file pstring2
```

例 111 从全局群集或区域群集配置中删除专用字符串

以下命令将从群集配置中删除所有专用字符串，无论对象实例是否仍存在于群集中。

```
# clpstring delete -F +
```

以下命令从名为 zc1 的区域群集中删除指定的专用字符串。

```
# clpstring delete -Z zc1 pstring1
```

例 112 显示在群集中创建的专用字符串的指定信息

以下命令显示群集中的专用字符串。

```
# clpstring show
=== Private Strings ===
```

```

Pstring Name:                pstring1
  Object Instance:           resource1
  Object Type:                resource

Pstring Name:                pstring2
  Object Instance:           object2
  Object Type:                resource

```

例 113 列出全局群集和区域群集中的专用字符串

以下命令显示全局群集和所有区域群集中的专用字符串名称。

```

# clpstring list -Z all
global:pstring1
global:pstring2
zc1:pstring1
zc1:pstring2
zc2:pstring

```

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

[Intro\(1CL\) \[17\]](#)、[cluster\(1CL\) \[461\]](#)、[scha_calls\(3HA\) \[897\]](#)、[Unresolved link to "attributes5"](#)、[Unresolved link to "rbac5"](#)

超级用户可以运行此命令的所有格式。

任何用户都可以运行带有以下选项的此命令：

- -? 选项
- -v 选项

要运行带有子命令的此命令，超级用户以外的用户需要具有 RBAC 授权。请参见下表。

子命令	RBAC 授权
create	solaris.cluster.modify
delete	solaris.cluster.modify
list	solaris.cluster.read

子命令	RBAC 授权
set	solaris.cluster.modify
show	solaris.cluster.read

名称

clquorum, clq — 管理 Oracle Solaris Cluster 法定设备和属性

```
/usr/cluster/bin/clquorum -V
/usr/cluster/bin/clquorum subcommand -?
/usr/cluster/bin/clquorum subcommand [options] -v devicename[...]
/usr/cluster/bin/clquorum add [-a] [-t type] [-p name=value [,...]] devicename[...]
/usr/cluster/bin/clquorum add -i {- | clconfigfile} [-t type] [-p name=value[, ...]] {+
| devicename[...]}
/usr/cluster/bin/clquorum disable [-t type[, ...]] {+ | devicename...}
/usr/cluster/bin/clquorum enable [-t type[, ...]] {+ | devicename[...]}
/usr/cluster/bin/clquorum export [-o {- | clconfigfile}] [-t type[, ...]] {+ | devicename[...]}
/usr/cluster/bin/clquorum list [-t type[, ...]] [-n node[, ...]] {+ | devicename[...]}
/usr/cluster/bin/clquorum remove -F [-t type[, ...]]
{+ | devicename[...]}
/usr/cluster/bin/clquorum reset
/usr/cluster/bin/clquorum show [-t type[, ...]] [-n node[, ...]]
{+ | devicename[...]}
/usr/cluster/bin/clquorum status [-t type[, ...]] [-n node[, ...]]
{+ | devicename[...]}

```

clquorum 命令可管理群集法定设备和群集法定属性。clq 命令是 clquorum 命令的简短格式。clquorum 命令和 clq 命令是相同的。您可以使用此命令的任一种格式。

此命令的一般格式如下所示：

```
clquorum [subcommand] [options] [operands]
```

仅当 *options* 指定 -? 选项或 -v 选项时，才可以省略 *subcommand*。

此命令的每个选项都有长格式和简短格式两种格式。在本手册页的 选项 部分给出了每个选项的两种格式，以及对该选项的描述。

法定设备是在发生裂脑和失忆情况时保护群集所必需的。（有关裂脑和失忆情况的信息，请参见《Oracle Solaris Cluster Concepts Guide》中有关法定和法定设备的章节。）每个法定设备必须通过 SCSI 电缆或 IP 网络连接到至少两个节点。

法定设备可以是共享的 SCSI 存储设备、共享的 NAS 存储设备 (Oracle ZFS Storage Appliance) 或法定服务器。如果法定设备存储着用户数据，则将上述设备作为法定设备

进行添加或删除时，存储的数据不会受影响。但是，如果使用的是复制的存储设备，则法定设备必须位于非复制卷上。

除非节点和法定设备处于维护状态，否则，节点和法定设备都会参与群集法定的构成。如果某个节点或法定设备处于维护状态，则其选票计数始终为零，并且不参与法定构成。

可以使用 `clquorum` 命令来执行以下任务：

- 将法定设备添加到 Oracle Solaris Cluster 配置中
- 从 Oracle Solaris Cluster 配置中删除法定设备
- 管理法定属性

子命令

支持以下子命令：

`add`

将指定共享设备作为法定设备添加。

仅可以在全局区域中运行该子命令。

每个法定设备必须至少与群集中的两个节点相连接。在群集配置中添加的法定设备包含到该设备所连接的每个节点的连接路径。以后，如果法定设备与群集节点之间的连接发生变化，您必须更新路径。可以通过删除法定设备然后重新将其添加到配置中来更新路径。如果添加了更多与法定设备连接的节点，或者断开了法定设备与一个或多个节点的连接，则可能会出现这种情况。有关法定管理的更多信息，请参见 [Unresolved link to "Oracle Solaris Cluster 系统管理指南 中的第 6 章 管理法定"](#)。

法定设备有多种类型。有关完整说明，请参见“选项”部分中的 `-t` 选项。`shared_disk` 类型是默认类型。

超级用户以外的用户需要具有 `solaris.cluster.modify` 基于角色的访问控制 (role-based access control, RBAC) 授权。请参见 [Unresolved link to "rbac5"](#)。

另请参见 `remove` 子命令的描述。

`disable`

将法定设备或节点置于法定维护状态。

仅可以在全局区域中运行该子命令。

在维护状态下，共享的设备或节点的选票计数为零。该共享设备或节点不再参与法定构成。此外，如果某个节点处于维护状态，那么，连接到该节点的任何法定设备的选票计数都会减一。

当您需要长时间关闭某个节点或设备以进行维护时，此功能十分有用。当某个节点重新引导到群集中后，除非设置了 `installmode`，否则，该节点会使自身脱离维护模式。

必须先关闭节点，然后才能将该节点置于维护状态。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

另请参见 `enable` 子命令的描述。

`enable`

使法定设备或节点脱离法定维护状态。

仅可以在全局区域中运行该子命令。

`enable` 子命令可使法定设备或节点脱离维护模式。该子命令可将法定设备或节点的已配置法定选票计数重置为默认值。此时，共享的设备或节点可以参与法定构成。

重置法定设备后，法定设备的选票计数将更改为 $N-1$ 。在此计算式中， N 是与设备连接的、选票计数非零的节点的数目。重置节点后，选票计数将重置为默认值。此时，与该节点连接的法定设备的选票计数将会加一。

除非启用了安装模式设置 `installmode`，否则，在引导时会自动启用每个节点的法定配置。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

另请参见 `disable` 子命令的描述。

`export`

导出群集法定的配置信息。

仅可以在全局区域中运行该子命令。

如果使用 `-o` 选项指定一个文件，则会将配置信息写入该文件。如果未指定文件，则会将信息写入标准输出 (`stdout`)。

`export` 子命令不会修改任何群集配置数据。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

`list`

显示群集中配置的法定设备的名称。

可以在全局区域中使用此子命令的所有格式。在区域群集中，只能结合 `-?` 或 `-v` 选项使用此子命令，或者不带任何选项使用此子命令。

如果不指定选项，`list` 子命令将显示群集中配置的所有法定设备。如果指定 `-t` 选项，该子命令只显示指定类型的法定设备。如果指定 `-n` 选项，该子命令将显示连接到任一指定节点的所有法定设备的名称。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

`remove`

从 Oracle Solaris Cluster 法定配置中删除指定的一个或多个法定设备。

仅可以在全局区域中运行该子命令。

结合强制选项 `-F` 使用此子命令可以删除双节点群集中的最后一个法定设备。如果不指定 `-F` 选项，`remove` 子命令将不会删除双节点群集的最后一个法定设备。

`remove` 子命令不会断开并删除物理设备。该子命令也不会影响设备上的任何用户数据（如果存在）。除非启用了 `installmode`，否则无法删除双节点群集中的最后一个法定设备。

只能删除法定设备。无法使用此子命令删除群集节点。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

另请参见 `add` 子命令的描述。

reset

将整个法定配置重置为默认选票计数。

仅可以在全局区域中运行该子命令。

如果启用了 `installmode`，则可以通过重置来清除该模式。除非已成功配置了至少一个法定设备，否则无法在双节点群集上重置 `installmode`。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

有关 `installmode` 属性的描述，另请参见 [cluster\(1CL\) \[461\]](#) 中的 `-p` 选项。

show

显示法定设备的属性。

仅可以在全局区域中运行该子命令。

如果不指定选项，`show` 子命令将显示群集中所有法定设备的属性。

如果使用 `-t` 选项指定了类型，该子命令只显示该类型设备的属性。请参见“选项”中的 `-t`。

如果使用 `-n` 选项指定了节点，该子命令将显示连接到任一指定节点的法定设备的属性。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

status

检查并显示法定设备的当前状态和选票计数。

可以在全局区域中使用此子命令的所有格式。在区域群集中，只能结合 `-?` 或 `-v` 选项使用此子命令，或者不带任何选项使用此子命令。

可以在全局区域中使用此子命令即时检查连接到指定节点的法定设备的状态。对于未连接到该节点的法定设备，此子命令将显示上次群集重新配置期间值为 `true` 的状态。

如果不指定选项，`status` 子命令将显示群集中所有法定设备的相关信息。

如果使用 `-t` 选项指定了类型，该子命令只显示该类型设备的相关信息。请参见“选项”中的 `-t`。

如果使用 `-n` 选项指定了节点，该子命令将显示连接到任一指定节点的法定设备的属性。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

支持以下选项：

`-?`

`--help`

显示帮助信息。使用该选项时，不会执行任何其他处理。

指定此选项时，子命令可有可无。

如果仅指定此选项而不使用子命令，将会显示此命令的子命令列表。

如果使用子命令且指定此选项，将会显示该子命令的用法选项。

`-a`

`--autoconfig`

对于使用共享磁盘的双节点群集，如果未配置任何法定设备，将自动选择并配置一个法定设备。

该群集中的所有共享磁盘都必须符合成为法定设备的条件。`autoconfig` 子命令不会检查某个可用设备是否符合成为法定设备的条件。`autoconfig` 子命令只针对共享磁盘执行检查。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权。请参见 [Unresolved link to "rbac5"](#) 手册页。

`-F`

强制删除指定的法定设备。

该强制选项仅可与 `remove` 子命令一起使用。使用强制选项可以删除双节点群集中的最后一个法定设备，或者删除某个发生故障的法定设备。结合此选项使用 `remove` 子命令时，在删除过程中，法定子系统不会处理法定设备。

`-i clconfigfile`

`--input=clconfigfile`

`--input clconfigfile`

指定用于管理法定设备的配置信息。此信息必须符合 [clconfiguration\(5CL\) \[1245\]](#) 手册页中定义的格式。

将 `-i` 与某个子命令以及其他命令行选项一起使用时，命令行选项的参数将会覆盖配置文件中的设置。

`-n node-name`
`--node=node-name`
`--node node-name`

指定法定设备连接到的节点名称。在 `list`、`status` 和 `show` 子命令中使用此选项可以使显示的信息仅限于连接到指定节点的那些法定设备。

可为 `node-name` 指定节点名称或节点 ID。

`-o {- | clconfigfile}`
`--output={- | clconfigfile}`
`--output {- | clconfigfile}`

将法定设备配置信息写入到文件或标准输出 (stdout)。此配置信息的格式符合 [clconfiguration\(5CL\) \[1245\]](#) 手册页中描述的格式。要指定标准输出，请指定 `-` 而不是文件名。

`-p name=value[...]`
`--property=name=value[...]`
`--property name=value[...]`

指定特定于某个设备类型的法定设备属性。将此选项与 `add` 子命令一起使用。有关这些属性的列表和描述，请参见 `-t` 选项的描述。

`-t device-type`
`--type=device-type`
`--type device-type`

指定法定设备类型。当指定了此选项时，操作数必须属于指定的类型。

对于 `add`、`export` 和 `remove` 子命令，当前支持的法定设备类型如下：

- 共享的本地磁盘，由 `shared_disk` 指定，可以是 SCSI-2、SCSI-3 或软件法定设备（禁用了隔离的 SCSI 磁盘）
- 在 Oracle Solaris Cluster 法定服务器计算机上运行的法定服务器进程，由 `quorum_server` 指定

默认类型为 `shared_disk`。

`add` 子命令不接受将 `-t node` 用作法定类型。

对于 `enable`、`disable`、`list`、`show` 和 `status` 子命令，可以指定类型 `node`、`shared_disk` 或 `quorum_server`。这些不同类型的法定设备具有以下属性：

`node`

无需设置特定属性即可让节点参与法定构成。

此类型只能与 `enable`、`disable`、`list`、`status` 和 `show` 子命令一起使用。不能用来添加 `node` 类型的法定设备。

`quorum_server`

`quorum_server` 类型的法定设备具有以下属性：

`qshost=quorum-server-host` : 指定运行法定服务器的计算机名称。此主机可以是计算机的 IP 地址,也可以是网络上的主机名。如果指定主机名,则必须在 `/etc/hosts` 文件和/或 `/etc/inet/ipnodes` 文件中指定计算机的 IP 地址。

`port=port` : 指定法定服务器与群集节点通信所用的端口号。

在添加法定服务器之前,必须先主机上安装法定服务器软件,并启动和运行法定服务器。有关详细信息,请参阅[Unresolved link to " Oracle Solaris Cluster 软件安装指南 "](#)。

`shared_disk`

使用此类型可配置 SCSI-2、SCSI-3 或软件法定设备。对于 `shared_disk` 法定设备,无需设置特定属性。`autoconfig` 子命令仅接受此法定设备类型。

`-V`

`--version`

显示该命令的版本。

请不要将此选项与其他子命令、选项或操作数一起指定。这些子命令、选项或操作数将被忽略。`-v` 选项只显示命令的版本。不会执行任何其他操作。

`-v`

`--verbose`

将详细信息显示到标准输出 (stdout)。

支持以下操作数 :

`devicename`

对于 `add`、`export` 和 `remove` 子命令,操作数是共享磁盘 (SCSI、法定服务器或 NAS 法定设备) 的名称。对于 `add` 子命令,如果未使用 `-i` 指定 `clconfigurationfile`,则必须至少将一个法定设备指定为操作数。

对于 `disable`、`enable`、`list`、`status` 和 `show` 子命令,操作数可以是节点的名称,也可以是共享磁盘 (SCSI、法定服务器或 NAS 法定设备) 的名称。

在任何情况下,如果指定了 `-t` 选项,则操作数类型必须与该选项的值匹配。

将以下值用作 `devicename` 操作数 :

- 对于节点,该操作数必须是节点名称或节点 ID。
- 对于 SCSI 法定设备,该操作数必须是设备标识符或完整的 DID 路径名,例如 `d1` 或 `/dev/did/rdisk/d1`。
- 对于法定服务器法定设备,该操作数必须指定一个或多个法定服务器的标识符。该标识符可以是法定服务器实例名称,并且在所有法定设备中必须是唯一的。

`+`

对于 `disable`、`enable`、`list`、`status` 和 `show` 子命令,指定为群集配置的所有法定设备。如果使用 `-t` 选项,加号 (+) 操作数可指定该类型的所有设备。

如果命令对于所有指定的操作数均成功执行，则返回零 (CL_NOERR)。如果某个操作数出错，命令会继续处理操作数列表中的下一个操作数。返回的退出代码始终反映最先发生的错误。

可能会返回以下退出值：

0 CL_NOERR

没有错误

您发出的命令已成功完成。

1 CL_ENOMEM

交换空间不足

某个群集节点耗尽了交换内存或其他操作系统资源。

3 CL_EINVAL

参数无效

您键入的命令不正确，或者使用 -i 选项提供的群集配置信息语法不正确。

6 CL_EACCESS

权限被拒绝

指定的对象不可访问。要发出命令，可能需要拥有超级用户或 RBAC 访问权限。有关更多信息，请参见 [Unresolved link to " su1M"](#) 和 [Unresolved link to " rbac5"](#) 手册页。

18 CL_EINTERNAL

遇到内部错误

内部错误表示软件缺陷或其他缺陷。

35 CL_EIO

I/O 错误

发生了物理输入/输出错误。

36 CL_ENOENT

没有这样的对象

由于以下某个原因，找不到您所指定的对象：

- 该对象不存在。
- 您尝试使用 -o 选项创建的配置文件路径中的某个目录不存在。
- 您尝试使用 -i 选项访问的配置文件包含错误。

39 CL_EEXIST

对象已存在

您所指定的设备、设备组、群集互连组件、节点、群集、资源、资源类型、资源组或专用字符串已存在。

41 CL_ETYPE

类型无效

您使用 -t 或 -p 选项指定的类型不存在。

例 114 添加 SCSI 法定设备

以下 clquorum 命令将配置连接到所有群集节点的某个 SCSI 法定设备。

```
# clquorum add /dev/did/rdisk/d4s2
```

使用 add 子命令时，shared_disk 类型是默认类型。要添加 shared_disk 法定设备，不需要指定 -t shared_disk。

例 115 添加法定服务器

以下 clquorum 命令将配置法定服务器 qs1：

```
# clquorum add -t quorum_server -p qshost=10.11.114.81 -p port=9000 qs1
```

例 116 删除法定设备

以下 clquorum 命令将删除 d4 法定设备。

```
# clquorum remove d4
```

无论法定设备的类型是 shared_disk 还是 quorum_server，用于删除该设备的命令都是相同的。

例 117 将法定设备置于维护状态

以下 clquorum 命令将法定设备 qs1 置于维护状态，并验证该设备是否处于维护状态。

```
# clquorum disable qs1
```

```
# clquorum status qs1
```

```
=== Cluster Quorum ===
```

```
--- Quorum Votes by Device ---
```

Device Name	Present	Possible	Status
qs1	1	1	Offline

例 118 重置法定设备的法定选票数

以下 `clquorum` 命令将法定设备 d4 的已配置法定选票计数重置为默认值。

```
# clquorum enable d4
```

例 119 显示群集中配置的法定设备

以下 `clquorum` 命令以简洁格式和详细格式显示法定设备。

```
# clquorum list
d4
pcow1
pcow2

# clquorum list -v
Quorums      Type
-----      ----
d4            shared_disk
pcow1        node
pcow2        node
```

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

[Intro\(1CL\) \[17\]](#)、[cluster\(1CL\) \[461\]](#)、[clconfiguration\(5CL\) \[1245\]](#)

超级用户可以运行此命令的所有格式。

任何用户都可以运行带有以下选项的此命令：

- `-?` 选项
- `-v` 选项

要运行带有其他子命令的此命令，超级用户以外的用户需要具有 RBAC 授权。请参见下表。

子命令	RBAC 授权
<code>add</code>	<code>solaris.cluster.modify</code>
<code>disable</code>	<code>solaris.cluster.modify</code>

子命令	RBAC 授权
enable	solaris.cluster.modify
export	solaris.cluster.read
list	solaris.cluster.read
remove	solaris.cluster.modify
reset	solaris.cluster.modify
show	solaris.cluster.read
status	solaris.cluster.read

名称

clquorum, clq — 管理 Oracle Solaris Cluster 法定设备和属性

```
/usr/cluster/bin/clquorum -V
/usr/cluster/bin/clquorum subcommand -?
/usr/cluster/bin/clquorum subcommand [options] -v devicename[...]
/usr/cluster/bin/clquorum add [-a] [-t type] [-p name=value [,...]] devicename[...]
/usr/cluster/bin/clquorum add -i {- | clconfigfile} [-t type] [-p name=value[, ...]] {+
| devicename[...]}
/usr/cluster/bin/clquorum disable [-t type[, ...]] {+ | devicename...}
/usr/cluster/bin/clquorum enable [-t type[, ...]] {+ | devicename[...]}
/usr/cluster/bin/clquorum export [-o {- | clconfigfile}] [-t type[, ...]] {+ | devicename[...]}
/usr/cluster/bin/clquorum list [-t type[, ...]] [-n node[, ...]] {+ | devicename[...]}
/usr/cluster/bin/clquorum remove -F [-t type[, ...]]
{+ | devicename[...]}
/usr/cluster/bin/clquorum reset
/usr/cluster/bin/clquorum show [-t type[, ...]] [-n node[, ...]]
{+ | devicename[...]}
/usr/cluster/bin/clquorum status [-t type[, ...]] [-n node[, ...]]
{+ | devicename[...]}

```

clquorum 命令可管理群集法定设备和群集法定属性。clq 命令是 clquorum 命令的简短格式。clquorum 命令和 clq 命令是相同的。您可以使用此命令的任一种格式。

此命令的一般格式如下所示：

```
clquorum [subcommand] [options] [operands]
```

仅当 *options* 指定 -? 选项或 -v 选项时，才可以省略 *subcommand*。

此命令的每个选项都有长格式和简短格式两种格式。在本手册页的 选项 部分给出了每个选项的两种格式，以及对该选项的描述。

法定设备是在发生裂脑和失忆情况时保护群集所必需的。（有关裂脑和失忆情况的信息，请参见《Oracle Solaris Cluster Concepts Guide》中有关法定和法定设备的章节。）每个法定设备必须通过 SCSI 电缆或 IP 网络连接到至少两个节点。

法定设备可以是共享的 SCSI 存储设备、共享的 NAS 存储设备 (Oracle ZFS Storage Appliance) 或法定服务器。如果法定设备存储着用户数据，则将上述设备作为法定设备

进行添加或删除时，存储的数据不会受影响。但是，如果使用的是复制的存储设备，则法定设备必须位于非复制卷上。

除非节点和法定设备处于维护状态，否则，节点和法定设备都会参与群集法定的构成。如果某个节点或法定设备处于维护状态，则其选票计数始终为零，并且不参与法定构成。

可以使用 `clquorum` 命令来执行以下任务：

- 将法定设备添加到 Oracle Solaris Cluster 配置中
- 从 Oracle Solaris Cluster 配置中删除法定设备
- 管理法定属性

子命令

支持以下子命令：

`add`

将指定共享设备作为法定设备添加。

仅可以在全局区域中运行该子命令。

每个法定设备必须至少与群集中的两个节点相连接。在群集配置中添加的法定设备包含到该设备所连接的每个节点的连接路径。以后，如果法定设备与群集节点之间的连接发生变化，您必须更新路径。可以通过删除法定设备然后重新将其添加到配置中来更新路径。如果添加了更多与法定设备连接的节点，或者断开了法定设备与一个或多个节点的连接，则可能会出现这种情况。有关法定管理的更多信息，请参见 [Unresolved link to "Oracle Solaris Cluster 系统管理指南 中的第 6 章 管理法定"](#)。

法定设备有多种类型。有关完整说明，请参见“选项”部分中的 `-t` 选项。`shared_disk` 类型是默认类型。

超级用户以外的用户需要具有 `solaris.cluster.modify` 基于角色的访问控制 (role-based access control, RBAC) 授权。请参见 [Unresolved link to "rbac5"](#)。

另请参见 `remove` 子命令的描述。

`disable`

将法定设备或节点置于法定维护状态。

仅可以在全局区域中运行该子命令。

在维护状态下，共享的设备或节点的选票计数为零。该共享设备或节点不再参与法定构成。此外，如果某个节点处于维护状态，那么，连接到该节点的任何法定设备的选票计数都会减一。

当您需要长时间关闭某个节点或设备以进行维护时，此功能十分有用。当某个节点重新引导到群集中后，除非设置了 `installmode`，否则，该节点会使自身脱离维护模式。

必须先关闭节点，然后才能将该节点置于维护状态。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

另请参见 `enable` 子命令的描述。

`enable`

使法定设备或节点脱离法定维护状态。

仅可以在全局区域中运行该子命令。

`enable` 子命令可使法定设备或节点脱离维护模式。该子命令可将法定设备或节点的已配置法定选票计数重置为默认值。此时，共享的设备或节点可以参与法定构成。

重置法定设备后，法定设备的选票计数将更改为 $N-1$ 。在此计算式中， N 是与设备连接的、选票计数非零的节点的数目。重置节点后，选票计数将重置为默认值。此时，与该节点连接的法定设备的选票计数将会加一。

除非启用了安装模式设置 `installmode`，否则，在引导时会自动启用每个节点的法定配置。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

另请参见 `disable` 子命令的描述。

`export`

导出群集法定的配置信息。

仅可以在全局区域中运行该子命令。

如果使用 `-o` 选项指定一个文件，则会将配置信息写入该文件。如果未指定文件，则会将信息写入标准输出 (`stdout`)。

`export` 子命令不会修改任何群集配置数据。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

`list`

显示群集中配置的法定设备的名称。

可以在全局区域中使用此子命令的所有格式。在区域群集中，只能结合 `-?` 或 `-v` 选项使用此子命令，或者不带任何选项使用此子命令。

如果不指定选项，`list` 子命令将显示群集中配置的所有法定设备。如果指定 `-t` 选项，该子命令只显示指定类型的法定设备。如果指定 `-n` 选项，该子命令将显示连接到任一指定节点的所有法定设备的名称。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

`remove`

从 Oracle Solaris Cluster 法定配置中删除指定的一个或多个法定设备。

仅可以在全局区域中运行该子命令。

结合强制选项 `-F` 使用此子命令可以删除双节点群集中的最后一个法定设备。如果不指定 `-F` 选项，`remove` 子命令将不会删除双节点群集中的最后一个法定设备。

`remove` 子命令不会断开并删除物理设备。该子命令也不会影响设备上的任何用户数据（如果存在）。除非启用了 `installmode`，否则无法删除双节点群集中的最后一个法定设备。

只能删除法定设备。无法使用此子命令删除群集节点。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

另请参见 `add` 子命令的描述。

reset

将整个法定配置重置为默认选票计数。

仅可以在全局区域中运行该子命令。

如果启用了 `installmode`，则可以通过重置来清除该模式。除非已成功配置了至少一个法定设备，否则无法在双节点群集上重置 `installmode`。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

有关 `installmode` 属性的描述，另请参见 [cluster\(1CL\) \[461\]](#) 中的 `-p` 选项。

show

显示法定设备的属性。

仅可以在全局区域中运行该子命令。

如果不指定选项，`show` 子命令将显示群集中所有法定设备的属性。

如果使用 `-t` 选项指定了类型，该子命令只显示该类型设备的属性。请参见“选项”中的 `-t`。

如果使用 `-n` 选项指定了节点，该子命令将显示连接到任一指定节点的法定设备的属性。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

status

检查并显示法定设备的当前状态和选票计数。

可以在全局区域中使用此子命令的所有格式。在区域群集中，只能结合 `-?` 或 `-v` 选项使用此子命令，或者不带任何选项使用此子命令。

可以在全局区域中使用此子命令即时检查连接到指定节点的法定设备的状态。对于未连接到该节点的法定设备，此子命令将显示上次群集重新配置期间值为 `true` 的状态。

如果不指定选项，`status` 子命令将显示群集中所有法定设备的相关信息。

如果使用 `-t` 选项指定了类型，该子命令只显示该类型设备的相关信息。请参见“选项”中的 `-t`。

如果使用 `-n` 选项指定了节点，该子命令将显示连接到任一指定节点的法定设备的属性。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

支持以下选项：

`-?`

`--help`

显示帮助信息。使用该选项时，不会执行任何其他处理。

指定此选项时，子命令可有可无。

如果仅指定此选项而不使用子命令，将会显示此命令的子命令列表。

如果使用子命令且指定此选项，将会显示该子命令的用法选项。

`-a`

`--autoconfig`

对于使用共享磁盘的双节点群集，如果未配置任何法定设备，将自动选择并配置一个法定设备。

该群集中的所有共享磁盘都必须符合成为法定设备的条件。`autoconfig` 子命令不会检查某个可用设备是否符合成为法定设备的条件。`autoconfig` 子命令只针对共享磁盘执行检查。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权。请参见 [Unresolved link to "rbac5"](#) 手册页。

`-F`

强制删除指定的法定设备。

该强制选项仅可与 `remove` 子命令一起使用。使用强制选项可以删除双节点群集中的最后一个法定设备，或者删除某个发生故障的法定设备。结合此选项使用 `remove` 子命令时，在删除过程中，法定子系统不会处理法定设备。

`-i clconfigfile`

`--input=clconfigfile`

`--input clconfigfile`

指定用于管理法定设备的配置信息。此信息必须符合 [clconfiguration\(5CL\) \[1245\]](#) 手册页中定义的格式。

将 `-i` 与某个子命令以及其他命令行选项一起使用时，命令行选项的参数将会覆盖配置文件中的设置。

`-n node-name`
`--node=node-name`
`--node node-name`

指定法定设备连接到的节点名称。在 `list`、`status` 和 `show` 子命令中使用此选项可以使显示的信息仅限于连接到指定节点的那些法定设备。

可为 `node-name` 指定节点名称或节点 ID。

`-o {- | clconfigfile}`
`--output={- | clconfigfile}`
`--output {- | clconfigfile}`

将法定设备配置信息写入到文件或标准输出 (stdout)。此配置信息的格式符合 [clconfiguration\(5CL\) \[1245\]](#) 手册页中描述的格式。要指定标准输出，请指定 `-` 而不是文件名。

`-p name=value[...]`
`--property=name=value[...]`
`--property name=value[...]`

指定特定于某个设备类型的法定设备属性。将此选项与 `add` 子命令一起使用。有关这些属性的列表和描述，请参见 `-t` 选项的描述。

`-t device-type`
`--type=device-type`
`--type device-type`

指定法定设备类型。当指定了此选项时，操作数必须属于指定的类型。

对于 `add`、`export` 和 `remove` 子命令，当前支持的法定设备类型如下：

- 共享的本地磁盘，由 `shared_disk` 指定，可以是 SCSI-2、SCSI-3 或软件法定设备（禁用了隔离的 SCSI 磁盘）
- 在 Oracle Solaris Cluster 法定服务器计算机上运行的法定服务器进程，由 `quorum_server` 指定

默认类型为 `shared_disk`。

`add` 子命令不接受将 `-t node` 用作法定类型。

对于 `enable`、`disable`、`list`、`show` 和 `status` 子命令，可以指定类型 `node`、`shared_disk` 或 `quorum_server`。这些不同类型的法定设备具有以下属性：

`node`

无需设置特定属性即可让节点参与法定构成。

此类型只能与 `enable`、`disable`、`list`、`status` 和 `show` 子命令一起使用。不能用来添加 `node` 类型的法定设备。

`quorum_server`

`quorum_server` 类型的法定设备具有以下属性：

`qshost=quorum-server-host` : 指定运行法定服务器的计算机名称。此主机可以是计算机的 IP 地址,也可以是网络上的主机名。如果指定主机名,则必须在 `/etc/hosts` 文件和/或 `/etc/inet/ipnodes` 文件中指定计算机的 IP 地址。

`port=port` : 指定法定服务器与群集节点通信所用的端口号。

在添加法定服务器之前,必须先主机上安装法定服务器软件,并启动和运行法定服务器。有关详细信息,请参阅[Unresolved link to " Oracle Solaris Cluster 软件安装指南 "](#)。

`shared_disk`

使用此类型可配置 SCSI-2、SCSI-3 或软件法定设备。对于 `shared_disk` 法定设备,无需设置特定属性。`autoconfig` 子命令仅接受此法定设备类型。

`-V`

`--version`

显示该命令的版本。

请不要将此选项与其他子命令、选项或操作数一起指定。这些子命令、选项或操作数将被忽略。`-v` 选项只显示命令的版本。不会执行任何其他操作。

`-v`

`--verbose`

将详细信息显示到标准输出 (stdout)。

支持以下操作数 :

`devicename`

对于 `add`、`export` 和 `remove` 子命令,操作数是共享磁盘 (SCSI、法定服务器或 NAS 法定设备) 的名称。对于 `add` 子命令,如果未使用 `-i` 指定 `clconfigurationfile`,则必须至少将一个法定设备指定为操作数。

对于 `disable`、`enable`、`list`、`status` 和 `show` 子命令,操作数可以是节点的名称,也可以是共享磁盘 (SCSI、法定服务器或 NAS 法定设备) 的名称。

在任何情况下,如果指定了 `-t` 选项,则操作数类型必须与该选项的值匹配。

将以下值用作 `devicename` 操作数 :

- 对于节点,该操作数必须是节点名称或节点 ID。
- 对于 SCSI 法定设备,该操作数必须是设备标识符或完整的 DID 路径名,例如 `d1` 或 `/dev/did/rdisk/d1`。
- 对于法定服务器法定设备,该操作数必须指定一个或多个法定服务器的标识符。该标识符可以是法定服务器实例名称,并且在所有法定设备中必须是唯一的。

`+`

对于 `disable`、`enable`、`list`、`status` 和 `show` 子命令,指定为群集配置的所有法定设备。如果使用 `-t` 选项,加号 (+) 操作数可指定该类型的所有设备。

如果命令对于所有指定的操作数均成功执行，则返回零 (CL_NOERR)。如果某个操作数出错，命令会继续处理操作数列表中的下一个操作数。返回的退出代码始终反映最先发生的错误。

可能会返回以下退出值：

0 CL_NOERR

没有错误

您发出的命令已成功完成。

1 CL_ENOMEM

交换空间不足

某个群集节点耗尽了交换内存或其他操作系统资源。

3 CL_EINVAL

参数无效

您键入的命令不正确，或者使用 -i 选项提供的群集配置信息语法不正确。

6 CL_EACCESS

权限被拒绝

指定的对象不可访问。要发出命令，可能需要拥有超级用户或 RBAC 访问权限。有关更多信息，请参见 [Unresolved link to " su1M"](#) 和 [Unresolved link to " rbac5"](#) 手册页。

18 CL_EINTERNAL

遇到内部错误

内部错误表示软件缺陷或其他缺陷。

35 CL_EIO

I/O 错误

发生了物理输入/输出错误。

36 CL_ENOENT

没有这样的对象

由于以下某个原因，找不到您所指定的对象：

- 该对象不存在。
- 您尝试使用 -o 选项创建的配置文件路径中的某个目录不存在。
- 您尝试使用 -i 选项访问的配置文件包含错误。

39 CL_EEXIST

对象已存在

您所指定的设备、设备组、群集互连组件、节点、群集、资源、资源类型、资源组或专用字符串已存在。

41 CL_ETYPE

类型无效

您使用 -t 或 -p 选项指定的类型不存在。

例 120 添加 SCSI 法定设备

以下 clquorum 命令将配置连接到所有群集节点的某个 SCSI 法定设备。

```
# clquorum add /dev/did/rdisk/d4s2
```

使用 add 子命令时，shared_disk 类型是默认类型。要添加 shared_disk 法定设备，不需要指定 -t shared_disk。

例 121 添加法定服务器

以下 clquorum 命令将配置法定服务器 qs1：

```
# clquorum add -t quorum_server -p qshost=10.11.114.81 -p port=9000 qs1
```

例 122 删除法定设备

以下 clquorum 命令将删除 d4 法定设备。

```
# clquorum remove d4
```

无论法定设备的类型是 shared_disk 还是 quorum_server，用于删除该设备的命令都是相同的。

例 123 将法定设备置于维护状态

以下 clquorum 命令将法定设备 qs1 置于维护状态，并验证该设备是否处于维护状态。

```
# clquorum disable qs1
```

```
# clquorum status qs1
```

```
=== Cluster Quorum ===
```

```
--- Quorum Votes by Device ---
```

Device Name	Present	Possible	Status
qs1	1	1	Offline

例 124 重置法定设备的法定选票数

以下 `clquorum` 命令将法定设备 `d4` 的已配置法定选票计数重置为默认值。

```
# clquorum enable d4
```

例 125 显示群集中配置的法定设备

以下 `clquorum` 命令以简洁格式和详细格式显示法定设备。

```
# clquorum list
d4
pcow1
pcow2

# clquorum list -v
Quorums      Type
-----      ----
d4            shared_disk
pcow1        node
pcow2        node
```

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

[Intro\(1CL\) \[17\]](#)、[cluster\(1CL\) \[461\]](#)、[clconfiguration\(5CL\) \[1245\]](#)

超级用户可以运行此命令的所有格式。

任何用户都可以运行带有以下选项的此命令：

- `-?` 选项
- `-v` 选项

要运行带有其他子命令的此命令，超级用户以外的用户需要具有 RBAC 授权。请参见下表。

子命令	RBAC 授权
<code>add</code>	<code>solaris.cluster.modify</code>
<code>disable</code>	<code>solaris.cluster.modify</code>

子命令	RBAC 授权
enable	solaris.cluster.modify
export	solaris.cluster.read
list	solaris.cluster.read
remove	solaris.cluster.modify
reset	solaris.cluster.modify
show	solaris.cluster.read
status	solaris.cluster.read

名称

clreslogicalhostname, clrslh — 管理 Oracle Solaris Cluster 逻辑主机名的资源

```
/usr/cluster/bin/clreslogicalhostname [subcommand] -?  
  
/usr/cluster/bin/clreslogicalhostname -V  
  
/usr/cluster/bin/clreslogicalhostname [subcommand [options]] -v  
[lresource]...  
  
/usr/cluster/bin/clreslogicalhostname create -g resourcegroup  
[-h lhost[,...]] [-N netif@node[,...]] [-p name=value]  
[-Z {zoneclustername | global}] [-d] lresource  
  
/usr/cluster/bin/clreslogicalhostname create -i  
{- | clconfiguration} [-a] [-g resourcegroup[,...]] [-p  
name=value] [-d] {+ | lresource...}  
  
/usr/cluster/bin/clreslogicalhostname delete [-g resourcegroup[,...]]  
[-Z {zoneclustername | global}] [-F] {+ | lresource...}  
  
/usr/cluster/bin/clreslogicalhostname disable [-g resourcegroup[,...]]  
[-R] [-n node[,...]] [-Z {zoneclustername | global}]  
{+ | lresource...}  
  
/usr/cluster/bin/clreslogicalhostname enable [-g resourcegroup[,...]]  
[-R] [-n node[,...]] [-Z {zoneclustername | global}]  
{+ | lresource...}  
  
/usr/cluster/bin/clreslogicalhostname export [-o {- | configfile}]  
{+ | lresource...}  
  
/usr/cluster/bin/clreslogicalhostname list [-s state[,...]]  
[-g resourcegroup[,...]] [-Z {zoneclustername  
[,...] | global | all}] {+ | lresource...}  
  
/usr/cluster/bin/clreslogicalhostname list-props [-l listtype]  
[-p name[,...]] [-Z {zoneclustername [,...] | global | all}]  
{+ | lresource...}  
  
/usr/cluster/bin/clreslogicalhostname monitor [-g resourcegroup[,...]]  
[-Z zoneclustername | all | global] {+ | lresource...}  
  
/usr/cluster/bin/clreslogicalhostname reset [-f errorflag] [-g  
resourcegroup[,...]] [-Z {zoneclustername | global}]  
{+ | lresource...}  
  
/usr/cluster/bin/clreslogicalhostname set [-i  
{- | clconfiguration}] [-g resourcegroup[,...]] [-p name  
{+|-}=value] [-Z {zoneclustername}] {+ | lresource...}  
  
/usr/cluster/bin/clreslogicalhostname show [-g resourcegroup[,...]]  
[-p name[,...]] [-Z {zoneclustername [,...] | global | all}]
```

```
[+ | lresource...]  
  
/usr/cluster/bin/clreslogicalhostname status [-s state[,...]]  
    [-n node[,...]] [-g resourcegroup[,...]] [-Z  
    {zoneclustername [,...] | global | all}] [+ | lresource...]  
  
/usr/cluster/bin/clreslogicalhostname unmonitor [-g  
    resourcegroup[,...]] [-Z {zoneclustername | global}]  
    [+ | lresource...]
```

`clreslogicalhostname` 命令可管理 Oracle Solaris Cluster 逻辑主机名的资源。`clrslh` 命令是 `clreslogicalhostname` 命令的简短格式。`clreslogicalhostname` 命令和 `clrslh` 命令是相同的。您可以使用此命令的任一种格式。

`clreslogicalhostname` 命令包含一些用于方便地创建逻辑主机名资源的内置选项。`clreslogicalhostname` 命令还支持自动创建 Solaris IP 多路径 (IP multipathing, IPMP) 组。

`clreslogicalhostname` 命令的一些子命令可修改资源配置。

- `disable`
- `enable`
- `monitor`
- `reset`
- `set`
- `unmonitor`

`clreslogicalhostname` 命令的某些子命令只获取有关资源的信息。可以从全局群集或区域群集使用这些子命令：以下命令只获取有关资源的信息：

- `export`
- `list`
- `list-props`
- `show`
- `status`

为了避免此命令产生不可预测的结果，请从全局群集节点运行该命令的所有格式。

此命令的一般格式如下所示：

```
clreslogicalhostname [subcommand] [options] [operands]
```

仅当 *options* 指定了 `-?`、`-o`、`-v` 或 `-V` 选项时，才能省略 *subcommand*。

此命令的每个选项都有长格式和简短格式两种格式。在本手册页的“选项”部分给出了每个选项的两种格式，以及对该选项的描述。

在区域群集中的操作

在区域群集中，可以将 `clreslogicalhostname` 命令与所有子命令（`export` 除外）一起使用。

也可以将 `-z` 选项与所有子命令（`export` 除外）一起使用来指定要将操作限制到的特定区域群集的名称。另外，还可以将区域群集名称附加到逻辑主机名资源 (`zoneclustername : lhresource`)，以便将操作限制到特定的区域群集。

您可以从全局群集节点访问所有区域群集信息，但无法从特定区域群集访问其他区域群集信息。如果不将操作限制到特定区域群集，则所用的子命令仅会在当前群集中运行。

支持以下子命令：

create

创建指定为命令操作数的逻辑主机名资源。

将 `create` 和 `-i` 选项一起使用以指定配置文件时，该子命令接受使用加号 (+) 作为操作数。使用 + 操作数时，将会创建配置文件中提供的尚不存在的所有资源。

使用 `create` 子命令之前，请确保 `/etc/netmasks` 文件具有所有逻辑主机名对应的 IP 地址子网和网络掩码条目。如有必要，请编辑 `/etc/netmasks` 文件以添加缺少的任何条目。

默认情况下，创建的资源处于启用状态，且启用了监视功能。但是，仅当资源的资源组联机后，该资源才会联机并受到监视。要在禁用状态下创建资源，请指定 `-d` 选项。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集在区域群集中创建逻辑主机名资源，请使用 `-z` 选项指定该区域群集的名称。

超级用户以外的用户需要具有 `solaris.cluster.modify` 基于角色的访问控制 (Role-Based Access Control, RBAC) 授权才能使用该子命令。

另请参见 `delete` 子命令的描述。

delete

删除指定为命令操作数的逻辑主机名资源。此子命令接受使用加号 (+) 作为操作数来指定删除所有资源。

`-g` 选项可以过滤操作数列表，以限制删除的资源。`-g` 选项只会删除操作数列表内属于 `resourcegrouplist` 中资源组的成员的资源。

- 默认情况下，仅当符合以下条件时，才会删除某个资源：
 - 已禁用该资源。
 - 已消除该资源的所有依赖性。
- 要确保删除所有指定的资源，请指定 `-F` 选项。`-F` 选项的效果如下所示：

- 删除所有指定的资源，即使是未禁用的资源。
- 从其他资源的资源依赖性设置中删除所有指定的资源。

将按照满足资源间依赖性所需的顺序禁用资源，而不管资源在命令行上的指定顺序。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集删除区域群集中的逻辑主机名资源，请使用 `-z` 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

另请参见 `create` 子命令的描述。

disable

禁用指定为命令操作数的逻辑主机名资源。此子命令接受使用加号 (+) 作为操作数来指定禁用所有资源。

`-g` 选项可以过滤操作数列表，以限制禁用的资源。`-g` 选项只会禁用操作数列表内属于 `resourcegroup` 中资源组的成员的资源。

要确保满足所有必需的资源依赖性，请指定 `-R` 选项。`-R` 选项将会禁用依赖于指定为命令操作数的资源的任何资源，即使这些资源未指定为命令操作数。`-g` 选项和 `-t` 选项不适用于仅为满足资源依赖性而禁用的资源。

将按照满足资源间依赖性所需的顺序禁用资源，而不管资源在命令行上的指定顺序。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集禁用在区域群集中注册的逻辑主机名资源，请使用 `-z` 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。

另请参见 `enable` 子命令的描述。

enable

启用指定为命令操作数的逻辑主机名资源。此子命令接受使用加号 (+) 作为操作数来指定启用所有资源。

`-g` 选项可以过滤操作数列表，以限制启用的资源。`-g` 选项仅启用操作数列表内属于 `resourcegroup` 中资源组的成员的资源。

要确保满足所有必需的资源依赖性，请指定 `-R` 选项。`-R` 选项将会启用依赖于指定为命令操作数的资源的任何资源，即使这些资源未指定为命令操作数。`-g` 选项不适用于仅为满足资源依赖性而启用的资源。

将按照满足资源间依赖性所需的顺序禁用资源，而不管资源在命令行上的指定顺序。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集启用在区域群集中注册的逻辑主机名资源，请使用 `-z` 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。

另请参见 `disable` 子命令的描述。

export

以 `clconfiguration(5CL) [1245]` 手册页所述的格式导出逻辑主机名资源配置。

只能在全局群集中使用该子命令。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

list

显示指定为命令操作数的逻辑主机名资源的列表。默认情况下，会显示所有资源。

`-g` 选项可以过滤操作数列表，以限制显示的资源。`-g` 选项只会显示操作数列表内属于 `resourcegroup` 中资源组的成员的资源。

此子命令接受使用加号 (+) 作为操作数来指定特定资源组中的所有资源，或者指定属于特定资源类型实例的所有资源。如果未提供操作数，则会显示指定资源组中的所有资源，或显示属于指定资源类型的实例的所有资源。

如果指定 `-v` 选项，还将显示列表中每个资源的资源组和资源类型。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集查看在区域群集中注册的逻辑主机名资源，请使用 `-z` 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

list-props

显示指定为命令操作数的逻辑主机名资源的属性列表。默认情况下，会显示所有资源的扩展属性。

以下选项可以过滤操作数列表，以限制要显示其属性的资源：

`-g resourcegroup` 仅显示操作数列表内属于 `resourcegroup` 中资源组成员的逻辑主机名资源的属性。

`-l` 选项指定要显示的资源属性的类型：

`-l all` 指定显示标准属性和扩展属性。

`-l extension` 指定只显示扩展属性。默认情况下，只显示扩展属性。

`-l standard` 指定只显示标准属性。

如果未指定 `-l` 选项，将只显示扩展属性，除非使用 `-p` 选项或 `-y` 选项明确指定标准属性。

`-p` 选项用于限制要显示的资源属性集。`-p` 选项只显示 `namelist` 中指定的属性。您可以在 `namelist` 中指定标准属性和扩展属性。

如果指定 `-v` 选项，还将显示每个属性的描述。

此子命令接受使用加号 (+) 作为操作数来指定特定资源组中的所有资源，或者指定属于特定资源类型实例的所有资源。如果未提供操作数，则会显示指定资源组中所有资源的属性，或显示属于指定资源类型的实例的所有资源的属性。

您可以在全局群集中或区域群集中使用该子命令。

要查看区域群集中的逻辑主机名资源的属性列表，请使用 `-z` 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

monitor

开启对指定为命令操作数的逻辑主机名资源的监视。此子命令接受使用加号 (+) 作为操作数来指定开启对所有资源的监视。

`-g` 选项可以过滤操作数列表，以限制监视的资源。`-g` 选项只会监视操作数列表内属于 `resourcegroup` 中资源组的成员的资源。

- 开启对某个资源的监视后，仅当符合以下条件时，才会监视该资源：
- 已启用该资源。
- 包含该资源的资源组至少已在一个群集节点上联机。

注 - 开启对某个资源的监视不会启用该资源。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集监视区域群集中的资源，请使用 `-z` 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。

另请参见 `unmonitor` 子命令的描述。

reset

清除与指定为命令操作数的逻辑主机名资源关联的错误标志。此子命令接受使用加号 (+) 作为操作数来指定清除所有资源的错误标志。

`-g` 选项可以过滤操作数列表，以限制重置的资源。`-g` 选项只会重置操作数列表内属于 `resourcegroup` 中资源组的成员的资源。

默认情况下，`reset` 子命令将会清除 `STOP_FAILED` 错误标志。要显式指定要清除的错误标志，请使用 `-f` 选项。`-f` 选项接受的唯一错误标志是 `STOP_FAILED` 错误标志。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集重置区域群集中的逻辑主机名资源，请使用 `-z` 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。

set

修改指定为命令操作数的逻辑主机名资源的特定属性。此子命令接受使用加号 (+) 作为操作数来指定修改所有资源的特定属性。

`-g` 选项可以过滤操作数列表，以限制修改的资源。`-g` 选项只会修改操作数列表内属于 `resourcegroup` 中资源组的成员的资源。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集设置区域群集中逻辑主机名资源的属性，请使用 `-z` 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

show

显示指定为命令操作数的逻辑主机名资源的配置。默认情况下，会显示所有资源的配置。

`-g` 选项可以过滤操作数列表，以限制要显示其配置的资源。`-g` 选项只会显示操作数列表内属于 `resourcegroup` 中资源组成员的资源的配置。

`-p` 选项用于限制要显示的资源属性集。`-p` 选项只显示 `namelist` 中指定的属性。您可以在 `namelist` 中指定标准属性和扩展属性。

此子命令接受使用加号 (+) 作为操作数来指定特定资源组中的所有资源，或者指定属于特定资源类型实例的所有资源。如果未提供操作数，则会显示指定资源组中所有资源的配置，或显示属于指定资源类型实例的所有资源的配置。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集查看区域群集中逻辑主机名资源的配置，请使用 `-z` 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

status

显示指定为命令操作数的逻辑主机名资源的状态。默认情况下，会显示所有资源的状态。

以下选项可以过滤操作数列表，以限制要显示其状态的资源的列表：

`-g resourcegroup` 仅显示操作数列表内属于 `resourcegroup` 中资源组成员的资源的状态。

`-n nodelist` 仅显示操作数列表内承载在 `nodelist` 中节点上的资源的状态。

`-s statelist`

仅显示操作数列表内处于 *statelist* 中的状况的资源的状态。

此子命令接受使用加号 (+) 作为操作数来指定特定资源组中的所有资源，或者指定属于特定资源类型实例的所有资源。如果未提供操作数，则会显示指定资源组中所有资源的状态，或显示属于指定资源类型实例的所有资源的状态。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集查看区域群集中逻辑主机名资源的状态，请使用 `-z` 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

`unmonitor`

关闭对指定为命令操作数的逻辑主机名资源的监视。此子命令接受使用加号 (+) 作为操作数来指定关闭对所有资源的监视。

如果关闭对某个被禁用资源的监视，该资源不会受到影响。该资源及其监视器已脱机。

注 - 关闭对某个资源的监视不会禁用该资源。但是，当您禁用某个资源时，不需要关闭对该资源的监视。禁用的资源及其监视器将保持脱机状态。

`-g` 选项可以过滤操作数列表，以限制要对其关闭监视的资源。`-g` 选项将对操作数列表内属于 *resourcegroup* 中资源组的成员的资源关闭监视。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集关闭对区域群集中逻辑主机名资源的监视，请使用 `-z` 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。

另请参见 `disable` 子命令和 `monitor` 子命令的描述。

支持以下选项：

`-?`

`--help`

显示帮助信息。使用该选项时，不会执行任何其他处理。

指定此选项时，子命令可有可无。

如果仅指定此选项而不使用子命令，将会显示此命令的子命令列表。

如果使用子命令且指定此选项，将会显示该子命令的用法选项。

此选项与特定子命令一起使用的效果如下所示：

create

与 -g 选项一起指定时，此选项将会显示指定资源组的所有资源属性的帮助信息。

set

显示指定为命令操作数的资源的属性的帮助信息。

-a

--automatic

当基于群集配置信息创建资源时，将自动执行以下附加操作：

- 注册资源类型
- 创建资源组
- 创建操作数列表中指定的资源所依赖的资源

- 群集配置信息包含的信息必须足以执行以下所有操作：
 - 启用要注册的资源类型
 - 启用要创建的资源组
 - 启用要创建的资源

此选项仅可以与 create 子命令一同指定。如果指定此选项，还必须指定 -i 选项并提供配置文件。

-d

--disable

创建资源后禁用该资源。此选项仅可以与 create 子命令一同指定。默认情况下，资源在启用状态下创建。

启用某个资源并不保证该资源联机。仅当资源所在的资源组已在至少一个节点上联机后，该资源才会联机。

-f *errorflag*

--flag *errorflag*

显式指定要通过 reset 子命令清除的错误标志。此选项仅可以与 reset 子命令一同指定。默认情况下，reset 子命令将会清除 STOP_FAILED 错误标志。

-f 选项接受的唯一错误标志是 STOP_FAILED 错误标志。

-F

--force

强制删除未禁用的资源。此选项仅可以与 delete 子命令一同指定。

-g *resourcegroup*[,...]

--resourcegroup *resourcegroup*[,...]

指定一个资源组或资源组列表。

对于除 `create` 以外的子命令，该命令仅对操作数列表内属于 `-g` 选项指定的资源组成员的资源起作用。

此选项与特定子命令一起使用的效果如下所示：

`create` 指定在特定资源组中创建资源。将 `-g` 与 `create` 子命令一起使用时，只能指定一个资源组。

`-h /host[,...]`

`--logicalhost /host[,...]`

指定此资源代表的逻辑主机名的列表。当有多个逻辑主机名要与新的逻辑主机名资源相关联，或者当逻辑主机名与资源本身不同名时，必须使用 `-h` 选项。列表中的所有逻辑主机名必须位于同一子网中。如果未指定 `-h` 选项，该资源将代表与资源本身同名的单个逻辑主机名。

您可以使用 `-h`，而无需使用 `-p` 来设置 `HostnameList` 属性。但是，不能在同一命令中使用 `-h` 并显式设置 `HostnameList`。

`-h` 只能与 `create` 子命令一起使用。

注 - 对于区域群集，必须在区域群集配置中的全局范围内的网络属性中，指定所有逻辑主机名或相应的 IP 地址。否则，资源组创建将会失败。

有关全局范围网络属性的更多信息，请参阅 [clzonecluster\(1CL\) \[517\]](#) 手册页。

`-i {- | clconfiguration}`

`--input {- | clconfiguration}`

指定用于创建或修改逻辑主机名资源的配置信息。此信息必须符合 [clconfiguration\(5CL\) \[1245\]](#) 手册页中定义的格式。此信息可包含在文件中或者通过标准输入提供。要指定标准输入，请指定 `-` 而不是文件名。

仅会创建或修改作为命令操作数提供的资源。在命令中指定的选项将会覆盖配置信息中设置的任何选项。如果配置信息中缺少配置参数，则必须在命令行中指定这些参数。

此选项与特定子命令一起使用的效果如下所示：

`create` 与 `-a` 选项一起指定时，此选项将注册所有必需的资源类型，并创建所有必需的资源组。您必须提供注册和配置所需的全部信息。所有其他配置数据将被忽略。

`-l listtype`

`--listtype listtype`

指定要通过 `list-props` 子命令显示的资源属性的类型。此选项仅可以与 `list-props` 子命令一同指定。

必须根据以下列表为 `listtype` 指定一个值：

`all` 指定显示标准属性和扩展属性。

`extension` 指定只显示扩展属性。默认情况下，只显示扩展属性。

`standard` 指定只显示标准属性。

如果未指定 `-l` 选项，则只显示扩展属性，除非使用 `-p` 选项明确指定标准属性。

`-n node[,...]`

`--node node[,...]`

指定目标全局群集或区域群集中的一个节点或一系列节点。可以使用节点名称或节点 ID 指定每个节点。如果指定 `-z` 选项，则仅可以使用 `-n` 选项指定区域群集主机名，而不能指定全局群集主机名。如果未指定 `-z` 选项，则仅可以使用 `-n` 选项指定全局群集主机名。

可以与此选项一起指定的子命令如下所述：

`disable` 只禁用操作数列表中承载在指定节点上的资源。

`enable` 只启用操作数列表中承载在指定节点上的资源。

`status` 仅报告操作数列表中承载在指定节点上的资源的状态。

`-N netif@node[,...]`

`--netiflist netif@node[,...]`

指定一个资源属性。通过 `-N` 选项，可以设置 `NetIfList` 属性，而不必使用 `-p` 选项来指定该属性。如果未指定 `-N`，`clreslogicalhostname` 命令将尝试根据可用 IPMP 组或公共适配器以及与 `HostnameList` 属性关联的子网来设置 `NetIfList` 属性。

可以采用 `ipmpgroup@node[,...]` 或 `publicNIC@node[,...]` 格式指定 `NetIfList` 属性。如果未使用 `-N`，或者将该选项与 `publicNIC@node` 一起使用，则 `clreslogicalhostname` 命令将尝试创建必要的 IPMP 组。系统将会创建包含基本默认值的单适配器 IPMP 组，以后，用户可以使用标准的 Solaris IPMP 接口修改这些组。只会在全局群集节点中自动创建 IPMP 组。

您可以使用 `-N`，而无需使用 `-p` 直接设置 `NetIfList` 属性。但是，不能在同一命令中使用 `-N` 并显式设置 `NetIfList`。

`-N` 只能与 `create` 子命令一起使用。

`-o {- | clconfiguration}`

`--output {- | clconfiguration}`

指定要将资源配置信息写入到的位置。此位置可以是文件，也可以是标准输出。要指定标准输出，请指定 `-` 而不是文件名。如果指定标准输出，将会抑制该命令的其他所有标准输出。此选项仅可以与 `export` 子命令一同指定。

只会写入作为命令操作数提供的资源的配置信息。信息是采用 [clconfiguration\(5CL\) \[1245\]](#) 手册页中定义的格式写入的。

```
-p name=value
-p name+=array-values
-p name-=array-values
--property name=value
--property name+=value-values
--property name-=value-values
```

设置资源的标准属性和扩展属性。此选项仅可以与 `create` 子命令和 `set` 子命令一同指定。

有关标准属性的描述，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

有关某个资源类型的扩展属性的描述，请参见该资源类型的文档。

与此选项一起使用的运算符如下所述：

=	将属性设置为指定值。 <code>create</code> 子命令和 <code>set</code> 子命令接受此运算符。
+=	将一个或多个值添加到某个字符串数组值。只有 <code>set</code> 子命令接受此运算符。只能为字符串数组值指定此运算符。
-=	从某个字符串数组值中删除一个或多个值。只有 <code>set</code> 子命令接受此运算符。只能为字符串数组值指定此运算符。

如果只需在群集节点的子集上设置每节点属性，请通过在属性名称后面附加节点列表（用大括号括住）来指定要设置该属性的节点，如下所示：

```
name{nodelist}
```

`nodelist` 是以逗号分隔的节点名称或节点 ID 的列表。有关每节点属性的更多信息，请参见 [rt_properties\(5\) \[1143\]](#) 手册页。

```
-p name[,...]
--property name[,...]
```

为 `list-props` 子命令和 `show` 子命令指定属性列表。

可将此选项用于资源的标准属性和扩展属性。

有关标准属性的描述，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

有关某个资源类型的扩展属性的描述，请参见该资源类型的文档。

如果未指定此选项，`list-props` 子命令和 `show` 子命令将会列出所有资源属性或大多数资源属性，具体取决于是否同时指定了 `-v` 选项。

```
-R
--recursive
```

以递归方式启用或禁用资源，以确保满足所有必需的依赖性。此选项仅可以与 `disable` 子命令和 `enable` 子命令一同指定。

此选项与这些子命令一起使用的效果如下所示：

disable	禁用依赖于指定为命令操作数的资源的任何资源，即使这些资源未指定为命令操作数。
enable	启用指定为命令操作数的资源所依赖的任何资源，即使这些资源未指定为命令操作数。

-s *state*[,...]

--state *state*[,...]

为 `list` 子命令和 `status` 子命令指定状况列表。

此选项将会限制输出，以便只包括在节点列表中的一个或多个节点上处于指定状况之一的资源。

可能的状况如下：

- degraded
- detached
- faulted
- monitor_failed
- not_online - 指定除 `online` 或 `online_not_monitored` 以外的任何状况
- offline
- online
- online_not_monitored
- start_failed
- stop_failed
- unknown
- unmonitored
- wait

-V

--version

显示该命令的版本。

指定此选项时，不要将其与子命令、操作数或其他选项结合使用。子命令、操作数或其他选项将被忽略。-V 选项只显示命令的版本。不会执行任何其他操作。

-v

--verbose

将详细消息显示到标准输出。

可以在任何格式的此命令中指定该选项。

请不要同时指定 -v 选项和 -o - 选项。否则 -v 选项将被忽略。-o - 选项会抑制其他所有标准输出。

-z {zoneclustername | global | all}
--zoneclustername={zoneclustername | global | all}
--zoneclustername {zoneclustername | global | all}

指定一个或多个存在资源且您要对其执行操作的群集。

除 export 子命令以外的所有子命令均支持此选项。

如果指定此选项，还必须指定以下列表中的参数之一：

zoneclustername	指定使用此选项的命令仅对名为 zoneclustername 的区域群集中的所有指定资源起作用。
global	指定使用此选项的命令仅对全局群集中的所有指定资源进行操作。
all	如果在全局群集中使用此参数，则会指定使用此参数的命令对所有群集（包括全局群集和所有区域群集）中的所有指定资源起作用。 如果在某区域群集中使用此参数，则会指定使用此参数的命令仅对该区域群集中的所有指定资源起作用。

支持以下操作数：

resource 指定应接受 Oracle Solaris Cluster 资源名称作为操作数。如果子命令接受多个资源，则可以使用加号 (+) 指定所有逻辑主机名资源。

如果命令对于所有指定的操作数均成功执行，则返回零 (CL_NOERR)。如果某个操作数出错，命令会继续处理操作数列表中的下一个操作数。返回的退出代码始终反映最先发生的错误。

此命令返回下列退出状态代码：

0 CL_NOERR

没有错误

您发出的命令已成功完成。

1 CL_ENOMEM

交换空间不足

某个群集节点耗尽了交换内存或其他操作系统资源。

3 CL_EINVAL

参数无效

您键入的命令不正确，或者使用 -i 选项提供的群集配置信息语法不正确。

6 CL_EACCESS

权限被拒绝

指定的对象不可访问。要发出命令，可能需要拥有超级用户或 RBAC 访问权限。有关更多信息，请参见 [Unresolved link to " su1M"](#) 和 [Unresolved link to " rbac5"](#) 手册页。

9 CL_ESTATE

对象处于错误状况

您尝试修改一个在特定时间无法修改，或任何时候均无法修改的属性、资源组或其他对象。

10 CL_EMETHOD

资源方法失败

资源方法失败。方法因以下某个原因而失败：

- 尝试创建资源或修改资源属性时，`validate` 方法失败。
- 尝试启用、禁用或删除资源时，`validate` 以外的其他方法失败。

15 CL_EPROP

属性无效

您使用 `-p`、`-y` 或 `-x` 选项指定的属性或值不存在或不被允许。

35 CL_EIO

I/O 错误

发生了物理输入/输出错误。

36 CL_ENOENT

没有这样的对象

由于以下某个原因，找不到您所指定的对象：

- 该对象不存在。
- 您尝试使用 `-o` 选项创建的配置文件路径中的某个目录不存在。
- 您尝试使用 `-i` 选项访问的配置文件包含错误。

39 CL_EEXIST

对象已存在

您所指定的设备、设备组、群集互连组件、节点、群集、资源、资源类型、资源组或专用字符串已存在。

这些退出值与 [scha_calls\(3HA\) \[897\]](#) 手册页中描述的返回码兼容。

例 126 创建逻辑主机名资源

此命令将在名为 `rg-failover` 的资源组内创建名为 `logicalhost1` 的资源。创建的资源处于启用状况，并启用了监视功能。

```
# clreslogicalhostname create -g rg-failover logicalhost1
```

以下两个命令中的任何一个都将在区域群集 ZC 中创建一个名为 `logicalhost1` 的资源。可以从全局群集节点或区域群集 ZC 执行这些命令。如果从区域群集执行命令，则使用区域群集名称显式定义资源范围的操作是可选的。

```
# clreslogicalhostname create -g rg-failover -Z ZC logicalhost1
```

```
# clreslogicalhostname create -g rg-failover ZC:logicalhost1
```

例 127 使用不同的逻辑主机名创建逻辑主机名资源

此命令将在名为 `rg-failover` 的资源组中创建一个名为 `rs-logicalhost1` 的资源。

逻辑主机名与资源名称不相同，但逻辑主机的名称和 IP 地址保持相同。

```
# clreslogicalhostname create -g rg-failover \  
-h logicalhost1 rs-logicalhost1
```

例 128 为逻辑主机名资源指定 IPMP 组

此命令将为 `logicalhost1` 资源设置 IPMP 组。

```
# clreslogicalhostname create -g rg-failover \  
-N ipmp0@black,ipmp0@white logicalhost1
```

例 129 删除逻辑主机名资源

此命令可删除名为 `logicalhost1` 的资源。

```
# clreslogicalhostname delete logicalhost1
```

例 130 列出逻辑主机名资源

此命令将列出所有逻辑主机名资源。

```
# clreslogicalhostname list  
logicalhost1  
logicalhost2
```

例 131 列出逻辑主机名资源及其资源组和资源类型

此命令将列出所有逻辑主机名资源及其资源组和资源类型。

```
# clreslogicalhostname list -v  
Resources    Resource Groups    Resource Types  
-----  
logicalhost1 rg-failover-1    SUNW.LogicalHostname  
logicalhost2 rg-failover-2    SUNW.LogicalHostname
```

例 132 列出逻辑主机名资源的扩展属性

此命令将列出所有逻辑主机名资源的扩展属性。

```
# clreslogicalhostname list-props -v
Properties      Descriptions
-----
NetIfList      List of IPMP groups on each node
HostnameList   List of hostnames this resource manages
CheckNameService Name service check flag
```

有关以下属性的描述，请参见 [Unresolved link to " attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

[Intro\(1CL\) \[17\]](#)、[cluster\(1CL\) \[461\]](#)、[clresource\(1CL\) \[225\]](#)、[clressharedaddress\(1CL\) \[289\]](#)、[clresource\(1CL\) \[225\]](#)、[link to " rbac5"](#)、[r_properties\(5\) \[1103\]](#)

超级用户可以运行此命令的所有格式。

任何用户都可以运行带有以下选项的此命令：

- -? 选项
- -V 选项

要运行带有子命令的此命令，超级用户以外的用户需要具有 RBAC 授权。请参见下表。

子命令	RBAC 授权
create	solaris.cluster.modify
delete	solaris.cluster.modify
disable	solaris.cluster.admin
enable	solaris.cluster.admin
export	solaris.cluster.read
list	solaris.cluster.read
list-props	solaris.cluster.read
monitor	solaris.cluster.admin
reset	solaris.cluster.admin
set	solaris.cluster.modify

子命令	RBAC 授权
show	solaris.cluster.read
status	solaris.cluster.read
unmonitor	solaris.cluster.admin

名称

clresource, clrs — 管理 Oracle Solaris Cluster 数据服务的资源

```
/usr/cluster/bin/clresource subcommand [-?]  
  
/usr/cluster/bin/clresource -V  
  
/usr/cluster/bin/clresource subcommand [options] -v [resource]...  
  
/usr/cluster/bin/clresource clear [-f errorflag] [-g  
  [resourcegroup,...] [-t [resourcetype,...] -n node  
  [,...]] [-Z {zoneclustername | global}]  
  {+ | resource...}  
  
/usr/cluster/bin/clresource create -g resourcegroup -t  
  resourcetype [-d] [-p "property-name{node-specifier,...}"=  
  value] [-x "extension-property{node-specifier,...}"=value] [-y  
  standard-property=value] [-Z {zoneclustername | global}]  
  resource  
  
/usr/cluster/bin/clresource create -i {- | clconfiguration} -t  
  resourcetype [-a] [-d] [-g [resourcegroup,...] [-p "  
  property-name{node-specifier,...}"=value] [-x "  
  extension-property{node-specifier,...}"=value] [-y  
  standard-property=value] {+ | resource...}  
  
/usr/cluster/bin/clresource delete [-F] [-g [resourcegroup,...]  
  [-t [resourcetype,...] [-Z {zoneclustername | global}]  
  {+ | resource...}  
  
/usr/cluster/bin/clresource disable [-r] [-g [resourcegroup,...]  
  [-t [resourcetype,...] [-n node[,...]]  
  [-Z {zoneclustername | global}] {+ | resource...}  
  
/usr/cluster/bin/clresource enable [-r] [-g [resourcegroup,...]  
  [-t [resourcetype,...] [-n node[,...]]  
  [-Z {zoneclustername | global}] {+ | resource...}  
  
/usr/cluster/bin/clresource export [-o {- | configfile}]  
  {+ | resource...}  
  
/usr/cluster/bin/clresource list [-g [resourcegroup,...] [-t  
  [resourcetype,...] [-n node[,...]] [-Z  
  {zoneclustername [,...] | global | all}] {+ | resource...}  
  
/usr/cluster/bin/clresource list-props [-l listtype] [-g  
  [resourcegroup,...] [-p "property-name{node-specifier,...}" ,...]  
  [-t [resourcetype,...] [-x "extension-property{node-specifier,...}" ,...]  
  [-y "standard-property{node-specifier,...}" ,...] [-Z  
  {zoneclustername [,...] | global | all}] {+ | resource...}  
  
/usr/cluster/bin/clresource monitor [-g [resourcegroup,...] [-t  
  [resourcetype,...] [-n node[,...]] [-Z  
  {zoneclustername | global}] {+ | resource...}
```

```

/usr/cluster/bin/clresource set [-g [resourcegroup,...] [-p "
    property-name[{node-specifier,...}]=value] [-t
    [resourcetype,...] [-x "extension-property[{node-specifier,...}]=
    value] [-y standard-property [+ = | - =]value] [-Z
    {zoneclustername | global}] {+ | resource...}

/usr/cluster/bin/clresource show [-g [resourcegroup,...] [-p
    property-name[{node-specifier,...}],...] [-t [resourcetype,...]
    [-x "extension-property[{node-specifier,...}],...] [-y "
    standard-property[{node-specifier,...}],...] [-Z
    {zoneclustername [,...] | global | all}] {+ | resource...}

/usr/cluster/bin/clresource status [-g [resourcegroup,...] [-s
    [state,...] [-t [resourcetype,...] [-n node[,...]]
    [-Z {zoneclustername [,...] | global | all}] {+ | resource...}

/usr/cluster/bin/clresource unmonitor [-g [resourcegroup,...]
    [-t [resourcetype,...] [-n node[,...]]
    [-Z {zoneclustername | global}] {+ | resource...}

```

clresource 命令可管理 Oracle Solaris Cluster 数据服务的资源。clrs 命令是 clresource 命令的简短格式。clresource 命令和 clrs 命令是相同的。您可以使用此命令的任一种格式。

此命令的一般格式如下所示：

```
clresource [subcommand] [options] [operands]
```

仅当 *options* 指定 -? 选项或 -v 选项时，才可以省略 *subcommand*。

此命令的每个选项都有长格式和简短格式两种格式。在本手册页的 *选项* 部分给出了每个选项的两种格式，以及对该选项的描述。

在区域群集中使用此命令

在区域群集中，可以将 clresource 命令与所有子命令（export 除外）一起使用。

也可以将 -z 选项与所有子命令（export 除外）一起使用来指定要将操作限制到的特定区域群集的名称。另外，还可以将区域群集名称附加到资源名称 (*zoneclustername* : *resource*)，以便将操作限制到特定的区域群集。

您可以从全局群集节点访问所有区域群集信息，但无法从特定区域群集访问其他区域群集信息。如果不将操作限制到特定区域群集，则所用的子命令仅会在当前群集中运行。

一个区域群集中的某个资源可能对另一个区域群集中的某个资源存在依赖性，或者对全局群集中的某个资源存在依赖性。另外，全局群集中的某个资源可能对该全局群集中的任一区域群集上的某个资源存在依赖性。只能从全局群集设置这种群集间依赖性。

您可以使用以下命令指定群集间依赖性：

```

# clresource set -p resource_dependencies=target-
zc
:target-rs source-zc:

```

source-rs

例如，如果需要指定从区域群集 ZC1 中的资源 R1 到区域群集 ZC2 中的资源 R2 的依赖性，请使用以下命令：

```
# clresource set -p resource_dependencies=ZC2:R2 ZC1:R1
```

如果需要指定区域群集 ZC1 资源 R1 对全局群集资源 R2 的依赖性，请使用以下命令：

```
# clresource set -p resource_dependencies=global:R2 ZC1:R1
```

支持现有资源依赖性 (Strong、Weak 、Restart 和 Offline-Restart) 。

资源状况和状态

资源状况和资源状态是基于每个节点进行维护的。一个给定的资源在每个群集节点上可以具有不同的状况和状态。

资源组管理器 (Resource Group Manager, RGM) 基于对资源调用的方法在每个节点上设置资源状况。例如，在给定节点上对某个资源成功运行 STOP 方法后，该资源在此节点上的状况将为 OFFLINE。如果 STOP 方法返回非零值或超时，则该资源的状况为 Stop_failed

可能的资源状况包括：

- Online
- Offline
- Start_failed
- Stop_failed
- Monitor_failed
- Online_not_monitored
- Starting
- Stopping
- Not_online

注 - 状况名称 (例如 Offline 和 Start_failed) 不区分大小写。指定状况名称时，可以使用大小写字母的任意组合。

除资源状况外，RGM 还可以维护资源状态，资源状态可由资源本身使用 API 设置。Status Message 字段实际上包括两个组成部分：状态关键字和状态消息。状态消息由资源有选择性地设置，它是在状态关键字后面输出的任意文本字符串。

资源状态可能值的描述如下：

DEGRADED	资源已联机，但其性能或可用性可能在某方面出现了下降。
FAULTED	资源遇到错误，无法正常运行。

OFFLINE	资源已脱机。
ONLINE	资源已联机，正在提供服务。
UNKNOWN	当前状态是未知或正在转换。

支持以下子命令：

clear

清除与指定为命令操作数的资源关联的错误标志。此子命令接受使用加号 (+) 作为操作数来指定清除所有资源的错误标志。

您可以在全局群集中或区域群集中使用该子命令。

以下选项可以过滤操作数列表，以限制要在其上清除错误标志的资源：

<code>-g resourcegroup</code>	只清除操作数列表内属于 <i>resourcegroup</i> 中资源组成员的资源。
<code>-n node</code>	清除指定的一个或多个节点上的资源。如果未提供 <code>-n</code> 选项，该命令将清除所有节点上的资源。
<code>-t resourcetype</code>	只清除操作数列表内属于 <i>resourcetype</i> 中资源类型的实例的资源。
<code>-z {zoneclustername global}</code>	只清除您指定的一个或多个特定群集中的资源。要从全局群集清除区域群集中的资源，请使用 <code>-z</code> 选项指定区域群集。

默认情况下，`clear` 子命令将会清除 `STOP_FAILED` 错误标志。要显式指定要清除的错误标志，请使用 `-f` 选项。`-f` 选项接受的唯一错误标志是 `STOP_FAILED` 错误标志。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。

create

创建作为命令操作数指定的资源。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集在特定区域群集中创建资源，可以使用 `-z` 选项指定区域群集的名称。

将 `create` 和 `-i` 选项一起使用以指定配置文件时，该子命令接受使用加号 (+) 作为操作数。使用 `+` 操作数时，将会创建配置文件中提供的尚不存在的所有资源。

默认情况下，创建的资源处于启用状态，且启用了监视功能。但是，仅当资源的资源组联机后，该资源才会联机并受到监视。要在禁用状态下创建资源，请指定 `-d` 选项。

创建资源时，可以使用以下选项设置属性值：

`-p property-name= value` 设置标准属性或扩展属性，前提是它们的名称是唯一的。

`-x extension-property= value` 设置扩展属性。

`-y standard-property= value` 设置标准属性。

`node-specifier` 是 `-p` 和 `-x` 选项的可选限定符。它指示创建资源时将只在指定节点上设置属性，而不在群集中其他节点上设置指定的属性。如果未使用 `node-specifier`，则会在群集中的所有节点上设置指定的属性。`node-specifier` 语法的示例包括：

`-x "myprop{phys-schost-1}"`

大括号 ({}) 表示您只想要在节点 `phys-schost-1` 上设置指定的属性。对于大多数 shell，必须将大括号包含在引号中。

可以使用以下语法在两个节点上设置属性：

`-x "myprop{phys-schost-1,phys-schost-2}"`

超级用户以外的用户需要具有 `solaris.cluster.modify` 基于角色的访问控制 (role-based access control, RBAC) 授权才能使用此子命令。

另请参见 `delete` 子命令的描述。

delete

删除作为命令操作数指定的资源。此子命令接受使用加号 (+) 作为操作数来指定删除所有资源。

您可以在全局群集中或区域群集中使用该子命令。

此子命令将按照为满足资源间依赖性而需要遵照的顺序删除多个资源，而不管在命令行上以哪种顺序指定了资源。

同时删除多个资源时，该命令将在多个步骤中执行。如果命令被中断（例如，某个节点发生故障），某些资源组的配置可能会处于无效状态。要更正此问题并完成资源的删除，请在某个正常运行的节点上再次发出同一命令。

以下选项可以过滤操作数列表，以限制删除的资源：

`-g resourcegroup` 只删除操作数列表内属于 `resourcegroup` 中资源组的成员的资源。

`-t resourcetype` 只删除操作数列表内属于 `resourcetype` 中资源类型的实例的资源。

`-z {zoneclustername | global}` 只删除您指定的一个或多个特定群集中的资源。要从全局群集中删除区域群集中的资源，请使用 `-z` 选项指定区域群集。

默认情况下，仅当符合以下条件时，才会删除某个资源：

- 必须已禁用该资源。

- 必须已消除该资源的所有依赖性。

要强制删除指定的资源，请指定 -F 选项。请慎用此选项，因为它会有以下效果：

- 删除所有指定的资源，即使是未禁用的资源。
- 从其他资源的资源依赖性设置中删除所有指定的资源。

这些效果可能会导致群集中的服务丧失。未删除的相关资源也可能被置于无效状况或错误状况。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

另请参见 `create` 子命令的描述。

disable

禁用指定为命令操作数的资源。此子命令接受使用加号 (+) 作为操作数来指定禁用所有资源。

您可以在全局群集中或区域群集中使用该子命令。

以下选项可以过滤操作数列表，以限制禁用的资源：

- | | |
|--|---|
| <code>-g resourcegroup</code> | 只禁用操作数列表内属于 <code>resourcegroup</code> 中资源组的成员的资源。 |
| <code>-n node</code> | 可以使用 <code>-n node</code> 来禁用一个或多个节点上的资源。 |
| <code>-t resourcetype</code> | 只禁用操作数列表内属于 <code>resourcetype</code> 中资源类型的实例的资源。 |
| <code>-z {zoneclustername global}</code> | 只禁用您指定的一个或多个特定群集中的资源。要从全局群集删除区域群集中的资源，请使用 <code>-z</code> 选项指定区域群集。 |

`-r` 选项将会禁用依赖于指定为命令操作数的资源的任何资源。即使这些资源未指定为命令操作数，也会将其禁用。`-g` 选项和 `-t` 选项不适用于仅为满足资源依赖性而禁用的资源。

此子命令不会影响资源的监视状态。如果该资源在启用的情况下是受监视的，则禁用后仍受监视。如果后来重新启用该资源，则还会继续监视该资源。

该子命令将按照为满足资源间依赖性而需要遵照的顺序禁用资源，而不管在命令行上以哪种顺序指定了资源。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。

另请参见 `enable` 子命令的描述。

enable

启用指定为命令操作数的资源。此子命令接受使用加号 (+) 作为操作数来指定启用所有资源。

您可以在全局群集中或区域群集中使用该子命令。

以下选项可以过滤操作数列表，以限制启用的资源：

<code>-g resourcegroup</code>	仅启用操作数列表内属于 <i>resourcegroup</i> 中资源组的成员的资源。
<code>-n node</code>	可以使用 <code>-n node</code> 来启用一个或多个节点上的资源。
<code>-t resourcetype</code>	仅启用操作数列表内属于 <i>resourcetype</i> 中资源类型的实例的资源。
<code>-z {zoneclustername global}</code>	仅启用您指定的一个或多个特定群集中的资源。要从全局群集启用区域群集中的资源，请使用 <code>-z</code> 选项指定区域群集。

要确保满足所有必需的资源依赖性，请指定 `-r` 选项。`-r` 选项将会启用指定为命令操作数的资源所依赖的任何资源。即使这些资源未指定为命令操作数，也会将其启用。`-g` 选项和 `-t` 选项不适用于仅为满足资源依赖性而禁用的资源。

该子命令将按照为满足资源间依赖性而需要遵照的顺序启用资源，而不管在命令行上以哪种顺序指定了资源。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。

另请参见 `disable` 子命令的描述。

export

以 [clconfiguration\(5CL\) \[1245\]](#) 手册页所述的格式导出群集资源配置。

只能在全局群集中使用该子命令。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

list

显示指定为命令操作数的资源的列表。默认情况下，会显示所有资源。

您可以在全局群集中或区域群集中使用该子命令。

以下选项可以过滤操作数列表，以限制显示的资源：

<code>-g resourcegroup</code>	只显示操作数列表内属于 <i>resourcegroup</i> 中资源组的成员的资源。
<code>-n node</code>	可以使用 <code>-n node</code> 来仅列出在一个或多个节点上处于联机状态的那些资源。
<code>-t resourcetype</code>	只显示属于 <i>resourcetype</i> 中资源类型的实例的资源。

`-z {zoneclustername | global | all}` 只显示您指定的一个或多个特定群集中的资源。要从全局群集显示区域群集中的资源，请使用 `-z` 选项指定区域群集。

此子命令接受使用加号 (+) 作为操作数来指定显示所有资源配置。可以通过指定 `-g` 选项或 `-t` 选项，将显示的信息限于特定资源组或资源类型。如果未提供操作数，则会显示指定资源组中的所有资源，或显示属于指定资源类型的实例的所有资源。

如果指定 `-v` 选项，还将显示列表中每个资源的资源组和资源类型。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

list-props

显示指定为命令操作数的资源的属性列表。

您可以在全局群集中或区域群集中使用该子命令。

以下选项可以过滤操作数列表，以限制要显示其属性的资源：

`-g resourcegroup` 仅显示操作数列表内属于 `resourcegroup` 中资源组成员的资源的属性。

`-t resourcetype` 仅显示操作数列表内属于 `resourcetype` 中资源类型实例的资源的属性。

`-l` 选项指定要显示的资源属性的类型：

`-l all` 指定显示标准属性和扩展属性。

`-l extension` 指定只显示扩展属性。默认情况下，只显示扩展属性。

`-l standard` 指定只显示标准属性。

如果未指定 `-l` 选项，则只显示扩展属性。要显示标准属性，请使用 `-p` 选项或 `-y` 选项显式指定这些属性。

以下选项用于限制要显示的资源属性集：

`-p property-name` 只显示 `property-name` 中指定的属性。您可以在 `property-name` 中指定标准属性和扩展属性。

`-x extension-property` 只显示一个或多个节点上的、已在 `extension-property` 中指定的扩展属性。

`-y standard-property` 只显示 `standard-property` 中指定的标准属性。

`node-specifier` 是 `-p`、`-x` 和 `-y` 选项的可选限定符。它指示将只显示指定节点上的属性，而不显示群集中其他节点上的指定属性。如果未使用 `node-specifier`，则会显示群集中所有节点上的指定属性。`node-specifier` 语法的示例包括：

`-x "myprop{phys-schost-1}"`

大括号 ({}) 表示您只想要显示节点 `phys-schost-1` 上的指定属性。对于大多数 shell，必须将大括号包含在引号中。

可以使用以下语法显示两个节点上的属性：

```
-x "myprop{phys-schost-1,phys-schost-2}"
```

如果指定 `-v` 选项，还将显示每个属性的描述。

此子命令接受使用加号 (+) 作为操作数来指定显示所有资源属性。如果未提供操作数，则会显示指定资源组中所有资源的属性，或显示属于指定资源类型的实例的所有资源的属性。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

```
-z {zoneclustername | global | all}
```

列出您指定的一个或多个特定群集中的资源的属性。要从全局群集列出区域群集中的资源，请使用 `-z` 选项指定区域群集。

`monitor`

开启对指定为命令操作数的资源的监视。此子命令接受使用加号 (+) 作为操作数来指定开启对所有资源的监视。

您可以在全局群集中或区域群集中使用该子命令。

以下选项可以过滤操作数列表，以限制要对其开启监视的资源：

<code>-g resourcegroup</code>	只对操作数列表内属于 <code>resourcegroup</code> 中资源组的成员的资源开启监视。
<code>-n node</code>	只对在一个或多个节点上处于联机状态的那些资源开启监视。
<code>-t resourcetype</code>	只对操作数列表内属于 <code>resourcetype</code> 中资源类型的实例的资源开启监视。
<code>-z {zoneclustername global}</code>	只对您指定的一个或多个特定群集中的资源开启监视。要从全局群集开启对区域群集中资源的监视，请使用 <code>-z</code> 选项指定区域群集。

开启对某个资源的监视后，仅当符合以下条件时，才会监视该资源：

- 已启用该资源。
- 包含该资源的资源组至少已在一个群集节点上联机。

注 - 开启对某个资源的监视不会启用该资源。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。

另请参见 `unmonitor` 子命令的描述。

set

设置指定为命令操作数的资源的特定属性。此子命令接受使用加号 (+) 作为操作数来指定修改所有资源的特定属性。

您可以在全局群集中或区域群集中使用该子命令。

以下选项可以过滤操作数列表，以限制要修改其属性的资源：

-g resourcegroup 只修改操作数列表内属于 *resourcegroup* 中资源组成员的资源的属性。

-t resourcetype 只修改操作数列表内属于 *resourcetype* 中资源类型实例的资源的属性。

使用以下选项设置属性值：

-p property-name= value 设置标准属性或扩展属性，前提是它们的名称是唯一的。

-x extension-property= value 设置扩展属性。

-y standard-property= value 设置标准属性。

node-specifier 是 **-p** 和 **-x** 选项的可选限定符，用于更新每节点扩展属性。它指示仅在指定节点上设置相应属性，而不在群集中其他节点上设置指定的属性。如果未使用 *node-specifier*，则会在群集中的所有节点上设置指定的属性。*node-specifier* 语法的示例包括：

-x "myprop{phys-schost-1}"

大括号 ({}) 表示您只想要在节点 *phys-schost-1* 上设置指定的属性。对于大多数 shell，必须将大括号包含在引号中。

可以使用以下语法在两个节点上设置属性：

-x "myprop{phys-schost-1,phys-schost-2}"

-z {zoneclustername | global}

只设置您指定的一个或多个特定群集中的资源的属性。要从全局群集设置区域群集中资源的属性，请使用 **-z** 选项指定区域群集。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

show

显示指定为命令操作数的资源的配置。默认情况下，会显示所有资源的配置。

您可以在全局群集中或区域群集中使用该子命令。

以下选项可以过滤操作数列表，以限制要显示其配置的资源列表：

-g resourcegroup 只显示操作数列表内属于 *resourcegroup* 中资源组成员的资源的配置。

`-n node` 可以使用 `-n node` 来仅显示在一个或多个节点上处于联机状态的那些资源的配置。

`-t resourcetype` 只显示操作数列表内属于 `resourcetype` 中资源类型实例的资源配置。

`-z {zoneclustername | global | all}`

只显示您指定的一个或多个特定群集中的资源。要从全局群集显示区域群集中的资源，请使用 `-z` 选项指定区域群集。

以下选项用于限制显示的资源属性集：

`-p property-name` 只显示 `property-name` 中指定的属性。您可以在 `property-name` 中指定标准属性和扩展属性。

`-x extension-property` 只显示一个或多个节点上的、已在 `extension-property` 中指定的扩展属性。

`-y standard-property` 只显示 `standard-property` 中指定的标准属性。

`node-specifier` 是 `-p`、`-x` 和 `-y` 选项的可选限定符。它指示将只显示指定节点上的属性，而不显示群集中其他节点上的指定属性。如果未使用 `node-specifier`，则会显示群集中所有节点上的指定属性。`node-specifier` 语法的示例包括：

`-x "myprop{phys-schost-1}"`

大括号 ({}) 表示您只想要显示节点 `phys-schost-1` 上的指定属性。对于大多数 shell，必须将大括号包含在引号中。

可以使用以下语法显示两个节点上的属性：

`-x "myprop{phys-schost-1,phys-schost-2}"`

此子命令接受使用加号 (+) 作为操作数来指定显示所有资源配置。可以通过指定 `-g` 选项或 `-t` 选项，将显示的信息限于特定资源组或资源类型。如果未提供操作数，该子命令将显示所有指定资源的配置。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

status

显示指定为命令操作数的资源的状态。默认情况下，会显示所有资源的状态。

以下选项可以过滤操作数列表，以限制要显示其状态的资源的列表：

`-g resourcegroup` 仅显示操作数列表内属于 `resourcegroup` 中资源组成员的资源的状态。

`-n node` 可以使用 `-n node` 来仅显示在一个或多个节点上处于联机状态的那些资源的状态。不能同时指定 `-n` 和 `-s` 选项。

-s <i>state</i>	仅显示操作数列表内处于 <i>state</i> 中的状况的资源的状况。不能同时指定 -n 和 -s 选项。
-t <i>resourcetype</i>	仅显示操作数列表内属于 <i>resourcetype</i> 中资源类型实例的资源的状况。
-z { <i>zoneclustername</i> global all}	显示您指定的一个或多个特定群集中的资源的状况。要从全局群集显示区域群集中资源的状况，请使用 -z 选项指定区域群集。

此子命令接受使用加号 (+) 作为操作数来指定显示所有资源的状况。可以通过指定 -g 选项或 -t 选项，将显示的信息限于特定资源组或资源类型。如果未提供操作数，将会显示所有指定资源的状况。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

unmonitor

关闭对指定为命令操作数的资源的监视。此子命令接受使用加号 (+) 作为操作数来指定关闭对所有资源的监视。

您可以在全局群集中或区域群集中使用该子命令。

如果关闭对某个被禁用资源的监视，该资源不会受到影响。该资源及其监视器已脱机。

注 - 关闭对某个资源的监视不会禁用该资源。但是，当您禁用某个资源时，不需要关闭对该资源的监视。禁用的资源及其监视器将保持脱机状态。

以下选项可以过滤操作数列表，以限制要对其关闭监视的资源：

-g <i>resourcegroup</i>	只对操作数列表内属于 <i>resourcegroup</i> 中资源组的成员的资源关闭监视。
-n <i>node</i>	只对在一个或多个节点上处于联机状态的那些资源关闭监视。
-t <i>resourcetype</i>	只对操作数列表内属于 <i>resourcetype</i> 中资源类型的实例的资源关闭监视。
-z { <i>zoneclustername</i> global}	只对您指定的一个或多个特定群集中的资源关闭监视。要从全局群集关闭对区域群集中资源的监视，请使用 -z 选项指定区域群集。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。

另请参见 `disable` 子命令和 `monitor` 子命令的描述。

支持以下选项：

-?

--help

显示帮助信息。使用该选项时，不会执行任何其他处理。

指定此选项时，子命令可有可无。

如果仅指定此选项而不使用子命令，将会显示此命令的子命令列表。

如果使用子命令且指定此选项，将会显示该子命令的用法选项。

-a

--automatic

当基于群集配置文件 ([clconfiguration\(5CL\) \[1245\]](#)) 创建资源时，将自动执行以下附加操作：

- 注册资源类型
- 创建资源组
- 创建操作数列表中指定的资源所依赖的资源

群集配置信息包含的信息必须足以执行以下所有操作：

- 启用要注册的资源类型
- 启用要创建的资源组
- 启用要创建的资源

此选项仅可以与 `create` 子命令一同指定。如果指定此选项，还必须指定 `-i` 选项并提供配置文件。

-d

--disable

创建资源后禁用该资源。此选项仅可以与 `create` 子命令一同指定。默认情况下，资源在启用状态下创建。

启用某个资源并不保证该资源联机。仅当某个资源的资源组已至少在一个节点上联机后，该资源才会联机。

-f *errorflag*

--flag=*errorflag*

--flag *errorflag*

显式指定要通过 `clear` 子命令清除的错误标志。此选项仅可以与 `clear` 子命令一同指定。默认情况下，`clear` 子命令将会清除 `STOP_FAILED` 错误标志。

- f 选项接受的唯一错误标志是 `STOP_FAILED` 错误标志。

-F

--force

强制删除未禁用的资源。此选项仅可以与 `delete` 子命令一同指定。

请慎用此选项，因为它会有以下效果：

- 删除所有指定的资源，即使是未禁用的资源。
- 从其他资源的资源依赖性设置中删除所有指定的资源。

这些效果可能会导致群集中的服务丧失。未删除的相关资源也可能被置于无效状况或错误状况。

`-g resourcegroup[,...]`
`--resourcegroup=resourcegroup[,...]`
`--resourcegroup resourcegroup[,...]`

指定一个资源组或资源组列表。

对于除 `create` 以外的子命令，该命令仅对操作数列表内属于指定的资源组成员的资源起作用。使用 `-g` 选项指定资源组。

随 `create` 子命令指定了 `-g` 选项时，`clresource` 将在指定的资源组中创建资源。使用此选项时，只能指定一个资源组。

`-i {- | clconfiguration}`
`--input={- | clconfiguration}`
`--input {- | clconfiguration}`

指定用于创建或修改资源的配置信息。此信息必须符合 [clconfiguration\(5CL\) \[1245\]](#) 手册页中定义的格式。此信息可包含在文件中或者通过标准输入提供。要指定标准输入，请指定 `-` 而不是文件名。

仅会创建或修改作为命令操作数提供的资源。在命令中指定的选项将会覆盖配置信息中设置的任何选项。如果配置信息中缺少配置参数，则必须在命令行中指定这些参数。

将 `-i` 选项与 `create` 子命令结合使用时，`clresource` 将会注册所有必需的资源类型，并创建所有必需的资源组。您必须提供注册和配置所需的全部信息。所有其他配置数据将被忽略。

`-l listtype`
`--listtype=listtype`
`--listtype listtype`

指定要通过 `list-props` 子命令显示的资源属性的类型。此选项仅可以与 `list-props` 子命令一同指定。

必须根据以下列表为 `listtype` 指定一个值：

<code>all</code>	指定显示标准属性和扩展属性。
<code>extension</code>	指定只显示扩展属性。默认情况下，只显示扩展属性。
<code>standard</code>	指定只显示标准属性。

如果未指定 `-l` 选项，则只显示扩展属性。要显示标准属性，请使用 `-p` 选项或 `-y` 选项显式指定这些属性。

```
-n node[,...]  
--node=node[,...]  
--node node[,...]
```

指定目标全局群集或区域群集中的一个节点或一系列节点。可以使用节点名称或节点 ID 指定每个节点。

如果指定 -z 选项，则仅可以使用 -n 选项指定区域群集主机名，而不能指定全局群集主机名。如果未指定 -z 选项，则仅可以使用 -n 选项指定全局群集主机名。

可以与此选项一起指定的子命令如下所述：

<code>disable</code>	只禁用操作数列表中承载在指定节点上的资源。
<code>enable</code>	只启用操作数列表中承载在指定节点上的资源。
<code>list</code>	只显示操作数列表中承载在指定节点上的那些资源的列表。
<code>monitor</code>	只监视操作数列表中承载在指定节点上的那些资源。
<code>show</code>	只显示操作数列表中承载在指定节点上的那些资源的配置信息。
<code>status</code>	仅报告操作数列表中承载在指定节点上的资源的状态。
<code>unmonitor</code>	只取消监视操作数列表中承载在指定节点上的那些资源。

```
-o {- | clconfiguration}  
--output={- | clconfiguration}  
--output {- | clconfiguration}
```

指定要将资源配置信息写入到的位置。此位置可以是文件，也可以是标准输出。要指定标准输出，请指定短划线 (-) 而不是文件名。如果指定标准输出，将会抑制该命令的其他所有标准输出。此选项仅可以与 `export` 子命令一同指定。

只会写入作为命令操作数提供的资源的配置信息。信息是采用 [clconfiguration\(5CL\) \[1245\]](#) 手册页中定义的格式写入的。

```
-p property-name=value  
-p property-name+=array-values  
-p property-name-=array-values  
--property=property-name=value  
--property=property-name+=array-values  
--property=property-name-=array-values  
--property property-name=value  
--property property-name+=array-values  
--property property-name-=array-values
```

设置作为命令操作数提供的资源的属性值。只能随 `create` 子命令和 `set` 子命令指定此选项的赋值格式。

使用 -p 选项可以指定任何标准属性或扩展属性。如果某个资源类型的某个扩展属性与该资源类型的某个标准属性同名，则使用 -p 选项会返回错误。在这种情况下，请使用 -x 选项指定扩展属性，使用 -y 选项指定标准属性。

有关标准属性的描述，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

有关某个资源类型的扩展属性的描述，请参见该资源类型的文档。

与此选项一起使用的运算符如下所述：

= 将属性设置为指定值。create 子命令和 set 子命令接受此运算符。

+= 将一个或多个值添加到某个字符串数组值。只有 set 子命令接受此运算符。仅可以为接受字符串值列表的属性指定此运算符，例如 Resource_dependencies。

-= 从某个字符串数组值中删除一个或多个值。只有 set 子命令接受此运算符。仅可以为接受字符串值列表的属性指定此运算符，例如 Resource_dependencies。

要在群集节点的子集上设置每节点扩展属性，请在设置属性时指定节点。将节点列表括在大括号中，并将其附加在属性名称的后面，如下所示：

```
name{node}
```

node 是以逗号分隔的节点名称或节点 ID 的列表。有关每节点扩展属性的更多信息，请参见 [rt_properties\(5\) \[1143\]](#) 手册页。

要设置每节点资源对群集节点子集的依赖性，请采用以下格式指定每个每节点依赖性：

```
myres1@node1,myres2@node2,myres3@node3
```

对于 gds-rs 资源，以下命令设置资源 trancos-3-rs 对节点 ptrancos1 的依赖性和资源 trancos-4-rs 对节点 ptrancos2 的依赖性：

```
# clresource set -p \  
resource_dependencies=trancos-3-rs@ptrancos1,trancos-4-rs@ptrancos2 gds-rs
```

```
phys-schost-1# clresource show -p resource_dependencies gds-rs  
=== Resources ===  
Resource: gds-rs  
Resource_dependencies: trancos-3-rs@ptrancos1 trancos-4-rs@ptrancos2
```

要设置本地节点范围的资源依赖性，请采用以下格式指定 LOCAL_NODE 限定符：

```
myres1{LOCAL_NODE},myres2{LOCAL_NODE}
```

对于 gds-rs 资源，以下命令设置本地节点对资源 trancos-3-rs 的依赖性：

```
# clresource set -p resource_dependencies=trancos-3-rs{LOCAL_NODE} gds-rs
```

```
phys-schost-1# clresource show -p resource_dependencies gds-rs
=== Resources ===
Resource: gds-rs
Resource_dependencies: trancos-3-rs{LOCAL_NODE}
```

有关每节点资源依赖性和依赖性范围限定符的更多信息，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

```
-p property-name[...]
--property=property-name[...]
--property property-name[...]
```

为 `list-props` 子命令和 `show` 子命令指定属性列表。

使用 `-p` 选项可以指定任何标准属性或扩展属性。如果某个资源类型的某个扩展属性与该资源类型的某个标准属性同名，则使用 `-p` 选项会返回错误。在这种情况下，请使用 `-x` 选项指定扩展属性，使用 `-y` 选项指定标准属性。

有关标准属性的描述，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

有关某个资源类型的扩展属性的描述，请参见该资源类型的文档。

如果未指定此选项，`list-props` 子命令和 `show` 子命令将会列出所有资源属性或大多数资源属性，具体取决于是否同时指定了 `-v` 选项。

```
-r
--recursive
```

以递归方式启用或禁用资源，以确保满足所有必需的依赖性。此选项仅可以与 `disable` 子命令和 `enable` 子命令一同指定。

此选项与这些子命令一起使用的效果如下所示：

<code>disable</code>	禁用依赖于指定为命令操作数的资源的任何资源。即使这些资源未指定为命令操作数，也会将其禁用。
<code>enable</code>	启用指定为命令操作数的资源所依赖的任何资源。即使这些资源未指定为命令操作数，也会将其启用。

```
-s state[...]
--state=state[...]
--state state[...]
```

为 `list` 子命令和 `status` 子命令指定状况列表。

此选项将会限制输出，以便只包括在节点列表中的一个或多个节点上处于指定状况之一的资源。

可能的状况如下：

- `Online`
- `Offline`
- `Start_failed`
- `Stop_failed`

-
- Monitor_failed
 - Online_not_monitored
 - Starting
 - Stopping
 - Not_online

注 - 状况名称 (例如 Offline 和 Start_failed) 不区分大小写。指定状况名称时, 可以使用大小写字母的任意组合。

-t resourcetype[,...]
--type=resourcetype[,...]
--type resourcetype[,...]

指定一个资源类型或资源类型列表。

对于接受此选项的所有子命令 (create 除外), 该命令只对同时满足以下两个条件的资源起作用:

- 资源在操作数列表中。
- 资源是 - t 选项指定的资源类型的实例。

将 -t 选项与 clresource create 一起指定时, 将会创建指定类型的资源。只能指定一个资源类型。

有关资源类型名称格式的描述, 请参见[Unresolved link to " Oracle Solaris Cluster 数据服务规划和管理指南 中的合法的 RGM 名称"](#)。

-u

如果使用 + 操作数, 此选项会指定命令要对其资源组已暂停的资源起作用。

如果指定 + 操作数时未指定 -u 选项, 命令将会忽略其资源组已暂停的所有资源。当随 clear、disable、enable、monitor、set 或 unmonitor 命令指定了 + 操作数时, -u 选项有效。

将 + 操作数与 clear、disable、enable、monitor、set 或 unmonitor 子命令一起使用时, 除非同时指定了 - u 选项, 否则命令将会忽略其资源组已暂停的所有资源。

-V

--version

显示该命令的版本。

指定此选项时, 不要将其与子命令、操作数或其他选项结合使用。子命令、操作数或其他选项将被忽略。-V 选项只显示命令的版本。不会执行任何其他操作。

-v

--verbose

将详细消息显示到标准输出。

可以在任何格式的此命令中指定该选项。

请不要同时指定 `-v` 选项和 `-o` 选项。否则 `-v` 选项将被忽略。`-o` 选项会抑制其他所有标准输出。

```
-x extension-property=value
-x extension-property+=array-value
-x extension-property-=array-value
--extension-property=extension-property=value
--extension-property=extension-property+=array-value
--extension-property=extension-property-=array-value
--extension-property extension-property=value
--extension-property extension-property+=array-value
--extension-property extension-property-=array-value
```

设置或修改作为命令操作数提供的资源的扩展属性值。

通常，使用 `-p` 选项可以指定任何标准属性或扩展属性。如果某个资源类型的某个扩展属性与该资源类型的某个标准属性同名，则使用 `-p` 选项会返回错误。在这种情况下，请使用 `-x` 选项指定扩展属性，使用 `-y` 选项指定标准属性。

只能随 `create` 子命令和 `set` 子命令指定此选项的赋值格式。

有关某个资源类型的扩展属性的描述，请参见该资源类型的文档。

与此选项一起使用的运算符如下所述：

- `=` 将属性设置为指定值。`create` 子命令和 `set` 子命令接受此运算符。
- `+=` 将一个或多个值添加到某个字符串数组值。只有 `set` 子命令接受此运算符。仅可以为接受字符串值列表的属性指定此运算符，例如 `Resource_dependencies`。
- `-=` 从某个字符串数组值中删除一个或多个值。只有 `set` 子命令接受此运算符。仅可以为接受字符串值列表的属性指定此运算符，例如 `Resource_dependencies`。

要在群集节点的子集上设置每节点扩展属性，请在设置属性时指定节点。将节点列表括在大括号中，并将其附加在属性名称的后面，如下所示：

```
name{node}
```

`node` 是以逗号分隔的节点名称或节点 ID 的列表。有关每节点属性的更多信息，请参见 [rt_properties\(5\) \[1143\]](#) 手册页。

```
-x extension-property[,...]
--extension-property=extension-property[,...]
--extension-property extension-property[,...]
```

为 `list-props` 子命令和 `show` 子命令指定扩展属性列表。

有关某个资源类型的扩展属性的描述，请参见该资源类型的文档。

使用 `-p` 选项可以指定任何标准属性或扩展属性。如果某个资源类型的某个扩展属性与该资源类型的某个标准属性同名，则使用 `-p` 选项会返回错误。在这种情况下，请使用 `-x` 选项指定扩展属性，使用 `-y` 选项指定标准属性。

如果未指定此选项，`list-props` 子命令和 `show` 子命令将会列出所有资源属性或大多数资源属性，具体取决于是否同时指定了 `-v` 选项。

```
-y standard-property=value
-y standard-property+=array-value
-y standard-property-=array-value
--standard-property=standard-property=value
--standard-property=standard-property+=array-value
--standard-property=standard-property-=array-value
--standard-property standard-property=value
--standard-property standard-property+=array-value
--standard-property standard-property-=array-value
```

设置或修改作为命令操作数提供的资源的标准属性值。

使用 `-p` 选项可以指定任何标准属性或扩展属性。如果某个资源类型的某个扩展属性与该资源类型的某个标准属性同名，则使用 `-p` 选项会返回错误。在这种情况下，请使用 `-x` 选项指定扩展属性，使用 `-y` 选项指定标准属性。

只能随 `create` 子命令和 `set` 子命令指定此选项的赋值格式。

有关标准属性的描述，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

与此选项一起使用的运算符如下所述：

<code>=</code>	将属性设置为指定值。 <code>create</code> 子命令和 <code>set</code> 子命令接受此运算符。
<code>+=</code>	将一个或多个值添加到某个字符串数组值。只有 <code>set</code> 子命令接受此运算符。仅可以为接受字符串值列表的属性指定此运算符，例如 <code>Resource_dependencies</code> 。
<code>-=</code>	从某个字符串数组值中删除一个或多个值。只有 <code>set</code> 子命令接受此运算符。仅可以为接受字符串值列表的属性指定此运算符，例如 <code>Resource_dependencies</code> 。

```
-y standard-property[...]
--standard-property=standard-property[...]
--standard-property standard-property[...]
```

为 `list-props` 子命令和 `show` 子命令指定标准属性列表。

有关标准属性的描述，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

使用 `-p` 选项可以指定任何标准属性或扩展属性。如果某个资源类型的某个扩展属性与该资源类型的某个标准属性同名，则使用 `-p` 选项会返回错误。在这种情况下，请使用 `-x` 选项指定扩展属性，使用 `-y` 选项指定标准属性。

如果未指定此选项，`list-props` 子命令和 `show` 子命令将会列出所有资源属性或大多数资源属性，具体取决于是否同时指定了 `-v` 选项。

`-z {zoneclustername | global | all}`
`--zoneclustername={zoneclustername | global | all}`
`--zoneclustername {zoneclustername | global | all}`

指定一个或多个存在资源且您要对其执行操作的群集。

除 `export` 子命令以外的所有子命令均支持此选项。

如果指定此选项，还必须指定以下列表中的参数之一：

<code>zoneclustername</code>	指定使用此选项的命令仅对名为 <code>zoneclustername</code> 的区域群集中的所有指定资源起作用。
<code>global</code>	指定使用此选项的命令仅对全局群集中的所有指定资源进行操作。
<code>all</code>	如果在全局群集中使用此参数，则会指定使用此参数的命令对所有群集（包括全局群集和所有区域群集）中的所有指定资源起作用。 如果在某区域群集中使用此参数，则会指定使用此参数的命令仅对该区域群集中的所有指定资源起作用。

仅支持以下操作数：

`resource` 指定要管理的一个或多个资源。如果子命令接受多个资源，则可以使用加号 (+) 指定所有资源。

如果命令对于所有指定的操作数均成功执行，则返回零 (`CL_NOERR`)。如果某个操作数出错，命令会继续处理操作数列表中的下一个操作数。返回的退出代码始终反映最先发生的错误。

可返回下列退出代码：

0 `CL_NOERR`

没有错误

您发出的命令已成功完成。

1 `CL_ENOMEM`

交换空间不足

某个群集节点耗尽了交换内存或其他操作系统资源。

3 `CL_EINVAL`

参数无效

您键入的命令不正确，或者使用 -i 选项提供的群集配置信息语法不正确。

6 CL_EACCESS

权限被拒绝

指定的对象不可访问。要发出命令，可能需要拥有超级用户或 RBAC 访问权限。有关更多信息，请参见 [Unresolved link to " su1M"](#) 和 [Unresolved link to " rbac5"](#) 手册页。

9 CL_ESTATE

对象处于错误状况

您尝试修改一个在特定时间无法修改，或任何时候均无法修改的属性、资源组或其他对象。

10 CL_EMETHOD

资源方法失败

资源方法失败。方法因以下某个原因而失败：

- 尝试创建资源或修改资源属性时，validate 方法失败。
- 尝试启用、禁用或删除资源时，validate 以外的其他方法失败。

15 CL_EPROP

属性无效

您使用 -p、-y 或 -x 选项指定的属性或值不存在或不被允许。

36 CL_ENOENT

没有这样的对象

由于以下某个原因，找不到您所指定的对象：

- 该对象不存在。
- 您尝试使用 -o 选项创建的配置文件路径中的某个目录不存在。
- 您尝试使用 -i 选项访问的配置文件包含错误。

39 CL_EEXIST

对象已存在

您所指定的设备、设备组、群集互连组件、节点、群集、资源、资源类型、资源组或专用字符串已存在。

41 CL_ETYPE

类型无效

您使用 -t 或 -p 选项指定的类型不存在。

这些退出值与 [scha_calls\(3HA\) \[897\]](#) 手册页中描述的返回码兼容。

例 133 创建资源

此示例将在名为 `rg-failover` 的资源组中创建一个名为 `rs-nfs` 的资源。该资源是 `SUNW.nfs` 资源类型的实例。创建的资源处于启用状况，并开启了资源监视功能。

```
# clresource create -g rg-failover -t SUNW.nfs rs-nfs
```

以下两个命令中的任何一个都在区域群集 `ZC` 内的资源组 `rg-failover` 中创建一个名为 `rs-nfs` 的资源。可以从全局群集节点执行这些命令，也可以在区域群集 `ZC` 内执行这些命令。如果从区域群集执行命令，则使用区域群集名称显式定义资源范围的操作是可选的。

```
# clresource create -g rg-failover -t SUNW.nfs -Z ZC rs-nfs
```

```
# clresource create -g rg-failover -t SUNW.nfs ZC:rs-nfs
```

例 134 开启对资源的监视

此示例将开启对名为 `rs-nfs` 的资源的监视。

```
# clresource monitor rs-nfs
```

开启对某个资源的监视后，在使用 `clresource unmonitor` 命令显式关闭监视之前，监视将一直保持开启状态。禁用和启用某个资源不影响其监视状态。

例 135 启用资源

此示例将启用资源组 `rg-failover` 和 `rg-failover2` 中的所有资源。

```
# clresource enable -g rg-failover,rg-failover2 +
```

此命令不会影响资源的监视状态。

例 136 设置资源属性

此示例会将 `SUNW.nfs` 资源类型的所有实例的 `r_description` 属性设置为 `HA-NFS res`。

```
# clresource set -t SUNW.nfs -p r_description="HA-NFS res" +
```

例 137 设置每节点资源属性

此示例将资源 `rs-oracle` 的每节点属性 `oracle_sid` 在不同节点上设置为不同的值，如下所述：

- 在节点 `phys-schost-1` 和节点 `phys-schost-2` 上，此属性设置为 `myora1`。
- 在节点 `phys-schost-3` 上，此属性设置为 `myora2`。

此示例假定大括号字符对于所用的 shell 具有特殊含义。因此，节点列表附加到的每个属性名称都包含在双引号中。

```
# clresource set -p "oracle_sid{phys-schost-1,phys-schost-2}"=myora1 \  
-p "oracle_sid{phys-schost-3}"=myora2 rs-oracle
```

例 138 设置每节点资源依赖性

此示例设置 gds-rs 每个节点的资源依赖性，以便它依赖于两个不同的逻辑主机资源。

```
# clresource set -p resource_dependencies=node-3-rs@pnode1,node-4-rs@pnode2 gds-rs  
  
# clresource show -p resource_dependencies gds-rs  
Resource: gds-rs  
Standard Properties:  
Resource_dependencies: node-3-rs@pnode1,node-4-rs@pnode2
```

例 139 向字符串数组属性添加值

此示例将值 rs-oracle 添加到资源 rs-myapp 的字符串数组属性 resource_dependencies 中。此字符串数组属性中的现有值不变。

```
# clresource set -p resource_dependencies+=rs-oracle rs-myapp  
  
# clresource show -p resource_dependencies rs-myapp  
Resource: rs-myapp  
Standard Properties:  
Resource_dependencies: rs-nfs rs-oracle
```

例 140 删除资源

此示例将删除名为 rs-nfs 的资源。

```
# clresource delete rs-nfs
```

例 141 更新整个群集配置

此示例将通过执行以下操作序列来更新整个群集配置：

1. 使群集中的所有资源组脱机，删除所有资源，然后删除所有资源组
2. 取消注册所有资源类型
3. 创建配置文件 /net/server/export/mycluster.xml 中指定的所有资源，注册这些资源的资源类型，然后创建所有必需的资源组

```
# clresourcegroup delete --force +  
# clresourcetype unregister +  
# clresource -i /net/server/export/mycluster.xml -a +
```

例 142 列出资源

此示例将列出所有资源。

```
# clresource list
logicalhost1
rs-nfs-1
rs-nfs-2
logicalhost2
rs-apache-1
```

例 143 列出资源及其组和类型

此示例将列出所有资源及其资源组和资源类型。

```
# clresource list -v

Resource Name      Resource Group      Resource Type
-----
logicalhost1      rg-failover-1      SUNW.LogicalHostname
rs-nfs-1          rg-failover-1      SUNW.nfs
logicalhost2      rg-failover-2      SUNW.LogicalHostname
rs-nfs-2          rg-failover-2      SUNW.nfs
rs-apache-1       rg-failover-1      SUNW.apache
```

例 144 列出特定类型的资源

此示例将列出 nfs 资源类型的所有实例。

```
# clresource list -t nfs
rs-nfs-1
rs-nfs-2
```

例 145 列出资源类型的扩展属性和描述

此示例将列出 nfs 资源类型的扩展属性以及每个扩展属性的描述。

```
# clresource list-props -t nfs -v
Properties          Descriptions
-----
Monitor_retry_count  Number of PMF restarts allowed for the fault monitor
Monitor_retry_interval  Time window (minutes) for fault monitor restarts
Rpcbind_nullrpc_timeout  Timeout(seconds) to use when probing rpcbind
Nfsd_nullrpc_timeout    Timeout(seconds) to use when probing nfsd
Mountd_nullrpc_timeout  Timeout(seconds) to use when probing mountd
Statd_nullrpc_timeout   Timeout(seconds) to use when probing statd
Lockd_nullrpc_timeout   Timeout(seconds) to use when probing lockd
Rpcbind_nullrpc_reboot  Boolean to indicate if we should reboot system when
                        null rpc call on rpcbind fails
Nfsd_nullrpc_restart    Boolean to indicate if we should restart nfsd when
                        null rpc call fails
```

Mountd_nullrpc_restart Boolean to indicate if we should restart mountd when
null rpc call fails

*Line breaks in the Descriptions column are added to enhance
the readability of this example. Actual output from the command does not
contain these line breaks.*

例 146 通过禁用然后启用资源清除 Start_failed 资源状态

Start_failed 资源状况指示 Start 或 Preinet_start 方法对某个资源失败或超时，但该资源的资源组仍已联机。即使该资源处于故障状况，可能不提供服务，其资源组仍然会联机。如果资源的 Failover_mode 属性设置为 None 或者设置为阻止资源组故障转移的另一个值，则可能会出现此状况。

与 Stop_failed 资源状况不同，Start_failed 资源状况不会阻止您或 Oracle Solaris Cluster 软件对资源组执行操作。无需发出 *command clear* 命令来清除 Start_failed 资源状况。只需执行一个可重新启动资源的命令。

以下命令通过禁用然后重新启用 resource-1 资源来清除该资源发生的 Start_failed 资源状态。

```
# clresource disable resource-1  
# clresource enable resource-1
```

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

[Intro\(1CL\) \[17\]](#), [clreslogicalhostname\(1CL\) \[207\]](#), [clresourcegroup\(1CL\) \[253\]](#),
[clresourcetype\(1CL\) \[277\]](#), [clressharedaddress\(1CL\) \[289\]](#), [cluster\(1CL\) \[461\]](#),
[scha_calls\(3HA\) \[897\]](#), [clconfiguration\(5CL\) \[1245\]](#), [Unresolved link to "attributes5"](#),
[r_properties\(5\) \[1103\]](#), [Unresolved link to "rbac5"](#)

[Unresolved link to "Oracle Solaris Cluster 数据服务规划和管理指南"](#)

超级用户可以运行此命令的所有格式。

任何用户都可以运行带有以下选项的此命令：

- -? 选项
- -V 选项

要运行带有子命令的此命令，超级用户以外的用户需要具有 RBAC 授权。请参见下表。

子命令	RBAC 授权
create	solaris.cluster.modify
delete	solaris.cluster.modify
disable	solaris.cluster.admin
enable	solaris.cluster.admin
export	solaris.cluster.read
list	solaris.cluster.read
list-props	solaris.cluster.read
set	solaris.cluster.modify
monitor	solaris.cluster.admin
clear	solaris.cluster.admin
show	solaris.cluster.read
status	solaris.cluster.read
unmonitor	solaris.cluster.admin

名称

clresourcegroup, clrg — 管理 Oracle Solaris Cluster 数据服务的资源组

```
/usr/cluster/bin/clresourcegroup -V
/usr/cluster/bin/clresourcegroup [subcommand] -?
/usr/cluster/bin/clresourcegroup subcommand [options] -v [resourcegroup ...]
/usr/cluster/bin/clresourcegroup add-node -n node[,...] [-S] [-Z {zoneclustername | global}]
{+ | resourcegroup...}
/usr/cluster/bin/clresourcegroup create [-S] [-n node[,...]] [-p name=value] [...] [-Z
{zoneclustername | global}] {+ | resourcegroup...}
/usr/cluster/bin/clresourcegroup create -i {- | clconfigfile} [-S] [-n node [,...]] [-p
name=value] [...] {+ | resourcegroup...}
/usr/cluster/bin/clresourcegroup delete [-F] [-Z {zoneclustername | global}] {+
| resourcegroup...}
/usr/cluster/bin/clresourcegroup evacuate -n node[,...] [-T seconds] [-Z {zoneclustername
| global}] {+}
/usr/cluster/bin/clresourcegroup export [-o {- | configfile}] {+ | resourcegroup...}
/usr/cluster/bin/clresourcegroup list [-n node[,...]] [-r resource[,...]] [-s state[,...]] [-t
resourcetype[,...]] [ [-Z {zoneclustername[,...] | global | all}] {+ | resourcegroup...}
/usr/cluster/bin/clresourcegroup manage [-Z {zoneclustername | global}] {+
| resourcegroup...}
/usr/cluster/bin/clresourcegroup offline [-n node [,...]] [-Z {zoneclustername | global}]
{+ | resourcegroup...}
/usr/cluster/bin/clresourcegroup online [-e] [-m] [-M] [-n node [,...]] [-Z {zoneclustername
| global}] {+ | resourcegroup...}
/usr/cluster/bin/clresourcegroup quiesce [-k] [-Z {zoneclustername | global}] {+
| resourcegroup...}
/usr/cluster/bin/clresourcegroup remaster [-Z {zoneclustername |
global}] {+ | resourcegroup...}
/usr/cluster/bin/clresourcegroup remove-node -n node
[,...] [-Z {zoneclustername | global}]
{+ | resourcegroup...}
/usr/cluster/bin/clresourcegroup restart [-n node[,...]]
[-Z zoneclustername |global] {+ | resourcegroup...}
/usr/cluster/bin/clresourcegroup resume [-Z {zoneclustername |
global}] {+ | resourcegroup...}
/usr/cluster/bin/clresourcegroup set [-n node[,...]] -p name[+|-]=value [...]
[-Z {zoneclustername | global}] {+ | resourcegroup...}
```

```

/usr/cluster/bin/clresourcegroup show [-n node[,...]]
    [-p name[,...]] [-r resource[,...]] [-t resourcetype[,...]]
    [-Z {zoneclustername[,...] | global | all}]
    [+ | resourcegroup...]

/usr/cluster/bin/clresourcegroup status [-n node[,...]]
    [-r resource [,]...] [-s state [,]...] [-t resourcetype
    [,]...] [-Z {zoneclustername[,...] | global | all}]
    [+ | resourcegroup...]

/usr/cluster/bin/clresourcegroup suspend [-k] [-Z
    {zoneclustername | global}] {+ | resourcegroup...}

/usr/cluster/bin/clresourcegroup switch -n node[,...]
    [-e] [-m] [-M] [-Z {zoneclustername | global}]
    {+ | resourcegroup...}

/usr/cluster/bin/clresourcegroup unmanage [-Z {zoneclustername |
    global}] {+ | resourcegroup...}

```

该命令管理 Oracle Solaris Cluster 数据服务资源组。

仅当 *options* 是 *-?* 选项或 *-v* 选项时，才可以省略 *subcommand*。

每个选项都有长和短两种格式。在 *选项* 中给出每个选项的两种格式以及该选项的描述。

clrg 命令是 *clresourcegroup* 命令的简短格式。

子命令需要至少一个操作数，但 *list*、*show* 和 *status* 除外。但是，很多子命令都支持加号操作数 (+)。该操作数将子命令应用于所有适用的对象。

可以在区域群集中使用该命令的某些格式。有关有效使用该命令的更多信息，请参见各个子命令的描述。为便于管理，请从全局群集节点使用此命令。

资源和资源组

资源状况、资源组状况和资源状态均是基于每个节点进行维护的。例如，一个给定资源在每个群集节点上可以具有不同的状况和状态。

注 - 状况名称 (例如 *Offline* 和 *Start_failed*) 不区分大小写。指定状况名称时，可以使用大小写字母的任意组合。

资源状态是由资源组管理器 (Resource Group Manager, RGM) 仅根据对资源调用的方法在每个节点上设置的。例如，在给定节点上对某个资源成功运行 *STOP* 方法后，该资源在此节点上的状态将为 *Offline*。如果 *STOP* 方法返回非零值或超时，则该资源的状态为 *Stop_failed*。

可能的资源状态包括：*Online*、*Offline*、*Start_failed*、*Stop_failed*、*Monitor_failed*、*Online_not_monitored*、*Starting* 和 *Stopping*。

可能的资源组状态包括：Unmanaged、Online、Offline、Pending_online、Pending_offline、Error_stop_failed、Online_faulted 和 Pending_online_blocked。

除资源状况外，RGM 还可以维护资源状态，资源状态可由资源本身使用 API 设置。Status Message 字段实际上包括两个组成部分：状态关键字和状态消息。状态消息由资源有选择性地设置，它是在状态关键字后面输出的任意文本字符串。

资源状态可能值的描述如下：

Degraded	资源已联机，但其性能或可用性可能在某方面出现了下降。
Faulted	资源遇到错误，无法正常运行。
Offline	资源已脱机。
Online	资源已联机，正在提供服务。
Unknown	当前状态是未知或正在转换。

在区域群集中使用此命令

在区域群集中，可以将 `clresourcegroup` 命令与所有子命令（`export` 除外）一起使用。

也可以将 `-z` 选项与所有子命令（`export` 除外）一起使用来指定要将操作限制到的特定区域群集的名称。另外，还可以将区域群集名称附加到资源组 (`zoneclustername:resourcegroup`)，以便将操作限定到特定的区域群集。

您可以从全局群集节点访问所有区域群集信息，但无法从特定区域群集访问其他区域群集信息。如果不将操作限制到特定区域群集，则所用的子命令仅会在当前群集中运行。

您可以指定某区域群集中的资源组与另一个区域群集中的资源组（或全局群集中的资源组）之间的关联性。可以使用以下命令指定不同区域群集中的资源组之间的关联性：

```
# clresourcegroup set -p RG_affinities={+|++|-|--}
target-zc:target-rg
source-zc:source-rg
```

关联性类型可以是以下类型之一：

- + (弱正)
- ++ (强正)
- +++ (包含故障转移委托的强正)
- - (弱负)
- -- (强负)

例如，如果需要指定区域群集 ZC1 中的资源组 RG1 与区域群集 ZC2 中的资源组 RG2 之间的强正关联性 (++)，请使用以下命令：

```
# clresourcegroup set -p RG_affinities=++ZC2:RG2 ZC1:RG1
```

要指定区域群集 ZC1 中的资源组 RG1 和区域群集 ZC2 中的资源组 RG2 之间包含故障转移委托的强正关联性 (+++), 请使用以下命令 :

```
# clresourcegroup set -p RG_affinities=+++ZC2:RG2 ZC1:RG1
```

要指定区域群集 ZC1 中的资源组 RG1 和全局群集中的资源组 RG2 之间的强负关联性 (--), 请使用以下命令 :

```
# clresourcegroup set -p RG_affinities=-global:RG2 ZC1:RG1
```

资源组可以跨群集节点或区域自动分布。有关更多信息, 请参见 [rg_properties\(5\) \[1129\]](#) 手册页中的 Load_factors、Priority 和 Preemption_mode 项。

支持以下子命令 :

add-node

将节点添加到某个资源组的 Nodelist 属性的末尾。

您可以在全局群集中或区域群集中使用该子命令。

列表中节点和区域的顺序指定了在这些节点或区域上使资源组联机的首选顺序。要将节点添加到 Nodelist 属性中的其他位置, 请使用 set 子命令。

要从全局群集节点为特定区域群集中的资源组添加节点, 您可以使用 -z 选项指定该区域群集的名称。

超级用户以外的用户需要具有 solaris.cluster.modify 基于角色的访问控制 (Role-Based Access Control, RBAC) 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

create

创建新的资源组。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集节点在特定区域群集中创建资源组, 您可以使用 -z 选项指定该区域群集的名称。

如果您使用 -i 选项指定了一个配置文件, 则可以指定加号操作数 (+)。该操作数指定您要创建该文件中提供的尚不存在的所有资源。

要设置新资源组的 Nodelist 属性, 请指定以下选项之一 :

- -n *node*
- -p Nodelist=-*node-*][,...]
- -i *clconfigfile*

列表中节点的顺序指定了在这些节点上使资源组联机的首选顺序。如果在创建时没有指定节点列表, 则 Nodelist 属性会设置为在群集中配置的所有节点。顺序是任意的。

默认情况下，创建资源组时 `RG_mode` 属性已设置为 `Failover`。但是，通过使用 `-s` 选项或 `-p RG_mode=Scalable` 选项，或将 `Maximum primaries` 设置为大于 1 的值，可以创建可伸缩的资源组。仅当创建资源组后才可以设置该资源组的 `RG_mode` 属性。

创建资源组时始终会将资源组置于不受管理状态。但在您发出 `manage` 子命令或使用 `-M` 选项发出 `online` 或 `switch` 子命令时，RGM 会将其状态更改为受管理状态。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

delete

删除资源组。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集节点删除特定区域群集中的资源组，您可以使用 `-z` 选项指定该区域群集的名称。

可以随该子命令指定加号操作数 (+) 以删除所有资源组。

如果资源组包含资源，除非指定了 `-F` 选项，否则无法删除这些资源组。如果指定了 `-F` 选项，则每个组及其中的所有资源都会被删除。还会同时删除所有依赖性和关联性。

该子命令将按照反映资源和资源组依赖性的顺序删除多个资源组。在命令行中指定资源组的顺序无关紧要。

`clresourcegroup delete` 命令的以下格式是通过几个步骤完成的：

- 同时删除多个资源组时
- 使用 `-F` 选项删除某个资源组时

如果以上任一格式的命令被中断（例如，节点出现故障），则某些资源组的配置可能会处于无效状态。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

evacuate

使通过 `-n` 选项指定的节点上的所有资源组脱机。

您可以在全局群集中或区域群集中使用该子命令。

从全局群集节点运行 `evacuate` 命令时，该子命令会清除该全局群集或区域群集中的所有资源组。在区域群集中，该子命令仅清除指定区域群集中的资源组。要从全局群集节点清除特定区域群集中的资源组，您可以使用 `-z` 选项指定该区域群集的名称。

资源组会按照反映资源和资源组依赖性的顺序进行脱机。

您可以将 `-t` 选项与该子命令一起使用，以指定用于阻止资源组切换回的秒数。如果未指定值，将使用默认值 60 秒。

在清除完成之后的 60 秒或指定的秒数内，防止资源组在清除的节点上进行故障转移或自动联机。

但是，如果您使用 `switch` 或 `online` 子命令使资源组联机，或清除的节点重新引导，则清除计时器将立即过期并再次允许自动故障转移。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

`export`

将资源组的配置信息写入一个文件或标准输出 (`stdout`)。

只能在全局群集中使用该子命令。

[clconfiguration\(5CL\) \[1245\]](#) 手册页中描述了该配置信息的格式。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

`list`

显示按您指定的限定符选项过滤的资源组的列表。

您可以在全局群集中或区域群集中使用该子命令。

可以使用 `-r resource` 来仅包括含有资源的资源组。可以使用 `-t resourcetype` 来仅包括含有 `resourcetype` 中某个资源类型的资源组。可以使用 `-n node` 来仅包括在一个或多个节点上处于联机状态的资源组。

如果指定了 `-s state`，则仅会列出具有指定状态的组。

如果未指定操作数或指定了加号操作数 (+)，则会列出按您指定的任何限定符选项过滤的所有资源组。

如果指定了详细选项 `-v`，则会显示状态（资源组是处于联机还是脱机状态）。即使资源组只在群集中的一个节点上处于联机状态，列出该资源组时该资源组也会显示为联机状态。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

`manage`

将您所指定的资源组置于被管理状态。

您可以在全局群集中或区域群集中使用该子命令。

如果从全局群集节点使用该子命令，则该子命令可以对任何资源组进行操作。如果在区域群集中使用该子命令，则该子命令只能对区域群集中的资源组成功进行操作。要从全局群集节点管理特定区域群集中的资源组，可以使用 `-z` 选项指定该区域群集的名称。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

`offline`

将您所指定的资源组置于脱机状态。

您可以在全局群集中或区域群集中使用该子命令。

如果从全局群集节点使用该子命令，则该子命令可以对任何资源组进行操作。如果在区域群集中使用该子命令，则该子命令只能对区域群集中的资源组成功进行操作。要从全局群集节点使特定区域群集中的资源组脱机，可以使用 `-z` 选项指定该区域群集的名称。

如果指定了 `-n` 选项，则只会使资源组在指定的节点上脱机。

如果未指定 `-n` 选项，则会使资源组在所有节点上脱机。

如果使用 `offline` 子命令使某资源组脱机，则重新引导节点后该资源组的 `Offline` 状态不会保留。换句话说，如果节点停止或加入群集，则资源组可以在某个节点中进入联机状态，即使您先前已将资源组切换到脱机状态。即使所有资源都被禁用，该资源组还是会进入联机状态。

同样，声明任何 `RG_dependencies` 或强 `RG_affinities` 的资源组可以在切换到其他资源组时自动进入联机状态。

要防止资源组自动进入联机状态，请使用 `suspend` 子命令暂停该资源组的自动恢复操作。要继续自动恢复操作，请使用 `resume` 子命令。

资源组会按照反映资源和资源组依赖性的顺序进行脱机。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5" 手册页](#)。

online

将您所指定的资源组置于联机状态。

您可以在全局群集中或区域群集中使用该子命令。

如果从全局群集节点使用该子命令，则该子命令可以对任何资源组进行操作。如果在区域群集中使用该子命令，则该子命令只能对区域群集中的资源组成功进行操作。要从全局群集节点使特定区域群集中的资源组联机，可以使用 `-z` 选项指定该区域群集的名称。

使用 `-n` 选项指定要使资源组在其上联机的节点的列表。如果未指定 `-n` 选项，则该子命令会使资源组在它们最偏爱的节点上联机，且不使这些组从其当前的任何主节点脱机。每个资源组的联机节点总数受 `Desired primaries` 和 `Maximum primaries` 属性限制。节点的首选排序由 `Nodelist`、`RG_affinities` 和 `Load_factors` 属性决定。有关这些属性的更多信息，请参见 [rg_properties\(5\) \[1129\]](#) 手册页。

当命令行中提供了多个资源组操作数并且未指定 `-n` 选项时，系统会按照由 `Priority` 属性确定的顺序为这些资源组操作数分配主节点，其中优先级最高的资源组会首先获得节点分配。分配完主节点后，所有资源组操作数会并行脱机，除非受到资源依赖性 or 资源组依赖性的约束。在命令行中指定资源组的顺序无关紧要。有关 `Priority` 属性的更多信息，请参见 [rg_properties\(5\) \[1129\]](#) 手册页。

优先级较低的资源组可能无法分配到其最偏爱的节点，或者当超出负载限制时可能受优先级较高的资源组影响而强制脱机。有关更多信息，请参见 [clnode\(1CL\) \[149\]](#) 手册页中的 `loadlimit` 子命令。

与 `switch` 子命令不同，该子命令不会尝试将 `Nodelist` 属性中列出的任何节点切换到 `Offline` 状态。

如果随该子命令指定了 `-e` 选项，则会启用资源组集中已联机的所有资源。

可以指定 `-m` 选项来监视处于联机状态的资源组集中的所有资源。但是，除非资源已启用并与某个 `MONITOR_START` 方法相关联，否则系统不会真正监视它们。

您还可以指定 `-M` 选项以指示将处于联机状态的所有资源组置于受管理状态。如果未指定 `-M` 选项，则该子命令对不受管理的资源组没有任何影响。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

quiesce

将您所指定的资源组置于停止状态。

您可以在全局群集中或区域群集中使用该子命令。

如果从全局群集节点使用该子命令，则该子命令可以对任何资源组进行操作。如果在区域群集中使用该子命令，则该子命令只能对区域群集中的资源组成功进行操作。要从全局群集节点对特定区域群集中的资源组进行操作，您可以使用 `-z` 选项指定该区域群集的名称。

如果 `START` 或 `STOP` 方法失败，该命令将阻止资源组从一个节点连续切换到另一个节点。如果停止方法失败并将资源的 `Failover_mode` 属性设置为 `HARD`，它还将阻止节点重新引导（通常情况下会发生重新引导）。在这种情况下，资源将改为 `STOP_FAILED` 状态。

使用 `-k` 选项可中止代表受影响资源组中的资源运行的方法。如果未指定 `-k` 选项，系统会允许方法继续运行，直到这些方法退出或超过所配置的超时时间。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

remaster

将指定的资源组从它们当前的主节点切换到它们最偏爱的节点。每个资源组的联机节点总数受 `Desired primaries` 和 `Maximum primaries` 属性限制。节点的首选排序由 `Nodelist`、`RG_affinities` 和 `Load_factors` 属性决定。有关更多信息，请参见 [clnode\(1CL\) \[149\]](#) 和 [rg_properties\(5\) \[1129\]](#) 手册页。

您可以在全局群集中或区域群集中使用该子命令。

如果从全局群集节点使用该子命令，则该子命令可以对任何资源组进行操作。如果在区域群集中使用该子命令，则该子命令只能对区域群集中的资源组成功进行操作。要从全局群集节点对特定区域群集中的资源组进行操作，可以使用 `-z` 选项指定该区域群集的名称。

与 `online` 子命令不同，该子命令可以使资源组在其当前主节点上脱机，并使其在更优先的主节点上联机。

当命令行中提供了多个资源组操作数时，系统会按照由 `Priority` 属性确定的顺序为这些资源组操作数分配主节点，其中优先级最高的资源组会先获得节点分配。在命令行中指定资源组的顺序无关紧要。有关更多信息，请参见 [rg_properties\(5\) \[1129\]](#) 手册页。

优先级较低的资源组可能无法分配到其最偏爱的节点，或者当超出负载限制时可能受优先级较高的资源组影响而强制脱机。有关更多信息，请参见 [clnode\(1CL\) \[149\]](#) 手册页的 `loadlimit` 子命令。

该子命令对不受管理的资源组不会产生任何影响。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

remove-node

从资源组的 `Nodelist` 属性中删除节点。

您可以在全局群集中或区域群集中使用该子命令。

您可以从全局群集节点或区域群集使用该子命令。要从全局群集节点为某区域群集中的某资源组删除节点，您可以使用 `-z` 选项指定该区域群集的名称。

在删除节点后，`remove-node` 可能会将 `Maximum primaries` 或 `Desired primaries` 属性的值重置为 `Nodelist` 属性中的新节点数。仅当任意一个值超过 `Nodelist` 属性中的新节点数时，`remove-node` 才会重置 `Maximum primaries` 或 `Desired primaries` 属性的值。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

restart

使某个资源组在当前承载该资源组的一组主节点上脱机后再重新联机。

您可以在全局群集中或区域群集中使用该子命令。

如果从全局群集节点使用该子命令，则该子命令可以对任何资源组进行操作。如果在区域群集中使用该子命令，则该子命令只能对区域群集中的资源组成功进行操作。要从全局群集节点对特定区域群集中的资源组进行操作，可以使用 `-z` 选项指定该区域群集的名称。

如果指定了 `-n` 选项，则资源组仅在您指定的节点列表中包含的当前主节点上重新启动。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

resume

在之前由 `suspend` 子命令暂停的指定资源组上继续执行自动恢复操作。

您可以在全局群集中或区域群集中使用该子命令。

如果从全局群集节点使用该子命令，则该子命令可以对任何资源组进行操作。如果在区域群集中使用该子命令，则该子命令只能对区域群集中的资源组成功进行操作。要从全局群集节点对特定区域群集中的资源组进行操作，可以使用 `-z` 选项指定该区域群集的名称。

在您没有明确发出命令继续执行自动恢复过程之前，已暂停的资源组不会自动重新启动或进行故障转移。无论是联机还是脱机，已暂停的数据服务都仍将处于其当前

状态。您仍可在指定的节点上将资源组手动切换到不同的状态。此外，您依然可以启用或禁用资源组中的单个资源。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

set

修改与您指定的资源组相关联的属性。

您可以在全局群集中或区域群集中使用该子命令。

如果从全局群集节点使用该子命令，则该子命令可以对任何资源组进行操作。如果在区域群集中使用该子命令，则该子命令只能对区域群集中的资源组成功进行操作。要从全局群集节点对特定区域群集中的资源组进行操作，可以使用 `-z` 选项指定该区域群集的名称。

您可以使用 `-p Nodelist=node` 或 `-n node` (简便方式) 修改 `Nodelist` 属性。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

show

为您指定的资源组生成配置报告 (按限定符选项过滤)。

您可以在全局群集中或区域群集中使用该子命令。

可以使用 `-r resource` 来仅包括含有资源的资源组。可以使用 `-t resourcetype` 来仅包括含有 `resourcetype` 中某个资源类型的资源组。可以使用 `-n node` 来仅包括在一个或多个节点上处于联机状态的资源组。您可以从全局群集使用 `-z` 选项来仅包括在指定区域群集中处于联机状态的那些资源组。

您可以使用 `-p` 选项显示选定的一组资源组属性 (而非所有资源组属性)。

如果未指定操作数或指定了加号操作数 (+)，则会列出按您指定的任何限定符选项过滤的所有资源组。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

status

为您指定的资源组生成状态报告 (按限定符选项过滤)。

您可以在全局群集中或区域群集中使用该子命令。

如果在区域群集中使用该命令，则该子命令仅应用于该区域群集中的资源组。

可以使用 `-r resource` 来仅包括含有资源的资源组。可以使用 `-t resourcetype` 来仅包括含有 `resourcetype` 中某个资源类型的资源组。可以使用 `-n node` 来仅包括在一个或多个节点上处于联机状态的资源组。您可以从全局群集节点使用 `-z` 选项指定一个区域群集，以便仅包括在指定区域群集中处于联机状态的那些资源组。

如果指定了 `-s state`，则仅会列出具有指定状态的组。

注 - 您可以在 `status` 子命令中指定 `-n` 选项或 `-s` 选项。但是您不能随 `status` 子命令同时指定这两个选项。

如果未指定操作数或指定了加号操作数 (+)，则会列出按您指定的任何限定符选项过滤的所有资源组。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

suspend

暂停自动恢复操作并停止指定的资源组。

您可以在全局群集中或区域群集中使用该子命令。

如果在全局群集节点中使用该子命令，则该子命令可以对任何资源组进行操作。如果在区域群集中使用该子命令，则该子命令只能对区域群集中的资源组成功进行操作。要从全局群集节点对特定区域群集中的资源组进行操作，可以使用 `-z` 选项指定该区域群集的名称。

在您没有明确发出命令继续执行自动恢复过程之前，已暂停的资源组不会自动重新启动或进行故障转移。无论是联机还是脱机，已暂停的数据服务都仍将处于其当前状态。资源组已暂停时，可以通过使用带有子命令（如 `switch`、`online`、`offline`、`disable` 或 `enable`）的 `clresourcegroup(1CL)` 或 `clresource(1CL)` 命令，在特定的节点上将资源组或其资源手动切换到其他状态。无需直接对资源执行操作（如中止应用程序进程或运行应用程序特定的命令），只需使用 `clresourcegroup(1CL)` 或 `clresource(1CL)` 命令即可。这样，群集框架就可以保持资源和资源组的当前状态的准确情形，以便在执行 `resume` 子命令时正确恢复可用性。

为了检查并修复群集中存在的问题或在资源组服务上执行维护，您可能需要暂停资源组的自动恢复过程。

还可以指定 `-k` 选项以立即中止代表受影响资源组中的资源运行的方法。使用 `-k` 选项可以加速资源组的停顿。如果未指定 `-k` 选项，系统会允许方法继续运行，直到这些方法退出或超过所配置的超时时间。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

switch

更改控制您所指定的资源组的一个节点或一组节点。

您可以在全局群集中或区域群集中使用该子命令。

如果在全局群集节点中使用该子命令，则该子命令可以对任何资源组进行操作。如果在区域群集中使用该子命令，则该子命令只能对区域群集中的资源组成功进行操作。

使用 `-n` 选项指定要使资源组在其上联机的节点的列表。您可以从全局群集节点使用 `-z` 选项指定一个区域群集，以便仅包括指定区域群集中的资源组列表。

如果某个资源组尚未联机，则该资源组将在由 `-n` 选项指定的一组节点上联机。但是，处于联机状态的组在新节点上联机之前，会在 `-n` 选项未指定的节点上脱机。

如果随该子命令指定了 `-e`，则会启用资源组集中已联机的所有资源。

您可以指定 `-m` 来监视处于联机状态的资源组集中的所有资源。但是，除非资源已启用并与某个 `MONITOR_START` 方法相关联，否则系统不会真正监视它们。

可以指定 `-M` 选项以指示将处于联机状态的所有资源组置于受管理状态。如果未指定 `-M` 选项，则该子命令对不受管理的资源组没有任何影响。

资源组会按照反映资源和资源组依赖性的顺序进行联机。在命令行中指定组的顺序无关紧要。

优先级较低的资源组可能无法切换到指定的节点，或者当超出负载限制时甚至可能受优先级较高的资源组影响而强制脱机。有关更多信息，请参见 [clnode\(1CL\) \[149\]](#) 手册页中的 `loadlimit` 子命令。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

unmanage

将您所指定的资源组置于不被管理的状态。

您可以在全局群集中或区域群集中使用该子命令。

如果从全局群集节点使用该子命令，则该子命令可以对任何资源组进行操作。如果在区域群集中使用该子命令，则该子命令只能对同一区域群集中的资源组成功进行操作。要从全局群集节点对特定区域群集中的资源组进行操作，可以使用 `-z` 选项指定该区域群集的名称。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

支持以下选项：

注 - 该部分显示了每个选项的短和长两种格式。

`-?`

`--help`

显示帮助信息。

指定此选项时，*subcommand* 可有可无。

如果没有随该选项指定 *subcommand*，将会显示所有可用子命令的列表。

如果随该选项指定了 *subcommand*，将会显示 *subcommand* 的用法。

如果在 `create` 或 `set` 子命令中指定了该选项，将显示所有资源组属性的帮助信息。

如果是与其他选项、子命令或操作数一起指定该选项，将忽略所有这些内容。不会发生任何其他处理。

`-e`
`--enable`
在资源组进入联机状态时，启用该资源组内的所有资源。
此选项只能与 `switch` 和 `online` 子命令一起使用。

`-F`
`--force`
强制删除资源组及其所有资源，即使这些资源处于启用或联机状态。此选项还从其他资源和资源组的任何依赖性属性设置或关联性属性设置中删除资源和资源组。
请谨慎使用 `-F` 选项和 `delete` 子命令。强制删除操作可能会导致引用被删除的资源组的其他资源组发生变化，例如当设置了依赖性 or 关联性时。在执行强制删除操作后，具有依赖性的资源可能会处于无效或错误状态。如果发生这种情况，您可能需要重新配置或重新启动受影响的具有依赖性的资源。

`-i {- | clconfigfile}`
`--input={- | clconfigfile}`
`--input {- | clconfigfile}`
指定您要使用位于 *clconfigfile* 文件中的配置信息。请参见 [clconfiguration\(5CL\) \[1245\]](#) 手册页。
随该选项指定一个破折号 (-) 将通过标准输入 (stdin) 提供配置信息。
如果指定了其他选项，则这些选项优先于 *clconfigfile* 中的选项和信息。
只有您指定的那些资源组才会受此选项的影响。

`-k`
`--kill`
中止代表您指定的资源组中的资源运行的 RGM 资源方法。
可以将此选项与 `quiesce` 和 `suspend` 子命令一起使用。如果未指定 `-k` 选项，系统会允许方法继续运行，直到这些方法退出或超过所配置的超时时间。

`-m`
`--monitor`
在资源组进入联机状态时，监视该资源组内的所有资源。
但是，除非资源已启用并与某个 `MONITOR_START` 方法相关联，否则系统不会真正监视它们。
此选项只能与 `switch` 和 `online` 子命令一起使用。

`-M`
`--manage`
指定通过 `switch` 或 `online` 子命令进行联机的所有资源组均置于受管理状态。

```
-n node[...]  
--node=node[...]  
--node node[...]
```

指定目标全局群集或区域群集中的一个节点或一系列节点。如果指定 `-z` 选项，则仅可以使用 `-n` 选项指定区域群集主机名，而不能指定全局群集主机名。如果未指定 `-z` 选项，则仅可以使用 `-n` 选项指定全局群集主机名。

可以将节点的名称或标识符指定为 `node`。

与 `list`、`show` 和 `status` 子命令结合使用时，该选项会限制输出。仅会包括当前在节点列表中的一个或多个节点上处于联机状态的那些资源组。

在 `create`、`add-node`、`remove-node` 和 `set` 子命令中指定该选项等同于设置 `Nodelist` 属性。`Nodelist` 属性中节点的顺序指定了在这些节点上使该组联机的顺序。如果没有在 `create` 子命令中指定节点列表，则 `Nodelist` 属性会设置为群集中的所有节点。顺序是任意的。

与 `switch` 和 `online` 子命令一起使用时，该选项指定要在其上使资源组联机的节点。

与 `evacuate` 和 `offline` 子命令一起使用时，该选项指定要在其上使资源组脱机的节点。

与 `restart` 子命令一起使用时，该选项指定要在其上重新启动资源组的节点。资源组在位于指定列表中的当前主节点上重新启动。

```
-o {- | clconfigfile}  
--output={- | clconfigfile}  
--output {- | clconfigfile}
```

将资源组配置信息写入一个文件或标准输出 (stdout)。[clconfiguration\(5CL\) \[1245\]](#) 手册页中描述了该配置信息的格式。

如果您使用该选项指定文件名称，该选项将创建新的文件。然后将配置信息置于该文件中。如果您使用该选项指定 `-`，则会将配置信息发送到标准输出 (stdout)。命令的所有其他标准输出将受到抑制。

仅可以将此选项与 `export` 子命令一起使用。

```
-p name  
--property=name  
--property name
```

指定资源组属性列表。

将此选项与 `show` 子命令一起使用。

有关您可以使用 `create` 或 `set` 子命令设置或修改的属性的信息，请参见 `-p name=value` 选项的相关描述。

如果没有指定该选项，`show` 子命令会列出大部分资源组属性。如果没有指定该选项但在 `show` 子命令中指定了 `-verbose` 选项，则该子命令会列出所有资源组属性。

[Unresolved link to "Oracle Solaris Cluster 数据服务规划和管理指南 中的资源组属性"](#)描述了您可以指定的资源组属性。

```
-p name=value
-p name+=array-values
-p name=array-values
--property=name=value
--property=name+=array-values
--property=name-=array-values
--property name=value
--property name+=array-values
--property name-=array-values
```

设置或修改资源组属性的值。

此选项只能与 `create` 和 `set` 子命令一起使用。

有关可以使用 `show` 子命令显示其相关信息的属性的信息，请参见 `-p name` 选项的描述。

允许多个 `-p` 实例。

与此选项一起使用的运算符如下所述：

<code>=</code>	将属性设置为指定值。 <code>create</code> 和 <code>set</code> 子命令接受此运算符。
<code>+=</code>	将一个或多个值添加到属性值列表。只有 <code>set</code> 子命令接受此运算符。仅可以为接受字符串值列表的属性指定此运算符，例如 <code>Nodelist</code> 。
<code>-=</code>	从属性值列表中删除一个或多个值。只有 <code>set</code> 子命令接受此运算符。仅可以为接受字符串值列表的属性指定此运算符，例如 <code>Nodelist</code> 。

```
-r resource[...]
--resource=resource[...]
--resource resource[...]
```

指定一个资源或资源列表。

此选项只能与 `list`、`show` 和 `status` 子命令一起使用。该选项限制这些命令的输出。只会输出包含资源列表中的一个或多个资源的那些资源组。

```
-s state[...]
--state=state[...]
--state state[...]
```

指定一个资源组状态或资源组状态列表。

此选项只能与 `status` 子命令一起使用。该选项限制输出，以便仅显示在任何指定节点上处于指定状态的那些资源组。您可以随该选项指定一个或多个以下参数（状态）：

`Error_stop_failed`

显示在您指定的任何节点上处于 `Error_stop_failed` 状态的任何指定资源组。

Not_online

显示在您指定的任何节点上处于 online 之外的任何状态的任何指定资源组。

Offline

仅当某个指定资源组在您指定的所有节点上处于 Offline 状态时才显示该资源组。

Online

显示在您指定的任何节点上处于 Online 状态的任何指定资源组。

Online_faulted

显示在您指定的任何节点上处于 Online_faulted 状态的任何指定资源组。

Pending_offline

显示在您指定的任何节点上处于 Pending_offline 状态的任何指定资源组。

Pending_online

显示在您指定的任何节点上处于 Pending_online 状态的任何指定资源组。

Pending_online_blocked

显示在您指定的任何节点上处于 Pending_online_blocked 状态的任何指定资源组。

Unmanaged

显示在您指定的任何节点上处于 Unmanaged 状态的任何指定资源组。

-S

--scalable

创建一个可伸缩资源组或更新 Maximum primaries 和 Desired primaries 属性。此选项只能与 create 和 add-node 子命令一起使用。

与 create 子命令一起使用时，该选项可以创建可伸缩的资源组，而非故障转移资源组。该选项还会将 Maximum primaries 和 Desired primaries 属性设置为生成的 Nodelist 属性中的节点数。

仅当资源组已经是可伸缩的时，才可以将该选项与 add-node 子命令一起使用。与 add-node 子命令一起使用时，该选项会将 Maximum primaries 和 Desired primaries 属性设置为生成的 Nodelist 属性中的节点数。

也可以使用 -p 选项设置 RG_mode、Maximum primaries 和 Desired primaries 属性。

-t *resourcetype*[...]

--type=*resourcetype*[...]

--type *resourcetype*[...]

指定一个资源类型或资源类型列表。

此选项只能与 `list`、`show` 和 `status` 子命令一起使用。该选项限制这些命令的输出。只会输出包含属于资源类型列表中的类型的一个或多个资源的那些资源组。

您将资源类型指定为 `[prefix.] type[:RT-version]`。例如，`nfs` 资源类型可以表示为 `SUNW.nfs:3.2`、`SUNW.nfs` 或 `nfs`。仅当在群集中注册的一个资源类型具有多个版本时，您才需要包括 `RT-version`。如果没有包括 `prefix`，则会采用 `SUNW`。

`-T seconds`

`--time=seconds`

`--time seconds`

指定在从节点中清除资源组之后阻止资源组切换回该节点的秒数。

仅可以将此选项与 `evacuate` 子命令一起使用。您必须将 `seconds` 指定为介于 0 和 65535 之间的整数值。如果未指定值，将使用默认值 60 秒。

在清除完成之后的 60 秒或指定的秒数内，防止资源组在清除的节点上进行故障转移或自动联机。

但是，如果您使用 `switch` 或 `online` 子命令使资源组联机，或清除的节点重新引导，则清除计时器将立即过期并再次允许自动故障转移。

`-T` 选项指定在完成清除后的 `T` 秒时间段内 RGM 不会在清除的节点上使资源组联机。通过将 `switch` 或 `online` 子命令与 `-n` 选项一起使用将资源组切换到清除的节点，可以覆盖 `-T` 计时器。完成此种切换后，`-T` 计时器对该节点会立即过期。但是，切换命令（例如不含 `-n` 标志的 `online` 或 `remaster`）会继续使用 `-T` 计时器，并且避免将任何资源组切换到清除的节点。

`-u`

如果使用 `+ 操作数`，此选项会指定命令要对其资源组已暂停的资源起作用。

如果指定 `+ 操作数` 时没有指定 `-u` 选项，则该命令会忽略所有暂停的资源组。在 `add-node`、`manage`、`offline`、`online`、`quiesce`、`remaster`、`remove-node`、`restart`、`set`、`switch` 或 `unmanage` 子命令中指定了 `+ 操作数` 时，`-u` 选项有效。

将 `+ 操作数` 与 `add-node`、`manage`、`offline`、`online`、`quiesce`、`remaster`、`remove-node`、`restart`、`set`、`switch` 或 `unmanage` 子命令一起使用时，该命令会忽略所有暂停的资源组，除非您还指定了 `-u` 选项。

`-v`

`--verbose`

在标准输出 (stdout) 中显示详细信息。

`-V`

`--version`

显示该命令的版本。

如果是与其他选项、子命令或操作数一起指定该选项，将忽略所有这些内容。仅显示该命令的版本。不会发生任何其他处理。

-z {zoneclustername | global | all}
--zoneclustername={zoneclustername | global | all}
--zoneclustername {zoneclustername | global | all}

指定其中存在资源组并且您要对其执行操作的一个或多个群集。

除 export 子命令以外的所有子命令均支持此选项。

如果指定此选项，还必须指定以下列表中的参数之一：

zoneclustername	指定使用此选项的命令仅对名为 zoneclustername 的区域群集中的所有指定资源组起作用。
global	指定使用此选项的命令仅对全局群集中的所有指定资源组起作用。
all	如果在全局群集中使用此参数，则会指定使用此参数的命令对所有群集（包括全局群集和所有区域群集）中的所有指定资源组起作用。 如果在某区域群集中使用此参数，则会指定使用此参数的命令仅对该区域群集中的所有指定资源类型起作用。

支持以下操作数：

resourcegroup	要管理的资源组的名称。
+	所有资源组。

[Intro\(1CL\) \[17\]](#) 手册页列出了该命令集中所有命令的一整套退出状态代码。返回的退出代码还与 [scha_calls\(3HA\) \[897\]](#) 手册页中介绍的返回代码兼容。

如果命令对于所有指定的操作数均成功执行，则返回零 (CL_NOERR)。如果某个操作数出错，命令会继续处理操作数列表中的下一个操作数。返回的退出代码始终反映最先发生的错误。

此命令返回下列退出状态代码：

0 CL_NOERR	没有错误 您发出的命令已成功完成。
1 CL_ENOMEM	交换空间不足 某个群集节点耗尽了交换内存或其他操作系统资源。
3 CL_EINVAL	参数无效

您键入的命令不正确，或者使用 `-i` 选项提供的群集配置信息语法不正确。

6 CL_EACCESS

权限被拒绝

指定的对象不可访问。要发出命令，可能需要拥有超级用户或 RBAC 访问权限。有关更多信息，请参见 [Unresolved link to "su1M"](#) 和 [Unresolved link to "rbac5"](#) 手册页。

35 CL_EIO

I/O 错误

发生了物理输入/输出错误。

36 CL_ENOENT

没有这样的对象

由于以下某个原因，找不到您所指定的对象：

- 该对象不存在。
- 您尝试使用 `-o` 选项创建的配置文件路径中的某个目录不存在。
- 您尝试使用 `-i` 选项访问的配置文件包含错误。

38 CL_EBUSY

对象忙

您尝试将一根电缆从上一个群集互连路径移到一个活动群集节点。或者，您尝试将一个节点从尚未删除引用的群集配置中移除。

39 CL_EEXIST

对象已存在

您所指定的设备、设备组、群集互连组件、节点、群集、资源、资源类型、资源组或专用字符串已存在。

例 147 创建新的故障转移资源组

以下示例中的第一个命令将创建故障转移资源组 `rg1` 和 `rg2`。第二个命令将配置文件 `cluster-1.xml` 中包含的资源添加到这些资源组中。

```
# clresourcegroup create rg1 rg2
# clresource create -g rg1,rg2 -i /net/server/export/cluster-1.xml +
```

以下两个示例中的任一个都从全局群集节点在区域群集 `ZC` 中创建故障转移资源组 `rg1` 和 `rg2`。

```
# clresourcegroup create -Z ZC rg1 rg2
# clresourcegroup create ZC:rg1 ZC:rg2
```

例 148 使所有资源组联机

以下命令使所有资源组联机，并启用和监视所有资源。

```
# clresourcegroup online -eM +
```

例 149 将节点添加到 Nodelist 属性中

以下命令将节点 phys-schost-4 添加到所有资源组的 Nodelist 属性中。

```
# clresourcegroup set -p Nodelist+=phys-schost-4 +
```

例 150 从节点清除所有资源组

以下命令从节点 phys-schost-3 清除所有资源组。

```
# clresourcegroup evacuate -n phys-schost-3 +
```

例 151 使资源组在所有节点上脱机

以下命令使资源组 rg1 在所有节点上脱机。

```
# clresourcegroup offline rg1
```

例 152 刷新整个资源组管理器配置

以下示例中的第一个命令将删除所有资源和资源组，即使它们已启用并处于联机状态。第二个命令取消注册所有资源类型。第三个命令创建配置文件 cluster-1.xml 中包含的资源。第三个命令还会注册资源类型并创建这些资源类型所依赖的所有资源组。

```
# clresourcegroup delete --force +
# clresourcetype unregister +
# clresource -i /net/server/export/cluster-1.xml -d +
```

例 153 列出所有资源组

以下命令可列出所有资源组。

```
# clresourcegroup list
rg1
rg2
```

例 154 列出所有资源组及其资源

以下命令可列出所有资源组及其资源。请注意，rg3 没有资源。

```
# clresourcegroup list -v
```

```

Resource Group Resource
-----
rg1          rs-2
rg1          rs-3
rg1          rs-4
rg1          rs-5
rg2          rs-1
rg3          -

```

例 155 列出包含特定资源的所有资源组

以下命令可列出包含 Oracle Solaris Cluster HA for NFS 资源的所有组。

```
# clresource list -t nfs
rg1
```

例 156 通过切换资源组清除 Start_failed 资源状态

Start_failed 资源状况指示 Start 或 Prenet_start 方法对某个资源失败或超时，但该资源的资源组仍已联机。即使该资源处于故障状况，可能不提供服务，其资源组仍然会联机。如果资源的 Failover_mode 属性设置为 None 或者设置为阻止资源组故障转移的另一个值，则可能会出现此状况。

与 Stop_failed 资源状况不同，Start_failed 资源状况不会阻止您或 Oracle Solaris Cluster 软件对资源组执行操作。无需发出 reset 子命令来清除 Start_failed 资源状态。只需执行一个可重新启动资源的命令。

以下命令清除 resource-grp-2 资源组中的资源发生的 Start_failed 资源状态。该命令通过将资源组切换到 schost-2 节点来清除此状态。

```
# clresourcegroup switch -n schost-2 resource-grp-2
```

例 157 通过重新启动资源组清除 Start_failed 资源状态

以下命令清除 resource-grp-2 资源组中的资源发生的 Start_failed 资源状态。该命令通过在初始承载该资源组的 schost-1 节点上重新启动资源组来清除此状态。

```
# clresourcegroup restart resource-grp-2
```

例 158 设置 load_factors 属性

以下命令为两个资源组设置负载因子。

```
# clresourcegroup set -p load_factors=factor1@50,factor2@1 rg1 rg2
```

从全局群集中，以下命令为某个区域群集中的两个资源组设置负载因子。

```
# clresourcegroup set -Z ZC load_factors=factor1@50,factor2@1 rg1 rg2
```

例 159 为资源组设置 priority 属性

以下命令设置资源组的优先级。

```
# clresourcegroup set -p priority=600 rg1
```

rg1 资源组在分配节点时将优先于优先级较低的资源组。在超出硬限制的节点上，rg1 会抢先于优先级较低的其他资源组。如果 rg1 的优先级比另一资源组的优先级高出至少 100，则它在超出软限制的节点上将抢先于该资源组。priority 的默认值为 500。

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

[clresource\(1CL\) \[225\]](#)、[clresourcetype\(1CL\) \[277\]](#)、[cluster\(1CL\) \[461\]](#)、[Intro\(1CL\) \[17\]](#)、[Unresolved link to "su1M"](#)、[scha_calls\(3HA\) \[897\]](#)、[Unresolved link to "attributes5"](#)、[Unresolved link to "rbac5"](#)、[rg_properties\(5\) \[1129\]](#)、[clconfiguration\(5CL\) \[1245\]](#)

超级用户可以运行此命令的所有格式。

所有用户都可以运行带有 -? (帮助) 或 -v (版本) 选项的此命令。

要运行带有其他子命令的 clresourcegroup 命令，超级用户以外的用户需要具有 RBAC 授权。请参见下表。

子命令	RBAC 授权
add-node	solaris.cluster.modify
create	solaris.cluster.modify
delete	solaris.cluster.modify
evacuate	solaris.cluster.admin
export	solaris.cluster.read
list	solaris.cluster.read
manage	solaris.cluster.admin
offline	solaris.cluster.admin
online	solaris.cluster.admin
quiesce	solaris.cluster.admin

子命令	RBAC 授权
remaster	solaris.cluster.admin
remove-node	solaris.cluster.modify
restart	solaris.cluster.admin
resume	solaris.cluster.admin
set	solaris.cluster.modify
show	solaris.cluster.read
status	solaris.cluster.read
suspend	solaris.cluster.admin
switch	solaris.cluster.admin
unmanage	solaris.cluster.admin

名称

clresourcetype, clrt — 管理 Oracle Solaris Cluster 数据服务的资源类型

```
/usr/cluster/bin/clresourcetype [subcommand -?]  
  
/usr/cluster/bin/clresourcetype subcommand -v [options]  
[-Z {zoneclustername | global}] {+ | resourcetype}...  
  
/usr/cluster/bin/clresourcetype add-node -n node[,...]  
[-Z {zoneclustername | global}] {+ | resourcetype}  
  
/usr/cluster/bin/clresourcetype export [-o {- | configfile}]  
{+ | resourcetype}  
  
/usr/cluster/bin/clresourcetype list [ -n node[,...]  
[-Z {zoneclustername[,...] | global | all}] {+ | resourcetype...}  
  
/usr/cluster/bin/clresourcetype list-props [-p [name,...]] [-Z  
{zoneclustername[,...] | global | all}] {+ | resourcetype...}  
  
/usr/cluster/bin/clresourcetype register [-i  
{- | clconfiguration}] [ {-n node  
[,...] | -N}] [-f rtrfile] [-p [name [+ | -]=value,...]  
[-Z {zoneclustername | global}] {+ | resourcetype...}  
  
/usr/cluster/bin/clresourcetype remove-node -n node  
[,...] [-Z {zoneclustername | global}]  
{+ | resourcetype...}  
  
/usr/cluster/bin/clresourcetype set [-n node  
[,...] | -N] [-p [name [+ | -]=value,...] [-Z  
{zoneclustername | global}] {+ | resourcetype...}  
  
/usr/cluster/bin/clresourcetype show [-n node[,...]  
[-Z {zoneclustername[,...] | global | all}] {+ | resourcetype...}  
  
/usr/cluster/bin/clresourcetype unregister [-Z {zoneclustername |  
global}] {+ | resourcetype...}
```

clresourcetype 命令用于管理 Oracle Solaris Cluster 数据服务的资源类型。clrt 命令是 clresourcetype 命令的简短格式。clresourcetype 命令和 clrt 命令是相同的。您可以使用此命令的任一种格式。

为便于管理，请从全局群集节点运行此命令。

在区域群集中，可以将 clresourcetype 命令与所有子命令（export 除外）一起使用。

也可以将 -z 选项与所有子命令（export 除外）一起使用来指定要将操作限制到的特定区域群集的名称。另外，还可以将区域群集名称附加到资源类型名称 (zoneclustername : resourcetype)，以便将操作限制到特定的区域群集。

您可以从全局群集节点访问所有区域群集信息，但无法从特定区域群集访问其他区域群集信息。如果不将操作限制到特定区域群集，则所用的子命令仅会在当前群集中运行。

此命令的一般格式如下所示：

```
clresourcetype [subcommand] [options] [operands]
```

仅当 *options* 指定 *-?* 选项或 *-v* 选项时，才可以省略 *subcommand*。

此命令的每个选项都有长格式和简短格式两种格式。在本手册页的 [选项](#) 部分给出了每个选项的两种格式，以及对该选项的描述。

支持以下子命令：

add-node

将指定的节点添加到节点列表（针对指定为命令操作数的资源类型）。

您可以在全局群集中或区域群集中使用该子命令。

从全局群集节点使用 `add-node` 命令时，可以使用 `-z` 选项指定区域群集的名称。

此子命令接受使用加号 (+) 作为操作数来指定所有资源类型。

超级用户以外的用户需要具有 `solaris.cluster.modify` 基于角色的访问控制 (role-based access control, RBAC) 授权才能使用此子命令。

另请参见 `remove-node` 子命令的描述。

export

以 [clconfiguration\(5CL\) \[1245\]](#) 手册页所述的格式导出群集资源类型配置。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

只能在全局群集中使用该子命令。

list

显示作为命令操作数指定的资源类型的列表。默认情况下，将显示已在群集中注册的所有资源类型。此子命令接受使用加号 (+) 作为操作数来指定已在群集中注册的所有资源类型。

您可以在全局群集中或区域群集中使用该子命令。

在全局群集节点中，此子命令仅显示已在全局群集节点中注册的资源类型。要从全局群集查看已在区域群集中注册的资源类型，可以使用 `-z` 选项指定区域群集。

如果指定 `-nodelist` 选项，将仅显示已注册用于 `nodelist` 中的节点上的资源类型。

如果指定 `-v` 选项，还将显示列表中每种资源类型的节点列表。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

list-props

显示指定资源类型的属性。此子命令接受使用加号 (+) 作为操作数来指定已在群集中注册的所有资源类型。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集查看已在区域群集中注册的资源类型属性，可以使用 -z 选项指定区域群集。

-p 选项可以限制要显示的属性集。

如果指定 -v 选项，还将显示每个属性的描述。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

register

注册作为命令操作数指定的资源类型。必须先注册资源类型，然后才能创建该类型的资源。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集向区域群集注册资源类型，请使用 -z 选项指定区域群集名称。

必须将用于定义每种资源类型的数据服务安装在要使用该资源类型的每个节点上。如果数据服务仅安装在群集节点子集上，请使用 -n *nodelist* 选项指定节点子集。如果资源类型将在群集中所有节点上可用，请指定 -N 选项。在使用 -N 选项时，资源类型还可用于以后可能添加到群集中的任何节点。省略 -N 选项和 -nnodelist 选项相当于指定 -N 选项。要明确指定属性名称，请使用 -p `Installed_nodes= nodelist` 选项。

有关已向群集注册的资源类型的信息，可从用于定义资源类型的资源类型注册 (Resource Type Registration, RTR) 文件获取。RTR 文件的位置和名称通常遵循以下约定：

- RTR 文件通常位于 `/opt/cluster/lib/rgm/rtreg` 目录中。
- RTR 文件的名称通常与资源类型的名称相匹配。

Oracle 提供的所有 RTR 文件的位置和文件名称都遵循这些约定。例如，用于定义 `SUNW.nfs` 资源类型的 RTR 文件包含在文件 `/opt/cluster/lib/rgm/rtreg/SUNW.nfs` 中。

如果 RTR 文件不符合这些约定，必须指定 - *rttrfile* 选项。

这些约定同样适用于从区域群集注册的资源类型。当用户为区域群集注册资源类型时，RTR 文件必须驻留在区域群集 *zonepath* 中。无法在区域群集 *zonepath* 界限之外注册 RTR 文件。在为区域群集注册 *Global_zone* 属性设置为 TRUE 的资源类型时，RTR 文件必须驻留在 `/opt/cluster/lib/rgm/rtreg` 或 `/usr/cluster/lib/rgm/rtreg` 目录中的全局群集节点内。如果指定这些位置之外的任何位置，资源类型将无法注册。



注意 - 不要注册 `Global_zone` 属性设置为 `TRUE` 的资源类型，除非该资源类型来自于已知的可信源。此属性设置为 `TRUE` 的资源类型会规避区域隔离并引起风险。

此子命令接受使用加号 (+) 作为操作数来指定尚未注册的所有资源类型。可用资源类型的完整列表按如下所示进行确定：

- 如果指定 `-iclconfiguration` 选项，`iclconfiguration` 将定义可用资源类型的完整列表。
- 如果没有指定 `-i` 选项，则可用资源类型的完整列表中仅包含由 Oracle 提供的资源类型。这些资源类型还必须已在节点列表中的所有节点上安装。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

另请参见 `unregister` 子命令的描述。

`remove-node`

从节点列表（操作对象列表中的资源类型均已向这些节点注册）中删除一个节点。此子命令接受使用加号 (+) 作为操作数来指定已在群集中注册的所有资源类型。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集删除区域群集的资源类型，请使用 `-z` 选项指定区域群集名称。

仅可对已为群集中的部分节点而不是所有节点注册的资源类型使用该子命令。因此，如果您在以下情况中使用此子命令，将会发生错误：

- 操作数列表中的资源类型已针对群集中的所有节点进行注册。有关为群集中所有节点注册资源类型的信息，请参见 `-N` 选项的描述。
- 操作数列表中资源类型的 `Installed_nodes` 属性尚未指定群集中节点的子集。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

另请参见 `add-node` 子命令的描述。

`set`

设置作为命令操作数指定的资源类型的属性。此子命令接受使用加号 (+) 作为操作数来指定已在群集中注册的所有资源类型。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集设置区域群集中资源类型的属性，请使用 `-z` 选项指定区域群集名称。

仅可以设置 `rt_properties(5)` [1143] 手册页中定义为 Tunable Any Time 的资源类型属性。

- 可以通过指定 `-nodelist` 选项而不指定 `-p` 选项来修改 `Installed_Nodes` 属性。或者，可以通过使用 `-pInstalled_Nodes= nodelist` 选项明确指定属性名称。
- 对于定义为 Tunable Any Time 的所有其他属性，必须通过使用 `-pproperty = value` 选项明确指定属性名称。

要限制资源类型在其上可用的节点的列表，请指定 `-nodelist` 选项。相反，要指定资源类型在群集中所有节点上可用，请指定 `-N` 选项。在使用 `-N` 选项时，资源类型还可用于以后可能添加到群集中的任何节点。必须指定 `-n` 选项或 `-N` 选项。如果省略这两个选项，该子命令不会更改任何配置信息。

show

显示已在群集中注册的资源类型的相关信息。默认情况下，将显示已注册的所有资源类型的以下信息：

- 与每个资源类型关联的属性的列表
- 用于定义这些属性的参数

如果指定 `-nodelist` 选项，将仅显示已注册用于 `nodelist` 中的节点上的资源类型。

如果指定 `-v` 选项，还将显示每种资源类型的以下信息：

- 为资源类型定义的方法
- 每种方法的超时参数

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集查看已在区域群集中注册的资源类型，可以使用 `-z` 选项指定区域群集名称。

此子命令接受使用加号 (+) 作为操作数来指定已在群集中注册的所有资源类型。如果未提供操作数，将显示已在群集中注册的所有资源类型的相关信息。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

unregister

取消注册作为命令操作数指定的资源类型。此子命令接受使用加号 (+) 作为操作数来指定不存在任何实例的所有已注册资源类型。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集向区域群集取消注册资源类型，请使用 `-z` 选项指定区域群集名称。

在卸载用于定义资源类型的数据服务之前取消注册资源类型。

如果某特定资源类型的资源存在，则无法取消注册该资源类型。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。

另请参见 `register` 子命令的描述。

支持以下选项：

`-?`

`--help`

显示帮助信息。使用该选项时，不会执行任何其他处理。

指定此选项时，子命令可有可无。

如果仅指定此选项而不使用子命令，将会显示此命令的子命令列表。

如果使用子命令且指定此选项，将会显示该子命令的用法选项。

```
-f rtrfile|rtrfiledir
--rtrfile=rtrfile|rtrfiledir
--rtrfile rtrfile|rtrfiledir
```

指定 RTR 文件或包含 RTR 文件的目录的完整路径以用于注册资源类型。此选项仅可以与 `register` 子命令一同指定。

如果指定某个文件，则仅可以注册一种资源类型。

仅当所用的 RTR 文件不符合以下约定时，才需要指定此选项：

- RTR 文件通常位于 `/opt/cluster/lib/rgm/rtreg` 目录中。
- RTR 文件的名称通常与资源类型的名称相匹配。

Oracle 提供的所有 RTR 文件的位置和文件名称都遵循这些约定。例如，用于定义 `SUNW.nfs` 资源类型的 RTR 文件包含在文件 `/opt/cluster/lib/rgm/rtreg/SUNW.nfs` 中。

如果使用 `-i` 选项，则可以为配置信息中指定的任何资源类型指定 `resourcetypeRTRFile` 元素。`resourcetypeRTRFile` 元素可指定要用来注册资源类型的 RTR 文件。但是，`export` 子命令不会在生成的配置信息中包括 `resourcetypeRTRFile` 元素。有关 `resourcetypeRTRFile` 元素的更多信息，请参见 [clconfiguration\(5CL\) \[1245\]](#) 手册页。

```
-i {- | clconfiguration}
--input={- | clconfiguration}
--input {- | clconfiguration}
```

指定用于注册资源类型或用来修改已注册资源类型的节点列表的配置信息。此信息必须符合 [clconfiguration\(5CL\) \[1245\]](#) 手册页中定义的格式。此信息可包含在文件中或者通过标准输入 (`stdin`) 提供。要指定标准输入，请指定 `-` 而不是文件名。

仅作为命令操作数提供的资源类型会受此选项影响。命令中指定的选项将覆盖 `clconfiguration` 文件中设置的任何选项。如果 `clconfiguration` 文件中缺少配置参数，则必须在命令行上指定这些参数。

```
-N
--allnodes
```

指定操作数列表中的资源类型将在群集中的所有节点上可用。`-N` 选项还可以使这些资源类型对以后可能添加到群集中的任何节点可用。该选项可通过清除 `Installed_nodes` 属性实现此结果。

如果指定 `-N` 选项，则无法在同一命令中指定 `-n` 选项。

仅可随 `register` 子命令或 `set` 子命令指定 `-N` 选项。

`-n node[,...]`
`--node=node[,...]`
`--node node[,...]`

指定目标全局群集或区域群集中的一个节点或一系列节点。可以使用节点名称或节点 ID 指定每个节点。

如果指定 `-z` 选项，则仅可以使用 `-n` 选项指定区域群集主机名，而不能指定全局群集主机名。如果未指定 `-z` 选项，则仅可以使用 `-n` 选项指定全局群集主机名。

如果指定 `-n` 选项，则无法在同一命令中指定 `-N` 选项。

可以与此选项一起指定的子命令如下所述：

`add-node`

将指定的节点添加到已注册资源类型的节点的列表中。

`list`

仅显示已注册用于指定节点上的资源类型。

`register`

仅注册用于指定节点上的资源类型。如果省略 `-n` 选项，`register` 子命令将注册用于所有节点上的资源类型。该子命令还会为以后将添加到群集中的任何节点注册资源类型。

`remove-node`

从已注册资源类型的节点的列表中删除指定的节点。

`set`

使资源类型仅在指定节点上可用。

`show`

仅显示已注册用于指定节点上的资源类型的相关信息。

`-o {- | clconfiguration}`
`--output={- | clconfiguration}`
`--output {- | clconfiguration}`

指定资源类型的相关配置信息将写入到的位置。此位置可以是某个文件或标准输出 (`stdout`)。要指定标准输出，请指定 `-` 而不是文件名。如果指定标准输出，将会抑制该命令的其他所有标准输出。此选项仅可以与 `export` 子命令一同指定。

仅会写入作为命令操作数提供的资源类型的配置信息。信息是采用 [clconfiguration\(5CL\) \[1245\]](#) 手册页中定义的格式写入的。

```
-p name=value
-p name+=array-values
-p name-=array-values
--property=name=value
--property=name+=array-values
--property=name-=array-values
--property name=value
--property name+=array-values
--property name-=array-values
```

设置作为命令操作数提供的资源类型的属性值。

与此选项一起使用的运算符如下所述：

= 将属性设置为指定值。

+= 将一个或多个值添加到某个字符串数组值。仅可以为接受字符串值列表的属性指定此运算符，例如 `Installed_nodes`。

-= 从某个字符串数组值中删除一个或多个值。仅可以为接受字符串值列表的属性指定此运算符，例如 `Installed_nodes`。

在 `set` 子命令中使用选项 `-p Installed_nodes+=nodeC,nodeD` 相当于在 `add-node` 子命令中使用选项 `-n nodeC,nodeD`。

```
-p name[...]
--property=name[...]
--property name[...]
```

指定 `list-props` 子命令的属性列表。

```
-V
--version
```

显示该命令的版本。

指定此选项时，不要将其与子命令、操作数或其他选项结合使用。子命令、操作数或其他选项将被忽略。`-v` 选项只显示命令的版本。不会执行任何其他操作。

```
-v
--verbose
```

将详细消息显示到标准输出 (`stdout`)。

可以在任何格式的此命令中指定该选项。

请不要同时指定 `-v` 选项和 `-o` 选项。`-v` 选项将被忽略。`-o` 选项会抑制其他所有标准输出。

```
-z {zoneclustername | global | all}
--zoneclustername={zoneclustername | global | all}
--zoneclustername {zoneclustername | global | all}
```

指定已注册资源类型且要对其执行操作的一个或多个群集。

除 `export` 子命令以外的所有子命令均支持此选项。

如果指定此选项，还必须指定以下列表中的参数之一：

<code>zoneclustername</code>	指定使用此选项的命令仅对名为 <code>zoneclustername</code> 的区域群集中的所有指定资源类型起作用。
<code>global</code>	指定使用此选项的命令仅对全局群集中的所有指定资源类型起作用。
<code>all</code>	如果在全局群集中使用此参数，则会指定使用此参数的命令对所有群集（包括全局群集和所有区域群集）中的所有指定资源类型起作用。 如果在某区域群集中使用此参数，则会指定使用此参数的命令仅对该区域群集中的所有指定资源类型起作用。

仅支持以下操作数：

<code>resourcetype</code>	指定要管理的一种或多种资源类型。如果子命令接受多种资源类型，则可以使用加号 (+) 指定所有资源类型。 有关资源类型名称格式的描述，请参见 Unresolved link to "Oracle Solaris Cluster 数据服务规划和管理指南 中的合法的 RGM 名称" 。
---------------------------	---

如果命令对于所有指定的操作数均成功执行，则返回零 (CL_NOERR)。如果某个操作数出错，命令会继续处理操作数列表中的下一个操作数。返回的退出代码始终反映最先发生的错误。

可返回下列退出代码：

0 CL_NOERR

没有错误

您发出的命令已成功完成。

1 CL_ENOMEM

交换空间不足

某个群集节点耗尽了交换内存或其他操作系统资源。

3 CL_EINVAL

参数无效

您键入的命令不正确，或者使用 `-i` 选项提供的群集配置信息语法不正确。

6 CL_EACCESS

权限被拒绝

指定的对象不可访问。要发出命令，可能需要拥有超级用户或 RBAC 访问权限。有关更多信息，请参见 [Unresolved link to " su1M"](#) 和 [Unresolved link to " rbac5"](#) 手册页。

18 CL_EINTERNAL

遇到内部错误

内部错误表示软件缺陷或其他缺陷。

35 CL_EIO

I/O 错误

发生了物理输入/输出错误。

36 CL_ENOENT

没有这样的对象

由于以下某个原因，找不到您所指定的对象：

- 该对象不存在。
- 您尝试使用 -o 选项创建的配置文件路径中的某个目录不存在。
- 您尝试使用 -i 选项访问的配置文件包含错误。

37 CL_EOP

不允许操作

您尝试对不受支持的配置执行操作，或者您所执行的操作不受支持。

41 CL_ETYPE

类型无效

您使用 -t 或 -p 选项指定的类型不存在。

这些退出值与 [scha_calls\(3HA\) \[897\]](#) 手册页中描述的返回码兼容。

例 160 注册资源类型

本示例注册其数据服务已安装在所有节点上但尚未注册的所有资源类型。该命令以简洁模式运行。

```
# clresourcetype register +
```

例 161 在选定的节点上注册资源类型

本示例注册其数据服务已安装在节点 `phys-schost-1` 和节点 `phys-schost-2` 上但尚未注册的所有资源类型。资源将仅在这些节点上可用。在本示例中，该命令没有返回任何错误。该命令以详细模式运行。

```
# clresourcetype register -v -n phys-schost-1,phys-schost-2 +
```

以下命令将注册其数据服务已安装在区域群集 ZC 的区域群集节点 zc-host-1 和 zc-host-2 上但尚未注册的所有资源类型。资源仅在这些区域群集节点上可用。

```
# clresourcetype register -n zc-host-1,zc-host-2 -Z ZC +
```

例 162 注册单个资源类型

本示例注册 SUNW.nfs:3.2 资源类型。此资源类型的数据服务将安装在所有群集节点上。

```
# clresourcetype register nfs:3.2
```

例 163 列出资源类型

本示例仅列出所有已注册的资源类型的名称。

```
# clresourcetype list
SUNW.LogicalHostname
SUNW.SharedAddress
SUNW.nfs
SUNW.apache
```

例 164 列出资源类型及其节点列表

本示例列出所有已注册的资源类型及其节点列表。

```
# clresourcetype list -v
Resource Type      Node List
-----
SUNW.LogicalHostname <all>
SUNW.SharedAddress <all>
SUNW.nfs            phys-schost-1,phys-schost-2,phys-schost-3
SUNW.apache         phys-schost-1,phys-schost-2,phys-schost-3
```

在从全局群集节点执行以下命令时，该命令会列出区域群集 ZC 中所有已注册的资源类型。

```
# clresourcetype list -Z ZC
SUNW.nfs
SUNW.apache
```

例 165 列出指定节点上的资源类型

本示例列出已在 phys-schost-4 上注册的所有资源类型。

```
# clrt list -n phys-schost-4
SUNW.LogicalHostname
SUNW.SharedAddress
```

有关以下属性的描述，请参见 [Unresolved link to " attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving（发展中）

[Intro\(1CL\) \[17\]](#), [clreslogicalhostname\(1CL \) \[207\]](#), [clresource\(1CL\) \[225\]](#),
[clresourcegroup\(1CL\) \[253\]](#), [clressharedaddress\(1CL\) \[289\]](#), [cluster\(1CL\) \[461\]](#),
[scha_calls\(3HA\) \[897\]](#), [clconfiguration\(5CL\) \[1245\]](#), [r_properties\(5\) \[1103\]](#),
[Unresolved link to " attributes5"](#), [Unresolved link to " rbac5"](#)

[Unresolved link to " Oracle Solaris Cluster 数据服务规划和管理指南 中的资源组属性"](#)

超级用户可以运行此命令的所有格式。

任何用户都可以运行带有以下选项的此命令：

- -? 选项
- -v 选项

要运行带有子命令的此命令，超级用户以外的用户需要具有 RBAC 授权。请参见下表。

子命令	RBAC 授权
add-node	solaris.cluster.modify
export	solaris.cluster.read
list	solaris.cluster.read
list-props	solaris.cluster.read
set	solaris.cluster.modify
register	solaris.cluster.modify
remove-node	solaris.cluster.modify
show	solaris.cluster.read
unregister	solaris.cluster.admin

名称

clressharedaddress, clrssa — 管理 Oracle Solaris Cluster 共享地址的资源

```
/usr/cluster/bin/clressharedaddress [subcommand] -?

/usr/cluster/bin/clressharedaddress -V

/usr/cluster/bin/clressharedaddress [subcommand [options]] -v
[saresource]...

/usr/cluster/bin/clressharedaddress create -g resourcegroup [-h
  lhost[,...]] [-N netif@node[,...]] [-X node[,...]]
  [-p name=value] [-Z {zoneclustername | global}] [-d] saresource

/usr/cluster/bin/clressharedaddress create -i
  {- | clconfiguration} [-a] [-g resourcegroup[,...]] [-X
  node[,...]] [-p name=value] [-d] {+ | saresource...}

/usr/cluster/bin/clressharedaddress delete [-g resourcegroup[,...]]
  [-Z {zoneclustername | global}] [-F] {+ | saresource...}

/usr/cluster/bin/clressharedaddress disable [-g resourcegroup[,...]]
  [-R] [-n node[,...]] [-Z
  {zoneclustername | global}] {+ | saresource...}

/usr/cluster/bin/clressharedaddress enable [-g resourcegroup[,...]]
  [-R] [-n node[,...]] [-Z
  {zoneclustername | global}] {+ | saresource...}

/usr/cluster/bin/clressharedaddress export [-o {- | configfile}]
  {+ | saresource...}

/usr/cluster/bin/clressharedaddress list [-s state[,...]]
  [-g resourcegroup[,...]] [-Z {zoneclustername
  [,...] | global | all}] {+ | saresource...}

/usr/cluster/bin/clressharedaddress list-props [-l listtype] [-p
  name[,...]] [-Z {zoneclustername [,...] | global | all}]
  {+ | lresource...}

/usr/cluster/bin/clressharedaddress monitor [-g resourcegroup[,...]]
  [-Z {zoneclustername | global}] {+ | saresource...}

/usr/cluster/bin/clressharedaddress reset [-f errorflag] [-g
  resourcegroup[,...]] [-Z {zoneclustername | global}]
  {+ | saresource...}

/usr/cluster/bin/clressharedaddress set [-i {- | clconfiguration}]
  [-g resourcegroup[,...]] [-X node[,...]] [-p
  name[+|-]=value] [-Z {zoneclustername | global}] {+ | saresource...}

/usr/cluster/bin/clressharedaddress show [-g resourcegroup[,...]]
```

```
[-p name[,...]] [-Z {zoneclustername [,...] | global | all}]
[+ | saresource...]

/usr/cluster/bin/clressharedaddress status [-s state[,...]]
[ -n node[,...]] [-g resourcegroup[,...]]
[-Z {zoneclustername [,...] | global | all}] [+ | saresource...]

/usr/cluster/bin/clressharedaddress unmonitor [-g resourcegroup[,...]]
[-Z {zoneclustername | global}] {+ | saresource...}
```

`clressharedaddress` 命令用于管理 Oracle Solaris Cluster 共享地址的资源。`clrssa` 命令是 `clressharedaddress` 命令的简短格式。`clressharedaddress` 命令和 `clrssa` 命令是相同的。您可以使用此命令的任一种格式。

还可以使用 [clresource\(1CL\) \[225\]](#) 命令管理共享地址的资源。

`clressharedaddress` 命令的一些子命令可修改资源配置。可以从全局群集或区域群集使用这些子命令。以下子命令可修改资源配置：

- `disable`
- `enable`
- `monitor`
- `reset`
- `set`
- `unmonitor`

`clressharedaddress` 命令的某些子命令只获取有关资源的信息。

- `export`
- `list`
- `list-props`
- `show`
- `status`

为了避免此命令产生不可预测的结果，请从全局群集节点运行该命令的所有格式。

此命令的一般格式为：

```
clressharedaddress [subcommand] [options] [operands]
```

仅当 `options` 指定选项 `-?` 或 `-v` 时，才可以省略 `subcommand`。

此命令的每个选项都有长格式和简短格式两种格式。在本手册页的 `选项` 部分给出了每个选项的两种格式，以及对该选项的描述。

在区域群集中的操作

在区域群集中，可以将 `clressharedaddress` 命令与所有子命令（`export` 除外）一起使用。

也可以将 `-z` 选项与所有子命令（`export` 除外）一起使用来指定要将操作限制到的特定区域群集的名称。另外，还可以将区域群集名称附加到共享地址资源 (`zoneclustername` : `saresource`)，以便将操作限制到特定的区域群集。

您可以从全局群集节点访问所有区域群集信息，但无法从特定区域群集访问其他区域群集信息。如果不将操作限制到特定区域群集，则所用的子命令仅会在当前群集中运行。

支持以下子命令：

create

创建作为命令操作数指定的共享地址资源。

将 `create` 和 `-i` 选项一起使用以指定配置文件时，该子命令接受使用加号 (+) 作为操作数。使用 + 操作数时，将会创建配置文件中提供的尚不存在的所有资源。

使用 `create` 子命令之前，请确保 `/etc/netmasks` 文件具有所有逻辑主机名对应的 IP 地址子网和网络掩码条目。如有必要，请编辑 `/etc/netmasks` 文件以添加缺少的任何条目。

默认情况下，创建的资源处于启用状态，且启用了监视功能。但是，仅当资源的资源组联机后，该资源才会联机并受到监视。要在禁用状态下创建资源，请指定 `-d` 选项。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集在区域群集中创建共享地址资源，请使用 `-z` 选项指定区域群集的名称。

超级用户以外的用户需要具有 `solaris.cluster.modify` 基于角色的访问控制 (Role-Based Access Control, RBAC) 授权才能使用该子命令。

另请参见 `delete` 子命令的描述。

delete

删除作为命令操作数指定的共享地址资源。此子命令接受使用加号 (+) 作为操作数来指定删除所有资源。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集删除区域群集中的共享地址资源，请使用 `-z` 选项指定区域群集的名称。

`-g` 选项可以过滤操作数列表，以限制删除的资源。`-g` 选项只会删除操作数列表内属于 `resourcegrouplist` 中资源组的成员的资源。

- 默认情况下，仅当符合以下条件时，才会删除某个资源：
- 必须已禁用该资源。
- 必须已消除该资源的所有依赖性。

- 要确保删除所有指定的资源，请指定 -F 选项。-F 选项的效果如下所示：
- 删除所有指定的资源，即使是未禁用的资源。
- 从其他资源的资源依赖性设置中删除所有指定的资源。

将按照满足资源间依赖性所需的顺序禁用资源，而不管资源在命令行上的指定顺序。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

另请参见 `create` 子命令的描述。

`disable`

禁用作为命令操作数指定的共享地址资源。此子命令接受使用加号 (+) 作为操作数来指定禁用所有资源。

-g 选项可以过滤操作数列表，以限制禁用的资源。-g 选项只会禁用操作数列表内属于 `resourcegroup` 中资源组的成员的资源。

要确保满足所有必需的资源依赖性，请指定 -R 选项。-R 选项将会禁用依赖于指定为命令操作数的资源的任何资源，即使这些资源未指定为命令操作数。-g 选项和 -t 选项不适用于仅为满足资源依赖性而禁用的资源。

将按照满足资源间依赖性所需的顺序禁用资源，而不管资源在命令行上的指定顺序。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集禁用在区域群集中注册的共享地址资源，请使用 -z 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。

另请参见 `enable` 子命令的描述。

`enable`

启用作为命令操作数指定的共享地址资源。此子命令接受使用加号 (+) 作为操作数来指定启用所有资源。

-g 选项可以过滤操作数列表，以限制启用的资源。-g 选项仅启用操作数列表内属于 `resourcegroup` 中资源组的成员的资源。

要确保满足所有必需的资源依赖性，请指定 -R 选项。-R 选项将会启用依赖于指定为命令操作数的资源的任何资源，即使这些资源未指定为命令操作数。-g 选项不适用于仅为满足资源依赖性而启用的资源。

将按照满足资源间依赖性所需的顺序禁用资源，而不管资源在命令行上的指定顺序。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集启用在区域群集中注册的共享地址资源，请使用 -z 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。

另请参见 `disable` 子命令的描述。

export

以 `clconfiguration(5CL) [1245]` 手册页所述的格式导出共享地址资源配置。

只能在全局群集中使用该子命令。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

list

显示作为命令操作数指定的共享地址资源的列表。默认情况下，会显示所有资源。

`-g` 选项可以过滤操作数列表，以限制显示的资源。`-g` 选项只会显示操作数列表内属于 `resourcegroup` 中资源组的成员的资源。

此子命令接受使用加号 (+) 作为操作数来指定特定资源组中的所有资源，或者指定属于特定资源类型实例的所有资源。如果未提供操作数，则会显示指定资源组中的所有资源，或显示属于指定资源类型的实例的所有资源。

如果指定 `-v` 选项，还将显示列表中每个资源的资源组和资源类型。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集查看在区域群集中注册的共享地址资源，请使用 `-z` 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

list-props

显示作为命令操作数指定的共享地址资源的属性列表。默认情况下，会显示所有资源的扩展属性。

以下选项可以过滤操作数列表，以限制要显示其属性的资源：

`-g resourcegroup` 仅显示操作数列表中属于 `resourcegroup` 中资源组的共享地址资源的属性。

`-l` 选项指定要显示的资源属性的类型：

`-l all` 指定显示标准属性和扩展属性。

`-l extension` 指定只显示扩展属性。默认情况下，只显示扩展属性。

`-l standard` 指定只显示标准属性。

如果未指定 `-l` 选项，将只显示扩展属性，除非使用 `-p` 选项或 `-y` 选项明确指定标准属性。

-p 选项用于限制要显示的资源属性集。-p 选项只显示 *namelist* 中指定的属性。您可以在 *namelist* 中指定标准属性和扩展属性。

如果指定 -v 选项，还将显示每个属性的描述。

此子命令接受使用加号 (+) 作为操作数来指定特定资源组中的所有资源，或者指定属于特定资源类型实例的所有资源。如果未提供操作数，则会显示指定资源组中所有资源的属性，或显示属于指定资源类型的实例的所有资源的属性。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集查看区域群集中共享地址资源的属性列表，请使用 -z 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

monitor

打开对作为命令操作数指定的共享地址资源的监视。此子命令接受使用加号 (+) 作为操作数来指定开启对所有资源的监视。

-g 选项可以过滤操作数列表，以限制监视的资源。-g 选项只会监视操作数列表内属于 *resourcegroup* 中资源组的成员的资源。

开启对某个资源的监视后，仅当符合以下条件时，才会监视该资源：

- 已启用该资源。
- 包含该资源的资源组至少已在一个群集节点上联机。

注 - 开启对某个资源的监视不会启用该资源。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集监视区域群集中的资源，请使用 -z 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。

另请参见 `unmonitor` 子命令的描述。

reset

清除与作为命令操作数指定的共享地址资源相关联的错误标志。此子命令接受使用加号 (+) 作为操作数来指定清除所有资源的错误标志。

-g 选项可以过滤操作数列表，以限制重置的资源。-g 选项只会重置操作数列表内属于 *resourcegroup* 中资源组的成员的资源。

默认情况下，`reset` 子命令将会清除 `STOP_FAILED` 错误标志。要显式指定要清除的错误标志，请使用 -f 选项。-f 选项接受的唯一错误标志是 `STOP_FAILED` 错误标志。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集重置区域群集中的共享地址资源，请使用 -z 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。

set

修改作为命令操作数指定的共享地址资源的指定属性。此子命令接受使用加号 (+) 作为操作数来指定修改所有资源的特定属性。

-g 选项可以过滤操作数列表，以限制修改的资源。-g 选项只会修改操作数列表内属于 *resourcegroup* 中资源组的成员的资源。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集设置区域群集中共享地址资源的属性，请使用 -z 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

show

显示作为命令操作数指定的共享地址资源的配置。默认情况下，会显示所有资源的配置。

-g 选项可以过滤操作数列表，以限制要显示其配置的资源。-g 选项只会显示操作数列表内属于 *resourcegroup* 中资源组成员的资源的配置。

-p 选项用于限制要显示的资源属性集。-p 选项只显示 *namelist* 中指定的属性。您可以在 *namelist* 中指定标准属性和扩展属性。

此子命令接受使用加号 (+) 作为操作数来指定特定资源组中的所有资源，或者指定属于特定资源类型实例的所有资源。如果未提供操作数，则会显示指定资源组中所有资源的配置，或显示属于指定资源类型实例的所有资源的配置。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集查看区域群集中共享地址资源的配置，请使用 -z 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

status

显示作为命令操作数指定的共享地址资源的状态。默认情况下，会显示所有资源的状态。

以下选项可以过滤操作数列表，以限制要显示其状态的资源的列表：

- | | |
|-------------------------|--|
| -g <i>resourcegroup</i> | 仅显示操作数列表内属于 <i>resourcegroup</i> 中资源组成员的资源的状态。 |
| -n <i>nodelist</i> | 仅显示操作数列表内承载在 <i>nodelist</i> 中节点上的资源的状态。 |

`-s statelist` 仅显示操作数列表内处于 *statelist* 中的状况的资源的状态。

此子命令接受使用加号 (+) 作为操作数来指定特定资源组中的所有资源，或者指定属于特定资源类型实例的所有资源。如果未提供操作数，则会显示指定资源组中所有资源的状态，或显示属于指定资源类型实例的所有资源的状态。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集查看区域群集中共享地址资源的状态，请使用 `-z` 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

`unmonitor`

关闭对作为命令操作数指定的共享地址资源的监视。此子命令接受使用加号 (+) 作为操作数来指定关闭对所有资源的监视。

如果关闭对某个被禁用资源的监视，该资源不会受到影响。该资源及其监视器已脱机。

注 - 关闭对某个资源的监视不会禁用该资源。但是，当您禁用某个资源时，不需要关闭对该资源的监视。禁用的资源及其监视器将保持脱机状态。

`-g` 选项可以过滤操作数列表，以限制要对其关闭监视的资源。`-g` 选项将对操作数列表内属于 *resourcegroup* 中资源组的成员的资源关闭监视。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集关闭对区域群集中共享地址资源的监视，请使用 `-z` 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。

另请参见 `disable` 子命令和 `monitor` 子命令的描述。

支持以下选项：

`-?`

`--help`

显示帮助信息。使用该选项时，不会执行任何其他处理。

指定此选项时，子命令可有可无。

如果仅指定此选项而不使用子命令，将会显示此命令的子命令列表。

如果使用子命令且指定此选项，将会显示该子命令的用法选项。

此选项与特定子命令一起使用的效果如下所示：

`create`

与 `-g` 选项一起指定时，此选项将会显示指定资源组的所有资源属性的帮助信息。

`set` 显示指定为命令操作数的资源的属性的帮助信息。

`-a`

`--automatic`

当基于群集配置信息创建资源时，将自动执行以下附加操作：

- 注册资源类型
- 创建资源组
- 创建操作数列表中指定的资源所依赖的资源

群集配置信息包含的信息必须足以执行以下所有操作：

- 启用要注册的资源类型
- 启用要创建的资源组
- 启用要创建的资源

此选项仅可以与 `create` 子命令一同指定。如果指定此选项，还必须指定 `-i` 选项并提供配置文件。

`-d`

`--disable`

创建资源后禁用该资源。此选项仅可以与 `create` 子命令一同指定。默认情况下，资源在启用状态下创建。

启用某个资源并不保证该资源联机。仅当资源所在的资源组已在至少一个节点上联机后，该资源才会联机。

`-f errorflag`

`--flag errorflag`

显式指定要通过 `reset` 子命令清除的错误标志。此选项仅可以与 `reset` 子命令一同指定。默认情况下，`reset` 子命令将会清除 `STOP_FAILED` 错误标志。

- f 选项接受的唯一错误标志是 `STOP_FAILED` 错误标志。

`-F`

`--force`

强制删除未禁用的资源。此选项仅可以与 `delete` 子命令一同指定。

`-g resourcegroup[,...]`

`--resourcegroup resourcegroup[,...]`

指定一个资源组或资源组列表。

对于除 `create` 以外的子命令，该命令仅对操作数列表内属于 `-g` 选项指定的资源组成员的资源起作用。

此选项与特定子命令一起使用的效果如下所示：

`create`

指定在特定资源组中创建资源。将 `-g` 与 `create` 子命令一起使用时，只能指定一个资源组。

`-h /host[,...]`

`--logicalhost /host[,...]`

指定主机名列表。无论是在多个逻辑主机需要与新的 SharedAddress 资源相关联时，还是在逻辑主机与资源本身具有不同的名称时，都必须使用 `-h` 选项。SharedAddress 资源的 HostnameList 内的所有逻辑主机必须位于同一子网上。如果未指定 HostnameList 属性，HostnameList 将与 SharedAddress 资源相同。

SharedAddress 资源的逻辑主机名必须位于同一子网上。

您可以使用 `-h` 选项，而不是通过 `-p` 设置 HostnameList 属性；但是，无法在同一命令中使用 `-h` 并明确设置 HostnameList。

`-h` 选项只能与 `create` 子命令一起使用。

`-i {- | clconfiguration}`

`--input {- | clconfiguration}`

指定用于创建或修改共享地址资源的配置信息。此信息必须符合 [clconfiguration\(5CL\) \[1245\]](#) 手册页中定义的格式。此信息可包含在文件中或者通过标准输入提供。要指定标准输入，请指定 `-` 而不是文件名。

仅会创建或修改作为命令操作数提供的资源。在命令中指定的选项将会覆盖配置信息中设置的任何选项。如果配置信息中缺少配置参数，则必须在命令行中指定这些参数。

此选项与特定子命令一起使用的效果如下所示：

<code>create</code>	与 <code>-a</code> 选项一起指定时，此选项将注册所有必需的资源类型，并创建所有必需的资源组。您必须提供注册和配置所需的全部信息。所有其他配置数据将被忽略。
---------------------	---

`-l listtype`

`--listtype listtype`

指定要通过 `list-props` 子命令显示的资源属性的类型。此选项仅可以与 `list-props` 子命令一同指定。

必须根据以下列表为 `listtype` 指定一个值：

<code>all</code>	指定显示标准属性和扩展属性。
------------------	----------------

<code>extension</code>	指定只显示扩展属性。默认情况下，只显示扩展属性。
------------------------	--------------------------

<code>standard</code>	指定只显示标准属性。
-----------------------	------------

如果未指定 `-l` 选项，则只显示扩展属性，除非使用 `-p` 选项明确指定标准属性。

`-n node[...]`

`--node node[...]`

指定目标全局群集或区域群集中的一个节点或一系列节点。可以使用节点名称或节点 ID 指定每个节点。

如果指定 `-z` 选项，则仅可以使用 `-n` 选项指定区域群集主机名，而不能指定全局群集主机名。如果没有指定 `-z` 选项，则仅可以使用 `-n` 选项指定全局群集主机名。

可以与此选项一起指定的子命令如下所述：

<code>disable</code>	只禁用操作数列表中承载在指定节点上的资源。
<code>enable</code>	只启用操作数列表中承载在指定节点上的资源。
<code>status</code>	仅报告操作数列表中承载在指定节点上的资源的状态。

`-N netif@node[...]`

`--netiflist netif@node[...]`

指定一个资源属性。通过 `-N` 选项，可以设置 `NetIfList` 属性，而不必使用 `-p` 选项来指定该属性。如果不指定 `-N`，`clressharedaddress` 命令将尝试基于可用 IPMP 组或公共适配器以及与 `HostnameList` 属性关联的子网来设置 `NetIfList` 属性。

可以采用 `ipmpgroup@node[...]` 格式指定 `NetIfList` 属性。但是，`-N` 接受 `ipmpgroup@ node[...]` 和 `publicNIC@ node[...]` 两种格式。如果未使用 `-N`，或者将该选项与 `publicNIC@node` 一起使用，则 `clressharedaddress` 命令将尝试创建必要的 IPMP 组。系统将创建一组包含一个或多个单适配器的 IPMP 组，以及一组默认以后使用标准 Oracle Solaris 接口修改为包括多个适配器的 IPMP 组。

您可以使用 `-N` 选项，而不是通过 `-p` 直接设置 `NetIfList` 属性；但是，无法在同一命令中使用 `-N` 并明确设置 `NetIfList`。

`-N` 只能与 `create` 子命令一起使用。

`-o { - | clconfiguration }`

`--output { - | clconfiguration }`

指定要将资源配置信息写入到的位置。此位置可以是某个文件或标准输出。要指定标准输出，请指定 `-` 而不是文件名。如果指定标准输出，将抑制命令的所有其他标准输出。此选项仅可以与 `export` 子命令一同指定。

只会写入作为命令操作数提供的资源的配置信息。信息是采用 [clconfiguration\(5CL\) \[1245\]](#) 手册页中定义的格式写入的。

```
-p name=value
-p name+=array-values
-p name-=array-values
--property name=value
--property name+=array-values
--property name-=array-values
```

设置资源的标准属性和扩展属性。此选项仅可以与 `create` 子命令和 `set` 子命令一同指定。

有关标准属性的描述，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

有关某个资源类型的扩展属性的描述，请参见该资源类型的文档。

与此选项一起使用的运算符如下所述：

=	将属性设置为指定值。 <code>create</code> 子命令和 <code>set</code> 子命令接受此运算符。
+=	将一个或多个值添加到某个字符串数组值。只有 <code>set</code> 子命令接受此运算符。只能为字符串数组值指定此运算符。
-=	从某个字符串数组值中删除一个或多个值。只有 <code>set</code> 子命令接受此运算符。只能为字符串数组值指定此运算符。

如果只需在群集节点的子集上设置每节点属性，请通过在属性名称后面附加节点列表（用大括号括住）来指定要设置该属性的节点，如下所示：

```
name{nodelist}
```

`nodelist` 是以逗号分隔的节点名称或节点 ID 的列表。有关每节点属性的更多信息，请参见 [rt_properties\(5\) \[1143\]](#) 手册页。

```
-p name[,...]
--property name[,...]
```

为 `list-props` 子命令和 `show` 子命令指定属性列表。

可将此选项用于资源的标准属性和扩展属性。

有关标准属性的描述，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

有关某个资源类型的扩展属性的描述，请参见该资源类型的文档。

如果未指定此选项，`list-props` 子命令和 `show` 子命令将会列出所有资源属性或大多数资源属性，具体取决于是否同时指定了 `-v` 选项。

```
-R
--recursive
```

以递归方式启用或禁用资源，以确保满足所有必需的依赖性。此选项仅可以与 `disable` 子命令和 `enable` 子命令一同指定。

此选项与这些子命令一起使用的效果如下所示：

disable	禁用依赖于指定为命令操作数的资源的任何资源，即使这些资源未指定为命令操作数。
enable	启用指定为命令操作数的资源所依赖的任何资源，即使这些资源未指定为命令操作数。

-s *state*[,...]

--state *state*[,...]

为 `list` 子命令和 `status` 子命令指定状况列表。

此选项将会限制输出，以便只包括在节点列表中的一个或多个节点上处于指定状况之一的资源。

可能的状况如下：

- degraded
- detached
- faulted
- monitor_failed
- not_online – 指定除 `online` 或 `online_not_monitored` 以外的任何状况
- offline
- online
- online_not_monitored
- start_failed
- stop_failed
- unknown
- unmonitored
- wait

-V

--version

显示该命令的版本。

指定此选项时，不要将其与子命令、操作数或其他选项结合使用。子命令、操作数或其他选项将被忽略。-V 选项只显示命令的版本。不会执行任何其他操作。

-v

--verbose

将详细消息显示到标准输出。

可以在任何格式的此命令中指定该选项。

请不要同时指定 -v 选项和 -o - 选项。否则 -v 选项将被忽略。-o - 选项会抑制其他所有标准输出。

-X *node*[...]

--auxnode *node*[...]

设置 AuxNodeListSharedAddress 资源属性。

AuxNodeList 列表中的节点可以托管与共享地址资源关联的逻辑主机集。但是，这些节点无法在故障转移期间充当主节点。

-Z {*zoneclustername* | global | all}

--zoneclustername={*zoneclustername* | global | all}

--zoneclustername {*zoneclustername* | global | all}

指定一个或多个存在资源且您要对其执行操作的群集。

除 export 子命令以外的所有子命令均支持此选项。

如果指定此选项，还必须指定以下列表中的参数之一：

zoneclustername 指定使用此选项的命令仅对名为 *zoneclustername* 的区域群集中的所有指定资源起作用。

global 指定使用此选项的命令仅对全局群集中的所有指定资源进行操作。

all 如果在全局群集中使用此参数，则会指定使用此参数的命令对所有群集（包括全局群集和所有区域群集）中的所有指定资源起作用。

如果在某区域群集中使用此参数，则会指定使用此参数的命令仅对该区域群集中的所有指定资源起作用。

支持以下操作数：

resource 指定应接受 Oracle Solaris Cluster 资源名称作为操作数。如果子命令接受多个资源，则可以使用加号 (+) 指定所有共享地址资源。

如果命令对于所有指定的操作数均成功执行，则返回零 (CL_NOERR)。如果某个操作数出错，命令会继续处理操作数列表中的下一个操作数。返回的退出代码始终反映最先发生的错误。

此命令返回下列退出状态代码：

0 CL_NOERR

没有错误

您发出的命令已成功完成。

1 CL_ENOMEM

交换空间不足

某个群集节点耗尽了交换内存或其他操作系统资源。

3 CL_EINVAL

参数无效

您键入的命令不正确，或者使用 `-i` 选项提供的群集配置信息语法不正确。

6 CL_EACCESS

权限被拒绝

指定的对象不可访问。要发出命令，可能需要拥有超级用户或 RBAC 访问权限。有关更多信息，请参见 [Unresolved link to "su1M"](#) 和 [Unresolved link to "rbac5"](#) 手册页。

9 CL_ESTATE

对象处于错误状况

您尝试修改一个在特定时间无法修改，或任何时候均无法修改的属性、资源组或其他对象。

10 CL_EMETHOD

资源方法失败

资源方法失败。方法因以下某个原因而失败：

- 尝试创建资源或修改资源属性时，`validate` 方法失败。
- 尝试启用、禁用或删除资源时，`validate` 以外的其他方法失败。

15 CL_EPROP

属性无效

您使用 `-p`、`-y` 或 `-x` 选项指定的属性或值不存在或不被允许。

35 CL_EIO

I/O 错误

发生了物理输入/输出错误。

36 CL_ENOENT

没有这样的对象

由于以下某个原因，找不到您所指定的对象：

- 该对象不存在。
- 您尝试使用 `-o` 选项创建的配置文件路径中的某个目录不存在。
- 您尝试使用 `-i` 选项访问的配置文件包含错误。

39 CL_EEXIST

对象已存在

您所指定的设备、设备组、群集互连组件、节点、群集、资源、资源类型、资源组或专用字符串已存在。

这些退出值与 [scha_calls\(3HA\) \[897\]](#) 手册页中描述的返回码兼容。

例 166 创建共享地址资源

此命令将在名为 `rg-failover` 的资源组内创建名为 `sharedhost1` 的资源。创建的资源处于启用状况，并启用了监视功能。

```
# clressharedaddress create -g rg-failover sharedhost1
```

以下两个命令均会在区域群集 `ZC` 中创建名为 `sharedhost1` 的资源。可在全局群集节点中或区域群集 `ZC` 内执行这些命令。

```
# clressharedaddress create -g rg-failover -Z ZC sharedhost1
```

```
# clressharedaddress create -g rg-failover ZC:sharedhost1
```

例 167 使用其他逻辑主机名创建共享地址资源

此命令将在名为 `rg-failover` 的资源组内创建名为 `rs-sharedhost1` 的资源。

该逻辑主机名与资源名称不同，但逻辑主机的名称和 IP 地址保持不变。

```
# clressharedaddress create -g rg-failover \  
-h sharedhost1 rs-sharedhost1
```

例 168 指定共享地址资源的 IPMP 组

此命令可设置 `sharedhost1` 资源的 IPMP 组。

```
# clressharedaddress create -g rg-failover \  
-N ipmp0@black,ipmp0@white sharedhost1
```

例 169 删除共享地址资源

此命令可删除名为 `sharedhost1` 的资源。

```
# clressharedaddress delete sharedhost1
```

例 170 列出共享地址资源

此命令可列出所有共享地址资源。

```
# clressharedaddress list  
sharedhost1  
sharedhost2
```


例 171 列出共享地址资源及其资源组和资源类型

此命令可列出所有共享地址资源及其资源组和资源类型。

```
# clressharedaddress list -v
Resources  Resource Groups Resource Types
-----
sharedhost1 rg-failover-1 SUNW.SharedAddress
sharedhost2 rg-failover-2 SUNW.SharedAddress
```

例 172 列出共享地址资源的扩展属性

此命令可列出所有共享地址资源的扩展属性。

```
# clressharedaddress list-props -v
Properties      Descriptions
-----
NetIfList      List of IPMP groups on each node
AuxNodeList    List of nodes on which this resource is available
HostnameList   List of hostnames this resource manages
CheckNameService Name service check flag
```

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

[Intro\(1CL\) \[17\]](#), [cluster\(1CL\) \[461\]](#), [clresource\(1CL\) \[225\]](#),
[clreslogicalhostname\(1CL\) \[207\]](#), [clresourcegroup\(1CL\) \[253\]](#),
[clresourcetype\(1CL\) \[277\]](#), [scha_calls\(3HA\) \[897\]](#), [clconfiguration\(5CL\) \[1245\]](#),
[Unresolved link to "rbac5"](#), [r_properties\(5\) \[1103\]](#)

超级用户可以运行此命令的所有格式。

任何用户都可以运行带有以下选项的此命令：

- -? 选项
- -v 选项

要运行带有子命令的此命令，超级用户以外的用户需要具有 RBAC 授权。请参见下表。

子命令	RBAC 授权
create	solaris.cluster.modify

子命令	RBAC 授权
delete	solaris.cluster.modify
disable	solaris.cluster.admin
enable	solaris.cluster.admin
export	solaris.cluster.read
list	solaris.cluster.read
list-props	solaris.cluster.read
monitor	solaris.cluster.admin
reset	solaris.cluster.admin
set	solaris.cluster.modify
show	solaris.cluster.read
status	solaris.cluster.read
unmonitor	solaris.cluster.admin

名称

clresourcegroup, clrg — 管理 Oracle Solaris Cluster 数据服务的资源组

```
/usr/cluster/bin/clresourcegroup -V
/usr/cluster/bin/clresourcegroup [subcommand] -?
/usr/cluster/bin/clresourcegroup subcommand [options] -v [resourcegroup ...]
/usr/cluster/bin/clresourcegroup add-node -n node[,...] [-S] [-Z {zoneclustername | global}]
{+ | resourcegroup...}
/usr/cluster/bin/clresourcegroup create [-S] [-n node[,...]] [-p name=value] [...] [-Z
{zoneclustername | global}] {+ | resourcegroup...}
/usr/cluster/bin/clresourcegroup create -i {- | clconfigfile} [-S] [-n node [,...]] [-p
name=value] [...] {+ | resourcegroup...}
/usr/cluster/bin/clresourcegroup delete [-F] [-Z {zoneclustername | global}] {+
| resourcegroup...}
/usr/cluster/bin/clresourcegroup evacuate -n node[,...] [-T seconds] [-Z {zoneclustername
| global}] {+}
/usr/cluster/bin/clresourcegroup export [-o {- | configfile}] {+ | resourcegroup...}
/usr/cluster/bin/clresourcegroup list [-n node[,...]] [-r resource[,...]] [-s state[,...]] [-t
resourcetype[,...]] [ [-Z {zoneclustername[,...] | global | all}] {+ | resourcegroup...}
/usr/cluster/bin/clresourcegroup manage [-Z {zoneclustername | global}] {+
| resourcegroup...}
/usr/cluster/bin/clresourcegroup offline [-n node [,...]] ] [-Z {zoneclustername | global}]
{+ | resourcegroup...}
/usr/cluster/bin/clresourcegroup online [-e] [-m] [-M] [-n node [,...]] [-Z {zoneclustername
| global}] {+ | resourcegroup...}
/usr/cluster/bin/clresourcegroup quiesce [-k] [-Z {zoneclustername | global}] {+
| resourcegroup...}
/usr/cluster/bin/clresourcegroup remaster [-Z {zoneclustername |
global}] {+ | resourcegroup...}
/usr/cluster/bin/clresourcegroup remove-node -n node
[,...] [-Z {zoneclustername | global}]
{+ | resourcegroup...}
/usr/cluster/bin/clresourcegroup restart [-n node[,...]]
[-Z zoneclustername |global] {+ | resourcegroup...}
/usr/cluster/bin/clresourcegroup resume [-Z {zoneclustername |
global}] {+ | resourcegroup...}
/usr/cluster/bin/clresourcegroup set [-n node[,...]] -p name[+|-]=value [...]
[-Z {zoneclustername | global}] {+ | resourcegroup...}
```

```
/usr/cluster/bin/clresourcegroup show [-n node[,...]]
      [-p name[,...]] [-r resource[,...]] [-t resourcetype[,...]]
      [-Z {zoneclustername[,...] | global | all}]
      [+ | resourcegroup...]

/usr/cluster/bin/clresourcegroup status [-n node[,...]]
      [-r resource [,]...] [-s state [,]...] [-t resourcetype
      [,]...] [-Z {zoneclustername[,...] | global | all}]
      [+ | resourcegroup...]

/usr/cluster/bin/clresourcegroup suspend [-k] [-Z
      {zoneclustername | global}] {+ | resourcegroup...}

/usr/cluster/bin/clresourcegroup switch -n node[,...]
      [-e] [-m] [-M] [-Z {zoneclustername | global}]
      {+ | resourcegroup...}

/usr/cluster/bin/clresourcegroup unmanage [-Z {zoneclustername |
      global}] {+ | resourcegroup...}
```

该命令管理 Oracle Solaris Cluster 数据服务资源组。

仅当 *options* 是 *-?* 选项或 *-v* 选项时，才可以省略 *subcommand*。

每个选项都有长和短两种格式。在 *选项* 中给出每个选项的两种格式以及该选项的描述。

clrg 命令是 *clresourcegroup* 命令的简短格式。

子命令需要至少一个操作数，但 *list*、*show* 和 *status* 除外。但是，很多子命令都支持加号操作数 (+)。该操作数将子命令应用于所有适用的对象。

可以在区域群集中使用该命令的某些格式。有关有效使用该命令的更多信息，请参见各个子命令的描述。为便于管理，请从全局群集节点使用此命令。

资源和资源组

资源状况、资源组状况和资源状态均是基于每个节点进行维护的。例如，一个给定资源在每个群集节点上可以具有不同的状况和状态。

注 - 状况名称 (例如 *Offline* 和 *Start_failed*) 不区分大小写。指定状况名称时，可以使用大小写字母的任意组合。

资源状态是由资源组管理器 (Resource Group Manager, RGM) 仅根据对资源调用的方法在每个节点上设置的。例如，在给定节点上对某个资源成功运行 *STOP* 方法后，该资源在此节点上的状态将为 *Offline*。如果 *STOP* 方法返回非零值或超时，则该资源的状态为 *Stop_failed*。

可能的资源状态包括：*Online*、*Offline*、*Start_failed*、*Stop_failed*、*Monitor_failed*、*Online_not_monitored*、*Starting* 和 *Stopping*。

可能的资源组状态包括：Unmanaged、Online、Offline、Pending_online、Pending_offline、Error_stop_failed、Online_faulted 和 Pending_online_blocked。

除资源状况外，RGM 还可以维护资源状态，资源状态可由资源本身使用 API 设置。Status Message 字段实际上包括两个组成部分：状态关键字和状态消息。状态消息由资源有选择性地设置，它是在状态关键字后面输出的任意文本字符串。

资源状态可能值的描述如下：

Degraded	资源已联机，但其性能或可用性可能在某方面出现了下降。
Faulted	资源遇到错误，无法正常运行。
Offline	资源已脱机。
Online	资源已联机，正在提供服务。
Unknown	当前状态是未知或正在转换。

在区域群集中使用此命令

在区域群集中，可以将 `clresourcegroup` 命令与所有子命令（`export` 除外）一起使用。

也可以将 `-z` 选项与所有子命令（`export` 除外）一起使用来指定要将操作限制到的特定区域群集的名称。另外，还可以将区域群集名称附加到资源组 (`zoneclustername :resourcegroup`)，以便将操作限定到特定的区域群集。

您可以从全局群集节点访问所有区域群集信息，但无法从特定区域群集访问其他区域群集信息。如果不将操作限制到特定区域群集，则所用的子命令仅会在当前群集中运行。

您可以指定某区域群集中的资源组与另一个区域群集中的资源组（或全局群集中的资源组）之间的关联性。可以使用以下命令指定不同区域群集中的资源组之间的关联性：

```
# clresourcegroup set -p RG_affinities={+|++|-|--}
target-zc:target-rg
source-zc:source-rg
```

关联性类型可以是以下类型之一：

- + (弱正)
- ++ (强正)
- +++ (包含故障转移委托的强正)
- - (弱负)
- -- (强负)

例如，如果需要指定区域群集 ZC1 中的资源组 RG1 与区域群集 ZC2 中的资源组 RG2 之间的强正关联性 (++)，请使用以下命令：

```
# clresourcegroup set -p RG_affinities=++ZC2:RG2 ZC1:RG1
```

要指定区域群集 ZC1 中的资源组 RG1 和区域群集 ZC2 中的资源组 RG2 之间包含故障转移委托的强正关联性 (+++), 请使用以下命令 :

```
# clresourcegroup set -p RG_affinities=+++ZC2:RG2 ZC1:RG1
```

要指定区域群集 ZC1 中的资源组 RG1 和全局群集中的资源组 RG2 之间的强负关联性 (--), 请使用以下命令 :

```
# clresourcegroup set -p RG_affinities=-global:RG2 ZC1:RG1
```

资源组可以跨群集节点或区域自动分布。有关更多信息, 请参见 [rg_properties\(5\) \[1129\]](#) 手册页中的 Load_factors、Priority 和 Preemption_mode 项。

支持以下子命令 :

add-node

将节点添加到某个资源组的 Nodelist 属性的末尾。

您可以在全局群集中或区域群集中使用该子命令。

列表中节点和区域的顺序指定了在这些节点或区域上使资源组联机的首选顺序。要将节点添加到 Nodelist 属性中的其他位置, 请使用 set 子命令。

要从全局群集节点为特定区域群集中的资源组添加节点, 您可以使用 -z 选项指定该区域群集的名称。

超级用户以外的用户需要具有 solaris.cluster.modify 基于角色的访问控制 (Role-Based Access Control, RBAC) 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

create

创建新的资源组。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集节点在特定区域群集中创建资源组, 您可以使用 -z 选项指定该区域群集的名称。

如果您使用 -i 选项指定了一个配置文件, 则可以指定加号操作数 (+)。该操作数指定您要创建该文件中提供的尚不存在的所有资源。

要设置新资源组的 Nodelist 属性, 请指定以下选项之一 :

- -n *node*
- -p Nodelist=-*node-*][,...]
- -i *clconfigfile*

列表中节点的顺序指定了在这些节点上使资源组联机的首选顺序。如果在创建时没有指定节点列表, 则 Nodelist 属性会设置为在群集中配置的所有节点。顺序是任意的。

默认情况下，创建资源组时 `RG_mode` 属性已设置为 `Failover`。但是，通过使用 `-s` 选项或 `-p RG_mode=Scalable` 选项，或将 `Maximum primaries` 设置为大于 1 的值，可以创建可伸缩的资源组。仅当创建资源组后才可以设置该资源组的 `RG_mode` 属性。

创建资源组时始终会将资源组置于不受管理状态。但在您发出 `manage` 子命令或使用 `-M` 选项发出 `online` 或 `switch` 子命令时，RGM 会将其状态更改为受管理状态。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

delete

删除资源组。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集节点删除特定区域群集中的资源组，您可以使用 `-z` 选项指定该区域群集的名称。

可以随该子命令指定加号操作数 (+) 以删除所有资源组。

如果资源组包含资源，除非指定了 `-F` 选项，否则无法删除这些资源组。如果指定了 `-F` 选项，则每个组及其中的所有资源都会被删除。还会同时删除所有依赖性和关联性。

该子命令将按照反映资源和资源组依赖性的顺序删除多个资源组。在命令行中指定资源组的顺序无关紧要。

`clresourcegroup delete` 命令的以下格式是通过几个步骤完成的：

- 同时删除多个资源组时
- 使用 `-F` 选项删除某个资源组时

如果以上任一格式的命令被中断（例如，节点出现故障），则某些资源组的配置可能会处于无效状态。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

evacuate

使通过 `-n` 选项指定的节点上的所有资源组脱机。

您可以在全局群集中或区域群集中使用该子命令。

从全局群集节点运行 `evacuate` 命令时，该子命令会清除该全局群集或区域群集中的所有资源组。在区域群集中，该子命令仅清除指定区域群集中的资源组。要从全局群集节点清除特定区域群集中的资源组，您可以使用 `-z` 选项指定该区域群集的名称。

资源组会按照反映资源和资源组依赖性的顺序进行脱机。

您可以将 `-t` 选项与该子命令一起使用，以指定用于阻止资源组切换回的秒数。如果未指定值，将使用默认值 60 秒。

在清除完成之后的 60 秒或指定的秒数内，防止资源组在清除的节点上进行故障转移或自动联机。

但是，如果您使用 `switch` 或 `online` 子命令使资源组联机，或清除的节点重新引导，则清除计时器将立即过期并再次允许自动故障转移。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

export

将资源组的配置信息写入一个文件或标准输出 (stdout)。

只能在全局群集中使用该子命令。

[clconfiguration\(5CL\) \[1245\]](#) 手册页中描述了该配置信息的格式。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

list

显示按您指定的限定符选项过滤的资源组的列表。

您可以在全局群集中或区域群集中使用该子命令。

可以使用 `-r resource` 来仅包括含有资源的资源组。可以使用 `-t resourcetype` 来仅包括含有 `resourcetype` 中某个资源类型的资源组。可以使用 `-n node` 来仅包括在一个或多个节点上处于联机状态的资源组。

如果指定了 `-s state`，则仅会列出具有指定状态的组。

如果未指定操作数或指定了加号操作数 (+)，则会列出按您指定的任何限定符选项过滤的所有资源组。

如果指定了详细选项 `-v`，则会显示状态（资源组是处于联机还是脱机状态）。即使资源组只在群集中的一个节点上处于联机状态，列出该资源组时该资源组也会显示为联机状态。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

manage

将您所指定的资源组置于被管理状态。

您可以在全局群集中或区域群集中使用该子命令。

如果从全局群集节点使用该子命令，则该子命令可以对任何资源组进行操作。如果在区域群集中使用该子命令，则该子命令只能对区域群集中的资源组成功进行操作。要从全局群集节点管理特定区域群集中的资源组，可以使用 `-z` 选项指定该区域群集的名称。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

offline

将您所指定的资源组置于脱机状态。

您可以在全局群集中或区域群集中使用该子命令。

如果从全局群集节点使用该子命令，则该子命令可以对任何资源组进行操作。如果在区域群集中使用该子命令，则该子命令只能对区域群集中的资源组成功进行操作。要从全局群集节点使特定区域群集中的资源组脱机，可以使用 `-z` 选项指定该区域群集的名称。

如果指定了 `-n` 选项，则只会使资源组在指定的节点上脱机。

如果未指定 `-n` 选项，则会使资源组在所有节点上脱机。

如果使用 `offline` 子命令使某资源组脱机，则重新引导节点后该资源组的 `Offline` 状态不会保留。换句话说，如果节点停止或加入群集，则资源组可以在某个节点中进入联机状态，即使您先前已将资源组切换到脱机状态。即使所有资源都被禁用，该资源组还是会进入联机状态。

同样，声明任何 `RG_dependencies` 或强 `RG_affinities` 的资源组可以在切换到其他资源组时自动进入联机状态。

要防止资源组自动进入联机状态，请使用 `suspend` 子命令暂停该资源组的自动恢复操作。要继续自动恢复操作，请使用 `resume` 子命令。

资源组会按照反映资源和资源组依赖性的顺序进行脱机。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5" 手册页](#)。

online

将您所指定的资源组置于联机状态。

您可以在全局群集中或区域群集中使用该子命令。

如果从全局群集节点使用该子命令，则该子命令可以对任何资源组进行操作。如果在区域群集中使用该子命令，则该子命令只能对区域群集中的资源组成功进行操作。要从全局群集节点使特定区域群集中的资源组联机，可以使用 `-z` 选项指定该区域群集的名称。

使用 `-n` 选项指定要使资源组在其上联机的节点的列表。如果未指定 `-n` 选项，则该子命令会使资源组在它们最偏爱的节点上联机，且不使这些组从其当前的任何主节点脱机。每个资源组的联机节点总数受 `Desired primaries` 和 `Maximum primaries` 属性限制。节点的首选排序由 `Nodelist`、`RG_affinities` 和 `Load_factors` 属性决定。有关这些属性的更多信息，请参见 [rg_properties\(5\) \[1129\]](#) 手册页。

当命令行中提供了多个资源组操作数并且未指定 `-n` 选项时，系统会按照由 `Priority` 属性确定的顺序为这些资源组操作数分配主节点，其中优先级最高的资源组会首先获得节点分配。分配完主节点后，所有资源组操作数会并行脱机，除非受到资源依赖性 or 资源组依赖性的约束。在命令行中指定资源组的顺序无关紧要。有关 `Priority` 属性的更多信息，请参见 [rg_properties\(5\) \[1129\]](#) 手册页。

优先级较低的资源组可能无法分配到其最偏爱的节点，或者当超出负载限制时可能受优先级较高的资源组影响而强制脱机。有关更多信息，请参见 [clnode\(1CL\) \[149\]](#) 手册页中的 `loadlimit` 子命令。

与 `switch` 子命令不同，该子命令不会尝试将 `Nodelist` 属性中列出的任何节点切换到 `Offline` 状态。

如果随该子命令指定了 `-e` 选项，则会启用资源组集中已联机的所有资源。

可以指定 `-m` 选项来监视处于联机状态的资源组集中的所有资源。但是，除非资源已启用并与某个 `MONITOR_START` 方法相关联，否则系统不会真正监视它们。

您还可以指定 `-M` 选项以指示将处于联机状态的所有资源组置于受管理状态。如果未指定 `-M` 选项，则该子命令对不受管理的资源组没有任何影响。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

quiesce

将您所指定的资源组置于停止状态。

您可以在全局群集中或区域群集中使用该子命令。

如果从全局群集节点使用该子命令，则该子命令可以对任何资源组进行操作。如果在区域群集中使用该子命令，则该子命令只能对区域群集中的资源组成功进行操作。要从全局群集节点对特定区域群集中的资源组进行操作，您可以使用 `-z` 选项指定该区域群集的名称。

如果 `START` 或 `STOP` 方法失败，该命令将阻止资源组从一个节点连续切换到另一个节点。如果停止方法失败并将资源的 `Failover_mode` 属性设置为 `HARD`，它还将阻止节点重新引导（通常情况下会发生重新引导）。在这种情况下，资源将改为 `STOP_FAILED` 状态。

使用 `-k` 选项可中止代表受影响资源组中的资源运行的方法。如果未指定 `-k` 选项，系统会允许方法继续运行，直到这些方法退出或超过所配置的超时时间。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

remaster

将指定的资源组从它们当前的主节点切换到它们最偏爱的节点。每个资源组的联机节点总数受 `Desired primaries` 和 `Maximum primaries` 属性限制。节点的首选排序由 `Nodelist`、`RG_affinities` 和 `Load_factors` 属性决定。有关更多信息，请参见 [clnode\(1CL\) \[149\]](#) 和 [rg_properties\(5\) \[1129\]](#) 手册页。

您可以在全局群集中或区域群集中使用该子命令。

如果从全局群集节点使用该子命令，则该子命令可以对任何资源组进行操作。如果在区域群集中使用该子命令，则该子命令只能对区域群集中的资源组成功进行操作。要从全局群集节点对特定区域群集中的资源组进行操作，可以使用 `-z` 选项指定该区域群集的名称。

与 `online` 子命令不同，该子命令可以使资源组在其当前主节点上脱机，并使其在更优先的主节点上联机。

当命令行中提供了多个资源组操作数时，系统会按照由 `Priority` 属性确定的顺序为这些资源组操作数分配主节点，其中优先级最高的资源组会先获得节点分配。在命令行中指定资源组的顺序无关紧要。有关更多信息，请参见 [rg_properties\(5\) \[1129\]](#) 手册页。

优先级较低的资源组可能无法分配到其最偏爱的节点，或者当超出负载限制时可能受优先级较高的资源组影响而强制脱机。有关更多信息，请参见 [clnode\(1CL\) \[149\]](#) 手册页的 `loadlimit` 子命令。

该子命令对不受管理的资源组不会产生任何影响。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

remove-node

从资源组的 `Nodelist` 属性中删除节点。

您可以在全局群集中或区域群集中使用该子命令。

您可以从全局群集节点或区域群集使用该子命令。要从全局群集节点为某区域群集中的某资源组删除节点，您可以使用 `-z` 选项指定该区域群集的名称。

在删除节点后，`remove-node` 可能会将 `Maximum primaries` 或 `Desired primaries` 属性的值重置为 `Nodelist` 属性中的新节点数。仅当任意一个值超过 `Nodelist` 属性中的新节点数时，`remove-node` 才会重置 `Maximum primaries` 或 `Desired primaries` 属性的值。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

restart

使某个资源组在当前承载该资源组的一组主节点上脱机后再重新联机。

您可以在全局群集中或区域群集中使用该子命令。

如果从全局群集节点使用该子命令，则该子命令可以对任何资源组进行操作。如果在区域群集中使用该子命令，则该子命令只能对区域群集中的资源组成功进行操作。要从全局群集节点对特定区域群集中的资源组进行操作，可以使用 `-z` 选项指定该区域群集的名称。

如果指定了 `-n` 选项，则资源组仅在您指定的节点列表中包含的当前主节点上重新启动。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

resume

在之前由 `suspend` 子命令暂停的指定资源组上继续执行自动恢复操作。

您可以在全局群集中或区域群集中使用该子命令。

如果从全局群集节点使用该子命令，则该子命令可以对任何资源组进行操作。如果在区域群集中使用该子命令，则该子命令只能对区域群集中的资源组成功进行操作。要从全局群集节点对特定区域群集中的资源组进行操作，可以使用 `-z` 选项指定该区域群集的名称。

在您没有明确发出命令继续执行自动恢复过程之前，已暂停的资源组不会自动重新启动或进行故障转移。无论是联机还是脱机，已暂停的数据服务都仍将处于其当前

状态。您仍可在指定的节点上将资源组手动切换到不同的状态。此外，您依然可以启用或禁用资源组中的单个资源。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

set

修改与您指定的资源组相关联的属性。

您可以在全局群集中或区域群集中使用该子命令。

如果从全局群集节点使用该子命令，则该子命令可以对任何资源组进行操作。如果在区域群集中使用该子命令，则该子命令只能对区域群集中的资源组成功进行操作。要从全局群集节点对特定区域群集中的资源组进行操作，可以使用 `-z` 选项指定该区域群集的名称。

您可以使用 `-p Nodelist=node` 或 `-n node` (简便方式) 修改 `Nodelist` 属性。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

show

为您指定的资源组生成配置报告 (按限定符选项过滤)。

您可以在全局群集中或区域群集中使用该子命令。

可以使用 `-r resource` 来仅包括含有资源的资源组。可以使用 `-t resourcetype` 来仅包括含有 `resourcetype` 中某个资源类型的资源组。可以使用 `-n node` 来仅包括在一个或多个节点上处于联机状态的资源组。您可以从全局群集使用 `-z` 选项来仅包括在指定区域群集中处于联机状态的那些资源组。

您可以使用 `-p` 选项显示选定的一组资源组属性 (而非所有资源组属性)。

如果未指定操作数或指定了加号操作数 (+)，则会列出按您指定的任何限定符选项过滤的所有资源组。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

status

为您指定的资源组生成状态报告 (按限定符选项过滤)。

您可以在全局群集中或区域群集中使用该子命令。

如果在区域群集中使用该命令，则该子命令仅应用于该区域群集中的资源组。

可以使用 `-r resource` 来仅包括含有资源的资源组。可以使用 `-t resourcetype` 来仅包括含有 `resourcetype` 中某个资源类型的资源组。可以使用 `-n node` 来仅包括在一个或多个节点上处于联机状态的资源组。您可以从全局群集节点使用 `-z` 选项指定一个区域群集，以便仅包括在指定区域群集中处于联机状态的那些资源组。

如果指定了 `-s state`，则仅会列出具有指定状态的组。

注 - 您可以在 `status` 子命令中指定 `-n` 选项或 `-s` 选项。但是您不能随 `status` 子命令同时指定这两个选项。

如果未指定操作数或指定了加号操作数 (+)，则会列出按您指定的任何限定符选项过滤的所有资源组。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

suspend

暂停自动恢复操作并停止指定的资源组。

您可以在全局群集中或区域群集中使用该子命令。

如果在全局群集节点中使用该子命令，则该子命令可以对任何资源组进行操作。如果在区域群集中使用该子命令，则该子命令只能对区域群集中的资源组成功进行操作。要从全局群集节点对特定区域群集中的资源组进行操作，可以使用 `-z` 选项指定该区域群集的名称。

在您没有明确发出命令继续执行自动恢复过程之前，已暂停的资源组不会自动重新启动或进行故障转移。无论是联机还是脱机，已暂停的数据服务都仍将处于其当前状态。资源组已暂停时，可以通过使用带有子命令（如 `switch`、`online`、`offline`、`disable` 或 `enable`）的 `clresourcegroup(1CL)` 或 `clresource(1CL)` 命令，在特定的节点上将资源组或其资源手动切换到其他状态。无需直接对资源执行操作（如中止应用程序进程或运行应用程序特定的命令），只需使用 `clresourcegroup(1CL)` 或 `clresource(1CL)` 命令即可。这样，群集框架就可以保持资源和资源组的当前状态的准确情形，以便在执行 `resume` 子命令时正确恢复可用性。

为了检查并修复群集中存在的问题或在资源组服务上执行维护，您可能需要暂停资源组的自动恢复过程。

还可以指定 `-k` 选项以立即中止代表受影响资源组中的资源运行的方法。使用 `-k` 选项可以加速资源组的停顿。如果未指定 `-k` 选项，系统会允许方法继续运行，直到这些方法退出或超过所配置的超时时间。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

switch

更改控制您所指定的资源组的一个节点或一组节点。

您可以在全局群集中或区域群集中使用该子命令。

如果在全局群集节点中使用该子命令，则该子命令可以对任何资源组进行操作。如果在区域群集中使用该子命令，则该子命令只能对区域群集中的资源组成功进行操作。

使用 `-n` 选项指定要使资源组在其上联机的节点的列表。您可以从全局群集节点使用 `-z` 选项指定一个区域群集，以便仅包括指定区域群集中的资源组列表。

如果某个资源组尚未联机，则该资源组将在由 `-n` 选项指定的一组节点上联机。但是，处于联机状态的组在新节点上联机之前，会在 `-n` 选项未指定的节点上脱机。

如果随该子命令指定了 `-e`，则会启用资源组集中已联机的所有资源。

您可以指定 `-m` 来监视处于联机状态的资源组集中的所有资源。但是，除非资源已启用并与某个 `MONITOR_START` 方法相关联，否则系统不会真正监视它们。

可以指定 `-M` 选项以指示将处于联机状态的所有资源组置于受管理状态。如果未指定 `-M` 选项，则该子命令对不受管理的资源组没有任何影响。

资源组会按照反映资源和资源组依赖性的顺序进行联机。在命令行中指定组的顺序无关紧要。

优先级较低的资源组可能无法切换到指定的节点，或者当超出负载限制时甚至可能受优先级较高的资源组影响而强制脱机。有关更多信息，请参见 [clnode\(1CL\) \[149\]](#) 手册页中的 `loadlimit` 子命令。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

unmanage

将您所指定的资源组置于不被管理的状态。

您可以在全局群集中或区域群集中使用该子命令。

如果从全局群集节点使用该子命令，则该子命令可以对任何资源组进行操作。如果在区域群集中使用该子命令，则该子命令只能对同一区域群集中的资源组成功进行操作。要从全局群集节点对特定区域群集中的资源组进行操作，可以使用 `-z` 选项指定该区域群集的名称。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

支持以下选项：

注 - 该部分显示了每个选项的短和长两种格式。

`-?`

`--help`

显示帮助信息。

指定此选项时，*subcommand* 可有可无。

如果没有随该选项指定 *subcommand*，将会显示所有可用子命令的列表。

如果随该选项指定了 *subcommand*，将会显示 *subcommand* 的用法。

如果在 `create` 或 `set` 子命令中指定了该选项，将显示所有资源组属性的帮助信息。

如果是与其他选项、子命令或操作数一起指定该选项，将忽略所有这些内容。不会发生任何其他处理。

`-e`
`--enable`
在资源组进入联机状态时，启用该资源组内的所有资源。
此选项只能与 `switch` 和 `online` 子命令一起使用。

`-F`
`--force`
强制删除资源组及其所有资源，即使这些资源处于启用或联机状态。此选项还从其他资源和资源组的任何依赖性属性设置或关联性属性设置中删除资源和资源组。
请谨慎使用 `-F` 选项和 `delete` 子命令。强制删除操作可能会导致引用被删除的资源组的其他资源组发生变化，例如当设置了依赖性或关联性时。在执行强制删除操作后，具有依赖性的资源可能会处于无效或错误状态。如果发生这种情况，您可能需要重新配置或重新启动受影响的具有依赖性的资源。

`-i {- | clconfigfile}`
`--input={- | clconfigfile}`
`--input {- | clconfigfile}`
指定您要使用位于 `clconfigfile` 文件中的配置信息。请参见 [clconfiguration\(5CL\) \[1245\]](#) 手册页。
随该选项指定一个破折号 (-) 将通过标准输入 (stdin) 提供配置信息。
如果指定了其他选项，则这些选项优先于 `clconfigfile` 中的选项和信息。
只有您指定的那些资源组才会受此选项的影响。

`-k`
`--kill`
中止代表您指定的资源组中的资源运行的 RGM 资源方法。
可以将此选项与 `quiesce` 和 `suspend` 子命令一起使用。如果未指定 `-k` 选项，系统会允许方法继续运行，直到这些方法退出或超过所配置的超时时间。

`-m`
`--monitor`
在资源组进入联机状态时，监视该资源组内的所有资源。
但是，除非资源已启用并与某个 `MONITOR_START` 方法相关联，否则系统不会真正监视它们。
此选项只能与 `switch` 和 `online` 子命令一起使用。

`-M`
`--manage`
指定通过 `switch` 或 `online` 子命令进行联机的所有资源组均置于受管理状态。

```
-n node[...]  
--node=node[...]  
--node node[...]
```

指定目标全局群集或区域群集中的一个节点或一系列节点。如果指定 `-z` 选项，则仅可以使用 `-n` 选项指定区域群集主机名，而不能指定全局群集主机名。如果未指定 `-z` 选项，则仅可以使用 `-n` 选项指定全局群集主机名。

可以将节点的名称或标识符指定为 `node`。

与 `list`、`show` 和 `status` 子命令结合使用时，该选项会限制输出。仅会包括当前在节点列表中的一个或多个节点上处于联机状态的那些资源组。

在 `create`、`add-node`、`remove-node` 和 `set` 子命令中指定该选项等同于设置 `Nodelist` 属性。`Nodelist` 属性中节点的顺序指定了在这些节点上使该组联机的顺序。如果没有在 `create` 子命令中指定节点列表，则 `Nodelist` 属性会设置为群集中的所有节点。顺序是任意的。

与 `switch` 和 `online` 子命令一起使用时，该选项指定要在其上使资源组联机的节点。

与 `evacuate` 和 `offline` 子命令一起使用时，该选项指定要在其上使资源组脱机的节点。

与 `restart` 子命令一起使用时，该选项指定要在其上重新启动资源组的节点。资源组在位于指定列表中的当前主节点上重新启动。

```
-o {- | clconfigfile}  
--output={- | clconfigfile}  
--output {- | clconfigfile}
```

将资源组配置信息写入一个文件或标准输出 (stdout)。[clconfiguration\(5CL\) \[1245\]](#) 手册页中描述了该配置信息的格式。

如果您使用该选项指定文件名称，该选项将创建新的文件。然后将配置信息置于该文件中。如果您使用该选项指定 `-`，则会将配置信息发送到标准输出 (stdout)。命令的所有其他标准输出将受到抑制。

仅可以将此选项与 `export` 子命令一起使用。

```
-p name  
--property=name  
--property name
```

指定资源组属性列表。

将此选项与 `show` 子命令一起使用。

有关您可以使用 `create` 或 `set` 子命令设置或修改的属性的信息，请参见 `-p name=value` 选项的相关描述。

如果没有指定该选项，`show` 子命令会列出大部分资源组属性。如果没有指定该选项但在 `show` 子命令中指定了 `-verbose` 选项，则该子命令会列出所有资源组属性。

[Unresolved link to "Oracle Solaris Cluster 数据服务规划和管理指南 中的资源组属性"](#)描述了您可以指定的资源组属性。

```
-p name=value
-p name+=array-values
-p name=array-values
--property=name=value
--property=name+=array-values
--property=name-=array-values
--property name=value
--property name+=array-values
--property name-=array-values
```

设置或修改资源组属性的值。

此选项只能与 `create` 和 `set` 子命令一起使用。

有关可以使用 `show` 子命令显示其相关信息的属性的信息，请参见 `-p name` 选项的描述。

允许多个 `-p` 实例。

与此选项一起使用的运算符如下所述：

<code>=</code>	将属性设置为指定值。 <code>create</code> 和 <code>set</code> 子命令接受此运算符。
<code>+=</code>	将一个或多个值添加到属性值列表。只有 <code>set</code> 子命令接受此运算符。仅可以为接受字符串值列表的属性指定此运算符，例如 <code>Nodelist</code> 。
<code>-=</code>	从属性值列表中删除一个或多个值。只有 <code>set</code> 子命令接受此运算符。仅可以为接受字符串值列表的属性指定此运算符，例如 <code>Nodelist</code> 。

```
-r resource[...]  
--resource=resource[...]  
--resource resource[...]
```

指定一个资源或资源列表。

此选项只能与 `list`、`show` 和 `status` 子命令一起使用。该选项限制这些命令的输出。只会输出包含资源列表中的一个或多个资源的那些资源组。

```
-s state[...]  
--state=state[...]  
--state state[...]
```

指定一个资源组状态或资源组状态列表。

此选项只能与 `status` 子命令一起使用。该选项限制输出，以便仅显示在任何指定节点上处于指定状态的那些资源组。您可以随该选项指定一个或多个以下参数（状态）：

`Error_stop_failed`

显示在您指定的任何节点上处于 `Error_stop_failed` 状态的任何指定资源组。

Not_online

显示在您指定的任何节点上处于 online 之外的任何状态的任何指定资源组。

Offline

仅当某个指定资源组在您指定的所有节点上处于 Offline 状态时才显示该资源组。

Online

显示在您指定的任何节点上处于 Online 状态的任何指定资源组。

Online_faulted

显示在您指定的任何节点上处于 Online_faulted 状态的任何指定资源组。

Pending_offline

显示在您指定的任何节点上处于 Pending_offline 状态的任何指定资源组。

Pending_online

显示在您指定的任何节点上处于 Pending_online 状态的任何指定资源组。

Pending_online_blocked

显示在您指定的任何节点上处于 Pending_online_blocked 状态的任何指定资源组。

Unmanaged

显示在您指定的任何节点上处于 Unmanaged 状态的任何指定资源组。

-S

--scalable

创建一个可伸缩资源组或更新 Maximum primaries 和 Desired primaries 属性。此选项只能与 create 和 add-node 子命令一起使用。

与 create 子命令一起使用时，该选项可以创建可伸缩的资源组，而非故障转移资源组。该选项还会将 Maximum primaries 和 Desired primaries 属性设置为生成的 Nodelist 属性中的节点数。

仅当资源组已经是可伸缩的时，才可以将该选项与 add-node 子命令一起使用。与 add-node 子命令一起使用时，该选项会将 Maximum primaries 和 Desired primaries 属性设置为生成的 Nodelist 属性中的节点数。

也可以使用 -p 选项设置 RG_mode、Maximum primaries 和 Desired primaries 属性。

-t *resourcetype*[...]

--type=*resourcetype*[...]

--type *resourcetype*[...]

指定一个资源类型或资源类型列表。

此选项只能与 `list`、`show` 和 `status` 子命令一起使用。该选项限制这些命令的输出。只会输出包含属于资源类型列表中的类型的一个或多个资源的那些资源组。

您将资源类型指定为 `[prefix.] type[:RT-version]`。例如，`nfs` 资源类型可以表示为 `SUNW.nfs:3.2`、`SUNW.nfs` 或 `nfs`。仅当在群集中注册的一个资源类型具有多个版本时，您才需要包括 `RT-version`。如果没有包括 `prefix`，则会采用 `SUNW`。

`-T seconds`

`--time=seconds`

`--time seconds`

指定在从节点中清除资源组之后阻止资源组切换回该节点的秒数。

仅可以将此选项与 `evacuate` 子命令一起使用。您必须将 `seconds` 指定为介于 0 和 65535 之间的整数值。如果未指定值，将使用默认值 60 秒。

在清除完成之后的 60 秒或指定的秒数内，防止资源组在清除的节点上进行故障转移或自动联机。

但是，如果您使用 `switch` 或 `online` 子命令使资源组联机，或清除的节点重新引导，则清除计时器将立即过期并再次允许自动故障转移。

`-T` 选项指定在完成清除后的 `T` 秒时间段内 RGM 不会在清除的节点上使资源组联机。通过将 `switch` 或 `online` 子命令与 `-n` 选项一起使用将资源组切换到清除的节点，可以覆盖 `-T` 计时器。完成此种切换后，`-T` 计时器对该节点会立即过期。但是，切换命令（例如不含 `-n` 标志的 `online` 或 `remaster`）会继续使用 `-T` 计时器，并且避免将任何资源组切换到清除的节点。

`-u`

如果使用 + 操作数，此选项会指定命令要对其资源组已暂停的资源起作用。

如果指定 + 操作数时没有指定 `-u` 选项，则该命令会忽略所有暂停的资源组。在 `add-node`、`manage`、`offline`、`online`、`quiesce`、`remaster`、`remove-node`、`restart`、`set`、`switch` 或 `unamange` 子命令中指定了 + 操作数时，`-u` 选项有效。

将 + 操作数与 `add-node`、`manage`、`offline`、`online`、`quiesce`、`remaster`、`remove-node`、`restart`、`set`、`switch` 或 `unamange` 子命令一起使用时，该命令会忽略所有暂停的资源组，除非您还指定了 `-u` 选项。

`-v`

`--verbose`

在标准输出 (stdout) 中显示详细信息。

`-V`

`--version`

显示该命令的版本。

如果是与其他选项、子命令或操作数一起指定该选项，将忽略所有这些内容。仅显示该命令的版本。不会发生任何其他处理。

```
-z {zoneclustername | global | all}
--zoneclustername={zoneclustername | global | all}
--zoneclustername {zoneclustername | global | all}
```

指定其中存在资源组并且您要对其执行操作的一个或多个群集。

除 `export` 子命令以外的所有子命令均支持此选项。

如果指定此选项，还必须指定以下列表中的参数之一：

<code>zoneclustername</code>	指定使用此选项的命令仅对名为 <code>zoneclustername</code> 的区域群集中的所有指定资源组起作用。
<code>global</code>	指定使用此选项的命令仅对全局群集中的所有指定资源组起作用。
<code>all</code>	如果在全局群集中使用此参数，则会指定使用此参数的命令对所有群集（包括全局群集和所有区域群集）中的所有指定资源组起作用。 如果在某区域群集中使用此参数，则会指定使用此参数的命令仅对该区域群集中的所有指定资源类型起作用。

支持以下操作数：

<code>resourcegroup</code>	要管理的资源组的名称。
<code>+</code>	所有资源组。

[Intro\(1CL\) \[17\]](#) 手册页列出了该命令集中所有命令的一整套退出状态代码。返回的退出代码还与 [scha_calls\(3HA\) \[897\]](#) 手册页中介绍的返回代码兼容。

如果命令对于所有指定的操作数均成功执行，则返回零 (`CL_NOERR`)。如果某个操作数出错，命令会继续处理操作数列表中的下一个操作数。返回的退出代码始终反映最先发生的错误。

此命令返回下列退出状态代码：

<code>0 CL_NOERR</code>	没有错误 您发出的命令已成功完成。
<code>1 CL_ENOMEM</code>	交换空间不足 某个群集节点耗尽了交换内存或其他操作系统资源。
<code>3 CL_EINVAL</code>	参数无效

您键入的命令不正确，或者使用 `-i` 选项提供的群集配置信息语法不正确。

6 CL_EACCESS

权限被拒绝

指定的对象不可访问。要发出命令，可能需要拥有超级用户或 RBAC 访问权限。有关更多信息，请参见 [Unresolved link to "su1M"](#) 和 [Unresolved link to "rbac5"](#) 手册页。

35 CL_EIO

I/O 错误

发生了物理输入/输出错误。

36 CL_ENOENT

没有这样的对象

由于以下某个原因，找不到您所指定的对象：

- 该对象不存在。
- 您尝试使用 `-o` 选项创建的配置文件路径中的某个目录不存在。
- 您尝试使用 `-i` 选项访问的配置文件包含错误。

38 CL_EBUSY

对象忙

您尝试将一根电缆从上一个群集互连路径移到一个活动群集节点。或者，您尝试将一个节点从尚未删除引用的群集配置中移除。

39 CL_EEXIST

对象已存在

您所指定的设备、设备组、群集互连组件、节点、群集、资源、资源类型、资源组或专用字符串已存在。

例 173 创建新的故障转移资源组

以下示例中的第一个命令将创建故障转移资源组 `rg1` 和 `rg2`。第二个命令将配置文件 `cluster-1.xml` 中包含的资源添加到这些资源组中。

```
# clresourcegroup create rg1 rg2
# clresource create -g rg1,rg2 -i /net/server/export/cluster-1.xml +
```

以下两个示例中的任一个都从全局群集节点在区域群集 `ZC` 中创建故障转移资源组 `rg1` 和 `rg2`。

```
# clresourcegroup create -Z ZC rg1 rg2
# clresourcegroup create ZC:rg1 ZC:rg2
```

例 174 使所有资源组联机

以下命令使所有资源组联机，并启用和监视所有资源。

```
# clresourcegroup online -eM +
```

例 175 将节点添加到 Nodelist 属性中

以下命令将节点 phys-schost-4 添加到所有资源组的 Nodelist 属性中。

```
# clresourcegroup set -p Nodelist+=phys-schost-4 +
```

例 176 从节点清除所有资源组

以下命令从节点 phys-schost-3 清除所有资源组。

```
# clresourcegroup evacuate -n phys-schost-3 +
```

例 177 使资源组在所有节点上脱机

以下命令使资源组 rg1 在所有节点上脱机。

```
# clresourcegroup offline rg1
```

例 178 刷新整个资源组管理器配置

以下示例中的第一个命令将删除所有资源和资源组，即使它们已启用并处于联机状态。第二个命令取消注册所有资源类型。第三个命令创建配置文件 cluster-1.xml 中包含的资源。第三个命令还会注册资源类型并创建这些资源类型所依赖的所有资源组。

```
# clresourcegroup delete --force +
# clresourcetype unregister +
# clresource -i /net/server/export/cluster-1.xml -d +
```

例 179 列出所有资源组

以下命令可列出所有资源组。

```
# clresourcegroup list
rg1
rg2
```

例 180 列出所有资源组及其资源

以下命令可列出所有资源组及其资源。请注意，rg3 没有资源。

```
# clresourcegroup list -v
```

```
Resource Group Resource
-----
rg1          rs-2
rg1          rs-3
rg1          rs-4
rg1          rs-5
rg2          rs-1
rg3          -
```

例 181 列出包含特定资源的所有资源组

以下命令可列出包含 Oracle Solaris Cluster HA for NFS 资源的所有组。

```
# clresource list -t nfs
rg1
```

例 182 通过切换资源组清除 Start_failed 资源状态

Start_failed 资源状况指示 Start 或 Prenet_start 方法对某个资源失败或超时，但该资源的资源组仍已联机。即使该资源处于故障状况，可能不提供服务，其资源组仍然会联机。如果资源的 Failover_mode 属性设置为 None 或者设置为阻止资源组故障转移的另一个值，则可能会出现此状况。

与 Stop_failed 资源状况不同，Start_failed 资源状况不会阻止您或 Oracle Solaris Cluster 软件对资源组执行操作。无需发出 reset 子命令来清除 Start_failed 资源状态。只需执行一个可重新启动资源的命令。

以下命令清除 resource-grp-2 资源组中的资源发生的 Start_failed 资源状态。该命令通过将资源组切换到 schost-2 节点来清除此状态。

```
# clresourcegroup switch -n schost-2 resource-grp-2
```

例 183 通过重新启动资源组清除 Start_failed 资源状态

以下命令清除 resource-grp-2 资源组中的资源发生的 Start_failed 资源状态。该命令通过在初始承载该资源组的 schost-1 节点上重新启动资源组来清除此状态。

```
# clresourcegroup restart resource-grp-2
```

例 184 设置 load_factors 属性

以下命令为两个资源组设置负载因子。

```
# clresourcegroup set -p load_factors=factor1@50,factor2@1 rg1 rg2
```

从全局群集中，以下命令为某个区域群集中的两个资源组设置负载因子。

```
# clresourcegroup set -Z ZC load_factors=factor1@50,factor2@1 rg1 rg2
```

例 185 为资源组设置 priority 属性

以下命令设置资源组的优先级。

```
# clresourcegroup set -p priority=600 rg1
```

rg1 资源组在分配节点时将优先于优先级较低的资源组。在超出硬限制的节点上，rg1 会抢先于优先级较低的其他资源组。如果 rg1 的优先级比另一资源组的优先级高出至少 100，则它在超出软限制的节点上将抢先于该资源组。priority 的默认值为 500。

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

[clresource\(1CL\) \[225\]](#)、[clresourcetype\(1CL\) \[277\]](#)、[cluster\(1CL\) \[461\]](#)、[Intro\(1CL\) \[17\]](#)、[Unresolved link to "su1M"](#)、[scha_calls\(3HA\) \[897\]](#)、[Unresolved link to "attributes5"](#)、[Unresolved link to "rbac5"](#)、[rg_properties\(5\) \[1129\]](#)、[clconfiguration\(5CL\) \[1245\]](#)

超级用户可以运行此命令的所有格式。

所有用户都可以运行带有 -? (帮助) 或 -v (版本) 选项的此命令。

要运行带有其他子命令的 clresourcegroup 命令，超级用户以外的用户需要具有 RBAC 授权。请参见下表。

子命令	RBAC 授权
add-node	solaris.cluster.modify
create	solaris.cluster.modify
delete	solaris.cluster.modify
evacuate	solaris.cluster.admin
export	solaris.cluster.read
list	solaris.cluster.read
manage	solaris.cluster.admin
offline	solaris.cluster.admin
online	solaris.cluster.admin
quiesce	solaris.cluster.admin

子命令	RBAC 授权
remaster	solaris.cluster.admin
remove-node	solaris.cluster.modify
restart	solaris.cluster.admin
resume	solaris.cluster.admin
set	solaris.cluster.modify
show	solaris.cluster.read
status	solaris.cluster.read
suspend	solaris.cluster.admin
switch	solaris.cluster.admin
unmanage	solaris.cluster.admin

名称

clresource, clrs — 管理 Oracle Solaris Cluster 数据服务的资源

```
/usr/cluster/bin/clresource subcommand [-?]  
  
/usr/cluster/bin/clresource -V  
  
/usr/cluster/bin/clresource subcommand [options] -v [resource]...  
  
/usr/cluster/bin/clresource clear [-f errorflag] [-g  
  [resourcegroup,...] [-t [resourcetype,...] [-n node  
  [,...]] [-Z {zoneclustername | global}]  
  {+ | resource...}  
  
/usr/cluster/bin/clresource create -g resourcegroup -t  
  resourcetype [-d] [-p "property-name{node-specifier,...}"=  
  value] [-x "extension-property{node-specifier,...}"=value] [-y  
  standard-property=value] [-Z {zoneclustername | global}]  
  resource  
  
/usr/cluster/bin/clresource create -i {- | clconfiguration} -t  
  resourcetype [-a] [-d] [-g [resourcegroup,...] [-p "  
  property-name{node-specifier,...}"=value] [-x "  
  extension-property{node-specifier,...}"=value] [-y  
  standard-property=value] {+ | resource...}  
  
/usr/cluster/bin/clresource delete [-F] [-g [resourcegroup,...]  
  [-t [resourcetype,...] [-Z {zoneclustername | global}]  
  {+ | resource...}  
  
/usr/cluster/bin/clresource disable [-r] [-g [resourcegroup,...]  
  [-t [resourcetype,...] [-n node[,...]]  
  [-Z {zoneclustername | global}] {+ | resource...}  
  
/usr/cluster/bin/clresource enable [-r] [-g [resourcegroup,...]  
  [-t [resourcetype,...] [-n node[,...]]  
  [-Z {zoneclustername | global}] {+ | resource...}  
  
/usr/cluster/bin/clresource export [-o {- | configfile}]  
  {+ | resource...}  
  
/usr/cluster/bin/clresource list [-g [resourcegroup,...] [-t  
  [resourcetype,...] [-n node[,...]] [-Z  
  {zoneclustername [,...] | global | all}] {+ | resource...}  
  
/usr/cluster/bin/clresource list-props [-l listtype] [-g  
  [resourcegroup,...] [-p "property-name{node-specifier,...}" ,...]  
  [-t [resourcetype,...] [-x "extension-property{node-specifier,...}" ,...]  
  [-y "standard-property{node-specifier,...}" ,...] [-Z  
  {zoneclustername [,...] | global | all}] {+ | resource...}  
  
/usr/cluster/bin/clresource monitor [-g [resourcegroup,...] [-t  
  [resourcetype,...] [-n node[,...]] [-Z  
  {zoneclustername | global}] {+ | resource...}
```

```

/usr/cluster/bin/clresource set [-g [resourcegroup,...] [-p "
    property-name[{node-specifier,...}]=value] [-t
    [resourcetype,...] [-x "extension-property[{node-specifier,...}]=
    value] [-y standard-property [+ = | - =]value] [-Z
    {zoneclustername | global}] {+ | resource...}

/usr/cluster/bin/clresource show [-g [resourcegroup,...] [-p
    property-name[{node-specifier,...}],...] [-t [resourcetype,...]
    [-x "extension-property[{node-specifier,...}],...] [-y "
    standard-property[{node-specifier,...}],...] [-Z
    {zoneclustername [,...] | global | all}] {+ | resource...}

/usr/cluster/bin/clresource status [-g [resourcegroup,...] [-s
    [state,...] [-t [resourcetype,...] [-n node[,...]]
    [-Z {zoneclustername [,...] | global | all}] {+ | resource...}

/usr/cluster/bin/clresource unmonitor [-g [resourcegroup,...]
    [-t [resourcetype,...] [-n node[,...]]
    [-Z {zoneclustername | global}] {+ | resource...}

```

clresource 命令可管理 Oracle Solaris Cluster 数据服务的资源。clrs 命令是 clresource 命令的简短格式。clresource 命令和 clrs 命令是相同的。您可以使用此命令的任一种格式。

此命令的一般格式如下所示：

```
clresource [subcommand] [options] [operands]
```

仅当 *options* 指定 -? 选项或 -v 选项时，才可以省略 *subcommand*。

此命令的每个选项都有长格式和简短格式两种格式。在本手册页的 选项 部分给出了每个选项的两种格式，以及对该选项的描述。

在区域群集中使用此命令

在区域群集中，可以将 clresource 命令与所有子命令（export 除外）一起使用。

也可以将 -z 选项与所有子命令（export 除外）一起使用来指定要将操作限制到的特定区域群集的名称。另外，还可以将区域群集名称附加到资源名称 (*zoneclustername* : *resource*)，以便将操作限制到特定的区域群集。

您可以从全局群集节点访问所有区域群集信息，但无法从特定区域群集访问其他区域群集信息。如果不将操作限制到特定区域群集，则所用的子命令仅会在当前群集中运行。

一个区域群集中的某个资源可能对另一个区域群集中的某个资源存在依赖性，或者对全局群集中的某个资源存在依赖性。另外，全局群集中的某个资源可能对该全局群集中的任一区域群集上的某个资源存在依赖性。只能从全局群集设置这种群集间依赖性。

您可以使用以下命令指定群集间依赖性：

```

# clresource set -p resource_dependencies=target-
zc
:target-rs source-zc:

```

source-rs

例如，如果需要指定从区域群集 ZC1 中的资源 R1 到区域群集 ZC2 中的资源 R2 的依赖性，请使用以下命令：

```
# clresource set -p resource_dependencies=ZC2:R2 ZC1:R1
```

如果需要指定区域群集 ZC1 资源 R1 对全局群集资源 R2 的依赖性，请使用以下命令：

```
# clresource set -p resource_dependencies=global:R2 ZC1:R1
```

支持现有资源依赖性 (Strong、Weak 、Restart 和 Offline-Restart) 。

资源状况和状态

资源状况和资源状态是基于每个节点进行维护的。一个给定的资源在每个群集节点上可以具有不同的状况和状态。

资源组管理器 (Resource Group Manager, RGM) 基于对资源调用的方法在每个节点上设置资源状况。例如，在给定节点上对某个资源成功运行 STOP 方法后，该资源在此节点上的状况将为 OFFLINE。如果 STOP 方法返回非零值或超时，则该资源的状况为 Stop_failed

可能的资源状况包括：

- Online
- Offline
- Start_failed
- Stop_failed
- Monitor_failed
- Online_not_monitored
- Starting
- Stopping
- Not_online

注 - 状况名称 (例如 Offline 和 Start_failed) 不区分大小写。指定状况名称时，可以使用大小写字母的任意组合。

除资源状况外，RGM 还可以维护资源状态，资源状态可由资源本身使用 API 设置。Status Message 字段实际上包括两个组成部分：状态关键字和状态消息。状态消息由资源有选择性地设置，它是在状态关键字后面输出的任意文本字符串。

资源状态可能值的描述如下：

DEGRADED	资源已联机，但其性能或可用性可能在某方面出现了下降。
FAULTED	资源遇到错误，无法正常运行。

OFFLINE	资源已脱机。
ONLINE	资源已联机，正在提供服务。
UNKNOWN	当前状态是未知或正在转换。

支持以下子命令：

clear

清除与指定为命令操作数的资源关联的错误标志。此子命令接受使用加号 (+) 作为操作数来指定清除所有资源的错误标志。

您可以在全局群集中或区域群集中使用该子命令。

以下选项可以过滤操作数列表，以限制要在其上清除错误标志的资源：

<code>-g resourcegroup</code>	只清除操作数列表内属于 <i>resourcegroup</i> 中资源组成员的资源。
<code>-n node</code>	清除指定的一个或多个节点上的资源。如果未提供 <code>-n</code> 选项，该命令将清除所有节点上的资源。
<code>-t resourcetype</code>	只清除操作数列表内属于 <i>resourcetype</i> 中资源类型的实例的资源。
<code>-z {zoneclustername global}</code>	只清除您指定的一个或多个特定群集中的资源。要从全局群集清除区域群集中的资源，请使用 <code>-z</code> 选项指定区域群集。

默认情况下，`clear` 子命令将会清除 `STOP_FAILED` 错误标志。要显式指定要清除的错误标志，请使用 `-f` 选项。`-f` 选项接受的唯一错误标志是 `STOP_FAILED` 错误标志。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。

create

创建作为命令操作数指定的资源。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集在特定区域群集中创建资源，可以使用 `-z` 选项指定区域群集的名称。

将 `create` 和 `-i` 选项一起使用以指定配置文件时，该子命令接受使用加号 (+) 作为操作数。使用 `+` 操作数时，将会创建配置文件中提供的尚不存在的所有资源。

默认情况下，创建的资源处于启用状态，且启用了监视功能。但是，仅当资源的资源组联机后，该资源才会联机并受到监视。要在禁用状态下创建资源，请指定 `-d` 选项。

创建资源时，可以使用以下选项设置属性值：

`-p property-name= value` 设置标准属性或扩展属性，前提是它们的名称是唯一的。

`-x extension-property= value` 设置扩展属性。

`-y standard-property= value` 设置标准属性。

`node-specifier` 是 `-p` 和 `-x` 选项的可选限定符。它指示创建资源时将只在指定节点上设置属性，而不在群集中其他节点上设置指定的属性。如果未使用 `node-specifier`，则会在群集中的所有节点上设置指定的属性。`node-specifier` 语法的示例包括：

`-x "myprop{phys-schost-1}"`

大括号 ({}) 表示您只想要在节点 `phys-schost-1` 上设置指定的属性。对于大多数 shell，必须将大括号包含在引号中。

可以使用以下语法在两个节点上设置属性：

`-x "myprop{phys-schost-1,phys-schost-2}"`

超级用户以外的用户需要具有 `solaris.cluster.modify` 基于角色的访问控制 (role-based access control, RBAC) 授权才能使用此子命令。

另请参见 `delete` 子命令的描述。

delete

删除作为命令操作数指定的资源。此子命令接受使用加号 (+) 作为操作数来指定删除所有资源。

您可以在全局群集中或区域群集中使用该子命令。

此子命令将按照为满足资源间依赖性而需要遵照的顺序删除多个资源，而不管在命令行上以哪种顺序指定了资源。

同时删除多个资源时，该命令将在多个步骤中执行。如果命令被中断（例如，某个节点发生故障），某些资源组的配置可能会处于无效状态。要更正此问题并完成资源的删除，请在某个正常运行的节点上再次发出同一命令。

以下选项可以过滤操作数列表，以限制删除的资源：

`-g resourcegroup` 只删除操作数列表内属于 `resourcegroup` 中资源组的成员的资源。

`-t resourcetype` 只删除操作数列表内属于 `resourcetype` 中资源类型的实例的资源。

`-z {zoneclustername | global}` 只删除您指定的一个或多个特定群集中的资源。要从全局群集中删除区域群集中的资源，请使用 `-z` 选项指定区域群集。

默认情况下，仅当符合以下条件时，才会删除某个资源：

- 必须已禁用该资源。

- 必须已消除该资源的所有依赖性。

要强制删除指定的资源，请指定 -F 选项。请慎用此选项，因为它会有以下效果：

- 删除所有指定的资源，即使是未禁用的资源。
- 从其他资源的资源依赖性设置中删除所有指定的资源。

这些效果可能会导致群集中的服务丧失。未删除的相关资源也可能被置于无效状况或错误状况。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

另请参见 `create` 子命令的描述。

disable

禁用指定为命令操作数的资源。此子命令接受使用加号 (+) 作为操作数来指定禁用所有资源。

您可以在全局群集中或区域群集中使用该子命令。

以下选项可以过滤操作数列表，以限制禁用的资源：

- | | |
|--|---|
| <code>-g resourcegroup</code> | 只禁用操作数列表内属于 <code>resourcegroup</code> 中资源组的成员的资源。 |
| <code>-n node</code> | 可以使用 <code>-n node</code> 来禁用一个或多个节点上的资源。 |
| <code>-t resourcetype</code> | 只禁用操作数列表内属于 <code>resourcetype</code> 中资源类型的实例的资源。 |
| <code>-z {zoneclustername global}</code> | 只禁用您指定的一个或多个特定群集中的资源。要从全局群集删除区域群集中的资源，请使用 <code>-z</code> 选项指定区域群集。 |

`-r` 选项将会禁用依赖于指定为命令操作数的资源的任何资源。即使这些资源未指定为命令操作数，也会将其禁用。`-g` 选项和 `-t` 选项不适用于仅为满足资源依赖性而禁用的资源。

此子命令不会影响资源的监视状态。如果该资源在启用的情况下是受监视的，则禁用后仍受监视。如果后来重新启用该资源，则还会继续监视该资源。

该子命令将按照为满足资源间依赖性而需要遵照的顺序禁用资源，而不管在命令行上以哪种顺序指定了资源。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。

另请参见 `enable` 子命令的描述。

enable

启用指定为命令操作数的资源。此子命令接受使用加号 (+) 作为操作数来指定启用所有资源。

您可以在全局群集中或区域群集中使用该子命令。

以下选项可以过滤操作数列表，以限制启用的资源：

<code>-g resourcegroup</code>	仅启用操作数列表内属于 <i>resourcegroup</i> 中资源组的成员的资源。
<code>-n node</code>	可以使用 <code>-n node</code> 来启用一个或多个节点上的资源。
<code>-t resourcetype</code>	仅启用操作数列表内属于 <i>resourcetype</i> 中资源类型的实例的资源。
<code>-z {zoneclustername global}</code>	仅启用您指定的一个或多个特定群集中的资源。要从全局群集启用区域群集中的资源，请使用 <code>-z</code> 选项指定区域群集。

要确保满足所有必需的资源依赖性，请指定 `-r` 选项。`-r` 选项将会启用指定为命令操作数的资源所依赖的任何资源。即使这些资源未指定为命令操作数，也会将其启用。`-g` 选项和 `-t` 选项不适用于仅为满足资源依赖性而禁用的资源。

该子命令将按照为满足资源间依赖性而需要遵照的顺序启用资源，而不管在命令行上以哪种顺序指定了资源。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。

另请参见 `disable` 子命令的描述。

export

以 [clconfiguration\(5CL\) \[1245\]](#) 手册页所述的格式导出群集资源配置。

只能在全局群集中使用该子命令。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

list

显示指定为命令操作数的资源的列表。默认情况下，会显示所有资源。

您可以在全局群集中或区域群集中使用该子命令。

以下选项可以过滤操作数列表，以限制显示的资源：

<code>-g resourcegroup</code>	只显示操作数列表内属于 <i>resourcegroup</i> 中资源组的成员的资源。
<code>-n node</code>	可以使用 <code>-n node</code> 来仅列出在一个或多个节点上处于联机状态的那些资源。
<code>-t resourcetype</code>	只显示属于 <i>resourcetype</i> 中资源类型的实例的资源。

`-z {zoneclustername | global | all}` 只显示您指定的一个或多个特定群集中的资源。要从全局群集显示区域群集中的资源，请使用 `-z` 选项指定区域群集。

此子命令接受使用加号 (+) 作为操作数来指定显示所有资源配置。可以通过指定 `-g` 选项或 `-t` 选项，将显示的信息限于特定资源组或资源类型。如果未提供操作数，则会显示指定资源组中的所有资源，或显示属于指定资源类型的实例的所有资源。

如果指定 `-v` 选项，还将显示列表中每个资源的资源组和资源类型。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

list-props

显示指定为命令操作数的资源的属性列表。

您可以在全局群集中或区域群集中使用该子命令。

以下选项可以过滤操作数列表，以限制要显示其属性的资源：

`-g resourcegroup` 仅显示操作数列表内属于 `resourcegroup` 中资源组成员的资源的属性。

`-t resourcetype` 仅显示操作数列表内属于 `resourcetype` 中资源类型实例的资源的属性。

`-l` 选项指定要显示的资源属性的类型：

`-l all` 指定显示标准属性和扩展属性。

`-l extension` 指定只显示扩展属性。默认情况下，只显示扩展属性。

`-l standard` 指定只显示标准属性。

如果未指定 `-l` 选项，则只显示扩展属性。要显示标准属性，请使用 `-p` 选项或 `-y` 选项显式指定这些属性。

以下选项用于限制要显示的资源属性集：

`-p property-name` 只显示 `property-name` 中指定的属性。您可以在 `property-name` 中指定标准属性和扩展属性。

`-x extension-property` 只显示一个或多个节点上的、已在 `extension-property` 中指定的扩展属性。

`-y standard-property` 只显示 `standard-property` 中指定的标准属性。

`node-specifier` 是 `-p`、`-x` 和 `-y` 选项的可选限定符。它指示将只显示指定节点上的属性，而不显示群集中其他节点上的指定属性。如果未使用 `node-specifier`，则会显示群集中所有节点上的指定属性。`node-specifier` 语法的示例包括：

`-x "myprop{phys-schost-1}"`

大括号 ({}) 表示您只想要显示节点 `phys-schost-1` 上的指定属性。对于大多数 shell，必须将大括号包含在引号中。

可以使用以下语法显示两个节点上的属性：

```
-x "myprop{phys-schost-1,phys-schost-2}"
```

如果指定 `-v` 选项，还将显示每个属性的描述。

此子命令接受使用加号 (+) 作为操作数来指定显示所有资源属性。如果未提供操作数，则会显示指定资源组中所有资源的属性，或显示属于指定资源类型的实例的所有资源的属性。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

```
-z {zoneclustername | global | all}
```

列出您指定的一个或多个特定群集中的资源的属性。要从全局群集列出区域群集中的资源，请使用 `-z` 选项指定区域群集。

`monitor`

开启对指定为命令操作数的资源的监视。此子命令接受使用加号 (+) 作为操作数来指定开启对所有资源的监视。

您可以在全局群集中或区域群集中使用该子命令。

以下选项可以过滤操作数列表，以限制要对其开启监视的资源：

`-g resourcegroup` 只对操作数列表内属于 `resourcegroup` 中资源组的成员的资源开启监视。

`-n node` 只对在一个或多个节点上处于联机状态的那些资源开启监视。

`-t resourcetype` 只对操作数列表内属于 `resourcetype` 中资源类型的实例的资源开启监视。

`-z {zoneclustername | global}` 只对您指定的一个或多个特定群集中的资源开启监视。要从全局群集开启对区域群集中资源的监视，请使用 `-z` 选项指定区域群集。

开启对某个资源的监视后，仅当符合以下条件时，才会监视该资源：

- 已启用该资源。
- 包含该资源的资源组至少已在一个群集节点上联机。

注 - 开启对某个资源的监视不会启用该资源。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。

另请参见 `unmonitor` 子命令的描述。

set

设置指定为命令操作数的资源的特定属性。此子命令接受使用加号 (+) 作为操作数来指定修改所有资源的特定属性。

您可以在全局群集中或区域群集中使用该子命令。

以下选项可以过滤操作数列表，以限制要修改其属性的资源：

-g resourcegroup 只修改操作数列表内属于 *resourcegroup* 中资源组成员的资源的属性。

-t resourcetype 只修改操作数列表内属于 *resourcetype* 中资源类型实例的资源的属性。

使用以下选项设置属性值：

-p property-name= value 设置标准属性或扩展属性，前提是它们的名称是唯一的。

-x extension-property= value 设置扩展属性。

-y standard-property= value 设置标准属性。

node-specifier 是 **-p** 和 **-x** 选项的可选限定符，用于更新每节点扩展属性。它指示仅在指定节点上设置相应属性，而不在群集中其他节点上设置指定的属性。如果未使用 *node-specifier*，则会在群集中的所有节点上设置指定的属性。*node-specifier* 语法的示例包括：

-x "myprop{phys-schost-1}"

大括号 ({}) 表示您只想要在节点 *phys-schost-1* 上设置指定的属性。对于大多数 shell，必须将大括号包含在引号中。

可以使用以下语法在两个节点上设置属性：

-x "myprop{phys-schost-1,phys-schost-2}"

-z {zoneclustername | global}

只设置您指定的一个或多个特定群集中的资源的属性。要从全局群集设置区域群集中资源的属性，请使用 **-z** 选项指定区域群集。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

show

显示指定为命令操作数的资源的配置。默认情况下，会显示所有资源的配置。

您可以在全局群集中或区域群集中使用该子命令。

以下选项可以过滤操作数列表，以限制要显示其配置的资源列表：

-g resourcegroup 只显示操作数列表内属于 *resourcegroup* 中资源组成员的资源的配置。

`-n node` 可以使用 `-n node` 来仅显示在一个或多个节点上处于联机状态的那些资源的配置。

`-t resourcetype` 只显示操作数列表内属于 `resourcetype` 中资源类型实例的资源配置。

`-z {zoneclustername | global | all}`

只显示您指定的一个或多个特定群集中的资源。要从全局群集显示区域群集中的资源，请使用 `-z` 选项指定区域群集。

以下选项用于限制显示的资源属性集：

`-p property-name` 只显示 `property-name` 中指定的属性。您可以在 `property-name` 中指定标准属性和扩展属性。

`-x extension-property` 只显示一个或多个节点上的、已在 `extension-property` 中指定的扩展属性。

`-y standard-property` 只显示 `standard-property` 中指定的标准属性。

`node-specifier` 是 `-p`、`-x` 和 `-y` 选项的可选限定符。它指示将只显示指定节点上的属性，而不显示群集中其他节点上的指定属性。如果未使用 `node-specifier`，则会显示群集中所有节点上的指定属性。`node-specifier` 语法的示例包括：

`-x "myprop{phys-schost-1}"`

大括号 ({}) 表示您只想要显示节点 `phys-schost-1` 上的指定属性。对于大多数 shell，必须将大括号包含在引号中。

可以使用以下语法显示两个节点上的属性：

`-x "myprop{phys-schost-1,phys-schost-2}"`

此子命令接受使用加号 (+) 作为操作数来指定显示所有资源配置。可以通过指定 `-g` 选项或 `-t` 选项，将显示的信息限于特定资源组或资源类型。如果未提供操作数，该子命令将显示所有指定资源的配置。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

status

显示指定为命令操作数的资源的状态。默认情况下，会显示所有资源的状态。

以下选项可以过滤操作数列表，以限制要显示其状态的资源的列表：

`-g resourcegroup` 仅显示操作数列表内属于 `resourcegroup` 中资源组成员的资源的状态。

`-n node` 可以使用 `-n node` 来仅显示在一个或多个节点上处于联机状态的那些资源的状态。不能同时指定 `-n` 和 `-s` 选项。

-s <i>state</i>	仅显示操作数列表内处于 <i>state</i> 中的状况的资源的状况。不能同时指定 -n 和 -s 选项。
-t <i>resourcetype</i>	仅显示操作数列表内属于 <i>resourcetype</i> 中资源类型实例的资源的状况。
-z { <i>zoneclustername</i> global all}	显示您指定的一个或多个特定群集中的资源的状况。要从全局群集显示区域群集中资源的状况，请使用 -z 选项指定区域群集。

此子命令接受使用加号 (+) 作为操作数来指定显示所有资源的状况。可以通过指定 -g 选项或 -t 选项，将显示的信息限于特定资源组或资源类型。如果未提供操作数，将会显示所有指定资源的状况。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

unmonitor

关闭对指定为命令操作数的资源的监视。此子命令接受使用加号 (+) 作为操作数来指定关闭对所有资源的监视。

您可以在全局群集中或区域群集中使用该子命令。

如果关闭对某个被禁用资源的监视，该资源不会受到影响。该资源及其监视器已脱机。

注 - 关闭对某个资源的监视不会禁用该资源。但是，当您禁用某个资源时，不需要关闭对该资源的监视。禁用的资源及其监视器将保持脱机状态。

以下选项可以过滤操作数列表，以限制要对其关闭监视的资源：

-g <i>resourcegroup</i>	只对操作数列表内属于 <i>resourcegroup</i> 中资源组的成员的资源关闭监视。
-n <i>node</i>	只对在一个或多个节点上处于联机状态的那些资源关闭监视。
-t <i>resourcetype</i>	只对操作数列表内属于 <i>resourcetype</i> 中资源类型的实例的资源关闭监视。
-z { <i>zoneclustername</i> global}	只对您指定的一个或多个特定群集中的资源关闭监视。要从全局群集关闭对区域群集中资源的监视，请使用 -z 选项指定区域群集。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。

另请参见 `disable` 子命令和 `monitor` 子命令的描述。

支持以下选项：

-?

--help

显示帮助信息。使用该选项时，不会执行任何其他处理。

指定此选项时，子命令可有可无。

如果仅指定此选项而不使用子命令，将会显示此命令的子命令列表。

如果使用子命令且指定此选项，将会显示该子命令的用法选项。

-a

--automatic

当基于群集配置文件 ([clconfiguration\(5CL\) \[1245\]](#)) 创建资源时，将自动执行以下附加操作：

- 注册资源类型
- 创建资源组
- 创建操作数列表中指定的资源所依赖的资源

群集配置信息包含的信息必须足以执行以下所有操作：

- 启用要注册的资源类型
- 启用要创建的资源组
- 启用要创建的资源

此选项仅可以与 `create` 子命令一同指定。如果指定此选项，还必须指定 `-i` 选项并提供配置文件。

-d

--disable

创建资源后禁用该资源。此选项仅可以与 `create` 子命令一同指定。默认情况下，资源在启用状态下创建。

启用某个资源并不保证该资源联机。仅当某个资源的资源组已至少在一个节点上联机后，该资源才会联机。

-f *errorflag*

--flag=*errorflag*

--flag *errorflag*

显式指定要通过 `clear` 子命令清除的错误标志。此选项仅可以与 `clear` 子命令一同指定。默认情况下，`clear` 子命令将会清除 `STOP_FAILED` 错误标志。

- f 选项接受的唯一错误标志是 `STOP_FAILED` 错误标志。

-F

--force

强制删除未禁用的资源。此选项仅可以与 `delete` 子命令一同指定。

请慎用此选项，因为它会有以下效果：

- 删除所有指定的资源，即使是未禁用的资源。
- 从其他资源的资源依赖性设置中删除所有指定的资源。

这些效果可能会导致群集中的服务丧失。未删除的相关资源也可能被置于无效状况或错误状况。

`-g resourcegroup[,...]`
`--resourcegroup=resourcegroup[,...]`
`--resourcegroup resourcegroup[,...]`

指定一个资源组或资源组列表。

对于除 `create` 以外的子命令，该命令仅对操作数列表内属于指定的资源组成员的资源起作用。使用 `-g` 选项指定资源组。

随 `create` 子命令指定了 `-g` 选项时，`clresource` 将在指定的资源组中创建资源。使用此选项时，只能指定一个资源组。

`-i {- | clconfiguration}`
`--input={- | clconfiguration}`
`--input {- | clconfiguration}`

指定用于创建或修改资源的配置信息。此信息必须符合 [clconfiguration\(5CL\) \[1245\]](#) 手册页中定义的格式。此信息可包含在文件中或者通过标准输入提供。要指定标准输入，请指定 `-` 而不是文件名。

仅会创建或修改作为命令操作数提供的资源。在命令中指定的选项将会覆盖配置信息中设置的任何选项。如果配置信息中缺少配置参数，则必须在命令行中指定这些参数。

将 `-i` 选项与 `create` 子命令结合使用时，`clresource` 将会注册所有必需的资源类型，并创建所有必需的资源组。您必须提供注册和配置所需的全部信息。所有其他配置数据将被忽略。

`-l listtype`
`--listtype=listtype`
`--listtype listtype`

指定要通过 `list-props` 子命令显示的资源属性的类型。此选项仅可以与 `list-props` 子命令一同指定。

必须根据以下列表为 `listtype` 指定一个值：

<code>all</code>	指定显示标准属性和扩展属性。
<code>extension</code>	指定只显示扩展属性。默认情况下，只显示扩展属性。
<code>standard</code>	指定只显示标准属性。

如果未指定 `-l` 选项，则只显示扩展属性。要显示标准属性，请使用 `-p` 选项或 `-y` 选项显式指定这些属性。

```
-n node[,...]  
--node=node[,...]  
--node node[,...]
```

指定目标全局群集或区域群集中的一个节点或一系列节点。可以使用节点名称或节点 ID 指定每个节点。

如果指定 -z 选项，则仅可以使用 -n 选项指定区域群集主机名，而不能指定全局群集主机名。如果未指定 -z 选项，则仅可以使用 -n 选项指定全局群集主机名。

可以与此选项一起指定的子命令如下所述：

<code>disable</code>	只禁用操作数列表中承载在指定节点上的资源。
<code>enable</code>	只启用操作数列表中承载在指定节点上的资源。
<code>list</code>	只显示操作数列表中承载在指定节点上的那些资源的列表。
<code>monitor</code>	只监视操作数列表中承载在指定节点上的那些资源。
<code>show</code>	只显示操作数列表中承载在指定节点上的那些资源的配置信息。
<code>status</code>	仅报告操作数列表中承载在指定节点上的资源的状态。
<code>unmonitor</code>	只取消监视操作数列表中承载在指定节点上的那些资源。

```
-o {- | clconfiguration}  
--output={- | clconfiguration}  
--output {- | clconfiguration}
```

指定要将资源配置信息写入到的位置。此位置可以是文件，也可以是标准输出。要指定标准输出，请指定短划线 (-) 而不是文件名。如果指定标准输出，将会抑制该命令的其他所有标准输出。此选项仅可以与 `export` 子命令一同指定。

只会写入作为命令操作数提供的资源的配置信息。信息是采用 [clconfiguration\(5CL\) \[1245\]](#) 手册页中定义的格式写入的。

```
-p property-name=value  
-p property-name+=array-values  
-p property-name-=array-values  
--property=property-name=value  
--property=property-name+=array-values  
--property=property-name-=array-values  
--property property-name=value  
--property property-name+=array-values  
--property property-name-=array-values
```

设置作为命令操作数提供的资源的属性值。只能随 `create` 子命令和 `set` 子命令指定此选项的赋值格式。

使用 -p 选项可以指定任何标准属性或扩展属性。如果某个资源类型的某个扩展属性与该资源类型的某个标准属性同名，则使用 -p 选项会返回错误。在这种情况下，请使用 -x 选项指定扩展属性，使用 -y 选项指定标准属性。

有关标准属性的描述，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

有关某个资源类型的扩展属性的描述，请参见该资源类型的文档。

与此选项一起使用的运算符如下所述：

= 将属性设置为指定值。create 子命令和 set 子命令接受此运算符。

+= 将一个或多个值添加到某个字符串数组值。只有 set 子命令接受此运算符。仅可以为接受字符串值列表的属性指定此运算符，例如 Resource_dependencies 。

-= 从某个字符串数组值中删除一个或多个值。只有 set 子命令接受此运算符。仅可以为接受字符串值列表的属性指定此运算符，例如 Resource_dependencies 。

要在群集节点的子集上设置每节点扩展属性，请在设置属性时指定节点。将节点列表括在大括号中，并将其附加在属性名称的后面，如下所示：

```
name{node}
```

node 是以逗号分隔的节点名称或节点 ID 的列表。有关每节点扩展属性的更多信息，请参见 [rt_properties\(5\) \[1143\]](#) 手册页。

要设置每节点资源对群集节点子集的依赖性，请采用以下格式指定每个每节点依赖性：

```
myres1@node1,myres2@node2,myres3@node3
```

对于 gds-rs 资源，以下命令设置资源 trancos-3-rs 对节点 ptrancos1 的依赖性和资源 trancos-4-rs 对节点 ptrancos2 的依赖性：

```
# clresource set -p \  
resource_dependencies=trancos-3-rs@ptrancos1,trancos-4-rs@ptrancos2 gds-rs
```

```
phys-schost-1# clresource show -p resource_dependencies gds-rs  
=== Resources ===  
Resource: gds-rs  
Resource_dependencies: trancos-3-rs@ptrancos1 trancos-4-rs@ptrancos2
```

要设置本地节点范围的资源依赖性，请采用以下格式指定 LOCAL_NODE 限定符：

```
myres1{LOCAL_NODE},myres2{LOCAL_NODE}
```

对于 gds-rs 资源，以下命令设置本地节点对资源 trancos-3-rs 的依赖性：

```
# clresource set -p resource_dependencies=trancos-3-rs{LOCAL_NODE} gds-rs
```

```
phys-schost-1# clresource show -p resource_dependencies gds-rs
=== Resources ===
Resource: gds-rs
Resource_dependencies: trancos-3-rs{LOCAL_NODE}
```

有关每节点资源依赖性和依赖性范围限定符的更多信息，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

```
-p property-name[...]
--property=property-name[...]
--property property-name[...]
```

为 `list-props` 子命令和 `show` 子命令指定属性列表。

使用 `-p` 选项可以指定任何标准属性或扩展属性。如果某个资源类型的某个扩展属性与该资源类型的某个标准属性同名，则使用 `-p` 选项会返回错误。在这种情况下，请使用 `-x` 选项指定扩展属性，使用 `-y` 选项指定标准属性。

有关标准属性的描述，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

有关某个资源类型的扩展属性的描述，请参见该资源类型的文档。

如果未指定此选项，`list-props` 子命令和 `show` 子命令将会列出所有资源属性或大多数资源属性，具体取决于是否同时指定了 `-v` 选项。

```
-r
--recursive
```

以递归方式启用或禁用资源，以确保满足所有必需的依赖性。此选项仅可以与 `disable` 子命令和 `enable` 子命令一同指定。

此选项与这些子命令一起使用的效果如下所示：

<code>disable</code>	禁用依赖于指定为命令操作数的资源的任何资源。即使这些资源未指定为命令操作数，也会将其禁用。
<code>enable</code>	启用指定为命令操作数的资源所依赖的任何资源。即使这些资源未指定为命令操作数，也会将其启用。

```
-s state[...]
--state=state[...]
--state state[...]
```

为 `list` 子命令和 `status` 子命令指定状况列表。

此选项将会限制输出，以便只包括在节点列表中的一个或多个节点上处于指定状况之一的资源。

可能的状况如下：

- `Online`
- `Offline`
- `Start_failed`
- `Stop_failed`

- Monitor_failed
- Online_not_monitored
- Starting
- Stopping
- Not_online

注 - 状况名称 (例如 Offline 和 Start_failed) 不区分大小写。指定状况名称时, 可以使用大小写字母的任意组合。

-t resourcetype[,...]
--type=resourcetype[,...]
--type resourcetype[,...]

指定一个资源类型或资源类型列表。

对于接受此选项的所有子命令 (create 除外), 该命令只对同时满足以下两个条件的资源起作用:

- 资源在操作数列表中。
- 资源是 - t 选项指定的资源类型的实例。

将 -t 选项与 clresource create 一起指定时, 将会创建指定类型的资源。只能指定一个资源类型。

有关资源类型名称格式的描述, 请参见[Unresolved link to " Oracle Solaris Cluster 数据服务规划和管理指南 中的合法的 RGM 名称"](#)。

-u

如果使用 + 操作数, 此选项会指定命令要对其资源组已暂停的资源起作用。

如果指定 + 操作数时未指定 -u 选项, 命令将会忽略其资源组已暂停的所有资源。当随 clear、disable、enable、monitor、set 或 unmonitor 命令指定了 + 操作数时, -u 选项有效。

将 + 操作数与 clear、disable、enable、monitor、set 或 unmonitor 子命令一起使用时, 除非同时指定了 - u 选项, 否则命令将会忽略其资源组已暂停的所有资源。

-V

--version

显示该命令的版本。

指定此选项时, 不要将其与子命令、操作数或其他选项结合使用。子命令、操作数或其他选项将被忽略。-v 选项只显示命令的版本。不会执行任何其他操作。

-v

--verbose

将详细消息显示到标准输出。

可以在任何格式的此命令中指定该选项。

请不要同时指定 `-v` 选项和 `-o` 选项。否则 `-v` 选项将被忽略。`-o` 选项会抑制其他所有标准输出。

```
-x extension-property=value
-x extension-property+=array-value
-x extension-property-=array-value
--extension-property=extension-property=value
--extension-property=extension-property+=array-value
--extension-property=extension-property-=array-value
--extension-property extension-property=value
--extension-property extension-property+=array-value
--extension-property extension-property-=array-value
```

设置或修改作为命令操作数提供的资源的扩展属性值。

通常，使用 `-p` 选项可以指定任何标准属性或扩展属性。如果某个资源类型的某个扩展属性与该资源类型的某个标准属性同名，则使用 `-p` 选项会返回错误。在这种情况下，请使用 `-x` 选项指定扩展属性，使用 `-y` 选项指定标准属性。

只能随 `create` 子命令和 `set` 子命令指定此选项的赋值格式。

有关某个资源类型的扩展属性的描述，请参见该资源类型的文档。

与此选项一起使用的运算符如下所述：

- `=` 将属性设置为指定值。`create` 子命令和 `set` 子命令接受此运算符。
- `+=` 将一个或多个值添加到某个字符串数组值。只有 `set` 子命令接受此运算符。仅可以为接受字符串值列表的属性指定此运算符，例如 `Resource_dependencies`。
- `-=` 从某个字符串数组值中删除一个或多个值。只有 `set` 子命令接受此运算符。仅可以为接受字符串值列表的属性指定此运算符，例如 `Resource_dependencies`。

要在群集节点的子集上设置每节点扩展属性，请在设置属性时指定节点。将节点列表括在大括号中，并将其附加在属性名称的后面，如下所示：

```
name{node}
```

`node` 是以逗号分隔的节点名称或节点 ID 的列表。有关每节点属性的更多信息，请参见 [rt_properties\(5\) \[1143\]](#) 手册页。

```
-x extension-property[,...]
--extension-property=extension-property[,...]
--extension-property extension-property[,...]
```

为 `list-props` 子命令和 `show` 子命令指定扩展属性列表。

有关某个资源类型的扩展属性的描述，请参见该资源类型的文档。

使用 `-p` 选项可以指定任何标准属性或扩展属性。如果某个资源类型的某个扩展属性与该资源类型的某个标准属性同名，则使用 `-p` 选项会返回错误。在这种情况下，请使用 `-x` 选项指定扩展属性，使用 `-y` 选项指定标准属性。

如果未指定此选项，`list-props` 子命令和 `show` 子命令将会列出所有资源属性或大多数资源属性，具体取决于是否同时指定了 `-v` 选项。

```
-y standard-property=value
-y standard-property+=array-value
-y standard-property-=array-value
--standard-property=standard-property=value
--standard-property=standard-property+=array-value
--standard-property=standard-property-=array-value
--standard-property standard-property=value
--standard-property standard-property+=array-value
--standard-property standard-property-=array-value
```

设置或修改作为命令操作数提供的资源的标准属性值。

使用 `-p` 选项可以指定任何标准属性或扩展属性。如果某个资源类型的某个扩展属性与该资源类型的某个标准属性同名，则使用 `-p` 选项会返回错误。在这种情况下，请使用 `-x` 选项指定扩展属性，使用 `-y` 选项指定标准属性。

只能随 `create` 子命令和 `set` 子命令指定此选项的赋值格式。

有关标准属性的描述，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

与此选项一起使用的运算符如下所述：

<code>=</code>	将属性设置为指定值。 <code>create</code> 子命令和 <code>set</code> 子命令接受此运算符。
<code>+=</code>	将一个或多个值添加到某个字符串数组值。只有 <code>set</code> 子命令接受此运算符。仅可以为接受字符串值列表的属性指定此运算符，例如 <code>Resource_dependencies</code> 。
<code>-=</code>	从某个字符串数组值中删除一个或多个值。只有 <code>set</code> 子命令接受此运算符。仅可以为接受字符串值列表的属性指定此运算符，例如 <code>Resource_dependencies</code> 。

```
-y standard-property[,...]
--standard-property=standard-property[,...]
--standard-property standard-property[,...]
```

为 `list-props` 子命令和 `show` 子命令指定标准属性列表。

有关标准属性的描述，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

使用 `-p` 选项可以指定任何标准属性或扩展属性。如果某个资源类型的某个扩展属性与该资源类型的某个标准属性同名，则使用 `-p` 选项会返回错误。在这种情况下，请使用 `-x` 选项指定扩展属性，使用 `-y` 选项指定标准属性。

如果未指定此选项，`list-props` 子命令和 `show` 子命令将会列出所有资源属性或大多数资源属性，具体取决于是否同时指定了 `-v` 选项。

`-z {zoneclustername | global | all}`
`--zoneclustername={zoneclustername | global | all}`
`--zoneclustername {zoneclustername | global | all}`

指定一个或多个存在资源且您要对其执行操作的群集。

除 `export` 子命令以外的所有子命令均支持此选项。

如果指定此选项，还必须指定以下列表中的参数之一：

<code>zoneclustername</code>	指定使用此选项的命令仅对名为 <code>zoneclustername</code> 的区域群集中的所有指定资源起作用。
<code>global</code>	指定使用此选项的命令仅对全局群集中的所有指定资源进行操作。
<code>all</code>	如果在全局群集中使用此参数，则会指定使用此参数的命令对所有群集（包括全局群集和所有区域群集）中的所有指定资源起作用。 如果在某区域群集中使用此参数，则会指定使用此参数的命令仅对该区域群集中的所有指定资源起作用。

仅支持以下操作数：

`resource` 指定要管理的一个或多个资源。如果子命令接受多个资源，则可以使用加号 (+) 指定所有资源。

如果命令对于所有指定的操作数均成功执行，则返回零 (`CL_NOERR`)。如果某个操作数出错，命令会继续处理操作数列表中的下一个操作数。返回的退出代码始终反映最先发生的错误。

可返回下列退出代码：

0 `CL_NOERR`

没有错误

您发出的命令已成功完成。

1 `CL_ENOMEM`

交换空间不足

某个群集节点耗尽了交换内存或其他操作系统资源。

3 `CL_EINVAL`

参数无效

您键入的命令不正确，或者使用 `-i` 选项提供的群集配置信息语法不正确。

6 CL_EACCESS

权限被拒绝

指定的对象不可访问。要发出命令，可能需要拥有超级用户或 RBAC 访问权限。有关更多信息，请参见 [Unresolved link to " su1M"](#) 和 [Unresolved link to " rbac5"](#) 手册页。

9 CL_ESTATE

对象处于错误状况

您尝试修改一个在特定时间无法修改，或任何时候均无法修改的属性、资源组或其他对象。

10 CL_EMETHOD

资源方法失败

资源方法失败。方法因以下某个原因而失败：

- 尝试创建资源或修改资源属性时，`validate` 方法失败。
- 尝试启用、禁用或删除资源时，`validate` 以外的其他方法失败。

15 CL_EPROP

属性无效

您使用 `-p`、`-y` 或 `-x` 选项指定的属性或值不存在或不被允许。

36 CL_ENOENT

没有这样的对象

由于以下某个原因，找不到您所指定的对象：

- 该对象不存在。
- 您尝试使用 `-o` 选项创建的配置文件路径中的某个目录不存在。
- 您尝试使用 `-i` 选项访问的配置文件包含错误。

39 CL_EEXIST

对象已存在

您所指定的设备、设备组、群集互连组件、节点、群集、资源、资源类型、资源组或专用字符串已存在。

41 CL_ETYPE

类型无效

您使用 `-t` 或 `-p` 选项指定的类型不存在。

这些退出值与 [scha_calls\(3HA\) \[897\]](#) 手册页中描述的返回码兼容。

例 186 创建资源

此示例将在名为 `rg-failover` 的资源组中创建一个名为 `rs-nfs` 的资源。该资源是 `SUNW.nfs` 资源类型的实例。创建的资源处于启用状况，并开启了资源监视功能。

```
# clresource create -g rg-failover -t SUNW.nfs rs-nfs
```

以下两个命令中的任何一个都在区域群集 `ZC` 内的资源组 `rg-failover` 中创建一个名为 `rs-nfs` 的资源。可以从全局群集节点执行这些命令，也可以在区域群集 `ZC` 内执行这些命令。如果从区域群集执行命令，则使用区域群集名称显式定义资源范围的操作是可选的。

```
# clresource create -g rg-failover -t SUNW.nfs -Z ZC rs-nfs
```

```
# clresource create -g rg-failover -t SUNW.nfs ZC:rs-nfs
```

例 187 开启对资源的监视

此示例将开启对名为 `rs-nfs` 的资源的监视。

```
# clresource monitor rs-nfs
```

开启对某个资源的监视后，在使用 `clresource unmonitor` 命令显式关闭监视之前，监视将一直保持开启状态。禁用和启用某个资源不影响其监视状态。

例 188 启用资源

此示例将启用资源组 `rg-failover` 和 `rg-failover2` 中的所有资源。

```
# clresource enable -g rg-failover,rg-failover2 +
```

此命令不会影响资源的监视状态。

例 189 设置资源属性

此示例会将 `SUNW.nfs` 资源类型的所有实例的 `r_description` 属性设置为 `HA-NFS res`。

```
# clresource set -t SUNW.nfs -p r_description="HA-NFS res" +
```

例 190 设置每节点资源属性

此示例将资源 `rs-oracle` 的每节点属性 `oracle_sid` 在不同节点上设置为不同的值，如下所述：

- 在节点 `phys-schost-1` 和节点 `phys-schost-2` 上，此属性设置为 `myora1`。
- 在节点 `phys-schost-3` 上，此属性设置为 `myora2`。

此示例假定大括号字符对于所用的 shell 具有特殊含义。因此，节点列表附加到的每个属性名称都包含在双引号中。

```
# clresource set -p "oracle_sid{phys-schost-1,phys-schost-2}"=myora1 \  
-p "oracle_sid{phys-schost-3}"=myora2 rs-oracle
```

例 191 设置每节点资源依赖性

此示例设置 gds-rs 每个节点的资源依赖性，以便它依赖于两个不同的逻辑主机资源。

```
# clresource set -p resource_dependencies=node-3-rs@pnode1,node-4-rs@pnode2 gds-rs  
  
# clresource show -p resource_dependencies gds-rs  
Resource: gds-rs  
Standard Properties:  
Resource_dependencies: node-3-rs@pnode1,node-4-rs@pnode2
```

例 192 向字符串数组属性添加值

此示例将值 rs-oracle 添加到资源 rs-myapp 的字符串数组属性 resource_dependencies 中。此字符串数组属性中的现有值不变。

```
# clresource set -p resource_dependencies+=rs-oracle rs-myapp  
  
# clresource show -p resource_dependencies rs-myapp  
Resource: rs-myapp  
Standard Properties:  
Resource_dependencies: rs-nfs rs-oracle
```

例 193 删除资源

此示例将删除名为 rs-nfs 的资源。

```
# clresource delete rs-nfs
```

例 194 更新整个群集配置

此示例将通过执行以下操作序列来更新整个群集配置：

1. 使群集中的所有资源组脱机，删除所有资源，然后删除所有资源组
2. 取消注册所有资源类型
3. 创建配置文件 /net/server/export/mycluster.xml 中指定的所有资源，注册这些资源的资源类型，然后创建所有必需的资源组

```
# clresourcegroup delete --force +  
# clresourcetype unregister +  
# clresource -i /net/server/export/mycluster.xml -a +
```

例 195 列出资源

此示例将列出所有资源。

```
# clresource list
logicalhost1
rs-nfs-1
rs-nfs-2
logicalhost2
rs-apache-1
```

例 196 列出资源及其组和类型

此示例将列出所有资源及其资源组和资源类型。

```
# clresource list -v
```

Resource Name	Resource Group	Resource Type
logicalhost1	rg-failover-1	SUNW.LogicalHostname
rs-nfs-1	rg-failover-1	SUNW.nfs
logicalhost2	rg-failover-2	SUNW.LogicalHostname
rs-nfs-2	rg-failover-2	SUNW.nfs
rs-apache-1	rg-failover-1	SUNW.apache

例 197 列出特定类型的资源

此示例将列出 nfs 资源类型的所有实例。

```
# clresource list -t nfs
rs-nfs-1
rs-nfs-2
```

例 198 列出资源类型的扩展属性和描述

此示例将列出 nfs 资源类型的扩展属性以及每个扩展属性的描述。

```
# clresource list-props -t nfs -v
```

Properties	Descriptions
Monitor_retry_count	Number of PMF restarts allowed for the fault monitor
Monitor_retry_interval	Time window (minutes) for fault monitor restarts
Rpcbind_nullrpc_timeout	Timeout(seconds) to use when probing rpcbind
Nfsd_nullrpc_timeout	Timeout(seconds) to use when probing nfsd
Mountd_nullrpc_timeout	Timeout(seconds) to use when probing mountd
Statd_nullrpc_timeout	Timeout(seconds) to use when probing statd
Lockd_nullrpc_timeout	Timeout(seconds) to use when probing lockd
Rpcbind_nullrpc_reboot	Boolean to indicate if we should reboot system when null rpc call on rpcbind fails
Nfsd_nullrpc_restart	Boolean to indicate if we should restart nfsd when null rpc call fails

Mountd_nullrpc_restart Boolean to indicate if we should restart mountd when
null rpc call fails

*Line breaks in the Descriptions column are added to enhance
the readability of this example. Actual output from the command does not
contain these line breaks.*

例 199 通过禁用然后启用资源清除 Start_failed 资源状态

Start_failed 资源状况指示 Start 或 Preinet_start 方法对某个资源失败或超时，但该资源的资源组仍已联机。即使该资源处于故障状况，可能不提供服务，其资源组仍然会联机。如果资源的 Failover_mode 属性设置为 None 或者设置为阻止资源组故障转移的另一个值，则可能会出现此状况。

与 Stop_failed 资源状况不同，Start_failed 资源状况不会阻止您或 Oracle Solaris Cluster 软件对资源组执行操作。无需发出 *command clear* 命令来清除 Start_failed 资源状况。只需执行一个可重新启动资源的命令。

以下命令通过禁用然后重新启用 resource-1 资源来清除该资源发生的 Start_failed 资源状态。

```
# clresource disable resource-1  
# clresource enable resource-1
```

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

[Intro\(1CL\) \[17\]](#), [clreslogicalhostname\(1CL\) \[207\]](#), [clresourcegroup\(1CL\) \[253\]](#),
[clresourcetype\(1CL\) \[277\]](#), [clressharedaddress\(1CL\) \[289\]](#), [cluster\(1CL\) \[461\]](#),
[scha_calls\(3HA\) \[897\]](#), [clconfiguration\(5CL\) \[1245\]](#), [Unresolved link to "attributes5"](#),
[r_properties\(5\) \[1103\]](#), [Unresolved link to "rbac5"](#)

[Unresolved link to "Oracle Solaris Cluster 数据服务规划和管理指南"](#)

超级用户可以运行此命令的所有格式。

任何用户都可以运行带有以下选项的此命令：

- -? 选项
- -V 选项

要运行带有子命令的此命令，超级用户以外的用户需要具有 RBAC 授权。请参见下表。

子命令	RBAC 授权
create	solaris.cluster.modify
delete	solaris.cluster.modify
disable	solaris.cluster.admin
enable	solaris.cluster.admin
export	solaris.cluster.read
list	solaris.cluster.read
list-props	solaris.cluster.read
set	solaris.cluster.modify
monitor	solaris.cluster.admin
clear	solaris.cluster.admin
show	solaris.cluster.read
status	solaris.cluster.read
unmonitor	solaris.cluster.admin

名称

clreslogicalhostname, clrslh — 管理 Oracle Solaris Cluster 逻辑主机名的资源

```
/usr/cluster/bin/clreslogicalhostname [subcommand] -?  
  
/usr/cluster/bin/clreslogicalhostname -V  
  
/usr/cluster/bin/clreslogicalhostname [subcommand [options]] -v  
[lresource]...  
  
/usr/cluster/bin/clreslogicalhostname create -g resourcegroup  
[-h lhost[,...]] [-N netif@node[,...]] [-p name=value]  
[-Z {zoneclustername | global}] [-d] lresource  
  
/usr/cluster/bin/clreslogicalhostname create -i  
{- | clconfiguration} [-a] [-g resourcegroup[,...]] [-p  
name=value] [-d] {+ | lresource...}  
  
/usr/cluster/bin/clreslogicalhostname delete [-g resourcegroup[,...]]  
[-Z {zoneclustername | global}] [-F] {+ | lresource...}  
  
/usr/cluster/bin/clreslogicalhostname disable [-g resourcegroup[,...]]  
[-R] [-n node[,...]] [-Z {zoneclustername | global}]  
{+ | lresource...}  
  
/usr/cluster/bin/clreslogicalhostname enable [-g resourcegroup[,...]]  
[-R] [-n node[,...]] [-Z {zoneclustername | global}]  
{+ | lresource...}  
  
/usr/cluster/bin/clreslogicalhostname export [-o {- | configfile}]  
{+ | lresource...}  
  
/usr/cluster/bin/clreslogicalhostname list [-s state[,...]]  
[-g resourcegroup[,...]] [-Z {zoneclustername  
[,...]} | global | all] {+ | lresource...}  
  
/usr/cluster/bin/clreslogicalhostname list-props [-l listtype]  
[-p name[,...]] [-Z {zoneclustername [,...]} | global | all]  
{+ | lresource...}  
  
/usr/cluster/bin/clreslogicalhostname monitor [-g resourcegroup[,...]]  
[-Z zoneclustername | all | global] {+ | lresource...}  
  
/usr/cluster/bin/clreslogicalhostname reset [-f errorflag] [-g  
resourcegroup[,...]] [-Z {zoneclustername | global}]  
{+ | lresource...}  
  
/usr/cluster/bin/clreslogicalhostname set [-i  
{- | clconfiguration}] [-g resourcegroup[,...]] [-p name  
{+|-}=value] [-Z {zoneclustername}] {+ | lresource...}  
  
/usr/cluster/bin/clreslogicalhostname show [-g resourcegroup[,...]]  
[-p name[,...]] [-Z {zoneclustername [,...]} | global | all]
```

```
[+ | lresource...]  
  
/usr/cluster/bin/clreslogicalhostname status [-s state[,...]]  
      [-n node[,...]] [-g resourcegroup[,...]] [-Z  
      {zoneclustername [,...] | global | all}] [+ | lresource...]  
  
/usr/cluster/bin/clreslogicalhostname unmonitor [-g  
      resourcegroup[,...]] [-Z {zoneclustername | global}]  
      [+ | lresource...]
```

`clreslogicalhostname` 命令可管理 Oracle Solaris Cluster 逻辑主机名的资源。`clrslh` 命令是 `clreslogicalhostname` 命令的简短格式。`clreslogicalhostname` 命令和 `clrslh` 命令是相同的。您可以使用此命令的任一种格式。

`clreslogicalhostname` 命令包含一些用于方便地创建逻辑主机名资源的内置选项。`clreslogicalhostname` 命令还支持自动创建 Solaris IP 多路径 (IP multipathing, IPMP) 组。

`clreslogicalhostname` 命令的一些子命令可修改资源配置。

- `disable`
- `enable`
- `monitor`
- `reset`
- `set`
- `unmonitor`

`clreslogicalhostname` 命令的某些子命令只获取有关资源的信息。可以从全局群集或区域群集使用这些子命令：以下命令只获取有关资源的信息：

- `export`
- `list`
- `list-props`
- `show`
- `status`

为了避免此命令产生不可预测的结果，请从全局群集节点运行该命令的所有格式。

此命令的一般格式如下所示：

```
clreslogicalhostname [subcommand] [options] [operands]
```

仅当 *options* 指定了 `-?`、`-o`、`-v` 或 `-V` 选项时，才能省略 *subcommand*。

此命令的每个选项都有长格式和简短格式两种格式。在本手册页的“选项”部分给出了每个选项的两种格式，以及对该选项的描述。

在区域群集中的操作

在区域群集中，可以将 `clreslogicalhostname` 命令与所有子命令（`export` 除外）一起使用。

也可以将 `-z` 选项与所有子命令（`export` 除外）一起使用来指定要将操作限制到的特定区域群集的名称。另外，还可以将区域群集名称附加到逻辑主机名资源 (`zoneclustername : lhresource`)，以便将操作限制到特定的区域群集。

您可以从全局群集节点访问所有区域群集信息，但无法从特定区域群集访问其他区域群集信息。如果不将操作限制到特定区域群集，则所用的子命令仅会在当前群集中运行。

支持以下子命令：

create

创建指定为命令操作数的逻辑主机名资源。

将 `create` 和 `-i` 选项一起使用以指定配置文件时，该子命令接受使用加号 (+) 作为操作数。使用 + 操作数时，将会创建配置文件中提供的尚不存在的所有资源。

使用 `create` 子命令之前，请确保 `/etc/netmasks` 文件具有所有逻辑主机名对应的 IP 地址子网和网络掩码条目。如有必要，请编辑 `/etc/netmasks` 文件以添加缺少的任何条目。

默认情况下，创建的资源处于启用状态，且启用了监视功能。但是，仅当资源的资源组联机后，该资源才会联机并受到监视。要在禁用状态下创建资源，请指定 `-d` 选项。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集在区域群集中创建逻辑主机名资源，请使用 `-z` 选项指定该区域群集的名称。

超级用户以外的用户需要具有 `solaris.cluster.modify` 基于角色的访问控制 (Role-Based Access Control, RBAC) 授权才能使用该子命令。

另请参见 `delete` 子命令的描述。

delete

删除指定为命令操作数的逻辑主机名资源。此子命令接受使用加号 (+) 作为操作数来指定删除所有资源。

`-g` 选项可以过滤操作数列表，以限制删除的资源。`-g` 选项只会删除操作数列表内属于 `resourcegroup` 中资源组的成员的资源。

- 默认情况下，仅当符合以下条件时，才会删除某个资源：
 - 已禁用该资源。
 - 已消除该资源的所有依赖性。
- 要确保删除所有指定的资源，请指定 `-F` 选项。`-F` 选项的效果如下所示：

- 删除所有指定的资源，即使是未禁用的资源。
- 从其他资源的资源依赖性设置中删除所有指定的资源。

将按照满足资源间依赖性所需的顺序禁用资源，而不管资源在命令行上的指定顺序。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集删除区域群集中的逻辑主机名资源，请使用 `-z` 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

另请参见 `create` 子命令的描述。

disable

禁用指定为命令操作数的逻辑主机名资源。此子命令接受使用加号 (+) 作为操作数来指定禁用所有资源。

`-g` 选项可以过滤操作数列表，以限制禁用的资源。`-g` 选项只会禁用操作数列表内属于 `resourcegroup` 中资源组的成员的资源。

要确保满足所有必需的资源依赖性，请指定 `-R` 选项。`-R` 选项将会禁用依赖于指定为命令操作数的资源的任何资源，即使这些资源未指定为命令操作数。`-g` 选项和 `-t` 选项不适用于仅为满足资源依赖性而禁用的资源。

将按照满足资源间依赖性所需的顺序禁用资源，而不管资源在命令行上的指定顺序。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集禁用在区域群集中注册的逻辑主机名资源，请使用 `-z` 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。

另请参见 `enable` 子命令的描述。

enable

启用指定为命令操作数的逻辑主机名资源。此子命令接受使用加号 (+) 作为操作数来指定启用所有资源。

`-g` 选项可以过滤操作数列表，以限制启用的资源。`-g` 选项仅启用操作数列表内属于 `resourcegroup` 中资源组的成员的资源。

要确保满足所有必需的资源依赖性，请指定 `-R` 选项。`-R` 选项将会启用依赖于指定为命令操作数的资源的任何资源，即使这些资源未指定为命令操作数。`-g` 选项不适用于仅为满足资源依赖性而启用的资源。

将按照满足资源间依赖性所需的顺序禁用资源，而不管资源在命令行上的指定顺序。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集启用在区域群集中注册的逻辑主机名资源，请使用 `-z` 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。

另请参见 `disable` 子命令的描述。

export

以 `clconfiguration(5CL) [1245]` 手册页所述的格式导出逻辑主机名资源配置。

只能在全局群集中使用该子命令。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

list

显示指定为命令操作数的逻辑主机名资源的列表。默认情况下，会显示所有资源。

`-g` 选项可以过滤操作数列表，以限制显示的资源。`-g` 选项只会显示操作数列表内属于 `resourcegroup` 中资源组的成员的资源。

此子命令接受使用加号 (+) 作为操作数来指定特定资源组中的所有资源，或者指定属于特定资源类型实例的所有资源。如果未提供操作数，则会显示指定资源组中的所有资源，或显示属于指定资源类型的实例的所有资源。

如果指定 `-v` 选项，还将显示列表中每个资源的资源组和资源类型。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集查看在区域群集中注册的逻辑主机名资源，请使用 `-z` 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

list-props

显示指定为命令操作数的逻辑主机名资源的属性列表。默认情况下，会显示所有资源的扩展属性。

以下选项可以过滤操作数列表，以限制要显示其属性的资源：

`-g resourcegroup` 仅显示操作数列表内属于 `resourcegroup` 中资源组成员的逻辑主机名资源的属性。

`-l` 选项指定要显示的资源属性的类型：

`-l all` 指定显示标准属性和扩展属性。

`-l extension` 指定只显示扩展属性。默认情况下，只显示扩展属性。

`-l standard` 指定只显示标准属性。

如果未指定 `-l` 选项，将只显示扩展属性，除非使用 `-p` 选项或 `-y` 选项明确指定标准属性。

`-p` 选项用于限制要显示的资源属性集。`-p` 选项只显示 *namelist* 中指定的属性。您可以在 *namelist* 中指定标准属性和扩展属性。

如果指定 `-v` 选项，还将显示每个属性的描述。

此子命令接受使用加号 (+) 作为操作数来指定特定资源组中的所有资源，或者指定属于特定资源类型实例的所有资源。如果未提供操作数，则会显示指定资源组中所有资源的属性，或显示属于指定资源类型的实例的所有资源的属性。

您可以在全局群集中或区域群集中使用该子命令。

要查看区域群集中的逻辑主机名资源的属性列表，请使用 `-z` 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

monitor

开启对指定为命令操作数的逻辑主机名资源的监视。此子命令接受使用加号 (+) 作为操作数来指定开启对所有资源的监视。

`-g` 选项可以过滤操作数列表，以限制监视的资源。`-g` 选项只会监视操作数列表内属于 *resourcegroup* 中资源组的成员的资源。

- 开启对某个资源的监视后，仅当符合以下条件时，才会监视该资源：
- 已启用该资源。
- 包含该资源的资源组至少已在一个群集节点上联机。

注 - 开启对某个资源的监视不会启用该资源。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集监视区域群集中的资源，请使用 `-z` 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。

另请参见 `unmonitor` 子命令的描述。

reset

清除与指定为命令操作数的逻辑主机名资源关联的错误标志。此子命令接受使用加号 (+) 作为操作数来指定清除所有资源的错误标志。

`-g` 选项可以过滤操作数列表，以限制重置的资源。`-g` 选项只会重置操作数列表内属于 *resourcegroup* 中资源组的成员的资源。

默认情况下，`reset` 子命令将会清除 `STOP_FAILED` 错误标志。要显式指定要清除的错误标志，请使用 `-f` 选项。`-f` 选项接受的唯一错误标志是 `STOP_FAILED` 错误标志。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集重置区域群集中的逻辑主机名资源，请使用 `-z` 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。

set

修改指定为命令操作数的逻辑主机名资源的特定属性。此子命令接受使用加号 (+) 作为操作数来指定修改所有资源的特定属性。

`-g` 选项可以过滤操作数列表，以限制修改的资源。`-g` 选项只会修改操作数列表内属于 `resourcegroup` 中资源组的成员的资源。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集设置区域群集中逻辑主机名资源的属性，请使用 `-z` 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

show

显示指定为命令操作数的逻辑主机名资源的配置。默认情况下，会显示所有资源的配置。

`-g` 选项可以过滤操作数列表，以限制要显示其配置的资源。`-g` 选项只会显示操作数列表内属于 `resourcegroup` 中资源组成员的资源的配置。

`-p` 选项用于限制要显示的资源属性集。`-p` 选项只显示 `namelist` 中指定的属性。您可以在 `namelist` 中指定标准属性和扩展属性。

此子命令接受使用加号 (+) 作为操作数来指定特定资源组中的所有资源，或者指定属于特定资源类型实例的所有资源。如果未提供操作数，则会显示指定资源组中所有资源的配置，或显示属于指定资源类型实例的所有资源的配置。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集查看区域群集中逻辑主机名资源的配置，请使用 `-z` 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

status

显示指定为命令操作数的逻辑主机名资源的状态。默认情况下，会显示所有资源的状态。

以下选项可以过滤操作数列表，以限制要显示其状态的资源的列表：

`-g resourcegroup` 仅显示操作数列表内属于 `resourcegroup` 中资源组成员的资源的状态。

`-n nodelist` 仅显示操作数列表内承载在 `nodelist` 中节点上的资源的状态。

`-s statelist`

仅显示操作数列表内处于 *statelist* 中的状况的资源的状态。

此子命令接受使用加号 (+) 作为操作数来指定特定资源组中的所有资源，或者指定属于特定资源类型实例的所有资源。如果未提供操作数，则会显示指定资源组中所有资源的状态，或显示属于指定资源类型实例的所有资源的状态。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集查看区域群集中逻辑主机名资源的状态，请使用 `-z` 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

`unmonitor`

关闭对指定为命令操作数的逻辑主机名资源的监视。此子命令接受使用加号 (+) 作为操作数来指定关闭对所有资源的监视。

如果关闭对某个被禁用资源的监视，该资源不会受到影响。该资源及其监视器已脱机。

注 - 关闭对某个资源的监视不会禁用该资源。但是，当您禁用某个资源时，不需要关闭对该资源的监视。禁用的资源及其监视器将保持脱机状态。

`-g` 选项可以过滤操作数列表，以限制要对其关闭监视的资源。`-g` 选项将对操作数列表内属于 *resourcegroup* 中资源组的成员的资源关闭监视。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集关闭对区域群集中逻辑主机名资源的监视，请使用 `-z` 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。

另请参见 `disable` 子命令和 `monitor` 子命令的描述。

支持以下选项：

`-?`

`--help`

显示帮助信息。使用该选项时，不会执行任何其他处理。

指定此选项时，子命令可有可无。

如果仅指定此选项而不使用子命令，将会显示此命令的子命令列表。

如果使用子命令且指定此选项，将会显示该子命令的用法选项。

此选项与特定子命令一起使用的效果如下所示：

create

与 -g 选项一起指定时，此选项将会显示指定资源组的所有资源属性的帮助信息。

set

显示指定为命令操作数的资源的属性的帮助信息。

-a

--automatic

当基于群集配置信息创建资源时，将自动执行以下附加操作：

- 注册资源类型
- 创建资源组
- 创建操作数列表中指定的资源所依赖的资源

- 群集配置信息包含的信息必须足以执行以下所有操作：
 - 启用要注册的资源类型
 - 启用要创建的资源组
 - 启用要创建的资源

此选项仅可以与 create 子命令一同指定。如果指定此选项，还必须指定 -i 选项并提供配置文件。

-d

--disable

创建资源后禁用该资源。此选项仅可以与 create 子命令一同指定。默认情况下，资源在启用状态下创建。

启用某个资源并不保证该资源联机。仅当资源所在的资源组已在至少一个节点上联机后，该资源才会联机。

-f *errorflag*

--flag *errorflag*

显式指定要通过 reset 子命令清除的错误标志。此选项仅可以与 reset 子命令一同指定。默认情况下，reset 子命令将会清除 STOP_FAILED 错误标志。

-f 选项接受的唯一错误标志是 STOP_FAILED 错误标志。

-F

--force

强制删除未禁用的资源。此选项仅可以与 delete 子命令一同指定。

-g *resourcegroup*[,...]

--resourcegroup *resourcegroup*[,...]

指定一个资源组或资源组列表。

`extension` 指定只显示扩展属性。默认情况下，只显示扩展属性。

`standard` 指定只显示标准属性。
如果未指定 `-l` 选项，则只显示扩展属性，除非使用 `-p` 选项明确指定标准属性。

`-n node[,...]`
`--node node[,...]`

指定目标全局群集或区域群集中的一个节点或一系列节点。可以使用节点名称或节点 ID 指定每个节点。如果指定 `-z` 选项，则仅可以使用 `-n` 选项指定区域群集主机名，而不能指定全局群集主机名。如果未指定 `-z` 选项，则仅可以使用 `-n` 选项指定全局群集主机名。

可以与此选项一起指定的子命令如下所述：

`disable` 只禁用操作数列表中承载在指定节点上的资源。

`enable` 只启用操作数列表中承载在指定节点上的资源。

`status` 仅报告操作数列表中承载在指定节点上的资源的状态。

`-N netif@node[,...]`
`--netiflist netif@node[,...]`

指定一个资源属性。通过 `-N` 选项，可以设置 `NetIfList` 属性，而不必使用 `-p` 选项来指定该属性。如果未指定 `-N`，`clreslogicalhostname` 命令将尝试根据可用 IPMP 组或公共适配器以及与 `HostnameList` 属性关联的子网来设置 `NetIfList` 属性。

可以采用 `ipmpgroup@node[,...]` 或 `publicNIC@node[,...]` 格式指定 `NetIfList` 属性。如果未使用 `-N`，或者将该选项与 `publicNIC@node` 一起使用，则 `clreslogicalhostname` 命令将尝试创建必要的 IPMP 组。系统将会创建包含基本默认值的单适配器 IPMP 组，以后，用户可以使用标准的 Solaris IPMP 接口修改这些组。只会在全局群集节点中自动创建 IPMP 组。

您可以使用 `-N`，而无需使用 `-p` 直接设置 `NetIfList` 属性。但是，不能在同一命令中使用 `-N` 并显式设置 `NetIfList`。

`-N` 只能与 `create` 子命令一起使用。

`-o {- | clconfiguration}`
`--output {- | clconfiguration}`

指定要将资源配置信息写入到的位置。此位置可以是文件，也可以是标准输出。要指定标准输出，请指定 `-` 而不是文件名。如果指定标准输出，将会抑制该命令的其他所有标准输出。此选项仅可以与 `export` 子命令一同指定。

只会写入作为命令操作数提供的资源的配置信息。信息是采用 [clconfiguration\(5CL\) \[1245\]](#) 手册页中定义的格式写入的。

```
-p name=value
-p name+=array-values
-p name-=array-values
--property name=value
--property name+=value-values
--property name-=value-values
```

设置资源的标准属性和扩展属性。此选项仅可以与 `create` 子命令和 `set` 子命令一同指定。

有关标准属性的描述，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

有关某个资源类型的扩展属性的描述，请参见该资源类型的文档。

与此选项一起使用的运算符如下所述：

=	将属性设置为指定值。create 子命令和 set 子命令接受此运算符。
+=	将一个或多个值添加到某个字符串数组值。只有 set 子命令接受此运算符。只能为字符串数组值指定此运算符。
-=	从某个字符串数组值中删除一个或多个值。只有 set 子命令接受此运算符。只能为字符串数组值指定此运算符。

如果只需在群集节点的子集上设置每节点属性，请通过在属性名称后面附加节点列表（用大括号括住）来指定要设置该属性的节点，如下所示：

```
name{nodelist}
```

nodelist 是以逗号分隔的节点名称或节点 ID 的列表。有关每节点属性的更多信息，请参见 [rt_properties\(5\) \[1143\]](#) 手册页。

```
-p name[,...]
--property name[,...]
```

为 `list-props` 子命令和 `show` 子命令指定属性列表。

可将此选项用于资源的标准属性和扩展属性。

有关标准属性的描述，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

有关某个资源类型的扩展属性的描述，请参见该资源类型的文档。

如果未指定此选项，`list-props` 子命令和 `show` 子命令将会列出所有资源属性或大多数资源属性，具体取决于是否同时指定了 `-v` 选项。

```
-R
--recursive
```

以递归方式启用或禁用资源，以确保满足所有必需的依赖性。此选项仅可以与 `disable` 子命令和 `enable` 子命令一同指定。

此选项与这些子命令一起使用的效果如下所示：

disable	禁用依赖于指定为命令操作数的资源的任何资源，即使这些资源未指定为命令操作数。
enable	启用指定为命令操作数的资源所依赖的任何资源，即使这些资源未指定为命令操作数。

-s *state*[,...]

--state *state*[,...]

为 `list` 子命令和 `status` 子命令指定状况列表。

此选项将会限制输出，以便只包括在节点列表中的一个或多个节点上处于指定状况之一的资源。

可能的状况如下：

- degraded
- detached
- faulted
- monitor_failed
- not_online – 指定除 `online` 或 `online_not_monitored` 以外的任何状况
- offline
- online
- online_not_monitored
- start_failed
- stop_failed
- unknown
- unmonitored
- wait

-V

--version

显示该命令的版本。

指定此选项时，不要将其与子命令、操作数或其他选项结合使用。子命令、操作数或其他选项将被忽略。-V 选项只显示命令的版本。不会执行任何其他操作。

-v

--verbose

将详细消息显示到标准输出。

可以在任何格式的此命令中指定该选项。

请不要同时指定 -v 选项和 -o - 选项。否则 -v 选项将被忽略。-o - 选项会抑制其他所有标准输出。

-z {zoneclustername | global | all}
--zoneclustername={zoneclustername | global | all}
--zoneclustername {zoneclustername | global | all}

指定一个或多个存在资源且您要对其执行操作的群集。

除 export 子命令以外的所有子命令均支持此选项。

如果指定此选项，还必须指定以下列表中的参数之一：

zoneclustername	指定使用此选项的命令仅对名为 zoneclustername 的区域群集中的所有指定资源起作用。
global	指定使用此选项的命令仅对全局群集中的所有指定资源进行操作。
all	如果在全局群集中使用此参数，则会指定使用此参数的命令对所有群集（包括全局群集和所有区域群集）中的所有指定资源起作用。 如果在某区域群集中使用此参数，则会指定使用此参数的命令仅对该区域群集中的所有指定资源起作用。

支持以下操作数：

resource 指定应接受 Oracle Solaris Cluster 资源名称作为操作数。如果子命令接受多个资源，则可以使用加号 (+) 指定所有逻辑主机名资源。

如果命令对于所有指定的操作数均成功执行，则返回零 (CL_NOERR)。如果某个操作数出错，命令会继续处理操作数列表中的下一个操作数。返回的退出代码始终反映最先发生的错误。

此命令返回下列退出状态代码：

0 CL_NOERR

没有错误

您发出的命令已成功完成。

1 CL_ENOMEM

交换空间不足

某个群集节点耗尽了交换内存或其他操作系统资源。

3 CL_EINVAL

参数无效

您键入的命令不正确，或者使用 -i 选项提供的群集配置信息语法不正确。

6 CL_EACCESS

权限被拒绝

指定的对象不可访问。要发出命令，可能需要拥有超级用户或 RBAC 访问权限。有关更多信息，请参见 [Unresolved link to " su1M"](#) 和 [Unresolved link to " rbac5"](#) 手册页。

9 CL_ESTATE

对象处于错误状况

您尝试修改一个在特定时间无法修改，或任何时候均无法修改的属性、资源组或其他对象。

10 CL_EMETHOD

资源方法失败

资源方法失败。方法因以下某个原因而失败：

- 尝试创建资源或修改资源属性时，`validate` 方法失败。
- 尝试启用、禁用或删除资源时，`validate` 以外的其他方法失败。

15 CL_EPROP

属性无效

您使用 `-p`、`-y` 或 `-x` 选项指定的属性或值不存在或不被允许。

35 CL_EIO

I/O 错误

发生了物理输入/输出错误。

36 CL_ENOENT

没有这样的对象

由于以下某个原因，找不到您所指定的对象：

- 该对象不存在。
- 您尝试使用 `-o` 选项创建的配置文件路径中的某个目录不存在。
- 您尝试使用 `-i` 选项访问的配置文件包含错误。

39 CL_EEXIST

对象已存在

您所指定的设备、设备组、群集互连组件、节点、群集、资源、资源类型、资源组或专用字符串已存在。

这些退出值与 [scha_calls\(3HA\) \[897\]](#) 手册页中描述的返回码兼容。

例 200 创建逻辑主机名资源

此命令将在名为 `rg-failover` 的资源组内创建名为 `logicalhost1` 的资源。创建的资源处于启用状况，并启用了监视功能。

```
# clreslogicalhostname create -g rg-failover logicalhost1
```

以下两个命令中的任何一个都将在区域群集 ZC 中创建一个名为 logicalhost1 的资源。可以从全局群集节点或区域群集 ZC 执行这些命令。如果从区域群集执行命令，则使用区域群集名称显式定义资源范围的操作是可选的。

```
# clreslogicalhostname create -g rg-failover -Z ZC logicalhost1
```

```
# clreslogicalhostname create -g rg-failover ZC:logicalhost1
```

例 201 使用不同的逻辑主机名创建逻辑主机名资源

此命令将在名为 rg-failover 的资源组中创建一个名为 rs-logicalhost1 的资源。

逻辑主机名与资源名称不相同，但逻辑主机的名称和 IP 地址保持相同。

```
# clreslogicalhostname create -g rg-failover \  
-h logicalhost1 rs-logicalhost1
```

例 202 为逻辑主机名资源指定 IPMP 组

此命令将为 logicalhost1 资源设置 IPMP 组。

```
# clreslogicalhostname create -g rg-failover \  
-N ipmp0@black,ipmp0@white logicalhost1
```

例 203 删除逻辑主机名资源

此命令可删除名为 logicalhost1 的资源。

```
# clreslogicalhostname delete logicalhost1
```

例 204 列出逻辑主机名资源

此命令将列出所有逻辑主机名资源。

```
# clreslogicalhostname list  
logicalhost1  
logicalhost2
```

例 205 列出逻辑主机名资源及其资源组和资源类型

此命令将列出所有逻辑主机名资源及其资源组和资源类型。

```
# clreslogicalhostname list -v  
Resources      Resource Groups Resource Types  
-----  
logicalhost1   rg-failover-1   SUNW.LogicalHostname  
logicalhost2   rg-failover-2   SUNW.LogicalHostname
```

例 206 列出逻辑主机名资源的扩展属性

此命令将列出所有逻辑主机名资源的扩展属性。

```
# clreslogicalhostname list-props -v
Properties      Descriptions
-----
NetIfList      List of IPMP groups on each node
HostnameList   List of hostnames this resource manages
CheckNameService Name service check flag
```

有关以下属性的描述，请参见 [Unresolved link to " attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

[Intro\(1CL\) \[17\]](#)、[cluster\(1CL\) \[461\]](#)、[clresource\(1CL\) \[225\]](#)、[clressharedaddress\(1CL\) \[289\]](#)、[clresource\(1CL\) \[225\]](#)、[link to " rbac5"](#)、[r_properties\(5\) \[1103\]](#)

超级用户可以运行此命令的所有格式。

任何用户都可以运行带有以下选项的此命令：

- -? 选项
- -V 选项

要运行带有子命令的此命令，超级用户以外的用户需要具有 RBAC 授权。请参见下表。

子命令	RBAC 授权
create	solaris.cluster.modify
delete	solaris.cluster.modify
disable	solaris.cluster.admin
enable	solaris.cluster.admin
export	solaris.cluster.read
list	solaris.cluster.read
list-props	solaris.cluster.read
monitor	solaris.cluster.admin
reset	solaris.cluster.admin
set	solaris.cluster.modify

子命令	RBAC 授权
show	solaris.cluster.read
status	solaris.cluster.read
unmonitor	solaris.cluster.admin

名称

clressharedaddress, clrssa — 管理 Oracle Solaris Cluster 共享地址的资源

```
/usr/cluster/bin/clressharedaddress [subcommand] -?  
  
/usr/cluster/bin/clressharedaddress -V  
  
/usr/cluster/bin/clressharedaddress [subcommand [options]] -v  
[saresource]...  
  
/usr/cluster/bin/clressharedaddress create -g resourcegroup [-h  
  lhost[,...]] [-N netif@node[,...]] [-X node[,...]]  
  [-p name=value] [-Z {zoneclustername | global}] [-d saresource  
  
/usr/cluster/bin/clressharedaddress create -i  
  {- | clconfiguration} [-a] [-g resourcegroup[,...]] [-X  
  node[,...]] [-p name=value] [-d] {+ | saresource...}  
  
/usr/cluster/bin/clressharedaddress delete [-g resourcegroup[,...]]  
  [-Z {zoneclustername | global}] [-F] {+ | saresource...}  
  
/usr/cluster/bin/clressharedaddress disable [-g resourcegroup[,...]]  
  [-R] [-n node[,...]] [-Z  
  {zoneclustername | global}] {+ | saresource...}  
  
/usr/cluster/bin/clressharedaddress enable [-g resourcegroup[,...]]  
  [-R] [-n node[,...]] [-Z  
  {zoneclustername | global}] {+ | saresource...}  
  
/usr/cluster/bin/clressharedaddress export [-o {- | configfile}]  
  {+ | saresource...}  
  
/usr/cluster/bin/clressharedaddress list [-s state[,...]]  
  [-g resourcegroup[,...]] [-Z {zoneclustername  
  [,...] | global | all}] {+ | saresource...}  
  
/usr/cluster/bin/clressharedaddress list-props [-l listtype] [-p  
  name[,...]] [-Z {zoneclustername [,...] | global | all}]  
  {+ | lresource...}  
  
/usr/cluster/bin/clressharedaddress monitor [-g resourcegroup[,...]]  
  [-Z {zoneclustername | global}] {+ | saresource...}  
  
/usr/cluster/bin/clressharedaddress reset [-f errorflag] [-g  
  resourcegroup[,...]] [-Z {zoneclustername | global}]  
  {+ | saresource...}  
  
/usr/cluster/bin/clressharedaddress set [-i {- | clconfiguration}]  
  [-g resourcegroup[,...]] [-X node[,...]] [-p  
  name[+|-]=value] [-Z {zoneclustername | global}] {+ | saresource...}  
  
/usr/cluster/bin/clressharedaddress show [-g resourcegroup[,...]]
```

```
[-p name[,...]] [-Z {zoneclustername [,...] | global | all}]
[+ | saresource...]

/usr/cluster/bin/clressharedaddress status [-s state[,...]]
[ -n node[,...]] [-g resourcegroup[,...]]
[-Z {zoneclustername [,...] | global | all}] [+ | saresource...]

/usr/cluster/bin/clressharedaddress unmonitor [-g resourcegroup[,...]]
[-Z {zoneclustername | global}] {+ | saresource...}
```

`clressharedaddress` 命令用于管理 Oracle Solaris Cluster 共享地址的资源。`clrssa` 命令是 `clressharedaddress` 命令的简短格式。`clressharedaddress` 命令和 `clrssa` 命令是相同的。您可以使用此命令的任一种格式。

还可以使用 [clresource\(1CL\) \[225\]](#) 命令管理共享地址的资源。

`clressharedaddress` 命令的一些子命令可修改资源配置。可以从全局群集或区域群集使用这些子命令。以下子命令可修改资源配置：

- `disable`
- `enable`
- `monitor`
- `reset`
- `set`
- `unmonitor`

`clressharedaddress` 命令的某些子命令只获取有关资源的信息。

- `export`
- `list`
- `list-props`
- `show`
- `status`

为了避免此命令产生不可预测的结果，请从全局群集节点运行该命令的所有格式。

此命令的一般格式为：

```
clressharedaddress [subcommand] [options] [operands]
```

仅当 `options` 指定选项 `-?` 或 `-v` 时，才可以省略 `subcommand`。

此命令的每个选项都有长格式和简短格式两种格式。在本手册页的 `选项` 部分给出了每个选项的两种格式，以及对该选项的描述。

在区域群集中的操作

在区域群集中，可以将 `clressharedaddress` 命令与所有子命令（`export` 除外）一起使用。

也可以将 `-z` 选项与所有子命令（`export` 除外）一起使用来指定要将操作限制到的特定区域群集的名称。另外，还可以将区域群集名称附加到共享地址资源 (`zoneclustername` : `saresource`)，以便将操作限制到特定的区域群集。

您可以从全局群集节点访问所有区域群集信息，但无法从特定区域群集访问其他区域群集信息。如果不将操作限制到特定区域群集，则所用的子命令仅会在当前群集中运行。

支持以下子命令：

create

创建作为命令操作数指定的共享地址资源。

将 `create` 和 `-i` 选项一起使用以指定配置文件时，该子命令接受使用加号 (+) 作为操作数。使用 + 操作数时，将会创建配置文件中提供的尚不存在的所有资源。

使用 `create` 子命令之前，请确保 `/etc/netmasks` 文件具有所有逻辑主机名对应的 IP 地址子网和网络掩码条目。如有必要，请编辑 `/etc/netmasks` 文件以添加缺少的任何条目。

默认情况下，创建的资源处于启用状态，且启用了监视功能。但是，仅当资源的资源组联机后，该资源才会联机并受到监视。要在禁用状态下创建资源，请指定 `-d` 选项。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集在区域群集中创建共享地址资源，请使用 `-z` 选项指定区域群集的名称。

超级用户以外的用户需要具有 `solaris.cluster.modify` 基于角色的访问控制 (Role-Based Access Control, RBAC) 授权才能使用该子命令。

另请参见 `delete` 子命令的描述。

delete

删除作为命令操作数指定的共享地址资源。此子命令接受使用加号 (+) 作为操作数来指定删除所有资源。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集删除区域群集中的共享地址资源，请使用 `-z` 选项指定区域群集的名称。

`-g` 选项可以过滤操作数列表，以限制删除的资源。`-g` 选项只会删除操作数列表内属于 `resourcegrouplist` 中资源组的成员的资源。

- 默认情况下，仅当符合以下条件时，才会删除某个资源：
- 必须已禁用该资源。
- 必须已消除该资源的所有依赖性。

- 要确保删除所有指定的资源，请指定 -F 选项。-F 选项的效果如下所示：
- 删除所有指定的资源，即使是未禁用的资源。
- 从其他资源的资源依赖性设置中删除所有指定的资源。

将按照满足资源间依赖性所需的顺序禁用资源，而不管资源在命令行上的指定顺序。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

另请参见 `create` 子命令的描述。

`disable`

禁用作为命令操作数指定的共享地址资源。此子命令接受使用加号 (+) 作为操作数来指定禁用所有资源。

-g 选项可以过滤操作数列表，以限制禁用的资源。-g 选项只会禁用操作数列表内属于 *resourcegroup* 中资源组的成员的资源。

要确保满足所有必需的资源依赖性，请指定 -R 选项。-R 选项将会禁用依赖于指定为命令操作数的资源的任何资源，即使这些资源未指定为命令操作数。-g 选项和 -t 选项不适用于仅为满足资源依赖性而禁用的资源。

将按照满足资源间依赖性所需的顺序禁用资源，而不管资源在命令行上的指定顺序。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集禁用在区域群集中注册的共享地址资源，请使用 -z 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。

另请参见 `enable` 子命令的描述。

`enable`

启用作为命令操作数指定的共享地址资源。此子命令接受使用加号 (+) 作为操作数来指定启用所有资源。

-g 选项可以过滤操作数列表，以限制启用的资源。-g 选项仅启用操作数列表内属于 *resourcegroup* 中资源组的成员的资源。

要确保满足所有必需的资源依赖性，请指定 -R 选项。-R 选项将会启用依赖于指定为命令操作数的资源的任何资源，即使这些资源未指定为命令操作数。-g 选项不适用于仅为满足资源依赖性而启用的资源。

将按照满足资源间依赖性所需的顺序禁用资源，而不管资源在命令行上的指定顺序。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集启用在区域群集中注册的共享地址资源，请使用 -z 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。

另请参见 `disable` 子命令的描述。

export

以 `clconfiguration(5CL) [1245]` 手册页所述的格式导出共享地址资源配置。

只能在全局群集中使用该子命令。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

list

显示作为命令操作数指定的共享地址资源的列表。默认情况下，会显示所有资源。

`-g` 选项可以过滤操作数列表，以限制显示的资源。`-g` 选项只会显示操作数列表内属于 `resourcegroup` 中资源组的成员的资源。

此子命令接受使用加号 (+) 作为操作数来指定特定资源组中的所有资源，或者指定属于特定资源类型实例的所有资源。如果未提供操作数，则会显示指定资源组中的所有资源，或显示属于指定资源类型的实例的所有资源。

如果指定 `-v` 选项，还将显示列表中每个资源的资源组和资源类型。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集查看在区域群集中注册的共享地址资源，请使用 `-z` 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

list-props

显示作为命令操作数指定的共享地址资源的属性列表。默认情况下，会显示所有资源的扩展属性。

以下选项可以过滤操作数列表，以限制要显示其属性的资源：

`-g resourcegroup` 仅显示操作数列表中属于 `resourcegroup` 中资源组的共享地址资源的属性。

`-l` 选项指定要显示的资源属性的类型：

`-l all` 指定显示标准属性和扩展属性。

`-l extension` 指定只显示扩展属性。默认情况下，只显示扩展属性。

`-l standard` 指定只显示标准属性。

如果未指定 `-l` 选项，将只显示扩展属性，除非使用 `-p` 选项或 `-y` 选项明确指定标准属性。

-p 选项用于限制要显示的资源属性集。-p 选项只显示 *namelist* 中指定的属性。您可以在 *namelist* 中指定标准属性和扩展属性。

如果指定 -v 选项，还将显示每个属性的描述。

此子命令接受使用加号 (+) 作为操作数来指定特定资源组中的所有资源，或者指定属于特定资源类型实例的所有资源。如果未提供操作数，则会显示指定资源组中所有资源的属性，或显示属于指定资源类型的实例的所有资源的属性。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集查看区域群集中共享地址资源的属性列表，请使用 -z 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

monitor

打开对作为命令操作数指定的共享地址资源的监视。此子命令接受使用加号 (+) 作为操作数来指定开启对所有资源的监视。

-g 选项可以过滤操作数列表，以限制监视的资源。-g 选项只会监视操作数列表内属于 *resourcegroup* 中资源组的成员的资源。

开启对某个资源的监视后，仅当符合以下条件时，才会监视该资源：

- 已启用该资源。
- 包含该资源的资源组至少已在一个群集节点上联机。

注 - 开启对某个资源的监视不会启用该资源。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集监视区域群集中的资源，请使用 -z 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。

另请参见 `unmonitor` 子命令的描述。

reset

清除与作为命令操作数指定的共享地址资源相关联的错误标志。此子命令接受使用加号 (+) 作为操作数来指定清除所有资源的错误标志。

-g 选项可以过滤操作数列表，以限制重置的资源。-g 选项只会重置操作数列表内属于 *resourcegroup* 中资源组的成员的资源。

默认情况下，`reset` 子命令将会清除 `STOP_FAILED` 错误标志。要显式指定要清除的错误标志，请使用 -f 选项。-f 选项接受的唯一错误标志是 `STOP_FAILED` 错误标志。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集重置区域群集中的共享地址资源，请使用 -z 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。

set

修改作为命令操作数指定的共享地址资源的指定属性。此子命令接受使用加号 (+) 作为操作数来指定修改所有资源的特定属性。

-g 选项可以过滤操作数列表，以限制修改的资源。-g 选项只会修改操作数列表内属于 *resourcegroup* 中资源组的成员的资源。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集设置区域群集中共享地址资源的属性，请使用 -z 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

show

显示作为命令操作数指定的共享地址资源的配置。默认情况下，会显示所有资源的配置。

-g 选项可以过滤操作数列表，以限制要显示其配置的资源。-g 选项只会显示操作数列表内属于 *resourcegroup* 中资源组成员的资源的配置。

-p 选项用于限制要显示的资源属性集。-p 选项只显示 *namelist* 中指定的属性。您可以在 *namelist* 中指定标准属性和扩展属性。

此子命令接受使用加号 (+) 作为操作数来指定特定资源组中的所有资源，或者指定属于特定资源类型实例的所有资源。如果未提供操作数，则会显示指定资源组中所有资源的配置，或显示属于指定资源类型实例的所有资源的配置。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集查看区域群集中共享地址资源的配置，请使用 -z 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

status

显示作为命令操作数指定的共享地址资源的状态。默认情况下，会显示所有资源的状态。

以下选项可以过滤操作数列表，以限制要显示其状态的资源的列表：

- | | |
|-------------------------|--|
| -g <i>resourcegroup</i> | 仅显示操作数列表内属于 <i>resourcegroup</i> 中资源组成员的资源的状态。 |
| -n <i>nodelist</i> | 仅显示操作数列表内承载在 <i>nodelist</i> 中节点上的资源的状态。 |

`-s statelist` 仅显示操作数列表内处于 `statelist` 中的状况的资源的状态。

此子命令接受使用加号 (+) 作为操作数来指定特定资源组中的所有资源，或者指定属于特定资源类型实例的所有资源。如果未提供操作数，则会显示指定资源组中所有资源的状态，或显示属于指定资源类型实例的所有资源的状态。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集查看区域群集中共享地址资源的状态，请使用 `-z` 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

`unmonitor`

关闭对作为命令操作数指定的共享地址资源的监视。此子命令接受使用加号 (+) 作为操作数来指定关闭对所有资源的监视。

如果关闭对某个被禁用资源的监视，该资源不会受到影响。该资源及其监视器已脱机。

注 - 关闭对某个资源的监视不会禁用该资源。但是，当您禁用某个资源时，不需要关闭对该资源的监视。禁用的资源及其监视器将保持脱机状态。

`-g` 选项可以过滤操作数列表，以限制要对其关闭监视的资源。`-g` 选项将对操作数列表内属于 `resourcegroup` 中资源组的成员的资源关闭监视。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集关闭对区域群集中共享地址资源的监视，请使用 `-z` 选项指定区域群集名称。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。

另请参见 `disable` 子命令和 `monitor` 子命令的描述。

支持以下选项：

`-?`

`--help`

显示帮助信息。使用该选项时，不会执行任何其他处理。

指定此选项时，子命令可有可无。

如果仅指定此选项而不使用子命令，将会显示此命令的子命令列表。

如果使用子命令且指定此选项，将会显示该子命令的用法选项。

此选项与特定子命令一起使用的效果如下所示：

`create`

与 `-g` 选项一起指定时，此选项将会显示指定资源组的所有资源属性的帮助信息。

`set` 显示指定为命令操作数的资源的属性的帮助信息。

`-a`

`--automatic`

当基于群集配置信息创建资源时，将自动执行以下附加操作：

- 注册资源类型
- 创建资源组
- 创建操作数列表中指定的资源所依赖的资源

群集配置信息包含的信息必须足以执行以下所有操作：

- 启用要注册的资源类型
- 启用要创建的资源组
- 启用要创建的资源

此选项仅可以与 `create` 子命令一同指定。如果指定此选项，还必须指定 `-i` 选项并提供配置文件。

`-d`

`--disable`

创建资源后禁用该资源。此选项仅可以与 `create` 子命令一同指定。默认情况下，资源在启用状态下创建。

启用某个资源并不保证该资源联机。仅当资源所在的资源组已在至少一个节点上联机后，该资源才会联机。

`-f errorflag`

`--flag errorflag`

显式指定要通过 `reset` 子命令清除的错误标志。此选项仅可以与 `reset` 子命令一同指定。默认情况下，`reset` 子命令将会清除 `STOP_FAILED` 错误标志。

- `f` 选项接受的唯一错误标志是 `STOP_FAILED` 错误标志。

`-F`

`--force`

强制删除未禁用的资源。此选项仅可以与 `delete` 子命令一同指定。

`-g resourcegroup[,...]`

`--resourcegroup resourcegroup[,...]`

指定一个资源组或资源组列表。

对于除 `create` 以外的子命令，该命令仅对操作数列表内属于 `-g` 选项指定的资源组成员的资源起作用。

此选项与特定子命令一起使用的效果如下所示：

`create`

指定在特定资源组中创建资源。将 `-g` 与 `create` 子命令一起使用时，只能指定一个资源组。

`-h /host[,...]`

`--logicalhost /host[,...]`

指定主机名列表。无论是在多个逻辑主机需要与新的 SharedAddress 资源相关联时，还是在逻辑主机与资源本身具有不同的名称时，都必须使用 `-h` 选项。SharedAddress 资源的 HostnameList 内的所有逻辑主机必须位于同一子网上。如果未指定 HostnameList 属性，HostnameList 将与 SharedAddress 资源相同。

SharedAddress 资源的逻辑主机名必须位于同一子网上。

您可以使用 `-h` 选项，而不是通过 `-p` 设置 HostnameList 属性；但是，无法在同一命令中使用 `-h` 并明确设置 HostnameList。

`-h` 选项只能与 `create` 子命令一起使用。

`-i {- | clconfiguration}`

`--input {- | clconfiguration}`

指定用于创建或修改共享地址资源的配置信息。此信息必须符合 [clconfiguration\(5CL\) \[1245\]](#) 手册页中定义的格式。此信息可包含在文件中或者通过标准输入提供。要指定标准输入，请指定 `-` 而不是文件名。

仅会创建或修改作为命令操作数提供的资源。在命令中指定的选项将会覆盖配置信息中设置的任何选项。如果配置信息中缺少配置参数，则必须在命令行中指定这些参数。

此选项与特定子命令一起使用的效果如下所示：

<code>create</code>	与 <code>-a</code> 选项一起指定时，此选项将注册所有必需的资源类型，并创建所有必需的资源组。您必须提供注册和配置所需的全部信息。所有其他配置数据将被忽略。
---------------------	---

`-l listtype`

`--listtype listtype`

指定要通过 `list-props` 子命令显示的资源属性的类型。此选项仅可以与 `list-props` 子命令一同指定。

必须根据以下列表为 `listtype` 指定一个值：

<code>all</code>	指定显示标准属性和扩展属性。
------------------	----------------

<code>extension</code>	指定只显示扩展属性。默认情况下，只显示扩展属性。
------------------------	--------------------------

<code>standard</code>	指定只显示标准属性。
-----------------------	------------

如果未指定 `-l` 选项，则只显示扩展属性，除非使用 `-p` 选项明确指定标准属性。

`-n node[...]`

`--node node[...]`

指定目标全局群集或区域群集中的一个节点或一系列节点。可以使用节点名称或节点 ID 指定每个节点。

如果指定 `-z` 选项，则仅可以使用 `-n` 选项指定区域群集主机名，而不能指定全局群集主机名。如果没有指定 `-z` 选项，则仅可以使用 `-n` 选项指定全局群集主机名。

可以与此选项一起指定的子命令如下所述：

<code>disable</code>	只禁用操作数列表中承载在指定节点上的资源。
<code>enable</code>	只启用操作数列表中承载在指定节点上的资源。
<code>status</code>	仅报告操作数列表中承载在指定节点上的资源的状态。

`-N netif@node[...]`

`--netiflist netif@node[...]`

指定一个资源属性。通过 `-N` 选项，可以设置 `NetIfList` 属性，而不必使用 `-p` 选项来指定该属性。如果不指定 `-N`，`clressharedaddress` 命令将尝试基于可用 IPMP 组或公共适配器以及与 `HostnameList` 属性关联的子网来设置 `NetIfList` 属性。

可以采用 `ipmpgroup@node[...]` 格式指定 `NetIfList` 属性。但是，`-N` 接受 `ipmpgroup@ node[...]` 和 `publicNIC@ node[...]` 两种格式。如果未使用 `-N`，或者将该选项与 `publicNIC@node` 一起使用，则 `clressharedaddress` 命令将尝试创建必要的 IPMP 组。系统将创建一组包含一个或多个单适配器的 IPMP 组，以及一组默认以后使用标准 Oracle Solaris 接口修改为包括多个适配器的 IPMP 组。

您可以使用 `-N` 选项，而不是通过 `-p` 直接设置 `NetIfList` 属性；但是，无法在同一命令中使用 `-N` 并明确设置 `NetIfList`。

`-N` 只能与 `create` 子命令一起使用。

`-o { - | clconfiguration }`

`--output { - | clconfiguration }`

指定要将资源配置信息写入到的位置。此位置可以是某个文件或标准输出。要指定标准输出，请指定 `-` 而不是文件名。如果指定标准输出，将抑制命令的所有其他标准输出。此选项仅可以与 `export` 子命令一同指定。

只会写入作为命令操作数提供的资源的配置信息。信息是采用 [clconfiguration\(5CL\) \[1245\]](#) 手册页中定义的格式写入的。

```
-p name=value
-p name+=array-values
-p name-=array-values
--property name=value
--property name+=array-values
--property name-=array-values
```

设置资源的标准属性和扩展属性。此选项仅可以与 `create` 子命令和 `set` 子命令一同指定。

有关标准属性的描述，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

有关某个资源类型的扩展属性的描述，请参见该资源类型的文档。

与此选项一起使用的运算符如下所述：

=	将属性设置为指定值。 <code>create</code> 子命令和 <code>set</code> 子命令接受此运算符。
+=	将一个或多个值添加到某个字符串数组值。只有 <code>set</code> 子命令接受此运算符。只能为字符串数组值指定此运算符。
-=	从某个字符串数组值中删除一个或多个值。只有 <code>set</code> 子命令接受此运算符。只能为字符串数组值指定此运算符。

如果只需在群集节点的子集上设置每节点属性，请通过在属性名称后面附加节点列表（用大括号括住）来指定要设置该属性的节点，如下所示：

```
name{nodelist}
```

`nodelist` 是以逗号分隔的节点名称或节点 ID 的列表。有关每节点属性的更多信息，请参见 [rt_properties\(5\) \[1143\]](#) 手册页。

```
-p name[,...]
--property name[,...]
```

为 `list-props` 子命令和 `show` 子命令指定属性列表。

可将此选项用于资源的标准属性和扩展属性。

有关标准属性的描述，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

有关某个资源类型的扩展属性的描述，请参见该资源类型的文档。

如果未指定此选项，`list-props` 子命令和 `show` 子命令将会列出所有资源属性或大多数资源属性，具体取决于是否同时指定了 `-v` 选项。

```
-R
--recursive
```

以递归方式启用或禁用资源，以确保满足所有必需的依赖性。此选项仅可以与 `disable` 子命令和 `enable` 子命令一同指定。

此选项与这些子命令一起使用的效果如下所示：

disable	禁用依赖于指定为命令操作数的资源的任何资源，即使这些资源未指定为命令操作数。
enable	启用指定为命令操作数的资源所依赖的任何资源，即使这些资源未指定为命令操作数。

`-s state[...]`

`--state state[...]`

为 `list` 子命令和 `status` 子命令指定状况列表。

此选项将会限制输出，以便只包括在节点列表中的一个或多个节点上处于指定状况之一的资源。

可能的状况如下：

- degraded
- detached
- faulted
- monitor_failed
- not_online - 指定除 `online` 或 `online_not_monitored` 以外的任何状况
- offline
- online
- online_not_monitored
- start_failed
- stop_failed
- unknown
- unmonitored
- wait

`-V`

`--version`

显示该命令的版本。

指定此选项时，不要将其与子命令、操作数或其他选项结合使用。子命令、操作数或其他选项将被忽略。`-V` 选项只显示命令的版本。不会执行任何其他操作。

`-v`

`--verbose`

将详细消息显示到标准输出。

可以在任何格式的此命令中指定该选项。

请不要同时指定 `-v` 选项和 `-o` 选项。否则 `-v` 选项将被忽略。`-o` 选项会抑制其他所有标准输出。

-X *node*[...]

--auxnode *node*[...]

设置 AuxNodeListSharedAddress 资源属性。

AuxNodeList 列表中的节点可以托管与共享地址资源关联的逻辑主机集。但是，这些节点无法在故障转移期间充当主节点。

-Z {*zoneclustername* | global | all}

--zoneclustername={*zoneclustername* | global | all}

--zoneclustername {*zoneclustername* | global | all}

指定一个或多个存在资源且您要对其执行操作的群集。

除 export 子命令以外的所有子命令均支持此选项。

如果指定此选项，还必须指定以下列表中的参数之一：

zoneclustername 指定使用此选项的命令仅对名为 *zoneclustername* 的区域群集中的所有指定资源起作用。

global 指定使用此选项的命令仅对全局群集中的所有指定资源进行操作。

all 如果在全局群集中使用此参数，则会指定使用此参数的命令对所有群集（包括全局群集和所有区域群集）中的所有指定资源起作用。

如果在某区域群集中使用此参数，则会指定使用此参数的命令仅对该区域群集中的所有指定资源起作用。

支持以下操作数：

resource 指定应接受 Oracle Solaris Cluster 资源名称作为操作数。如果子命令接受多个资源，则可以使用加号 (+) 指定所有共享地址资源。

如果命令对于所有指定的操作数均成功执行，则返回零 (CL_NOERR)。如果某个操作数出错，命令会继续处理操作数列表中的下一个操作数。返回的退出代码始终反映最先发生的错误。

此命令返回下列退出状态代码：

0 CL_NOERR

没有错误

您发出的命令已成功完成。

1 CL_ENOMEM

交换空间不足

某个群集节点耗尽了交换内存或其他操作系统资源。

3 CL_EINVAL

参数无效

您键入的命令不正确，或者使用 `-i` 选项提供的群集配置信息语法不正确。

6 CL_EACCESS

权限被拒绝

指定的对象不可访问。要发出命令，可能需要拥有超级用户或 RBAC 访问权限。有关更多信息，请参见 [Unresolved link to "su1M"](#) 和 [Unresolved link to "rbac5"](#) 手册页。

9 CL_ESTATE

对象处于错误状况

您尝试修改一个在特定时间无法修改，或任何时候均无法修改的属性、资源组或其他对象。

10 CL_EMETHOD

资源方法失败

资源方法失败。方法因以下某个原因而失败：

- 尝试创建资源或修改资源属性时，`validate` 方法失败。
- 尝试启用、禁用或删除资源时，`validate` 以外的其他方法失败。

15 CL_EPROP

属性无效

您使用 `-p`、`-y` 或 `-x` 选项指定的属性或值不存在或不被允许。

35 CL_EIO

I/O 错误

发生了物理输入/输出错误。

36 CL_ENOENT

没有这样的对象

由于以下某个原因，找不到您所指定的对象：

- 该对象不存在。
- 您尝试使用 `-o` 选项创建的配置文件路径中的某个目录不存在。
- 您尝试使用 `-i` 选项访问的配置文件包含错误。

39 CL_EEXIST

对象已存在

您所指定的设备、设备组、群集互连组件、节点、群集、资源、资源类型、资源组或专用字符串已存在。

这些退出值与 [scha_calls\(3HA\) \[897\]](#) 手册页中描述的返回码兼容。

例 207 创建共享地址资源

此命令将在名为 `rg-failover` 的资源组内创建名为 `sharedhost1` 的资源。创建的资源处于启用状况，并启用了监视功能。

```
# clressharedaddress create -g rg-failover sharedhost1
```

以下两个命令均会在区域群集 `ZC` 中创建名为 `sharedhost1` 的资源。可在全局群集节点中或区域群集 `ZC` 内执行这些命令。

```
# clressharedaddress create -g rg-failover -Z ZC sharedhost1
```

```
# clressharedaddress create -g rg-failover ZC:sharedhost1
```

例 208 使用其他逻辑主机名创建共享地址资源

此命令将在名为 `rg-failover` 的资源组内创建名为 `rs-sharedhost1` 的资源。

该逻辑主机名与资源名称不同，但逻辑主机的名称和 IP 地址保持不变。

```
# clressharedaddress create -g rg-failover \  
-h sharedhost1 rs-sharedhost1
```

例 209 指定共享地址资源的 IPMP 组

此命令可设置 `sharedhost1` 资源的 IPMP 组。

```
# clressharedaddress create -g rg-failover \  
-N ipmp0@black,ipmp0@white sharedhost1
```

例 210 删除共享地址资源

此命令可删除名为 `sharedhost1` 的资源。

```
# clressharedaddress delete sharedhost1
```

例 211 列出共享地址资源

此命令可列出所有共享地址资源。

```
# clressharedaddress list  
sharedhost1  
sharedhost2
```


例 212 列出共享地址资源及其资源组和资源类型

此命令可列出所有共享地址资源及其资源组和资源类型。

```
# clressharedaddress list -v
Resources  Resource Groups Resource Types
-----
sharedhost1 rg-failover-1 SUNW.SharedAddress
sharedhost2 rg-failover-2 SUNW.SharedAddress
```

例 213 列出共享地址资源的扩展属性

此命令可列出所有共享地址资源的扩展属性。

```
# clressharedaddress list-props -v
Properties      Descriptions
-----
NetIfList      List of IPMP groups on each node
AuxNodeList    List of nodes on which this resource is available
HostnameList   List of hostnames this resource manages
CheckNameService Name service check flag
```

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

[Intro\(1CL\) \[17\]](#), [cluster\(1CL\) \[461\]](#), [clresource\(1CL\) \[225\]](#),
[clreslogicalhostname\(1CL\) \[207\]](#), [clresourcegroup\(1CL\) \[253\]](#),
[clresourcetype\(1CL\) \[277\]](#), [scha_calls\(3HA\) \[897\]](#), [clconfiguration\(5CL\) \[1245\]](#),
[Unresolved link to "rbac5"](#), [r_properties\(5\) \[1103\]](#)

超级用户可以运行此命令的所有格式。

任何用户都可以运行带有以下选项的此命令：

- -? 选项
- -v 选项

要运行带有子命令的此命令，超级用户以外的用户需要具有 RBAC 授权。请参见下表。

子命令	RBAC 授权
create	solaris.cluster.modify

子命令	RBAC 授权
delete	solaris.cluster.modify
disable	solaris.cluster.admin
enable	solaris.cluster.admin
export	solaris.cluster.read
list	solaris.cluster.read
list-props	solaris.cluster.read
monitor	solaris.cluster.admin
reset	solaris.cluster.admin
set	solaris.cluster.modify
show	solaris.cluster.read
status	solaris.cluster.read
unmonitor	solaris.cluster.admin

名称

clresourcetype, clrt — 管理 Oracle Solaris Cluster 数据服务的资源类型

```
/usr/cluster/bin/clresourcetype [subcommand -?]  
  
/usr/cluster/bin/clresourcetype subcommand -v [options]  
[-Z {zoneclustername | global}] {+ | resourcetype}...  
  
/usr/cluster/bin/clresourcetype add-node -n node[,...]  
[-Z {zoneclustername | global}] {+ | resourcetype}  
  
/usr/cluster/bin/clresourcetype export [-o {- | configfile}]  
{+ | resourcetype}  
  
/usr/cluster/bin/clresourcetype list [ -n node[,...]  
[-Z {zoneclustername[,...] | global | all}] {+ | resourcetype...}  
  
/usr/cluster/bin/clresourcetype list-props [-p [name,...]] [-Z  
{zoneclustername[,...] | global | all}] {+ | resourcetype...}  
  
/usr/cluster/bin/clresourcetype register [-i  
{- | clconfiguration}] [ {-n node  
[,...] | -N}] [-f rtrfile] [-p [name [+ | -]=value,...]]  
[-Z {zoneclustername | global}] {+ | resourcetype...}  
  
/usr/cluster/bin/clresourcetype remove-node -n node  
[,...] [-Z {zoneclustername | global}]  
{+ | resourcetype...}  
  
/usr/cluster/bin/clresourcetype set [-n node  
[,...] | -N] [-p [name [+ | -]=value,...]] [-Z  
{zoneclustername | global}] {+ | resourcetype...}  
  
/usr/cluster/bin/clresourcetype show [-n node[,...]  
[-Z {zoneclustername[,...] | global | all}] {+ | resourcetype...}  
  
/usr/cluster/bin/clresourcetype unregister [-Z {zoneclustername |  
global}] {+ | resourcetype...}
```

clresourcetype 命令用于管理 Oracle Solaris Cluster 数据服务的资源类型。clrt 命令是 clresourcetype 命令的简短格式。clresourcetype 命令和 clrt 命令是相同的。您可以使用此命令的任一种格式。

为便于管理，请从全局群集节点运行此命令。

在区域群集中，可以将 clresourcetype 命令与所有子命令（export 除外）一起使用。

也可以将 -z 选项与所有子命令（export 除外）一起使用来指定要将操作限制到的特定区域群集的名称。另外，还可以将区域群集名称附加到资源类型名称 (zoneclustername : resourcetype)，以便将操作限制到特定的区域群集。

您可以从全局群集节点访问所有区域群集信息，但无法从特定区域群集访问其他区域群集信息。如果不将操作限制到特定区域群集，则所用的子命令仅会在当前群集中运行。

此命令的一般格式如下所示：

```
clresourcetype [subcommand] [options] [operands]
```

仅当 *options* 指定 *-?* 选项或 *-v* 选项时，才可以省略 *subcommand*。

此命令的每个选项都有长格式和简短格式两种格式。在本手册页的 [选项](#) 部分给出了每个选项的两种格式，以及对该选项的描述。

支持以下子命令：

add-node

将指定的节点添加到节点列表（针对指定为命令操作数的资源类型）。

您可以在全局群集中或区域群集中使用该子命令。

从全局群集节点使用 `add-node` 命令时，可以使用 `-z` 选项指定区域群集的名称。

此子命令接受使用加号 (+) 作为操作数来指定所有资源类型。

超级用户以外的用户需要具有 `solaris.cluster.modify` 基于角色的访问控制 (role-based access control, RBAC) 授权才能使用此子命令。

另请参见 `remove-node` 子命令的描述。

export

以 [clconfiguration\(5CL\) \[1245\]](#) 手册页所述的格式导出群集资源类型配置。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

只能在全局群集中使用该子命令。

list

显示作为命令操作数指定的资源类型的列表。默认情况下，将显示已在群集中注册的所有资源类型。此子命令接受使用加号 (+) 作为操作数来指定已在群集中注册的所有资源类型。

您可以在全局群集中或区域群集中使用该子命令。

在全局群集节点中，此子命令仅显示已在全局群集节点中注册的资源类型。要从全局群集查看已在区域群集中注册的资源类型，可以使用 `-z` 选项指定区域群集。

如果指定 `-nodelist` 选项，将仅显示已注册用于 `nodelist` 中的节点上的资源类型。

如果指定 `-v` 选项，还将显示列表中每种资源类型的节点列表。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

list-props

显示指定资源类型的属性。此子命令接受使用加号 (+) 作为操作数来指定已在群集中注册的所有资源类型。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集查看已在区域群集中注册的资源类型属性，可以使用 -z 选项指定区域群集。

-p 选项可以限制要显示的属性集。

如果指定 -v 选项，还将显示每个属性的描述。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

register

注册作为命令操作数指定的资源类型。必须先注册资源类型，然后才能创建该类型的资源。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集向区域群集注册资源类型，请使用 -z 选项指定区域群集名称。

必须将用于定义每种资源类型的数据服务安装在要使用该资源类型的每个节点上。如果数据服务仅安装在群集节点子集上，请使用 -n *nodelist* 选项指定节点子集。如果资源类型将在群集中所有节点上可用，请指定 -N 选项。在使用 -N 选项时，资源类型还可用于以后可能添加到群集中的任何节点。省略 -N 选项和 -nnodelist 选项相当于指定 -N 选项。要明确指定属性名称，请使用 -p `Installed_nodes= nodelist` 选项。

有关已向群集注册的资源类型的信息，可从用于定义资源类型的资源类型注册 (Resource Type Registration, RTR) 文件获取。RTR 文件的位置和名称通常遵循以下约定：

- RTR 文件通常位于 `/opt/cluster/lib/rgm/rtreg` 目录中。
- RTR 文件的名称通常与资源类型的名称相匹配。

Oracle 提供的所有 RTR 文件的位置和文件名称都遵循这些约定。例如，用于定义 `SUNW.nfs` 资源类型的 RTR 文件包含在文件 `/opt/cluster/lib/rgm/rtreg/SUNW.nfs` 中。

如果 RTR 文件不符合这些约定，必须指定 - *rttrfile* 选项。

这些约定同样适用于从区域群集注册的资源类型。当用户为区域群集注册资源类型时，RTR 文件必须驻留在区域群集 *zonepath* 中。无法在区域群集 *zonepath* 界限之外注册 RTR 文件。在为区域群集注册 *Global_zone* 属性设置为 TRUE 的资源类型时，RTR 文件必须驻留在 `/opt/cluster/lib/rgm/rtreg` 或 `/usr/cluster/lib/rgm/rtreg` 目录中的全局群集节点内。如果指定这些位置之外的任何位置，资源类型将无法注册。



注意 - 不要注册 `Global_zone` 属性设置为 `TRUE` 的资源类型，除非该资源类型来自于已知的可信源。此属性设置为 `TRUE` 的资源类型会规避区域隔离并引起风险。

此子命令接受使用加号 (+) 作为操作数来指定尚未注册的所有资源类型。可用资源类型的完整列表按如下所示进行确定：

- 如果指定 `-iclconfiguration` 选项，`iclconfiguration` 将定义可用资源类型的完整列表。
- 如果没有指定 `-i` 选项，则可用资源类型的完整列表中仅包含由 Oracle 提供的资源类型。这些资源类型还必须已在节点列表中的所有节点上安装。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

另请参见 `unregister` 子命令的描述。

`remove-node`

从节点列表（操作对象列表中的资源类型均已向这些节点注册）中删除一个节点。此子命令接受使用加号 (+) 作为操作数来指定已在群集中注册的所有资源类型。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集删除区域群集的资源类型，请使用 `-z` 选项指定区域群集名称。

仅可对已为群集中的部分节点而不是所有节点注册的资源类型使用该子命令。因此，如果您在以下情况中使用此子命令，将会发生错误：

- 操作数列表中的资源类型已针对群集中的所有节点进行注册。有关为群集中所有节点注册资源类型的信息，请参见 `-N` 选项的描述。
- 操作数列表中资源类型的 `Installed_nodes` 属性尚未指定群集中节点的子集。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。

另请参见 `add-node` 子命令的描述。

`set`

设置作为命令操作数指定的资源类型的属性。此子命令接受使用加号 (+) 作为操作数来指定已在群集中注册的所有资源类型。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集设置区域群集中资源类型的属性，请使用 `-z` 选项指定区域群集名称。

仅可以设置 `rt_properties(5)` [1143] 手册页中定义为 Tunable Any Time 的资源类型属性。

- 可以通过指定 `-nodelist` 选项而不指定 `-p` 选项来修改 `Installed_Nodes` 属性。或者，可以通过使用 `-pInstalled_Nodes= nodelist` 选项明确指定属性名称。
- 对于定义为 Tunable Any Time 的所有其他属性，必须通过使用 `-pproperty = value` 选项明确指定属性名称。

要限制资源类型在其上可用的节点的列表，请指定 `-nodelist` 选项。相反，要指定资源类型在群集中所有节点上可用，请指定 `-N` 选项。在使用 `-N` 选项时，资源类型还可用于以后可能添加到群集中的任何节点。必须指定 `-n` 选项或 `-N` 选项。如果省略这两个选项，该子命令不会更改任何配置信息。

show

显示已在群集中注册的资源类型的相关信息。默认情况下，将显示已注册的所有资源类型的以下信息：

- 与每个资源类型关联的属性的列表
- 用于定义这些属性的参数

如果指定 `-nodelist` 选项，将仅显示已注册用于 `nodelist` 中的节点上的资源类型。

如果指定 `-v` 选项，还将显示每种资源类型的以下信息：

- 为资源类型定义的方法
- 每种方法的超时参数

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集查看已在区域群集中注册的资源类型，可以使用 `-z` 选项指定区域群集名称。

此子命令接受使用加号 (+) 作为操作数来指定已在群集中注册的所有资源类型。如果未提供操作数，将显示已在群集中注册的所有资源类型的相关信息。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。

unregister

取消注册作为命令操作数指定的资源类型。此子命令接受使用加号 (+) 作为操作数来指定不存在任何实例的所有已注册资源类型。

您可以在全局群集中或区域群集中使用该子命令。

要从全局群集向区域群集取消注册资源类型，请使用 `-z` 选项指定区域群集名称。

在卸载用于定义资源类型的数据服务之前取消注册资源类型。

如果某特定资源类型的资源存在，则无法取消注册该资源类型。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。

另请参见 `register` 子命令的描述。

支持以下选项：

`-?`

`--help`

显示帮助信息。使用该选项时，不会执行任何其他处理。

指定此选项时，子命令可有可无。

如果仅指定此选项而不使用子命令，将会显示此命令的子命令列表。

如果使用子命令且指定此选项，将会显示该子命令的用法选项。

```
-f rtrfile|rtrfiledir
--rtrfile=rtrfile|rtrfiledir
--rtrfile rtrfile|rtrfiledir
```

指定 RTR 文件或包含 RTR 文件的目录的完整路径以用于注册资源类型。此选项仅可以与 `register` 子命令一同指定。

如果指定某个文件，则仅可以注册一种资源类型。

仅当所用的 RTR 文件不符合以下约定时，才需要指定此选项：

- RTR 文件通常位于 `/opt/cluster/lib/rgm/rtreg` 目录中。
- RTR 文件的名称通常与资源类型的名称相匹配。

Oracle 提供的所有 RTR 文件的位置和文件名称都遵循这些约定。例如，用于定义 `SUNW.nfs` 资源类型的 RTR 文件包含在文件 `/opt/cluster/lib/rgm/rtreg/SUNW.nfs` 中。

如果使用 `-i` 选项，则可以为配置信息中指定的任何资源类型指定 `resourcetypeRTRFile` 元素。`resourcetypeRTRFile` 元素可指定要用来注册资源类型的 RTR 文件。但是，`export` 子命令不会在生成的配置信息中包括 `resourcetypeRTRFile` 元素。有关 `resourcetypeRTRFile` 元素的更多信息，请参见 [clconfiguration\(5CL\) \[1245\]](#) 手册页。

```
-i {- | clconfiguration}
--input={- | clconfiguration}
--input {- | clconfiguration}
```

指定用于注册资源类型或用来修改已注册资源类型的节点列表的配置信息。此信息必须符合 [clconfiguration\(5CL\) \[1245\]](#) 手册页中定义的格式。此信息可包含在文件中或者通过标准输入 (`stdin`) 提供。要指定标准输入，请指定 `-` 而不是文件名。

仅作为命令操作数提供的资源类型会受此选项影响。命令中指定的选项将覆盖 `clconfiguration` 文件中设置的任何选项。如果 `clconfiguration` 文件中缺少配置参数，则必须在命令行上指定这些参数。

```
-N
--allnodes
```

指定操作数列表中的资源类型将在群集中的所有节点上可用。`-N` 选项还可以使这些资源类型对以后可能添加到群集中的任何节点可用。该选项可通过清除 `Installed_nodes` 属性实现此结果。

如果指定 `-N` 选项，则无法在同一命令中指定 `-n` 选项。

仅可随 `register` 子命令或 `set` 子命令指定 `-N` 选项。

`-n node[,...]`
`--node=node[,...]`
`--node node[,...]`

指定目标全局群集或区域群集中的一个节点或一系列节点。可以使用节点名称或节点 ID 指定每个节点。

如果指定 `-z` 选项，则仅可以使用 `-n` 选项指定区域群集主机名，而不能指定全局群集主机名。如果未指定 `-z` 选项，则仅可以使用 `-n` 选项指定全局群集主机名。

如果指定 `-n` 选项，则无法在同一命令中指定 `-N` 选项。

可以与此选项一起指定的子命令如下所述：

`add-node`

将指定的节点添加到已注册资源类型的节点的列表中。

`list`

仅显示已注册用于指定节点上的资源类型。

`register`

仅注册用于指定节点上的资源类型。如果省略 `-n` 选项，`register` 子命令将注册用于所有节点上的资源类型。该子命令还会为以后将添加到群集中的任何节点注册资源类型。

`remove-node`

从已注册资源类型的节点的列表中删除指定的节点。

`set`

使资源类型仅在指定节点上可用。

`show`

仅显示已注册用于指定节点上的资源类型的相关信息。

`-o {- | clconfiguration}`
`--output={- | clconfiguration}`
`--output {- | clconfiguration}`

指定资源类型的相关配置信息将写入到的位置。此位置可以是某个文件或标准输出 (`stdout`)。要指定标准输出，请指定 `-` 而不是文件名。如果指定标准输出，将会抑制该命令的其他所有标准输出。此选项仅可以与 `export` 子命令一同指定。

仅会写入作为命令操作数提供的资源类型的配置信息。信息是采用 [clconfiguration\(5CL\) \[1245\]](#) 手册页中定义的格式写入的。

```
-p name=value  
-p name+=array-values  
-p name-=array-values  
--property=name=value  
--property=name+=array-values  
--property=name-=array-values  
--property name=value  
--property name+=array-values  
--property name-=array-values
```

设置作为命令操作数提供的资源类型的属性值。

与此选项一起使用的运算符如下所述：

= 将属性设置为指定值。

+= 将一个或多个值添加到某个字符串数组值。仅可以为接受字符串值列表的属性指定此运算符，例如 `Installed_nodes`。

-= 从某个字符串数组值中删除一个或多个值。仅可以为接受字符串值列表的属性指定此运算符，例如 `Installed_nodes`。

在 `set` 子命令中使用选项 `-p Installed_nodes+=nodeC,nodeD` 相当于在 `add-node` 子命令中使用选项 `-n nodeC,nodeD`。

```
-p name[...]  
--property=name[...]  
--property name[...]
```

指定 `list-props` 子命令的属性列表。

```
-V  
--version
```

显示该命令的版本。

指定此选项时，不要将其与子命令、操作数或其他选项结合使用。子命令、操作数或其他选项将被忽略。`-v` 选项只显示命令的版本。不会执行任何其他操作。

```
-v  
--verbose
```

将详细消息显示到标准输出 (`stdout`)。

可以在任何格式的此命令中指定该选项。

请不要同时指定 `-v` 选项和 `-o` 选项。`-v` 选项将被忽略。`-o` 选项会抑制其他所有标准输出。

```
-z {zoneclustername | global | all}  
--zoneclustername={zoneclustername | global | all}  
--zoneclustername {zoneclustername | global | all}
```

指定已注册资源类型且要对其执行操作的一个或多个群集。

除 `export` 子命令以外的所有子命令均支持此选项。

如果指定此选项，还必须指定以下列表中的参数之一：

<code>zoneclustername</code>	指定使用此选项的命令仅对名为 <code>zoneclustername</code> 的区域群集中的所有指定资源类型起作用。
<code>global</code>	指定使用此选项的命令仅对全局群集中的所有指定资源类型起作用。
<code>all</code>	<p>如果在全局群集中使用此参数，则会指定使用此参数的命令对所有群集（包括全局群集和所有区域群集）中的所有指定资源类型起作用。</p> <p>如果在某区域群集中使用此参数，则会指定使用此参数的命令仅对该区域群集中的所有指定资源类型起作用。</p>

仅支持以下操作数：

<code>resourcetype</code>	<p>指定要管理的一种或多种资源类型。如果子命令接受多种资源类型，则可以使用加号 (+) 指定所有资源类型。</p> <p>有关资源类型名称格式的描述，请参见Unresolved link to "Oracle Solaris Cluster 数据服务规划和管理指南 中的合法的 RGM 名称"。</p>
---------------------------	--

如果命令对于所有指定的操作数均成功执行，则返回零 (CL_NOERR)。如果某个操作数出错，命令会继续处理操作数列表中的下一个操作数。返回的退出代码始终反映最先发生的错误。

可返回下列退出代码：

0 CL_NOERR

没有错误

您发出的命令已成功完成。

1 CL_ENOMEM

交换空间不足

某个群集节点耗尽了交换内存或其他操作系统资源。

3 CL_EINVAL

参数无效

您键入的命令不正确，或者使用 `-i` 选项提供的群集配置信息语法不正确。

6 CL_EACCESS

权限被拒绝

指定的对象不可访问。要发出命令，可能需要拥有超级用户或 RBAC 访问权限。有关更多信息，请参见 [Unresolved link to " su1M"](#) 和 [Unresolved link to " rbac5"](#) 手册页。

18 CL_EINTERNAL

遇到内部错误

内部错误表示软件缺陷或其他缺陷。

35 CL_EIO

I/O 错误

发生了物理输入/输出错误。

36 CL_ENOENT

没有这样的对象

由于以下某个原因，找不到您所指定的对象：

- 该对象不存在。
- 您尝试使用 -o 选项创建的配置文件路径中的某个目录不存在。
- 您尝试使用 -i 选项访问的配置文件包含错误。

37 CL_EOP

不允许操作

您尝试对不受支持的配置执行操作，或者您所执行的操作不受支持。

41 CL_ETYPE

类型无效

您使用 -t 或 -p 选项指定的类型不存在。

这些退出值与 [scha_calls\(3HA\) \[897\]](#) 手册页中描述的返回码兼容。

例 214 注册资源类型

本示例注册其数据服务已安装在所有节点上但尚未注册的所有资源类型。该命令以简洁模式运行。

```
# clresourcetype register +
```

例 215 在选定的节点上注册资源类型

本示例注册其数据服务已安装在节点 `phys-schost-1` 和节点 `phys-schost-2` 上但尚未注册的所有资源类型。资源将仅在这些节点上可用。在本示例中，该命令没有返回任何错误。该命令以详细模式运行。

```
# clresourcetype register -v -n phys-schost-1,phys-schost-2 +
```

以下命令将注册其数据服务已安装在区域群集 ZC 的区域群集节点 zc-host-1 和 zc-host-2 上但尚未注册的所有资源类型。资源仅在这些区域群集节点上可用。

```
# clresourcetype register -n zc-host-1,zc-host-2 -Z ZC +
```

例 216 注册单个资源类型

本示例注册 SUNW.nfs:3.2 资源类型。此资源类型的数据服务将安装在所有群集节点上。

```
# clresourcetype register nfs:3.2
```

例 217 列出资源类型

本示例仅列出所有已注册的资源类型的名称。

```
# clresourcetype list
SUNW.LogicalHostname
SUNW.SharedAddress
SUNW.nfs
SUNW.apache
```

例 218 列出资源类型及其节点列表

本示例列出所有已注册的资源类型及其节点列表。

```
# clresourcetype list -v
Resource Type           Node List
-----
SUNW.LogicalHostname   <all>
SUNW.SharedAddress     <all>
SUNW.nfs               phys-schost-1,phys-schost-2,phys-schost-3
SUNW.apache            phys-schost-1,phys-schost-2,phys-schost-3
```

在从全局群集节点执行以下命令时，该命令会列出区域群集 ZC 中所有已注册的资源类型。

```
# clresourcetype list -Z ZC
SUNW.nfs
SUNW.apache
```

例 219 列出指定节点上的资源类型

本示例列出已在 phys-schost-4 上注册的所有资源类型。

```
# clrt list -n phys-schost-4
SUNW.LogicalHostname
SUNW.SharedAddress
```

有关以下属性的描述，请参见 [Unresolved link to " attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving（发展中）

[Intro\(1CL\) \[17\]](#), [clreslogicalhostname\(1CL \) \[207\]](#), [clresource\(1CL\) \[225\]](#),
[clresourcegroup\(1CL\) \[253\]](#), [clressharedaddress\(1CL\) \[289\]](#), [cluster\(1CL\) \[461\]](#),
[scha_calls\(3HA\) \[897\]](#), [clconfiguration\(5CL\) \[1245\]](#), [r_properties\(5\) \[1103\]](#),
[Unresolved link to " attributes5"](#), [Unresolved link to " rbac5"](#)

[Unresolved link to " Oracle Solaris Cluster 数据服务规划和管理指南 中的资源组属性"](#)

超级用户可以运行此命令的所有格式。

任何用户都可以运行带有以下选项的此命令：

- -? 选项
- -v 选项

要运行带有子命令的此命令，超级用户以外的用户需要具有 RBAC 授权。请参见下表。

子命令	RBAC 授权
add-node	solaris.cluster.modify
export	solaris.cluster.read
list	solaris.cluster.read
list-props	solaris.cluster.read
set	solaris.cluster.modify
register	solaris.cluster.modify
remove-node	solaris.cluster.modify
show	solaris.cluster.read
unregister	solaris.cluster.admin

名称

clsetup — 以交互方式配置 Oracle Solaris Cluster

```
/usr/cluster/bin/clsetup -V  
/usr/cluster/bin/clsetup -?  
/usr/cluster/bin/clsetup [-f logfile]
```

clsetup 命令可提供以下配置功能，具体取决于发出命令时群集所处的状态。用户必须是超级用户才能运行此命令。

此命令没有简短格式。

- 在安装后运行 clsetup 命令时，该命令会执行初始设置任务，例如配置法定设备和重置 installmode 属性。如果在使用 scinstall 或 cluster create 命令创建群集时取消选择自动法定配置，则必须在安装完群集后立即运行 clsetup 命令。在运行 clsetup 命令并重置 installmode 属性之前，确保所有节点都已加入群集。

如果在创建群集时使用自动法定配置，则无需在群集安装后运行 clsetup 命令。自动法定配置功能还会重置群集的 installmode 属性。

您可以从群集中的任一节点发出此格式的 clsetup 命令。

- 在正常群集操作期间运行 clsetup 命令时，clsetup 命令会提供交互式菜单驱动的实用程序来执行群集配置任务。下面是此实用程序管理的其中一些群集组件：
 - 法定设备
 - 资源组
 - 数据服务
 - 群集互连
 - 设备组和卷
 - 专用主机名
 - 新节点
 - 区域群集
 - 其他群集属性

您可以从群集中的任一节点发出此格式的 clsetup 命令。

- 在从处于非群集模式的节点运行 clsetup 命令时，clsetup 命令会提供菜单驱动的实用程序来更改和显示专用 IP 地址范围。

在启动此格式的 clsetup 实用程序之前，必须将所有节点重新引导到非群集模式。

支持以下选项：

-?

--help

显示命令的帮助信息。

-f *logfile*

--file *logfile*

指定可以记录命令的日志文件的名称。如果指定此选项，则可以运行并记录或者仅记录由 `clsetup` 生成的大多数命令集，具体取决于用户响应。

-V

--version

显示命令集的版本。不会执行任何命令行处理，且该命令不会进入其交互式菜单。

有关以下属性的描述，请参见 [Unresolved link to " attributes5 "](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

[Intro\(1CL\) \[17\]](#), [cldevicegroup\(1CL\) \[65\]](#), [clnode\(1CL\) \[149\]](#), [clquorum\(1CL\) \[195\]](#), [clreslogicalhostname\(1CL\) \[207\]](#), [clresourcegroup\(1CL\) \[253\]](#), [clresourcetype\(1CL\) \[277\]](#), [clressharedaddress\(1CL\) \[289\]](#), [cluster\(1CL\) \[461\]](#), [cltelemetryattribute\(1CL\) \[447\]](#), [clzonecluster\(1CL\) \[517\]](#)

[Unresolved link to " Oracle Solaris Cluster 软件安装指南 "](#)、[Unresolved link to " Oracle Solaris Cluster 系统管理指南 "](#)

名称

clsnmphost — 管理 Oracle Solaris Cluster SNMP 主机的列表

```
/usr/cluster/bin/clsnmphost -V
/usr/cluster/bin/clsnmphost [subcommand] -?
/usr/cluster/bin/clsnmphost subcommand [[options]] -v [host]
/usr/cluster/bin/clsnmphost add [-c community[,...]]
    [-n node[,...]] host [...]
/usr/cluster/bin/clsnmphost add -i {- | clconfigfile}
    [-c community[,...]] [-n node[,...]] host [...]
/usr/cluster/bin/clsnmphost export [-o {- | clconfigfile}]
    [-c community[,...]] [-n node[,...]] [+ | host...]
/usr/cluster/bin/clsnmphost list [-c community[,...]]
    [-n node[,...]] [+ | host...]
/usr/cluster/bin/clsnmphost remove [-c community[,...]]
    [-n node[,...]] {+ | host...}
/usr/cluster/bin/clsnmphost show [-c community[,...]]
    [-n node[,...]] [+ | host...]
```

clsnmphost 命令可管理将接收 SNMP 事件通知的简单网络管理协议 (Simple Network Management Protocol, SNMP) 主机和团体名称。SNMP 主机使用群集管理信息库 (Management Information Base, MIB) 提供访问控制机制。当 MIB 发送 SNMP 陷阱通知时, 使用此命令配置的 SNMP 主机可以确定要将陷阱通知发送到的主机。有关群集 MIB 的更多信息, 请参见 [clsnmpmib\(1CL\) \[417\]](#) 手册页。

此命令没有简短格式。

此命令的一般格式如下所示：

```
clsnmphost [subcommand] [options] [operands]
```

仅当 *options* 指定选项 *-?* 或 *-v* 时, 才可以省略 *subcommand*。

此命令的每个选项都有长格式和简短格式两种格式。在本手册页“选项”部分给出了每个选项的两种格式, 以及对该选项的描述。

有关更多信息, 请参见 [Intro\(1CL\) \[17\]](#) 手册页。

仅可以在全局区域中使用此命令。

子命令

支持以下子命令：

add

将 SNMP 主机添加到将接收群集 MIB 陷阱通知并可以访问 MIB 表的主机列表中。仅可以在全局区域中运行该子命令。

如果在不带 `-n` 选项的情况下使用 `add` 子命令，则仅影响当前节点。如果在不带 `-c` 选项的情况下使用 `add`，则该子命令会使用 `public` 作为默认团体名称。使用 IP 地址或主机名指定主机。

如果指定的团体名称不存在，此命令将创建该团体。使用 `-i` 选项从 `clconfigfile` 导入一个或多个主机配置。

超级用户以外的用户需要具有 `solaris.cluster.modify` 基于角色的访问控制 (role-based access control, RBAC) 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

export

导出指定节点的 SNMP 主机信息。

仅可以在全局区域中运行该子命令。

使用 `-n` 选项可指定一个或多个要导出其 SNMP 主机信息的节点。如果在不带 `-n` 选项的情况下使用 `export`，则该子命令将仅导出当前节点的 SNMP 主机信息。

有关 `export` 子命令的输出格式的更多信息，请参见 [clconfiguration\(5CL\) \[1245\]](#) 手册页。默认情况下，所有输出都会发送到标准输出。使用 `-o` 选项后跟文件名可将输出重定向到该文件。

通过使用 `-c` 选项，可以将 `export` 子命令的输出限定为仅与特定团体中主机相关的信息。将一个或多个主机指定为操作数可将输出信息限定为仅与这些主机相关的信息。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

list

列出在指定节点上配置的 SNMP 主机。

仅可以在全局区域中运行该子命令。

如果在不带 `-n` 选项的情况下使用 `list` 子命令，则仅列出当前节点上的 SNMP 主机。默认情况下，该子命令会列出节点上的所有主机。要将输出限定为与特定主机有关的信息，请将一个或多个主机指定为操作数。也可以使用 `-c` 选项仅列出指定团体中的这些主机。

superuser (超级用户) 以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

remove

从节点配置中删除一个 SNMP 主机。

仅可以在全局区域中运行该子命令。

要从配置中删除某个主机，必须将该主机名指定为操作数。如果在不带 `-n` 选项的情况下使用 `remove` 子命令，则仅删除当前节点上的 SNMP 主机。要删除所有主机，请使用加号 (+)。要从特定团体中删除一个或多个主机，请使用 `-c` 选项。

`superuser` (超级用户) 以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5" 手册页](#)。

show

显示指定节点上的 SNMP 主机信息。

仅可以在全局区域中运行该子命令。

如果在不带 `-n` 选项的情况下使用 `show` 子命令，则仅显示当前节点上的 SNMP 主机信息。默认情况下，`show` 子命令会显示所有主机及其团体的信息。要将输出限定为仅与团体中特定主机相关的信息，请使用 `-c` 选项，或将一个或多个主机名指定为操作数。

`superuser` (超级用户) 以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5" 手册页](#)。

支持以下选项：

`-?`

`--help`

显示帮助信息。

指定此选项时，子命令可有可无。

- 如果在不带子命令的情况下使用此选项，将会显示可用子命令的列表。
- 如果将该选项与子命令结合使用，将会显示该子命令的用法选项。

使用该选项时，不会执行任何其他处理。

`-c community`

`--community community`

指定将与主机名结合使用的 SNMP 团体名称。还可以将该选项与其他子命令结合使用以缩小子命令作用范围。例如，与 `remove` 子命令结合使用时，`-c` 选项可用于从特定 `community` 中删除一个或多个主机。如果使用不带 `-c` 选项的 `add` 子命令，该子命令会使用 `public` 作为默认团体名称。

`-i {- | clconfigfile}`

`--input {- | clconfigfile}`

指定可用于验证或修改 SNMP 主机配置的配置信息。此信息必须符合 [clconfiguration\(5CL\) \[1245\]](#) 手册页中定义的格式。此信息可包含在文件中或者通过标准输入提供。要指定标准输入，请指定减号 (-) 而不是文件名。

`-n node[...]`

`--node[s] node[...]`

指定一个节点或一系列节点。可以使用节点名称或节点 ID 指定每个节点。`clsnmpshost` 命令的所有格式都接受此选项。

`-o {- | clconfigfile}`

`--output {- | clconfigfile}`

以 [clconfiguration\(5CL\) \[1245\]](#) 手册页定义的格式写入群集 SNMP 主机配置信息。可以将此信息写入文件或标准输出中。

要写入标准输出，请指定减号 (-) 而不是文件名。如果指定标准输出，将抑制命令的所有其他标准输出。

如果提供文件名，配置会被复制到以该名称命名的新文件中。

`-o` 选项仅对 `export` 子命令有效。如果没有指定 `-o` 选项，输出将显示到标准输出。

`-V`

`--version`

显示该命令的版本。

不要将该选项与子命令、操作数或其他选项一起指定，因为子命令、操作数或其他选项会被忽略。`-V` 选项只显示命令的版本。不会执行任何其他操作。

`-v`

`--verbose`

将详细信息显示到标准输出。

可以将该选项与命令的任何格式一同指定，尽管一些子命令可能不会生成扩展输出。例如，指定详细选项时 `export` 子命令不会生成扩展输出。

支持以下操作数：

`+` 指定所有 SNMP 主机条目。

`host` 指定供访问群集上的 SNMP MIB 的主机的 IP 地址、IPv6 地址或名称。

如果命令对于所有指定的操作数均成功执行，则返回零 (`CL_NOERR`)。如果某个操作数出错，命令会继续处理操作数列表中的下一个操作数。返回的退出代码始终反映最先发生的错误。

此命令返回下列退出状态代码：

`0 CL_NOERR`

没有错误

您发出的命令已成功完成。

1 CL_ENOMEM

交换空间不足

某个群集节点耗尽了交换内存或其他操作系统资源。

3 CL_EINVAL

参数无效

您键入的命令不正确，或者使用 `-i` 选项提供的群集配置信息语法不正确。

6 CL_EACCESS

权限被拒绝

指定的对象不可访问。要发出命令，可能需要拥有超级用户或 RBAC 访问权限。有关更多信息，请参见 [Unresolved link to " su1M"](#) 和 [Unresolved link to " rbac5"](#) 手册页。

18 CL_EINTERNAL

遇到内部错误

内部错误表示软件缺陷或其他缺陷。

35 CL_EIO

I/O 错误

发生了物理输入/输出错误。

36 CL_ENOENT

没有这样的对象

由于以下某个原因，找不到您所指定的对象：

- 该对象不存在。
- 您尝试使用 `-o` 选项创建的配置文件路径中的某个目录不存在。
- 您尝试使用 `-i` 选项访问的配置文件包含错误。

例 220 通过指定主机名添加主机

以下命令可将主机 `myhost` 添加到当前节点 `private` 上团体的 SNMP 主机列表中。

```
# clsnmphost add -c private phys-schost-1
```

将主机添加到 `public` 以外的其他团体时，必须指定团体名称。

例 221 通过指定主机 IP 和 IPv6 地址添加主机

以下命令可将主机添加到当前节点上团体 `public` 的 SNMP 主机列表中。该命令的首个版本通过指定主机的 IP 地址来添加主机。该命令的第二个版本通过指定主机的 IPv6 地址来添加主机。

```
# clsnmphost add -c public 192.168.12.12
Or
```

```
# clsnmphost add -c public fe:1::5
```

例 222 删除主机

以下命令可从团体 `private` 中删除所有主机。

```
# clsnmphost remove -c private +
```

例 223 列出当前节点上的主机

以下命令可列出当前节点上的所有主机。

```
# clsnmphost list
phys-schost-1
192.168.12.12
```

例 224 列出主机及其团体名称

以下命令使用详细选项 `-v` 列出当前节点上的所有主机及其团体名称。

```
# clsnmphost list -v

--- SNMP hosts on node phys-schost-1 ---

Host Name      Community
-----
phys-schost-1  private
192.168.12.12  public
```

例 225 显示 SNMP 主机配置

以下命令可显示节点 `phys-cluster-2` 上 SNMP 主机的所有配置信息。

```
# clsnmphost show -n phys-schost-2

--- SNMP Host Configuration on phys-schost-2 ---

SNMP Host Name:      phys-schost-2
Community:           private
```

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core

属性类型	属性值
接口稳定性	Evolving (发展中)

[clsnmpmib\(1CL\) \[417\]](#)、[cluster\(1CL\) \[461\]](#)、[Intro\(1CL\) \[17\]](#)、[sceventmib\(1M\) \[689\]](#)、[Unresolved link to "su1M"](#)、[scha_calls\(3HA\) \[897\]](#)、[Unresolved link to "attributes5"](#)、[Unresolved link to "rbac5"](#)、[clconfiguration\(5CL\) \[1245\]](#)

超级用户可以运行此命令的所有格式。

所有用户都可以运行带有 `-?` (帮助) 或 `-v` (版本) 选项的此命令。

要运行带有其他子命令的 `clsnmpmib` 命令，超级用户以外的用户需要具有 RBAC 授权。请参见下表。

子命令	RBAC 授权
<code>add</code>	<code>solaris.cluster.modify</code>
<code>export</code>	<code>solaris.cluster.read</code>
<code>list</code>	<code>solaris.cluster.read</code>
<code>remove</code>	<code>solaris.cluster.modify</code>
<code>show</code>	<code>solaris.cluster.read</code>

名称

clsnmpmib, ctmib — 管理 Oracle Solaris Cluster SNMP MIB

```
/usr/cluster/bin/clsnmpmib -V
/usr/cluster/bin/clsnmpmibsubcommand [] -?
/usr/cluster/bin/clsnmpmib [subcommand] [options] -v [mib]
/usr/cluster/bin/clsnmpmib disable [-n node[,...]] {+ | mib ...}
/usr/cluster/bin/clsnmpmib enable [-n node[,...]] {+ | mib ...}
/usr/cluster/bin/clsnmpmib export [-n node[,...]]
    [-o {- | clconfigfile}] {+ | mib ...}
/usr/cluster/bin/clsnmpmib list [-n node[,...]] {+ | mib ...}
/usr/cluster/bin/clsnmpmib set [-p name=value] [...]
    [-n node[,...]] {+ | mib ...}
/usr/cluster/bin/clsnmpmib show [-n node[,...]] {+ | mib ...}
```

clsnmpmib 命令可管理当前节点上的现有 Oracle Solaris Cluster 简单网络管理协议 (Simple Network Management Protocol, SNMP) 管理信息库 (Management Information Base, MIB)。要创建可以管理 MIB 的 SNMP 主机，请参见 [clsnmphost\(1CL\) \[409\]](#) 手册页。要定义可以使用 SNMP 版本 3 (SNMP Version 3, SNMPv3) 协议访问 MIB 的 SNMPv3 用户，请参见 [clsnmpuser\(1CL\) \[425\]](#) 手册页。

此命令的一般格式如下所示：

```
clsnmpmib [subcommand] [options] [operands]
```

仅当 *options* 指定选项 *-?* 或 *-v* 时，才可以省略 *subcommand*。

此命令的每个选项都有长格式和简短格式两种格式。在“选项”部分给出了每个选项的两种格式，以及对该选项的描述。

有关更多信息，请参见 [Intro\(1CL\) \[17\]](#) 手册页。

Oracle Solaris Cluster MIB

Oracle Solaris Cluster 当前支持一个 MIB，即事件 MIB。Oracle Solaris Cluster SNMP 事件 MIB 可实时向 SNMP 管理器通知群集事件。一经启用，Oracle Solaris Cluster 事件 MIB 就会自动向 clsnmphost 命令定义的所有主机发送陷阱通知。Oracle Solaris Cluster 事件 MIB 在端口 11162 上发送陷阱通知。可在端口 11161 上查看 SNMP 树。

可以使用 `clsnmpmib set` 子命令指定 `min_severity` 或 `log_number` 值。由于群集会生成大量事件通知，因此只有严重程度为 `min_severity` 或更高的事件才作为陷阱通知发送。默认情况下，`min_severity` 值设置为 `NOTICE`。`log_number` 值指定在停止使用较旧的条目之前要在 MIB 表中记录的事件的数量。MIB 可维护一个只读表，其中包含最近发生的已作为陷阱发送的事件。事件数受 `log_number` 值限制。系统重新引导后此信息将不再存在。

仅可以在全局区域中使用此命令。

支持以下子命令：

`disable`

在指定节点禁用一个或多个群集 MIB。

仅可以在全局区域中运行该子命令。

如果没有指定 `-n` 选项，则仅禁用当前节点上的 MIB。禁用 MIB 时，无法访问 MIB 表，并且 MIB 不会发送任何陷阱通知。

超级用户以外的用户需要具有 `solaris.cluster.modify` 基于角色的访问控制 (Role-Based Access Control, RBAC) 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

`enable`

在指定节点启用一个或多个群集 MIB。

仅可以在全局区域中运行该子命令。

如果没有指定 `-n` 选项，则仅启用当前节点上的 MIB。要限制启用的 MIB，请使用 `mib` 操作数。

启用 MIB 时，也就启用了它的所有功能。但是，要使所有 MIB 功能完全发挥作用，可能有必要进行一些进一步的配置。例如，如果未配置任何主机，MIB 将无法发送陷阱通知。有关配置 SNMP 主机的信息，请参见 [clsnmphost\(1CL\) \[409\]](#) 手册页。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

`export`

导出群集 MIB 配置信息。

仅可以在全局区域中运行该子命令。

使用 `-n` 选项可指定一个或多个要从中导出 MIB 配置信息的节点。如果在不带 `-n` 选项的情况下使用 `export`，则该子命令仅从当前节点导出 MIB 配置信息。默认情况下，该子命令导出当前节点上所有 MIB 的配置信息。要进一步完善输出，请指定您需要其配置信息的一个或多个 MIB 的名称。

有关 `export` 子命令的输出格式的更多信息，请参见 [clconfiguration\(5CL\) \[1245\]](#) 手册页。默认情况下，所有输出都会发送到标准输出。使用 `-o` 选项后跟文件名可将输出重定向到该文件。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

list

显示指定节点上的群集 MIB 列表。

仅可以在全局区域中运行该子命令。

使用 `-n` 选项可指定要列出的群集 MIB 的节点。如果在不带 `-n` 选项的情况使用 `list` 子命令，则该子命令仅列出当前节点上的 MIB。要限制列出的 MIB，请指定要列出的一个或多个 MIB 的名称。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

set

更改指定节点上的一个或多个 MIB 使用的 SNMP 协议 `min_severity` 或 `log_number` 设置。

仅可以在全局区域中运行该子命令。

默认情况下，该子命令会更改节点上的所有 MIB。如果没有指定节点，则仅修改当前节点上 MIB 的 SNMP 属性。必须通过使用 `-p` 选项指定 SNMP 属性。

所有 MIB 都使用以下默认属性值：`protocol:SNMPv2`，`min_severity:NOTICE`，`log_number:100`。除非使用 `mib` 操作数指定 MIB 名称，否则 `set` 子命令会更改所有 MIB 的协议、`min_severity` 或 `log_number` 设置。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

show

显示指定节点上 MIB 的信息。

仅可以在全局区域中运行该子命令。

`show` 子命令显示 MIB 的名称、其 SNMP 协议版本、`min_severity` 值或 `log_number` 值。默认情况下，该子命令会显示节点上所有 MIB 的信息。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

支持以下选项：

`-?`

`--help`

显示帮助信息。

指定此选项时，子命令可有可无。

- 如果在不带子命令的情况下使用此选项，将会显示可用子命令的列表。
- 如果将该选项与子命令结合使用，将会显示该子命令的用法选项。

使用该选项时，不会执行任何其他处理。

`-n node[...]`

`--node[s] node[...]`

指定一个节点或一系列节点。可以使用节点名称或节点 ID 指定每个节点。`clsnmpmib` 命令的所有格式都接受此选项。可以使用 `-n` 选项指定要在其上执行操作的节点。如果不使用 `-n` 选项，则该命令会采用当前节点。

`-o {- | clconfigfile}`

`--output {- | clconfigfile}`

指定要写入群集 MIB 配置信息的位置。此位置可以是文件，也可以是标准输出。要指定标准输出，请指定减号 (-) 而不是文件名。如果指定标准输出，将会抑制该命令的其他所有标准输出。如果没有指定 `-o` 选项，输出将发送到标准输出。此选项仅可以与 `export` 子命令一同指定。

配置信息是采用 [clconfiguration\(5CL\) \[1245\]](#) 手册页中定义的格式写入的。

`-p name= value`

`--property=name =value`

`--property name=value`

`version` 指定 MIB 将使用的 SNMP 协议的版本。Oracle Solaris Cluster 支持 SNMPv2 和 SNMPv3 协议版本。

`min_severity` 指定最低严重程度值。只有其值等于或大于 `min_severity` 值的事件才会记录在 MIB 表中，对于这些事件，将向配置的主机发送陷阱。

`log_number` 指定在停止使用较旧的条目之前要在 MIB 表中记录的事件的数量。

允许使用多个 `-p name= value` 实例。

可以使用该选项设置以下属性：

`version`

指定 MIB 将使用的 SNMP 协议的版本。您可以按如下方式指定 `value`：

- `version=SNMPv2`
- `version=snmpv2`
- `version=2`
- `version=SNMPv3`
- `version=snmpv3`
- `version=3`

`min_severity`

指定要用于 MIB 的最低严重性级别值。您可以按如下方式指定值：

- `min_severity=NOTICE`

-
- min_severity=WARNING
 - min_severity=ERROR
 - min_severity=CRITICAL
 - min_severity=FATAL

允许使用大写或小写的值。

log_number

指定在停止使用较旧的条目之前要在 MIB 表中记录的事件的数量。默认值为 100。值范围为 100 到 500。您可以按如下方式指定value：

- log_number=*number*

-V

--version

显示该命令的版本。

不要将该选项与子命令、操作数或其他选项一起指定，因为子命令、操作数或其他选项会被忽略。-V 选项只显示命令的版本。不会执行任何其他操作。

-v

--verbose

将详细信息显示到标准输出。

可以将该选项与命令的任何格式一同指定，尽管一些子命令可能不会生成扩展输出。例如，指定详细选项时 export 子命令不会生成扩展输出。

支持以下操作数：

mib 指定要应用子命令的一个或多个 MIB 的名称。如果您没有指定此操作数，该子命令将使用默认的加号 (+)，这意味着所有 MIB。如果使用 *mib* 操作数，请在空格分隔列表中所有其他命令行选项之后指定 MIB。

+ 所有群集 MIB。

如果命令对于所有指定的操作数均成功执行，则返回零 (CL_NOERR)。如果某个操作数出错，命令会继续处理操作数列表中的下一个操作数。返回的退出代码始终反映最先发生的错误。

此命令返回下列退出状态代码：

0 CL_NOERR

没有错误

您发出的命令已成功完成。

1 CL_ENOMEM

交换空间不足

某个群集节点耗尽了交换内存或其他操作系统资源。

3 CL_EINVAL

参数无效

您键入的命令不正确，或者使用 -i 选项提供的群集配置信息语法不正确。

6 CL_EACCESS

权限被拒绝

指定的对象不可访问。要发出命令，可能需要拥有超级用户或 RBAC 访问权限。有关更多信息，请参见 [Unresolved link to " su1M"](#) 和 [Unresolved link to " rbac5"](#) 手册页。

18 CL_EINTERNAL

遇到内部错误

内部错误表示软件缺陷或其他缺陷。

35 CL_EIO

I/O 错误

发生了物理输入/输出错误。

36 CL_ENOENT

没有这样的对象

由于以下某个原因，找不到您所指定的对象：

- 该对象不存在。
- 您尝试使用 -o 选项创建的配置文件路径中的某个目录不存在。
- 您尝试使用 -i 选项访问的配置文件包含错误。

例 226 列出 MIB

以下命令可列出群集节点上的所有 MIB。

```
# clsnmpmib list
Event
```

例 227 启用 MIB

以下命令可启用当前节点上的事件 MIB。

```
# clsnmpmib enable event
```

群集 MIB 的名称不区分大小写。

例 228 更改协议

以下命令可将 phys-cluster-2 群集节点上 Event (事件) MIB 的协议更改为 SNMPv3。

```
# clsnmpmib set -n phys-cluster-2 -p version=SNMPv3 Event
```

如果使用 -n 选项，也可以使用节点 ID 代替节点名称。

例 229 显示配置

以下命令可显示群集节点 phys-cluster-1 和 phys-cluster-2 上的配置信息。

```
# clsnmpmib show -n phys-cluster-1,phys-cluster-2
```

```
--- SNMP MIB Configuration on myhost ---
```

```
SNMP MIB Name:          phys-cluster-1
State:                  Event
Enabled:                yes
Protocol:               SNMPv3
min_severity:          1
log_number:             100
SNMP MIB Name:          phys-cluster-2
State:                  Event
Enabled:                yes
Protocol:               SNMPv3
min_severity:          3
log_number:             250
```

例 230 更改 Min Severity 值

以下命令可将 phys-cluster-2 群集节点上事件 MIB 的 min_severity 更改为 WARNING。

```
# clsnmpmib set -n phys-cluster-2 -p min_severity=WARNING Event
```

如果使用 -n 选项，也可以使用节点 ID 代替节点名称。

例 231 更改 Log_Number 值

以下命令可将 phys-cluster-2 群集节点上事件 MIB 的 log_number 更改为 250。

```
# clsnmpmib set -n phys-cluster-2 -p log_number=250 Event
```

如果使用 -n 选项，也可以使用节点 ID 代替节点名称。

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

/usr/cluster/lib/mib/sun-cluster-event-mib.mib

Oracle Solaris Cluster SNMP 事件 MIB 定义文件

[clsnmphost\(1CL\) \[409\]](#)、[clsnmpuser\(1CL\) \[425\]](#)、[Intro\(1CL\) \[17\]](#)、[cluster\(1CL\) \[461\]](#)、[sceventmib\(1M\) link to "attributes5"](#)、[Unresolved link to "rbac5"](#)、[clconfiguration\(5CL\) \[1245\]](#)

超级用户可以运行此命令的所有格式。

所有用户都可以运行带有 -? (帮助) 或 -v (版本) 选项的此命令。

要运行带有其他子命令的 `clsnmpmib` 命令，超级用户以外的用户需要具有 RBAC 授权。请参见下表。

子命令	RBAC 授权
disable	solaris.cluster.modify
enable	solaris.cluster.modify
export	solaris.cluster.read
list	solaris.cluster.read
set	solaris.cluster.modify
show	solaris.cluster.read

名称

clsnmpuser — 管理 Oracle Solaris Cluster SNMP 用户

```
/usr/cluster/bin/clsnmpuser -V

/usr/cluster/bin/clsnmpuser [subcommand] -?

/usr/cluster/bin/clsnmpuser [subcommand] [options] -v [operand]

/usr/cluster/bin/clsnmpuser create -i {- | clconfigfile}
    [-a authentication] -f passwdfile [-n node[,...]] {+ | user ...}

/usr/cluster/bin/clsnmpuser delete [-a authentication] [-n node[,...]]
    {+ | user ...}

/usr/cluster/bin/clsnmpuser export [-o {- | clconfigfile}]
    [-a authentication] [-n node[,...]] [{+ | user ...}]

/usr/cluster/bin/clsnmpuser list [-a authentication] [-n node[,...]]
    {-d | + | user ...}

/usr/cluster/bin/clsnmpuser set [-a authentication] [-n node[,...]]
    {+ | user ...}

/usr/cluster/bin/clsnmpuser set-default {-l seclevel [,...]}
    {+ | user ...}

/usr/cluster/bin/clsnmpuser show [-a authentication] [-n node[,...]]
    [-d | + | user ...]
```

clsnmpuser 命令可管理简单网络管理协议 (Simple Network Management Protocol, SNMP) 用户 (这些用户可以管理群集管理信息库 (Management Information Base, MIB) 中的控制机制) 的角色。有关群集 MIB 的更多信息, 请参见 [clsnmpmib\(1CL\) \[417\]](#) 手册页。如果群集包含一个被配置为使用 SNMP 版本 3 (SNMP Version 3, SNMPv3) 的 MIB, 您必须定义一个 SNMP 用户。SNMP 用户与 Solaris OS 用户并不相同, SNMP 用户不需要采用与现有 OS 用户相同的用户名。

此命令没有简短格式。

此命令的一般格式如下所示：

```
clsnmpuser [subcommand] [options] [operands]
```

仅当 *options* 指定选项 -? 或 -v 时, 才可以省略 *subcommand*。

此命令的每个选项都有长格式和简短格式两种格式。在“选项”部分给出了每个选项的两种格式, 以及对该选项的描述。

有关更多信息, 请参见 [Intro\(1CL\) \[17\]](#) 手册页。

仅可以在全局区域中使用此命令。

支持以下子命令：

create

创建用户并将该用户添加到指定节点上的 SNMP 用户配置中。

仅可以在全局区域中运行该子命令。

将 `-n` 选项与该子命令一起使用可指定要在其上创建 SNMP 用户的群集节点。如果没有指定 `-n` 选项，将创建用户并仅将该用户添加到当前节点上的 SNMP 配置中。

要创建并添加在 `clconfiguration` 文件中配置的所有用户，请使用 `-i` 选项和 `-n` 选项。

要将验证类型分配给要创建的 SNMP 用户，请指定 `-a` 选项。

通过指定 `-f` 选项，可以包括 SNMP 用户的密码。如果要使用 `-i` 选项，则 `-f` 选项是必需的。

如果指定 `-i` 选项，则使用 `clconfiguration(5CL) [1245]` 文件中的配置信息。指定 `-i` 选项时，还可以指定加号 (+) 操作数或用户列表。

超级用户以外的用户需要具有 `solaris.cluster.modify` 基于角色的访问控制 (role-based access control, RBAC) 授权才能使用该命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

delete

从指定节点删除一个 SNMPv3 用户。

仅可以在全局区域中运行该子命令。

使用 `delete` 子命令并仅指定一个用户名时，该子命令会删除该用户的所有实例。要按验证类型删除用户，请使用 `-a` 选项。如果没有使用 `-n` 选项，则仅从当前节点删除用户。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

export

从指定节点导出 SNMP 用户信息。

仅可以在全局区域中运行该子命令。

如果没有使用 `-n` 选项，则仅从当前节点导出 SNMP 用户信息。有关 `export` 子命令的输出格式，请参见 `clconfiguration(5CL) [1245]` 手册页。默认情况下，所有输出都会发送到标准输出。使用 `-o` 选项后跟文件名可将输出重定向到该文件。

可以使用 `-a` 选项仅提供具有特定验证类型的用户的输出。如果将一个或多个用户指定为操作数，则输出将被限定为仅与这些用户相关的信息。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

list

打印在指定节点配置的 SNMPv3 用户的列表。

仅可以在全局区域中运行该子命令。

默认情况下，list 子命令会显示指定节点上的所有 SNMPv3 用户。要仅显示默认的 SNMP 用户，请指定 -d 选项，不带任何操作数。要将输出限定为指定的验证类型，请使用 -a 选项。

超级用户以外的用户需要具有 solaris.cluster.read RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

set

更改指定节点上的用户配置。

仅可以在全局区域中运行该子命令。

如果没有指定 -n 选项，则仅修改当前节点上的用户配置。

超级用户以外的用户需要具有 solaris.cluster.modify RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

set-default

指定默认 SNMP 用户的名称以及 MIB 发送陷阱通知时使用的安全性级别。

仅可以在全局区域中运行该子命令。

使用 -l 选项可指定安全性级别。

如果将 MIB 配置为使用 SNMPv3，则必须指定一个特定用户名和用于验证陷阱的安全性级别。如果配置中包含多个用户，则必须指定 MIB 发送陷阱通知时将使用的默认用户。

如果配置中仅包含一个用户，则该用户将自动成为默认 SNMP 用户。如果删除了默认 SNMP 用户，则另一个现有用户（如果有）将成为默认用户。

超级用户以外的用户需要具有 solaris.cluster.modify RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

show

显示指定节点上的用户相关信息。

仅可以在全局区域中运行该子命令。

默认情况下，show 子命令会显示节点上所有用户的相关信息。要仅显示默认 SNMP 用户的相关信息，请指定 -d 选项，并且不提供操作数。要将输出限定为特定的验证类型，请使用 -a 选项。如果没有使用 -n 选项，则该命令仅显示当前节点上的用户信息。

超级用户以外的用户需要具有 solaris.cluster.read RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

支持以下选项：

-?
--help
显示帮助信息。
指定此选项时，子命令可有可无。

- 如果在不带子命令的情况下使用此选项，将会显示可用子命令的列表。
- 如果将该选项与子命令结合使用，将会显示该子命令的用法选项。

使用该选项时，不会执行任何其他处理。

-a *authentication*
--authentication *authentication*
指定用于对用户进行授权的验证协议。验证协议的值可以是 SHA 或 MD5。

-d
--default
指定 MIB 发送陷阱通知时使用的默认 SNMP 用户。

-f *passwdfile*
--file *passwdfile*
指定包含一个或多个 SNMP 用户密码的文件。如果在创建新用户时未指定该选项，则此命令会提示您输入一个密码。该选项仅对 create 子命令有效。
必须按以下格式在单独的行上指定用户密码：
user:password
密码不能包含以下字符，也不能包含空格：

- ; (分号)
- : (冒号)
- \ (反斜杠)
- \n (换行符)

-i {- | *clconfigfile*}
--input {- | *clconfigfile*}
指定要用于验证或修改 SNMP 主机配置的配置信息。此信息必须符合 [clconfiguration\(5CL\) \[1245\]](#) 手册页中定义的格式。此信息可包含在文件中或者通过标准输入提供。要指定标准输入，请指定减号 (-) 而不是文件名。

-l *secllevel*
--securitylevel *secllevel*
指定用户的安全性级别。为 *secllevel* 指定以下值之一：

- noAuthNoPriv
- AuthNoPriv
- authPriv

有关 SNMP 安全级别的更多信息，请参见 [Unresolved link to " snmpcmd1" 手册页](#)。

`-n node[,...]`

`--node[s] node[,...]`

指定一个节点或一系列节点。可以使用节点名称或节点 ID 指定每个节点。

该命令的所有格式都接受此选项。

`-o { - | clconfigfile }`

`--output { - | clconfigfile }`

以 [clconfiguration\(5CL\) \[1245\]](#) 手册页描述的格式写入群集 SNMP 主机配置信息。可以将此信息写入文件或标准输出中。

要写入标准输出，请指定减号 (-) 而不是文件名。如果指定标准输出，将抑制命令的所有其他标准输出。

`-V`

`--version`

显示该命令的版本。

不要将该选项与子命令、操作数或其他选项一起指定，因为子命令、操作数或其他选项会被忽略。-V 选项只显示命令的版本。不会执行任何其他操作。

`-v`

`--verbose`

显示详细消息和信息。

可以将该选项与命令的任何格式一同指定，尽管一些子命令可能不会生成扩展输出。例如，如果指定详细选项，`export` 子命令不会生成扩展输出。

支持以下操作数：

`+` 指定所有 SNMP 用户。

`user` 指定 SNMP 用户的名称。

如果命令对于所有指定的操作数均成功执行，则返回零 (CL_NOERR)。如果某个操作数出错，命令会继续处理操作数列表中的下一个操作数。返回的退出代码始终反映最先发生的错误。

此命令返回下列退出状态代码：

`0 CL_NOERR`

没有错误

您发出的命令已成功完成。

1 CL_ENOMEM

交换空间不足

某个群集节点耗尽了交换内存或其他操作系统资源。

3 CL_EINVAL

参数无效

您键入的命令不正确，或者使用 `-i` 选项提供的群集配置信息语法不正确。

6 CL_EACCESS

权限被拒绝

指定的对象不可访问。要发出命令，可能需要拥有超级用户或 RBAC 访问权限。有关更多信息，请参见 [Unresolved link to "su1M"](#) 和 [Unresolved link to "rbac5"](#) 手册页。

18 CL_EINTERNAL

遇到内部错误

内部错误表示软件缺陷或其他缺陷。

35 CL_EIO

I/O 错误

发生了物理输入/输出错误。

36 CL_ENOENT

没有这样的对象

由于以下某个原因，找不到您所指定的对象：

- 该对象不存在。
- 您尝试使用 `-o` 选项创建的配置文件路径中的某个目录不存在。
- 您尝试使用 `-i` 选项访问的配置文件包含错误。

例 232 创建 SNMPv3 用户

以下命令可创建新用户 `newuser1`，并将该用户添加到当前节点上的配置中。验证类型为 SHA。

```
# clsnmpuser create -a SHA newuser1
Enter password for user 'newuser1':
```

此示例要求您输入要创建的用户密码。要自动执行该过程，请使用 `-f` 选项。

例 233 列出用户

以下命令可列出验证类型为 MD5 的所有用户。

```
# clsnmpuser list -a MD5 +
user1
mySNMPusername
```

加号 (+) 是可选的，因为它是默认选项。

例 234 显示用户

以下命令可显示当前节点上所有用户的用户信息。

```
# clsnmpuser show

--- SNMP User Configuration on phys-schost-1 ---

SNMP User Name:                newuser1
Authentication Protocol:      SHA
Default User:                  Yes
Default Security Level:       authPriv
```

例 235 更改用户的验证协议和状态

以下命令可修改用户 newuser1 的验证协议和默认用户状态。

```
# clsnmpuser set -a MD5 newuser1
```

例 236 删除 SNMP 用户

以下命令可删除所有 SNMP 用户。

```
# clsnmpuser delete +
```

此示例中使用加号 (+) 指定所有用户。

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

[clsnmp\(1CL\) \[409\]](#)、[clsnmpmib\(1CL\) \[417\]](#)、[cluster\(1CL\) \[461\]](#)、[Intro\(1CL\) \[17\]](#)、[sceventmib\(1M\) \[68\]](#)
[link to "snmpcmd1"](#)、[Unresolved link to "su1M"](#)、[scha_calls\(3HA\) \[897\]](#)、[Unresolved link to "attributes5"](#)、[Unresolved link to "rbac5"](#)、[clconfiguration\(5CL\) \[1245\]](#)

超级用户可以运行此命令的所有格式。

所有用户都可以运行带有 -? (帮助) 或 -v (版本) 选项的此命令。

要运行带有其他子命令的 `clsnmpmib` 命令，超级用户以外的用户需要具有 RBAC 授权。请参见下表。

子命令	RBAC 授权
<code>create</code>	<code>solaris.cluster.modify</code>
<code>delete</code>	<code>solaris.cluster.modify</code>
<code>export</code>	<code>solaris.cluster.read</code>
<code>list</code>	<code>solaris.cluster.read</code>
<code>set</code>	<code>solaris.cluster.modify</code>
<code>set-default</code>	<code>solaris.cluster.modify</code>
<code>show</code>	<code>solaris.cluster.read</code>

名称

cltelemetryattribute, chta — 配置系统资源监视

```
/usr/cluster/bin/cltelemetryattribute -V

/usr/cluster/bin/cltelemetryattribute [Subcommand] -?

/usr/cluster/bin/cltelemetryattribute subcommand [options] -v
    [telemetry-attribute ...]

/usr/cluster/bin/cltelemetryattribute disable [-i
    {- | clconfigfile}] [-t object-type] {+ | telemetry-attribute ...}

/usr/cluster/bin/cltelemetryattribute enable [-i
    {- | clconfigfile}] [-t object-type] {+ | telemetry-attribute ...}

/usr/cluster/bin/cltelemetryattribute export [-o
    {- | clconfigfile}] [-t object-type[,...]]
    [+ | telemetry-attribute ...]

/usr/cluster/bin/cltelemetryattribute list [-t object-type[,...]]
    [+ | telemetry-attribute ...]

/usr/cluster/bin/cltelemetryattribute print [-b object-instance[,...]]
    [-a] [-d period] [-u] [-n node[,...]] [-t object-type[,...]]
    [+ | telemetry-attribute ...]

/usr/cluster/bin/cltelemetryattribute set-threshold -b
    object-instance [-n node] {-p name=value} [-p name=value]
    [...] -t object-type telemetry-attribute

/usr/cluster/bin/cltelemetryattribute show [-b object-instance[,...]]
    [-n node[,...]] [-t object-type[,...]]
    [+ | telemetry-attribute ...]

/usr/cluster/bin/cltelemetryattribute status -b object-instance
    [-n node] {-p name} -t object-type [+ | telemetry-attribute ...]
```

此命令配置系统资源的监视。

您可以针对不同类型的对象监视系统资源的使用情况，其中包括以下对象：

- 磁盘
- 文件系统
- IP 地址
- 网络接口
- 节点
- Solaris 区域

- 资源组

您可以监视的系统资源的方面称为遥测属性。

该命令执行以下操作：

- 启用或禁用遥测属性
- 设置或修改遥测属性的阈值
- 显示一个列表，其中包括被监视的属性、应用的阈值以及收集的有关对象的数据

通过标识对应的遥测属性，选择您要监视的系统资源使用情况的方面。要针对某个对象监视系统资源使用情况，请对该类型的对象启用相应的遥测属性。Oracle Solaris Cluster 软件在群集中针对该类型的所有对象收集这些属性的使用情况数据。

对于系统资源，某个特定值可能对群集的性能很关键。可以设置遥测属性的阈值，以便在超过临界值时可以通知您。有关阈值的信息，请参见 `set-threshold` 子命令和 `-p` 选项的描述。

仅当 `options` 是 `-?` 选项或 `-v` 选项时，才可以省略 `subcommand`。

每个选项都有长和短两种格式。在 `选项` 中给出每个选项的两种格式以及该选项的描述。

`clta` 命令是 `cltelemetryattribute` 命令的简短格式。

在完善系统资源监视的配置之前，必须对监视进行初始化。请参见 [sctelemetry\(1M\) \[763\] 手册页](#)。

仅可以在全局区域中使用此命令。

子命令

支持以下子命令：

`disable`

对指定的对象类型禁用指定的遥测属性。

仅可以在全局区域中运行该子命令。

Oracle Solaris Cluster 软件收集设置为 `enabled` 状态的系统资源的使用情况数据。如果将某个对象类型的系统资源设置为 `disabled` 状态，Oracle Solaris Cluster 软件不会收集与该对象实例对应的任何实例的数据。

满足以下两个条件时，`cltelemetryattribute` 命令也会禁用这些属性的数据收集：

- 使用 `-i` 选项指定配置文件。
- 在输入文件中将遥测属性设置为 `disabled`。

通过使用 `export` 子命令创建配置文件。

将遥测属性设置为 `disabled` 时，其已配置的阈值设置保持不变。

超级用户以外的用户需要具有 `solaris.cluster.modify` 基于角色的访问控制 (Role-Based Access Control, RBAC) 授权才能使用该子命令。请参见 [Unresolved link to "rbac5" 手册页](#)。

enable

对指定的对象类型启用指定遥测属性的数据收集。

仅可以在全局区域中运行该子命令。

默认情况下，为选定的对象类型启用选定的属性。

要启用遥测属性的数据收集，请将遥测属性设置为 `enabled`。

Oracle Solaris Cluster 软件只收集已为其启用遥测属性的对象类型的数据。启用对象类型的属性时，Oracle Solaris Cluster 软件会针对所有节点上该类型的所有对象实例收集该属性的数据。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5" 手册页](#)。

export

将对象类型和对象实例的遥测属性配置导出到一个文件或标准输出 (stdout) 中。

仅可以在全局区域中运行该子命令。

配置包括属性对于某个对象类型是处于启用还是禁用状态。配置还可能包括为阈值配置设置的限制。

通过使用 `-o` 选项指定一个文件，以将配置信息写入文件。如果不指定 `-o` 选项，`cltelemetryattribute` 命令会将配置信息写入标准输出 (stdout)。

`export` 子命令不会修改群集配置数据。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5" 手册页](#)。

list

显示可以为指定的对象类型配置的遥测属性。

仅可以在全局区域中运行该子命令。

如果指定详细选项 `-v`，则 `list` 子命令显示可将属性应用到的对象类型。

阈值属性的显示格式如下：

```
Threshold: severity, direction  
          , value, rearm
```

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5" 手册页](#)。

print

显示对指定对象实例或对象类型启用的指定遥测属性的系统资源使用情况。

仅可以在全局区域中运行该子命令。

输出包含以下数据：

- 日期和时间戳
- 对象实例
- 对象类型
- 遥测属性
- 节点
- 值

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

set-threshold

针对节点上的指定对象修改指定遥测属性的阈值设置。

仅可以在全局区域中运行该子命令。

使用 `-p` 选项指定要修改的阈值。也可以使用 `-p` 选项指定要修改的阈值属性。只能修改 `value` 和 `rearm` 阈值属性。

必须为指定的阈值更改这些属性中的至少一个。如果至少配置了一个属性，则运行 `status` 子命令时将显示输出。

要取消激活阈值，请将 `value` 和 `rearm` 指定为空白，如下所示：

```
-y value=,rearm=
```

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

show

显示为对象类型或对象实例上的遥测特性配置的属性。

仅可以在全局区域中运行该子命令。

这些属性包括是否为对象类型启用系统资源。如果指定详细选项 `-v`，则 `show` 子命令会显示为对象实例启用的遥测属性的阈值设置。

阈值属性的显示格式如下：

```
Threshold: severity, direction  
, value, rearm
```

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

status

显示在标准输出上配置了阈值的对象类型的当前状态。如果没有至少设置一个属性，则运行 `status` 子命令时不会显示输出。

仅可以在全局区域中运行该子命令。

在不带参数的情况下使用此子命令可显示当前具有警告或致命严重级别的所有活动阈值状态。可能的阈值输出包括阈值的当前严重级别。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

支持以下选项：

`-?`

`--help`

显示帮助信息。

指定此选项时，*subcommand* 可有可无。

如果没有随该选项指定 *subcommand*，将会显示所有可用子命令的列表。

如果随该选项指定了 *subcommand*，将会显示 *subcommand* 的用法。

如果随该选项指定了 `set-threshold` 子命令，将会显示所有资源组属性的帮助信息。

如果是与其他选项、子命令或操作数一起指定该选项，将忽略所有这些内容。不会发生任何其他处理。

`-a`

`--average`

输出三小时期间内收集的数据的平均值，以及与该平均值关联的标准偏差。

如果一起指定 `-a` 选项和 `-d` 选项，则显示在指定 *period* 中三小时间隔内的平均数据。

如果未指定 `-a` 选项，则输出的数据为最新数据。

`-b object-instance`

`--object-instance=object-instance`

`--object-instance object-instance`

指定要显示其信息，或要为其设置阈值的对象实例。

对象实例始终为特定类型。例如，群集节点 `phys-schost-1` 是 `node` 类型的对象实例。仅当为对象类型启用相应的遥测属性时，Oracle Solaris Cluster 软件才监视对象实例的系统资源。

`-d period`

`--date-range=period`

`--date-range period`

指定要让 Oracle Solaris Cluster 软件收集监视数据的时间段。

为 *period* 参数指定的日期和时间格式必须符合国际标准化组织 (International Organization for Standardization, ISO) 8601 国际日期格式。

begin-time, end-time] 时间段为逗号 (,) 分隔的两个时间之间的时段。

begin-time+ 时间段为指定的开始时间与当前时间之间的时段。

`end-time` - 时间段为 Oracle Solaris Cluster 软件启动并开始收集数据与指定的结束时间之间的时段。

`period` 格式示例如下所示：

`-d 2006-04-30T18:00,2006-06-16T18:00`

从 2006 年 4 月 30 日下午 6:00 到 2006 年 6 月 16 日下午 6:00

`-d 2006-06-16+`

从 2006 年 6 月 16 日午夜 12:00 起

`-d 2006-07-31T18:00+`

从 2006 年 7 月 31 日下午 6:00 起

`-d 2006-06-16T18:00-`

从 Oracle Solaris Cluster 软件启动开始到 2006 年 6 月 16 日下午 6:00

`-d 2006-05-31T12:00,2006-06-16T11:59`

从 2006 年 5 月 31 日午夜 12:00 到 2006 年 6 月 16 日晚上 11:59

只能将该选项与 `print` 子命令一起使用。

`-i {- | clconfigfile}`

`--input={- | clconfigfile}`

`--input {- | clconfigfile}`

指定您要使用位于 `clconfigfile` 文件中的配置信息来指定遥测属性和阈值配置。请参见 [clconfiguration\(5CL\) \[1245\]](#) 手册页。

随该选项指定一个破折号 (-) 将通过标准输入 (stdin) 提供配置信息。如果指定了其他选项，则这些选项优先于 `clconfigfile` 中的选项和信息。

`-n node`

`--node=node`

`--node node`

指定 Oracle Solaris Cluster 在其上收集使用情况数据的节点名称。您可以指定一个名称或节点标识符。

对 `node` 类型的对象实例、资源或资源组指定子命令时，请勿使用 `-n` 选项。

`-o {- | clconfigfile}`

`--output={- | clconfigfile}`

`--output {- | clconfigfile}`

将遥测属性和阈值配置数据写入文件或标准输出

(stdout)。[clconfiguration\(5CL\) \[1245\]](#) 手册页中描述了该配置信息的格式。

如果您使用该选项指定文件名称，该选项将创建新的文件。然后将配置信息置于该文件中。如果您使用该选项指定 `-`，则会将配置信息发送到标准输出 (stdout)。命令的所有其他标准输出将受到抑制。

仅可以将此选项与 `export` 子命令一起使用。

`-p name`
`--property=name`
`--property name`

指定 `status` 子命令的属性列表。

有关您可以使用 `set-threshold` 子命令来为其设置阈值的属性的信息，请参见 `-p name=value` 选项的描述。

`-p name=value`
`--property=name=value`
`--property name value`

指定阈值的属性。

允许使用多个 `-p name= value` 实例。

有关可以使用 `status` 子命令显示其相关信息的属性的信息，请参见 `-p name` 选项的描述。

对于每个阈值，必须指定 `severity` 属性和 `direction` 属性来标识阈值。设置阈值后将无法修改这些属性。

为每个阈值设置一个 `value`。还可以为每个阈值设置一个 `rearm`。使用 `set-threshold` 子命令来修改 `value` 和 `rearm` 属性。可以使用此选项指定的属性和值如下所示：

`severity`

阈值的严重级别。您可以为该属性设置的可能值是 `fatal` 和 `warning`。严重级别为 `fatal` 的阈值比严重级别为 `warning` 的阈值更严重。

在 Oracle Solaris Cluster Manager 中，严重级别显示为可视报警。

`direction`

将应用的阈值的方向。您可以为该属性设置的可能值是 `falling` 和 `rising`。通过将 `direction` 属性设置为 `falling`，您指定 `fatal` 严重级别具有比 `warning` 严重级别低的值。通过将 `direction` 属性设置为 `rising`，您指定 `fatal` 严重级别具有比 `warning` 严重级别高的值。

`value`

您要为遥测属性的阈值设置的值。如果超过该阈值，则遥测属性的严重级别会发生更改。最多可以将四个阈值与对象的特定遥测属性关联。

使用 `set-threshold` 子命令来设置或修改 `value` 属性。

`rearm`

清除遥测属性严重级别的方法。通过指定 `rearm` 值，当遥测特性的值超过与 `direction` 属性中的设置相反的方向中的 `rearm` 值时，清除遥测特性的严重级别。如果不指定 `rearm` 值，则等同于将阈值与 `rearm` 值设置为相同的值。

通知的频率遵循滞后原则，即，频率由双值函数决定。当函数增加时，应用一个值。当函数与另一个值相同时，应用该值。

设置 `rearm` 和 `value` 的值来满足系统需要。如果不指定可选的 `rearm` 属性，则将 `value` 视为默认值。但是，如果将 `rearm` 属性设置为与 `value` 属性相同的值，或者，如果未将值分配给 `rearm`，则监视的遥测特性值每次高于或低于为 `value` 设置的值时，将收到一条通知。为了避免收到大量的通知，请将 `rearm` 设置为 `value` 以外的其他值。

如果使用 `set-threshold` 子命令指定 `rearm`，则 `cltelemetryattribute` 命令可确保 `rearm` 的值符合以下要求：

- 如果 `direction` 为 `rising`，则 `value` 具有大于或等于 `rearm` 的值。
- 如果 `direction` 为 `falling`，则 `value` 具有小于或等于 `value` 的值。

使用 `set-threshold` 子命令来更改 `rearm`。

```
-t object-type
--object-type=object-type
--object-type object-type
```

指定 Oracle Solaris Cluster 软件将收集其使用情况数据的对象类型。所有对象实例都为特定类型。

使用此选项可将子命令的输出限制为指定类型的对象。

您可以为其监视系统资源的对象类型和与每个对象类型关联的遥测属性如下所示：

对象类型	描述	遥测属性
disk	磁盘	rbyte.rate、wbyte.rate、read.rate、write.rate、
filesystem	文件系统	block.used、inode.used
ipaddr	IP 地址	ipacket.rate、opacket.rate
netif	网络接口	ipacket.rate、opacket.rate、rbyte.rate、wbyte.rate
node	节点	cpu.idle、cpu.iowait、cpu.used、loadavg.1mn、loadavg.5mn、loadavg.15mn、mem.used、mem.free、swap.used、swap.free
resourcegroup	资源组	cpu.used、mem.used、swap.used
zone	Zone	cpu.idle、cpu.iowait、cpu.used、loadavg.1mn、loadavg.5mn、loadavg.15mn

您可以监视的遥测属性如下所示：

遥测属性	描述
block.used	设备上使用的块的百分比
cpu.idle	空闲 CPU 的数量

遥测属性	描述
cpu.iowait	等待输入/输出完成的 CPU 的数量
cpu.used	使用的 CPU 的数量
inode.used	设备上使用的 inode 的百分比
ipacket.rate	每秒传入的包数量
loadavg.1mn	在过去的一分钟内等待 CPU 的进程数量
loadavg.5mn	在过去的五分钟内等待 CPU 的进程数量
loadavg.15mn	在过去的十五分钟内等待 CPU 的进程数量
mem.free	空闲内存的兆字节数
mem.used	使用的内存的兆字节数
opacket.rate	每秒传出的包数量
rbyte.rate	每秒读取的兆位数
read.rate	每秒读取操作的数量
swap.free	空闲交换内存的兆字节数
swap.used	使用的交换内存的兆字节数
wbyte.rate	每秒写入的兆位数
write.rate	每秒写入操作的数量

您并不能监视上表中针对所有对象类型列出的所有遥测属性。使用 `list` 子命令来显示您可以收集其数据的对象类型，以及对每种类型的对象监视的遥测属性。

`-u`
`--utc`

以国际协调时间 (Coordinated Universal Time, UTC) 或格林尼治标准时间 (Greenwich Mean Time, GMT) 显示与使用情况数据一起显示的日期和时间。通过指定此选项，可以绕过日期和时间与本地日期和时间之间的相互转换。默认情况下，Oracle Solaris Cluster 软件显示本地日期和时间。

只能将该选项与 `print` 子命令一起使用。

`-v`
`--verbose`

在标准输出 (stdout) 中显示详细信息。

`-V`
`--version`

显示该命令的版本。

如果是与其他选项、子命令或操作数一起指定该选项，将忽略所有这些内容。仅显示该命令的版本。不会发生任何其他处理。

支持以下操作数：

telemetry-attribute

需要其使用情况数据的特定遥测属性。

Oracle Solaris Cluster 软件包含可收集其使用情况数据的特定对象类型。对于每个对象类型，您可以启用遥测属性监视。Oracle Solaris Cluster 软件只收集已启用属性的数据。

+

所有遥测组。

[Intro\(1CL\) \[17\]](#) 手册页列出了该命令集中所有命令的一整套退出状态代码。

如果命令对于所有指定的操作数均成功执行，则返回零 (CL_NOERR)。如果某个操作数出错，命令会继续处理操作数列表中的下一个操作数。返回的退出代码始终反映最先发生的错误。

此命令返回下列退出状态代码：

0 CL_NOERR

没有错误

您发出的命令已成功完成。

1 CL_ENOMEM

交换空间不足

某个群集节点耗尽了交换内存或其他操作系统资源。

3 CL_EINVAL

参数无效

您键入的命令不正确，或者使用 `-i` 选项提供的群集配置信息语法不正确。

6 CL_EACCESS

权限被拒绝

指定的对象不可访问。要发出命令，可能需要拥有超级用户或 RBAC 访问权限。有关更多信息，请参见 [Unresolved link to "su1M"](#) 和 [Unresolved link to "rbac5"](#) 手册页。

18 CL_EINTERNAL

遇到内部错误

内部错误表示软件缺陷或其他缺陷。

35 CL_EIO

I/O 错误

发生了物理输入/输出错误。

36 CL_ENOENT

没有这样的对象

由于以下某个原因，找不到您所指定的对象：

- 该对象不存在。
- 您尝试使用 -o 选项创建的配置文件路径中的某个目录不存在。
- 您尝试使用 -i 选项访问的配置文件包含错误。

38 CL_EBUSY

对象忙

您尝试将一根电缆从上一个群集互连路径移到一个活动群集节点。或者，您尝试将一个节点从尚未删除引用的群集配置中移除。

39 CL_EEXIST

对象已存在

您所指定的设备、设备组、群集互连组件、节点、群集、资源、资源类型、资源组或专用字符串已存在。

41 CL_ETYPE

未知类型

您使用 -t 或 -p 选项指定的类型不存在。

例 237 显示为对象类型配置的系统资源

以下命令显示适用于对象类型（在本例中为磁盘）的系统资源。

```
# cltelemetryattribute list -t disk
rbyte.rate
wbyte.rate
write.rate
read.rate
```

例 238 启用对象类型的遥测属性

以下命令在群集中的所有磁盘上启用指定遥测属性的数据收集。

```
# cltelemetryattribute enable -t disk rbyte.rate wbyte.rate
```

例 239 设置对象类型的遥测属性的阈值

以下命令在群集中的磁盘 d4 上设置遥测属性 wbyte.rate 的阈值。rearm 的默认值设置为值 value。因此，当写入磁盘 d4 的字节数超过或低于 100 时，Oracle Solaris Cluster 软件将发出致命通知。

```
# cltelemetryattribute set-threshold -t disk -b d4 \
-p severity=fatal,direction=rising,value=100 wbyte.rate
```

例 240 显示已配置遥测属性的非详细列表

以下命令显示在群集中的所有磁盘上配置的遥测属性的非详细列表。

```
# cltelemetryattribute show -t disk

=== Telemetry Attributes ===

Telemetry Attribute:          read.rate
Unit:                          read/s
Enabled Object Types:         disk

Telemetry Attribute:          write.rate
Unit:                          writes/s
Enabled Object Types:         disk

Telemetry Attribute:          wbyte.rate
Unit:                          KBytes/s
Enabled Object Types:         disk

Telemetry Attribute:          rbyte.rate
Unit:                          KBytes/s
Enabled Object Types:         disk
```

例 241 显示遥测属性的配置の詳細列表

以下命令显示在群集中的所有磁盘上配置的遥测属性的详细列表。

```
# cltelemetryattribute show -v -t disk

=== Telemetry Attributes ===

Telemetry Attribute:          read.rate
Unit:                          read/s
Enabled Object Types:         disk

Telemetry Attribute:          write.rate
Unit:                          writes/s
Enabled Object Types:         disk

Telemetry Attribute:          wbyte.rate
Unit:                          KBytes/s
Enabled Object Types:         disk

Telemetry Attribute:          rbyte.rate
Unit:                          KBytes/s
Enabled Object Types:         disk

% cltelemetryattribute show -v -t disk

=== Telemetry Attributes ===

Telemetry Attribute:          read.rate
```

```

Unit: read/s
Enabled Object Types: disk

Telemetry Attribute: write.rate
Unit: writes/s
Enabled Object Types: disk

--- Object Instances of Type "disk" ---

Object Instance: d4
Thresholds: <Direction, Severity, Value, Rearm>
Threshold 1: <rising, fatal, 1000, 500>

Telemetry Attribute: wbyte.rate
Unit: KBytes/s
Enabled Object Types: disk

Telemetry Attribute: rbyte.rate
Unit: KBytes/s
Enabled Object Types: disk

```

例 242 显示遥测属性的状态

以下命令显示在群集中的所有磁盘上配置的遥测属性的状态。

```

# cltelemetryattribute status

=== Telemetry Attributes Thresholds ===

Attribute  Obj-Instance  Obj-Type  Node  Threshold  Status
-----
mem.used   phys-schost-1  node      16-v2-4  <rising, fatal, 1000, 1000>  warning

```

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

[cluster\(1CL\) \[461\]](#)、[Intro\(1CL\) \[17\]](#)、[sctelemetry\(1M\) \[763\]](#)、[Unresolved link to "su1M"](#)、[Unresolved link to "attributes5"](#)、[Unresolved link to "rbac5"](#)、[SUNW.SCTelemetry\(5\) \[1229\]](#)、[clconfiguration\(5CL\) \[1245\]](#)

超级用户可以运行此命令的所有格式。

所有用户都可以运行带有 `-?` (帮助) 或 `-v` (版本) 选项的此命令。

要运行带有其他子命令的 `cltelemetryattribute` 命令，超级用户以外的用户需要具有 RBAC 授权。请参见下表。

子命令	RBAC 授权
<code>disable</code>	<code>solaris.cluster.modify</code>
<code>enable</code>	<code>solaris.cluster.modify</code>
<code>export</code>	<code>solaris.cluster.read</code>
<code>list</code>	<code>solaris.cluster.read</code>
<code>print</code>	<code>solaris.cluster.read</code>
<code>set-threshold</code>	<code>solaris.cluster.modify</code>
<code>show</code>	<code>solaris.cluster.read</code>
<code>status</code>	<code>solaris.cluster.read</code>

名称

cltelemetryattribute, chta — 配置系统资源监视

```
/usr/cluster/bin/cltelemetryattribute -V

/usr/cluster/bin/cltelemetryattribute [Subcommand] -?

/usr/cluster/bin/cltelemetryattribute subcommand [options] -v
    [telemetry-attribute ...]

/usr/cluster/bin/cltelemetryattribute disable [-i
    {- | clconfigfile}] [-t object-type] {+ | telemetry-attribute ...}

/usr/cluster/bin/cltelemetryattribute enable [-i
    {- | clconfigfile}] [-t object-type] {+ | telemetry-attribute ...}

/usr/cluster/bin/cltelemetryattribute export [-o
    {- | clconfigfile}] [-t object-type[,...]]
    [+ | telemetry-attribute ...]

/usr/cluster/bin/cltelemetryattribute list [-t object-type[,...]]
    [+ | telemetry-attribute ...]

/usr/cluster/bin/cltelemetryattribute print [-b object-instance[,...]]
    [-a] [-d period] [-u] [-n node[,...]] [-t object-type[,...]]
    [+ | telemetry-attribute ...]

/usr/cluster/bin/cltelemetryattribute set-threshold -b
    object-instance [-n node] {-p name=value} [-p name=value]
    [...] -t object-type telemetry-attribute

/usr/cluster/bin/cltelemetryattribute show [-b object-instance[,...]]
    [-n node[,...]] [-t object-type[,...]]
    [+ | telemetry-attribute ...]

/usr/cluster/bin/cltelemetryattribute status -b object-instance
    [-n node] {-p name} -t object-type [+ | telemetry-attribute ...]
```

此命令配置系统资源的监视。

您可以针对不同类型的对象监视系统资源的使用情况，其中包括以下对象：

- 磁盘
- 文件系统
- IP 地址
- 网络接口
- 节点
- Solaris 区域

- 资源组

您可以监视的系统资源的方面称为遥测属性。

该命令执行以下操作：

- 启用或禁用遥测属性
- 设置或修改遥测属性的阈值
- 显示一个列表，其中包括被监视的属性、应用的阈值以及收集的有关对象的数据

通过标识对应的遥测属性，选择您要监视的系统资源使用情况的方面。要针对某个对象监视系统资源使用情况，请对该类型的对象启用相应的遥测属性。Oracle Solaris Cluster 软件在群集中针对该类型的所有对象收集这些属性的使用情况数据。

对于系统资源，某个特定值可能对群集的性能很关键。可以设置遥测属性的阈值，以便在超过临界值时可以通知您。有关阈值的信息，请参见 `set-threshold` 子命令和 `-p` 选项的描述。

仅当 `options` 是 `-?` 选项或 `-v` 选项时，才可以省略 `subcommand`。

每个选项都有长和短两种格式。在 `选项` 中给出每个选项的两种格式以及该选项的描述。

`clta` 命令是 `cltelemetryattribute` 命令的简短格式。

在完善系统资源监视的配置之前，必须对监视进行初始化。请参见 [sctelemetry\(1M\) \[763\] 手册页](#)。

仅可以在全局区域中使用此命令。

子命令

支持以下子命令：

`disable`

对指定的对象类型禁用指定的遥测属性。

仅可以在全局区域中运行该子命令。

Oracle Solaris Cluster 软件收集设置为 `enabled` 状态的系统资源的使用情况数据。如果将某个对象类型的系统资源设置为 `disabled` 状态，Oracle Solaris Cluster 软件不会收集与该对象实例对应的任何实例的数据。

满足以下两个条件时，`cltelemetryattribute` 命令也会禁用这些属性的数据收集：

- 使用 `-i` 选项指定配置文件。
- 在输入文件中将遥测属性设置为 `disabled`。

通过使用 `export` 子命令创建配置文件。

将遥测属性设置为 `disabled` 时，其已配置的阈值设置保持不变。

超级用户以外的用户需要具有 `solaris.cluster.modify` 基于角色的访问控制 (Role-Based Access Control, RBAC) 授权才能使用该子命令。请参见 [Unresolved link to "rbac5" 手册页](#)。

enable

对指定的对象类型启用指定遥测属性的数据收集。

仅可以在全局区域中运行该子命令。

默认情况下，为选定的对象类型启用选定的属性。

要启用遥测属性的数据收集，请将遥测属性设置为 `enabled`。

Oracle Solaris Cluster 软件只收集已为其启用遥测属性的对象类型的数据。启用对象类型的属性时，Oracle Solaris Cluster 软件会针对所有节点上该类型的所有对象实例收集该属性的数据。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5" 手册页](#)。

export

将对象类型和对象实例的遥测属性配置导出到一个文件或标准输出 (`stdout`) 中。

仅可以在全局区域中运行该子命令。

配置包括属性对于某个对象类型是处于启用还是禁用状态。配置还可能包括为阈值配置设置的限制。

通过使用 `-o` 选项指定一个文件，以将配置信息写入文件。如果不指定 `-o` 选项，`cltelemetryattribute` 命令会将配置信息写入标准输出 (`stdout`)。

`export` 子命令不会修改群集配置数据。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5" 手册页](#)。

list

显示可以为指定的对象类型配置的遥测属性。

仅可以在全局区域中运行该子命令。

如果指定详细选项 `-v`，则 `list` 子命令显示可将属性应用到的对象类型。

阈值属性的显示格式如下：

```
Threshold: severity, direction  
          , value, rearm
```

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5" 手册页](#)。

print

显示对指定对象实例或对象类型启用的指定遥测属性的系统资源使用情况。

仅可以在全局区域中运行该子命令。

输出包含以下数据：

- 日期和时间戳
- 对象实例
- 对象类型
- 遥测属性
- 节点
- 值

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

set-threshold

针对节点上的指定对象修改指定遥测属性的阈值设置。

仅可以在全局区域中运行该子命令。

使用 `-p` 选项指定要修改的阈值。也可以使用 `-p` 选项指定要修改的阈值属性。只能修改 `value` 和 `rearm` 阈值属性。

必须为指定的阈值更改这些属性中的至少一个。如果至少配置了一个属性，则运行 `status` 子命令时将显示输出。

要取消激活阈值，请将 `value` 和 `rearm` 指定为空白，如下所示：

```
-y value=,rearm=
```

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

show

显示为对象类型或对象实例上的遥测特性配置的属性。

仅可以在全局区域中运行该子命令。

这些属性包括是否为对象类型启用系统资源。如果指定详细选项 `-v`，则 `show` 子命令会显示为对象实例启用的遥测属性的阈值设置。

阈值属性的显示格式如下：

```
Threshold: severity, direction  
, value, rearm
```

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

status

显示在标准输出上配置了阈值的对象类型的当前状态。如果没有至少设置一个属性，则运行 `status` 子命令时不会显示输出。

仅可以在全局区域中运行该子命令。

在不带参数的情况下使用此子命令可显示当前具有警告或致命严重级别的所有活动阈值状态。可能的阈值输出包括阈值的当前严重级别。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5" 手册页](#)。

支持以下选项：

-?

--help

显示帮助信息。

指定此选项时，*subcommand* 可有可无。

如果没有随该选项指定 *subcommand*，将会显示所有可用子命令的列表。

如果随该选项指定了 *subcommand*，将会显示 *subcommand* 的用法。

如果随该选项指定了 `set-threshold` 子命令，将会显示所有资源组属性的帮助信息。

如果是与其他选项、子命令或操作数一起指定该选项，将忽略所有这些内容。不会发生任何其他处理。

-a

--average

输出三小时期间内收集的数据的平均值，以及与该平均值关联的标准偏差。

如果一起指定 `-a` 选项和 `-d` 选项，则显示在指定 *period* 中三小时间隔内的平均数据。

如果未指定 `-a` 选项，则输出的数据为最新数据。

-b *object-instance*

--object-instance=*object-instance*

--object-instance *object-instance*

指定要显示其信息，或要为其设置阈值的对象实例。

对象实例始终为特定类型。例如，群集节点 `phys-schost-1` 是 `node` 类型的对象实例。仅当为对象类型启用相应的遥测属性时，Oracle Solaris Cluster 软件才监视对象实例的系统资源。

-d *period*

--date-range=*period*

--date-range *period*

指定要让 Oracle Solaris Cluster 软件收集监视数据的时间段。

为 *period* 参数指定的日期和时间格式必须符合国际标准化组织 (International Organization for Standardization, ISO) 8601 国际日期格式。

begin-time, *end-time*] 时间段为逗号 (,) 分隔的两个时间之间的时段。

begin-time+ 时间段为指定的开始时间与当前时间之间的时段。

end-time - 时间段为 Oracle Solaris Cluster 软件启动并开始收集数据与指定的结束时间之间的时段。

period 格式示例如下所示：

-d 2006-04-30T18:00,2006-06-16T18:00

从 2006 年 4 月 30 日下午 6:00 到 2006 年 6 月 16 日下午 6:00

-d 2006-06-16+

从 2006 年 6 月 16 日午夜 12:00 起

-d 2006-07-31T18:00+

从 2006 年 7 月 31 日下午 6:00 起

-d 2006-06-16T18:00-

从 Oracle Solaris Cluster 软件启动开始到 2006 年 6 月 16 日下午 6:00

-d 2006-05-31T12:00,2006-06-16T11:59

从 2006 年 5 月 31 日午夜 12:00 到 2006 年 6 月 16 日晚上 11:59

只能将该选项与 *print* 子命令一起使用。

-i {- | *clconfigfile*}

--input={- | *clconfigfile*}

--input {- | *clconfigfile*}

指定您要使用位于 *clconfigfile* 文件中的配置信息来指定遥测属性和阈值配置。请参见 [clconfiguration\(5CL\) \[1245\]](#) 手册页。

随该选项指定一个破折号 (-) 将通过标准输入 (stdin) 提供配置信息。如果指定了其他选项，则这些选项优先于 *clconfigfile* 中的选项和信息。

-n *node*

--node=*node*

--node *node*

指定 Oracle Solaris Cluster 在其上收集使用情况数据的节点名称。您可以指定一个名称或节点标识符。

对 *node* 类型的对象实例、资源或资源组指定子命令时，请勿使用 -n 选项。

-o {- | *clconfigfile*}

--output={- | *clconfigfile*}

--output {- | *clconfigfile*}

将遥测属性和阈值配置数据写入文件或标准输出

(stdout)。[clconfiguration\(5CL\) \[1245\]](#) 手册页中描述了该配置信息的格式。

如果您使用该选项指定文件名称，该选项将创建新的文件。然后将配置信息置于该文件中。如果您使用该选项指定 -，则会将配置信息发送到标准输出 (stdout)。命令的所有其他标准输出将受到抑制。

仅可以将此选项与 *export* 子命令一起使用。

`-p name`
`--property=name`
`--property name`

指定 `status` 子命令的属性列表。

有关您可以使用 `set-threshold` 子命令来为其设置阈值的属性的信息，请参见 `-p name=value` 选项的描述。

`-p name=value`
`--property=name=value`
`--property name value`

指定阈值的属性。

允许使用多个 `-p name= value` 实例。

有关可以使用 `status` 子命令显示其相关信息的属性的信息，请参见 `-p name` 选项的描述。

对于每个阈值，必须指定 `severity` 属性和 `direction` 属性来标识阈值。设置阈值后将无法修改这些属性。

为每个阈值设置一个 `value`。还可以为每个阈值设置一个 `rearm`。使用 `set-threshold` 子命令来修改 `value` 和 `rearm` 属性。可以使用此选项指定的属性和值如下所示：

`severity`

阈值的严重级别。您可以为该属性设置的可能值是 `fatal` 和 `warning`。严重级别为 `fatal` 的阈值比严重级别为 `warning` 的阈值更严重。

在 Oracle Solaris Cluster Manager 中，严重级别显示为可视报警。

`direction`

将应用的阈值的方向。您可以为该属性设置的可能值是 `falling` 和 `rising`。通过将 `direction` 属性设置为 `falling`，您指定 `fatal` 严重级别具有比 `warning` 严重级别低的值。通过将 `direction` 属性设置为 `rising`，您指定 `fatal` 严重级别具有比 `warning` 严重级别高的值。

`value`

您要为遥测属性的阈值设置的值。如果超过该阈值，则遥测属性的严重级别会发生更改。最多可以将四个阈值与对象的特定遥测属性关联。

使用 `set-threshold` 子命令来设置或修改 `value` 属性。

`rearm`

清除遥测属性严重级别的方法。通过指定 `rearm` 值，当遥测特性的值超过与 `direction` 属性中的设置相反的方向中的 `rearm` 值时，清除遥测特性的严重级别。如果不指定 `rearm` 值，则等同于将阈值与 `rearm` 值设置为相同的值。

通知的频率遵循滞后原则，即，频率由双值函数决定。当函数增加时，应用一个值。当函数与另一个值相同时，应用该值。

设置 `rearm` 和 `value` 的值来满足系统需要。如果不指定可选的 `rearm` 属性，则将 `value` 视为默认值。但是，如果将 `rearm` 属性设置为与 `value` 属性相同的值，或者，如果未将值分配给 `rearm`，则监视的遥测特性值每次高于或低于为 `value` 设置的值时，将收到一条通知。为了避免收到大量的通知，请将 `rearm` 设置为 `value` 以外的其他值。

如果使用 `set-threshold` 子命令指定 `rearm`，则 `cltelemetryattribute` 命令可确保 `rearm` 的值符合以下要求：

- 如果 `direction` 为 `rising`，则 `value` 具有大于或等于 `rearm` 的值。
- 如果 `direction` 为 `falling`，则 `value` 具有小于或等于 `value` 的值。

使用 `set-threshold` 子命令来更改 `rearm`。

```
-t object-type
--object-type=object-type
--object-type object-type
```

指定 Oracle Solaris Cluster 软件将收集其使用情况数据的对象类型。所有对象实例都为特定类型。

使用此选项可将子命令的输出限制为指定类型的对象。

您可以为其监视系统资源的对象类型和与每个对象类型关联的遥测属性如下所示：

对象类型	描述	遥测属性
disk	磁盘	rbyte.rate、wbyte.rate、read.rate、write.rate、
filesystem	文件系统	block.used、inode.used
ipaddr	IP 地址	ipacket.rate、opacket.rate
netif	网络接口	ipacket.rate、opacket.rate、rbyte.rate、wbyte.rate
node	节点	cpu.idle、cpu.iowait、cpu.used、loadavg.1mn、loadavg.5mn、loadavg.15mn、mem.used、mem.free、swap.used、swap.free
resourcegroup	资源组	cpu.used、mem.used、swap.used
zone	Zone	cpu.idle、cpu.iowait、cpu.used、loadavg.1mn、loadavg.5mn、loadavg.15mn

您可以监视的遥测属性如下所示：

遥测属性	描述
block.used	设备上使用的块的百分比
cpu.idle	空闲 CPU 的数量

遥测属性	描述
cpu.iowait	等待输入/输出完成的 CPU 的数量
cpu.used	使用的 CPU 的数量
inode.used	设备上使用的 inode 的百分比
ipacket.rate	每秒传入的包数量
loadavg.1mn	在过去的一分钟内等待 CPU 的进程数量
loadavg.5mn	在过去的五分钟内等待 CPU 的进程数量
loadavg.15mn	在过去的十五分钟内等待 CPU 的进程数量
mem.free	空闲内存的兆字节数
mem.used	使用的内存的兆字节数
opacket.rate	每秒传出的包数量
rbyte.rate	每秒读取的兆位数
read.rate	每秒读取操作的数量
swap.free	空闲交换内存的兆字节数
swap.used	使用的交换内存的兆字节数
wbyte.rate	每秒写入的兆位数
write.rate	每秒写入操作的数量

您并不能监视上表中针对所有对象类型列出的所有遥测属性。使用 `list` 子命令来显示您可以收集其数据的对象类型，以及对每种类型的对象监视的遥测属性。

`-u`
`--utc`

以国际协调时间 (Coordinated Universal Time, UTC) 或格林尼治标准时间 (Greenwich Mean Time, GMT) 显示与使用情况数据一起显示的日期和时间。通过指定此选项，可以绕过日期和时间与本地日期和时间之间的相互转换。默认情况下，Oracle Solaris Cluster 软件显示本地日期和时间。

只能将该选项与 `print` 子命令一起使用。

`-v`
`--verbose`

在标准输出 (stdout) 中显示详细信息。

`-V`
`--version`

显示该命令的版本。

如果是与其他选项、子命令或操作数一起指定该选项，将忽略所有这些内容。仅显示该命令的版本。不会发生任何其他处理。

支持以下操作数：

telemetry-attribute

需要其使用情况数据的特定遥测属性。

Oracle Solaris Cluster 软件包含可收集其使用情况数据的特定对象类型。对于每个对象类型，您可以启用遥测属性监视。Oracle Solaris Cluster 软件只收集已启用属性的数据。

+

所有遥测组。

[Intro\(1CL\) \[17\]](#) 手册页列出了该命令集中所有命令的一整套退出状态代码。

如果命令对于所有指定的操作数均成功执行，则返回零 (CL_NOERR)。如果某个操作数出错，命令会继续处理操作数列表中的下一个操作数。返回的退出代码始终反映最先发生的错误。

此命令返回下列退出状态代码：

0 CL_NOERR

没有错误

您发出的命令已成功完成。

1 CL_ENOMEM

交换空间不足

某个群集节点耗尽了交换内存或其他操作系统资源。

3 CL_EINVAL

参数无效

您键入的命令不正确，或者使用 `-i` 选项提供的群集配置信息语法不正确。

6 CL_EACCESS

权限被拒绝

指定的对象不可访问。要发出命令，可能需要拥有超级用户或 RBAC 访问权限。有关更多信息，请参见 [Unresolved link to "su1M"](#) 和 [Unresolved link to "rbac5"](#) 手册页。

18 CL_EINTERNAL

遇到内部错误

内部错误表示软件缺陷或其他缺陷。

35 CL_EIO

I/O 错误

发生了物理输入/输出错误。

36 CL_ENOENT

没有这样的对象

由于以下某个原因，找不到您所指定的对象：

- 该对象不存在。
- 您尝试使用 -o 选项创建的配置文件路径中的某个目录不存在。
- 您尝试使用 -i 选项访问的配置文件包含错误。

38 CL_EBUSY

对象忙

您尝试将一根电缆从上一个群集互连路径移到一个活动群集节点。或者，您尝试将一个节点从尚未删除引用的群集配置中移除。

39 CL_EEXIST

对象已存在

您所指定的设备、设备组、群集互连组件、节点、群集、资源、资源类型、资源组或专用字符串已存在。

41 CL_ETYPE

未知类型

您使用 -t 或 -p 选项指定的类型不存在。

例 243 显示为对象类型配置的系统资源

以下命令显示适用于对象类型（在本例中为磁盘）的系统资源。

```
# cltelemetryattribute list -t disk
rbyte.rate
wbyte.rate
write.rate
read.rate
```

例 244 启用对象类型的遥测属性

以下命令在群集中的所有磁盘上启用指定遥测属性的数据收集。

```
# cltelemetryattribute enable -t disk rbyte.rate wbyte.rate
```

例 245 设置对象类型的遥测属性的阈值

以下命令在群集中的磁盘 d4 上设置遥测属性 wbyte.rate 的阈值。rearm 的默认值设置为值 value。因此，当写入磁盘 d4 的字节数超过或低于 100 时，Oracle Solaris Cluster 软件将发出致命通知。

```
# cltelemetryattribute set-threshold -t disk -b d4 \
-p severity=fatal,direction=rising,value=100 wbyte.rate
```

例 246 显示已配置遥测属性的非详细列表

以下命令显示在群集中的所有磁盘上配置的遥测属性的非详细列表。

```
# cltelemetryattribute show -t disk

=== Telemetry Attributes ===

Telemetry Attribute:          read.rate
Unit:                          read/s
Enabled Object Types:         disk

Telemetry Attribute:          write.rate
Unit:                          writes/s
Enabled Object Types:         disk

Telemetry Attribute:          wbyte.rate
Unit:                          KBytes/s
Enabled Object Types:         disk

Telemetry Attribute:          rbyte.rate
Unit:                          KBytes/s
Enabled Object Types:         disk
```

例 247 显示遥测属性的配置の詳細列表

以下命令显示在群集中的所有磁盘上配置的遥测属性的详细列表。

```
# cltelemetryattribute show -v -t disk

=== Telemetry Attributes ===

Telemetry Attribute:          read.rate
Unit:                          read/s
Enabled Object Types:         disk

Telemetry Attribute:          write.rate
Unit:                          writes/s
Enabled Object Types:         disk

Telemetry Attribute:          wbyte.rate
Unit:                          KBytes/s
Enabled Object Types:         disk

Telemetry Attribute:          rbyte.rate
Unit:                          KBytes/s
Enabled Object Types:         disk

% cltelemetryattribute show -v -t disk

=== Telemetry Attributes ===

Telemetry Attribute:          read.rate
```

```

Unit: read/s
Enabled Object Types: disk

Telemetry Attribute: write.rate
Unit: writes/s
Enabled Object Types: disk

--- Object Instances of Type "disk" ---

Object Instance: d4
Thresholds: <Direction, Severity, Value, Rearm>
Threshold 1: <rising, fatal, 1000, 500>

Telemetry Attribute: wbyte.rate
Unit: KBytes/s
Enabled Object Types: disk

Telemetry Attribute: rbyte.rate
Unit: KBytes/s
Enabled Object Types: disk

```

例 248 显示遥测属性的状态

以下命令显示在群集中的所有磁盘上配置的遥测属性的状态。

```

# cltelemetryattribute status

=== Telemetry Attributes Thresholds ===

Attribute  Obj-Instance  Obj-Type  Node  Threshold  Status
-----
mem.used   phys-schost-1  node      16-v2-4  <rising, fatal, 1000, 1000>  warning

```

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

[cluster\(1CL\) \[461\]](#)、[Intro\(1CL\) \[17\]](#)、[sctelemetry\(1M\) \[763\]](#)、[Unresolved link to "su1M"](#)、[Unresolved link to "attributes5"](#)、[Unresolved link to "rbac5"](#)、[SUNW.SCTelemetry\(5\) \[1229\]](#)、[clconfiguration\(5CL\) \[1245\]](#)

超级用户可以运行此命令的所有格式。

所有用户都可以运行带有 `-?` (帮助) 或 `-v` (版本) 选项的此命令。

要运行带有其他子命令的 `cltelemetryattribute` 命令，超级用户以外的用户需要具有 RBAC 授权。请参见下表。

子命令	RBAC 授权
<code>disable</code>	<code>solaris.cluster.modify</code>
<code>enable</code>	<code>solaris.cluster.modify</code>
<code>export</code>	<code>solaris.cluster.read</code>
<code>list</code>	<code>solaris.cluster.read</code>
<code>print</code>	<code>solaris.cluster.read</code>
<code>set-threshold</code>	<code>solaris.cluster.modify</code>
<code>show</code>	<code>solaris.cluster.read</code>
<code>status</code>	<code>solaris.cluster.read</code>

名称

cluster — 管理群集的全局配置和状态

```
/usr/cluster/bin/cluster -V

/usr/cluster/bin/cluster [subcommand] -?

/usr/cluster/bin/cluster subcommand
    [options] -v [clustername ...]

/usr/cluster/bin/cluster check
    [-F] [-C checkid[,...]]|-E checkid[,...]]
    [-e explorerpath[,...]] [-j jarpath[,...]]
    [-k keyword[,...]] [-n node[,...]] [-o outputdir]
    [-s severitylevel] [clustername]

/usr/cluster/bin/cluster create -i {- | clconfigfile}
    [clustername]

/usr/cluster/bin/cluster export [-o {- | configfile}]
    [-t objecttype[,...]] [clustername]

/usr/cluster/bin/cluster monitor-heartbeat [-v] [clustername]

/usr/cluster/bin/cluster list [clustername]

/usr/cluster/bin/cluster list-checks [-F] [-K]
    [-C checkid[,...]]|-E checkid[,...]] [-j jar-path[,...]]
    [-o outputdir] [clustername]

/usr/cluster/bin/cluster list-cmds [clustername]

/usr/cluster/bin/cluster rename -c newclustername [clustername]

/usr/cluster/bin/cluster restore-netprops [clustername]

/usr/cluster/bin/cluster set {-p name=value} [-p name=value] [...]
    [clustername]

/usr/cluster/bin/cluster set-netprops {-p name=value}
    [-p name=value] [...] [clustername]

/usr/cluster/bin/cluster show [-t objecttype[,...]] [clustername]

/usr/cluster/bin/cluster show-netprops [clustername]

/usr/cluster/bin/cluster shutdown [-y] [-g graceperiod]
    [-m message] [clustername]

/usr/cluster/bin/cluster status [-t objecttype[,...]] [clustername]
```

cluster 命令显示和管理群集范围的配置和状态信息。此命令还可以关闭全局群集。

以下 cluster 子命令可以在区域群集中运行：

-
- `cluster show` – 列出区域群集、节点、资源组、资源类型和资源属性。
 - `cluster status` – 显示区域群集组件的状态。
 - `cluster shutdown` – 按顺序关闭区域群集。
 - `cluster list` – 显示区域群集的名称。
 - `cluster list-cmds` – 列出以下命令，支持在区域群集内使用这些命令：
 - `clnode`
 - `clreslogicalhostname`
 - `clresource`
 - `clresourcegroup`
 - `clresourcetype`
 - `clressharedaddress`
 - `cluster`

几乎所有与 `cluster` 命令一起使用的子命令都在群集模式下运行。您可以从群集的任何节点中运行这些子命令。但是，`create`、`set-netprops` 和 `restore-netprops` 子命令例外。您必须在非群集模式下运行这些子命令。

仅当 `options` 是 `-?` 选项或 `-v` 选项时，才可以省略 `subcommand`。

`cluster` 命令没有短格式。

每个选项都有长和短两种格式。在 选项 中给出每个选项的两种格式以及该选项的描述。

在全局区域中使用此命令。

支持以下子命令：

`check`

检查并报告是否正确配置了群集。

仅可以在全局区域中运行该子命令。

该子命令有以下三种模式：基本检查、交互式检查和功能检查。

- 基本检查在未指定 `-k interactive` 或 `-k functional` 关键字时运行。基本检查读取并评估某些配置信息，以确定可能的错误或未满足的要求。
- 交互式检查由 `-k interactive` 选项指定。如果未指定 `-C-E` 选项，则系统会运行所有可用的交互式检查。

交互式检查与基本检查类似，但交互式检查需要检查无法确定的用户的信息。例如，检查可能会提示用户指定固件版本。交互检查不会中断群集功能。
- 功能检查由 `-k functional -Ccheckid` 选项指定。`-k functional` 选项要求 `-C` 选项的功能检查的检查 ID 不得超过一个。`-E` 选项与 `-k functional` 选项一起使用时无效。

功能检查执行群集配置的特定功能或行为，例如触发故障转移或使节点发生紧急情况。这些检查需要用户输入提供某些群集配置信息（例如要故障转移到的节点），并确定是开始执行检查还是继续执行检查。

因为某些功能检查涉及到中断群集服务，所以在您阅读完检查的详细描述并确定是否要先使群集脱离生产环境之前，请勿启动任何功能检查。使用 `cluster list-checks -v -C checkID` 命令可以显示功能检查的完整描述。

当从运行的群集的活动成员发出该子命令时，该子命令会运行配置检查。这些检查可以确保群集满足成功运行该群集所需的最低要求。

当从没有作为活动群集成员运行的节点发出该子命令时，该子命令会在该节点上运行预安装检查。这些检查可以确定应当进行修复的漏洞，以做好准备安装群集并避免可能出现的可用性降低情况。

每个配置检查都会生成一组报告，这些报告保存在指定或默认的输出目录中。而每个报告都包含一个摘要，显示执行的检查总数以及失败次数（按严重级别分组）。

每个报告均以普通文本和 XML 这两种格式生成。XML 格式的 DTD 位于 `/usr/cluster/lib/cfgchk/checkresults.dtd` 文件中。生成的报告只有英文版本。

超级用户以外的用户需要具有 `solaris.cluster.read` 基于角色的访问控制 (Role-Based Access Control, RBAC) 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

create

通过使用存储在 `clconfigfile` 文件中的配置信息创建新的群集。[clconfiguration\(5CL\) \[1245\]](#) 手册页中描述了该配置信息的格式。

仅可以在全局区域中运行该子命令。

您必须在非群集模式下运行该子命令。还必须从尚未配置为群集一部分的主机运行该子命令。Oracle Solaris Cluster 软件必须已安装在将要成为群集的一部分的每个节点上。

如果没有指定群集名称，则该群集的名称会取自 `clconfigfile` 文件。

超级用户以外的用户需要具有 `solaris.cluster.modify` 基于角色的访问控制 (Role-Based Access Control, RBAC) 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

export

导出配置信息。

仅可以在全局区域中运行该子命令。

如果使用 `-o` 选项指定一个文件，则会将配置信息写入该文件。如果未指定 `-o` 选项，则会将输出写入标准输出 (stdout)。

以下选项限制了导出的信息：

`-t objecttype[,...]`

仅导出指定类型的组件的配置信息。

仅可以导出在其上发出 `cluster` 命令的群集的配置信息。如果指定除了在其上发出 `cluster` 命令的群集以外的某群集的名称，该子命令会失败。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

list

显示群集的名称。

您可以在全局区域中或区域群集中使用该子命令。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

list-checks

显示一个列表，其中包含每个可用检查的检查 ID 和说明。

仅可以在全局区域中使用此命令。

检查 ID 以指明检查类型的字母作为开头。

F	功能检查
I	交互式检查
M	多个节点上的基本检查
S	单个节点上的基本检查

`-v` 选项显示检查操作的详细信息，其中包括检查的关键字。显示功能检查的详细描述很重要，这样可以确定是否在运行该检查之前从生产环境中删除该群集。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

list-cmds

显示所有可用 Oracle Solaris Cluster 命令的列表。

您可以在全局区域中或区域群集中使用该子命令。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

monitor-heartbeat

在动态重新配置 (Dynamic Reconfiguration, DR) 期间手动为群集节点重新启用心跳超时监视。

仅可以在全局区域中运行该子命令。独占 IP 区域群集不支持 `monitor-heartbeat` 子命令。

对 CPU 或内存板执行 DR 操作时，受影响的节点将变得无响应，所以将在所有其他节点上暂停对该节点的心跳监视。DR 完成后，将自动重新启用对该受影响节点的心跳监视。如果 DR 操作未完成，您可能需要使用 `monitor-heartbeat` 子命令手

动重新启用心跳监视。如果受影响的节点无法重新加入群集，将被取消群集成员资格。

有关重新启用心跳超时监视的说明，请参见[Unresolved link to " Oracle Solaris Cluster 4.2 Hardware Administration Manual 中的Kernel Cage Dynamic Reconfiguration Recovery"](#)。有关 DR 的一般信息，请参见[Unresolved link to " Oracle Solaris Cluster Concepts Guide 中的Dynamic Reconfiguration Support"](#)。

rename

重命名群集。

仅可以在全局区域中使用此命令。

将 -c 选项与该子命令一起使用可为群集指定新的名称。

注 - 如果您的群集配置为活动的 Oracle Solaris Cluster Geographic Edition 伙伴关系成员，请参见[Unresolved link to " Oracle Solaris Cluster Geographic Edition System Administration Guide 中的Renaming a Cluster That Is in a Partnership"](#)。本节介绍如何正确重命名配置为 Oracle Solaris Cluster Geographic Edition 伙伴关系成员的群集。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。请参见 [Unresolved link to " rbac5" 手册页](#)。

restore-netprops

重置群集的群集专用网络设置。

仅可以在全局区域中运行该子命令。您必须在非群集模式下运行该子命令。

仅当 `set-netprops` 子命令失败并存在以下条件时才使用该子命令：

- 您要尝试修改专用网络属性。
- 失败表明节点上的群集配置不一致。在这种情况下，需要运行 `restore-netprops` 子命令。

必须在群集的每个节点中运行该子命令。该子命令可以修复群集配置。该子命令还会删除因修改 IP 地址范围失败而导致的非一致性。如果失败，则为更改配置设置所做的任何尝试都不能保证正常进行。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。请参见 [Unresolved link to " rbac5" 手册页](#)。

set

修改群集的属性。

仅可以在全局区域中运行该子命令。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。请参见 [Unresolved link to " rbac5" 手册页](#)。

set-netprops

修改专用网络属性。

仅可以在全局区域中运行该子命令。

您必须在非群集模式下运行该子命令。但设置 `num_zoneclusters` 属性时，也可以在群集模式下运行该子命令。

超级用户以外的用户需要具有 `solaris.cluster.modify` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

show

显示有关群集组件的详细配置信息。

仅可以在全局区域中运行该子命令。

以下选项限制了显示的信息：

`-t objecttype[,...]`

仅显示指定类型的组件的配置信息。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

show-netprops

显示关于群集专用网络属性的信息。

仅可以在全局区域中运行该子命令。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

shutdown

按顺序关闭全局群集。

仅可以在全局区域中运行该子命令。

如果在全局群集中发出该命令，则 Oracle Solaris Cluster 软件会关闭整个全局群集，包括与该全局群集关联的所有区域群集。不能在区域群集中使用 `cluster` 命令。

如果提供除了在其上发出 `cluster` 命令的群集以外的某群集的名称，该子命令会失败。

仅从群集中的某个节点运行该子命令。

该子命令将执行以下操作：

- 使群集中正在运行的所有资源组脱机。如果任何转换失败，该子命令将无法完成任务并显示一条错误消息。
- 卸载所有群集文件系统。如果卸载失败，该子命令将无法完成任务并显示一条错误消息。
- 关闭所有活动的设备服务。如果设备的任何转换失败，该子命令将无法完成任务并显示一条错误消息。
- 停止群集中的所有节点。

在该子命令开始关闭群集之前，它会在所有节点上发出一条警告消息。发出警告之后，该子命令会发出一条最终消息，提示您确认是否要关闭该群集。要阻止发出该最终消息，请使用 `-y` 选项。

默认情况下，`shutdown` 子命令会在关闭群集之前等待 60 秒。可以使用 `-g` 选项指定不同的延迟时间。

要指定与警告一起显示的消息字符串，请使用 `-m` 选项。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

status

显示群集组件的状态。

您可以在全局区域中或区域群集中使用该子命令。

选项 `-t objecttype[...]` 仅显示指定类型的组件的状态信息。

超级用户以外的用户需要具有 `solaris.cluster.read` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

支持以下选项：

注 - 本节中介绍了每种选项的长和短两种格式。

`-?`

`--help`

显示帮助信息。

指定此选项时，*subcommand* 可有可无。

如果没有指定 *subcommand*，将显示所有可用子命令的列表。

如果指定了 *subcommand*，将显示该子命令的用法。

如果指定此选项和其他选项，将会忽略其他选项。

`-C checkid[...]`

`--checkID=checkid[...]`

`--checkID checkid[...]`

指定要运行的检查。系统不会运行未指定的检查。如果 `-E` 选项与 `-C` 选项一起指定，则 `-C` 选项会被忽略。

对于 `-k functional` 关键字，需要使用 `-C` 选项并且必须仅指定一个要运行的 *checkid* ID。

此选项只能与 `check` 和 `list-checks` 子命令一起使用。

`-c newclustername`

`--newclustername=newclustername`

`--newclustername newclustername`

为群集指定新的名称。

将此选项与 `rename` 子命令一起使用可更改群集的名称。

`-E checkid[,...]`
`--excludeCheckID=checkid[,...]`
`--excludeCheckID checkid[,...]`

指定要排除的检查。除了指定的那些检查，所有检查都在运行。如果 `-C` 选项与 `-E` 选项一起指定，则 `-C` 选项会被忽略。

`-E` 选项与 `-k functional` 关键字一起使用时无效。

此选项只能与 `check` 和 `list-checks` 子命令一起使用。

`-e explorerpath[,...]`
`--explorer=explorerpath[,...]`
`--explorer explorerpath[,...]`

指定到已解压缩的 Oracle Explorer 或 Sun Explorer 归档的路径，用作系统的备用数据源。`explorerpath` 的值必须是全限定路径位置。

此选项只能与 `check` 子命令一起使用。

`-F`
`--force`

忽略 `/var/cluster/logs/cluster_check/cfgchk.lck` 文件（如果存在），强制执行该子命令。仅当确定未运行 `check` 和 `list-checks` 子命令时，才使用该选项。

`-g graceperiod`
`--graceperiod=graceperiod`
`--graceperiod graceperiod`

在依据默认设置 60 秒关闭群集之前更改时间。

指定 `graceperiod`（秒）。

`-i {- | clconfigfile}`
`--input={- | clconfigfile}`
`--input {- | clconfigfile}`

使用 `clconfigfile` 文件中的配置信息。请参见 [clconfiguration\(5CL\) \[1245\]](#) 手册页。

要通过标准输入（`stdin`）提供配置信息，请使用破折号（`-`）指定该选项。

如果指定其他选项，则这些选项优先于群集配置文件中的选项和信息。

`-j jarpath[,...]`
`--jar=jarpath[,...]`
`--jar jarpath[,...]`

指定到包含检查的其他 `jar` 文件的路径。`jarpath` 必须是全限定路径。

此选项只能与 `check` 和 `list-checks` 子命令一起使用。

```
-k keyword[,...]  
--list-keywords=keyword  
--keyword keyword
```

列出可用检查中的所有关键字。此选项会覆盖所有其他选项。
此选项只能与 `list-checks` 子命令一起使用。

```
-k keyword[,...]  
--keyword=keyword  
--keyword keyword
```

仅运行包含指定关键字的检查。使用 `cluster list-checks -k` 命令可以确定分配给可用检查的关键字。

`-k functional` 关键字需要 `-C` 选项具有单个 `checkid`。您不能一次指定多个功能检查或者在同一命令中指定任何其他关键字。

此选项只能与 `check` 和 `list-checks` 子命令一起使用。

```
-m message  
--message=message  
--message message
```

指定要与发出 `shutdown` 子命令时显示的警告一起显示的消息字符串。

标准警告消息是 `system will be shut down in ...`。

如果 `message` 包含多个单词，请使用单引号 (') 或双引号 (") 标记将其隔开。

`shutdown` 命令会在群集关闭前 7200、3600、1800、1200、600、300、120、60 和 30 秒发出消息。

```
-n node[,...]  
--node=node[,...]  
--node node[,...]
```

仅在指定的节点或一组节点上运行检查。`node` 的值可以是节点名称，也可以是节点 ID 号。

此选项只能与 `check` 子命令一起使用。

```
-o {- | clconfigfile}  
--output={- | clconfigfile}  
--output {- | clconfigfile }
```

将群集配置信息写入一个文件或标准输出 (stdout)。[clconfiguration\(5CL\) \[1245\]](#) 手册页中描述了该配置信息的格式。

如果您使用该选项指定文件名称，该选项将创建新的文件。然后将配置信息置于该文件中。如果您使用该选项指定 `-`，则会将配置信息发送到标准输出 (stdout)。命令的所有其他标准输出将受到抑制。

此格式的 `-o` 选项只能与 `export` 子命令一起使用。

`-o outputdir`
`--output=outputdir`
`--output outputdir`

指定要将 `check` 子命令生成的报告保存到的目录。

此格式的 `-o` 选项只能与 `check` 和 `list-checks` 子命令一起使用。

必须已存在或能够创建输出目录 `outputdir`。位于 `outputdir` 中的先前报告会被新的报告覆盖。

如果未指定 `-o` 选项，则目录 `/var/cluster/logs/cluster_check/datestamp/` 会默认用作 `outputdir`。

`-p name=value`
`--property=name=value`
`--property name=value`

修改群集范围的属性。

允许使用多个 `-p name= value` 实例。

将此选项与 `set` 和 `set-netprops` 子命令一起使用可修改以下属性：

`concentrate_load`

指定资源组管理器 (Resource Group Manager, RGM) 跨可用节点分布资源组负载的方式。仅可以在全局群集中设置 `concentrate_load` 属性。在区域群集中，`concentrate_load` 属性的默认值为 `FALSE`。如果将该值设置为 `FALSE`，则 RGM 会尝试将资源组负载均匀地分布到该资源组的节点列表中列出的所有可用节点或区域上。如果在全局群集中将该值设置为 `TRUE`，则资源组负载会在不超过任何配置的硬负载限制或软负载限制的情况下集中分布到尽可能少的节点或区域上。默认值为 `FALSE`。

如果资源组 RG2 对资源组 RG1 声明了 `++` 或 `+++` 关联，请避免为 RG2 设置任何非零负载因子。而应为 RG1 设置较大的负载因子，以抵偿 RG2 在与 RG1 相同的节点上联机时施加的额外负载。这样可使 `Concentrate_load` 功能按预期方式工作。或者，可以对 RG2 设置负载因子，但是避免对这些负载因子设置任何硬负载限制；请仅设置软限制。这样 RG2 在超出软负载限制的情况下也能联机。

每个节点的硬负载和软负载限制均使用 `clnode create-loadlimit`、`clnode set-loadlimit` 和 `clnode delete-loadlimit` 命令创建和修改。有关说明，请参见 [clnode\(1CL\) \[149\]](#) 手册页。

`global_fencing`

为所有共享设备指定全局默认隔离算法。

该属性可以接受的值为 `nofencing`、`nofencing-noscrub`、`pathcount` 或 `prefer3`。

检查和删除任何永久性组保留 (Persistent Group Reservation, PGR) 密钥之后，`nofencing` 设置会关闭共享设备的隔离功能。

`nofencing-noscrub` 设置可以关闭共享设备的隔离功能，无需首先检查或删除 PGR 密钥。

`pathcount` 设置根据附加到共享设备的 DID 路径的数目确定隔离协议。对于使用不少于 3 个 DID 路径的设备，该属性设置为 SCSI-3 协议。

`prefer3` 设置为所有设备的设备隔离功能指定 SCSI-3 协议。`pathcount` 设置已指定给不支持 SCSI-3 协议的所有设备。

默认情况下，该属性已设置为 `prefer3`。

`heartbeat_quantum`

定义发送心跳的频率（毫秒）。

默认情况下，Oracle Solaris Cluster 软件使用的心跳量程为 1 秒（或 1,000 毫秒）。指定一个介于 100 和 10,000 毫秒之间的值。

`heartbeat_timeout`

定义时间间隔（毫秒），在这段时间之后，如果没有从同级节点接收到任何心跳，则相应路径会声明为关闭。

默认情况下，Oracle Solaris Cluster 软件使用的心跳超时时间为 10 秒（或 10,000 毫秒）。指定一个介于 2,500 和 60,000 毫秒之间的值。

通过 `set` 子命令，可以跨所有适配器修改群集的全局心跳参数。

Oracle Solaris Cluster 软件依靠专用互连上的心跳检测群集节点之间的通信故障。如果减少心跳超时时间，Oracle Solaris Cluster 软件可以更快地检测故障。减少心跳超时时间值时检测故障所需的时间也会减少。因此，Oracle Solaris Cluster 软件可以更快地从故障中恢复。更快的恢复会提高群集的可用性。

使用 `set` 子命令减小心跳参数的值时，即使在理想条件下，也始终存在可能发生可疑路径超时和节点紧急情况的风险。始终测试并全面将心跳参数的较低值限定在相关工作负载条件之下后，才能在群集中真正实施这些值。

为 `heartbeat_timeout` 指定的值必须始终大于或等于为 `heartbeat_quantum` 指定的值的五倍 ($\text{heartbeat_timeout} \geq (5 * \text{heartbeat_quantum})$)。

`installmode`

指定群集的安装模式设置。可以为 `installmode` 属性指定 `enabled` 或 `disabled`。

当 `installmode` 属性处于启用状态时，节点不会尝试在引导时重置其法定配置。并且，在这种模式下，很多管理功能都被阻止。首次安装群集时，`installmode` 属性处于启用状态。

在所有节点首次加入群集并且共享法定设备已添加到配置中之后，您必须明确禁用 `installmode` 属性。禁用 `installmode` 属性之后，法定投票计数会设置为默认值。如果在创建群集期间自动配置了法定设备，则 `installmode` 属性也会在配置法定设备之后被禁用。

resource_security

指定 RGM 资源执行程序所用的安全策略。允许的 resource_security 值为 SECURE、WARN、OVERRIDE 或 COMPATIBILITY。

Start 和 Validate 之类的资源方法始终以 root 用户身份运行。如果方法可执行文件不归 root 所有或具有组或全局写权限，将存在不安全状况。在这种情况下，如果 resource_security 属性设置为 SECURE，在运行时资源方法执行将失败并返回错误。如果 resource_security 设置为任何其他值，将允许资源方法执行，但显示警告消息。为了最大限度地保证安全性，请将 resource_security 设置为 SECURE。

resource_security 设置还会修改声明了 application_user 资源属性的资源类型的行为。声明了 application_user 资源属性的资源类型通常是一个代理，该代理使用 scha_check_app_user(1HA) [563] 接口对应用程序可执行文件的所有权和权限执行额外的检查。有关更多信息，请参见 r_properties(5) [1103] 手册页的 application_user 部分。

udp_session_timeout

指定时间间隔（秒），超过该间隔后所有非活动的 UDP 会话都会被删除。

（可选）可以将该属性设置为任意整数。

该属性仅应用于 UDP 服务以及启用了循环负载均衡方案的负载均衡策略 Lb_weighted。

默认情况下，该属性已设置为 480 秒（8 分钟）。

专用网络属性

仅可以使用 set-netprops 子命令修改专用网络属性。

仅当默认的专用网络地址与某个已经在使用的地址发生冲突时，才必须修改这些专用网络设置。当现有地址范围不足以满足持续增长的群集配置时，也必须修改这些专用网络设置。

修改网络属性时，期望群集的所有节点都可用并且位于非群集模式下。仅在群集的某一个节点上修改专用网络设置，因为设置会被传播到所有节点。

设置 private_netaddr 属性时，还可以设置 private_netmask 属性或 max_nodes 和 max_privatenets 属性，或者所有属性。如果尝试设置 private_netmask 属性以及 max_nodes 或 max_privatenets 属性，将出现一个错误。您必须始终同时设置 max_nodes 和 max_privatenets 这两个属性。

默认专用网络地址为 172.16.0.0，其默认网络掩码为 255.255.240.0。

如果因群集配置不一致而无法设置属性，请在非群集模式下，在每个节点上运行 cluster restore-netprops 命令。

专用网络属性如下所示：

max_nodes

指定您希望成为群集一部分的节点的最大数量。该属性只能通过结合 `private_netaddr` 和 `max_privatenets` 属性（或 `private_netmask` 属性）来设置。`max_nodes` 的最大值为 64。最小值为 2。

max_privatenets

指定您希望在群集中使用的专用网络的最大数量。仅可以结合 `private_netaddr` 和 `max_nodes` 属性（或 `private_netmask` 属性）设置该属性。`max_privatenets` 的最大值为 128。最小值为 2。

num_zoneclusters

指定要为全局群集配置的区域群集的数量。Oracle Solaris Cluster 软件组合使用该值、节点数和为全局群集指定的专用网络数来计算专用网络的网络掩码。

Oracle Solaris Cluster 软件使用专用网络的网络掩码来确定保留给群集使用的专用网络 IP 地址的范围。

可以在群集模式或非群集模式下设置该属性。

如果不为该属性指定值，默认设置为 12。可以为该属性指定 0。

private_netaddr

指定专用网络地址。

private_netmask

指定群集专用网络掩码。在这种情况下指定的值必须大于或等于默认的网络掩码 `255.255.240.0`。仅可以结合 `private_netaddr` 属性设置该属性。

如果要分配比默认地址范围更小的 IP 地址范围，可以改用除 `private_netmask` 属性之外的属性或者 `max_nodes` 和 `max_privatenets` 属性。

num_xip_zoneclusters

指定可以对物理群集配置的独占 IP 区域群集的数量。该命令会调用名为 `modify_xip_zc` 的 shell 脚本，然后使用可配置的独占 IP 区域群集的数量对应的条目更新 `clprivnet` 配置文件。`num_xip_zoneclusters` 属性必须是 `num_zoneclusters` 属性的子集。

对于专用网络属性的每种组合，该命令将执行以下任务：

`-p private_netaddr=netaddr`

该命令将默认网络掩码 `255.255.240.0` 分配给专用互连。默认 IP 地址范围最多可以容纳 64 个节点和 10 个专用网络。

`-p private_netaddr=netaddr,private_netmask=netmask`

如果指定的网络掩码小于默认网络掩码，则该命令会失败然后退出，并显示一条错误消息。

如果指定的网络掩码大于或等于默认的网络掩码，则该命令会将指定的网络掩码分配给专用互连。生成的 IP 地址范围最多可以容纳 64 个节点和 10 个专用网络。

要分配比默认地址范围更小的 IP 地址范围，可以改用除 `private_netmask` 属性之外的属性或者 `max_nodes` 和 `max_privatenets` 属性。

```
-p private_netaddr=netaddr,max_nodes=nodes,  
max_privatenets=privatenets,num_xip_zoneclusters=xip_zoneclusters
```

该命令可以计算能够支持指定数量的节点和专用网络的最小网络掩码。该命令随后将计算的网路掩码分配给专用互连。该命令还指定可以对物理群集配置的独占 IP 区域群集的数量。

```
-p private_netaddr=netaddr,private_netmask=netmask,  
max_nodes=nodes,max_privatenets=privatenets
```

该命令可以计算能够支持指定数量的节点和专用网络的最小网络掩码。

该命令会将计算结果与指定的网络掩码进行比较。如果指定的网络掩码小于计算的网路掩码，则该命令会失败然后退出，并显示一条错误消息。如果指定的网络掩码大于或等于计算的网路掩码，则该命令会将指定的网络掩码分配给专用互连。

```
-s severitylevel  
--severity=severitylevel  
--severity severitylevel
```

仅报告级别至少为 `severitylevel` 的违规。

此选项只能与 `check` 子命令一起使用。

每个检查都分配有一个严重级别。指定严重级别可以从报告中排除严重级别较轻的失败检查。`severity` 的值是按最低到最高严重级别顺序列出的以下值之一：

information

warning

low

medium

high

critical

没有指定该选项时，会默认使用 `information` 严重级别。严重级别 `information` 表明所有严重级别的失败检查都将包括在报告中。

`-t objecttype[...]`
`--type=objecttype[...]`
`--type objecttype[...]`

指定 `export`、`show` 和 `status` 子命令的对象类型。

使用该选项可以将 `export`、`show` 和 `status` 子命令的输出仅限制到指定类型的对象。支持以下对象或组件类型。请注意，对某些对象类型而言，状态并不适用。

对象类型/短对象类型	可用状态
<code>access/access</code>	否
<code>device/dev</code>	是
<code>devicegroup/dg</code>	是
<code>global/global</code>	否
<code>interconnect/intr</code>	是
<code>nasdevice/nas</code>	否
<code>node/node</code>	是
<code>quorum/quorum</code>	是
<code>reslogicalhostname/rslh</code>	是
<code>resource/rs</code>	是
<code>resourcegroup/rg</code>	是
<code>resourcetype/rt</code>	否
<code>ressharedaddress/rssa</code>	是

`-v`
`--verbose`

在标准输出 (`stdout`) 中显示详细信息。与 `check` 子命令一起使用时，显示执行期间的详细进度。与 `list-checks` 子命令一起使用时，提供关于检查的更多详细信息。

`-V`
`--version`

显示该命令的版本。

如果是与其他选项、子命令或操作数一起指定该选项，将忽略所有这些内容。仅显示该命令的版本。不会发生任何其他处理。

`-y`
`--yes`

禁止发出用于询问您是否确认要关闭的提示。群集会立即关闭，无需用户干预。

支持以下操作数：

clustername

要管理的群集的名称。

对于除了 `create` 以外的所有子命令，您指定的 *clustername* 必须与您在其上发出 `cluster` 命令的群集的名称匹配。

使用 `create` 子命令指定新的唯一群集名称。

[Intro\(1CL\) \[17\]](#) 手册页列出了该命令集中所有命令的一整套退出状态代码。返回的退出代码还与 [scha_calls\(3HA\) \[897\]](#) 手册页中介绍的返回代码兼容。

如果命令对于所有指定的操作数均成功执行，则返回零 (`CL_NOERR`)。如果某个操作数出错，命令会继续处理操作数列表中的下一个操作数。返回的退出代码始终反映最先发生的错误。

此命令返回下列退出状态代码：

0 `CL_NOERR`

没有错误

您发出的命令已成功完成。

1 `CL_ENOMEM`

交换空间不足

某个群集节点耗尽了交换内存或其他操作系统资源。

3 `CL_EINVAL`

参数无效

您键入的命令不正确，或者使用 `-i` 选项提供的群集配置信息语法不正确。

6 `CL_EACCESS`

权限被拒绝

指定的对象不可访问。要发出命令，可能需要拥有超级用户或 RBAC 访问权限。有关更多信息，请参见 [Unresolved link to "su1M"](#) 和 [Unresolved link to "rbac5"](#) 手册页。

35 `CL_EIO`

I/O 错误

发生了物理输入/输出错误。

36 `CL_ENOENT`

没有这样的对象

由于以下某个原因，找不到您所指定的对象：

- 该对象不存在。
- 您尝试使用 `-o` 选项创建的配置文件路径中的某个目录不存在。

- 您尝试使用 `-i` 选项访问的配置文件包含错误。

此外，`check` 子命令在其放置检查报告的同一输出目录中创建名为 `cluster_check_exit_code.log` 的文本文件。如果该子命令本身退出 `CL_NOERR`，则会在该文件中报告一个代码，指出所有违规检查的最高严重级别。以下是可能的检查代码：

100	未报告任何违规。报告中可能存在严重级别为 <code>information</code> 或 <code>warning</code> 的检查输出。
101	<code>critical</code>
102	<code>high</code>
103	<code>medium</code>
104	<code>low</code>

例 249 显示群集配置信息

以下命令可显示所有可用的群集配置信息。

```
# cluster show
=== Cluster ===

Cluster Name:                schost
clusterid:                   0x4FA7C35F
installmode:                 disabled
heartbeat_timeout:          9999
heartbeat_quantum:          1000
private_netaddr:             172.16.0.0
private_netmask:             255.255.240.0
max_nodes:                   64
max_privatenets:             10
num_zoneclusters:           12
udp_session_timeout:         480
concentrate_load:            True
resource_security:           SECURE
global_fencing:              prefer3
Node List:                   phys-schost-1, phys-schost-2

=== Host Access Control ===

Cluster name:                schost
  Allowed hosts:              None
  Authentication Protocol:    sys

=== Cluster Nodes ===

Node Name:                   phys-schost-1
  Node ID:                    1
```

```

Enabled: yes
privatehostname: clusternode1-priv
reboot_on_path_failure: disabled
globalzoneshares: 1
defaultpsetmin: 1
quorum_vote: 1
quorum_defaultvote: 1
quorum_resv_key: 0x4FA7C35F00000001
Transport Adapter List: net3, net1

Node Name: phys-schost-2
Node ID: 2
Enabled: yes
privatehostname: clusternode2-priv
reboot_on_path_failure: disabled
globalzoneshares: 1
defaultpsetmin: 1
quorum_vote: 1
quorum_defaultvote: 1
quorum_resv_key: 0x4FA7C35F00000002
Transport Adapter List: net3, net1

=== Transport Cables ===

Transport Cable: phys-schost-1:net3,switch1@1
Endpoint1: phys-schost-1:net3
Endpoint2: switch1@1
State: Enabled

Transport Cable: phys-schost-1:net1,switch2@1
Endpoint1: phys-schost-1:net1
Endpoint2: switch2@1
State: Enabled

Transport Cable: phys-schost-2:net3,switch1@2
Endpoint1: phys-schost-2:net3
Endpoint2: switch1@2
State: Enabled

Transport Cable: phys-schost-2:net1,switch2@2
Endpoint1: phys-schost-2:net1
Endpoint2: switch2@2
State: Enabled

=== Transport Switches ===

Transport Switch: switch1
State: Enabled
Type: switch
Port Names: 1 2
Port State(1): Enabled
Port State(2): Enabled

Transport Switch: switch2

```

```
State: Enabled
Type: switch
Port Names: 1 2
Port State(1): Enabled
Port State(2): Enabled
```

```
=== Quorum Devices ===
```

```
Quorum Device Name: d4
Enabled: yes
Votes: 1
Global Name: /dev/did/rdisk/d4s2
Type: shared_disk
Access Mode: scsi3
Hosts (enabled): phys-schost-1, phys-schost-2
```

```
=== Device Groups ===
```

```
=== Registered Resource Types ===
```

```
Resource Type: SUNW.LogicalHostname:4
RT_description: Logical Hostname Resource Type
RT_version: 4
API_version: 2
RT_basedir: /usr/cluster/lib/rgm/rt/hafoip
Single_instance: False
Proxy: False
Init_nodes: All potential masters
Installed_nodes: <All>
Failover: True
Pkglist: <NULL>
RT_system: True
Global_zone: True
```

```
Resource Type: SUNW.SharedAddress:2
RT_description: HA Shared Address Resource Type
RT_version: 2
API_version: 2
RT_basedir: /usr/cluster/lib/rgm/rt/hascip
Single_instance: False
Proxy: False
Init_nodes: <Unknown>
Installed_nodes: <All>
Failover: True
Pkglist: <NULL>
RT_system: True
Global_zone: True
```

```
=== Resource Groups and Resources ===
```

```
=== DID Device Instances ===
```

```
DID Device Name: /dev/did/rdisk/d1
Full Device Path: phys-schost-2:/dev/rdisk/
```

```

Full Device Path:          c0t600A0B8000485B6A000058584EDCBD7Ed0
                           phys-schost-1:/dev/rdisk/
                           c0t600A0B8000485B6A000058584EDCBD7Ed0
Replication:              none
default_fencing:         global

DID Device Name:         /dev/did/rdisk/d2
Full Device Path:        phys-schost-2:/dev/rdisk/
                           c0t600A0B8000485B6A0000585A4EDCBDA4d0
Full Device Path:        phys-schost-1:/dev/rdisk/
                           c0t600A0B8000485B6A0000585A4EDCBDA4d0
Replication:              none
default_fencing:         global

DID Device Name:         /dev/did/rdisk/d3
Full Device Path:        phys-schost-2:/dev/rdisk/
                           c0t600A0B8000485B6A0000585C4EDCBDCAd0
Full Device Path:        phys-schost-1:/dev/rdisk/
                           c0t600A0B8000485B6A0000585C4EDCBDCAd0
Replication:              none
default_fencing:         global

DID Device Name:         /dev/did/rdisk/d4
Full Device Path:        phys-schost-2:/dev/rdisk/
                           c0t600A0B8000485B6A0000585E4EDCBDF1d0
Full Device Path:        phys-schost-1:/dev/rdisk/
                           c0t600A0B8000485B6A0000585E4EDCBDF1d0
Replication:              none
default_fencing:         global

DID Device Name:         /dev/did/rdisk/d5
Full Device Path:        phys-schost-2:/dev/rdisk/
                           c0t600A0B8000485B6A000058604EDCBE1Cd0
Full Device Path:        phys-schost-1:/dev/rdisk/
                           c0t600A0B8000485B6A000058604EDCBE1Cd0
Replication:              none
default_fencing:         global

DID Device Name:         /dev/did/rdisk/d6
Full Device Path:        phys-schost-2:/dev/rdisk/
                           c0t600A0B8000486F08000073014EDCBED0d0
Full Device Path:        phys-schost-1:/dev/rdisk/
                           c0t600A0B8000486F08000073014EDCBED0d0
Replication:              none
default_fencing:         global

DID Device Name:         /dev/did/rdisk/d7
Full Device Path:        phys-schost-2:/dev/rdisk/
                           c0t600A0B8000486F08000073034EDCBEFAd0
Full Device Path:        phys-schost-1:/dev/rdisk/
                           c0t600A0B8000486F08000073034EDCBEFAd0
Replication:              none
default_fencing:         global

```

```

DID Device Name: /dev/did/rdisk/d8
  Full Device Path: phys-schost-2:/dev/rdisk/
                    c0t600A0B8000486F08000073054EDCBF1Fd0
  Full Device Path: phys-schost-1:/dev/rdisk/
                    c0t600A0B8000486F08000073054EDCBF1Fd0
  Replication: none
  default_fencing: global

DID Device Name: /dev/did/rdisk/d9
  Full Device Path: phys-schost-2:/dev/rdisk/
                    c0t600A0B8000486F08000073074EDCBF46d0
  Full Device Path: phys-schost-1:/dev/rdisk/
                    c0t600A0B8000486F08000073074EDCBF46d0
  Replication: none
  default_fencing: global

DID Device Name: /dev/did/rdisk/d10
  Full Device Path: phys-schost-2:/dev/rdisk/
                    c0t600A0B8000486F08000073094EDCBF71d0
  Full Device Path: phys-schost-1:/dev/rdisk/
                    c0t600A0B8000486F08000073094EDCBF71d0
  Replication: none
  default_fencing: global

DID Device Name: /dev/did/rdisk/d11
  Full Device Path: phys-schost-1:/dev/rdisk/c3t0d0
  Replication: none
  default_fencing: global

DID Device Name: /dev/did/rdisk/d12
  Full Device Path: phys-schost-1:/dev/rdisk/c4t0d0
  Replication: none
  default_fencing: global

DID Device Name: /dev/did/rdisk/d13
  Full Device Path: phys-schost-1:/dev/rdisk/c4t1d0
  Replication: none
  default_fencing: global

DID Device Name: /dev/did/rdisk/d14
  Full Device Path: phys-schost-2:/dev/rdisk/c3t0d0
  Replication: none
  default_fencing: global

DID Device Name: /dev/did/rdisk/d15
  Full Device Path: phys-schost-2:/dev/rdisk/c4t0d0
  Replication: none
  default_fencing: global

DID Device Name: /dev/did/rdisk/d16
  Full Device Path: phys-schost-2:/dev/rdisk/c4t1d0
  Replication: none
  default_fencing: global

```

=== NAS Devices ===

Nas Device: qualfugu
Type: sun_uss
userid: osc_agent

=== Zone Clusters ===

Zone Cluster Name: zc1
zonename: zc1
zonpath: /zones/zc1
autoboot: TRUE
brand: solaris10
bootargs: <NULL>
pool: <NULL>
limitpriv: <NULL>
scheduling-class: <NULL>
ip-type: shared
enable_priv_net: TRUE
resource_security: COMPATIBILITY

--- Solaris Resources for zc1 ---

Resource Name: net
address: schost-1
physical: auto

Resource Name: net
address: schost-2
physical: auto

--- Zone Cluster Nodes for zc1 ---

Node Name: phys-schost-1
physical-host: phys-schost-1
hostname: vzschost1a

--- Solaris Resources for phys-schost-1 ---

Node Name: phys-schost-2
physical-host: phys-schost-2
hostname: vzschost2a

--- Solaris Resources for phys-schost-2 ---

Zone Cluster Name: zc2
zonename: zc2
zonpath: /zones/zc2
autoboot: TRUE
brand: solaris
bootargs: <NULL>
pool: <NULL>
limitpriv: <NULL>
scheduling-class: <NULL>

```

ip-type:                shared
enable_priv_net:        TRUE
resource_security:      COMPATIBILITY

--- Solaris Resources for zc2 ---

--- Zone Cluster Nodes for zc2 ---

Node Name:              phys-schost-1
physical-host:          phys-schost-1
hostname:               vzscho1b

--- Solaris Resources for phys-schost-1 ---

Node Name:              phys-schost-2
physical-host:          phys-schost-2
hostname:               vzscho2b

--- Solaris Resources for phys-schost-2 ---

Zone Cluster Name:      zc3
zonename:               zc3
zonepath:               /zones/zc3
autoboot:               TRUE
brand:                  solaris
bootargs:               <NULL>
pool:                   <NULL>
limitpriv:              <NULL>
scheduling-class:      <NULL>
ip-type:                shared
enable_priv_net:        TRUE
resource_security:      COMPATIBILITY

--- Solaris Resources for zc3 ---

--- Zone Cluster Nodes for zc3 ---

Node Name:              phys-schost-2
physical-host:          phys-schost-2
hostname:               vzscho1c

--- Solaris Resources for phys-schost-2 ---

```

例 250 显示关于选定群集组件的配置信息

以下命令可显示关于资源、资源类型以及资源组的信息。仅显示关于群集的信息。

```

# cluster show -t resource, resourcetype, resourcegroup
Single_instance:        False
Proxy:                  False
Init_nodes:             <Unknown>

```

```

Installed_nodes:          <All>
Failover:                 True
Pkglist:                  <NULL>
RT_system:                True

Resource Type:            SUNW.qfs
RT_description:           SAM-QFS Agent on SunCluster
RT_version:               3.1
API_version:              3
RT_basedir:               /opt/SUNWsamfs/sc/bin
Single_instance:         False
Proxy:                    False
Init_nodes:               All potential masters
Installed_nodes:         <All>
Failover:                 True
Pkglist:                  <NULL>
RT_system:                False

```

=== Resource Groups and Resources ===

```

Resource Group:          qfs-rg
RG_description:          <NULL>
RG_mode:                 Failover
RG_state:                Managed
Failback:                False
Nodelist:                phys-schost-2 phys-schost-1

```

--- Resources for Group qfs-rg ---

```

Resource:                qfs-res
Type:                    SUNW.qfs
Type_version:            3.1
Group:                   qfs-rg
R_description:           default
Resource_project_name:   default
Enabled{phys-schost-2}:  True
Enabled{phys-schost-1}:  True
Monitored{phys-schost-2}: True
Monitored{phys-schost-1}: True

```

例 251 显示群集状态

以下命令可显示所有群集节点的状态。

```

# cluster status -t node
=== Cluster Nodes ===

--- Node Status ---

Node Name                Status
-----                -
phys-schost-1            Online
phys-schost-2            Online

```

```
--- Node Status ---
```

```
Node Name                               Status  
-----                               -
```

或者，也可以使用 `clnode` 命令显示相同信息。

```
# clnode status  
=== Cluster Nodes ===
```

```
--- Node Status ---
```

```
Node Name                               Status  
-----                               -  
phys-schost-1                           Online  
phys-schost-2                           Online
```

例 252 创建群集

以下命令可以从群集配置文件 `suncluster.xml` 创建名为 `cluster-1` 的群集。

```
# cluster create -i /suncluster.xml cluster-1
```

例 253 更改群集的名称

以下命令可以将群集的名称更改为 `cluster-2` 。

```
# cluster rename -c cluster-2
```

例 254 禁用群集的 `installmode` 属性

以下命令可以禁用群集的 `installmode` 属性。

```
# cluster set -p installmode=disabled
```

例 255 修改专用网络

以下命令可修改群集的专用网络设置。该命令将专用网络地址设置为 `172.10.0.0`。该命令还可以计算和设置能够支持指定的八个节点和四个专用网络的最小专用网络掩码，并指定您要为全局群集配置八个区域群集。该命令还标识可以在非群集模式下对物理群集配置的独占 IP 区域群集的数量。

```
# cluster set-netprops \  
-p private_netaddr=172.10.0.0 \  
-p max_nodes=8 \  
-p max_privatenets=4 \  
-p num_zoneclusters=8 \  
-p num_xip_zoneclusters=3
```

还可以在非群集模式下按如下所示指定该命令：

```
# cluster set-netprops \  
-p private_netaddr=172.10.0.0 \  
-p max_nodes=8,\  
-p max_privatenets=4 \  
-p num_zoneclusters=8 \  
-p num_xip_zoneclusters=3
```

例 256 列出可用检查

以下命令可列出群集上所有可用的检查（以单行的格式显示）。可用的实际检查因发行版或更新而异。

```
# cluster list-checks  
M6336822 : (Critical) Global filesystem /etc/vfstab entries are  
not consistent across all Oracle Solaris Cluster nodes.  
S6708689 : (Variable) One or more Oracle Solaris Cluster resources  
cannot be validated  
M6708613 : (Critical) vxio major numbers are not consistent across  
all Oracle Solaris Cluster nodes.  
S6708255 : (Critical) The nsswitch.conf file 'hosts' database  
entry does not have 'cluster' specified first.  
S6708479 : (Critical) The /etc/system rpcmod:svc_default_stksize  
parameter is missing or has an incorrect value for Oracle Solaris Cluster.  
F6984121 : (Critical) Perform cluster shutdown  
F6984140 : (Critical) Induce node panic  
...
```

例 257 在群集上运行基本检查

以下命令在详细模式下在 schost 群集的所有节点上运行所有可用的基本检查，其中 phys-schost-1 是群集成员。输出会重定向至文件 basicchks.18Nov2011.schost。

```
phys-schost-1# cluster check -v -o basicchks.18Nov2011.schost
```

例 258 在群集上运行交互式检查

以下命令可以运行除了具有 vfstab 关键字的那些检查之外的所有可用交互式检查。检查的输出会保存到文件 interactive.chk.18Nov2011。

```
# cluster check -k interactive -E vfstab -o interactive.chk.18Nov2011 cluster-1
```

User supplies information when prompted

例 259 在群集上运行功能检查

以下命令可显示功能检查 F6968101 的详细描述并在包含 phys-schost-1、phys-schost-2 和 phys-schost-3 群集成员的群集上运行该检查。检查的输出会保存到文件

F6968101.failovertest.19Nov2011。因为该检查涉及到故障转移群集节点，所以在使群集脱离生产环境之前，请勿启动该检查。

```
phys-schost-1# cluster list-checks -v -C F6968101
```

```
initializing...
F6968101: (Critical) Perform resource group switchover
Keywords: SolarisCluster4.x, functional
Applicability: Applicable if multi-node cluster running live.
Check Logic: Select a resource group and destination node.
Perform '/usr/cluster/bin/clresourcegroup switch' on specified
resource group either to specified node or to all nodes in succession.
Version: 1.118
Revision Date: 13/07/09
```

```
cleaning up...
```

Take the cluster out of production

```
phys-schost-1# cluster check -k functional -C F6968101 \
-o F6968101.failovertest.19Nov2011
```

```
initializing...
initializing xml output...
loading auxiliary data...
starting check run...
phys-schost-1, phys-schost-2, phys-schost-3: F6968101.... starting:
Perform resource group switchover
```

```
=====
```

```
>>> Functional Check <<<
```

Follow onscreen directions

```
...
```

例 260 在指定的节点上运行有限的检查

以下命令在详细模式下运行严重级别为 high 或更高的所有检查。这些检查仅在节点 phys-schost-1 上运行。

```
# cluster check -v -n phys-schost-1 -s high
```

```
initializing...
initializing xml output...
loading auxiliary data...
filtering out checks with severity less than High
starting check run...
phys-schost-1: M6336822.... starting: Global filesystem /etc/vfstab entries...
phys-schost-1: M6336822 not applicable
phys-schost-1: S6708689.... starting: One or more Oracle Solaris Cluster...
phys-schost-1: S6708689 passed
```

```
...
```

```
phys-schost-1: S6708606 skipped: severity too low
```

```

phys-schost-1: S6708638      skipped: severity too low
phys-schost-1: S6708641.... starting: Cluster failover/switchover might...
phys-schost-1: S6708641      passed
...

```

/usr/cluster/lib/cfgchk/checkresults.dtd

/var/cluster/logs/cluster_check/

/outputdir/cluster_check_exit_code.log

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

[Intro\(1CL\) \[17\]](#)、[Unresolved link to "init1M"](#)、[Unresolved link to "su1M"](#)、[scha_calls\(3HA\) \[897\]](#)、[Unresolved link to "attributes5"](#)、[Unresolved link to "rbac5"](#)、[clconfiguration\(5CL\) \[1245\]](#)

超级用户可以运行此命令的所有格式。

所有用户都可以运行带有 `-?` (帮助) 或 `-v` (版本) 选项的此命令。

要运行带有子命令的 `cluster` 命令，超级用户以外的用户需要具有 RBAC 授权。请参见下表。

子命令	RBAC 授权
check	solaris.cluster.read
create	solaris.cluster.modify
export	solaris.cluster.read
list	solaris.cluster.read
list-checks	solaris.cluster.read
list-cmds	solaris.cluster.read
rename	solaris.cluster.modify
restore-netprops	solaris.cluster.modify
set	solaris.cluster.modify
set-netprops	solaris.cluster.modify

子命令	RBAC 授权
show	solaris.cluster.read
show-netprops	solaris.cluster.read
shutdown	solaris.cluster.admin
status	solaris.cluster.read

名称

clzonecluster, clzc — 创建和管理区域群集

```
/usr/cluster/bin/clzonecluster [subcommand] -?  
  
/usr/cluster/bin/clzonecluster -V  
  
/usr/cluster/bin/clzonecluster subcommand [options] -v  
    [zone-cluster-name]  
  
/usr/cluster/bin/clzonecluster apply [-n node-name[,...]] [-d]  
    {+ | zone-cluster-name [...]}  
  
/usr/cluster/bin/clzonecluster boot [-n node-name[,...]] [-o]  
    {+ | zone-cluster-name [...]}  
  
/usr/cluster/bin/clzonecluster clone -Z target-zone-cluster-name  
    [-m method][-n node-name[,...]] {source-zone-cluster-name}  
  
/usr/cluster/bin/clzonecluster configure [-f command-file]  
    zone-cluster-name  
  
/usr/cluster/bin/clzonecluster delete [-F] zone-cluster-name  
  
/usr/cluster/bin/clzonecluster export [-f command-file]  
    zone-cluster-name  
  
/usr/cluster/bin/clzonecluster halt [-n node-name[,...]]  
    {+ | zone-cluster-name}  
  
/usr/cluster/bin/clzonecluster install [-c config_profile.xml]  
    [-M manifest.xml] zone-cluster-name  
  
/usr/cluster/bin/clzonecluster install [-n node-name]  
    -a absolute_path_to_archive [-x cert|ca-cert|key=file]...  
    -z zone zone-cluster-name  
  
/usr/cluster/bin/clzonecluster install [-n node-name]  
    -d absolute_root_path zone-cluster-name  
  
/usr/cluster/bin/clzonecluster install-cluster  
    [-d dvd-image] [-n node-name[,...]]  
    [-p patchdir=patch-dir[,patchlistfile=file-name]]  
    -s software-component[,...] [-v] zone-cluster-name  
  
/usr/cluster/bin/clzonecluster install-cluster  
    [-p patchdir=patch-dir[,patchlistfile=file-name]  
    [-n node-name[,...]] [-v] zone-cluster-name  
  
/usr/cluster/bin/clzonecluster list [+ | zone-cluster-name [...]]  
  
/usr/cluster/bin/clzonecluster move -f zone-path zone-cluster-name  
  
/usr/cluster/bin/clzonecluster ready [-n node-name[,...]]  
    {+ | zone-cluster-name [...]}
```

```
/usr/cluster/bin/clzonecluster reboot [-n node-name[,...]] [-o]
    {+ | zone-cluster-name [...]}

/usr/cluster/bin/clzonecluster set {-p name=value}
    [-p name=value] [...] [zone-cluster-name]

/usr/cluster/bin/clzonecluster show [+ | zone-cluster-name [...]]

/usr/cluster/bin/clzonecluster show-rev [-v] [-n node-name[,...]]
    [+ | zone-cluster-name ...]

/usr/cluster/bin/clzonecluster status [+ | zone-cluster-name [...]]

/usr/cluster/bin/clzonecluster uninstall [-F] [-n node-name
    [...]] zone-cluster-name

/usr/cluster/bin/clzonecluster verify [-n node-name[,...]]
    {+ | zone-cluster-name [...]}
```

`clzonecluster` 命令为 Oracle Solaris Cluster 配置创建和修改区域群集。`clzc` 命令是 `clzonecluster` 命令的简短格式；这两个命令是相同的。`clzonecluster` 命令可以识别群集，并支持单个管理源。可以从一个节点发出命令的所有格式来影响单个区域群集节点或所有节点。

仅当 *options* 是 `-?` 选项或 `-v` 选项时，才可以省略 *subcommand*。

除了 `list`、`show` 和 `status` 子命令以外，子命令都至少需要一个操作数。但是，许多子命令接受使用加号操作数 (+) 来将子命令应用到所有适用的对象。`clzonecluster` 命令可以在区域群集的任何节点上运行，并且可以影响任何或所有区域群集。

每个选项都有长和短两种格式。在 选项 中给出每个选项的两种格式以及该选项的描述。

注 - 创建区域群集之后无法更改区域群集名称。

支持以下子命令：

`apply`

应用对区域群集进行的配置更改。

`apply` 子命令可应用对区域群集进行的永久实时重新配置。您应运行 `clzonecluster configure` 以进行配置更改，然后运行 `apply` 子命令以应用对特定区域群集进行的更改。`apply` 子命令使用 `-n` 选项来指定将应用重新配置的节点的列表。

只能从全局群集节点使用 `apply` 子命令。

`boot`

引导区域群集。

`boot` 子命令引导区域群集。`boot` 子命令使用 `-n` 标志来为节点的特定列表引导区域群集。

只能从全局群集节点使用 `boot` 子命令。

clone

克隆区域群集。

`clone` 命令通过复制已安装的现有区域群集来安装区域群集。该子命令是另一种安装区域群集的方式。`clone` 子命令本身不创建新的区域群集。确保用于克隆的源区域群集在克隆之前处于“已安装”（未运行）状态。必须先使用 `configure` 子命令创建新的区域群集。然后使用 `clone` 子命令将克隆的配置应用到新的区域群集。

只能从全局群集节点使用 `clone` 子命令。

configure

启动交互式实用程序配置 `solaris10` 或 `labeled` 标记区域群集。

`configure` 子命令使用 `zonecfg` 命令在指定的每台计算机上配置区域。`configure` 子命令允许您指定适用于区域群集的每个节点的属性。这些属性与 `zonecfg` 命令为各个区域建立的属性具有相同的意义。`configure` 子命令支持对 `zonecfg` 命令未知的属性进行配置。如果未指定 `-f` 选项，`configure` 子命令将启动交互式 shell。`-f` 选项将命令文件作为其参数。`configure` 子命令使用该文件以非交互方式创建或修改区域群集。

通过 `configure` 子命令，还可以使用统一归档文件配置区域群集并选择恢复归档文件或克隆归档文件。将 `-a archive` 选项与 `create` 子命令一起使用。例如：

```
# /usr/cluster/bin/clzc configure sczone1
sczone1: No such zone cluster configured
Use 'create' to begin configuring a new zone cluster.
clzc:sczone1> create -a archive -z archived zone
```

只能从全局群集节点使用 `configure` 子命令。有关更多信息，请参见[Unresolved link to "Oracle Solaris Cluster 软件安装指南"](#)。

要指定 `solaris10` 标记区域群集，在配置该区域群集时可以使用默认模板。默认模板位于 `/etc/cluster/zone_cluster/ORCLcls10default.xml`。您可以使用 `-t` 选项指定默认 `solaris10` 区域群集模板或群集上的其他现有 `solaris10` 区域群集。如果指定了其他 `solaris10` 区域群集，将从指定的区域群集导入区域群集配置。还必须在 `sysid` 属性中指定 `root` 用户密码，以保证 `verify` 或 `commit` 操作不会失败。键入以下命令应用模板：

```
# /usr/cluster/bin/clzc configure sczone2
sczone2: No such zone cluster configured
Use 'create' to begin configuring a new zone cluster.
clzc:sczone2> create -t ORCLcls10default
clzc:sczone2> info
zonename: sczone2
zonepath:
autoboot: true
hostid:
brand: solaris10
```

`configure` 命令的交互和非交互格式都支持使用多个子命令来编辑区域群集配置。有关可用配置子命令的列表，请参见 [Unresolved link to "zonecfg1M"](#)。

交互式 `configure` 实用程序使您能够创建和修改区域群集的配置。区域群集配置由许多资源类型和属性组成。`configure` 实用程序使用范围概念来确定在何处应用子命令。`configure` 实用程序使用的范围级别有三个：群集、资源和节点特定资源。默认范围为群集。以下列表描述范围的三个级别：

- 群集范围 – 影响整个区域群集的属性。如果 `zoneclustername` 是 `sczone`，`clzonecluster` 命令的交互式 shell 类似于以下内容：

```
clzc:sczone>
```

- 节点范围 – 嵌套在节点资源范围内的特殊资源范围。节点范围中的设置影响区域群集中的特定节点。例如，可以将网络资源添加到区域群集中的特定节点。`clzonecluster` 命令的交互式 shell 类似于以下内容：

```
clzc:sczone:node:net>
```

- 资源范围 – 适用于某个特定资源的属性。资源范围提示附加有资源类型的名称。例如，`clzonecluster` 命令的交互式 shell 类似于以下内容：

```
clzc:sczone:net>
```

delete

删除特定区域群集。

该子命令删除特定区域群集。使用通配符操作数 (*) 时，`delete` 命令将删除在全局群集上配置的区域群集。在运行 `delete` 子命令之前，区域群集必须处于已配置状态。不论区域群集处于什么状态，将 `-F` 选项与 `delete` 命令一起使用都会尝试删除区域群集。

只能从全局群集节点使用 `delete` 子命令。

export

将区域群集配置导出到命令文件。

导出的 `commandfile` 可用作 `configure` 子命令的输入。根据需要修改文件以反映要创建的配置。有关更多信息，请参见 [Unresolved link to "clconfiguration\(5CL\)"](#) 手册页。

只能从全局群集节点使用 `export` 子命令。

halt

停止一个区域群集或该区域群集中的特定节点。

指定特定区域群集时，`halt` 子命令仅应用于该特定区域群集。您可以停止整个区域群集，或仅停止区域群集的特定节点。如果未指定区域群集，`halt` 子命令将应用于所有区域群集。也可以停止指定计算机上的所有区域群集。

`halt` 子命令使用 `-n` 选项来停止特定节点上的区域群集。默认情况下，`halt` 子命令停止所有节点上的所有区域群集。如果指定 + 操作数而不是区域名称，将停止所有区域群集。

只能从全局群集节点使用 `halt` 子命令。

`install`

安装区域群集。

该子命令安装区域群集。

如果使用 `install -M manifest.xml` 选项，则您指定的清单将用于区域群集的所有节点上的安装。清单文件描述管理员进行安装所需的 `solaris` 软件包信息，如 `certificate_file`、`key_file`、发布者以及任何其他软件包。`manifest.xml` 文件还必须指定用于区域群集安装的 Oracle Solaris Cluster 组软件包 `ha-cluster-full`、`ha-cluster-framework-full`、`ha-cluster-data-services-full` 或 `ha-cluster-minimal`。有关自动化安装程序清单的更多信息，请参见 [Unresolved link to "安装 Oracle Solaris 11.2 系统中的创建定制 AI 清单"](#)。

如果不使用 `-M` 选项（默认选项），则会将位于 `/usr/share/auto_install/manifest/zone_default.xml` 的自动化安装程序清单用于安装。使用此 `zone_default.xml` 清单时，在发出命令的区域群集节点的全局区域中安装的所有 `ha-cluster/*` 软件包均会安装在区域群集的所有节点中。如果在安装区域群集时使用定制清单并且未指定 Oracle Solaris Cluster 组软件包，安装将失败。

要安装的所有区域群集节点的底层全局区域与发出 `install` 子命令的区域群集节点的全局区域中安装的 Oracle Solaris Cluster 软件包集必须完全相同。如果存在不满足此要求的区域群集节点，则无法在此节点上完成区域群集安装。

只能从全局群集节点使用 `install` 子命令。`-M` 和 `-c` 选项只能用于 `solaris` 标记区域群集。

如果区域群集的标记是 `solaris10`，则必须使用 `-a` 或 `-d` 选项。

`-a archive`

`solaris` 或 `solaris10` 标记区域群集的统一归档文件的绝对路径、`solaris10` 标记区域群集的 `flar` 归档文件位置或者要用于安装的 Oracle Solaris 10 映像归档文件。有关支持的归档类型的详细信息，请参见 [Unresolved link to "solaris105"](#) 手册页。该归档绝对路径应能在要安装区域群集的群集的所有物理节点上访问。统一归档文件安装可以使用恢复归档文件或克隆归档文件。

`-d path`

已安装的 Oracle Solaris 10 系统的根目录路径。该路径应能在要安装区域群集的群集的所有物理节点上访问。

`[-x cert[ca-cert|key=file]..`

如果具有 HTTPS 统一归档文件位置，请指定 SSL 证书、证书颁发机构 (Certificate Authority, CA) 证书和密钥文件。可以多次指定 `-x` 选项。

`-z zone`

如果统一归档文件包含多个区域，请指定配置源或安装源的区域名称。

将使用同一归档或已安装的 Oracle Solaris 10 系统作为区域群集中所有 solaris10 标记区域的安装源。安装将使用在区域群集配置期间在 sysid 资源类型中指定的系统标识参数覆盖源归档或已安装的 Oracle Solaris 10 中的系统标识参数。

install-cluster

install-cluster 子命令在 solaris10 标记区域群集节点中安装支持 Oracle Solaris 10 OS 的 Oracle Solaris Cluster 软件。安装的软件包括核心软件包、群集软件包组件（例如，区域群集中支持的代理和 Geographic Edition 软件）和修补程序。

注 - install-cluster 子命令不支持在 solaris10 标记区域群集节点中安装 Oracle Solaris Cluster 版本 3.3 或 3.3 5/11 软件。有关 solaris10 标记区域群集支持的发行版的更多信息，请查阅[Unresolved link to "Oracle Solaris Cluster 4.2 发行说明"](#)。

当 solaris10 标记区域安装的 Oracle Solaris 10 系统未装有群集软件时，可使用该子命令。要使用该子命令，必须使用 clzonecluster install 命令将 Oracle Solaris 10 系统的 Solaris OS 软件安装到 solaris10 区域，且该区域必须引导为 online 状态。

如果尚未在 solaris10 标记区域中安装群集核心软件包，则可以通过对群集发行版 DVD 目录指定 -d 选项，对群集软件组件指定 -s 选项以及对修补程序指定 -p 选项来一次安装这些核心软件包、所有群集软件包组件以及所有修补程序。用于安装群集软件组件和修补程序的选项是可选的。

如果已安装群集核心软件包，则仍可以使用该子命令安装区域群集中支持的修补程序和任意群集软件组件。指定修补信息时，必须使用 -o 选项将区域群集的群集节点引导为 offline-running 状态。

solaris10 标记区域群集仅支持共享 IP 区域类型（有关独占 IP 和共享 IP 区域群集的更多信息，请参见[Unresolved link to "Oracle Solaris Cluster 软件安装指南"](#)）。只能从全局区域运行该子命令。

list

显示已配置的区域群集的名称。

该子命令报告在群集中配置的区域群集的名称。

- 如果从全局群集节点运行 list 子命令，该子命令将显示全局群集中的所有区域群集的列表。
- 如果从区域群集节点运行 list 子命令，该子命令将只显示区域群集的名称。

要查看在其中配置区域群集的节点的列表，请使用 -v 选项。

move

将区域路径移至新的区域路径。

该子命令将区域路径移至新的区域路径。

只能从全局群集节点使用 move 子命令。

ready

为应用程序准备区域。
该子命令为运行应用程序准备区域。
只能从全局群集节点使用 ready 子命令。

reboot

重新引导区域群集。
该子命令重新引导区域群集，其类似于发出 halt 子命令，然后发出 boot 子命令。
有关更多信息，请参见 halt 子命令和 boot 子命令。
只能从全局群集节点使用 reboot 子命令。

set

设置使用 -p 选项为区域群集指定的属性的值。您可以从全局区域或区域群集使用 set 子命令。有关可设置的属性的信息，请参见“选项”部分中关于 -p 的描述。

show

显示区域群集的属性。
区域群集的属性包括区域群集名称、标记、IP 类型、节点列表、区域路径和允许的地址。show 子命令从区域群集运行，但仅应用于该特定区域群集。当您从区域群集中使用该子命令时，区域路径始终为 /。如果指定了区域群集名称，该命令将只应用于该区域群集。
只能从全局群集节点使用 show 子命令。

show- rev

显示有关区域群集中每个节点的群集发行版信息。
此功能对于列出区域群集中安装的发行版本和修补程序很有用。例如：

```
# clzonecluster show-rev
=== Zone Clusters ===
Zone Cluster Name:   zc1
Release at vznoda1a on node pnode1:  3.3u2_40u1_zc:2012-04-01
Release at vznoda2a on node pnode2:  3.3u2_40u1_zc:2012-04-01
```

您可以从全局群集节点或区域群集节点使用 show- rev 子命令。

status

确定区域群集节点是否为区域群集的成员并显示区域群集是 solaris、solaris10 还是 labeled 标记区域群集。
区域状态可以为以下状态之一：
Configured、Installed、Ready、Running、Shutting Down 和 Unavailable。将显示全局群集中所有区域群集的状态，以便您能够看到自己的虚拟群集的状态。
要检查区域活动，请使用 zoneadm 命令。

只能从全局群集节点使用 `status` 子命令。

`uninstall`

卸载区域群集。

该子命令卸载区域群集。`uninstall` 子命令使用 `zoneadm` 命令。

只能从全局群集节点使用 `uninstall` 子命令。

`verify`

检查所指定的信息的语法是否正确。

该子命令在区域群集中的每个节点上调用 `zoneadm verify` 命令，以确保能够安全地安装每个区域群集成员。有关更多信息，请参见[Unresolved link to "zoneadm1M"](#)。

只能从全局群集节点使用 `verify` 子命令。

注 - 每个选项的长短两种格式都显示在此部分中。

支持以下选项：

`-?`

`--help`

显示帮助信息。

指定此选项时，*subcommand* 可有可无。

如果没有指定 *subcommand*，将显示所有可用子命令的列表。

如果指定了 *subcommand*，将显示该子命令的用法。

如果指定此选项和其他选项，将会忽略其他选项。

`-a absolute_path_to_archive zoneclustername`

指定以下项的路径：已安装的 Oracle Solaris 10 系统、已安装的 Solaris 10 本机区域或 `solaris10` 标记区域的 `flash_archive`、`cpio`、`pax`、`xus-tar`、`zfs archive` 或 `level 0 ufsdump`。还可以指定统一归档文件的绝对路径。有关更多信息，请参见以下手册页：[Unresolved link to "solaris105"](#)、[Unresolved link to "flash_archive4"](#)、[Unresolved link to "cpio1"](#) 和 [Unresolved link to "pax1"](#)。

`-c config_profile.xml`

`--configprofile config_profile.xml`

为 `solaris` 标记区域群集指定配置文件模板。从系统信息库安装后，该模板会将系统配置信息应用到区域群集的所有节点。如果未指定 `config_profile.xml`，您必须在每个节点上从全局区域运行 `zlogin -C zoneclustername` 命令来手动配置每个区域群集节点。所有配置文件都必须具有 `.xml` 扩展名。

`-c` 选项可替换配置文件模板中区域群集节点的主机名。引导区域群集节点后，该配置文件会应用到该区域群集节点。

`-d absolute_root_path`

`--dirpath dirpatch`

`-d` 选项与 `cluster` 子命令一起使用时，指定已安装的 Oracle Solaris 10 系统的根目录路径。该路径应能在要安装区域群集的群集的所有物理节点上访问。

`-d`

`--dvd-directory dvd-directory`

指定 DVD 映像目录。

`-d` 选项与 `install-cluster` 子命令一起使用时，指定支持 `solaris10` 标记区域的 Oracle Solaris Cluster 发行版的 DVD 映像目录。DVD 映像包括核心软件包和其他群集软件组件（例如区域群集中支持的代理和 Geographic Edition 软件）。必须能够从运行该命令的节点的全局区域中访问该 DVD 目录。

`-d`

`--dry_run`

当 `-d` 选项与 `apply` 子命令一起使用时，将在模拟运行模式下运行重新配置。模拟运行模式不会更改配置，它会让正在运行的区域保持现状。使用模拟运行模式可以检查真实重新配置将执行的操作。

`-f{commandfile | zonepath}`

`--file-argument {commandfile | zonepath}`

与 `configure` 子命令一起使用时，`-f` 选项指定命令文件参数。例如，`clzonecluster configure-fcommandfile`。与 `move` 子命令一起使用时，`-f` 选项指定 `zonepath`。

`-F`

在 `delete` 和 `uninstall` 操作期间，可以使用 `-F` 选项。`-F` 选项强制抑制 Are you sure you want to *do this operation* [y/n]? 问题。

`-m method`

`--method method`

使用 `-m` 选项克隆区域群集。克隆的唯一有效方法为 `copy` 命令。在运行 `clone` 子命令之前，必须停止源区域群集。

`-M manifest.xml`

`--manifest manifest.xml`

使用 `-M` 选项可为 `solaris` 标记区域群集的所有节点指定一个清单。该清单为区域群集安装指定 Oracle Solaris 软件包信息和 Oracle Solaris Cluster 软件包。

`-n nodename[...]`

`--nodelist nodename[...]`

指定子命令的节点列表。

例如，`clzonecluster boot-n phys-schost-1、phys-schost-2 zoneclustername`。

-o

--offline

将区域群集引导或重新引导为 offline-running 模式。

如果区域群集节点不是区域群集的成员但 Oracle Solaris 区域状态为正在运行，则会出现 offline-running 模式。区域群集与物理群集共享引导模式（群集或非群集模式），因此，群集处于脱机模式与处于非群集模式不同。

要将区域群集引导为 offline-running 模式，请键入以下内容。

```
clzonecluster boot [-n phys-schost-1,...] [-o] zoneclustername
```

要将区域群集重新引导为 offline-running 模式，请键入以下内容。

```
clzonecluster reboot [-n phys-schost-1,...] [-o] zoneclustername
```

要将 offline-running 区域重新引导为 online-running 模式，请运行不带 -o 选项的 clzonecluster reboot 命令。

-p name=value

--property=name=value

--property name=value

-p 选项与 install-cluster 子命令和 set 子命令一起使用。有关将 -p 与 install-cluster 子命令一起使用的用法信息，请参见关于 -p patchdir=patchdir[,patchlistfile=patchlistfile] 的描述。

-p 选项与 set 子命令一起使用来指定属性值。允许使用多个 -p name= value 实例。

将该选项与 set 子命令一起使用可修改以下属性：

resource_security

指定 RGM 资源执行程序所用的安全策略。允许的 resource_security 值为 SECURE、WARN、OVERRIDE 或 COMPATIBILITY。

Start 和 Validate 之类的资源方法始终以 root 用户身份运行。如果方法可执行文件不归 root 所有或具有组或全局写权限，将存在不安全状况。在这种情况下，如果 resource_security 属性设置为 SECURE，在运行时资源方法执行将失败并返回错误。如果 resource_security 设置为任何其他值，将允许资源方法执行，但显示警告消息。为了最大限度地保证安全性，请将 resource_security 设置为 SECURE。

resource_security 设置还会修改声明了 application_user 资源属性的资源类型的行为。有关更多信息，请参见 [r_properties\(5\) \[1103\]](#) 手册页的 application_user 部分。

-p patchdir=patchdir[,patchlistfile=patchlistfile]

--patch-specification=patchdir=patchdir[,patchlistfile=patchlistfile]

--patch-specification patchdir=patchdir[,patchlistfile=patchlistfile]

-p 选项指定的 patchdir 和 patchlistfile 属性仅与 install-cluster 子命令一起使用。如果在安装核心软件包后安装修补程序，则必须将区域群集引导为 offline-running 状态才能应用修补程序。

允许使用多个 `-p name= value` 实例。

`patchdir`

指定要应用到 `solaris10` 标记区域的 Oracle Solaris Cluster 修补程序所在的目录。`patchdir` 目录是必需项，且必须能够从区域群集的所有节点上的 `solaris10` 标记区域内部进行访问。

`patchlistfile`

指定 `patchlistfile`。`patchlistfile` 指定包含要安装的修补程序列表的文件。如果未指定可选的 `patchlistfile`，该命令将尝试将安装 `patchdir` 目录内的所有修补程序。您也可以在 `patchdir` 目录中创建一个 `patchlistfile` 列出修补程序 ID（每行一个），从而指出要安装的修补程序。

`-s`

`--software-component {all | software-component[,...]}`

指定要从 DVD 映像安装的软件组件。

这些组件是核心软件包以外的附加组件，可以为在区域群集中支持的数据服务或 Geographic Edition 软件。使用 `-s all` 时，无法指定其他任何组件，将安装所有数据服务和 Geographic Edition 软件。对于数据服务代理，组件名称即为代理名称。对于 Geographic Edition 软件，指定为 `-s geo`。如果未指定 `-s` 选项，则只安装群集框架软件。

`-v`

`--verbose`

在标准输出 (stdout) 中显示详细信息。

`-V`

`--version`

显示该命令的版本。

如果是与其他选项、子命令或操作数一起指定该选项，将忽略所有这些内容。仅显示该命令的版本。不会发生任何其他处理。

`[-x cert|ca-cert|key=file] ...`

如果具有 HTTPS 统一归档文件位置，请指定 SSL 证书、CA 证书和密钥文件。可以多次指定 `-x` 选项。

`-Z target-zoneclustername`

`--zonecluster target-zoneclustername`

您要克隆的区域群集名称。

使用源区域群集名称来克隆。在使用该子命令之前，必须停止源区域群集。

`-z zone`

如果统一归档文件包含多个区域，请指定安装源的区域名称。

资源和属性

clzonecluster 命令支持区域群集的多个资源和属性。

必须使用 clzonecluster 命令配置 clzonecluster 命令支持的任何资源和属性。有关配置 clzonecluster 命令不支持的资源或属性的更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

下面的子部分（“资源”和“属性”）介绍了 clzonecluster 命令支持的那些资源和属性。

资源

以下列出在资源范围中支持的资源类型以及可以在何处找到更多信息：

admin

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。该资源可同时在群集范围和节点范围内使用。该资源会向下传递到单个 Oracle Solaris 区域级别。同时在群集和节点范围中指定该资源时，节点范围资源信息会向下传递到区域群集特定节点的 Oracle Solaris 区域。

admin 资源的 auths 属性可设置为以下值之一：

clone	等同于 solaris.zone.clonefrom
login	等同于 solaris.zone.login
manage	等同于 solaris.zone.manage

capped-cpu

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。该资源可同时在群集范围和节点范围内使用。该资源会向下传递到单个 Oracle Solaris 区域级别。同时在群集和节点范围中指定该资源时，节点范围资源信息会向下传递到区域群集特定节点的 Oracle Solaris 区域。

capped-memory

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。该资源可在群集范围和节点范围内使用。该资源会向下传递到单个 Oracle Solaris 区域级别。同时在群集和节点范围中指定该资源时，节点范围资源信息会向下传递到区域群集特定节点的 Oracle Solaris 区域。

dataset

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。该资源可在群集范围或节点范围内使用。不能同时在群集和节点范围中指定数据集。

在群集范围中该资源用于为高可用性 ZFS 文件系统导出要在区域群集中使用的 ZFS 数据集。导出的数据集由 Oracle Solaris Cluster 软件管理，在群集范围中指定时，不会传递到单个 Oracle Solaris 区域级别。数据集无法在区域群集之间共享。

在节点范围中该资源用于将本地 ZFS 数据集导出到特定区域群集节点。在节点范围中指定时，导出的数据集不由 Oracle Solaris Cluster 软件管理，而会传递到单个 Oracle Solaris 区域级别。

dedicated-cpu

有关更多信息，请参见 [Unresolved link to "zonecfg1M"](#) 手册页。可以在每个节点上使用专用于区域群集的固定数量的 CPU。

该资源可在群集范围和节点范围内使用。该资源会向下传递到单个 Oracle Solaris 区域级别。同时在群集和节点范围中指定该资源时，节点范围资源信息会向下传递到区域群集特定节点的 Oracle Solaris 区域。

device

有关更多信息，请参见 [Unresolved link to "zonecfg1M"](#) 手册页。该资源会向下传递到单个 Oracle Solaris 区域级别并可在群集范围或节点范围中指定。在节点范围中该资源用于添加特定于区域群集节点的设备。可以将设备只添加到一个区域群集。无法同时在群集范围和节点范围中添加相同设备。

fs

有关更多信息，请参见 [Unresolved link to "zonecfg1M"](#) 手册页。可以在群集范围或节点范围中指定该资源。不能同时在群集和节点范围中指定 fs 资源。

在群集范围中该资源通常用于导出要在区域群集中使用的文件系统。导出的文件系统由 Oracle Solaris Cluster 软件管理，而不会向下传递到单个 Oracle Solaris 区域级别，但 `cluster-control` 属性设置为 `false` 的 `lofs` 文件系统除外。有关 `cluster-control` 属性的更多信息，请参见本手册页中“资源”部分中关于 fs 的描述。

在节点范围中该资源用于将本地文件系统导出到特定区域群集节点。在节点范围中指定时，导出的文件系统不由 Oracle Solaris Cluster 软件管理，而会传递到单个 Oracle Solaris 区域级别。

可以通过直接挂载或回送挂载将文件系统导出到区域群集。直接挂载使文件系统在区域群集中可访问，方法是在某个位置挂载指定的文件系统，该位置位于区域根目录或在其路径中具有区域根目录的某个子目录下方。直接挂载意味着文件系统只属于该区域群集。区域群集在 Oracle Solaris Trusted Extensions 上运行时，必须对以读写权限挂载的文件使用直接挂载。区域群集支持对 UFS、QFS 独立文件系统、QFS 共享文件系统和 ZFS（导出为数据集）使用直接挂载。

回送挂载是一种机制，使已在某个位置挂载的文件系统显示为在其他位置挂载。可以通过使用每个区域群集一个回送挂载将单个文件系统导出到多个区域群集。这样便可以在多个区域群集之间共享单个文件系统。在多个区域群集之间共享文件系统前，管理员必须考虑安全隐患。不管如何挂载真实的文件系统，回送挂载可以将访问权限限制为只读。

fs: cluster-control

`cluster-control` 属性仅适用于在群集范围中指定的回送挂载。`cluster-control` 属性的默认值为 `true`。

当属性值为 `true` 时，Oracle Solaris Cluster 管理该文件系统，但不会将文件系统信息传递给 `zonecfg` 命令。区域引导后，Oracle Solaris Cluster 根据需要在区域群集节点中挂载和取消挂载文件系统。

Oracle Solaris Cluster 可以为 QFS 共享文件系统、UFS、QFS 独立文件系统和 UFS 上的 PxFS 管理回送挂载。

当属性值为 `false` 时，Oracle Solaris Cluster 不管理文件系统。群集软件将该文件系统信息和所有关联的信息传递到 `zonecfg` 命令，该命令在每台计算机上创建区域群集区域。在这种情况下，当区域引导时，Oracle Solaris 软件将挂载文件系统。管理员可以将该选项用于 UFS 文件系统。

管理员可以在群集范围中指定回送挂载。为回送挂载配置 `cluster-control` 属性值 `false` 对通用本地目录（如包含可执行文件的目录）的只读挂载很有用。此信息会传递到执行实际挂载的 `zonecfg` 命令。使用值为 `true` 的 `cluster-control` 属性配置回送挂载对于将全局文件系统 (PxFS) 或 QFS 共享文件系统提供给受群集控制的区域群集很有用。

QFS 共享文件系统、UFS、QFS 独立文件系统和 ZFS 最多可配置在一个区域群集中。

net

有关 net 资源的更多信息，请参见 [Unresolved link to "zonecfg1M"](#) 手册页。

由 Oracle Solaris Cluster 管理的任何网络资源（如逻辑主机或共享地址）是在群集范围中指定的。由应用程序管理的任何网络资源（如 Oracle RAC VIP）是在群集范围中指定的。这些网络资源不会传递到单个 Oracle Solaris 区域级别。

管理员可以指定网络接口卡 (Network Interface Card, NIC) 以用于指定的 IP 地址。系统自动选择满足下列两个要求的 NIC：

- NIC 已连接到同一个子网。
- 已为该区域群集配置 NIC。

node

节点资源执行下列两个目标：

- 标识范围级别。在节点范围中指定的任何资源只属于该特定节点。
- 标识区域群集的节点。管理员可以标识将运行区域的计算机，方法是在该计算机上标识全局群集全局区域。为每个区域群集节点指定 IP 地址和 NIC 是可选的。管理员还可以指定用于标识网络信息以访问该节点的信息。

注 - 如果管理员没有为每个区域群集节点配置 IP 地址，将出现以下两种情况：

1. 该特定区域群集将无法配置要在区域群集中使用的 NAS 设备。群集在与 NAS 设备通信时将使用区域群集节点的 IP 地址，所以缺失 IP 地址会导致群集不支持隔离 NAS 设备。
 2. 群集软件将激活所有 NIC 上的所有逻辑主机 IP 地址。
-

privnet

该资源可在节点范围中使用。该资源指定可用作区域群集的专用适配器的数据链路设备。该资源在指定给区域群集之前必须在全局区域中可用。配置独占 IP 区域群集时，默认将 `enable_priv_net` 属性设置为 `true`，以在区域群集节点间启用专用网络通信。

```
add node
add
privnet
set
physical=vnic1
end
add
privnet
set
physical=vnic5
end
end
```

资源属性 `privnet` 的顺序用于构成区域群集节点间的路径。第一个节点中指定的第一个 `privnet` 适配器将尝试用第二个节点中指定的第一 `privnet` 路径构成路径。在添加和删除操作后将保留 `privnet` 资源的顺序。

注 - `privnet` 资源无法在多个独占 IP 区域中共享。必须将其指定给一个特定的独占 IP 区域。

rctl

有关更多信息，请参见 [Unresolved link to "zonecfg1M"](#) 手册页。该资源可同时在群集范围和节点范围内使用。该资源会向下传递到单个 Oracle Solaris 区域级别。同时在群集和节点范围中指定该资源时，节点范围资源信息会向下传递到区域群集特定节点的 Oracle Solaris 区域。

sysid

请参见 [Unresolved link to "sysidcfg4"](#) 手册页。该资源为 `solaris10` 区域群集的所有区域指定系统标识参数。

属性

每个资源类型具有一个或多个属性。支持群集的以下属性：

(群集)

`admin`

有关更多信息，请参见 [Unresolved link to "zonecfg1M"](#) 手册页。

(群集)

`allowed-address`

指定可在适配器上激活的 IP 地址。仅允许特定的 IP 地址。该可选属性用于节点范围的网络资源。例如：

```
set allowed-address=1.2.2.3/24
```

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

(群集)

attr

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。区域群集将使用设置为 cluster 的属性 name、设置为 boolean 的属性 type 和设置为 true 的属性 value。使用 create 选项配置区域群集时，将为这些属性设置默认值。这些属性是用于区域群集配置的强制性属性，不能更改。

(群集)

autoboot

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

(群集)

bootargs

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

(群集)

brand

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。仅支持 solaris、solaris10 和 labeled 标记类型。

(群集)

cpu-shares

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

(群集)

device

[Unresolved link to "zonecfg1M"](#)。

(群集)

enable_priv_net

设置为 true 时，在区域群集的节点之间将启用 Oracle Solaris Cluster 专用网络通信。

- 如果 ip-type 设置为 shared，区域群集节点之间的通信使用全局群集的专用网络。
- 如果 ip-type 设置为 exclusive，区域群集节点之间的通信使用指定的 privnet 资源。如果未指定这些资源，将会通过在全局群集的专用网络上创建虚拟网络接口 (vnic) 来自动生成这些资源。

区域群集节点的 Oracle Solaris Cluster 专用主机名和 IP 地址是由系统自动生成的。如果值设置为 false，将禁用专用网络。默认值为 true。

(群集)

ip-type

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。仅支持 shared 和 exclusive 值。

(群集)

limitpriv

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

(群集)

max-lwps

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

(群集)

max-msg-ids

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

(群集)

max-sem-ids

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

(群集)

max-shm-ids

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

(群集)

monitor_quantum

定义量程值的毫秒数。

(群集)

monitor_timeout

指定监视器超时的毫秒数。

(群集)

max-shm-memory

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

(群集)

pool

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

(群集)

zonename

区域群集的名称，也是区域群集中每个区域的名称。

(群集)

zonepath

区域群集中每个区域的区域路径。

admin

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

capped-cpu

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

capped-memory

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

dataset

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

dedicated-cpu

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

device

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

fs

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

inherit pkg-dir

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

net

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

node

包括 physical-host、hostname 和 net。

- physical-host – 该属性指定将托管区域群集节点的全局群集节点。
- hostname – 该属性指定由物理主机属性指定的全局群集节点上区域群集节点的公共主机名。
- net – 该资源指定区域群集节点用于公共网络通信的网络地址和物理接口名称，该区域群集节点在由物理主机指定的全局群集节点上。

rctl

请参见 [Unresolved link to "zonecfg1M"](#)。

sysid

使用 `/usr/bin/sysconfig configure` 命令。

请参见 [Unresolved link to "sysidcfg4"](#)。包括

`root_password`、`name_service`、`security_policy`、`system_locale`、`timezone`、`terminal` 和 `nfs4_domain`。管理员稍后可以按照正常 Oracle Solaris 过程手动更改任何 `sysidcfg config` 值（一次更改一个节点）。

- `root_password` – 该属性为区域群集的所有节点指定公用 root 密码的加密值。请勿指定明文密码。必须使用 `/etc/shadow` 中的加密密码字符串。这是一个必需属性。
- `name_service` – 该可选属性指定要在区域群集中使用的命名服务。但是，全局区域的 `/etc/sysidcfg` 文件中的设置可能已过时。为确保该属性具有正确的设置，请使用 `clzonecluster` 命令手动输入该值。
- `security_policy` – 该值在默认情况下设置为 `none`。
- `system_locale` – 默认情况下该值从 `clzonecluster` 命令的环境中获得。
- `timezone` – 该属性指定要在区域群集中使用的时区。默认情况下该值从 `clzonecluster` 命令的环境中获得。
- `terminal` – 该值在默认情况下设置为 `xterm`。
- `nfs4_domain` – 该值在默认情况下设置为 `dynamic`。

在所有示例中，`zoneclustername` 为 `sczone`。第一个全局群集节点为 `phys-schost-1`，第二个节点为 `phys-schost-2`。第一个区域群集节点为 `zc-host-1`，第二个节点为 `zc-host-2`。

例 261 创建新的区域群集

以下示例说明了如何创建双节点 `solaris10` 标记区域群集。zpool "tank" 委派给区域用作高可用性 ZFS 文件系统。内存限制用于限制可用于区域群集的内存量。除 root 用户密码之外，将使用默认系统标识值。

```
phys-schost-1# clzonecluster configure sczone
sczone: No such zone cluster configured
Use 'create' to begin configuring a new zone cluster.
clzc:sczone> create -b
clzc:sczone> set zonepath=/zones/timuzc
clzc:sczone> set brand=solaris10
clzc:sczone> set autoboot=true
clzc:sczone> set bootargs="-m verbose"
clzc:sczone> set limitpriv="default,proc_priocntl,proc_clock_highres"

clzc:sczone> set enable_priv_net=true
clzc:sczone> set ip-type=shared
```

```

clzc:sczone> add dataset
clzc:sczone:dataset> set name=tank
clzc:sczone:dataset> end
clzc:sczone> add capped-memory
clzc:sczone:capped-memory> set physical=3G
clzc:sczone:capped-memory> end
clzc:sczone> add rctl
clzc:sczone:rctl> set name=zone.max-swap
clzc:sczone:rctl> add value (priv=privileged,limit=4294967296,action=deny)

clzc:sczone:rctl> end
clzc:sczone> add rctl
clzc:sczone:rctl> set name=zone.max-locked-memory
clzc:sczone:rctl> add value (priv=privileged,limit=3221225472,action=deny)

clzc:sczone:rctl> end
clzc:sczone> add attr
clzc:sczone:attr> set name=cluster
clzc:sczone:attr> set type=boolean
clzc:sczone:attr> set value=true
clzc:sczone:attr> end
clzc:sczone> add node
clzc:sczone:node> set physical-host=ptimu1
clzc:sczone:node> set hostname=zc-host-1
clzc:sczone:node> add net
clzc:sczone:node:net> set address=vztimula
clzc:sczone:node:net> set physical=sc_ipmp0
clzc:sczone:node:net> end
clzc:sczone:node> end
clzc:sczone> add node
clzc:sczone:node> set physical-host=ptimu2
clzc:sczone:node> set hostname=zc-host-2
clzc:sczone:node> add net
clzc:sczone:node:net> set address=vztimu2a
clzc:sczone:node:net> set physical=sc_ipmp0
clzc:sczone:node:net> end
clzc:sczone:node> end
clzc:sczone> add fs
clzc:sczone:fs> set dir=/opt/local
clzc:sczone:fs> set special=/usr/local
clzc:sczone:fs> set type=lofs
clzc:sczone:fs> add options [ro,nodevices]
clzc:sczone:fs> set cluster-control=false
clzc:sczone:fs> end
clzc:sczone> add sysid
clzc:sczone> set root_password=Ziith.NOLOrRg
clzc:sczone> set name_service="NIS{domain_name=mycompany.com name_server=
ns101c-90(10.100.10.10)}"
clzc:sczone> set nfs4_domain=dynamic
clzc:sczone> set security_policy=NONE
clzc:sczone> set system_locale=C
clzc:sczone> set terminal=xterms

```

```
clzc:sczone> set timezone=US/Pacific
clzc:sczone> end
```

如果使用 `create` 子命令（而不是上面所示的 `create -b` 子命令），将使用默认模板，该模板已设置 `attr` 属性。

区域群集现已配置。下列命令安装然后从全局群集节点引导区域群集：

```
phys-schost-1# clzonecluster install -a absolute_path_to_archive install sczone
phys-schost-1# clzonecluster boot sczone
```

例 262 从统一归档文件创建区域群集

以下示例说明如何从统一归档文件创建和安装区域群集。统一归档文件可以从全局区域、非全局区域或区域群集节点创建。支持从克隆归档文件和恢复归档文件这两种统一归档文件配置和安装区域群集。如果统一归档文件是从非群集区域创建的，则必须设置以下属性：`enable_priv_net=true`。您还可以根据需要更改任何区域属性。

```
phys-schost-1# clzonecluster configure sczone
sczone: No such zone cluster configured
Use 'create' to begin configuring a new zone cluster.
clzc:sczone> create -a absolute_path_to_archive -z archived_zone_1
clzc:sczone> set zonepath=/zones/sczone

clzc:sczone> set enable_priv_net=true
clzc:sczone> set ip-type=shared

clzc:sczone> add attr
clzc:sczone:attr> set name=cluster
clzc:sczone:attr> set type=boolean
clzc:sczone:attr> set value=true
clzc:sczone:attr> end

clzc:sczone> add node
clzc:sczone:node> set physical-host=psoft1
clzc:sczone:node> set hostname=zc-host-1
clzc:sczone:node> add net
clzc:sczone:node:net> set address=vzsoft1a
clzc:sczone:node:net> set physical=sc_ipmp0
clzc:sczone:node:net> end
clzc:sczone:node> end
clzc:sczone> add node
clzc:sczone:node> set physical-host=psoft2
clzc:sczone:node> set hostname=zc-host-2
clzc:sczone:node> add net
clzc:sczone:node:net> set address=vzsoft2a
clzc:sczone:node:net> set physical=sc_ipmp0
clzc:sczone:node:net> end
clzc:sczone:node> end
```

区域群集现已配置。以下命令从全局群集节点上的统一归档文件安装区域群集：

```
phys-schost-1# clzonecluster install -a absolute_path_to_archive -z archived-zone sczone
```

区域群集现已安装。以下命令将引导区域群集：

```
phys-schost-1# clzonecluster boot sczone
```

例 263 修改现有的区域群集

以下示例显示如何修改在示例 1 中创建的区域群集的配置。额外的公用 IP 地址将添加到 phys-schost-2 上的区域群集节点中。

UFS 文件系统将导出到区域群集，以用作高度可用的文件系统。假设 UFS 文件系统是在 Oracle Solaris Volume Manager 元设备上创建的。

```
phys-schost-1# clzonecluster configure sczone
clzc:sczone> add device
clzc:sczone:device> set match=/dev/md/1/dsk/d100
clzc:sczone:device> end
clzc:sczone> add device
clzc:sczone:device> set match=/dev/md/oraset/dsk/d100
clzc:sczone:device> end
clzc:sczone> select node physical-host=phys-schost-2
clzc:sczone:node> add net
clzc:sczone:node:net> set address=192.168.0.3/24
clzc:sczone:node:net> set physical=bge0
clzc:sczone:node:net> end
clzc:sczone:node> end
clzc:sczone> add fs
clzc:sczone:fs> set dir=/qfs/ora_home
clzc:sczone:fs> set special=oracle_home
clzc:sczone:fs> set type=samfs
clzc:sczone:fs> end
clzc:sczone> exit
```

例 264 使用现有区域群集作为模板来创建新的区域群集

以下示例显示如何使用在示例 1 中创建的 sczone1 区域群集作为模板来创建名为 sczone 的区域群集。新区域群集的配置将与原始区域群集相同。需要修改新区域群集的某些属性来避免冲突。如果管理员删除资源类型而不指定特定资源，系统将删除该类型的所有资源。例如，**remove net** 将导致删除所有网络资源。

```
phys-schost-1# clzonecluster configure sczone1
sczone1: No such zone cluster configured
Use 'create' to begin configuring a new zone cluster.

clzc:sczone1> create -t sczone
clzc:sczone1>set zonepath=/zones/sczone1

clzc:sczone1> select node physical-host=phys-schost-1
clzc:sczone1:node> set hostname=zc-host-3
```

```
clzc:sczone1:node> select net address=zc-host-1
clzc:sczone1:node:net> set address=zc-host-3
clzc:sczone1:node:net> end
clzc:sczone1:node> end
clzc:sczone1> select node physical-host=phys-schost-2
clzc:sczone1:node> set hostname=zc-host-4
clzc:sczone1:node> select net address=zc-host-2
clzc:sczone1:node:net> set address=zc-host-4
clzc:sczone1:node:net> end
clzc:sczone1:node> remove net address=192.168.0.3/24
clzc:sczone1:node> end
clzc:sczone1> remove dataset name=tank/home
clzc:sczone1> remove net
clzc:sczone1> remove device
clzc:sczone1> remove fs dir=/qfs/ora_home
clzc:sczone1> exit
```

支持以下操作数：

zoneclustername 区域群集的名称。可以指定新区域群集的名称。所有子命令均支持 *zoneclustername* 操作数。

+ 群集中的所有节点。只有一部分子命令支持 + 操作数。

[Intro\(1CL\) \[17\]](#) 手册页列出了该命令集中所有命令的一整套退出状态代码。

如果命令对于所有指定的操作数均成功执行，则返回零 (CL_NOERR)。如果某个操作数出错，命令会继续处理操作数列表中的下一个操作数。返回的退出代码始终反映最先发生的错误。

此命令返回下列退出状态代码：

0 CL_NOERR

没有错误。
您发出的命令已成功完成。

1 CL_ENOMEM

交换空间不足。
某个群集节点耗尽了交换内存或其他操作系统资源。

3 CL_EINVAL

参数无效。
您键入的命令不正确，或者使用 -i 选项提供的群集配置信息语法不正确。

18 CL_EINTERNAL

遇到内部错误。

36 CL_ENOENT

没有这样的对象

由于以下某个原因，找不到您所指定的对象：

- 该对象不存在。
- 您尝试使用 -o 选项创建的配置文件路径中的某个目录不存在。
- 您尝试使用 -i 选项访问的配置文件包含错误。

37 CL_EOP

不允许操作

您尝试对不受支持的配置执行操作，或者您所执行的操作不受支持。

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

[clnode\(1CL\) \[149\]](#)、[cluster\(1CL\) \[461\]](#)、[Intro\(1CL\) \[17\]](#)、[scinstall\(1M\) \[697\]](#)、[Unresolved link to "zoneadm1M"](#)、[Unresolved link to "zonecfg1M"](#)、[Unresolved link to "clconfiguration\(5CL\)"](#)

超级用户可以运行此命令的所有格式。

所有用户都可以运行带有 -? (帮助) 或 -v (版本) 选项的此命令。

要运行带有子命令的 `clzonecluster` 命令，超级用户以外的用户需要具有 RBAC 授权。请参见下表。

子命令	RBAC 授权
boot	solaris.cluster.admin
check	solaris.cluster.read
clone	solaris.cluster.admin
configure	solaris.cluster.admin
delete	solaris.cluster.admin
export	solaris.cluster.admin

子命令	RBAC 授权
halt	solaris.cluster.admin
install	solaris.cluster.admin
list	solaris.cluster.read
monitor	solaris.cluster.modify
move	solaris.cluster.admin
ready	solaris.cluster.admin
reboot	solaris.cluster.admin
show	solaris.cluster.read
status	solaris.cluster.read
uninstall	solaris.cluster.admin
unmonitor	solaris.cluster.modify
verify	solaris.cluster.admin

名称

clzonecluster, clzc — 创建和管理区域群集

```
/usr/cluster/bin/clzonecluster [subcommand] -?  
  
/usr/cluster/bin/clzonecluster -V  
  
/usr/cluster/bin/clzonecluster subcommand [options] -v  
    [zone-cluster-name]  
  
/usr/cluster/bin/clzonecluster apply [-n node-name[,...]] [-d]  
    {+ | zone-cluster-name [...]}  
  
/usr/cluster/bin/clzonecluster boot [-n node-name[,...]] [-o]  
    {+ | zone-cluster-name [...]}  
  
/usr/cluster/bin/clzonecluster clone -Z target-zone-cluster-name  
    [-m method][-n node-name[,...]] {source-zone-cluster-name}  
  
/usr/cluster/bin/clzonecluster configure [-f command-file]  
    zone-cluster-name  
  
/usr/cluster/bin/clzonecluster delete [-F] zone-cluster-name  
  
/usr/cluster/bin/clzonecluster export [-f command-file]  
    zone-cluster-name  
  
/usr/cluster/bin/clzonecluster halt [-n node-name[,...]]  
    {+ | zone-cluster-name}  
  
/usr/cluster/bin/clzonecluster install [-c config_profile.xml]  
    [-M manifest.xml] zone-cluster-name  
  
/usr/cluster/bin/clzonecluster install [-n node-name]  
    -a absolute_path_to_archive [-x cert|ca-cert|key=file]...  
    -z zone zone-cluster-name  
  
/usr/cluster/bin/clzonecluster install [-n node-name]  
    -d absolute_root_path zone-cluster-name  
  
/usr/cluster/bin/clzonecluster install-cluster  
    [-d dvd-image] [-n node-name[,...]]  
    [-p patchdir=patch-dir[,patchlistfile=file-name]]  
    -s software-component[,...] [-v] zone-cluster-name  
  
/usr/cluster/bin/clzonecluster install-cluster  
    [-p patchdir=patch-dir[,patchlistfile=file-name]  
    [-n node-name[,...]] [-v] zone-cluster-name  
  
/usr/cluster/bin/clzonecluster list [+ | zone-cluster-name [...]]  
  
/usr/cluster/bin/clzonecluster move -f zone-path zone-cluster-name  
  
/usr/cluster/bin/clzonecluster ready [-n node-name[,...]]  
    {+ | zone-cluster-name [...]}
```

```
/usr/cluster/bin/clzonecluster reboot [-n node-name[,...]] [-o]
    {+ | zone-cluster-name [...]}

/usr/cluster/bin/clzonecluster set {-p name=value}
    [-p name=value] [...] [zone-cluster-name]

/usr/cluster/bin/clzonecluster show [+ | zone-cluster-name [...]]

/usr/cluster/bin/clzonecluster show-rev [-v] [-n node-name[,...]]
    [+ | zone-cluster-name ...]

/usr/cluster/bin/clzonecluster status [+ | zone-cluster-name [...]]

/usr/cluster/bin/clzonecluster uninstall [-F] [-n node-name
    [...]] zone-cluster-name

/usr/cluster/bin/clzonecluster verify [-n node-name[,...]]
    {+ | zone-cluster-name [...]}
```

`clzonecluster` 命令为 Oracle Solaris Cluster 配置创建和修改区域群集。`clzc` 命令是 `clzonecluster` 命令的简短格式；这两个命令是相同的。`clzonecluster` 命令可以识别群集，并支持单个管理源。可以从一个节点发出命令的所有格式来影响单个区域群集节点或所有节点。

仅当 *options* 是 `-?` 选项或 `-v` 选项时，才可以省略 *subcommand*。

除了 `list`、`show` 和 `status` 子命令以外，子命令都至少需要一个操作数。但是，许多子命令接受使用加号操作数 (+) 来将子命令应用到所有适用的对象。`clzonecluster` 命令可以在区域群集的任何节点上运行，并且可以影响任何或所有区域群集。

每个选项都有长和短两种格式。在 选项 中给出每个选项的两种格式以及该选项的描述。

注 - 创建区域群集之后无法更改区域群集名称。

支持以下子命令：

`apply`

应用对区域群集进行的配置更改。

`apply` 子命令可应用对区域群集进行的永久实时重新配置。您应运行 `clzonecluster configure` 以进行配置更改，然后运行 `apply` 子命令以应用对特定区域群集进行的更改。`apply` 子命令使用 `-n` 选项来指定将应用重新配置的节点的列表。

只能从全局群集节点使用 `apply` 子命令。

`boot`

引导区域群集。

`boot` 子命令引导区域群集。`boot` 子命令使用 `-n` 标志来为节点的特定列表引导区域群集。

只能从全局群集节点使用 `boot` 子命令。

clone

克隆区域群集。

`clone` 命令通过复制已安装的现有区域群集来安装区域群集。该子命令是另一种安装区域群集的方式。`clone` 子命令本身不创建新的区域群集。确保用于克隆的源区域群集在克隆之前处于“已安装”（未运行）状态。必须先使用 `configure` 子命令创建新的区域群集。然后使用 `clone` 子命令将克隆的配置应用到新的区域群集。

只能从全局群集节点使用 `clone` 子命令。

configure

启动交互式实用程序配置 `solaris10` 或 `labeled` 标记区域群集。

`configure` 子命令使用 `zonecfg` 命令在指定的每台计算机上配置区域。 `configure` 子命令允许您指定适用于区域群集的每个节点的属性。这些属性与 `zonecfg` 命令为各个区域建立的属性具有相同的意义。 `configure` 子命令支持对 `zonecfg` 命令未知的属性进行配置。如果未指定 `-f` 选项， `configure` 子命令将启动交互式 shell。 `-f` 选项将命令文件作为其参数。 `configure` 子命令使用该文件以非交互方式创建或修改区域群集。

通过 `configure` 子命令，还可以使用统一归档文件配置区域群集并选择恢复归档文件或克隆归档文件。将 `-a archive` 选项与 `create` 子命令一起使用。例如：

```
# /usr/cluster/bin/clzc configure sczone1
sczone1: No such zone cluster configured
Use 'create' to begin configuring a new zone cluster.
clzc:sczone1> create -a archive -z archived zone
```

只能从全局群集节点使用 `configure` 子命令。有关更多信息，请参见[Unresolved link to "Oracle Solaris Cluster 软件安装指南"](#)。

要指定 `solaris10` 标记区域群集，在配置该区域群集时可以使用默认模板。默认模板位于 `/etc/cluster/zone_cluster/ORCLcls10default.xml`。您可以使用 `-t` 选项指定默认 `solaris10` 区域群集模板或群集上的其他现有 `solaris10` 区域群集。如果指定了其他 `solaris10` 区域群集，将从指定的区域群集导入区域群集配置。还必须在 `sysid` 属性中指定 `root` 用户密码，以保证 `verify` 或 `commit` 操作不会失败。键入以下命令应用模板：

```
# /usr/cluster/bin/clzc configure sczone2
sczone2: No such zone cluster configured
Use 'create' to begin configuring a new zone cluster.
clzc:sczone2> create -t ORCLcls10default
clzc:sczone2> info
zonename: sczone2
zonepath:
autoboot: true
hostid:
brand: solaris10
```

`configure` 命令的交互和非交互格式都支持使用多个子命令来编辑区域群集配置。有关可用配置子命令的列表，请参见 [Unresolved link to "zonecfg1M"](#)。

交互式 `configure` 实用程序使您能够创建和修改区域群集的配置。区域群集配置由许多资源类型和属性组成。`configure` 实用程序使用范围概念来确定在何处应用子命令。`configure` 实用程序使用的范围级别有三个：群集、资源和节点特定资源。默认范围为群集。以下列表描述范围的三个级别：

- 群集范围 – 影响整个区域群集的属性。如果 `zoneclustername` 是 `sczone`，`clzonecluster` 命令的交互式 shell 类似于以下内容：

```
clzc:sczone>
```

- 节点范围 – 嵌套在节点资源范围内的特殊资源范围。节点范围中的设置影响区域群集中的特定节点。例如，可以将网络资源添加到区域群集中的特定节点。`clzonecluster` 命令的交互式 shell 类似于以下内容：

```
clzc:sczone:node:net>
```

- 资源范围 – 适用于某个特定资源的属性。资源范围提示附加有资源类型的名称。例如，`clzonecluster` 命令的交互式 shell 类似于以下内容：

```
clzc:sczone:net>
```

delete

删除特定区域群集。

该子命令删除特定区域群集。使用通配符操作数 (*) 时，`delete` 命令将删除在全局群集上配置的区域群集。在运行 `delete` 子命令之前，区域群集必须处于已配置状态。不论区域群集处于什么状态，将 `-F` 选项与 `delete` 命令一起使用都会尝试删除区域群集。

只能从全局群集节点使用 `delete` 子命令。

export

将区域群集配置导出到命令文件。

导出的 `commandfile` 可用作 `configure` 子命令的输入。根据需要修改文件以反映要创建的配置。有关更多信息，请参见 [Unresolved link to "clconfiguration\(5CL\)"](#) 手册页。

只能从全局群集节点使用 `export` 子命令。

halt

停止一个区域群集或该区域群集中的特定节点。

指定特定区域群集时，`halt` 子命令仅应用于该特定区域群集。您可以停止整个区域群集，或仅停止区域群集的特定节点。如果未指定区域群集，`halt` 子命令将应用于所有区域群集。也可以停止指定计算机上的所有区域群集。

`halt` 子命令使用 `-n` 选项来停止特定节点上的区域群集。默认情况下，`halt` 子命令停止所有节点上的所有区域群集。如果指定 + 操作数而不是区域名称，将停止所有区域群集。

只能从全局群集节点使用 `halt` 子命令。

`install`

安装区域群集。

该子命令安装区域群集。

如果使用 `install -M manifest.xml` 选项，则您指定的清单将用于区域群集的所有节点上的安装。清单文件描述管理员进行安装所需的 `solaris` 软件包信息，如 `certificate_file`、`key_file`、发布者以及任何其他软件包。`manifest.xml` 文件还必须指定用于区域群集安装的 Oracle Solaris Cluster 组软件包 `ha-cluster-full`、`ha-cluster-framework-full`、`ha-cluster-data-services-full` 或 `ha-cluster-minimal`。有关自动化安装程序清单的更多信息，请参见 [Unresolved link to "安装 Oracle Solaris 11.2 系统中的创建定制 AI 清单"](#)。

如果不使用 `-M` 选项（默认选项），则会将位于 `/usr/share/auto_install/manifest/zone_default.xml` 的自动化安装程序清单用于安装。使用此 `zone_default.xml` 清单时，在发出命令的区域群集节点的全局区域中安装的所有 `ha-cluster/*` 软件包均会安装在区域群集的所有节点中。如果在安装区域群集时使用定制清单并且未指定 Oracle Solaris Cluster 组软件包，安装将失败。

要安装的所有区域群集节点的底层全局区域与发出 `install` 子命令的区域群集节点的全局区域中安装的 Oracle Solaris Cluster 软件包集必须完全相同。如果存在不满足此要求的区域群集节点，则无法在此节点上完成区域群集安装。

只能从全局群集节点使用 `install` 子命令。`-M` 和 `-c` 选项只能用于 `solaris` 标记区域群集。

如果区域群集的标记是 `solaris10`，则必须使用 `-a` 或 `-d` 选项。

`-a archive`

`solaris` 或 `solaris10` 标记区域群集的统一归档文件的绝对路径、`solaris10` 标记区域群集的 `flar` 归档文件位置或者要用于安装的 Oracle Solaris 10 映像归档文件。有关支持的归档类型的详细信息，请参见 [Unresolved link to "solaris105"](#) 手册页。该归档绝对路径应能在要安装区域群集的群集的所有物理节点上访问。统一归档文件安装可以使用恢复归档文件或克隆归档文件。

`-d path`

已安装的 Oracle Solaris 10 系统的根目录路径。该路径应能在要安装区域群集的群集的所有物理节点上访问。

`[-x cert[ca-cert|key=file]..`

如果具有 HTTPS 统一归档文件位置，请指定 SSL 证书、证书颁发机构 (Certificate Authority, CA) 证书和密钥文件。可以多次指定 `-x` 选项。

`-z zone`

如果统一归档文件包含多个区域，请指定配置源或安装源的区域名称。

将使用同一归档或已安装的 Oracle Solaris 10 系统作为区域群集中所有 solaris10 标记区域的安装源。安装将使用在区域群集配置期间在 sysid 资源类型中指定的系统标识参数覆盖源归档或已安装的 Oracle Solaris 10 中的系统标识参数。

install-cluster

install-cluster 子命令在 solaris10 标记区域群集节点中安装支持 Oracle Solaris 10 OS 的 Oracle Solaris Cluster 软件。安装的软件包括核心软件包、群集软件包组件（例如，区域群集中支持的代理和 Geographic Edition 软件）和修补程序。

注 - install-cluster 子命令不支持在 solaris10 标记区域群集节点中安装 Oracle Solaris Cluster 版本 3.3 或 3.3 5/11 软件。有关 solaris10 标记区域群集支持的发行版的更多信息，请查阅[Unresolved link to "Oracle Solaris Cluster 4.2 发行说明"](#)。

当 solaris10 标记区域安装的 Oracle Solaris 10 系统未装有群集软件时，可使用该子命令。要使用该子命令，必须使用 clzonecluster install 命令将 Oracle Solaris 10 系统的 Solaris OS 软件安装到 solaris10 区域，且该区域必须引导为 online 状态。

如果尚未在 solaris10 标记区域中安装群集核心软件包，则可以通过对群集发行版 DVD 目录指定 -d 选项，对群集软件组件指定 -s 选项以及对修补程序指定 -p 选项来一次安装这些核心软件包、所有群集软件包组件以及所有修补程序。用于安装群集软件组件和修补程序的选项是可选的。

如果已安装群集核心软件包，则仍可以使用该子命令安装区域群集中支持的修补程序和任意群集软件组件。指定修补信息时，必须使用 -o 选项将区域群集的群集节点引导为 offline-running 状态。

solaris10 标记区域群集仅支持共享 IP 区域类型（有关独占 IP 和共享 IP 区域群集的更多信息，请参见[Unresolved link to "Oracle Solaris Cluster 软件安装指南"](#)）。只能从全局区域运行该子命令。

list

显示已配置的区域群集的名称。

该子命令报告在群集中配置的区域群集的名称。

- 如果从全局群集节点运行 list 子命令，该子命令将显示全局群集中的所有区域群集的列表。
- 如果从区域群集节点运行 list 子命令，该子命令将只显示区域群集的名称。

要查看在其中配置区域群集的节点的列表，请使用 -v 选项。

move

将区域路径移至新的区域路径。

该子命令将区域路径移至新的区域路径。

只能从全局群集节点使用 move 子命令。

ready

为应用程序准备区域。
该子命令为运行应用程序准备区域。
只能从全局群集节点使用 ready 子命令。

reboot

重新引导区域群集。
该子命令重新引导区域群集，其类似于发出 halt 子命令，然后发出 boot 子命令。
有关更多信息，请参见 halt 子命令和 boot 子命令。
只能从全局群集节点使用 reboot 子命令。

set

设置使用 -p 选项为区域群集指定的属性的值。您可以从全局区域或区域群集使用 set 子命令。有关可设置的属性的信息，请参见“选项”部分中关于 -p 的描述。

show

显示区域群集的属性。
区域群集的属性包括区域群集名称、标记、IP 类型、节点列表、区域路径和允许的地址。show 子命令从区域群集运行，但仅应用于该特定区域群集。当您从区域群集中使用该子命令时，区域路径始终为 /。如果指定了区域群集名称，该命令将只应用于该区域群集。
只能从全局群集节点使用 show 子命令。

show-rev

显示有关区域群集中每个节点的群集发行版信息。
此功能对于列出区域群集中安装的发行版本和修补程序很有用。例如：

```
# clzonecluster show-rev
=== Zone Clusters ===
Zone Cluster Name:   zc1
Release at vznoda1a on node pnode1:  3.3u2_40u1_zc:2012-04-01
Release at vznoda2a on node pnode2:  3.3u2_40u1_zc:2012-04-01
```

您可以从全局群集节点或区域群集节点使用 show-rev 子命令。

status

确定区域群集节点是否为区域群集的成员并显示区域群集是 solaris、solaris10 还是 labeled 标记区域群集。
区域状态可以为以下状态之一：Configured、Installed、Ready、Running、Shutting Down 和 Unavailable。将显示全局群集中所有区域群集的状态，以便您能够看到自己的虚拟群集的状态。
要检查区域活动，请使用 zoneadm 命令。

只能从全局群集节点使用 `status` 子命令。

`uninstall`

卸载区域群集。

该子命令卸载区域群集。`uninstall` 子命令使用 `zoneadm` 命令。

只能从全局群集节点使用 `uninstall` 子命令。

`verify`

检查所指定的信息的语法是否正确。

该子命令在区域群集中的每个节点上调用 `zoneadm verify` 命令，以确保能够安全地安装每个区域群集成员。有关更多信息，请参见[Unresolved link to "zoneadm1M"](#)。

只能从全局群集节点使用 `verify` 子命令。

注 - 每个选项的长短两种格式都显示在此部分中。

支持以下选项：

`-?`

`--help`

显示帮助信息。

指定此选项时，*subcommand* 可有可无。

如果没有指定 *subcommand*，将显示所有可用子命令的列表。

如果指定了 *subcommand*，将显示该子命令的用法。

如果指定此选项和其他选项，将会忽略其他选项。

`-a absolute_path_to_archive zoneclustername`

指定以下项的路径：已安装的 Oracle Solaris 10 系统、已安装的 Solaris 10 本机区域或 `solaris10` 标记区域的 `flash_archive`、`cpio`、`pax`、`xus-tar`、`zfs archive` 或 `level 0 ufsdump`。还可以指定统一归档文件的绝对路径。有关更多信息，请参见以下手册页：[Unresolved link to "solaris105"](#)、[Unresolved link to "flash_archive4"](#)、[Unresolved link to "cpio1"](#) 和 [Unresolved link to "pax1"](#)。

`-c config_profile.xml`

`--configprofile config_profile.xml`

为 `solaris` 标记区域群集指定配置文件模板。从系统信息库安装后，该模板会将系统配置信息应用到区域群集的所有节点。如果未指定 `config_profile.xml`，您必须在每个节点上从全局区域运行 `zlogin -C zoneclustername` 命令来手动配置每个区域群集节点。所有配置文件都必须具有 `.xml` 扩展名。

`-c` 选项可替换配置文件模板中区域群集节点的主机名。引导区域群集节点后，该配置文件会应用到该区域群集节点。

`-d absolute_root_path`

`--dirpath dirpatch`

`-d` 选项与 `cluster` 子命令一起使用时，指定已安装的 Oracle Solaris 10 系统的根目录路径。该路径应能在要安装区域群集的群集的所有物理节点上访问。

`-d`

`--dvd-directory dvd-directory`

指定 DVD 映像目录。

`-d` 选项与 `install-cluster` 子命令一起使用时，指定支持 `solaris10` 标记区域的 Oracle Solaris Cluster 发行版的 DVD 映像目录。DVD 映像包括核心软件包和其他群集软件组件（例如区域群集中支持的代理和 Geographic Edition 软件）。必须能够从运行该命令的节点的全局区域中访问该 DVD 目录。

`-d`

`--dry_run`

当 `-d` 选项与 `apply` 子命令一起使用时，将在模拟运行模式下运行重新配置。模拟运行模式不会更改配置，它会让正在运行的区域保持现状。使用模拟运行模式可以检查真实重新配置将执行的操作。

`-f{commandfile | zonepath}`

`--file-argument {commandfile | zonepath}`

与 `configure` 子命令一起使用时，`-f` 选项指定命令文件参数。例如，`clzonecluster configure-fcommandfile`。与 `move` 子命令一起使用时，`-f` 选项指定 `zonepath`。

`-F`

在 `delete` 和 `uninstall` 操作期间，可以使用 `-F` 选项。`-F` 选项强制抑制 `Are you sure you want to do this operation [y/n]?` 问题。

`-m method`

`--method method`

使用 `-m` 选项克隆区域群集。克隆的唯一有效方法为 `copy` 命令。在运行 `clone` 子命令之前，必须停止源区域群集。

`-M manifest.xml`

`--manifest manifest.xml`

使用 `-M` 选项可为 `solaris` 标记区域群集的所有节点指定一个清单。该清单为区域群集安装指定 Oracle Solaris 软件包信息和 Oracle Solaris Cluster 软件包。

`-n nodename[...]`

`--nodelist nodename[...]`

指定子命令的节点列表。

例如，`clzonecluster boot-n phys-schost-1、phys-schost-2 zoneclustername`。

-o

--offline

将区域群集引导或重新引导为 offline-running 模式。

如果区域群集节点不是区域群集的成员但 Oracle Solaris 区域状态为正在运行，则会出现 offline-running 模式。区域群集与物理群集共享引导模式（群集或非群集模式），因此，群集处于脱机模式与处于非群集模式不同。

要将区域群集引导为 offline-running 模式，请键入以下内容。

```
clzonecluster boot [-n phys-schost-1,...] [-o] zoneclustername
```

要将区域群集重新引导为 offline-running 模式，请键入以下内容。

```
clzonecluster reboot [-n phys-schost-1,...] [-o] zoneclustername
```

要将 offline-running 区域重新引导为 online-running 模式，请运行不带 -o 选项的 clzonecluster reboot 命令。

-p name=value

--property=name=value

--property name=value

-p 选项与 install-cluster 子命令和 set 子命令一起使用。有关将 -p 与 install-cluster 子命令一起使用的用法信息，请参见关于 -p patchdir=patchdir[,patchlistfile=patchlistfile] 的描述。

-p 选项与 set 子命令一起使用来指定属性值。允许使用多个 -p name= value 实例。

将该选项与 set 子命令一起使用可修改以下属性：

resource_security

指定 RGM 资源执行程序所用的安全策略。允许的 resource_security 值为 SECURE、WARN、OVERRIDE 或 COMPATIBILITY。

Start 和 Validate 之类的资源方法始终以 root 用户身份运行。如果方法可执行文件不归 root 所有或具有组或全局写权限，将存在不安全状况。在这种情况下，如果 resource_security 属性设置为 SECURE，在运行时资源方法执行将失败并返回错误。如果 resource_security 设置为任何其他值，将允许资源方法执行，但显示警告消息。为了最大限度地保证安全性，请将 resource_security 设置为 SECURE。

resource_security 设置还会修改声明了 application_user 资源属性的资源类型的行为。有关更多信息，请参见 [r_properties\(5\) \[1103\]](#) 手册页的 application_user 部分。

-p patchdir=patchdir[,patchlistfile=patchlistfile]

--patch-specification=patchdir=patchdir[,patchlistfile=patchlistfile]

--patch-specification patchdir=patchdir[,patchlistfile=patchlistfile]

-p 选项指定的 patchdir 和 patchlistfile 属性仅与 install-cluster 子命令一起使用。如果在安装核心软件包后安装修补程序，则必须将区域群集引导为 offline-running 状态才能应用修补程序。

允许使用多个 `-p name= value` 实例。

`patchdir`

指定要应用到 `solaris10` 标记区域的 Oracle Solaris Cluster 修补程序所在的目录。`patchdir` 目录是必需项，且必须能够从区域群集的所有节点上的 `solaris10` 标记区域内部进行访问。

`patchlistfile`

指定 `patchlistfile`。`patchlistfile` 指定包含要安装的修补程序列表的文件。如果未指定可选的 `patchlistfile`，该命令将尝试将安装 `patchdir` 目录内的所有修补程序。您也可以在 `patchdir` 目录中创建一个 `patchlistfile` 列出修补程序 ID（每行一个），从而指出要安装的修补程序。

`-s`

`--software-component {all | software-component[,...]}`

指定要从 DVD 映像安装的软件组件。

这些组件是核心软件包以外的附加组件，可以为在区域群集中支持的数据服务或 Geographic Edition 软件。使用 `-s all` 时，无法指定其他任何组件，将安装所有数据服务和 Geographic Edition 软件。对于数据服务代理，组件名称即为代理名称。对于 Geographic Edition 软件，指定为 `-s geo`。如果未指定 `-s` 选项，则只安装群集框架软件。

`-v`

`--verbose`

在标准输出 (stdout) 中显示详细信息。

`-V`

`--version`

显示该命令的版本。

如果是与其他选项、子命令或操作数一起指定该选项，将忽略所有这些内容。仅显示该命令的版本。不会发生任何其他处理。

`[-x cert|ca-cert|key=file] ...`

如果具有 HTTPS 统一归档文件位置，请指定 SSL 证书、CA 证书和密钥文件。可以多次指定 `-x` 选项。

`-Z target-zoneclustername`

`--zonecluster target-zoneclustername`

您要克隆的区域群集名称。

使用源区域群集名称来克隆。在使用该子命令之前，必须停止源区域群集。

`-z zone`

如果统一归档文件包含多个区域，请指定安装源的区域名称。

资源和属性

clzonecluster 命令支持区域群集的多个资源和属性。

必须使用 clzonecluster 命令配置 clzonecluster 命令支持的任何资源和属性。有关配置 clzonecluster 命令不支持的资源或属性的更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

下面的子部分（“资源”和“属性”）介绍了 clzonecluster 命令支持的那些资源和属性。

资源

以下列出在资源范围中支持的资源类型以及可以在何处找到更多信息：

admin

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。该资源可同时在群集范围和节点范围内使用。该资源会向下传递到单个 Oracle Solaris 区域级别。同时在群集和节点范围中指定该资源时，节点范围资源信息会向下传递到区域群集特定节点的 Oracle Solaris 区域。

admin 资源的 auths 属性可设置为以下值之一：

clone	等同于 solaris.zone.clonefrom
login	等同于 solaris.zone.login
manage	等同于 solaris.zone.manage

capped-cpu

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。该资源可同时在群集范围和节点范围内使用。该资源会向下传递到单个 Oracle Solaris 区域级别。同时在群集和节点范围中指定该资源时，节点范围资源信息会向下传递到区域群集特定节点的 Oracle Solaris 区域。

capped-memory

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。该资源可在群集范围和节点范围内使用。该资源会向下传递到单个 Oracle Solaris 区域级别。同时在群集和节点范围中指定该资源时，节点范围资源信息会向下传递到区域群集特定节点的 Oracle Solaris 区域。

dataset

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。该资源可在群集范围或节点范围内使用。不能同时在群集和节点范围中指定数据集。

在群集范围中该资源用于为高可用性 ZFS 文件系统导出要在区域群集中使用的 ZFS 数据集。导出的数据集由 Oracle Solaris Cluster 软件管理，在群集范围中指定时，不会传递到单个 Oracle Solaris 区域级别。数据集无法在区域群集之间共享。

在节点范围中该资源用于将本地 ZFS 数据集导出到特定区域群集节点。在节点范围中指定时，导出的数据集不由 Oracle Solaris Cluster 软件管理，而会传递到单个 Oracle Solaris 区域级别。

dedicated-cpu

有关更多信息，请参见 [Unresolved link to "zonecfg1M"](#) 手册页。可以在每个节点上使用专用于区域群集的固定数量的 CPU。

该资源可在群集范围和节点范围内使用。该资源会向下传递到单个 Oracle Solaris 区域级别。同时在群集和节点范围中指定该资源时，节点范围资源信息会向下传递到区域群集特定节点的 Oracle Solaris 区域。

device

有关更多信息，请参见 [Unresolved link to "zonecfg1M"](#) 手册页。该资源会向下传递到单个 Oracle Solaris 区域级别并可在群集范围或节点范围中指定。在节点范围中该资源用于添加特定于区域群集节点的设备。可以将设备只添加到一个区域群集。无法同时在群集范围和节点范围中添加相同设备。

fs

有关更多信息，请参见 [Unresolved link to "zonecfg1M"](#) 手册页。可以在群集范围或节点范围中指定该资源。不能同时在群集和节点范围中指定 fs 资源。

在群集范围中该资源通常用于导出要在区域群集中使用的文件系统。导出的文件系统由 Oracle Solaris Cluster 软件管理，而不会向下传递到单个 Oracle Solaris 区域级别，但 `cluster-control` 属性设置为 `false` 的 `lofs` 文件系统除外。有关 `cluster-control` 属性的更多信息，请参见本手册页中“资源”部分中关于 fs 的描述。

在节点范围中该资源用于将本地文件系统导出到特定区域群集节点。在节点范围中指定时，导出的文件系统不由 Oracle Solaris Cluster 软件管理，而会传递到单个 Oracle Solaris 区域级别。

可以通过直接挂载或回送挂载将文件系统导出到区域群集。直接挂载使文件系统在区域群集中可访问，方法是在某个位置挂载指定的文件系统，该位置位于区域根目录或在其路径中具有区域根目录的某个子目录下方。直接挂载意味着文件系统只属于该区域群集。区域群集在 Oracle Solaris Trusted Extensions 上运行时，必须对以读写权限挂载的文件使用直接挂载。区域群集支持对 UFS、QFS 独立文件系统、QFS 共享文件系统和 ZFS（导出为数据集）使用直接挂载。

回送挂载是一种机制，使已在某个位置挂载的文件系统显示为在其他位置挂载。可以通过使用每个区域群集一个回送挂载将单个文件系统导出到多个区域群集。这样便可以在多个区域群集之间共享单个文件系统。在多个区域群集之间共享文件系统前，管理员必须考虑安全隐患。不管如何挂载真实的文件系统，回送挂载可以将访问权限限制为只读。

fs: cluster-control

`cluster-control` 属性仅适用于在群集范围中指定的回送挂载。`cluster-control` 属性的默认值为 `true`。

当属性值为 `true` 时，Oracle Solaris Cluster 管理该文件系统，但不会将文件系统信息传递给 `zonecfg` 命令。区域引导后，Oracle Solaris Cluster 根据需要在区域群集节点中挂载和取消挂载文件系统。

Oracle Solaris Cluster 可以为 QFS 共享文件系统、UFS、QFS 独立文件系统和 UFS 上的 PxFS 管理回送挂载。

当属性值为 `false` 时，Oracle Solaris Cluster 不管理文件系统。群集软件将该文件系统信息和所有关联的信息传递到 `zonecfg` 命令，该命令在每台计算机上创建区域群集区域。在这种情况下，当区域引导时，Oracle Solaris 软件将挂载文件系统。管理员可以将该选项用于 UFS 文件系统。

管理员可以在群集范围中指定回送挂载。为回送挂载配置 `cluster-control` 属性值 `false` 对通用本地目录（如包含可执行文件的目录）的只读挂载很有用。此信息会传递到执行实际挂载的 `zonecfg` 命令。使用值为 `true` 的 `cluster-control` 属性配置回送挂载对于将全局文件系统 (PxFS) 或 QFS 共享文件系统提供给受群集控制的区域群集很有用。

QFS 共享文件系统、UFS、QFS 独立文件系统和 ZFS 最多可配置在一个区域群集中。

net

有关 net 资源的更多信息，请参见 [Unresolved link to "zonecfg1M"](#) 手册页。

由 Oracle Solaris Cluster 管理的任何网络资源（如逻辑主机或共享地址）是在群集范围中指定的。由应用程序管理的任何网络资源（如 Oracle RAC VIP）是在群集范围中指定的。这些网络资源不会传递到单个 Oracle Solaris 区域级别。

管理员可以指定网络接口卡 (Network Interface Card, NIC) 以用于指定的 IP 地址。系统自动选择满足下列两个要求的 NIC：

- NIC 已连接到同一个子网。
- 已为该区域群集配置 NIC。

node

节点资源执行下列两个目标：

- 标识范围级别。在节点范围中指定的任何资源只属于该特定节点。
- 标识区域群集的节点。管理员可以标识将运行区域的计算机，方法是在该计算机上标识全局群集全局区域。为每个区域群集节点指定 IP 地址和 NIC 是可选的。管理员还可以指定用于标识网络信息以访问该节点的信息。

注 - 如果管理员没有为每个区域群集节点配置 IP 地址，将出现以下两种情况：

1. 该特定区域群集将无法配置要在区域群集中使用的 NAS 设备。群集在与 NAS 设备通信时将使用区域群集节点的 IP 地址，所以缺失 IP 地址会导致群集不支持隔离 NAS 设备。
 2. 群集软件将激活所有 NIC 上的所有逻辑主机 IP 地址。
-

privnet

该资源可在节点范围中使用。该资源指定可用作区域群集的专用适配器的数据链路设备。该资源在指定给区域群集之前必须在全局区域中可用。配置独占 IP 区域群集时，默认将 `enable_priv_net` 属性设置为 `true`，以在区域群集节点间启用专用网络通信。

```
add node
add
privnet
set
physical=vnic1
end
add
privnet
set
physical=vnic5
end
end
```

资源属性 `privnet` 的顺序用于构成区域群集节点间的路径。第一个节点中指定的第一个 `privnet` 适配器将尝试用第二个节点中指定的第一 `privnet` 路径构成路径。在添加和删除操作后将保留 `privnet` 资源的顺序。

注 - `privnet` 资源无法在多个独占 IP 区域中共享。必须将其指定给一个特定的独占 IP 区域。

rctl

有关更多信息，请参见 [Unresolved link to "zonecfg1M"](#) 手册页。该资源可同时在群集范围和节点范围内使用。该资源会向下传递到单个 Oracle Solaris 区域级别。同时在群集和节点范围中指定该资源时，节点范围资源信息会向下传递到区域群集特定节点的 Oracle Solaris 区域。

sysid

请参见 [Unresolved link to "sysidcfg4"](#) 手册页。该资源为 `solaris10` 区域群集的所有区域指定系统标识参数。

属性

每个资源类型具有一个或多个属性。支持群集的以下属性：

(群集)

`admin`

有关更多信息，请参见 [Unresolved link to "zonecfg1M"](#) 手册页。

(群集)

`allowed-address`

指定可在适配器上激活的 IP 地址。仅允许特定的 IP 地址。该可选属性用于节点范围的网络资源。例如：

```
set allowed-address=1.2.2.3/24
```

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

(群集)

attr

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。区域群集将使用设置为 cluster 的属性 name、设置为 boolean 的属性 type 和设置为 true 的属性 value。使用 create 选项配置区域群集时，将为这些属性设置默认值。这些属性是用于区域群集配置的强制性属性，不能更改。

(群集)

autoboot

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

(群集)

bootargs

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

(群集)

brand

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。仅支持 solaris、solaris10 和 labeled 标记类型。

(群集)

cpu-shares

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

(群集)

device

[Unresolved link to "zonecfg1M"](#)。

(群集)

enable_priv_net

设置为 true 时，在区域群集的节点之间将启用 Oracle Solaris Cluster 专用网络通信。

- 如果 ip-type 设置为 shared，区域群集节点之间的通信使用全局群集的专用网络。
- 如果 ip-type 设置为 exclusive，区域群集节点之间的通信使用指定的 privnet 资源。如果未指定这些资源，将会通过在全局群集的专用网络上创建虚拟网络接口 (vnic) 来自动生成这些资源。

区域群集节点的 Oracle Solaris Cluster 专用主机名和 IP 地址是由系统自动生成的。如果值设置为 false，将禁用专用网络。默认值为 true。

(群集)

`ip-type`

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。仅支持 shared 和 exclusive 值。

(群集)

`limitpriv`

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

(群集)

`max-lwps`

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

(群集)

`max-msg-ids`

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

(群集)

`max-sem-ids`

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

(群集)

`max-shm-ids`

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

(群集)

`monitor_quantum`

定义量程值的毫秒数。

(群集)

`monitor_timeout`

指定监视器超时的毫秒数。

(群集)

`max-shm-memory`

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

(群集)

`pool`

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

(群集)

zonename

区域群集的名称，也是区域群集中每个区域的名称。

(群集)

zonepath

区域群集中每个区域的区域路径。

admin

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

capped-cpu

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

capped-memory

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

dataset

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

dedicated-cpu

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

device

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

fs

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

inherit pkg-dir

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

net

有关更多信息，请参见 [Unresolved link to "zonecfg1M" 手册页](#)。

node

包括 physical-host、hostname 和 net。

- physical-host – 该属性指定将托管区域群集节点的全局群集节点。
- hostname – 该属性指定由物理主机属性指定的全局群集节点上区域群集节点的公共主机名。

-
- net – 该资源指定区域群集节点用于公共网络通信的网络地址和物理接口名称，该区域群集节点在由物理主机指定的全局群集节点上。

rctl

请参见 [Unresolved link to "zonecfg1M"](#)。

sysid

使用 `/usr/bin/sysconfig configure` 命令。

请参见 [Unresolved link to "sysidcfg4"](#)。包括

`root_password`、`name_service`、`security_policy`、`system_locale`、`timezone`、`terminal` 和 `nfs4_domain`。管理员稍后可以按照正常 Oracle Solaris 过程手动更改任何 `sysidcfg config` 值（一次更改一个节点）。

- `root_password` – 该属性为区域群集的所有节点指定公用 root 密码的加密值。请勿指定明文密码。必须使用 `/etc/shadow` 中的加密密码字符串。这是一个必需属性。
- `name_service` – 该可选属性指定要在区域群集中使用的命名服务。但是，全局区域的 `/etc/sysidcfg` 文件中的设置可能已过时。为确保该属性具有正确的设置，请使用 `clzonecluster` 命令手动输入该值。
- `security_policy` – 该值在默认情况下设置为 `none`。
- `system_locale` – 默认情况下该值从 `clzonecluster` 命令的环境中获得。
- `timezone` – 该属性指定要在区域群集中使用的时区。默认情况下该值从 `clzonecluster` 命令的环境中获得。
- `terminal` – 该值在默认情况下设置为 `xterm`。
- `nfs4_domain` – 该值在默认情况下设置为 `dynamic`。

[Intro\(1CL\) \[17\]](#) 手册页列出了该命令集中所有命令的一整套退出状态代码。

如果命令对于所有指定的操作数均成功执行，则返回零 (`CL_NOERR`)。如果某个操作数出错，命令会继续处理操作数列表中的下一个操作数。返回的退出代码始终反映最先发生的错误。

此命令返回下列退出状态代码：

0 `CL_NOERR`

没有错误

您发出的命令已成功完成。

1 `CL_ENOMEM`

交换空间不足

某个群集节点耗尽了交换内存或其他操作系统资源。

3 `CL_EINVAL`

参数无效

您键入的命令不正确，或者使用 `-i` 选项提供的群集配置信息语法不正确。

6 CL_EACCESS

权限被拒绝

指定的对象不可访问。要发出命令，可能需要拥有超级用户或 RBAC 访问权限。有关更多信息，请参见 [Unresolved link to " su1M"](#) 和 [Unresolved link to " rbac5"](#) 手册页。

18 CL_EINTERNAL

遇到内部错误

内部错误表示软件缺陷或其他缺陷。

35 CL_EIO

I/O 错误

发生了物理输入/输出错误。

36 CL_ENOENT

没有这样的对象

由于以下某个原因，找不到您所指定的对象：

- 该对象不存在。
- 您尝试使用 `-o` 选项创建的配置文件路径中的某个目录不存在。
- 您尝试使用 `-i` 选项访问的配置文件包含错误。

38 CL_EBUSY

对象忙

您尝试将一根电缆从上一个群集互连路径移到一个活动群集节点。或者，您尝试将一个节点从尚未删除引用的群集配置中移除。

39 CL_EEXIST

对象已存在

您所指定的设备、设备组、群集互连组件、节点、群集、资源、资源类型、资源组或专用字符串已存在。

41 CL_ETYPE

未知类型

您使用 `-t` 或 `-p` 选项指定的类型不存在。

在所有示例中，`zoneclustername` 为 `sczone`。第一个全局群集节点为 `phys-schost-1`，第二个节点为 `phys-schost-2`。第一个区域群集节点为 `zc-host-1`，第二个节点为 `zc-host-2`。

例 265 创建新的区域群集

以下示例说明了如何创建双节点 solaris10 标记区域群集。zpool "tank" 委派给区域用作高可用性 ZFS 文件系统。内存限制用于限制可用于区域群集的内存量。除 root 用户密码之外，将使用默认系统标识值。

```
phys-schost-1# clzonecluster configure sczone
sczone: No such zone cluster configured
Use 'create' to begin configuring a new zone cluster.
clzc:sczone> create -b
clzc:sczone> set zonepath=/zones/timuzc
clzc:sczone> set brand=solaris10
clzc:sczone> set autoboot=true
clzc:sczone> set bootargs="-m verbose"
clzc:sczone> set limitpriv="default,proc_priocntl,proc_clock_highres"

clzc:sczone> set enable_priv_net=true
clzc:sczone> set ip-type=shared
clzc:sczone> add dataset
clzc:sczone:dataset> set name=tank
clzc:sczone:dataset> end
clzc:sczone> add capped-memory
clzc:sczone:capped-memory> set physical=3G
clzc:sczone:capped-memory> end
clzc:sczone> add rctl
clzc:sczone:rctl> set name=zone.max-swap
clzc:sczone:rctl> add value (priv=privileged,limit=4294967296,action=deny)

clzc:sczone:rctl> end
clzc:sczone> add rctl
clzc:sczone:rctl> set name=zone.max-locked-memory
clzc:sczone:rctl> add value (priv=privileged,limit=3221225472,action=deny)

clzc:sczone:rctl> end
clzc:sczone> add attr
clzc:sczone:attr> set name=cluster
clzc:sczone:attr> set type=boolean
clzc:sczone:attr> set value=true
clzc:sczone:attr> end
clzc:sczone> add node
clzc:sczone:node> set physical-host=ptimu1
clzc:sczone:node> set hostname=zc-host-1
clzc:sczone:node> add net
clzc:sczone:node:net> set address=vztimu1a
clzc:sczone:node:net> set physical=sc_ipmp0
clzc:sczone:node:net> end
clzc:sczone:node> end
clzc:sczone> add node
clzc:sczone:node> set physical-host=ptimu2
clzc:sczone:node> set hostname=zc-host-2
clzc:sczone:node> add net
clzc:sczone:node:net> set address=vztimu2a
```

```

clzc:sczone:node:net> set physical=sc_ipmp0
clzc:sczone:node:net> end
clzc:sczone:node> end
clzc:sczone> add fs
clzc:sczone:fs> set dir=/opt/local
clzc:sczone:fs> set special=/usr/local
clzc:sczone:fs> set type=lofs
clzc:sczone:fs> add options [ro,nodevices]
clzc:sczone:fs> set cluster-control=false
clzc:sczone:fs> end
clzc:sczone> add sysid
clzc:sczone> set root_password=ZiitH.NOLOrRg
clzc:sczone> set name_service="NIS{domain_name=mycompany.com name_server=
    ns101c-90(10.100.10.10)}"
clzc:sczone> set nfs4_domain=dynamic
clzc:sczone> set security_policy=NONE
clzc:sczone> set system_locale=C
clzc:sczone> set terminal=xterms
clzc:sczone> set timezone=US/Pacific
clzc:sczone> end

```

如果使用 `create` 子命令（而不是上面所示的 `create -b` 子命令），将使用默认模板，该模板已设置 `attr` 属性。

区域群集现已配置。下列命令安装然后从全局群集节点引导区域群集：

```

phys-schost-1# clzonecluster install -a absolute_path_to_archive install sczone

phys-schost-1# clzonecluster boot sczone

```

例 266 从统一归档文件创建区域群集

以下示例说明如何从统一归档文件创建和安装区域群集。统一归档文件可以从全局区域、非全局区域或区域群集节点创建。支持从克隆归档文件和恢复归档文件这两种统一归档文件配置和安装区域群集。如果统一归档文件是从非群集区域创建的，则必须设置以下属性：`enable_priv_net=true`。您还可以根据需要更改任何区域属性。

```

phys-schost-1# clzonecluster configure sczone
sczone: No such zone cluster configured
Use 'create' to begin configuring a new zone cluster.
clzc:sczone> create -a absolute_path_to_archive -z archived_zone_1
clzc:sczone> set zonepath=/zones/sczone

clzc:sczone> set enable_priv_net=true
clzc:sczone> set ip-type=shared

clzc:sczone> add attr
clzc:sczone:attr> set name=cluster
clzc:sczone:attr> set type=boolean
clzc:sczone:attr> set value=true
clzc:sczone:attr> end

```

```

clzc:sczone> add node
clzc:sczone:node> set physical-host=psoft1
clzc:sczone:node> set hostname=zc-host-1
clzc:sczone:node> add net
clzc:sczone:node:net> set address=vzsoft1a
clzc:sczone:node:net> set physical=sc_ipmp0
clzc:sczone:node:net> end
clzc:sczone:node> end
clzc:sczone> add node
clzc:sczone:node> set physical-host=psoft2
clzc:sczone:node> set hostname=zc-host-2
clzc:sczone:node> add net
clzc:sczone:node:net> set address=vzsoft2a
clzc:sczone:node:net> set physical=sc_ipmp0
clzc:sczone:node:net> end
clzc:sczone:node> end

```

区域群集现已配置。以下命令从全局群集节点上的统一归档文件安装区域群集：

```
phys-schost-1# clzonecluster install -a absolute_path_to_archive -z archived-zone sczone
```

区域群集现已安装。以下命令将引导区域群集：

```
phys-schost-1# clzonecluster boot sczone
```

例 267 修改现有的区域群集

以下示例显示如何修改在示例 1 中创建的区域群集的配置。额外的公用 IP 地址将添加到 phys-schost-2 上的区域群集节点中。

UFS 文件系统将导出到区域群集，以用作高度可用的文件系统。假设 UFS 文件系统是在 Oracle Solaris Volume Manager 元设备上创建的。

```

phys-schost-1# clzonecluster configure sczone
clzc:sczone> add device
clzc:sczone:device> set match=/dev/md/1/dsk/d100
clzc:sczone:device> end
clzc:sczone> add device
clzc:sczone:device> set match=/dev/md/oraset/dsk/d100
clzc:sczone:device> end
clzc:sczone> select node physical-host=phys-schost-2
clzc:sczone:node> add net
clzc:sczone:node:net> set address=192.168.0.3/24
clzc:sczone:node:net> set physical=bge0
clzc:sczone:node:net> end
clzc:sczone:node> end
clzc:sczone> add fs
clzc:sczone:fs> set dir=/qfs/ora_home
clzc:sczone:fs> set special=oracle_home
clzc:sczone:fs> set type=samfs

```

```
clzc:sczone:fs> end
clzc:sczone> exit
```

例 268 使用现有区域群集作为模板来创建新的区域群集

以下示例显示如何使用在示例 1 中创建的 `sczone1` 区域群集作为模板来创建名为 `sczone` 的区域群集。新区域群集的配置将与原始区域群集相同。需要修改新区域群集的某些属性来避免冲突。如果管理员删除资源类型而不指定特定资源，系统将删除该类型的所有资源。例如，`remove net` 将导致删除所有网络资源。

```
phys-schost-1# clzonecluster configure sczone1
sczone1: No such zone cluster configured
Use 'create' to begin configuring a new zone cluster.

clzc:sczone1> create -t sczone
clzc:sczone1> set zonepath=/zones/sczone1

clzc:sczone1> select node physical-host=phys-schost-1
clzc:sczone1:node> set hostname=zc-host-3
clzc:sczone1:node> select net address=zc-host-1
clzc:sczone1:node:net> set address=zc-host-3
clzc:sczone1:node:net> end
clzc:sczone1:node> end
clzc:sczone1> select node physical-host=phys-schost-2
clzc:sczone1:node> set hostname=zc-host-4
clzc:sczone1:node> select net address=zc-host-2
clzc:sczone1:node:net> set address=zc-host-4
clzc:sczone1:node:net> end
clzc:sczone1:node> remove net address=192.168.0.3/24
clzc:sczone1:node> end
clzc:sczone1> remove dataset name=tank/home
clzc:sczone1> remove net
clzc:sczone1> remove device
clzc:sczone1> remove fs dir=/qfs/ora_home
clzc:sczone1> exit
```

支持以下操作数：

`zoneclustername` 区域群集的名称。可以指定新区域群集的名称。所有子命令均支持 `zoneclustername` 操作数。

`+` 群集中的所有节点。只有一部分子命令支持 `+` 操作数。

[Intro\(1CL\) \[17\]](#) 手册页列出了该命令集中所有命令的一整套退出状态代码。

如果命令对于所有指定的操作数均成功执行，则返回零 (`CL_NOERR`)。如果某个操作数出错，命令会继续处理操作数列表中的下一个操作数。返回的退出代码始终反映最先发生的错误。

此命令返回下列退出状态代码：

0 CL_NOERR

没有错误。
您发出的命令已成功完成。

1 CL_ENOMEM

交换空间不足。
某个群集节点耗尽了交换内存或其他操作系统资源。

3 CL_EINVAL

参数无效。
您键入的命令不正确，或者使用 -i 选项提供的群集配置信息语法不正确。

18 CL_EINTERNAL

遇到内部错误。

36 CL_ENOENT

没有这样的对象
由于以下某个原因，找不到您所指定的对象：

- 该对象不存在。
- 您尝试使用 -o 选项创建的配置文件路径中的某个目录不存在。
- 您尝试使用 -i 选项访问的配置文件包含错误。

37 CL_EOP

不允许操作
您尝试对不受支持的配置执行操作，或者您所执行的操作不受支持。

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

[clnode\(1CL\) \[149\]](#)、[cluster\(1CL\) \[461\]](#)、[Intro\(1CL\) \[17\]](#)、[scinstall\(1M\) \[697\]](#)、[Unresolved link to "zoneadm1M"](#)、[Unresolved link to "zonecfg1M"](#)、[Unresolved link to "clconfiguration\(5CL\)"](#)

超级用户可以运行此命令的所有格式。

所有用户都可以运行带有 -? (帮助) 或 -v (版本) 选项的此命令。

要运行带有子命令的 `clzonecluster` 命令，超级用户以外的用户需要具有 RBAC 授权。请参见下表。

子命令	RBAC 授权
<code>boot</code>	<code>solaris.cluster.admin</code>
<code>check</code>	<code>solaris.cluster.read</code>
<code>clone</code>	<code>solaris.cluster.admin</code>
<code>configure</code>	<code>solaris.cluster.admin</code>
<code>delete</code>	<code>solaris.cluster.admin</code>
<code>export</code>	<code>solaris.cluster.admin</code>
<code>halt</code>	<code>solaris.cluster.admin</code>
<code>install</code>	<code>solaris.cluster.admin</code>
<code>list</code>	<code>solaris.cluster.read</code>
<code>monitor</code>	<code>solaris.cluster.modify</code>
<code>move</code>	<code>solaris.cluster.admin</code>
<code>ready</code>	<code>solaris.cluster.admin</code>
<code>reboot</code>	<code>solaris.cluster.admin</code>
<code>show</code>	<code>solaris.cluster.read</code>
<code>status</code>	<code>solaris.cluster.read</code>
<code>uninstall</code>	<code>solaris.cluster.admin</code>
<code>unmonitor</code>	<code>solaris.cluster.modify</code>
<code>verify</code>	<code>solaris.cluster.admin</code>

OSC4 1ha

名称

rt_callbacks — 用于管理作为 Oracle Solaris Cluster 资源的服务的回调接口

```
method-path -R resource -T type -G group [-Z zonename]
```

```
validate-path [-c | -u] -R resource -T type -G group [-r prop=  
val] [-x prop=val] [-g prop=val] [-Z zonename] [-X prop{nodeid}=  
val]
```

Oracle Solaris Cluster 资源类型的回调接口定义资源组管理器 (Resource Group Manager, RGM) 用于控制作为群集资源的服务的接口。资源类型的实现器提供充当回调方法的程序或脚本。

method-path 在资源类型注册文件中声明的程序的路径。该程序在 `clresourcetype` 命令中已注册为资源类型的以下回调方法之一：START、STOP、INIT、FINI、BOOT、PRENET_START、POSTNET_STOP、MONITOR_START、或 UPDATE。请参见 [clresourcetype\(1CL\) \[277\]](#) 和 [rt_reg\(4\) \[1067\]](#) 手册页。

validate-path 在资源类型注册文件中已声明为资源类型的 VALIDATE 方法的程序的路径。该程序已在 `clresourcetype` 命令中注册。

回调方法是传递的指定选项，预计用于采取特定操作以控制群集上的服务操作。

资源类型开发者在资源类型注册文件中声明回调方法程序的路径。群集管理员使用 `clresourcetype` 命令在群集配置中注册资源类型。随后群集管理员可以使用该注册的资源类型创建资源。这些资源是在 RGM 管理的资源组中配置的。

RGM 通过自动调用 RGM 所管理的资源组中的资源的回调方法来响应事件。回调方法预计用于对由资源表示的服务采取特定操作。这些操作的示例包括在某个群集节点上停止和启动服务。

回调方法返回的退出状态代码告知 RGM 回调方法是成功还是失败。如果方法失败，RGM 会采取操作，或者报告资源状态为故障。因此，群集管理员可以记下该故障并采取相应操作。

支持以下选项：

-c

指定在群集管理员创建资源时调用该方法，以验证所有资源和资源组属性的初始设置。

RGM 可指定 -c 选项或 -u 选项，但从不会同时指定这两个选项。

当群集管理员创建资源且调用 VALIDATE 方法时，所有系统定义的属性、扩展属性和资源组属性均将传递给 VALIDATE 方法。当群集管理员更新资源且调用 VALIDATE 方法时，仅将要更新的属性传递给 VALIDATE 方法。

-g prop =val

指定要传递给 VALIDATE 方法的资源组属性的值。

prop 资源组属性的名称。

val 群集管理员创建或更新资源时要传递给该方法的值。

-G group

指定在其中配置资源的资源组的名称。

-r prop =val

指定要传递给 VALIDATE 方法的系统定义的资源属性的值。

prop 系统定义的资源属性的名称。

val 群集管理员创建或更新资源时要传递给该方法的值。

-R resource

指定为其调用该方法的资源的名称。

-T type

指定资源的资源类型名称。

-u

指定在群集管理员更新现有资源或现有资源组的属性时调用该方法。

RGM 可指定 -c 选项或 -u 选项，但从不会同时指定这两个选项。

当群集管理员创建资源且调用 VALIDATE 方法时，所有系统定义的属性、扩展属性和资源组属性均将传递给 VALIDATE 方法。当群集管理员更新资源且调用 VALIDATE 方法时，仅将要更新的属性传递给 VALIDATE 方法。

-x prop =val

为本地节点指定资源扩展属性的值。

prop 资源扩展属性的名称。扩展属性是由资源类型实现定义的。此扩展属性是在资源类型注册文件的参数表中声明的。

val 群集管理员创建或更新资源时要传递给该方法的值。

-x prop{ nodeid}=val

为指定的节点指定每节点资源扩展属性的值。

<i>prop</i>	资源扩展属性的名称。扩展属性是由资源类型实现定义的。此扩展属性在资源类型注册文件的参数表中声明为每节点属性。
<i>node</i>	整数节点 ID。它用于指定在其上设置每节点属性值的节点。
<i>val</i>	群集管理员创建或更新资源时要传递给该方法的值。

-z zonename

指定在其中配置要运行的资源组的非全局区域的名称。

如果 `Global_zone` 资源类型属性设置为 `TRUE`，即使包含该资源的资源组在非全局区域中运行，这些方法也将在全局区域中执行。该选项提供在其中配置要运行的资源组的非全局区域的名称。

只要满足以下任一条件，就不会传递 `-z` 选项。

- `Global_zone` 属性设置为 `FALSE`。
- 资源组配置为在全局区域中运行。

回调方法是由调用它们的 RGM 机制定义的。这些方法预计用于对群集资源执行操作。这些方法预计还用于返回退出状态，后者将报告方法是成功还是失败。以下部分将介绍每个回调方法。

BOOT

如果节点在引导或重新引导时加入或重新加入群集，就会调用该方法。该方法会在 `Init_nodes` 资源类型属性指定的节点上调用。与 `INIT` 类似，该方法用于初始化在包含该资源的资源组联机之后加入群集的节点上的资源。对于位于受管资源组中的资源，会调用该方法；但对于位于不受管资源组中的资源，不会调用该方法。

FINI

当包含该资源的资源组取消 RGM 管理时，会调用该方法。该方法会在 `Init_nodes` 资源类型属性指定的节点上调用。该方法会取消配置资源并清除 `INIT` 方法或 `BOOT` 方法设置的任何持久性设置。

INIT

当包含该资源的资源组置于 RGM 管理之下时，会调用该方法。该方法会在 `Init_nodes` 资源类型属性指定的节点上调用。该方法将初始化资源。

MONITOR_CHECK

在将包含该资源的资源组重新定位到新的节点之前，会调用该方法。在故障监视器执行 `scha_control` 命令或 `scha_control()` 函数的 `GIVEOVER` 选项时，会调用该方法。请参见 [scha_control\(1HA\) \[579\]](#) 和 [scha_control\(3HA\) \[939\]](#) 手册页。

该方法会在作为资源组的潜在新主节点的任意节点上调用。`MONITOR_CHECK` 方法用于评估节点的运行状况是否足以运行资源。`MONITOR_CHECK` 方法必须以这样一种方式实现，即它不与同时运行的其他方法相冲突。

如果 MONITOR_CHECK 方法失败，它将禁止将资源组重定位到调用了回调的节点。

MONITOR_START

资源启动后，会在启动该资源的同一节点上调用该方法。该方法会启动资源的监视器。

MONITOR_START 失败将导致 RGM 将资源状态设置为 MONITOR_FAILED。

MONITOR_STOP

资源停止之前，会在运行该资源的同一节点上调用该方法。该方法会停止资源的监视器。在群集管理员禁用监视时，也会调用该方法。

RGM 在 MONITOR_STOP 方法失败时所采取的操作取决于资源的 Failover_mode 属性设置。如果 Failover_mode 设置为 HARD，RGM 会尝试通过重新引导节点来强制停止资源。否则，RGM 将资源的状态设置为 STOP_FAILED。

POSTNET_STOP

作为 STOP 方法的辅助方法，该方法旨在执行在将相关网络地址配置为关闭之后需要执行的关闭操作。该方法会在已调用 STOP 方法的节点上调用。在资源组中的网络地址已配置为关闭且资源的 STOP 方法已调用之后，会调用该方法。但是，该方法会在网络地址处于非活动状态之前被调用。POSTNET_STOP 方法在资源的 STOP 方法之后以及依赖于该资源的任意资源的 POSTNET_STOP 方法之后调用。

RGM 在 POSTNET_STOP 方法失败时所采取的操作取决于资源的 Failover_mode 属性设置。如果 Failover_mode 设置为 HARD，RGM 会尝试通过中止节点来强制停止资源。否则，RGM 将资源的状态设置为 STOP_FAILED。

PRENET_START

作为 START 方法的辅助方法，该方法旨在执行在将相关网络地址配置为打开之前需要执行的启动操作。该方法会在要调用 START 方法的节点上调用。同一资源组中的网络地址已处于活动状态之后会调用该方法。但是，该方法在地址配置为打开之前以及资源的 START 方法调用之前调用。PRENET_START 方法在资源的 START 方法之前以及依赖于该资源的任意资源的 PRENET_START 方法之前调用。

RGM 在 PRENET_START 方法失败时所采取的操作取决于资源的 Failover_mode 属性设置。如果 Failover_mode 设置为 SOFT 或 HARD，RGM 会尝试将包含该资源的资源组重定位到其他节点。否则，RGM 将资源的状态设置为 START_FAILED。

START

当包含该资源的资源组在某个群集节点上联机时，将在该节点调用此方法。群集管理员可以使用 clresourcegroup 命令切换打开关闭状态。START 方法激活节点上的资源。

RGM 在 START 方法失败时所采取的操作取决于资源的 Failover_mode 属性设置。如果 Failover_mode 设置为 SOFT 或 HARD，RGM 会尝试将资源的组重定位到其他节点。否则，RGM 将资源的状态设置为 START_FAILED。

STOP

当包含该资源的资源组在某个群集节点上脱机时，将在该节点调用此方法。群集管理员可以使用 `clresourcegroup` 命令切换打开关闭状态。如果资源处于活动状态，此方法将停用该资源。

RGM 在 STOP 方法失败时所采取的操作取决于资源的 `Failover_mode` 属性设置。如果 `Failover_mode` 设置为 `HARD`，RGM 会尝试通过重新引导节点来强制停止资源。否则，RGM 将资源的状态设置为 `STOP_FAILED`。

UPDATE

调用该方法以通知运行的资源其属性已更改。在 RGM 成功设置资源或其资源组的属性之后，会调用 `UPDATE` 方法。该方法会在资源处于联机状态的节点上调用。该方法可调用 `scha_resource_get` 和 `scha_resourcegroup_get` 命令读取可影响活动资源的属性值并相应地调整运行的资源。

VALIDATE

在创建资源或者更新资源或包含该资源的资源组时，会调用此方法。`VALIDATE` 会在由资源类型的 `Init_nodes` 属性指定的一组群集节点上调用。

在应用资源的创建或更新之前，会调用 `VALIDATE` 方法。如果该方法在某个节点上失败且生成了失败退出状态代码，则会取消创建或更新。

当群集管理员创建资源且调用 `VALIDATE` 方法时，所有系统定义的属性、扩展属性和资源组属性均将传递给 `VALIDATE` 方法。当群集管理员更新资源且调用 `VALIDATE` 方法时，仅将要更新的属性传递给 `VALIDATE` 方法。您可以使用 `scha_resource_get` 和 `scha_resourcegroup_get` 命令检索未更新的资源属性。

在 `VALIDATE` 命令行中用两个不同的 `-r` 选项传递资源依赖性属性。一个 `-r` 选项仅列出资源名称，表示在本地节点上适用的依赖性。另一个 `-r` 选项包括带有限定符（如 `{LOCAL_NODE}`）的整个依赖性列表。属性名称

`Resource_dependencies`、`Resource_dependencies_offline_restart`、`Resource_dependencies_restart` 和 `Resource_dependencies_weak` 提供本地节点的不带限定符的依赖性名称。对应的属性名称 `Resource_dependencies_Q`、`Resource_dependencies_Q_offline_restart`、`Resource_dependencies_Q_restart` 和 `Resource_dependencies_Q_weak` 提供带有限定符的相同依赖性列表。

例如，可设置以下内容：

```
# clresource set -p Resource_dependencies=r1@node1,r2@node2,r3{local_node},r4
```

在节点 1 上，以下参数将传递给 `VALIDATE` 方法：

```
... -r Resource_dependencies=r1,r3,r4
```

```
-r Resource_dependencies_Q=r1@node1,r2@node2,r3{local_node},r4
```

```
...
```

在节点 2 上，`Resource_dependencies` 属性的值将为 `r2,r3,r4`，而 `Resource_dependencies_Q` 属性的值在所有节点上均相同。类似地，属性名称 `Resource_dependencies_Q_weak`、`Resource_dependencies_Q_restart`

和 `Resource_dependencies_Q_offline_restart` 分别对应于依赖性属性 `Resource_dependencies_weak`、`Resource_dependencies_restart` 和 `Resource_dependencies_offline_restart`。

如果不显式设置 `Network_resources_used` 属性，它的值将从四个 `Resource_dependencies` 属性中派生，其中包含在这四个属性中出现的所有网络地址资源。每个节点上的 `Network_resources_used` 属性派生值反映了存在的任何每节点依赖性，各节点的值可能会有所不同。

当您实现 `VALIDATE` 方法时，您向 `stdout` 或 `stderr` 写入的任意消息均将传回至用户命令。在解释验证失败的原因或为用户提供与资源相关的说明时，此操作很有用。

Oracle Solaris Cluster 资源管理回调方法是由 RGM 使用超级用户权限执行的。实现方法的程序预计要使用相应执行权限进行安装，为安全起见，这些程序应不可写。

为执行回调方法而设置的环境变量如下所示：

```
HOME=/
PATH=/usr/bin:/usr/cluster/bin
LD_LIBRARY_PATH=/usr/cluster/lib
```

SIGNALS

如果回调方法的调用操作超出其超时周期，该进程将首先发送一个 `SIGTERM` 信号。如果 `SIGTERM` 信号无法在十秒内停止方法执行，进程将发送一个 `SIGKILL` 信号。

返回以下退出状态代码。

0 命令已成功完成。

非零值 发生错误。

失败退出状态的特定值不影响 RGM 对故障执行的操作。但是，在方法失败时，退出状态会记录在群集日志中。资源类型实现可能会定义不同的非零退出代码以通过群集日志向群集管理员传达相关错误信息。

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口	Evolving (发展中)
稳定性	

[clresourcegroup\(1CL\) \[253\]](#)、[clresourcetype\(1CL\) \[277\]](#)、[scha_cmds\(1HA\) \[573\]](#)、[scha_control\(1HA\) \[573\]](#)、[Unresolved link to "signal3C"](#)、[Unresolved link to "](#)

[stdio3C](#)、[scha_calls\(3HA\) \[897\]](#)、[scha_control\(3HA\) \[939\]](#)、[rt_reg\(4\) \[1067\]](#)、[Unresolved link to "attributes5"](#)

名称

scdsbuilder — 启动 GUI 版本的 Oracle Solaris Cluster Agent Builder

scdsbuilder

scdsbuilder 命令启动 GUI 版本的 Oracle Solaris Cluster Agent Builder。

使用 Agent Builder 之前，请验证以下要求：

- Java 运行时环境包含在 \$PATH 变量中。从版本 1.6 开始，Agent Builder 依赖于 Java Development Kit。如果 Java Development Kit 未包含在 \$PATH 变量中，Agent Builder 命令 (scdsbuilder) 会返回并显示错误消息。
- 已安装 Oracle Solaris OS 的开发者系统支持软件组。
- cc 编译器包含在 \$PATH 变量中。Agent Builder 使用 \$PATH 变量中第一个出现的 cc 来确定生成资源类型的 C 二进制代码所使用的编译器。如果 cc 未包含在 \$PATH 中，Agent Builder 将禁用生成 C 代码的选项。

注 - 可以将标准 cc 编译器以外的编译器用于 Agent Builder。要使用其他编译器，请在 \$PATH 中创建从 cc 到其他编译器的符号链接，如 gcc。或者，将 makefile（当前为 CC=cc）中的编译器指定信息更改为另一个编译器的完整路径。例如，在由 Agent Builder 生成的 makefile 中，将 CC=cc 更改为 CC=pathname/gcc。在这种情况下，无法直接运行 Agent Builder。必须改用 make 和 make pkg 命令来生成数据服务代码和软件包。

此命令返回下列退出状态代码：

0 命令已成功完成。

非零值 发生错误。

install-directory /rtconfig

包含来自上一个会话的信息。此信息简化了工具的退出和重新启动功能。

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api

属性类型	属性值
接口稳定性	Evolving (发展中)

[cc\(1\)](#)、[scdscreate\(1HA\) \[559\]](#)、[scdsconfig\(1HA\) \[555\]](#)、[Unresolved link to "attributes5"](#)

[Unresolved link to "Oracle Solaris Cluster Data Services Developer's Guide"](#)

名称

scdsconfig — 配置资源类型模板

```
scdsconfig -s start-command [-u start-method-timeout] [-e  
    validate-command] [-y validate-method-timeout] [-t stop-command]  
    [-v stop-method-timeout] [-m probe-command] [-n probe-timeout]  
    [-d working-directory]
```

scdsconfig 命令可配置使用 [scdscreate\(1HA\) \[559\]](#) 命令创建的资源类型模板。通过 scdsconfig 命令，可以为网络感知（客户机/服务器模型）和非网络感知（无客户机）应用程序配置基于 C、通用数据服务 (Generic Data Service, GDS) 或 Korn shell 的模板。

scdsconfig 命令可配置特定于应用程序的命令以启动、停止、验证和探测应用程序。还可以使用 scdsconfig 命令设置 start、stop、validate 和 probe 命令的超时值。scdsconfig 命令既支持网络感知（客户机/服务器模型）应用程序，也支持非网络感知（无客户机）应用程序。可以从运行 scdscreate 命令的同一目录中运行 scdsconfig 命令。还可以通过使用 -d 选项指定同一目录。scdsconfig 命令将用户指定的参数置于生成代码中的正确位置，以配置资源类型模板。如果生成源代码的类型为 C，则此命令还将编译该代码。scdsconfig 命令可将输出置于稍后可以安装的 Solaris 软件包中。此命令在 scdscreate 命令创建的 `$vendor-id $resource-type-name` 目录下的 pkg 子目录中创建软件包。

支持以下选项：

`-d working-directory`

指定运行 scdscreate 命令的目录。

如果从运行 scdscreate 命令的目录以外的其他目录中运行 scdsconfig 命令，必须指定此选项。

`-e validate-command`

指定要调用的命令的绝对路径以验证应用程序。如果未指定绝对路径，将不会验证应用程序。如果应用程序成功运行，`validate-command` 将返回退出状态 0。除 0 以外的退出状态指示应用程序无法正确执行。在这种情况下，根据应用程序的过去的失败历史记录，将出现两种结果的其中一种：

- 该资源类型的资源在同一节点上重新启动。
- 包含该资源的资源组已故障转移到其他正常运行的节点。

`-m probe-command`

指定命令以定期检查网络感知或非网络感知应用程序的运行状况。它必须是完整的命令行，可以直接传递到 shell 以探测应用程序。如果应用程序成功运行，`probe-command` 将返回退出状态 0。除 0 以外的退出状态指示应用程序无法正确执行。在这种情况下，根据应用程序的过去的失败历史记录，将出现两种结果的其中一种：

- 该资源类型的资源在同一节点上重新启动。
- 包含该资源的资源组故障转移到其他正常运行的节点。

-n *probe-timeout*

指定探测命令的超时值（秒）。超时必须将系统过载考虑在内以防止误报。默认值为 30 秒。

-s *start-command*

指定启动应用程序的命令。启动命令必须是完整的命令行，可以直接传递到 shell 以启动应用程序。可以包含命令行参数以指定主机名、端口号或启动应用程序所需的其他配置数据。要创建具有多个独立进程树的资源类型，请指定包含命令列表（一条命令占一行，用以启动不同进程树）的文本文件。

-t *stop-command*

为应用程序指定停止命令。停止命令必须是完整的命令行，可以直接传递到 shell 以停止应用程序。如果忽略此选项，则生成的代码将通过发出信号来停止应用程序。将 80% 的超时值分配给停止命令以停止应用程序。如果停止命令无法在此期限内停止应用程序，则会将 15% 的超时值分配给 SIGKILL 以停止应用程序。如果 SIGKILL 还无法停止应用程序，则停止方法将返回错误。

-u *start-method-timeout*

指定启动命令的超时值（秒）。超时必须将系统过载考虑在内以防止误报。默认值为 300 秒。

-v *stop-method-timeout*

指定停止命令的超时值（秒）。超时必须将系统过载考虑在内以防止误报。默认值为 300 秒。

-y *validate-method-timeout*

指定验证命令的超时值（秒）。超时必须将系统过载考虑在内以防止误报。默认值为 300 秒。

返回以下退出状态代码：

0 命令已成功完成。

非零值 发生错误。

working-directory /rtconfig

包含来自上一个会话的信息。简化工具的退出和重新启动功能。

有关下列属性的说明，请参见 [Unresolved link to " attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

Unresolved link to "[ksh1](#)"、[scdsbuilder\(1HA\) \[553\]](#)、[scdscreate\(1HA\) \[559\]](#)、Unresolved link to "[attributes5](#)"

名称

scdscreate — 创建 Oracle Solaris Cluster 资源类型模板

```
scdscreate -V vendor-id -T resource-type-name -a [-s] [-n  
RT-version] [-d working-directory] [-k | -g]
```

scdscreate 命令可创建用于使应用程序具有高可用性 (highly available, HA) 或可伸缩性的模板。通过此命令，您可以为网络感知（客户机/服务器模型）和非网络感知（无客户机）应用程序创建基于 C、通用数据服务 (Generic Data Service, GDS) 或 Korn shell 的模板。

可以使用两种根本不同的方法的其中一种创建模板：

GDS

scdscreate 可创建由三个组成一组的驱动脚本，这些脚本来自群集上预安装的单个资源类型 SUNW.gds。这些脚本命名为 start *RT-Name*、stop *RT-Name* 和 remove *RT-Name*，可启动、停止和删除该应用程序的实例。在此模型中，在群集上预安装的 SUNW.gds 资源类型的实现是不可改变的。

生成的源代码

scdscreate 可创建 Oracle Solaris Cluster 资源类型的模板，其实例在资源组管理器 (Resource Group Manager, RGM) 的控制下运行以使应用程序具有高可用性和可伸缩性。

任一模型均可以创建网络感知（客户机/服务器模型）和非网络感知（无客户机）应用程序的模板。

scdscreate 可在 *working-directory* 下创建格式为 *\$vendor-id\$resource-type-name* 的目录。此目录包含驱动脚本或为资源类型生成的源、二进制和软件包文件。scdscreate 还可创建配置文件 *rtconfig*，可以在其中存储资源类型的配置信息。scdscreate 仅允许每个目录创建一个资源类型。必须在不同的目录中创建不同的资源类型。

支持以下选项：

-a

此参数指定正在创建的资源类型不是网络感知的。scdscreate 在创建的模板中禁用所有网络相关代码。

-n *RT-version*

此可选参数指定生成的资源类型的版本。如果忽略此参数，且正在创建基于 C 或 Korn shell 的应用程序，则在默认情况下将使用文本字符串 1.0。如果忽略此参

数，且正在创建基于 GDS 的应用程序，则在默认情况下将使用 GDS 的 `RT_version` 字符串。`RT-version` 可区分同一基本资源类型的多个注册版或升级版。

不能将以下字符包括在 `RT-version` 中：空格、制表符、斜杠 (/)、反斜杠 (\)、星号 (*)、问号 (?)、逗号 (,)、分号 (;)、左方括号 (l) 或右方括号 (l)。

`-d working-directory`

在当前目录以外的其他目录中创建资源类型的模板。如果忽略此参数，`scdscreate` 将在当前目录中创建模板。

`-g`

此可选参数可生成基于 GDS 格式的模板以使应用程序具有高可用性和可伸缩性。

`-k`

此可选参数采用 Korn shell 命令语法而不是采用 C 生成源代码。请参见 [Unresolved link to "ksh1"](#)。

`-s`

此可选参数指示资源类型是可伸缩的。可以将可伸缩资源类型的实例（资源）配置到故障转移资源组中，进而关闭可伸缩性功能。如果忽略此参数，`scdscreate` 将创建故障转移资源类型的模板。

`-T resource-type-name`

资源类型名称和资源类型版本与供应商 ID 一起唯一标识正在创建的资源类型。

`-v vendor-id`

供应商 ID 通常为正在创建资源类型的供应商的股票代码或其他标识符。`scdscreate` 将后跟句点 (.) 的供应商 ID 附加到资源类型名称的开头。此语法可确保资源类型名称保持唯一（如果多个供应商使用相同的资源类型名称）。

0 命令已成功完成。

非零值 发生错误。

`working-directory/rtconfig`

包含来自上一个会话的信息并简化 `scdscreate` 的退出和重新启动功能。

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[Unresolved link to "ksh1", scdsbuilder\(1HA\) \[553\], scdsconfig\(1HA\) \[555\], Unresolved link to "attributes5", rt_properties\(5\) \[1143\]](#)

[Unresolved link to "Oracle Solaris Cluster Data Services Developer's Guide "](#)

名称

`scha_check_app_user` — 获取应用程序用户名并检查所有权和权限

```
scha_check_app_user [-R resource] [-U username] [-Z  
zoneclustername] cmd-path
```

`scha_check_app_user` 命令可获取供特定资源（受资源组管理器 (Resource Group Manager, RGM) 控制的资源）使用的配置应用程序用户名。它还可检查 `cmd-path` 指定的可执行文件的所有权和权限。此可执行文件通常是要由资源方法或监视器（使用 [Unresolved link to "su1M"](#) 之类的包装将用户 ID 设置为配置的用户）执行的应用程序。资源方法或监视器在执行应用程序之前调用 `scha_check_app_user`。根据 `scha_check_app_user` 的输出，方法或监视器可能会在检测到与安全有关的问题时返回错误或输出警告消息。

`scha_check_app_user` 命令将配置用户的名称写入到标准输出（文件描述符 1），将任何安全警告或错误消息写入到标准错误（文件描述符 2）。退出代码指示配置的安全策略是否允许执行命令。如果退出代码为 0，调用者可以尝试以应用程序用户身份执行命令。如果退出代码非零，调用者不应尝试以应用程序用户身份执行命令，而应返回错误。

调用 `scha_check_app_user` 的脚本可以使用该命令的输出来确定以下情况：

- 应以什么用户 ID 执行命令
- 是允许执行命令还是抛出错误
- 发现安全问题时向用户传递回什么错误或警告消息

`scha_check_app_user` 命令与 [r_properties\(5\) \[1103\]](#) 手册页中介绍的 `Resource_security` 和 `Application_user` 属性一起工作。

`scha_check_app_user` 命令的行为取决于 `Resource_security` 属性的设置。`Resource_security` 属性在全局群集和每个区域群集中可能具有不同的值。`scha_check_app_user` 使用的 `Resource_security` 值是该属性在执行命令的群集中的值。

应当在要执行应用程序的相同环境中调用 `scha_check_app_user` 命令。例如，如果应用程序在全局区域中执行，则 `scha_check_app_user` 也应当在全局区域中执行。

通常的使用案例为下面的一种：

- 资源及其资源组配置在全局群集中，`scha_check_app_user` 程序在全局群集中执行。
- 资源及其资源组配置在区域群集中，`scha_check_app_user` 程序在该区域群集的某个区域中执行。

在这两种使用案例中，均无需在命令中指定 `-z zoneclustername` 选项。

当应用程序要在全局区域中执行，但该应用程序与区域群集中的某个资源关联时，需要使用 `-z zoneclustername` 选项。通常，这并不是必需的，但对于将 `Global_zone` 属性设置为 `TRUE` 的资源类型，则可能是必需的。有关更多信息，请参见 [rt_properties\(5\) \[1143\]](#)。

有关使用 `-z` 以及其他命令选项的详细信息，请参见“选项”部分。

支持以下选项：

`-z zoneclustername`

指定资源配置在哪个群集中。仅当命令在全局区域中执行但需要访问区域群集中某个资源的 `Application_user` 属性时，才需要指定此选项。无法在某一区域群集中使用 `-z` 选项来访问其他区域群集。

如果省略 `-z` 选项，则认为资源存在于执行 `scha_check_app_user` 命令的群集（全局群集或区域群集）中。

如果 `scha_check_app_user` 命令在全局区域中执行，并且同时指定了 `-z` 和 `-R` 选项，则使用 `-R` 指定的 `resource` 位于 `-z` 指定的区域群集中而不在全局群集中。在这种情况下，代理开发者应当提醒最终用户注意以下情况：即使资源配置在区域群集中，`Application_user` 属性指定的用户名也需要在全局区域中有效。

如果 `scha_check_app_user` 命令在全局区域中执行，并且指定了 `-z` 选项，则 `cmd-path` 参数标识的是全局区域（而不是 `-z` 指定的区域）中的文件路径名。

`-u username`

如果指定了该选项，则采用此用户名作为应用程序用户名，而不考虑可执行文件所有者、`Application_user` 属性设置或 `Resource_security` 属性设置。当调用者有自己的应用程序用户名确定机制并且调用者只想检查可执行程序的所有权和权限时，可以使用 `-u` 选项。如果调用者的实际用户 ID 为非 `root` 用户，而 `-u` 选项指定了 `root` 用户，则会产生错误。

如果将 `-u` 选项与 `-z` 选项一起使用，则指定的 `username` 必须在执行命令的区域中有效，而不一定要在 `-z` 选项指定的 `zoneclustername` 中有效。

`-R resource`

与此命令执行关联的 RGM 资源的名称。如果未同时指定 `-u` 选项，则从此资源的 `Application_user` 属性获取应用程序用户名。如果该资源没有 `Application_user` 属性，或者未设置该属性，则应用程序用户名是可执行文件的所有者。

如果未指定 `-u` 而 `Resource_security` 设置为 `COMPATIBILITY`，则无论 `Application_user` 属性设置为何值，应用程序用户名都将设置为调用进程的实际用户 ID。如果未指定 `-u` 而 `Resource_security` 属性设置为 `OVERRIDE`，则无论 `Application_user` 属性设置为何值，应用程序用户名都将设置为可执行文件的所有者。

如果同时指定了 `-R` 选项与 `-z`，则资源的 `Application_user` 属性指定的用户名必须在执行命令的区域中有效，而不一定要在 `-z` 选项指定的 `zoneclustername` 中有效。

cmd-path

调用者要以应用程序用户身份执行的可执行文件的全路径名。如果指定了 `-z` 选项，将相对于执行命令的区域（而不是 `-z` 选项指定的 `zoneclustername`）来计算 *cmd-path*。

如果既没有指定 `-R` 也没有指定 `-u`，则应用程序用户名是可执行文件的所有者，除非 `Resource_security` 设置为 `COMPATIBILITY`，在这种情况下，应用程序用户名设置为调用进程的实际用户 ID。

如果计算的应用程序用户是 `root` 用户（超级用户），但调用者的实际用户 ID 为非 `root` 用户，则应用程序用户名将成为调用者的实际用户 ID。

例 269 在脚本中将 `scha_check_app_user` 与 `su` 一起使用

以下 `bash` 脚本在使用 `su(1M)` 执行名为 `mycommand` 的命令（该命令与名为 `myresource` 的 RGM 资源关联）之前调用 `scha_check_app_user`。

```
COMMANDPATH=/opt/mypkg/bin/mycommand
RESOURCENAME=myresource
TMPFILE=$(/usr/bin/mktemp)

# Here we are redirecting the error/warning messages into
# a temp file and will write them later.
# Instead, we could just let them flow out to stderr.
APPUSER=$(/usr/cluster/bin/scha_check_app_user \
    -R $RESOURCENAME $COMMANDPATH 2>$TMPFILE)
errcode=$?

if [[ $errcode -ne 0 ]]; then
    # Security checks failed -- do not execute the program
    printf "Security checks failed on program %s:\n" $COMMANDPATH
    # Output the error messages
    /usr/bin/cat $TMPFILE
    /usr/bin/rm $TMPFILE
    exit errcode
fi

# There may still be warning messages in TMPFILE.
# Write them for the user.
/usr/bin/cat $TMPFILE
/usr/bin/rm $TMPFILE

# Application user name is in $APPUSER.
# Execute mycommand with any necessary arguments.
#
# Note that the su command might still fail, for example, if
# this script lacks the necessary privilege to execute as
# the application user.
#
# Other command wrappers such as "su -" or "pfexec" could be used
# here instead of plain "su".
```

```
su $APPUSER $COMMANDPATH arg1 arg2
```

返回以下退出状态代码。[scha_calls\(3HA\) \[897\]](#) 中介绍了错误代码。

0 SCHA_ERR_NOERR

通过了安全检查，命令可以用应用程序用户身份执行。但是，在输出写入到 `stderr` 时，会指示获取或检查应用程序用户时发生了错误。应当将任何此类警告消息传递回用户。

3 SCHA_ERR_INVALID

使用了无效参数调用该命令。在这种情况下，不会将应用程序用户写入到 `stdout`。一条错误消息将写入到 `stderr`，其中具体列出了多个可能错误中的某一个。

6 SCHA_ERR_ACCESS

路径参数标识的文件不是可执行文件；或者 `-u` 选项指定了 `root` 用户，而调用者的实际用户 ID 为非 `root` 用户；或者 `Resource_security` 为 `SECURE`，但存在以下状况之一：

- 可执行文件是全局可写的。
- 应用程序用户是 `root`，可执行文件是组可写的。

`SCHA_ERR_ACCESS` 退出代码表示存在安全违规，调用者不应执行命令。

14 SCHA_ERR_RSRC

`rname` 参数标识的资源名称无效。在这种情况下，不会将应用程序用户写入到 `stdout`。一条错误消息将写入到 `stderr`。

16 SCHA_ERR_CHECKS

`Resource_security` 为 `SECURE`，而 `Application_user` 名称未映射到有效用户 ID。`SCHA_ERR_CHECKS` 退出代码表示存在安全违规，调用者不应执行命令。

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[Unresolved link to "su1M"](#)、[Unresolved link to "pfexec1"](#)、[cluster\(1CL\) \[461\]](#)、[scha_cmds\(1HA\) \[573\]](#)、[scha_calls\(3HA\) \[897\]](#)、[scha_strerror\(3HA\) \[10\]](#)、[link to "attributes5"](#)、[Unresolved link to "Oracle Solaris Cluster Data Services Developer's Guide"](#)

名称

`scha_cluster_get` — 访问群集信息

```
scha_cluster_get -o optag [-Z zoneclustername] [args]
```

`scha_cluster_get` 命令用于访问和生成群集相关信息。您可以访问有关群集、节点、主机名、资源组、资源类型和状态的信息。

该命令仅供在资源类型回调方法的 shell 脚本实现中使用。资源类型的这些回调方法代表群集的资源组管理器 (Resource Group Manager, RGM) 所控制的服务。此命令可与 `scha_resource_get` () 函数提供相同的信息。

此命令可在单独的行中以格式化字符串的形式将输出发送到标准输出 (stdout)，如 [scha_cmds\(1HA\) \[573\]](#) 手册页中所述。可以将输出存储在 shell 变量中。还可以使用 shell 命令 (例如 `awk` 命令) 解析输出。

支持以下选项：

`-o optag`

`optag` 参数指示要访问的信息。可能需要一个附加参数，用于指示要检索其信息的群集节点，具体取决于 `optag`。

注 - `optag` 选项 (例如 `NODENAME_LOCAL` 和 `NODENAME_NODEID`) 不区分大小写。指定 `optag` 选项时，您可以使用任意大小写字母组合。

支持以下 `optag` 值：

`ALL_LOADLIMITS`

在连续的行上生成全局群集或区域群集中定义的所有 `loadlimit` 名称。

`ALL_NODEIDS`

在连续的行上生成群集中所有节点的数字节点标识符。

`ALL_NODENAMES`

在连续的行上生成群集中所有节点的名称。

`ALL_PRIVATELINK_HOSTNAMES`

在连续的行上生成在群集互连上定位所有群集节点所需的主机名。

`ALL_PSTRINGS`

在连续的行上生成群集中所有专用字符串的名称 (而不是值)。有关专用字符串的更多信息，请参见 [clpstring\(1CL\) \[175\]](#) 手册页。

ALL_RESOURCEGROUPS

在连续的行上生成群集中要管理的所有资源组的名称。

ALL_RESOURCETYPES

在连续的行上生成群集中已注册的所有资源类型的名称。

ALL_ZONES

在连续的行上生成群集中所有节点上的所有区域（包括全局区域）的 `nodename:zonename` 字符串。

仅当发生以下情况时，此查询的输出中才会包括非全局区域：

- 自群集联机后，至少引导过一次非全局区域。
- 非全局区域已成功启动服务管理工具 (Service Management Facility, SMF) 服务 `/system/cluster/sc_ng_zones`。

不执行 SMF 服务 `/system/cluster/sc_ng_zones` 的非全局区域无法控制资源组，因此不包括在输出中。

ALL_ZONES_NODEID

在连续的行上生成已将其数字节点标识符指定为参数的群集节点上所有区域（包括全局区域）的 `nodename:zonename` 字符串。

仅当发生以下情况时，此查询的输出中才会包括非全局区域：

- 自群集联机后，至少引导过一次非全局区域。
- 非全局区域已成功启动服务管理工具 (Service Management Facility, SMF) 服务 `/system/cluster/sc_ng_zones`。

不执行 SMF 服务 `/system/cluster/sc_ng_zones` 的非全局区域无法控制资源组，因此不包括在输出中。

CLUSTERNAME

生成群集名称。

HARD_LOADLIMIT

在连续的行上生成成为全局群集或区域群集中所有节点设置的具有特定 `limitname` 的硬限制。这需要一个不带标志的附加 `string` 参数（为负载限制名称字符串）。

字符串数组输出的每个元素的格式均为 `"%s=%d"`，其中，左侧字符串是 `nodename` 或 `nodename:zonename`，右侧整数是该节点上指定的限制名称的硬负载限制值。如果没有指定硬限制，将显示值 `-1` 作为硬限制。

LOADLIMIT_PROPS

在连续的行上生成全局群集或区域群集中所有节点的硬限制和软限制（用 / 分隔）。这需要一个不带标志的附加 `string` 参数（为负载限制名称字符串）。

字符串数组输出的每个元素都是 "%s=%d/%d" 格式的字符串，其中，左侧的字符串是 *nodename* 或 *nodename:zonename*，第一个整数是软限制，第二个整数是硬限制。如果没有指定硬限制，将显示值 -1 作为硬限制。如果没有指定软限制，将显示值 0 作为软限制。

LOADLIMITS_NODE

在连续的行上生成为特定节点设置的负载限制（用 / 分隔）设置。这需要一个不带标志的附加 string 参数（为负载限制名称字符串）。

字符串数组输出的每个元素都是 "%s=%d/%d" 格式的字符串，其中，字符串是指定节点上定义的限制名称，第一个整数是软限制值，第二个整数是硬限制值。如果没有指定硬限制，将显示值 -1 作为硬限制。如果没有指定软限制，将显示值 0 作为软限制。

NODEID_LOCAL

生成执行命令的节点的数字节点标识符。

NODEID_NODENAME

生成名称所指示的节点的数字节点标识符。需要一个不带标志的附加参数（为群集节点的名称）。

NODENAME_LOCAL

生成执行命令的群集节点的名称。

NODENAME_NODEID

生成数字标识符所指示的群集节点的名称。需要一个不带标志的附加参数（为数字群集节点标识符）。

NODESTATE_LOCAL

根据执行命令的节点的状态生成 UP 或 DOWN。

NODESTATE_NODE

根据指定节点的状态生成 UP 或 DOWN。需要一个不带标志的附加参数（为群集节点的名称）。

PRIVATELINK_HOSTNAME_LOCAL

生成在整个群集互连上定位运行命令的节点所需的主机名。

PRIVATELINK_HOSTNAME_NODE

生成在群集互连上定位指定节点所需的主机名。需要一个不带标志的附加参数（为群集节点的名称）。

PSTRING

生成专用字符串的明文值。需要一个不带标志的附加参数（为专用字符串的名称）。超级用户以外的用户需要具有 `solaris.cluster.modify` 基于角色的访问

控制 (role-based access control, RBAC) 授权才能使用该查询标记。有关专用字符串的更多信息，请参见 [clpstring\(1CL\) \[175\]](#) 手册页。

RESOURCE_SECURITY

生成 `resource_security` 群集属性的当前设置。

RG_FAILOVER_LOG

为在群集上配置的资源组生成故障转移和手动切换事件日志。该选项后跟您要检索日志的天数。您最多可以检索七天的日志文件。如果未指定天数，将使用默认值一天。请参见 [例 272 “使用 `scha_cluster` 命令显示事件日志”](#)。

SOFT_LOADLIMIT

在连续的行上生成为群集中所有节点设置的具有特定 `limitname` 的软负载限制。这需要一个不带标志的附加 `string` 参数（为负载限制名称字符串）。

字符串数组输出的每个元素的格式均为 `"%s=%d"`，其中，左侧字符串是 `nodename` 或 `nodename:zonename`，右侧整数是该节点上指定的限制名称的软负载限制值。如果没有指定软限制，将显示值 `0` 作为软限制。

SYSLOG_FACILITY

生成 RGM 用于日志消息的 [Unresolved link to "syslog3C"](#) 设备号。该值为 `24`，对应于 `daemon` 设备。您可以在 [Unresolved link to "logger1"](#) 命令中使用该值作为设备级别，以在群集日志中记录消息。

`-z zoneclustername`

指定要对其执行操作的群集。当命令是在全局区域中执行但却需要在指定区域群集上进行操作时，此选项适用。无法在某区域群集内执行该命令来访问其他区域群集。

`zoneclustername` 指定在名为 `zoneclustername` 的区域群集中执行查询。

如果省略 `-z` 选项，则将在执行命令的群集中执行查询。

如果要在全局群集中查询每区域属性（例如，节点状态）的值，请勿使用 `-z` 选项，而应使用查询标记的每区域格式。例如，使用 `NODESTATE_NODE` 而不是 `NODESTATE`，并提供 `nodename:zonename` 格式的附加命令行参数。

例 270 在 shell 脚本中使用 `scha_cluster` 命令

以下 shell 脚本使用 `scha_cluster` 命令显示每个群集节点是处于运行状态还是停机状态：

```
#!/bin/sh
nodenames=`scha_cluster_get -O All_Nodenames`
for node in $nodenames
do
    state=`scha_cluster_get -O NodeState_Node $node`
```

```
printf "State of node: %s\n exit: %d\n value: %s\n" "$node" $? "$state"
done
```

例 271 使用 `scha_cluster` 命令显示节点的负载限制

以下命令显示为节点 `node1` 定义的所有负载限制。

```
# scha_cluster_get -O LOADLIMITS_NODE node1
factor1=50/100
factor2=0/4
```

例 272 使用 `scha_cluster` 命令显示事件日志

以下命令显示过去七天群集的故障转移和手动切换事件日志。输出以 `rg_name`、资源组故障转移或手动切换到的 `node_name` 和 `timestamp` 的格式列出。

```
# scha_cluster_get -O RG_FAILOVER_LOG 7
rg1,psnow4,Thu Oct 17 04:17:31 2013
rg2,psnow4,Thu Oct 17 04:17:31 2013
rg1,psnow4,Thu Oct 17 04:30:56 2013
rg2,psnow4,Thu Oct 17 04:30:56 2013
rg_fo_nfs,psnow4,Tue Oct 22 22:42:43 2013
rg_fo_nfs,psnow4,Tue Oct 22 22:46:08 2013
testrg-lh-1,psnow4,Tue Oct 22 22:47:43 2013
rg_test,psnow4,Tue Oct 22 22:51:50 2013
rg_test,psnow4,Tue Oct 22 22:51:55 2013
rg_test,psnow4,Tue Oct 22 22:52:01 2013
RG1,psnow4,Tue Oct 22 23:09:14 2013
RG1,psnow4,Tue Oct 22 23:15:39 2013
RG4,psnow4,Tue Oct 22 23:16:10 2013
RG3,psnow4,Tue Oct 22 23:16:16 2013
RG2,psnow4,Tue Oct 22 23:16:20 2013
RG1,psnow4,Tue Oct 22 23:16:26 2013
RG4,psnow4,Tue Oct 22 23:16:51 2013
RG2,psnow4,Tue Oct 22 23:16:51 2013
RG1,psnow4,Tue Oct 22 23:17:07 2013
RG3,psnow4,Tue Oct 22 23:17:10 2013
RG1,psnow4,Tue Oct 22 23:18:08 2013
RG2,psnow4,Tue Oct 22 23:18:08 2013
RG3,psnow4,Tue Oct 22 23:18:08 2013
```

返回以下退出状态代码：

0 成功完成。

非零值 发生错误。

[scha_calls\(3HA\) \[897\]](#) 中介绍了错误代码。

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Stable (稳定)

[Unresolved link to "awk1"](#)、[Unresolved link to "logger1"](#)、[Unresolved link to "sh1"](#)、[scha_cmds\(1HA\) \[573\]](#)、[Unresolved link to "syslog3C"](#)、[scha_calls\(3HA\) \[897\]](#)、[scha_cluster_get\(3HA\) \[909\]](#)、[Unresolved link to "attributes5"](#)、[rg_properties\(5\) \[1129\]](#)

名称

scha_cmds —

scha_cluster_get、scha_control、scha_resource_get、scha_resourcegroup_get、scha_resourcetype_get、的命令标准输出

```
scha-command -o optag...
```

Oracle Solaris Cluster

[scha_cluster_get\(1HA\) \[567\]](#)、[scha_control\(1HA\) \[579\]](#)、[scha_resource_get\(1HA\) \[585\]](#)、[scha_resourcegroup_get\(1HA\) \[591\]](#) 和 [scha_resource_setstatus\(1HA\) \[593\]](#) 命令是资源类型的回调方法的命令行实现。请参见 [rt_callbacks\(1HA\) \[545\]](#)。

资源类型代表群集的资源组管理器 (Resource Group Manager, RGM) 设备控制的服务。这些命令向 [scha_calls\(3HA\) \[897\]](#) C 函数的功能提供命令行界面。

get 命令访问群集配置信息。所有这些命令具有相同的通用界面。这些命令都使用 `-ooptag` 操作数。该操作数指示要访问的信息。这些命令都将结果作为格式化的字符串发送到标准输出 (stdout)。可能需要其他参数，具体取决于相应的命令和 `optag` 的值。有关不同 `optag` 结果的格式的信息，请参见“结果格式”部分。

注 - 所有 `scha` 命令的 `optag` 选项不区分大小写。指定 `optag` 选项时，您可以使用任意大小写字母组合。

[scha_control\(1HA\) \[579\]](#) 命令也使用 `-o optag` 选项，该选项指示控制操作，但不会将输出生成到标准输出。

[scha_resource_setstatus\(1HA\) \[593\]](#) 命令用于设置由 RGM 管理的资源的 STATUS 和 STATUS_MSG 属性。

结果格式

通过命令输出到标准输出的字符串的格式取决于 `-o` 选项中包含的 `optag` 指示的结果类型。下表中指定了每种类型的格式。 [Unresolved link to "formats5"](#) 中介绍了格式表示法。

结果类型	标准输出的格式
boolean	TRUE\n 或 FALSE\n
enum	%s\n, enum 值的字符串名称。
extension	%s\n, 扩展属性的类型特性，为以下值之一：STRING、INT、BOOLEAN、ENUM 或 STRINGARRAY。 根据类型信息，以每个类型的格式输出属性值，如下所示：STRING 作为 string、INT 作为 int、BOOLEAN 作为 boolean、ENUM 作为 enum、STRINGARRAY 作为 string_array。

结果类型	标准输出的格式
int	%d\n
status	%s\n%s\n, 第一个字符串是状态, 为以下 enum 值之一: DEGRADED、FAULTED、OFFLINE、ONLINE 或 UNKNOWN。 第二个字符串是状态消息。
string	%s\n
string_array	数组中的每个元素按格式 %s\n 进行输出。可以为 GLOBAL_RESOURCES_USED 和 INSTALLED_NODES 属性返回指示所有节点或资源的星号。
unsigned_int	%u\n
unsigned_int_array	数组中的每个元素按格式 %u\n 进行输出。

optag 结果类型

下表指定不同命令的有效 *optag* 值, 以及根据在前面的表中指定的格式输出的结果类型。

scha_cluster_get(1HA) [567] 的 <i>optag</i> 值	结果类型
ALL_NODEIDS	unsigned_int_array
ALL_NODENAMES	string_array
ALL_PRIVATELINK_HOSTNAMES	string_array
ALL_RESOURCEGROUPS	string_array
ALL_RESOURCETYPES	string_array
CLUSTERNAME	string
NODEID_LOCAL	unsigned_int
NODEID_NODENAME	unsigned_int
NODENAME_LOCAL	string
NODENAME_NODEID	string
NODESTATE_LOCAL	enum (UP、DOWN)
NODESTATE_NODE	enum (UP、DOWN)
PRIVATELINK_HOSTNAME_LOCAL	string
PRIVATELINK_HOSTNAME_NODE	string
SYSLOG_FACILITY	int

scha_control(1HA) [579] 的 <i>optag</i> 值
CHANGE_STATE_OFFLINE
CHANGE_STATE_ONLINE
CHECK_GIVEOVER
CHECK_RESTART

[scha_control\(1HA\) \[579\]](#) 的 *optag* 值

GIVEOVER
IGNORE_FAILED_START
RESOURCE_DISABLE
RESOURCE_IS_RESTARTED
RESOURCE_RESTART
RESTART

[scha_resource_get\(1HA\) \[585\]](#) 的 *optag* 值 结果类型

AFFINITY_TIMEOUT	int
ALL_EXTENSIONS	string_array
BOOT_TIMEOUT	int
CHEAP_PROBE_INTERVAL	int
CHEAP_PROBE_INTERVAL	int
EXTENSION	extension
EXTENSION_NODE	extension
FAILOVER_MODE	enum (NONE、HARD、SOFT、RESTART_ONLY、LOG_ONLY)
FINI_TIMEOUT	int
GROUP	string
INIT_TIMEOUT	int
LOAD_BALANCING_POLICY	string
LOAD_BALANCING_WEIGHTS	string_array
MONITORED_SWITCH	enum (DISABLED、ENABLED)
MONITORED_SWITCH_NODE	enum (DISABLED、ENABLED)
MONITOR_CHECK_TIMEOUT	int
MONITOR_START_TIMEOUT	int
MONITOR_STOP_TIMEOUT	int
NETWORK_RESOURCES_USED	string_array
NUM_RESOURCE_RESTARTS	int
NUM_RG_RESTARTS	int
ON_OFF_SWITCH	enum (DISABLED、ENABLED)
ON_OFF_SWITCH_NODE	enum (DISABLED、ENABLED)
PORT_LIST	string_array
POSTNET_STOP_TIMEOUT	int
PRENET_START_TIMEOUT	int
R_DESCRIPTION	string
RESOURCE_DEPENDENCIES	string_array
RESOURCE_DEPENDENCIES_OFFLINE_RESTART	string_array
RESOURCE_DEPENDENCIES_RESTART	string_array

scha_resource_get(1HA) [585] 的 <i>optag</i> 值	结果类型
RESOURCE_DEPENDENCIES_WEAK	string_array
RESOURCE_PROJECT_NAME	string
RESOURCE_STATE	enum (ONLINE、OFFLINE、START_FAILED、STOP_FAILED、MONITOR_FAILED、ONLINE_NOT_MONITORED、STARTING、STOPPING)
RESOURCE_STATE_NODE	enum (有关值，请参见 RESOURCE_STATE)
RETRY_COUNT	int
RETRY_INTERVAL	int
SCALABLE	boolean
START_TIMEOUT	int
STATUS	status
STATUS_NODE	status
STOP_TIMEOUT	int
THOROUGH_PROBE_INTERVAL	int
TYPE	string
TYPE_VERSION	string
UDP_AFFINITY	boolean
UPDATE_TIMEOUT	int
VALIDATE_TIMEOUT	int
WEAK_AFFINITY	boolean

scha_resource_get (1HA) 和 scha_resourcetype_get(1HA) 的 <i>optag</i> 值	结果类型
API_VERSION	int
BOOT	string
FAILOVER	boolean
FINI	string
GLOBAL_ZONE	boolean
INIT	string
INIT_NODES	enum (RG_PRIMARYES、RT_INSTALLED_NODES)
INSTALLED_NODES	string_array。返回一个星号 (*) 以指示所有节点。
IS_LOGICAL_HOSTNAME	boolean
IS_SHARED_ADDRESS	boolean
MONITOR_CHECK	string
MONITOR_START	string
MONITOR_STOP	string
PER_NODE	boolean
PKGLIST	string_array
POSTNET_STOP	string

<code>scha_resource_get</code> (1HA) 和 <code>scha_resourcetype_get</code> (1HA) 的 <i>optag</i> 值	结果类型
PRENET_START	string
PROXY	boolean
RT_BASEDIR	string
RT_DESCRIPTION	string
RT_SYSTEM	boolean
RT_VERSION	string
SINGLE_INSTANCE	boolean
START	string
STOP	string
UPDATE	string
VALIDATE	string

<code>scha_resourcegroup_get</code> (1HA) 的 <i>optag</i> 值	结果类型
AUTO_START_ON_NEW_CLUSTER	boolean
DESIRED_PRIMARYS	int
FAILBACK	boolean
GLOBAL_RESOURCES_USED	string_array (返回星号 (*) 以指示所有资源)
IMPLICIT_NETWORK_DEPENDENCIES	boolean
MAXIMUM_PRIMARYS	int
MODELIST	string_array
PATHPREFIX	string
PINGPONG_INTERVAL	int
RESOURCE_LIST	string_array
RG_AFFINITIES	string_array
RG_DEPENDENCIES	string_array
RG_DESCRIPTION	string
RG_IS_FROZEN	boolean
RG_MODE	enum (FAILOVER、SCALABLE)
RG_PROJECT_NAME	string
RG_SLM_CPU	decimal
RG_SLM_CPU_MIN	decimal
RG_SLM_PSET_TYPE	enum (DEFAULT、DEDICATED_STRONG、DEDICATED_WEAK)
RG_SLM_TYPE	enum (AUTOMATED、MANUAL)
RG_STATE	enum (UNMANAGED、ONLINE、OFFLINE、PENDING_ONLINE、PENDING_OFFLINE、ERROR_STOP_FAILED、ONLINE_FAULTED、PENDING_ONLINE_BLOCKED)
RG_STATE_NODE	enum (有关值，请参见 RG_STATE)

<code>scha_resourcegroup_get</code> (1HA) 的 <i>optag</i> 值	结果类型
RG_SYSTEM	boolean
SUSPEND_AUTOMATIC_RECOVERY	boolean

一组退出状态代码适用于所有 `scha` 命令。

退出状态代码是 `scha_calls(3HA)` [897] 中介绍的相应 C 函数的 `scha_err_t` 返回代码数字值。

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Stable (稳定)

[Unresolved link to "](#)

`awk1`"、[rt_callbacks\(1HA\)](#) [545]、[scha_cluster_get\(1HA\)](#) [567]、[scha_control\(1HA\)](#) [579]、[scha_reso](#)
[link to "attributes5"](#)、[Unresolved link to "](#)
[formats5](#)"、[r_properties\(5\)](#) [1103]、[rg_properties\(5\)](#) [1129]、[rt_properties\(5\)](#) [1143]

名称

scha_control — 请求资源和资源组控制

```
scha_control -o optag -G group -R resource [-Z zonename]
```

scha_control 命令可以请求重新启动或重定位受资源组管理器 (Resource Group Manager, RGM) 控制的资源或资源组。可以在资源监视器的 shell 脚本实现中使用该命令。此命令可提供与 [scha_control\(3HA\) \[939\]](#) 函数相同的功能。

该命令的退出代码指示是否拒绝了所请求的操作。如果接受了请求，则在资源组或资源已完成脱机并已重新联机之前，该命令不会返回。调用 [scha_control\(1HA\) \[579\]](#) 的故障监视器可能会由于资源或资源组脱机而停止。因此，故障监视器可能永远无法收到成功的请求的返回状态。

要使用该命令，您需要具有 solaris.cluster.resource.admin 基于角色的访问控制 (role-based access control, RBAC) 授权。请参见 [Unresolved link to "rbac5"](#)。

您还必须能够承担已向其分配 Oracle Solaris Cluster 命令权限配置文件的角色，才能使用该命令。经授权的用户可以通过 [Unresolved link to "pfsh1"](#)、[Unresolved link to "pfcsh1"](#) 或 [Unresolved link to "pfksh1"](#) 配置文件 shell 在命令行中发出授权的 Oracle Solaris Cluster 命令。配置文件 shell 是一种特殊的 shell，通过该 shell，可以访问分配给 Oracle Solaris Cluster 命令权限配置文件的 Oracle Solaris Cluster 特权命令。当您运行 [Unresolved link to "su1M"](#) 以承担某个角色时，即会启动配置文件 shell。也可以使用 [Unresolved link to "pfexec1"](#) 发出授权的 Oracle Solaris Cluster 命令。

支持以下选项：

-G group

需要重新启动或重定位的资源组的名称，或者需要重新启动或重定位的资源所在的资源组的名称。如果该资源组未在发出请求的节点上联机，则请求会被拒绝。

-o optag

请求 optag 选项。

注 - optag 选项 (例如 CHECK_GIVEOVER 和 CHECK_RESTART) 不区分大小写。指定 optag 选项时，您可以使用任意大小写字母组合。

支持以下 optag 值：

CHANGE_STATE_OFFLINE

请求使 -R 选项指定的代理资源在本地节点上脱机。代理资源是一种 Oracle Solaris Cluster 资源，它可以从另一群集 (例如 Oracle Clusterware) 导入某个

资源的状态。在 Oracle Solaris Cluster 软件的上下文中，这种状态变化反映了外部资源的状态变化。

当您使用该 *optag* 值更改某个代理资源的状态时，不会执行该代理资源的方法。

如果节点上的“被依赖”资源发生故障，并且该资源无法恢复，则监视器将使该资源在该节点上脱机。监视器将会通过调用 `scha_control` 命令并使用 `CHANGE_STATE_OFFLINE` *optag* 值来使该资源脱机。监视器还会通过触发重新启动操作，使被依赖资源的所有 `offline-restart` 依赖项脱机。当群集管理员解决故障并重新启用被依赖资源后，监视器还会使被依赖资源的 `offline-restart` 依赖项重新联机。

CHANGE_STATE_ONLINE

请求使 `-R` 选项指定的代理资源在本地节点上联机。代理资源是一种 Oracle Solaris Cluster 资源，它可以从另一群集（例如 Oracle Clusterware）导入某个资源的状态。在 Oracle Solaris Cluster 软件的上下文中，这种状态变化反映了外部资源的状态变化。

当您使用该 *optag* 值更改某个代理资源的状态时，不会执行该代理资源的方法。

CHECK_GIVEOVER

对 `-G` 选项指定的资源组执行与 `GIVEOVER` 相同的有效性检查，但不实际重定位该资源组。

CHECK_RESTART

对 `-G` 选项指定的资源组执行与 `RESTART` 相同的有效性检查，但不实际重新启动该资源组。

GIVEOVER

请求使 `-G` 选项指定的资源组在本地节点上脱机，然后在 `RGM` 选择的另一个节点上重新联机。请注意，如果该资源组当前已在两个或更多个节点上联机，并且没有其他可用节点可供该资源组在其上联机，则可使它在本地节点上脱机，且无需再让它在其他位置联机。请求可能会被拒绝，具体取决于各项检查的结果。例如，如果组是由于在 `PINGPONG_INTERVAL` 属性指定的时间间隔内某个节点上发出的 `GIVEOVER` 请求而脱机，则可能会拒绝将该节点用作主机。

如果群集管理员配置了一个或多个资源组的 `RG_Affinities` 属性，且您对一个资源组发出了 `scha_control GIVEOVER` 请求，那么，最终可能会重定位多个资源组。[rg_properties\(5\) \[1129\]](#) 中介绍了 `RG_Affinities` 属性。

通过故障监视器发出 `scha_control` 命令或调用 `scha_control()` 函数而使包含资源的资源组重定位到新的节点之前，将会调用 `MONITOR_CHECK` 方法。

您可以在作为资源组潜在新主机的任何节点上调用 `MONITOR_CHECK` 方法。`MONITOR_CHECK` 方法旨在评估某个节点的运行状况是否良好并足以运行某个资源。`MONITOR_CHECK` 方法必须以这样一种方式实现，即它不与同时运行的其他方法相冲突。

如果 `MONITOR_CHECK` 失败，则会禁止将资源组重定位到调用了回调的节点。

IGNORE_FAILED_START

请求在当前执行的 `Prenet_start` 或 `Start` 方法失败时资源组不进行故障转移，无论 `Failover_mode` 属性的设置如何。

换言之，该 `optag` 值将覆盖当 `Failover_Mode` 属性设置为 `SOFT` 或 `HARD` 的资源无法启动时通常会采取的恢复操作。通常，资源组会故障转移到另一个节点。而资源的行为却如同 `Failover_Mode` 设置为 `NONE` 一样。如果未发生其他错误，资源将进入 `START_FAILED` 状态，而资源组最终将进入 `ONLINE_FAULTED` 状态。

仅当通过 `Start` 或 `Prenet_start` 方法（该方法随后会退出并返回非零状态，或者超时）调用时，该 `optag` 值才有意义。该 `optag` 值仅对 `Start` 或 `Prenet_start` 方法的当前调用有效。在 `Start` 方法确定资源在其他节点上无法成功启动的情况下，应使用该 `optag` 值调用 `scha_control` 命令。如果由任何其他方法调用该 `optag` 值，将返回错误 `SCHA_ERR_INVALID`。该 `optag` 值可防止可能会发生的资源组“交替”故障转移。

RESOURCE_DISABLE

在调用 `scha_control` 命令的节点上禁用 `-R` 选项指定的资源。

如果节点上的“被依赖”资源发生故障，并且该资源无法恢复，则监视器将使该资源在该节点上脱机。监视器将会通过调用 `scha_control` 命令并使用 `RESOURCE_DISABLE` `optag` 值来使该资源脱机。监视器还会通过触发重新启动操作，使被依赖资源的所有 `offline-restart` 依赖项脱机。当群集管理员解决故障并重新启用被依赖资源后，监视器还会使被依赖资源的 `offline-restart` 依赖项重新联机。

RESOURCE_IS_RESTARTED

请求在本地节点上递增 `-R` 选项指定的资源的资源重新启动计数器，但不实际重新启动该资源。

未调用 `scha_control` 的 `RESOURCE_RESTART` 选项而直接重新启动资源（例如，使用 `pmfadm(1M)` [625] 的资源监视器可以使用该选项来通知 RGM 已重新启动该资源。`scha_resource_get(1HA)` [585] 的后续 `NUM_RESOURCE_RESTARTS` 查询会反映此递增情况。

如果资源的类型无法声明 `RETRY_INTERVAL` 标准属性，则不允许使用 `scha_control` 命令的 `RESOURCE_IS_RESTARTED` 选项。因此，`scha_control` 命令将会失败，并生成退出状态代码 13 (`SCHA_ERR_RT`)。

RESOURCE_RESTART

请求使 `-R` 选项指定的资源在本地节点上脱机，然后在该节点上再次联机，而无需停止该资源组中的任何其他资源。这将通过在本地节点上对该资源应用以下方法序列来使其停止后再启动：

MONITOR_STOP

```
STOP
START
MONITOR_START
```

如果资源类型没有声明 `STOP` 和 `START` 方法，则会改用 `POSTNET_STOP` 和 `PRENET_START` 来重新启动该资源：

```
MONITOR_STOP
POSTNET_STOP
PRENET_START
MONITOR_START
```

如果资源的类型没有声明 `MONITOR_STOP` 和 `MONITOR_START` 方法，则只会调用 `STOP` 和 `START` 方法或者 `POSTNET_STOP` 和 `PRENET_START` 方法来执行重新启动。

如果重新启动资源时某个方法调用失败，RGM 可能会设置一个错误状态、重定位资源组或重新引导节点，具体取决于对该资源的 `FAILOVER_MODE` 属性的设置。有关更多信息，请参见 [r_properties\(5\) \[1103\]](#) 中的 `FAILOVER_MODE` 属性。

使用该选项重新启动资源的资源监视器可以使用 [scha_resource_get\(1HA\) \[585\]](#) 的 `NUM_RESOURCE_RESTARTS` 查询来统计最近的重新启动尝试次数。

具有 `PRENET_START` 和/或 `POSTNET_STOP` 方法的资源类型应慎用 `RESOURCE_RESTART` 函数。只会对此类资源应用 `MONITOR_STOP`、`STOP`、`START` 和 `MONITOR_START` 方法。该资源所依赖的网络地址资源不会重新启动，将保持联机状态。

如果节点上的“被依赖”资源发生故障，并且该资源无法恢复，则监视器将使该资源在该节点上脱机。监视器将会通过调用 `scha_control` 命令并使用 `RESOURCE_RESTART optag` 值来使该资源脱机。监视器还会通过触发重新启动操作，使被依赖资源的所有 `offline-restart` 依赖项脱机。当群集管理员解决故障并重新启用被依赖资源后，监视器还会使被依赖资源的 `offline-restart` 依赖项重新联机。

RESTART

请求使 `-G` 选项指定的资源组脱机，然后再联机，而无需强制重定位到其他节点。如果该资源组中的某个资源无法重新启动，该请求最终可能会导致重定位该组。使用该选项重新启动资源组的资源监视器可以使用 [scha_resource_get\(1HA\) \[585\]](#) 的 `NUM_RG_RESTARTS` 查询来统计最近的重新启动尝试次数。

`CHECK_GIVEOVER` 和 `CHECK_RESTART optag` 值用于供资源监视器直接对资源采取操作（例如，终止并重新启动进程，或者重新引导节点），而不是通过调用 `scha_control` 命令来执行移交或重新启动。如果检查失败，监视器应休眠片刻，然后重新启动其探测器，而不是调用其重新启动或故障转移操作。有关更多信息，请参见 [scha_control\(3HA\) \[939\]](#)。

-R resource

资源组中某个资源的名称，该资源很可能是其监视器正在发出 [scha_control\(1HA\) \[579\]](#) 请求的资源。如果指定的资源不在资源组中，则请求会被拒绝。

所指定的资源的 `Failover_mode` 属性设置可能会抑制所请求的 `scha_control` 操作。如果 `Failover_mode` 为 `RESTART_ONLY`，则允许除 `scha_controlGIVEOVER` 和 `scha_controlCHECK_GIVEOVER` 以外的所有请求。`GIVEOVER` 和 `CHECK_GIVEOVER` 请求将返回 `SCHA_ERR_CHECKS` 退出代码，并且所请求的移交操作不会执行，而只生成 `syslog` 消息。

如果在资源上设置了 `Retry_count` 和 `Retry_interval` 属性，则资源重新启动次数限制为 `Retry_interval` 中的 `Retry_count` 尝试次数。如果 `Failover_mode` 为 `LOG_ONLY`，则任何 `scha_control` 移交、重新启动或禁用请求将返回 `SCHA_ERR_CHECKS` 退出代码，并且所请求的移交或重新启动操作不会执行，而只生成 `syslog` 消息。

-Z zonename

资源组配置为在其中运行的区域的名称。

如果 `Global_zone` 属性设置为 `TRUE`，则即使包含资源的资源组在非全局区域中运行，也会在全局区域中执行方法。该选项提供在其中配置要运行的资源组的非全局区域的名称。

请仅对 `Global_zone` 属性设置为 `TRUE` 的资源类型使用 `-Z` 选项。如果 `Global_zone` 属性设置为 `FALSE`，则不需要此选项。有关 `Global_zone` 属性的更多信息，请参见 [rt_properties\(5\) \[1143\]](#) 手册页。

返回以下退出状态代码：

- 0 命令已成功完成。
- 非零值 发生错误。
[scha_calls\(3HA\) \[897\]](#) 中介绍了故障错误代码。

有关下列属性的说明，请参见 [Unresolved link to " attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Stable (稳定)

[pmfadm\(1M\) \[625\]](#)、[rt_callbacks\(1HA\) \[545\]](#)、[scha_cmds\(1HA\) \[573\]](#)、[scha_resource_get\(1HA\) \[585\]](#)、[Unresolved link to " attributes5"](#)、[r_properties\(5\) \[1103\]](#)、[Unresolved link to " rbac5"](#)、[rg_properties\(5\) \[1129\]](#)、[rt_properties\(5\) \[1143\]](#)

名称

scha_resource_get — 访问资源信息

```
scha_resource_get [-O] -O optag -R resource [-G group] [-Z  
zoneclustername] [args]
```

scha_resource_get 命令可访问有关受资源组管理器 (Resource Group Manager, RGM) 控制的资源的信息。您可以使用该命令查询资源类型的属性 (如 [rt_properties\(5\) \[1143\]](#) 中所述) 以及资源的属性 (如 [r_properties\(5\) \[1103\]](#) 中所述)。

在代表受群集 RGM 控制的服务的资源类型回调方法的 shell 脚本实现中使用 scha_resource_get 命令。该命令提供的信息与 [scha_resource_get\(3HA\) \[963\]](#) C 函数提供的信息相同。

该命令将信息生成到 stdout，并在各个行中以格式化字符串表示，如 [scha_cmds\(1HA\) \[573\]](#) 中所述。可以将输出存储在 shell 变量中，并通过使用 shell 工具或 [Unresolved link to "awk1"](#) 进行解析，以供脚本进一步使用。

要使用该命令，您需要具有 solaris.cluster.resource.read 基于角色的访问控制 (role-based access control, RBAC) 授权。请参见 [Unresolved link to "rbac5"](#)。

您还必须能够承担已向其分配 Oracle Solaris Cluster 命令权限配置文件的角色，才能使用该命令。经授权的用户可以通过 [Unresolved link to "pfsh1"](#)、[Unresolved link to "pfcsh1"](#) 或 [Unresolved link to "pfksh1"](#) 配置文件 shell 在命令行中发出授权的 Oracle Solaris Cluster 命令。配置文件 shell 是一种特殊的 shell，通过该 shell，可以访问分配给 Oracle Solaris Cluster 命令权限配置文件的 Oracle Solaris Cluster 特权命令。当您运行 [Unresolved link to "su1M"](#) 以承担某个角色时，即会启动配置文件 shell。也可以使用 [Unresolved link to "pfexec1"](#) 发出授权的 Oracle Solaris Cluster 命令。

支持以下选项：

-G *group*

配置了资源的资源组的名称。尽管该参数是可选的，但如果包括该参数，命令将会更有效地运行。

-O *optag*

表示要访问的信息。根据您指定的 *optag* 值，可能需要包括一个附加值，以指示要检索其信息的群集节点。

注 - *optag* 值 (例如 AFFINITY_TIMEOUT 和 BOOT_TIMEOUT) 不区分大小写。指定 *optag* 值时，您可以使用大小字母的任意组合。

以下 *optag* 值将检索相应的资源属性。将生成资源的已命名属性的值。NUM_RG_RESTARTS、NUM_RESOURCE_RESTARTS、MONITORED_SWITCH、ON_OFF_SWITCH、RESOURCE_STATE 和 STATUS 属性在执行命令的节点上引用该值。有关与以下 *optag* 值相对应的资源属性的描述，请参见 [r_properties\(5\) \[1103\]](#) 手册页。请注意，以下列表中的某些 *optag* 值在列表后面而不是在 [r_properties\(5\) \[1103\]](#) 手册页中进行了介绍。

AFFINITY_TIMEOUT
ALL_EXTENSIONS
APPLICATION_USER
BOOT_TIMEOUT
CHEAP_PROBE_INTERVAL
EXTENSION
EXTENSION_NODE
FAILOVER_MODE
FINI_TIMEOUT
GLOBAL_ZONE_OVERRIDE
GROUP
INIT_TIMEOUT
LOAD_BALANCING_POLICY
LOAD_BALANCING_WEIGHT
SMONITORED_SWITCH
MONITORED_SWITCH_NODE
MONITOR_CHECK_TIMEOUT
MONITOR_START_TIMEOUT
MONITOR_STOP_TIMEOUT
NETWORK_RESOURCES_USED
NUM_RESOURCE_RESTARTS
NUM_RESOURCE_RESTARTS_ZONE
NUM_RG_RESTARTS
NUM_RG_RESTARTS_ZONE
ON_OFF_SWITCH
ON_OFF_SWITCH_NODE
PORT_LIST
POSTNET_STOP_TIMEOUT
PRE_EVICT
PRENET_START_TIMEOUT
RESOURCE_DEPENDENCIES
RESOURCE_DEPENDENCIES_OFFLINE_RESTART
RESOURCE_DEPENDENCIES_RESTART
RESOURCE_DEPENDENCIES_WEAK
RESOURCE_PROJECT_NAME
RESOURCE_STATE
RESOURCE_STATE_NODE
RETRY_COUNT
RETRY_INTERVAL
R_DESCRIPTION
SCALABLE
START_TIMEOUT
STATUS
STATUS_NODE
STOP_TIMEOUT
THOROUGH_PROBE_INTERVAL
TYPE

TYPE_VERSION
UDP_AFFINITY
UPDATE_TIMEOUT
VALIDATE_TIMEOUT
WEAK_AFFINITY

[r_properties\(5\) \[1103\]](#) 手册页中未介绍以下 *optag* 值。

ALL_EXTENSIONS

在连续的若干行中，生成资源的所有扩展属性的名称。

EXTENSION

在连续的若干行中，生成属性类型，其后跟有该属性的值。如果属性是每节点扩展属性，则返回的值是执行 `scha_resource_get` 的节点上的该属性的值。需要给出资源扩展属性名称的不带标志的参数。Shell 脚本可能需要放弃该类型以获取值，如示例中所示。

当用户请求未指定明确值的节点上的该属性的值时，将会返回在资源类型注册 (Resource Type Registration, RTR) 文件中声明的默认值。请参见 [rt_reg\(4\) \[1067\]](#) 手册页。

EXTENSION_NODE

在连续的若干行中，生成已命名节点的属性类型，其后跟有该属性的值。该值需要两个给出特定节点上资源的扩展名称的不带标志的参数，其顺序如下：

- 扩展属性名称
- 节点名称

Shell 脚本可能需要放弃该类型以获取值。

当用户请求未指定明确值的节点上的该属性的值时，将会返回在 RTR 文件中声明的默认值。请参见 [rt_reg\(4\) \[1067\]](#) 手册页。

GROUP

生成配置了资源的资源组的名称。

RESOURCE_DEPENDENCIES_NODE

生成指定节点的资源的 `RESOURCE_DEPENDENCIES` 属性值。需要给出节点名称的不带标志的参数。

RESOURCE_DEPENDENCIES_OFFLINE_RESTART_NODE

生成指定节点的资源的 `RESOURCE_DEPENDENCIES_OFFLINE_RESTART` 属性值。需要给出节点名称的不带标志的参数。

RESOURCE_DEPENDENCIES_RESTART_NODE

生成指定节点的资源的 `RESOURCE_DEPENDENCIES_RESTART` 属性值。需要给出节点名称的不带标志的参数。

RESOURCE_DEPENDENCIES_WEAK_NODE

生成指定节点的资源的 RESOURCE_DEPENDENCIES_WEAK 属性值。需要给出节点名称的不带标志的参数。

RESOURCE_STATE_NODE

生成指定节点的资源的 RESOURCE_STATE 属性值。需要给出节点名称的不带标志的参数。

STATUS_NODE

生成指定节点的资源的 STATUS 属性值。需要给出节点名称的不带标志的参数。
以下 *optag* 值将检索相应的资源类型属性。将生成资源类型的已命名属性的值。

注 - *optag* 值 (例如 API_VERSION 和 BOOT) 不区分大小写。指定 *optag* 值时, 您可以使用大小字母的任意组合。

有关资源类型属性的说明, 请参见 [rt_properties\(5\) \[1143\]](#)。

API_VERSION
BOOT
FAILOVER
FINI
GLOBAL_ZONE
INIT
INIT_NODES
INSTALLED_NODES
IS_LOGICAL_HOSTNAME
IS_SHARED_ADDRESS
MONITOR_CHECK
MONITOR_START
MONITOR_STOP
PKGLIST
POSTNET_STOP
PRENET_START
PROXY
RT_BASEDIR
RT_DESCRIPTION
RT_SYSTEM
RT_VERSION
SINGLE_INSTANCE
START
STOP
UPDATE
VALIDATE

如果该资源类型声明了 GLOBAL_ZONE_OVERRIDE 资源属性, 则 GLOBAL_ZONE *optag* 检索的值是 GLOBAL_ZONE_OVERRIDE 属性的当前值, 而不是 GLOBAL_ZONE 属性的值。有关更多信息, 请参见 [rt_properties\(5\) \[1143\]](#) 手册页中的 Global_zone 属性和 [r_properties\(5\) \[1103\]](#) 手册页中的 Global_zone_override 属性的描述。

-Q

在资源依赖性列表中包括所有指定限定符。 [r_properties\(5\) \[1103\]](#) 手册页中介绍了 {LOCAL_NODE}、 {ANY_NODE}、 @nodename 和 {FROM_RG_AFFINITIES} 限定符。如果省略了 -Q 选项，则资源依赖性列表的返回值只包含在本地节点上应用的依赖性的资源名称，没有任何声明的限定符。

-R *resource*

受 RGM 群集工具管理的资源的名称。

-z *zoneclustername*

指定包含资源组并想要对其执行操作的群集。当命令是在全局区域中执行但却需要在指定区域群集上进行操作时，此选项适用。无法在某区域群集内执行该命令来访问其他区域群集。

zoneclustername 指定在名为 *zoneclustername* 的区域群集中执行查询。

如果省略 -z 选项，则将在执行命令的群集中执行查询。

例 273 使用 `scha_resource_get` 命令的脚本样例

为以下脚本传递提供所需资源名称和资源组名称的 -R 和 -G 参数。接下来，`scha_resource_get` 命令访问资源的 `Retry_count` 属性和资源的 `enum` 类型的 `LogLevel` 扩展属性。

```
#!/bin/sh

while getopts R:G: opt
do
    case $opt in
        R)    resource="$OPTARG";;
        G)    group="$OPTARG";;
        esac
    done

    retry_count=`scha_resource_get -O Retry_count -R $resource \
-G $group`
    printf "retry count for resource %s is %d\n" $resource \
$retry_count

    LogLevel_info=`scha_resource_get -O Extension -R $resource \
-G $group LogLevel`

    # Get the enum value that follows the type information
    # of the extension property. Note that the preceding
    # assignment has already changed the newlines separating
    # the type and the value to spaces for parsing by awk.

    loglevel=`echo $LogLevel_info | awk '{print $2}'`
```

例 274 使用带有 -Q 选项以及不带有该选项的 `scha_resource_get` 命令查询资源依赖性

该示例显示如何使用 `clresource` 命令创建名为 `myres` 的资源，包含多个具有 `{LOCAL_NODE}` 范围限定符、`{ANY_NODE}` 范围限定符或不具有任何范围限定符的资源依赖性。然后，该示例显示如何使用 `scha_resource_get` 命令查询 `Resource_dependencies` 属性。如果没有 -Q 选项，则只返回资源名称。如果有 -Q 选项，还会返回已声明的范围限定符。

```
# clresource create -g mygrp -t myrestype \  
-p Resource_dependencies=myres2{LOCAL_NODE},myres3{ANY_NODE},myres4 \  
myres  
# scha_resource_get -O Resource_dependencies -R myres -G mygrp  
  
myres2  
myres3  
myres4  
# scha_resource_get -Q -O Resource_dependencies -R myres -G mygrp  
  
myres2{LOCAL_NODE}  
myres3{ANY_NODE}  
myres4  
#
```

例 275 查看资源依赖性属性

以下示例说明如何使用 `scha_resource_get` command 检索依赖于两个不同逻辑主机名资源的每节点资源依赖性。要设置每节点资源依赖性，必须使用 `clresource set` 命令。该示例使用名为 `gds-rs` 的可伸缩资源，并设置 `gds-rs` 对 `ptrancos1` 上 `trancos-3-rs` 的依赖性以及对 `ptrancos2` 上 `trancos-4-rs` 的依赖性。

在 `ptrancos1` 节点中：

```
ptrancos1(/root)$ scha_resource_get -O RESOURCE_DEPENDENCIES -R gds-rs  
  
trancos-3-rs  
ptrancos1(/root)$ scha_resource_get -O RESOURCE_DEPENDENCIES_NODE -R gds-rs ptrancos1  
  
trancos-3-rs  
ptrancos1(/root)$ scha_resource_get -O RESOURCE_DEPENDENCIES_NODE -R gds-rs ptrancos2  
  
trancos-4-rs  
ptrancos1(/root)$ scha_resource_get -Q -O RESOURCE_DEPENDENCIES -R gds-rs  
  
trancos-3-rs@ptrancos1  
trancos-4-rs@ptrancos2  
ptrancos1(/root)$ scha_resource_get -O NETWORK_RESOURCES_USED -R gds-rs  
  
trancos-3-rs
```

在 `ptrancos2` 节点中：

```

ptrancos2(/root)$ scha_resource_get -0 RESOURCE_DEPENDENCIES -R gds-rs

trancos-4-rs
ptrancos2(/root)$ scha_resource_get -0 RESOURCE_DEPENDENCIES_NODE -R gds-rs ptrancos1

trancos-3-rs
ptrancos2(/root)$ scha_resource_get -0 RESOURCE_DEPENDENCIES_NODE -R gds-rs ptrancos2

trancos-4-rs
ptrancos2(/root)$ scha_resource_get -Q -0 RESOURCE_DEPENDENCIES -R gds-rs

trancos-3-rs@ptrancos1
trancos-4-rs@ptrancos2
ptrancos2(/root)$ scha_resource_get -0 NETWORK_RESOURCES_USED -R gds-rs

trancos-4-rs

```

返回以下退出状态代码：

0 命令已成功完成。

非零值 发生错误。
[scha_calls\(3HA\) \[897\]](#) 中介绍了故障错误代码。

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[Unresolved link to "awk1"](#)、[scha_cmds\(1HA\) \[573\]](#)、[scha_calls\(3HA\) \[897\]](#)、[scha_resource_get\(3HA\) \[963\]](#)、[rt_reg\(4\) \[1067\]](#)、[link to "attributes5"](#)、[property_attributes\(5\) \[1087\]](#)、[r_properties\(5\) \[1103\]](#)、[rt_properties\(5\) \[1143\]](#)

名称

scha_resource_setstatus — 设置资源状态

```
scha_resource_setstatus -R resource -G group -s status [-m msg]
                        [-Z zonename]
```

scha_resource_setstatus 命令可设置由资源组管理器 (Resource Group Manager, RGM) 管理的资源的 Status 和 Status_msg 属性。该命令供资源的监视器用来指示它所识别的资源状态。该命令提供的功能与 [scha_resource_setstatus\(3HA\) \[1003\]](#) C 函数的功能相同。

执行 [scha_resource_setstatus\(1HA\) \[593\]](#) 命令时，会使用所指定的值更新资源的 Status 和 Status_msg 属性。Oracle Solaris Cluster 将在群集系统日志中记录对资源状态所做的更改，您可以使用群集管理工具查看该日志。

需要具有 solaris.cluster.resource.admin RBAC 授权才能使用该命令。请参见 [Unresolved link to "rbac5"](#)。

您还必须能够承担已向其分配 Oracle Solaris Cluster 命令权限配置文件的角色，才能使用该命令。经授权的用户可以通过 [Unresolved link to "pfsh1"](#)、[Unresolved link to "pfcsh1"](#) 或 [Unresolved link to "pfksh1"](#) 配置文件 shell 在命令行中发出授权的 Oracle Solaris Cluster 命令。配置文件 shell 是一种特殊的 shell，通过该 shell，可以访问分配给 Oracle Solaris Cluster 命令权限配置文件的 Oracle Solaris Cluster 特权命令。当您运行 [Unresolved link to "su1M"](#) 以承担某个角色时，即会启动配置文件 shell。也可以使用 [Unresolved link to "pfexec1"](#) 发出授权的 Oracle Solaris Cluster 命令。

支持以下选项：

-G group

指定包含该资源的资源组。

-m msg

指定要分配给资源的 Status_msg 属性的文本字符串。如果未指定该选项，资源的 Status_msg 值将设置为 NULL。

-R resource

指定要设置其状态的资源。

-s status

指定 status 的值：OK、DEGRADED、FAULTED、UNKNOWN 或 OFFLINE。

-Z zonename

指定在其中配置要运行的资源组的区域群集的名称。

如果 `Global_zone` 属性设置为 `TRUE`，则即使在区域群集中配置了包含资源的资源组，也会在全局区域中执行方法。`-Z` 选项设置的是运行资源组的非全局区域而不是运行方法的全局区域的状态。

请仅对 `Global_zone` 属性设置为 `TRUE` 的资源类型使用 `-Z` 选项。如果 `Global_zone` 属性设置为 `FALSE`，则不需要此选项。有关 `Global_zone` 属性的更多信息，请参见 [rt_properties\(5\) \[1143\]](#) 手册页。

例 276 设置资源 R1 的状态并生成 `Status_msg`

以下命令将资源组 RG2 中资源 R1 的状态设置为 `OK`，并将 `Status_msg` 设置为 `Resource R1 is OK`：

```
scha_resource_setstatus -R R1 -G RG2 -s OK -m "Resource R1 is OK"
```

例 277 设置资源 R1 的状态但不生成 `Status_msg`

以下命令将资源组 RG2 中 R1 的状态设置为 `DEGRADED`，并将 `Status_msg` 设置为 `NULL`：

```
scha_resource_setstatus -R R1 -G RG2 -s DEGRADED
```

例 278 设置区域 Zone1 中资源 R1 的状态并生成 `Status_msg`

以下示例显示了一个作为 shell 脚本实现的资源类型方法或监视器。该 shell 脚本显示了如何将区域 `$localzone` 中资源组 `$rg` 中的资源 `$resource` 的状态设置为 `OK`。该 shell 脚本还将 `Status_msg` 设置为 `"Resource R1 is OK"`。在本例中，必须指定 `-Z` 选项，因为资源类型属性 `Global_zone` 被假定设置为 `TRUE`。

```
resource=
rg=""
localzone=""
zflag=""
while getopts R:G:Z:
do
    case $c in
        R) resource=$OPTARG;;
        G) rg=$OPTARG;;
        Z) zflag="-Z"
           localzone=$OPTARG;;
    esac
done
...
scha_resource_setstatus -R $resource -G $rg $zflag $localzone -s OK -m
    "Resource R1 is OK"
```

返回以下退出状态代码：

0 命令已成功完成。

非零值

发生错误。

[scha_calls\(3HA\) \[897\]](#) 中介绍了故障错误代码。

有关下列属性的说明，请参见 [Unresolved link to " attributes5"](#) :

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Stable (稳定)

[scha_cmds\(1HA\) \[573\]](#)、[scha_calls\(3HA\) \[897\]](#)、[scha_resource_setstatus\(3HA\) \[1003\]](#)、[Unresolved link to " attributes5"](#)、[Unresolved link to " rbac5"](#)、[rt_properties\(5\) \[1143\]](#)

名称

scha_resourcegroup_get — 访问资源组信息

```
scha_resourcegroup_get -o optag -G group [-Z zonecluster] [args]
```

scha_resourcegroup_get 命令可以访问受资源组管理器 (Resource Group Manager, RGM) 群集工具控制的资源组的相关信息。

该命令设计为在资源类型的回调方法的 shell 脚本实现中使用。这些资源类型代表由群集的 RGM 控制的服务。该命令提供的信息与 [scha_resourcegroup_get\(3HA\) \[1013\]](#) C 函数提供的信息相同。

该命令会以格式化字符串将信息生成到标准输出 (stdout)，如 [scha_cmds\(1HA\) \[573\]](#) 中所述。输出内容是单个字符串，或者是位于各自的行上的多个字符串。可以将输出存储在 shell 变量中，并使用 shell 工具或 [Unresolved link to "awk1"](#) 进行解析，以进一步供脚本使用。

要使用该命令，您需要具有 `solaris.cluster.resource.read` 基于角色的访问控制 (role-based access control, RBAC) 授权。请参见 [Unresolved link to "rbac5"](#)。

您还必须能够承担已向其分配 Oracle Solaris Cluster 命令权限配置文件的角色，才能使用该命令。经授权的用户可以通过 [Unresolved link to "pfsh1"](#)、[Unresolved link to "pfcsh1"](#) 或 [Unresolved link to "pfksh1"](#) 配置文件 shell 在命令行中发出授权的 Oracle Solaris Cluster 命令。配置文件 shell 是一种特殊的 shell，通过该 shell，可以访问分配给 Oracle Solaris Cluster 命令权限配置文件的 Oracle Solaris Cluster 特权命令。当您运行 [Unresolved link to "su1M"](#) 以承担某个角色时，即会启动配置文件 shell。也可以使用 [Unresolved link to "pfexec1"](#) 发出授权的 Oracle Solaris Cluster 命令。

支持以下选项：

`-G group`

资源组的名称。

`-o optag`

指定要访问的信息。根据您指定的 `optag`，可能需要包括一个附加操作数以指明要检索其信息的节点或区域。

注 - `optag` 值 (例如 `DESIRED_PRIMARYS` 和 `FAILBACK`) 不区分大小写。指定 `optag` 选项时，您可以使用任意大小写字母组合。

下列 `optag` 值将检索相应的资源组属性。将生成资源组的指定属性的值。RG_STATE 属性在执行命令的特定节点上引用该值。

ALL_LOAD_FACTORS
ALL_LOAD_FACTOR_NAMES

AUTO_START_ON_NEW_CLUSTER
DESIRED_PRIMARYES
FAILBACK
GLOBAL_RESOURCES_USED
IMPLICIT_NETWORK_DEPENDENCIES
LOAD_FACTOR
MAXIMUM_PRIMARYES
NODELIST
PATHPREFIX
PINGPONG_INTERVAL
PREEMPTION_MODE
PRIORITY
RESOURCE_LIST
RG_AFFINITIES
RG_DEPENDENCIES
RG_DESCRIPTION
RG_IS_FROZEN
RG_MODE
RG_PROJECT_NAME
RG_SLM_TYPE
RG_SLM_PSET_TYPE
RG_SLM_CPU
RG_SLM_CPU_MIN
RG_STATE
RG_STATE_NODE
RG_SYSTEM
SUSPEND_AUTOMATIC_RECOVERY
TARGET_NODES

-z zoneclustername

指定包含资源组并想要对其执行操作的群集。当命令是在全局区域中执行但却需要在指定区域群集上进行操作时，此选项适用。无法在某区域群集内执行该命令来访问其他区域群集。

zoneclustername 指定在名为 *zoneclustername* 的区域群集中执行查询。

如果省略 -z 选项，则将在执行命令的群集中执行查询。

要查询每区域属性（例如，全局群集中的资源组状态）的值，请不要使用 -z 选项，而应使用查询标记的每区域格式。例如，使用 *RG_STATE_NODE* 而不是 *RG_STATE*，并提供 *nodename:zonename* 格式的附加命令行参数。

注 - *RG_STATE_NODE* 需要一个不带标志的参数以指定节点。该 *optag* 值将为指定的节点生成资源组的 *RG_STATE* 属性值。如果该不带标志的参数指定非全局区域，则格式为 *nodename:zonename*。

例 279 使用 *scha_resourcegroup_get* 的样例脚本

向下面的脚本传递了一个 -G 参数，该参数提供所需的资源组名称。随后，使用 *scha_resourcegroup_get* 命令来获取资源组中的资源的列表。

```
#!/bin/sh

while getopts G: opt
do
    case $opt in
        G)    group="$OPTARG";;
        esac
    done

    resource_list=`scha_resourcegroup_get -O Resource_list -G $group`

    for resource in $resource_list
    do
        printf "Group: %s contains resource: %s\n" "$group" "$resource"
    done
```

例 280 使用 `scha_resourcegroup_get` 命令查询资源组上的所有负载因子

使用以下命令可以查看名为 `rg1` 的资源组上的所有负载因子。

```
# scha_resourcegroup_get -O ALL_LOAD_FACTORS -G rg1
factor1=50
factor2=1
factor3=0
```

例 281 使用 `scha_resourcegroup_get` 命令列出资源组的所有已定义负载因子名称

使用以下命令可以检索名为 `rg1` 的资源组上的所有已定义负载因子的列表。

```
# scha_resourcegroup_get -O ALL_LOAD_FACTOR_NAMES -G rg1

factor1
factor2
factor3
```

例 282 使用 `scha_resourcegroup_get` 命令查询资源组的特定负载因子

使用以下命令可以查看名为 `rg1` 的资源组上的某个特定负载因子。

```
# scha_resourcegroup_get -O LOAD_FACTOR -G rg1 factor1

50
```

例 283 使用 `scha_resourcegroup_get` 命令查询资源组的优先级

使用以下命令可以查看为资源组 `rg1` 设置的优先级。

```
# scha_resourcegroup_get -O PRIORITY -G rg1

501
```

例 284 使用 `scha_resourcegroup_get` 命令查询资源组的抢占模式

使用以下命令可以查看为资源组 `rg1` 设置的抢占模式。

```
# scha_resourcegroup_get -O PREEMPTION_MODE -G rg1  
Has_Cost
```

返回以下退出状态代码：

0 命令已成功完成。

非零值 发生错误。

[scha_calls\(3HA\) \[897\]](#) 中介绍了故障错误代码。

有关下列属性的说明，请参见 [Unresolved link to " attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Stable (稳定)

[Unresolved link to "](#)

[awk1"](#)、[clnode\(1CL\) \[149\]](#)、[scha_cmds\(1HA\) \[573\]](#)、[scha_calls\(3HA\) \[897\]](#)、[scha_resourcegroup_get\(3HA\) \[897\]](#)、[Unresolved link to " attributes5"](#)、[rg_properties\(5\) \[1129\]](#)、[Unresolved link to " rbac5"](#)

名称

scha_resourcetype_get — 访问资源类型信息

```
scha_resourcetype_get -o optag -T type [-Z zoneclustername]
```

scha_resourcetype_get 命令可以访问向资源组管理器 (Resource Group Manager, RGM) 注册的资源类型的相关信息。

请在代表由 RGM 控制的服务的资源类型的回调方法 shell 脚本实现中使用该命令。该命令提供的信息与 [scha_resourcetype_get\(3HA\) \[1041\]](#) C 函数提供的信息相同。

该命令会以格式化字符串将信息输出到标准输出 (stdout)，如 [scha_cmds\(1HA\) \[573\]](#) 手册页中所述。输出内容是单个字符串，或者是位于各自的行上的多个字符串。可以将输出存储在 shell 变量中。您还可以使用 [Unresolved link to "awk1"](#) 命令或其他 shell 命令解析输出，以进一步供脚本使用。

需要具有 `solaris.cluster.resource.read` RBAC 授权才能使用该命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

经授权的用户可以通过 [Unresolved link to "pfsh1"](#)、[Unresolved link to "pfcsh1"](#) 或 [Unresolved link to "pfksh1"](#) 配置文件 shell 在命令行上发出 Oracle Solaris Cluster 特权命令。配置文件 shell 是一种特殊的 shell，通过该 shell，可以访问分配给 Oracle Solaris Cluster 命令权限配置文件的 Oracle Solaris Cluster 特权命令。当您运行 [Unresolved link to "su1M"](#) 以承担某个角色时，即会启动配置文件 shell。也可以使用 [Unresolved link to "pfexec1"](#) 来发出特权 Oracle Solaris Cluster 命令。

支持以下选项：

`-o optag`

表示要访问的信息。

注 - `optag` 选项 (例如 `API_VERSION` 和 `BOOT`) 不区分大小写。指定 `optag` 选项时，您可以使用任意大小写字母组合。

以下 `optag` 值将检索相应的资源类型属性。输出是资源类型的指定属性的值。

```
API_VERSION  
BOOT  
FAILOVER  
FINI  
GLOBAL_ZONE  
INIT
```

INIT_NODES
 INSTALLED_NODES
 IS_LOGICAL_HOSTNAME
 IS_SHARED_ADDRESS
 MONITOR_CHECK
 MONITOR_START
 MONITOR_STOP
 PKGLIST
 POSTNET_STOP
 PRENET_START
 PROXY
 RESOURCE_LIST
 RT_BASEDIR
 RT_DESCRIPTION
 RT_SYSTEM
 RT_VERSION
 SINGLE_INSTANCE
 START
 STOP
 UPDATE
 VALIDATE

-T type

是所注册的供 RGM 群集工具使用的资源类型的名称。

-z zoneclustername

指定要对其执行操作的群集。当命令是在全局区域中执行但却需要在指定区域群集上进行操作时，此选项适用。无法在某区域群集内执行该命令来访问其他区域群集。

zoneclustername 指定在名为 *zoneclustername* 的区域群集中执行查询。

如果省略 **-z** 选项，则将在执行命令的群集中执行查询。

返回以下退出值：

0 命令已成功完成。

非零值 发生错误。

[scha_calls\(3HA\) \[897\]](#) 中介绍了故障错误代码。

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Stable (稳定)

Unresolved link to "

awk1", [scha_cmds\(1HA\) \[573\]](#), [scha_calls\(3HA\) \[897\]](#), [scha_resourcetype_get\(3HA\) \[1041\]](#), Unresolved
link to " [attributes5](#)", [rt_properties\(5\) \[1143\]](#)

OSC4 1m

名称

ccradm — CCR 表文件管理命令

```
/usr/cluster/lib/sc/ccradm subcommand [-?]  
  
/usr/cluster/lib/sc/ccradm addkey [-Z zoneclustername] -v value -k key ccrtablefile  
  
/usr/cluster/lib/sc/ccradm addtab [-Z zoneclustername] ccrtablefile  
  
/usr/cluster/lib/sc/ccradm changekey [-Z zoneclustername] -v value -k key ccrtablefile  
  
/usr/cluster/lib/sc/ccradm delkey [-Z zoneclustername] -k key ccrtablefile  
  
/usr/cluster/lib/sc/ccradm recover [-Z zoneclustername] -f -o ccrtablefile  
  
/usr/cluster/lib/sc/ccradm remtab [-Z zoneclustername] ccrtablefile  
  
/usr/cluster/lib/sc/ccradm replace [-Z zoneclustername] -i newdatafile ccrtablefile  
  
/usr/cluster/lib/sc/ccradm showkey [-Z zoneclustername] -k key ccrtablefile
```

ccradm 命令支持群集配置系统信息库 (Cluster Configuration Repository, CCR) 信息的管理。

CCR 信息位于 `/etc/cluster/ccr` 目录下。关于全局群集的 CCR 信息位于 `/etc/cluster/ccr/global` 目录下。关于区域群集 *zoneclustername* 的 CCR 信息位于 `/etc/cluster/ccr/zoneclustername` 目录下。CCR 信息只应通过受支持的编程接口访问。文件权限有意设置为不允许直接访问 CCR 信息。

CCR 信息以表的形式存储，每个表存储在自己对应的文件中。CCR 表文件的每一行由两个 ASCII 字符串组成，一个代表关键字，另一个代表值。每个 CCR 文件都以生成号 `ccr_gennum` 和校验和 `ccr_checksum` 开头。

`ccr_gennum` 表示 CCR 表文件的当前生成号。由系统管理 `ccr_gennum`。最大的编号代表最新版本的文件。

`ccr_checksum` 表示 CCR 表内容的校验和，用于对表中数据进行一致性检查。系统不会使用校验和无效的 CCR 表文件。

ccrtablefile 文件代表本地节点上的 CCR 表（变量为文件名）。指定了 `-z` 选项时，*ccrtablefile* 属于指定的区域群集。未指定 `-z` 选项时，*ccrtablefile* 属于全局群集。请注意，全局群集和区域群集可以有各自的 *ccrtablefile* 文件，其文件名相同但包含的信息不同。

仅可以在全局区域中使用此命令。

该命令具有以下子命令：

addkey

向指定群集的 CCR 表文件添加关键字值对。与 -s 选项一起使用时，数据为字符串值。与 -f 选项一起使用时，值为文件中的第一个字符串，且该文件只包含一个字符串。如果文件不是这种格式，该命令将返回错误。

该子命令只能在群集模式下使用。

addtab

在 CCR 中为指定群集创建一个表。表最初只包含 ccr_gennum 和 ccr_checksum。

该子命令只能在群集模式下使用。

changekey

根据指定的关键字和新值修改 CCR 表文件中该关键字的值。如果在 CCR 表文件中找不到该关键字，该命令将返回 ESPIPE。与 -s 选项一起使用时，数据为字符串值。与 -f 选项一起使用时，值为文件中的第一个字符串，且该文件只包含一个字符串。如果文件不是这种格式，该命令将返回错误。

该子命令只能在群集模式下使用。

delkey

根据指定的关键字，从 CCR 表文件中删除关键字值对。如果在 CCR 表文件中找不到该关键字，该命令将返回 ESPIPE。

该子命令只能在群集模式下使用。

recover



注意 - 该子命令只供精通 CCR 内部操作的工程师使用。该子命令支持手动恢复操作。普通用户不应使用该子命令。

recover 子命令总是设置 CCR 表文件的 ccr_gennum 值、重新计算该文件的校验和并设置 ccr_checksum 值。

使用不带 -o 选项的 recover 子命令时，recover 子命令会将生成号设置为 INIT_VERSION，即 -1。生成号 INIT_VERSION 表示该 CCR 表文件仅在本地节点重新加入群集后才生效，重新加入时群集将使用群集内另一节点的 CCR 表文件的内容替换该 CCR 表文件的内容。前提条件是，群集内必须有其他节点为该 CCR 表文件设置了覆盖版本，或至少有一个其他节点有该 CCR 表文件的有效副本。CCR 表文件的校验和有效并且其生成号大于等于零时，该 CCR 表文件才有效。

如果 CCR 表文件在所有节点上的生成号都为 INIT_VERSION，则恢复过程完成后，该 CCR 表仍无效。因此，您必须至少对群集中一个节点的 CCR 表文件使用带有 -o 选项的 recover 子命令。

使用带 -o 选项的 recover 子命令时，recover 子命令会将生成号设置为 OVRD_VERSION，即 -2。生成号 OVRD_VERSION 表示系统会将本地节点上的 CCR 表文件的内容传播到群集中的其他所有节点。将内容传播到其他节点后，系统会将生成号更改为 0。应该只有一个节点的 CCR 表文件具有 OVRD_VERSION 值。如果在多个

节点上的相同 CCR 表文件中设置了 OVRD_VERSION，系统将随机使用其中一个 CCR 表文件的内容。

该子命令只能在非群集模式下使用。

rentab

从 CCR 中删除表。

该子命令只能在群集模式下使用。

replace



注意 - 该子命令只供精通 CCR 内部操作的工程师使用。该子命令支持手动恢复操作。普通用户不应使用该子命令。

将 *ccrdatafile* 的内容替换为 *newdatafile* 的内容。将重新计算校验和并将生成号重置为 0。

该子命令只能在群集模式下使用。

showkey

显示 CCR 表文件中指定关键字的值。如果在 CCR 表文件中找不到该关键字，该命令将返回 ESPIPE。showkey 命令仅在标准输出中写入指定关键字的行尾前的值字符串。如果发生了错误，该命令不会写入任何内容。

该子命令只能在群集模式下使用。

该命令具有以下选项：

-?

--help

显示帮助信息。

指定此选项时，子命令可有可无。

如果没有指定子命令，将显示包含所有可用子命令的列表。

如果指定了子命令，将显示该子命令的用法。

如果指定此选项和其他选项，将会忽略其他选项。

-f

--force

指定在节点以群集成员身份引导时，强制执行 recover 子命令。

-i *newdatafile*

--input=*newdatafile*

--input *newdatafile*

指定要用于恢复操作的 CCR 表文件。

-k
--key

指定要添加、删除或修改的关键字的名称。

-o
--override

覆盖选项与 `recover` 子命令一起使用。该选项将生成号设置为 `OVRD_VERSION`。

该选项用于将一个 CCR 表文件指定为主副本。在恢复过程中，该 CCR 表文件主版本将覆盖其余节点上该文件的其他版本。如果多个节点上的 CCR 表文件的生成号均为 `OVRD_VERSION`，系统将只选择其中一个文件，并在其中一个节点的控制台上输出警告消息。恢复后，表的生成号将重置为 0。

该选项只能在非群集模式下使用。

-v *value*
--value=*value*
--value *value*

指定 CCR 表的关键字值。值字符串中不能包含空格字符。这表示不能包含空格、制表符、回车或换行符。

-Z {*zoneclustername* | *global*}
--zoneclustername= {*zoneclustername* | *global*}
--zoneclustername {*zoneclustername* | *global*}

指定 CCR 事务必须在哪个群集中执行。所有子命令都支持该选项。如果指定此选项，还必须指定以下列表中的参数之一：

zoneclustername

指定使用此选项的命令仅对名为 *zoneclustername* 的区域群集中的所有指定资源组起作用。

global

指定使用此选项的命令仅对全局群集中的所有指定资源组起作用。

-z 选项只能在群集模式下使用。

仅支持以下操作数：

ccrtablefile

指定要管理的 CCR 表文件。只能指定一个 *ccrtablefile*。

`ccradm` 命令可用于对 CCR 表文件执行管理操作。

例 285 修复损坏的 CCR 表并重新计算校验和。

仅应在紧急修复过程中按照 Oracle 授权人员的指导执行以下步骤修复损坏的 CCR 表。

本示例修复 CCR 表 `ccr-file`。

1. 以非群集模式重新引导所有节点。
2. 编辑所有节点上的该文件，使其包含正确的数据。该文件必须在所有节点上保持一致。因为该文件在所有节点上均相同，也可以将其指定为所有节点的覆盖版本。
3. 通过对所有节点运行以下命令（其中 `ccr-file` 是 CCR 表的名称），重新计算校验和并将该 CCR 表文件指定为覆盖版本。

```
# ccradm recover -o ccr-file
```

4. 以群集模式重新引导所有节点。

例 286 根据备份版本恢复损坏的 CCR 表

本示例使用 CCR 表 `yyy` 的备份版本（`yyy.bak` 文件）中的内容替换该表。在群集模式下，从一个节点运行以下命令。

```
# ccradm replace -Z global -i /etc/cluster/ccr/global/yyy.bak /etc/cluster/ccr/global/yyy
```

例 287 创建 CCR 表

本示例在区域群集 `zc1` 中创建一个临时 CCR 表 `foo`。在群集模式下，从一个节点运行以下命令。

```
# ccradm addtab -Z zc1 foo
```

例 288 删除 CCR 表

本示例显示如何从全局群集中删除 CCR 表 `foo`。在群集模式下，从一个节点运行以下命令。

```
# ccradm remtab foo
```

例 289 修改 CCR 表

本示例将全局群集 CCR 表 `rgm_rg_nfs-rg` 中的 `Pingpong_interval` 属性的值更改为 `5400`。在群集模式下，从一个节点运行以下命令。

```
# ccradm changekey -s 5400 -k Pingpong_interval rgm_rg_nfs-rg
```

例 290 显示 CCR 表中的关键字值

本示例显示 CCR 表 `rgm_rg_nfs-rg` 中 `Pingpong_interval` 属性的值。

```
# ccradm showkey -k Pingpong_interval rgm_rg_nfs-rg
5400
```

返回以下退出值：

0

未发生错误。

>0

发生错误。

有关下列属性的描述，请参见 [Unresolved link to "attributes5"](#)。

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

名称

cl_eventd — 群集事件守护进程

```
/usr/cluster/lib/sc/cl_eventd [-v]
```

cl_eventd 守护进程在引导时启动，以监视由其他群集组件生成的系统事件。该守护进程还将这些事件转发至其他群集节点。只有类 EC_Cluster 的事件才会转发到其他群集节点。

支持以下选项：

-v 将其他故障排除和调试信息发送至 [Unresolved link to "syslogd1M"](#)。

```
/usr/cluster/lib/sc/cl_eventd 群集事件守护进程
```

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core

[Unresolved link to "syseventd1M"](#)、[Unresolved link to "syslog3C"](#)

cl_eventd 守护进程不提供可公开访问的界面。

名称

cl_pnmd — 公共网络管理 (Public Network Management, PNM) 服务守护进程

```
/usr/cluster/bin/cl_pnmd [-d [-t tracefile]]
```

cl_pnmd 是适用于公共网络管理 (Public Network Management, PNM) 模块的服务器守护进程。它通常是在系统引导时启动的。当它启动时，便启动 PNM 服务。

[Unresolved link to "in.mpathd1M"](#) 守护进程为本地主机中的所有 IP 网络多路径 (IP Network Multipathing, IPMP) 组执行适配器测试和节点内故障转移。

cl_pnmd 保持跟踪本地主机的 IPMP 状态，并有利于所有 IPMP 组的节点间故障转移。

支持以下选项：

- d 在 stderr 上显示调试消息。
- t *tracefile* 与 -d 选项结合使用时，将导致所有调试消息重定向到 *tracefile*。如果省略了 *tracefile*，则使用 /var/cluster/run/cl_pnmd.log。

cl_pnmd 是守护进程，没有与外部进行直接的 stdin、stdout 或 stderr 连接。所有诊断消息均是通过 [Unresolved link to "syslog3C"](#) 记录的。

cl_pnmd 必须以超级用户模式运行。

由于生成的调试消息卷，在较长一段时间内请勿使用 -t 选项。

cl_pnmd 是通过 pnm 启动脚本启动的。服务管理工具启动和停止守护进程。

SIGTERM 信号可用来正常地中止 cl_pnmd。不应使用其他信号来中止守护进程。

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

[Unresolved link to "ifconfig1M"](#)、[Unresolved link to "in.mpathd1M"](#)、[Unresolved link to "syslog3C"](#)、[Unresolved link to "attributes5"](#)

名称

dcscfg — 查询 DCS

```
/usr/cluster/lib/sc/dcs_cfg -c info [-s service-name | -C  
    service-class | -d device-path]
```

```
/usr/cluster/dtk/bin/dcs_cfg -c status [-s service-name]
```

```
/usr/cluster/lib/sc/dcs_cfg -c remove -s service-name
```

dcscfg 命令是一个紧急命令行接口，用于直接更新设备配置系统 (Device Configuration System, DCS)。只应在授权的 Oracle 支持人员指导下使用 dcscfg 的更新选项。请使用 cldevicegroup 命令执行 DCS 的所有常规更改。

要查询设备服务，请使用该命令的 info 或 status 格式。info 格式提供关于服务的常规配置信息。status 格式提供关于服务的当前状态的信息。不带有附加限定选项的 info 和 status 命令可显示正在使用的所有服务类和服务。

仅可以在全局区域中使用此命令。

支持以下选项：

-c command

指定要运行的 *command*：

info

显示有关指定的服务名称或所有服务（如果未指定任何服务）的信息。输出因服务类型而异，可以包括服务类、辅助项、切换回原状态、副本、实体或设备。

status

显示指定服务或所有服务（如果未指定任何服务名称）的服务状态。

remove

从 DCS 中删除指定的服务名称。此操作仅从群集中删除服务名称。不会从 Oracle Solaris 中删除服务。例如，如果使用 dcscfg 删除元集，Solaris Volume Manager 不会删除磁盘集。

-C service-class

指定服务类。有效的服务类包括：SUNWmd、DISK、TAPE 和 SUNWlocal。

-d device-path

指定设备路径。

-s *service-name*

指定服务名称。有效的服务名称包括元集和磁盘。

例 291 显示有关磁盘的信息

此示例显示有关磁盘 dsk/d5 的信息：

```
# dcs_config -c info -s dsk/d5
Service name: dsk/d5
Service class: DISK
Switchback Enabled: False
Number of secondaries: All
Replicas: (Node id --> 1, Preference --> 0)(Node id --> 2, Preference --> 0)
Devices: (239, 160-167)
Properties:
  gdev --> d5
  autogenerated --> 1
```

例 292 删除无法识别的 Solaris Volume Manager 元集

在此示例中，群集软件可识别元集 nfs-set，但 Solaris Volume Manager 不能。cldevicegroup status 命令可显示元集：

```
=== Cluster Device Groups ===

--- Device Group Status ---
Device Group Name      Primary      Secondary      Status
-----
nfs-set                -            -              Offline
```

metaset 命令不识别设置：

```
# metaset -s nfs-set
metaset: setname "nfs-set": no such set
```

从一个节点运行以下 dcs_config 命令将从群集中删除 nfs-set：

```
# dcs_config -c remove -s nfs-set
```

例 293 显示 metaset 的状态

此示例显示 nfs-set 元集的状态。

```
# dcs_config -c status -s nfs-set
Service Name: nfs-set
Active replicas: (1. State - Primary)(2. State - Secondary)
Service state: SC_STATE_ONLINE
```

有关下列属性的描述，请参见 [Unresolved link to "attributes5"](#)。

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

[cldevicegroup\(1CL\) \[65\]](#)、[Unresolved link to "metaset1M"](#)

名称

halockrun — 在持有文件锁时运行子程序

```
/usr/cluster/bin/halockrun [-nsv] [-e exitcode] lockfilename  
  prog [args]
```

halockrun 实用程序提供了一个便捷的方式对文件声明文件锁并在持有该锁时运行程序。因为该实用程序支持脚本锁定，所以当以脚本语言（例如 Bourne shell）编程时，该实用程序非常有用。请参见 [Unresolved link to "sh1"](#)。

halockrun 打开文件 *lockfilename* 并对整个文件声明独占模式文件锁。请参见 [Unresolved link to "fcntl2"](#)。然后它将带有参数 *args* 的程序 *prog* 作为子进程运行并等待子进程退出。子进程退出时，halockrun 释放该锁并以该子进程退出时所使用的同一退出代码退出。

整体效果为子进程 *prog* 作为临界部分运行并且该临界部分格式正确，因为无论子进程如何终止，都会释放该锁。

如果无法打开或创建文件 *lockfilename*，halockrun 将在 `stderr` 上显示错误消息并退出，退出代码为 99。

支持以下选项：

`-e exitcode`

通常，由 halockrun 检测到的错误以退出代码 99 退出。`-e` 选项可以将该特殊退出代码更改为其他值。

`-n`

应在非阻塞模式中请求锁：如果无法立即授予锁，halockrun 将立即以退出代码 1 退出，而不运行 *prog*。该行为不会受到 `-e` 选项影响。

如果没有 `-n` 选项，将在阻塞模式中请求锁，因此 halockrun 实用程序会阻止等待该锁变为可用。

`-s`

在共享模式而不是在独占模式中声明文件锁。

`-v`

详细输出，位于 `stderr` 上。

错误由 halockrun 自身检测，因此从不会启动子进程，导致 halockrun 以退出代码 99 退出。（使用 `-e` 选项可以将该退出代码值更改为其他值。请参见“选项”部分。

否则，`halockrun` 以该子进程退出时所使用的同一退出代码退出。

有关下列属性的说明，请参见 [Unresolved link to " attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

[Unresolved link to " fcntl2"](#), [Unresolved link to " attributes5"](#)

名称

hatimerun — 超时时运行子程序

```
/usr/cluster/bin/hatimerun -t timeOutSecs [-adv] [-e exitcode] prog args
```

```
/usr/cluster/bin/hatimerun -t timeOutSecs [-dv] [-e exitcode] [-k signalname] prog args
```

hatimerun 实用程序为其他子程序的执行超时提供一个方便的工具。当以脚本语言（如 Bourne shell）编程时，该实用程序很有用。请参见 [Unresolved link to "sh1"](#)。

hatimerun 实用程序将包含参数 *args* 的程序 *prog* 作为超时时段内的子进程及其自己的进程组运行。超时是由 `-ttimeOutSecs` 选项指定的（秒）。如果超时到期，则 hatimerun 使用 SIGKILL 信号终止子进程的进程组，然后退出，并显示退出代码 99。

支持以下选项：

- a 彻底更改 hatimerun 的含义：不同于在超时到期时终止子进程，hatimerun 实用程序只是退出，并显示退出代码 99，保留子进程异步运行。
同时提供 -a 选项和 -k 选项是非法的。
- d 启用超时延迟。此选项延迟启动超时时钟，直到程序 `_prog_` 开始执行。在负载沉重的系统上，在子进程派生与指定的程序开始执行这两者之间可能有几秒的延迟。使用 -d 选项避免了将额外的执行前时间计入所分配的超时时间。
- e 将超时案例的退出代码更改为 99 以外的某个其他值。
- k 指定用于终止子进程组的信号。可能的信号名称与 [Unresolved link to "kill1"](#) 命令识别的名称相同。需特别指出的是，信号名称应为在 `<signal.h>` 描述中定义的符号名称之一。信号名称是以不区分大小写的方式识别的，没有 SIG 前缀。向 -k 选项提供数字参数也是合法的，在这种情况下使用信号编号。
同时提供 -a 选项和 -k 选项是非法的。
- t 指定超时周期（秒）。
- v 详细输出，位于 `stderr` 上。

如果发生超时，则 hatimerun 将退出，并显示退出代码 99（可以使用 -e 选项将其覆盖为某个其他值）。

如果未发生超时，但是 `hatimerun` 实用程序检测到某个其他错误（与子程序检测到的错误相反），则 `hatimerun` 将退出，并显示退出代码 88。

否则，`hatimerun` 将以子程序的退出状态退出。

`hatimerun` 实用程序捕捉信号 `SIGTERM`。它通过终止子进程（就像发生了超时）响应信号，然后退出，并显示退出代码 98。

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving（发展中）

[Unresolved link to "kill1"](#), [Unresolved link to "sh1"](#), [Unresolved link to "attributes5"](#)

名称

pmfadm — 进程监视器工具管理

```
/usr/cluster/bin/pmfadm -c nametag [-a action] [[-e ENV_VAR=  
    env.var...] | -E] [-n retries] [-t period] [-C level#]  
    command [args-to-command...]  
  
/usr/cluster/bin/pmfadm -k nametag [-w timeout] [signal]  
  
/usr/cluster/bin/pmfadm -L [-h host]  
  
/usr/cluster/bin/pmfadm -l nametag [-h host]  
  
/usr/cluster/bin/pmfadm -m nametag [-n retries] [-t period]  
  
/usr/cluster/bin/pmfadm -q nametag [-h host]  
  
/usr/cluster/bin/pmfadm -s nametag [-w timeout] [signal]
```

pmfadm 实用程序为进程监视器设备提供了管理命令行界面。

进程监视器工具可以监视进程及其后代进程，并在它们无法保持活动状态时重新启动它们。可以指定允许的故障总数，并将其限制在特定的时间段。在指定时间段内达到最大故障数目之后，会在控制台记录一条消息，并不再重新启动进程。

如果已指定 *action* 程序，则在达到允许的故障数目时会调用该程序。如果 *action* 程序以非零状态退出，将会从进程监视器设备中删除进程 *nametag*。否则，将使用传递到 pmfadm 中的原始参数重新启动该进程。

在进程监视器控制下启动的进程以启动请求的用户的有效用户 ID (euid) 和有效组 ID (egid) 运行。只有原始用户或 root 用户可以处理与这些进程关联的 *nametag*。但是，任何调用方（本地或远程）都能使用状态信息。

初始启动的进程产生的所有进程及其后续产生的进程都会受到监视。仅当最后一个进程或子进程退出时，进程监视器才会尝试重新启动该进程。

支持以下选项：

-a *action*

当进程无法保持活动状态时要调用的操作程序。该程序必须在 -a 选项的单个参数中指定，但可以用引号括起的包含多个组件的字符串。在任一情况下，均按照所指定的那样执行字符串，并有两个附加参数：发生的事件（当前仅包括 *failed*）以及与该进程关联的 *nametag*。在执行命令之前，会重新实例化当前目录和 PATH 环境变量。不会保留（也不应假定保留）任何其他环境变量。

如果操作程序以状态 0 退出，则使用指定给 pmfadm 的原始参数再次启动进程。任何其他退出状态都会导致该 *nametag* 不再存在于进程监视器范围内。

如果不指定 `-a` 选项，其结果等同于指定始终以非零状态退出的操作脚本。

`-c level#`

当启动进程时，可监视该进程及其子进程的最高级别 `level#`（含该级别）。`level#` 的值必须为大于或等于零的整数。原始进程在级别 0 执行，它的子进程在级别 1 执行，而这些子进程的子进程在级别 2 执行，依此类推。任何新派生操作都会产生新的子进程级别。

该选项可对监视哪些进程提供更多的控制。它对于监视派生新进程的服务器非常有用。

如果不指定该选项，将监视所有子进程，并且直到原始进程及其所有子进程停止后，才会重新启动原始进程。

如果服务器派生新的进程以处理客户端请求，则可能只需要监视该服务器。如果服务器停止，即使某些客户端进程仍在运行，也需要重新启动该服务器。适当的监视级别为 `-c 0`。

如果父进程在派生子进程之后退出，则将需要监视该子进程。用于监视子进程的级别为 `-c 1`。当两个进程都停止时，将重新启动服务器。

`-c nametag`

启动进程，并以 `nametag` 作为标识符。跟随命令行标志的所有参数作为相关进程执行。在执行命令之前，进程监视器设备会重新实例化当前目录和 `PATH` 环境变量。不会保留（也不应假定保留）任何其他环境变量。

如果 `nametag` 已存在，则 `pmfadm` 将以退出状态 1 退出，不会产生其他影响。

命令行参数中不支持 `I/O` 重定向。如果有必要进行此操作，则应创建执行该重定向的脚本，并将其作为 `pmfadm` 执行的命令使用。

`-E`

将整个 `pmfadm` 环境传递到新的进程。默认不使用该选项，在这种情况下，会传递 `rpc.pmfd` 环境与 `pmfadm` 环境的路径。

`-e` 和 `-E` 选项互斥，即，不能在单一命令中使用这两个选项。

`-e ENV_VAR=env.value`

传递到新进程的执行环境中的 `ENV_VAR=env.value` 格式的环境变量。该选项可以重复，因此可以传递多个环境变量。默认不使用该选项，在这种情况下，会传递 `rpc.pmfd` 环境与 `pmfadm` 环境的路径。

`-h host`

要连接的主机的名称。默认值为 `localhost`。

`-k nametag`

将指定的信号发送到与 `nametag` 关联的进程，其中包括与操作程序（如果其当前正在运行）关联的任何进程。如果未指定任何信号，则将发送默认信号 `SIGKILL`。如果进程及其后代进程退出，并仍有重试次数，则进程监视器将重新启动该进程。指定的信号是由 `kill` 命令所识别的同一组名称。

-
- L
返回发出该命令的用户所具有的所有运行标记的列表，如果用户为 root 用户，将显示在服务器上运行的所有标记。
- l *nametag*
输出有关 *nametag* 的状态信息。该命令的输出主要用于诊断，并且并非一成不变。
- m *nametag*
修改 *nametag* 的重试次数或观察重试的时间段。更改这些参数之后，将会清除早期失败的历史记录。
- n *retries*
在指定的时间段内允许的重试次数。该字段的默认值为 0，表示进程退出之后将不再重新启动。允许的最大值为 100。值 -1 表示重试次数不受限制。
- q *nametag*
指示 *nametag* 是否已注册并在进程监视器下运行。如果是，则返回 0，否则，返回 1。其他返回值表示出现错误。
- s *nametag*
停止重新启动与 *nametag* 关联的命令。如果指定信号，则将其发送到所有进程，其中包括操作脚本及其进程（如果它们当前正在执行）。如果未指定信号，则不进行发送。停止监视进程并不意味着这些进程不再存在。在这些进程及其所有后代进程退出之前，它们仍保持运行。指定的信号是由 kill 命令所识别的同一组名称。
- t *period*
计算故障的分钟数。该标志的默认值为 -1，它表示不限定。如果指定该参数，则不会将指定时段之外发生的进程故障计算在内。
- w *timeout*
当与 -s *nametag* 或 -k *nametag* 标志结合使用时，等待指定的秒数直到与 *nametag* 关联的进程退出。如果超时到期，则 pmfadm 以退出状态 2 退出。该标志的默认值为 0，表示命令立即返回，不等待任何进程退出。
如果给定值 -1，pmfadm 将无限等待，直至与该标志关联的进程退出。pmfadm 进程不会释放它所使用的 RPC 服务器线程，直至达到 RPC 超时期限。因此，应避免不必要地将 -w *timeout* 值设置为 -1。

例 294 启动将不会重新启动的睡眠进程

以下示例将启动名为 sleep.once 的睡眠进程，该进程退出之后将不会再重新启动：

```
example% pmfadm -c sleep.once /bin/sleep 5
```

例 295 启动睡眠进程并重新启动它

以下示例启动睡眠进程并重新启动它，最多重新启动一次：

```
example% pmfadm -c sleep.twice -n 1 /bin/sleep 5
```

例 296 启动睡眠进程并重新启动它

以下示例启动睡眠进程并重新启动它，最多每分钟重新启动两次。当超过可接受的故障次数而无法继续运行时，它将调用 /bin/true：

```
example% pmfadm -c sleep.forever -n 2 -t 1 -a /bin/true /bin/sleep 60
```

例 297 列出 sleep.forever Nametag 的当前状态

以下命令可列出 sleep.forever nametag 的当前状态：

```
example% pmfadm -l sleep.forever
```

例 298 将 SIGHUP 发送到所有进程

以下命令将 SIGHUP 发送到所有与 sleep.forever 关联的进程，最多等待 5 秒直到所有进程退出。

```
example% pmfadm -w 5 -k sleep.forever HUP
```

例 299 停止监视进程并发送 SIGHUP

以下命令停止监视（重新启动）与 sleep.forever 关联的进程，并将 SIGHUP 发送到与其相关的任何进程。在提供信号之后，该命令会立即返回，但可能在所有进程退出之前。

```
example% pmfadm -s sleep.forever HUP
```

例 300 列出用户所具有的所有运行的标记

如果用户发出以下命令：

```
example% pmfadm -c sleep.once /bin/sleep 30
example% pmfadm -c sleep.twice /bin/sleep 60
example% pmfadm -c sleep.forever /bin/sleep 90
```

以下命令：

```
example% pmfadm -L
```

的输出为

```
sleep.once sleep.twice sleep.forever
```

返回以下退出值：

- 0 成功完成。
- 1 *nametag* 不存在，或者尝试创建已存在的 *nametag*。
- 2 命令超时。
- 其他非零值 发生错误。

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

[Unresolved link to "kill1"](#), [rpc.pmfd\(1M\) \[633\]](#), [Unresolved link to "attributes5"](#)

名称

rpc.pmfd, pmfd — 基于 RPC 的进程监视器服务器

`/usr/cluster/lib/sc/rpc.pmfd`

rpc.pmfd 守护进程是 Oracle 的 ONC RPC 服务器，用于提供 Oracle Solaris Cluster 软件所使用的进程监视器工具。该守护进程在系统启动时首次启动。

必须以超级用户身份启动 rpc.pmfd 守护进程，以便能够以提交已排队等候监视的命令的用户身份运行这些命令。

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

[Unresolved link to "truss1"](#)[Unresolved link to "attributes5"](#)

诊断消息通常会记录到控制台。

为了避免与其他控制进程冲突，truss 命令不允许跟踪它检测到由另一进程通过 /proc 接口控制的进程。

名称

rpc.pmfd, pmfd — 基于 RPC 的进程监视器服务器

`/usr/cluster/lib/sc/rpc.pmfd`

rpc.pmfd 守护进程是 Oracle 的 ONC RPC 服务器，用于提供 Oracle Solaris Cluster 软件所使用的进程监视器工具。该守护进程在系统启动时首次启动。

必须以超级用户身份启动 rpc.pmfd 守护进程，以便能够以提交已排队等候监视的命令的用户身份运行这些命令。

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

[Unresolved link to "truss1"](#)[Unresolved link to "attributes5"](#)

诊断消息通常会记录到控制台。

为了避免与其他控制进程冲突，truss 命令不允许跟踪它检测到由另一进程通过 /proc 接口控制的进程。

名称

sc_zonesd — Oracle Solaris Cluster 区域管理守护进程

/usr/cluster/lib/sc/sc_zonesd

sc_zonesd 守护进程是 Oracle Solaris Cluster 软件使用的系统守护进程。该守护进程在系统启动时首次启动。

该守护进程仅在全局区域中运行。

所有诊断消息均是通过 syslog 函数进行记录的。

sc_zonesd 守护进程必须以超级用户模式启动。

sc_zonesd 守护进程受 SMF 服务 sc_zones 控制。如果该守护进程终止或 SMF 服务被禁用，群集节点将发生紧急情况。

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Private (专用)

[Unresolved link to "syslog3C"](#)、[Unresolved link to "attributes5"](#)

名称

scconf — 更新 Oracle Solaris Cluster 软件配置

```
scconf -a [-Hv] [-h node_options] [-A adapter_options] [-B  
switch_options] [-m cable_options] [-P privatehostname_options]  
[-q quorum_options] [-D devicegroup_options] [-T  
authentication_options]  
  
scconf -c [-Hv] [-C cluster_options] [-A adapter_options] [-B  
switch_options] [-m cable_options] [-P privatehostname_options]  
[-q quorum_options] [-D devicegroup_options] [-S slm_options]  
[-T authentication_options] [-w heartbeat_options]  
  
scconf -r [-Hv] [-h node_options] [-A adapter_options] [-B  
switch_options] [-m cable_options] [-P privatehostname_options]  
[-q quorum_options] [-D devicegroup_options] [-T  
authentication_options]  
  
scconf -p [-Hv [v]]  
  
scconf [-H]
```

注 - Oracle Solaris Cluster 软件包括面向对象的命令集。虽然 Oracle Solaris Cluster 软件仍支持原始的命令集，但是 Oracle Solaris Cluster 过程文档仅使用面向对象的命令集。有关面向对象的命令集的更多信息，请参见 [Intro\(1CL\) \[17\]](#) 手册页。

scconf 命令可管理 Oracle Solaris Cluster 软件配置。您可以使用 scconf 将项添加到配置中、更改以前配置的项的属性，以及从配置中删除项。在此命令的三种格式中，选项都按照它们在命令行上的键入顺序进行处理。与每个选项关联的所有更新必须已成功完成，然后才会考虑下一个选项。

当组成原始磁盘组或 Solaris Volume Manager 元集的磁盘使用基于控制器的复制来实现数据可用性时，还可以使用 scconf 命令来注册这些元集和磁盘组。在使用 scconf 命令注册磁盘组和元集之前，请确保磁盘组中的所有磁盘要么都是复制的，要么都是非复制的，而不能同时包含这两种磁盘。另外，您必须执行带 -T 或 -t 选项的 scdidadm 命令，或者执行 cldevice replicate 命令。这些命令会将 DID 设备配置为使用基于控制器的复制。有关更多信息，请参见 [scdidadm\(1M\) \[677\]](#) 手册页或 [cldevice\(1CL\) \[51\]](#) 手册页。

只能从活动群集节点上运行 scconf 命令。使用哪个节点运行该命令没有区别，只要该节点在群集中处于活动状态即可。无论使用哪个节点，运行该命令的结果始终相同。

使用 scconf 的 -p 选项可以显示当前配置的列表。

scconf 命令的所有格式都接受 -H 选项。指定 -H 将会显示帮助信息，其他所有选项将被忽略且不执行。不带任何选项调用 scconf 时也会显示帮助信息。

仅可以在全局区域中使用此命令。

基本选项

以下选项是 `scconf` 命令的所有格式共有的：

`-h`

如果在命令行上的任何位置指定了该选项，都将会显示帮助信息。其他所有选项将被忽略且不执行。不带任何选项调用 `scconf` 时也会显示帮助信息。

仅可以在全局区域中使用此选项。

下列选项用于修饰 `scconf` 命令的基本格式和功能。不能在同一命令行上组合使用这些选项。

`-a`

指定 `scconf` 命令的 `add` 格式。

仅可以在全局区域中使用此选项。

使用 `-a` 选项可以添加或初始化用于定义 Oracle Solaris Cluster 的软件配置的大多数项。将附加选项（例如，适配器、交换机或设备组选项）与 `-a` 一起使用可以指定要添加的元素及其关联属性。可以在同一命令行上组合使用任意数目的附加选项，只要这些附加选项是可以与 `-a` 选项一起使用的。

`-c`

指定 `scconf` 命令的 `change` 格式。

仅可以在全局区域中使用此选项。

使用 `-c` 选项可以更改已配置为 Oracle Solaris Cluster 软件配置一部分的项的属性。将附加选项与 `-c` 一起使用可以指定新属性或更改的属性。可以在同一命令行上组合使用任意数目的附加选项，只要这些附加选项是可以与 `-c` 选项一起使用的。

`-p`

指定 `scconf` 命令的 `print` 格式。

仅可以在全局区域中使用此选项。

`-p` 选项显示您可以使用 `scconf` 配置的当前 Oracle Solaris Cluster 配置元素及其关联属性的列表。可将此选项与一个或多个 `-v` 选项组合使用，以显示更详细的列表。

`-r`

指定 `scconf` 命令的 `remove` 格式。

仅可以在全局区域中使用此选项。

使用 `-r` 选项可以从 Oracle Solaris Cluster 软件配置中删除项。将附加选项与 `-r` 一起使用可以指定要从配置中删除的项。可以在同一命令行上组合使用任意数目的附加选项，只要这些附加选项是可以与 `-r` 选项一起使用的。

附加选项

可将下列附加选项与前面所述的一个或多个基本选项组合使用。请参阅“用法概要”部分，以查看可与 `scconf` 的每种格式一起使用的选项。

附加选项如下：

`-A adapter_options`

添加、删除或更改群集传输适配器的属性。要使这些操作成功，给定适配器所在的节点不需要在群集中处于活动状态。下面描述了接受 `-A` 的该命令三种格式使用 `-A adapter_options` 的方式。

- 使用以下语法为命令的 `add` 格式指定 `-A adapter_options`：

```
-A name=adaptername,node=
node[,vlanid=vlanid][,state=
state] \
[,other_options]
```

- 使用以下语法为命令的 `change` 格式指定 `-A adapter_options`：

```
-A name=adaptername,node=
node[,state=state] \
[,other_options]
```

- 使用以下语法为命令的 `remove` 格式指定 `-A adapter_options`：

```
-A name=name,node=node
```

`-A` 选项支持以下子选项：

`name=adaptername`

指定特定节点上某个适配器的名称。每次指定 `-A` 选项时，都必须包括该子选项。

`adaptername` 是基于 `device name` 构造的，后面紧接一个 `physical-unit` 编号（例如 `hme0`）。

`node=node`

指定特定节点上某个适配器的名称。每次指定 `-A` 选项时，`node` 子选项都是必需的。

可以使用节点名称或节点 ID 指定 `node`。

`state=state`

更改适配器的状态。可以将该子选项与命令的 `change` 格式一起使用。`state` 可设置为 `enabled` 或 `disabled`。

将适配器添加到配置后，其状态始终设置为 `disabled`。默认情况下，将电缆添加到适配器上的任一端口会将端口和适配器的状态都更改为 `enabled`。请参见 `-m cable_options`。

禁用一个适配器还会禁用与该适配器关联的所有端口。但是，启用一个适配器不会启用该适配器的端口。要启用某个适配器端口，必须启用该端口连接到的电缆。

trtype=type

指定传输类型。-A 与命令的 *add* 格式一起使用时，必须包括该子选项。

dlpi 就是一个传输 *type* 示例。请参见 [sctransp_dlpi\(7p\) \[1279\]](#)。

[*vlanid=vlanid*]

指定带标记的 VLAN 适配器的 VLAN ID。

[*other_options*]

如果有其他选项适用于特定的适配器类型，则可以在命令的 *add* 和 *change* 格式中与 -A 一起使用这些选项。有关特殊选项的信息，请参阅群集传输适配器手册页。

要将该命令选项与 -a、-c 或 -r 一起使用，需要具有 `solaris.cluster.transport.modify RBAC` 授权。请参见 [Unresolved link to "rbac5"](#)。

-B *switch_options*

添加、删除或更改群集传输交换机（又称传输结点）的属性。

这种设备的示例包括但不限于以太网集线器、其他各种类型的交换机和环网。

下面描述了接受 -B 的三种命令格式使用 -B *switch_options* 的方式。

- 使用以下语法为命令的 *add* 格式指定 -B *switch_options*：

```
-B type=type,name=name  
[,other_options]
```

- 使用以下语法为命令的 *change* 格式指定 -B *switch_options*：

```
-B name=name[,state=state]  
[,other_options]
```

- 使用以下语法为命令的 *remove* 格式指定 -B *switch_options*：

```
-B name=name
```

-B 选项支持以下子选项：

name=name

指定群集传输交换机的名称。每次指定 -B 选项时，都必须包括 *name* 子选项。*name* 的长度为最多 256 个字符。它由字母或数字组成，第一个字符为字母。每个传输交换机名称在群集名称空间中必须唯一。

`state=State`

更改群集传输交换机的状态。该子选项可以与 `-B change` 命令一起使用。state 可设置为 `enabled` 或 `disabled`。

将交换机添加到配置后，其状态始终设置为 `disabled`。默认情况下，将电缆添加到交换机上的任一端口会将端口和交换机的状态都更改为 `enabled`。请参见 `-m cable_options`。

禁用一个交换机还会禁用与该交换机关联的所有端口。但是，启用一个交换机不会启用该交换机的端口。要启用某个交换机端口，必须启用该端口连接到的电缆。

`type=type`

指定群集传输交换机类型。`-B` 与命令的 `add` 格式一起使用时，必须包括该子选项。

以太网集线器就是 `switch` 类型的群集传输交换机的示例。[sconf_transp_jct_etherswitch\(1M\) \[673\]](#) 手册页中提供了更多信息。

[*other_options*]

如果有其他选项适用于特定的交换机类型，则可以在命令的 `add` 和 `change` 格式中与 `-B` 一起使用这些选项。有关特殊选项的信息，请参阅 [sconf_transp_jct_etherswitch\(1M\) \[673\]](#) 群集传输交换机手册页。

要将该命令选项与 `-a`、`-c` 或 `-r` 一起使用，需要具有 `solaris.cluster.transport.modify RBAC` 授权。请参见 [Unresolved link to "rbac5"](#)。

`-c cluster_options`

更改群集自身的名称。该选项只能与命令的 `change` 格式一起使用。

按以下方式为命令的 `change` 格式指定 `cluster_options`：

`-C cluster=clustername`

此命令格式会将群集的名称更改为 `clustername`。

`-D devicegroup_options`

向配置添加设备组，更改或重设现有设备组的属性，或者从 Oracle Solaris Cluster 设备组配置中删除组。其他设备组选项 (*other_options*) 在添加或更改设备组及其选项方面发挥着重要作用。配置任何设备组时，请特别注意与类型相关的设备组选项的手册页（例如 [sconf_dg_svm\(1M\) \[659\]](#) 和 [sconf_dg_rawdisk\(1M\) \[655\]](#)）。并非所有设备组类型都支持 `-D` 选项的所有三种格式。例如，`svm` 设备组通常只能与命令的 `change` 格式一起使用来更改特定属性，例如节点优先级列表的顺序。

命令的 `add` 格式可用于创建设备组，或者向现有设备组添加节点。对于某些设备组类型，`add` 格式还可用于向组中添加设备。命令的 `change` 格式可以注册更新，以更改与某个组关联的特定属性。命令的 `remove` 格式可用于删除整个设备组，或者删除组的一个或多个组件。

接受 `-D` 的三种 `scconf` 命令格式使用 `-D devicegroup_options` 的方式如下：

添加：

```
-D type=type,name=name[,nodelist=node[:node...]]
    [,preferenced={true | false}]
    [,numsecondaries=integer]
    [,failback={enabled | disabled}][,other_options]
```

更改：

```
-D name=name[,nodelist=node[:node...]]
    [,preferenced={true | false}]
    [,numsecondaries=integer]
    [,failback={enabled | disabled}][,other_options]
```

删除：

```
-D name=name,nodelist=node[:node...]
```

`-D` 选项支持以下子选项：

`name=name`

设备组的名称。使用命令的所有三种格式时都必须提供该名称。

`nodelist=node[: node]...`

潜在主节点的列表，将组添加到群集时，某些设备组类型需要该列表。有关更多信息，请参阅与类型相关的设备组的手册页。

将 `preferenced` 子选项设置为 `true` 时，`nodelist` 子选项是必需的。

使用命令的 `add` 格式时，默认情况下，`nodelist` 是一个有序列表，指明了节点尝试作为设备组的主节点进行接管时应遵循的优先顺序。但是，如果 `preferenced` 子选项设置为 `false`（请参见下一子部分），则访问组中设备的第一个节点自动成为该组的主节点。向现有设备组添加节点时，不能使用 `preferenced` 子选项。但是，首次创建组时可以使用 `preferenced` 子选项，或者将其与命令的 `change` 格式一起使用。

要更改主节点顺序优先级，必须在 `nodelist` 中按所需顺序指定群集节点的完整列表。还必须将 `preferenced` 子选项设置为 `true`。

与命令的 `remove` 格式一起使用时，`nodelist` 子选项可用于从设备组中删除指定的节点。只要不提供 `nodelist`，就可以删除整个设备组。单单从一个设备组中删除所有节点并不一定会删除该组。

`type=type`

设备组的类型。要指明需要创建的设备组的类型（例如 `rawdisk`），必须将该类型与命令的 `add` 格式一起使用。

[failback={enabled | disabled}]

与命令的 add 或 change 格式一起使用时，可启用或禁用设备组的 failback 行为。

指定当设备组主节点脱离群集并在随后返回时系统的行为。

当节点脱离群集时，设备组将故障转移到辅助节点。当发生故障的节点重新加入群集时，设备组可继续由辅助节点控制，也可以故障恢复到原始主节点。

如果 failback 为 enabled，则设备组将变为由原始主节点控制。如果 failback 为 disabled，则设备组将继续由辅助节点控制。

failback 默认为 disabled。

[numsecondaries=*integer*]

使您可以根据需要动态更改设备组的辅助节点数。设备组是一种 HA 服务，它要求一个节点充当主节点，一个或多个节点充当辅助节点。如果当前主节点发生故障，设备组的辅助节点能够接管并充当主节点。

该整数应大于 0，但小于指定组中的节点总数。默认值为 1。

系统管理员可以使用 numsecondaries 子选项来更改设备组的辅助节点数，同时保持指定的可用性级别。如果设备组中的某个节点已从辅助节点列表中删除，则将它重新转换为辅助节点之前，它无法接管并充当主节点。更改辅助节点数之前，需要评估此操作对辅助全局文件系统的影响。

numsecondaries 子选项只应用于设备组中当前处于群集模式的节点，并可与节点的 preferenced 子选项一起使用。如果启用了设备的 preferenced 子选项，则首先会从辅助节点列表中删除优先级最低的节点。如果设备组中没有任何节点标记为首选的，群集会随机选择节点进行删除。

如果由于节点故障，设备组的实际辅助节点数减少到所需级别以下，则会将从前从辅助节点列表中删除的节点添加回辅助节点列表，前提是这些节点当前位于群集中，属于该设备组，并且当前不是主节点或辅助节点。最先会转换设备组中优先级最高的节点，一直转换到与所需辅助节点数相符。

如果设备组中某个节点的优先级高于现有辅助节点，并加入了群集，那么，优先级最低的节点将从辅助节点列表中删除，并由新添加的节点取代。仅当实际辅助节点数超过所需级别时，才会发生这种取代。

要将所需辅助节点数设置为系统默认值（您不必知道该默认值），请发出下列命令之一：

```
# scconf -aD type=svm,name=foo, \  
nodelist=node1:node2,numsecondaries=
```

或

```
# scconf -cD name=foo,numsecondaries=
```

在创建设备组时，numsecondaries 子选项只能与 -a 选项一起使用。要向现有设备组添加主机，不能将 numsecondaries 子选项与 -a 选项一起使用。

[preferenced={true | false}]

指示设备组潜在主节点的优先顺序状态。只要 `preferenced` 子选项未设置为 `false`，新建设备组的节点列表就会按照节点尝试作为设备组主节点进行接管时遵循的顺序来指明优先顺序。

如果将 `preferenced` 子选项设置为 `true`，则还必须使用 `odelist` 子选项指定整个节点列表。

如果 `preferenced` 子选项没有与用于创建设备组的 `add` 一起指定，则它在默认情况下设置为 `false`。但是，如果 `preferenced` 子选项没有与 `change` 一起指定，则在指定了 `odelist` 时它在默认情况下设置为 `true`。

`preferenced` 子选项不能与用于向已建立的设备组添加节点的 `add` 格式一起使用。在此情况下，将使用已建立的节点优先级列表设置。

[other_options]

可以将其他与设备组类型相关的选项与命令的 `add` 或 `change` 格式一起使用。有关更多信息，请参阅相应的手册页（例如 [scconf_dg_svm\(1M\) \[659\]](#) 和 [scconf_dg_rawdisk\(1M\) \[655\]](#)）。

要将该命令选项与 `-a`、`-c` 或 `-r` 一起使用，需要具有 `solaris.cluster.device.modify` RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

-h node_options

在群集配置数据库中添加或删除一个节点。与 `scconf` 的 `add` 格式一起使用时，将在群集配置数据库中添加新名称以及内部生成的节点 ID。此外，将为新节点分配一个磁盘预留键，以及一个值为零的法定选票计数。为通过群集互连访问节点而分配的名称将初始化为 `clusternodenodeid-priv`。请参见 `-p` 选项，以了解有关显示配置元素及其关联属性的更多信息。

`scconf` 不能单独用来向群集添加新节点。`scconf` 只可用于更新配置数据库本身。`scconf` 不会将配置数据库复制到新节点或者在新节点上创建必要的节点标识符。要向群集添加节点，请使用 [scinstall\(1M\) \[697\]](#)。

与 `scconf` 的 `remove` 格式一起使用时，必须先删除对节点的所有引用（包括最后一条传输电缆）、所有资源组引用以及所有设备组引用，然后才能使用 `scconf` 从群集配置中彻底删除该节点。

要删除的节点不能是为任何法定设备而配置的。此外，不能从三节点群集中删除某个节点，除非至少配置了一个共享的法定设备。

Oracle Solaris Cluster 文档中的系统管理过程更详细地介绍了如何删除群集节点。

每次指定 `-h` 选项都必须指定 `node=node` 子选项。对于命令的 `add` 格式，指定的 `node` 必须是节点名称。

使用以下语法为命令的 `add` 格式指定 `-hnode_options`：

```
-h node=nodename
```

对于命令的 `remove` 格式，可以使用节点名称或节点 ID 指定 `node`。使用以下语法为命令的 `remove` 格式指定 `-h node_options`：

`-h node=node`

要将该命令选项与 `-a`、`-c` 或 `-r` 一起使用，需要具有 `solaris.cluster.node.modify RBAC` 授权。请参见 [Unresolved link to "rbac5"](#)。

`-m cable_options`

帮助建立群集互连拓扑。该选项的作用是可以配置用于连接群集传输适配器和交换机上的各个端口的电缆。每条新电缆通常可以映射两个群集传输适配器之间的连接，或者映射某个适配器与某个传输交换机端口之间的连接。接受 `-m` 的该命令的每种格式使用 `-m cable_options` 的方式如下：

- 使用以下语法为命令的 `add` 格式指定 `-m cable_options`：

```
-m endpoint=[node:]  
name[@port],  
    endpoint=[node:]name[@  
port][,noenable]
```

- 使用以下语法为命令的 `change` 格式指定 `-m cable_options`：

```
-m endpoint=[node:]  
name[@port],state=  
state
```

- 使用以下语法为命令的 `remove` 格式指定 `-m cable_options`：

```
-m endpoint=[node:]  
name[@port]
```

`-m` 选项支持以下子选项：

`endpoint=[node:] name[@port]`

每次指定 `-m` 选项时都必须包括该选项。对于命令的 `add` 格式，必须指定两个 `endpoint` 选项。选项参数的 `name` 组件用于指定电缆某一个端点上的群集传输适配器或群集传输交换机的名称。如果指定了 `node` 组成部分，则 `name` 是群集传输适配器的名称。否则，`name` 是群集传输交换机的名称。

如果未指定 `port` 组件，则会尝试采用默认端口名称。适配器的默认端口始终是 `0`。交换机端点的默认端口名称等于电缆另一端连接的节点的节点 ID。有关 `port` 分配和其他要求的更多信息，请参阅群集传输适配器和群集传输交换机手册页（例如 [sconf_transp_jct_etherswitch\(1M\) \[673\]](#)）。添加电缆之前，必须已配置电缆两个端点上的适配器和交换机（参见 `-A` 和 `-B`）。

`noenable`

在将电缆添加到配置时可以使用。默认情况下，当您添加一条电缆时，该电缆及其连接到的两个端口，以及这些端口所在的适配器或交换机的状态都将设置为 `enable`。但是，如果在添加电缆时指定了 `noenable`，则添加的电缆及其两个端点将处于禁用状态。端口所在的适配器或交换机的状态保持不变。

`state=State`

更改电缆及其连接到的两个端点的状态。启用某条电缆后，该电缆及其两个端口，以及与这两个端口关联的适配器或交换机全部都被启用。但是，如果禁用某条电缆，只会禁用该电缆及其两个端口。与这两个端口关联的适配器或交换机的状态保持不变。默认情况下，将电缆添加到配置中时，该电缆及其端点的状态始终设置为 `enabled`。要添加处于 `disabled` 状态的电缆，请在添加操作过程中使用 `noenable` 子选项。

要将该命令选项与 `-a`、`-c` 或 `-r` 一起使用，需要具有 `solaris.cluster.transport.modify RBAC` 授权。请参见 [Unresolved link to "rbac5"](#)。

`-P privatehostname_options`

对于节点，该选项可添加或更改专用主机名。

与命令的 `add (-a)` 格式一起使用时，`-P` 选项将指定下列操作之一：

- 当指定了某个节点时，该命令将会分配指定的主机名别名，以用于通过专用群集互连或传输对指定的节点进行 IP 访问。如果未分配或者已重置主机名别名，则节点的默认专用主机名为 `clusternodeid-priv`。
- 该主机名必须未被企业中的其他任何节点使用。

为群集配置的专用 IP 地址范围必须支持群集中使用的专用 IP 地址数目的增长。请确保专用 IP 地址范围能够支持所添加的专用 IP 地址，再分配其中某个专用 IP 地址。有关更多信息，请参见 [scprivipadm\(1M\) \[725\]](#) 手册页。

与命令的 `change (-c)` 格式一起使用时，`-P` 选项将更改指定节点的主机名别名。

绝不当将专用主机名存储在 [Unresolved link to "hosts4"](#) 数据库中。一个特殊的 `nsswitch` 工具（请参见 [Unresolved link to "nsswitch.conf4"](#)）可针对专用主机名执行所有主机名查找操作。

接受 `-P` 的该命令的每种格式使用 `privatehostname_options` 的方式如下：

添加：

```
-P node=node[,privatehostname=hostalias]
```

更改：

```
-P node=node[,privatehostname=hostalias]
```

删除：

```
-P node=node
```

`-P` 选项支持以下子选项：

`node=node`

提供要为其分配指定专用主机名或主机别名（通过 `privatehostname` 子选项提供）的节点的名称或 ID。

`privatehostname=hostalias`

提供用于通过专用群集互连或传输访问节点的主机别名。如果未指定 `privatehostname` 子选项，则将指定 `node` 的专用主机名重置为默认值。

要将该命令选项与 `-a`、`-c` 或 `-r` 一起使用，需要具有 `solaris.cluster.transport.modify RBAC` 授权。请参见 [Unresolved link to "rbac5"](#)。

`-q quorum_options`

管理共享群集法定设备和各个群集法定属性。请特别注意与类型相关的法定设备选项的手册页（例如 [scconf_quorum_dev_scsi\(1M\) \[665\]](#)）。



注意 - 使用基于控制器的复制的设备不能用作 Oracle Solaris Cluster 环境中的法定设备。如果使用 `-q` 选项指定了一个使用基于控制器的复制的设备，`scconf` 命令将返回错误。

命令的 `add` 和 `remove` 格式可在配置中添加和删除共享法定设备。命令的 `change` 格式可更改各个群集法定配置属性或状态。适用于这三种命令格式的 `-q quorum_options` 可用于更改群集法定配置，如下所述：

添加：

```
-q name=devicename,  
type={scsi}
```

For SCSI quorum devices only:

```
-q autoconfig[,noop]
```

更改：

```
-q node=node,{  
maintstate | reset}  
-q name=devicename,{  
maintstate | reset}  
-q reset  
-q installmode
```

For SCSI quorum devices only:

```
-q autoconfig[,noop]
```

删除：

```
-q name=devicename
```

如果在执行法定相关操作时 `scconf` 中断或发生故障，群集配置数据库中的法定配置信息可能会出现不一致。如果发生这种情况，再次运行相同的 `scconf` 命令或使用 `reset` 子选项运行该命令以重置法定信息。

`-q` 选项支持以下子选项：

autoconfig

与命令的 `add` 格式一起使用时，在双节点群集中会自动选择并分配一个法定设备。该法定设备是从可用设备中选择的。如果已配置了一个法定设备，该命令将会中止。

与命令的 `change` 格式一起使用时，在双节点群集中会自动选择并分配一个设备，用于取代所有现有法定设备。该法定设备是从可用设备中选择的。

群集中的所有可用设备都必须符合成为法定设备的条件。autoconfig 子选项不会评估某个可用设备是否符合成为法定设备的条件。

如果群集包含两个以上的节点，autoconfig 子选项将不会更改法定配置。如果计划将某个 NAS 设备配置为法定设备，请不要使用 autoconfig 子选项。

installmode

强制群集返回到安装模式。在 `installmode` 下，节点在引导时不会尝试重置其法定配置。并且，在这种模式下，很多管理功能都被阻止。首次安装一个群集后，它是以 `installmode` 进行设置的。当所有节点首次加入该群集，并且共享法定设备已添加到配置中后，请发出 `scconf -c -q reset` 以将选票计数重置为默认值，并清除 `installmode` 设置。

name=devicename

指定在群集中添加或删除共享法定设备时要使用的已连接共享存储设备的名称。该子选项还可以与命令的 `change` 格式一起使用来更改法定设备的状态。

每个法定设备必须至少与群集中的两个节点建立网络连接或端口连接。不可以将非共享磁盘用作法定设备。

将 `scconf` 的 `change` 格式与 `-qname` 一起使用可以将设备置于维护状态，或者将设备的法定配置重置为默认值。在维护状态下，设备的选票计数为零，因此，不参与法定构成。重置为默认值后，设备的选票计数更改为 $N-1$ ，其中 N 是可与设备建立端口连接的、选票计数非零的节点的数目。

node=node

与命令的 `add` 格式一起使用时，选择要配置的应与所添加的共享法定设备建立端口连接的节点。该子选项还可以与命令的 `change` 格式一起使用来更改节点的法定状态。

将 `node` 子选项与法定更新命令的 `change` 格式一起使用时，该子选项用于将节点置于维护状态或者将节点的法定配置重置为默认值。

将节点置于维护状态之前，必须先关闭该节点。如果尝试将某个群集成员置于维护状态，`scconf` 将返回错误。

在维护状态下，节点的选票计数为零，因此，不参与法定构成。此外，配置为与节点建立端口连接的任何共享法定设备的选票计数将会减 1，以反映节点的新状态。节点重置为默认值后，其选票计数将重置为 1，而共享法定设备的选票计数将重新往上调整。除非群集处于 `installmode` 模式，否则，在引导时将自动重置每个节点的法定配置。

可以使用节点名称或节点 ID 指定 `node`。

`type=type`

与命令的 `add` 格式一起使用时，指定要创建的法定设备的类型。

`scsi`

指定共享磁盘法定设备。有关特定于 SCSI 类型的选项，请参见 [`scconf_quorum_dev_scsi\(1M\)` \[665\]](#)。

`{maintstate}`

当与命令的 `change` 格式一起使用并用作 `globaldev` 或 `node` 子选项的标志时，可将共享的法定设备或节点置于法定维护状态。在维护状态下，共享的设备或节点不再参与法定构成。当您需要长时间关闭某个节点或设备以进行维护时，此功能十分有用。当某个节点重新引导回群集后，在正常情况下，该节点会从维护模式中自行删除。

随同一 `-q` 选项同时指定 `maintstate` 和 `reset` 是非法的。

`[,noop]`

与 `autoconfig` 子选项一起使用时有效。该命令在标准输出中显示将由 `autoconfig` 子选项添加或更改的法定设备的列表。`autoconfig,noop` 子选项不会更改法定配置。

`{reset}`

当随命令的 `change` 格式用作一个标志时，可重置共享法定设备或共享节点的已配置法定选票。该选项可以与 `globaldev` 或 `node` 子选项结合使用，也可以作为自身的子选项。

如果单独使用，则会将整个法定配置重置为默认的选票设置。此外，如果设置了 `installmode`，则会通过全局法定配置重置来将其清除。除非已成功配置了至少一个共享法定设备，否则无法在双节点群集上重置 `installmode`。

otheroptions

可以使用其他特定于法定设备类型的选项。有关详细信息，请参见 [`scconf_quorum_dev_scsi\(1M\)` \[665\]](#)。

要将该命令选项与 `-a`、`-c` 或 `-r` 一起使用，需要具有 `solaris.cluster.quorum.modify` RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

`-s slm_options`

与 `sconf` 命令的 `change` 格式一起使用时，可设置属性以配置系统资源控制。如果未向这些属性分配值，这些属性将自动设置为默认值。

`-s` 选项的语法为：

```
-S [node=node] \  
[,globalzonestores=integer] \  
[,defaultpsetmin=integer]
```

`-s` 选项支持以下子选项：

`globalzonestshares=globalzonestshares`

设置分配给全局区域的份额数。`globalzonestshares` 的下限为 1，上限为 65,535。要了解该上限，请参见 [Unresolved link to "prctl1"](#) 手册页中有关 `zone.cpu-shares` 属性的信息。`globalzonestshares` 的默认值为 1。在正在运行的群集上，如果不再存在任何使用全局区域中配置的 CPU 控制的联机资源组，则分配给全局区域的 CPU 份额数将设置为 `globalzonestshares` 的值。

`defaultpsetmin=defaultpsetmin`

设置默认处理器集中可用的最小 CPU 数。默认值为 1。`defaultpsetmin` 的最小值为 1。Oracle Solaris Cluster 将在可用 CPU 数的限值内分配 CPU 数目，该数目尽可能地接近您为 `defaultpsetmin` 设置的数目。如果分配的数目小于您请求的数目，Oracle Solaris Cluster 将定期尝试分配您请求的 CPU 数目。此操作可能会销毁某些 `dedicated_weak` 处理器集。有关 `dedicated_weak` 处理器集的信息，请参见 [scrgadm\(1M\) \[731\]](#) 手册页。

`node=node`

标识要设置其属性的节点。通过指定节点的名称，在您希望从 CPU 控制受益的每个节点上设置这些属性。每次使用 `-s` 选项时，只能指定一个节点。

要将该命令选项与 `-c` 一起使用，需要具有 `solaris.cluster.node.modify` RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

`-T authentication_options`

为尝试将自身添加到群集配置的节点建立验证策略。具体而言，就是当某台计算机请求将自身作为群集节点添加到群集中时（参见 [scinstall\(1M\) \[697\]](#)），执行检查以确定该节点是否有权加入。如果该节点具有权限，将对该正在加入的节点进行验证。默认情况下，允许任何计算机将自身加入群集。

接受 `-T` 的三种命令格式使用 `-T authentication_options` 的方式如下：

添加：

```
-T node=nodename[,...][,authtype=authtype]
```

更改：

```
-T authtype=authtype
```

删除：

```
-T {node=nodename[,...] | all}
```

`-T` 选项支持以下子选项：

`all`

可以通过指定 `scconf -r -T all` 来清除包含所有节点名称的列表。清除验证列表意味着任何节点都可以在群集中安装和配置自身。

`node=nodename`

在可以将自身安装和配置为群集中节点的节点列表中添加或删除主机名。命令的 `add` 格式至少需要一个 `node` 子选项，该子选项对于 `remove` 格式是可选的。如果验证列表为空，任何主机都可以请求将自身添加到群集配置。但是，如果列表至少包含一个名称，将会使用验证列表验证所有这种请求。

可接受非法的 `nodename`，包括以点 (.) 表示的节点名称。如果将以 . 表示的 `nodename` 添加到验证列表，则点字符是一个特殊字符，其他所有名称将被删除。此功能可阻止主机尝试在群集中安装和配置自身。

`authtype=authtype`

与命令的 `add` 或 `change` 格式一起使用。

当前支持的验证类型 (`authtype`) 只有 `des` 和 `sys` (或 `unix`)。默认的验证类型为提供最少的安全验证的 `sys`。

如果使用 `des` (或 Diffie-Hellman) 验证，则应该在实际运行 `scinstall` 命令添加节点之前，将要添加的每个群集节点的条目添加到 `publickey` 数据库中。

要将该命令选项与 `-a`、`-c` 或 `-r` 一起使用，需要具有 `solaris.cluster.node.modify` RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

`-v`

与 `-p` 选项一起使用时，可请求更详细的群集配置列表。如果与其他选项一起使用，在遇到错误时可能会显示更多信息。

要将该命令选项与 `-p` 一起使用，需要具有

`solaris.cluster.device.read`、`solaris.cluster.transport.read`、`solaris.cluster.resource.read`、`solaris.cluster.system.read` RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

`-w heartbeat_options`

更改某个群集的全局心跳参数，这可以有效地更改群集的所有适配器的心跳参数。

Oracle Solaris Cluster 依赖经由专用互连的心跳来检测群集节点之间的通信故障。减小心跳超时可使 Oracle Solaris Cluster 更快地检测故障，因为减小心跳超时时，检测故障所需的时间更少。这样，Oracle Solaris Cluster 在发生故障后可以更快地恢复，从而提高群集可用性。

`-w` 选项支持以下子选项：

`heartbeat_quantum=quantum_milliseconds`

定义发送心跳的频率。默认情况下，Oracle Solaris Cluster 使用 1 秒 (1,000 毫秒) 心跳量程。指定一个介于 100 和 10,000 毫秒之间的值。

`heartbeat_timeout=timeout_milliseconds`

一个时间间隔，经过该时间间隔后，如果未从对等节点收到心跳，则将相应的路径声明为已关闭。默认情况下，Oracle Solaris Cluster 使用 10 秒 (10,000 毫秒) 心跳超时。指定一个介于 2,500 和 60,000 毫秒之间的值。

注 - 如果使用 `-w` 减小心跳参数值，则即使在理想的条件下，也总是会存在路径虚假超时风险并且可能会发生节点紧急情况。始终测试并全面将心跳参数的较低值限定在相关工作负载条件之下后，才能在群集中真正实施这些值。

使用 `-w` 选项，一次只能更改一个心跳子选项。减小心跳参数的值时，请先更改 `heartbeat_quantum`，然后更改 `heartbeat_timeout`。增大心跳参数的值时，请先更改 `heartbeat_timeout`，然后更改 `heartbeat_quantum`。

注 - 为 `heartbeat_timeout` 指定的值必须总是大于或等于为 `heartbeat_quantum` 指定的值的五倍 (`heartbeat_timeout >= (5 * heartbeat_quantum)`)。

您需要具有 `solaris.cluster.system.modify` RBAC 授权才能使用 `-w`。请参见 [Unresolved link to "rbac5"](#)。

例 301 减小心跳

以下示例说明了如何将心跳量程从 Oracle Solaris Cluster 默认值 1,000 毫秒减小为 100 毫秒。该示例还说明了如何将心跳超时从 Oracle Solaris Cluster 默认值 10,000 毫秒减小为 2,500 毫秒。

```
phys-schost-1# scconf -c -w heartbeat_quantum=100
phys-schost-1# scconf -c -w heartbeat_timeout=2500
```

由于 `heartbeat_timeout` 必须总是大于或等于 `heartbeat_quantum` 的五倍，因此首先需要设置 `heartbeat_quantum`。否则将不符合要求。换言之，如果 `heartbeat_quantum` 当前设置为默认值 1,000 毫秒，而您将 `heartbeat_timeout` 设置成了 2,500 毫秒，那么，`heartbeat_timeout` 就小于 `heartbeat_quantum` 的五倍。`scconf` 命令因此将会失败。

但是，将 `heartbeat_quantum` 设置为适当的值后，如果要求尚未满足，还可以将 `heartbeat_timeout` 设置为更小的值。

例 302 增大心跳

以下示例说明了如何将心跳超时和心跳量程参数从您在前一示例中设置的值增大到 Oracle Solaris Cluster 默认值。

```
phys-schost-1# scconf -c -w heartbeat_timeout=10000
phys-schost-1# scconf -c -w heartbeat_quantum=1000
```

先设置 `heartbeat_timeout`，以满足 `heartbeat_timeout` 总是大于或等于 `heartbeat_quantum` 的五倍这一要求。将 `heartbeat_timeout` 设置为所需值后，可以将 `heartbeat_quantum` 设置为更大的新值。

例 303 典型的安装后设置操作

下列命令提供了您可能会在新的双节点群集上执行的一组典型的安装后设置操作示例。这些命令将会向群集添加一个共享法定设备，清除 `installmode`，配置另一组群集传输连接，并阻止其他计算机尝试将自身添加到该群集：

```
phys-red# scconf -a -q globaldev=d0
phys-red# scconf -c -q reset
phys-red# scconf -a \
  -A trtype=dlpi,name=hme1,node=phys-red \
  -A trtype=dlpi,name=hme1,node=phys-green \
  -m endpoint=phys-red:hme1,endpoint=phys-green:hme1
phys-red# scconf -a -T node=.
```

返回以下退出值：

0 命令已成功完成。
非零值 发生错误。

有关下列属性的说明，请参见 [Unresolved link to " attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	过时

[Intro\(1CL\) \[17\]](#)、[cldevice\(1CL\) \[51\]](#)、[scconf_dg_rawdisk\(1M\) \[655\]](#)、[scconf_dg_svm\(1M\) \[659\]](#)、[scconf_q link to " hosts4"](#)、[Unresolved link to " nsswitch.conf4"](#)、[Unresolved link to " publickey4"](#)、[Unresolved link to " attributes5"](#)、[sctransp_dlpi\(7p\) \[1279\]](#)

仅当群集中的所有节点均在运行时才使用 `-w` 选项。如果群集中有任一节点关闭，请不要使用 `-w`。否则，节点可能会挂起或发生紧急情况。

包含一个或多个单 CPU 节点或者包含八个以上节点的群集如果以较小的心跳参数值运行，将更有可能会遇到超时和节点紧急情况。

注 - 如果使用 `-w` 减小心跳参数值，则即使在理想的条件下，也总是会存在路径虚假超时风险并且可能会发生节点紧急情况。始终测试并全面将心跳参数的较低值限定在相关工作负载条件之下后，才能在群集中真正实施这些值。

使用 `scconf` 更改配置后，应该备份每个节点上的根文件系统，或者保留所有更改的日志。如果需要恢复在正常系统备份之间的配置更改，可以使用日志来返回到最近的配置。

随 `scconf` 命令指定的选项列表始终按照您在命令行上指定它们的顺序执行。但是，只要可能，`scconf` 就会根据群集配置数据库将某些传输选项（`-A`、`-B` 和 `-m`）作为单个事务进行处理。请尽量在单个命令行上将此类型的所有相关选项组合在一起，以减少群集的开销。

名称

scconf_dg_rawdisk — 添加、更改或更新原始磁盘设备组配置

```
scconf -a -D type=rawdisk, [generic_options] [,globaldev=gdev1,globaldev=gdev1,...] [,localonly=true]

scconf -a -D type=rawdisk, [generic_options] [,globaldev=gdev1,globaldev=gdev1,...] [,localonly=true | false]

scconf -c -D name=diskgroup,autogen=true

scconf -r -D device_service_name [,nodelist=node[:node]...]
[,globaldev=gdev1,...]
```

注 - Oracle Solaris Cluster 软件包括面向对象的命令集。虽然 Oracle Solaris Cluster 软件仍支持原始的命令集，但是 Oracle Solaris Cluster 过程文档仅使用面向对象的命令集。有关面向对象的命令集的更多信息，请参见 [Intro\(1CL\) \[17\]](#) 手册页。

以下信息针对的是 scconf 命令。要使用等效的面向对象的命令，请参见 [cldevicegroup\(1CL\) \[65\]](#) 手册页。

scconf_dg_rawdisk 实用程序用于添加、更改或更新原始磁盘设备组配置。

原始磁盘是指不会将其用作卷管理器卷或元设备一部分的磁盘。原始磁盘设备组允许您在磁盘设备组内定义一组磁盘。

在系统引导时，默认情况下，系统会为配置中的每个磁盘 ID 伪驱动程序 (DID) 设备创建原始磁盘设备组。依据约定，原始磁盘设备组名称是在初始化时分配的，而且是从 DID 名称中衍生的。对于添加到原始磁盘设备组的每个节点，scconf 实用程序会验证设备组中的每个设备是否已通过物理方式连接到节点。

scconf -a (添加) 命令可用于创建在其中配置了多个磁盘设备的原始磁盘设备组。引导时，会为群集中的每个磁盘设备创建原始磁盘设备组。

在添加新的原始磁盘设备组之前，必须先从引导时创建的设备组中移除要在新组中使用的设备。然后，才能创建包含这些设备的新原始磁盘设备组。通过在 scconf 的 globaldev 选项中指定这些设备的列表，并在 nodelist 选项中指定潜在主节点首选项列表，可实现此操作。如果设备组已存在，将只添加新的节点和全局设备，同时会忽略已成为现有设备组一部分的节点或设备。

如果 preferenced 子选项没有与用于创建新设备组的 -a 选项一起指定，则它在默认情况下设置为 false。但是，如果为现有设备组指定了值为 true 或 false 的 preferenced 子选项，则会返回错误。系统这样设置是为了维护现有的节点列表首选项状态。

如果设备组只能由特定节点进行控制，则在配置时应将 `otheroption` 设置为 `localonly=true`。在节点列表中只能指定一个节点来创建 `localonly` 设备组。

`scconf -c` (更改) 命令用于更改潜在主节点首选项的顺序、启用或禁用故障恢复、设置所需的辅助节点数以及将更多全局设备添加到设备组。

如果要更改节点首选项列表的顺序，则必须在 `nodelist` 中指定当前存在于设备组中的所有节点。此外，如果更改节点首选项的顺序，则还必须将 `preferenced` 子选项设置为 `true`。

如果在更改时未指定 `preferenced` 子选项，则使用已建立的 `true` 或 `false` 设置。

新节点无法使用命令的更改格式来添加。更改选项还可用于将设备组更改为 `localonly` 设备组，反之亦然。要将设备组更改为 `localonly` 设备组，请将 `otheroption` 设置为 `localonly=true`。指定 `localonly=false` 可将其设置回 `localonly` 设备组以外的设备组。必须已将 `nodelist` 设置为单个节点的列表，否则会导致错误。将 `localonly` 设置为 `true` 时，可以使用命令的 `change` 格式来指定 `nodelist`。然而这样做是多余的，因为该列表只能包含已配置的单个节点。指定已配置节点外的任何节点都将导致错误。

`scconf -r` (删除) 命令可用于删除节点、全局设备以及群集设备组配置中的设备组名称。如果节点或全局设备是通过设备组名称来指定的，则会将其从设备组中首先删除。将最后一个设备和节点从设备组中删除后，该设备组也会从群集配置中删除。如果只指定设备组名称（根本不指定节点或设备），则会删除整个设备组。

如果原始磁盘设备名称已在原始磁盘设备组中注册，该名称将无法在 Solaris Volume Manager 设备组中注册。

如需受支持的一般选项的列表，请参见 [scconf\(1M\) \[637\]](#) 手册页。

以下操作选项用于说明该命令所执行的操作。一个命令只允许一个操作选项。

支持以下操作选项：

- a 将新的原始磁盘设备组添加到群集配置中。您也可以使用该选项来更改设备组配置。
- c 更改节点首选项列表的顺序、更改首选项和故障恢复策略、更改所需的辅助节点数并使用 `globaldev` 选项将更多设备添加到设备组。还可用于将设备组设置为仅限本地使用。
- r 从群集删除原始磁盘设备组名称。
autogen 标志是 `scconf` 命令的指示符。该命令不会列出具有 `autogen` 属性的设备，除非使用 `-v` 命令行选项。通过 `scconf` 命令的更改格式来使用设备时，系统会重置设备的 `autogen` 属性或将其设置为 `false`，除非还指定了 `autogen=true`。

例 304 使用 `scconf` 命令

以下 `scconf` 命令用于创建原始磁盘设备组、更改潜在主节点的顺序、更改首选项和故障恢复策略、更改所需的辅助节点数以及从群集配置中删除原始磁盘设备组。

```
host1# scconf -a -D type=rawdisk,name=rawdisk_groupname,  
nodelist=host1:host2:host3,preferenced=false,failback=enabled,  
numsecondaries=,globaldev=d1,globaldev=d2
```

```
host1# scconf -a -D type=rawdisk,name=rawdisk_groupname,  
nodelist=host1,globaldev=d1,globaldev=d2,localonly=true,  
globaldev=d1,globaldev=d2
```

```
host1# scconf -c -D name=rawdisk_groupname,  
nodelist=host3:host2:host1,preferenced=true,failback=disabled,  
numsecondaries=2,globaldev=d4,globaldev=d5
```

```
host1# scconf -c -D name=rawdisk_groupname,localonly=true
```

```
host1# scconf -r -D name=rawdisk_groupname
```

```
host1# scconf -r -D name=rawdisk_groupname,nodelist=host1,host2
```

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
体系结构	SPARC
可用性	ha-cluster/system/core
接口稳定性	过时

[Intro\(1CL\) \[17\]](#)、[cldevicegroup\(1CL\) \[65\]](#)、[scconf\(1M\) \[637\]](#)、[Unresolved link to "attributes5"](#)

名称

scconf_dg_svm — 更改 Solaris Volume Manager 设备组配置。

```
scconf -c -D [generic_options]
```

注 - Oracle Solaris Cluster 软件包括面向对象的命令集。虽然 Oracle Solaris Cluster 软件仍支持原始的命令集，但是 Oracle Solaris Cluster 过程文档仅使用面向对象的命令集。有关面向对象的命令集的更多信息，请参见 [Intro\(1CL\) \[17\]](#) 手册页。

以下信息针对的是 `scconf` 命令。要使用等效的面向对象的命令，请参见 [cldevicegroup\(1CL\) \[65\]](#) 手册页。

Solaris Volume Manager 设备组可通过名称、节点（可在其上对该组进行访问）、磁盘集中各设备的全局列表以及属性集（可用于控制诸如潜在主节点首选项和故障恢复行为之类的操作）来定义。

就 Solaris Volume Manager 设备组来说，只有一个磁盘集可分配给其中的一个设备组，且组名必须始终与磁盘集本身的名称相匹配。

在 Solaris Volume Manager 中，多主机设备或共享设备是指由两个或多个主机以及可由所有主机访问并在所有主机上具有相同设备名称的硬盘驱动器组成的组合。这一使用相同名称的设备命名要求是通过使用原始磁盘设备形成磁盘集来实现的。设备 ID 伪驱动程序 (DID) 允许多主机设备在群集中拥有一致的名称。只有已配置为磁盘集本身一部分的主机可配置到 Solaris Volume Manager 设备组的 `nodelist` 中。在将驱动器添加到共享磁盘集时，这些驱动器必须不属于任何其他共享磁盘集。

Solaris Volume Manager `metaset` 命令用于创建磁盘集，并且一开始就可以将其创建并将注册为 Solaris Volume Manager 设备组。接下来，必须使用 `scconf` 命令设置节点首选项列表、`preferenced`、`failback` 和 `numsecondaries` 子选项。

如果要更改节点首选项列表的顺序或故障恢复模式，您必须在 `nodelist` 中指定当前存在于设备组中的所有节点。此外，如果更改节点首选项的顺序，则还必须将 `preferenced` 子选项设置为 `true`。

如果在命令的 "change" 格式中未指定 `preferenced` 子选项，则使用已建立的 `true` 或 `false` 设置。

无法使用 `scconf` 命令从群集配置中删除 Solaris Volume Manager 设备组。请改用 Solaris Volume Manager 的 `metaset` 命令。可通过删除 Solaris Volume Manager 磁盘集删除设备组。

有关支持的一般选项的列表，请参见 [scconf\(1M\) \[637\]](#)。有关用来创建和删除磁盘集及设备组的 `metaset` 相关命令的列表，请参见 [Unresolved link to "metaset1M"](#)。

一个命令只允许一个操作选项。支持以下操作选项。

-c 更改节点首选项列表的顺序、更改首选项和故障恢复策略，以及更改所需的辅助节点数。

例 305 创建和注册磁盘集

以下 `metaset` 命令用于创建磁盘集 `diskset` 并将磁盘集注册为 Solaris Volume Manager 设备组。

其次，`scconf` 命令可用于指定设备组的潜在主节点的顺序、更改首选项和故障恢复选项，以及更改所需的辅助节点数。

```
host1# metaset -s diskset1 -a -h host1 host2
host1# scconf -c -D name=diskset1,nodeList=host2:host1,
preferred=true,failback=disabled,numsecondaries=1
```

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	过时

[Intro\(1CL\) \[17\]](#)、[cldevicegroup\(1CL\) \[65\]](#)、[scconf\(1M\) \[637\]](#)、[Unresolved link to "metaset1M"](#)

名称

scconf_quorum_dev_quorum_server — 添加、删除和配置法定服务器类型的法定设备。

```
scconf [-q quorum-options]
```

注 - Oracle Solaris Cluster 软件包括面向对象的命令集。虽然 Oracle Solaris Cluster 软件仍支持原始的命令集，但是 Oracle Solaris Cluster 过程文档仅使用面向对象的命令集。有关面向对象的命令集的更多信息，请参见 [Intro\(1CL\) \[17\]](#) 手册页。

Oracle Solaris Cluster 提供将法定服务器配置为法定设备的选项。该配置信息由设备名称（在法定设备内必须是唯一的）、运行法定服务器的主机的地址和法定服务器在其上侦听请求的端口号组成。如果群集需要多个法定设备，请配置多个法定服务器或将存储设备用于其他法定设备。对于一个群集，一个法定服务器仅可用作一个法定设备。

要将群集配置为使用法定服务器，必须安装和配置法定服务器软件，且该软件必须在可访问所有群集节点的计算机上运行。该命令在群集节点上运行时，法定服务器本身必须经过配置且在运行中。有关配置法定服务器的信息，请参见 [Unresolved link to "clquorumserver1CL"](#)。

配置法定服务器类型的法定设备需要以下参数。有关支持的一般选项的列表，请参见 [Unresolved link to "scconf\(1M\)"](#)。

使用命令的 `add` 和 `remove` 形式将共享法定设备添加到配置文件和从该配置文件中将其删除。使用命令的 `change` 形式更改各种群集法定配置属性或状态。以下法定服务器特定选项可用于更改群集法定配置：

添加法定服务器类型的法定设备

添加法定设备前：

- 法定服务器必须在法定服务器主机上运行。
- 必须在 `/etc/inet/hosts` 文件中输入法定服务器主机名。
- 必须设置法定服务器主机的网络掩码。

有关主机文件和网络掩码要求的信息，请参见 [Unresolved link to "Oracle Solaris Cluster 系统管理指南"](#) 中有关添加法定服务器法定设备的步骤。添加法定设备后，所有的参数都不可更改。

```
# scconf -q -a name=devicename,type=quorum_server,qshost=qhost,port=portnumber
```

```
name=devicename
```

法定服务器名称。在系统的所有法定设备中，该名称必须是唯一的。

`type=quorum_server`

表示要创建的磁盘设备组的类型。对于法定服务器类型的法定设备，该参数的值必须是 `quorum_server`。

`qhost=qhost`

网络中计算机的主机名，该计算机可由所有群集节点访问并且可运行法定服务器。取决于主机的 IPv4 或 IPv6 配置，该主机名必须在 `/etc/hosts` 文件、`/etc/inet/ipnodes` 文件或二者中具有一个条目。

`port=portnumber`

法定服务器侦听请求所在的端口。

注 - 如果需要更改法定服务器的端口号而保持相同主机名，请先删除法定设备，进行更改，然后重新添加法定设备。

更改法定服务器类型的法定设备的配置参数

```
# scconf -c -q name=devicename,maintstate | reset
```

如果必须更改其他参数（如 `qshost` 或 `port`），请使用新参数添加新的法定设备，然后删除现有法定设备。

删除法定服务器类型的法定设备

```
# scconf -q name=devicename
```

如果执行与法定相关的操作时，`scconf` 命令中断或失败，则在群集配置数据库中法定配置信息可能变得不一致。如果出现不一致的情况，再次运行相同的 `scconf` 命令或使用 `reset` 选项运行 `scconf` 命令以重置法定信息。

例 306 添加法定服务器类型的法定设备

以下 `scconf` 命令可用于添加其端口号配置为 9000 的法定服务器法定设备。

```
# scconf -q -a name=qd1,type=quorum_server,qshost=scclient1,port=9000
```

例 307 删除法定服务器类型的法定设备

以下 `scconf` 命令可用于删除名为 `qd1` 的法定服务器法定设备。

```
# scconf -r -q name=qd1
```

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
稳定性	过时

[Intro\(1CL\) \[17\]](#)、[clquorum\(1CL\) \[195\]](#)、[Unresolved link to "clquorumserver1CL"](#)、[cluster\(1CL\) \[461\]](#)、[Unresolved link to "sccnf\(1M\)"](#)、[Unresolved link to "gateways4"](#)、[Unresolved link to "hosts4"](#)

名称

`scconf_quorum_dev_scsi` — 添加和删除共享 SCSI 法定设备和更改各种 SCSI 群集法定配置属性或状态。

```
scconf {-a|-c|-r} -q globaldev=devicename otheroptions
```

```
scconf {-a|-c|-r} -q name=devicename otheroptions
```

注 - Oracle Solaris Cluster 软件包括面向对象的命令集。虽然 Oracle Solaris Cluster 软件仍支持原始的命令集，但是 Oracle Solaris Cluster 过程文档仅使用面向对象的命令集。有关面向对象的命令集的更多信息，请参见 [Intro\(1CL\) \[17\]](#) 手册页。

SCSI 法定设备被认为是任何 Oracle Solaris Cluster 支持的、连接到两个或更多个群集节点的连接存储器。设备必须由 DID 管理，并且提供的设备名称必须是 DID 设备名称。

SCSI 法定设备再没有其他可指定的属性了。

以下选项专用于共享磁盘法定设备。有关支持的一般选项的列表，请参见 [scconf\(1M\) \[637\]](#)。

命令的 `add` 和 `remove` 形式用于将共享法定设备添加到配置或从配置中将其删除。命令的 `change` 形式用于更改群集法定配置的各种属性。适用于这三种命令格式的 `-q quorum-options` 可用于更改群集法定配置，如下所述：

添加共享法定设备：

```
-q -a globaldev=devicename [,  
node=node ,node=  
node[, ...]]
```

或

```
-q -a name= devicename ,type=scsi
```

或

```
-q -a autoconfig[,noop]
```

更改法定配置的属性或状态：

```
-q -c globaldev=devicename ,{  
maintstate | reset}
```

或

```
-q -c autoconfig[,noop]
```

删除共享法定设备：

```
-q -r globaldev=devicename
```

或

```
-q -r name=devicename
```

autoconfig

与命令的 `add` 格式一起使用时，在双节点群集中会自动选择并分配一个法定设备。该法定设备是从可用设备中选择的。如果已配置了一个法定设备，该命令将会中止。

与命令的 `change` 格式一起使用时，在双节点群集中会自动选择并分配一个设备，用于取代所有现有法定设备。该法定设备是从可用设备中选择的。

群集中的所有可用设备都必须符合成为法定设备的条件。`autoconfig` 子选项不会评估某个可用设备是否符合成为法定设备的条件。

如果群集包含两个以上的节点，`autoconfig` 子选项将不会更改法定配置。如果计划将某个 NAS 设备配置为法定设备，请不要使用 `autoconfig` 子选项。

[,noop]

与 `autoconfig` 子选项一起使用时有效。该命令在标准输出中显示将由 `autoconfig` 子选项添加或更改的法定设备的列表。`autoconfig,noop` 子选项不会更改法定配置。

如果在执行法定相关操作时 `scconf` 中断或发生故障，群集配置数据库中的法定配置信息可能会出现不一致。如果出现不一致的情况，再次运行相同的 `scconf` 命令或使用 `reset` 选项运行该命令可重置法定信息。

使用命令的 `add` 形式，如果在没有 `node` 列表的情况下指定 `name`，则会添加法定设备，并且为设备所连接的每个节点定义一个端口。但如果指定 `node` 列表，则必须至少提供两个节点，并且必须为列表中的每个节点提供端口以便设备能够与节点连接。

例 308 添加 SCSI 法定设备

以下 `scconf` 命令可用于添加 SCSI 法定设备。

```
-a -q globaldev=/dev/did/rdisk/d4s2  
    or  
-a -q name=/dev/did/rdisk/d4s2,type=scsi
```

例 309 更改 SCSI 法定设备

以下 `scconf` 命令可用于更改 SCSI 法定设备配置。

```
-c -q globaldev=/dev/did/rdisk/d4s2,reset
```

or
-c -q name=/dev/did/rdisk/d4s2,reset

例 310 删除 SCSI 法定设备

以下 `scconf` 命令可用于删除 SCSI 法定设备 `qd1`。

```
-r -q globaldev=qd1
```

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	过时

[Intro\(1CL\) \[17\]](#)、[clquorum\(1CL\) \[195\]](#)、[cluster\(1CL\) \[461\]](#)、[scconf\(1M\) \[637\]](#)

名称

scconf_transp_adap_bge — 配置 bge 传输适配器

注 - Oracle Solaris Cluster 软件包括面向对象的命令集。虽然 Oracle Solaris Cluster 软件仍支持原始的命令集，但是 Oracle Solaris Cluster 过程文档仅使用面向对象的命令集。有关面向对象的命令集的更多信息，请参见 [Intro\(1CL\) \[17\]](#) 手册页。

可将 bge 适配器配置为群集传输适配器。这类适配器只能用于 dlpi 传输类型。bge 适配器支持 VLAN 功能。

bge 适配器可连接到传输交换机或其他节点中的另一 bge 适配器。在任一情况下，连接都是通过传输电缆建立的。

使用传输交换机并使用 scconf 命令、scinstall 命令或其他工具配置传输电缆的端点时，会要求您指定传输交换机的端口名称。您可提供任何端口名称，或接受默认名称，只要该名称对交换机是唯一的。

默认方式是将端口名称设置为电缆另一端适配器所在节点的 ID。

这种类型的群集传输适配器没有用户可配置的属性。

[Intro\(1CL\) \[17\]](#)、[clinterconnect\(1CL\) \[97\]](#)、[clnode\(1CL\) \[149\]](#)、[scconf\(1M\) \[637\]](#)、[scinstall\(1M\) \[697\]](#)、[link to " bge7D"](#)

名称

scconf_transp_adap_e1000g — 配置 Intel PRO/1000 网络适配器

注 - Oracle Solaris Cluster 软件包括面向对象的命令集。虽然 Oracle Solaris Cluster 软件仍支持原始的命令集，但是 Oracle Solaris Cluster 过程文档仅使用面向对象的命令集。有关面向对象的命令集的更多信息，请参见 [Intro\(1CL\) \[17\]](#) 手册页。

可将 e1000g Intel PRO/1000 网络适配器配置为群集传输适配器。这类适配器只能用于 dlp_i 传输类型。

基于 e1000g 的网络适配器可连接到传输交换机或其他节点中的另一以太网适配器。在任一情况下，连接都是通过传输电缆建立的。

使用传输交换机并使用 scconf、scinstall 或其他工具配置传输电缆的端点时，会要求您指定传输交换机的端口名称。您可提供任何端口名称，或接受默认名称，只要该名称对交换机是唯一的。

默认方式是将端口名称设置为电缆另一端适配器所在节点的标识符。

这种类型的群集传输适配器没有用户可配置的属性。

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
体系结构	x86
可用性	过时

[Intro\(1CL\) \[17\]](#)、[clinterconnect\(1CL\) \[97\]](#)、[clnode\(1CL\) \[149\]](#)、[scconf\(1M\) \[637\]](#)、[scinstall\(1M\) \[697\]](#)、[link to "e1000g7D"](#)

名称

scconf_transp_jct_etherswitch — 配置以太网群集传输交换机

注 - Oracle Solaris Cluster 软件包括面向对象的命令集。虽然 Oracle Solaris Cluster 软件仍支持原始的命令集，但是 Oracle Solaris Cluster 过程文档仅使用面向对象的命令集。有关面向对象的命令集的更多信息，请参见 [Intro\(1CL\) \[17\]](#) 手册页。

可以将以太网交换机配置为群集传输交换机，也称为传输结点。这类交换机属于 switch 交换机类型。没有用户可配置的属性。

[Intro\(1CL\) \[17\]](#)、[clinterconnect\(1CL\) \[97\]](#)、[clnode\(1CL\) \[149\]](#)

名称

scconf_transp_jct_ibswitch — 配置 InfiniBand 群集传输交换机

注 - Oracle Solaris Cluster 软件包括面向对象的命令集。虽然 Oracle Solaris Cluster 软件仍支持原始的命令集，但是 Oracle Solaris Cluster 过程文档仅使用面向对象的命令集。有关面向对象的命令集的更多信息，请参见 [Intro\(1CL\) \[17\]](#) 手册页。

可以将 InfiniBand 交换机配置为群集传输交换机，也称为传输结点。这类交换机属于 switch 交换机类型。没有用户可配置的属性。

[Intro\(1CL\) \[17\]](#)、[clinterconnect\(1CL\) \[97\]](#)、[clnode\(1CL\) \[149\]](#)

名称

scdidadm — 设备标识符配置和管理实用程序包装

```
/usr/cluster/bin/scdidadm -b combined-did-instance
/usr/cluster/bin/scdidadm -C
/usr/cluster/bin/scdidadm -c
/usr/cluster/bin/scdidadm -F
    {pathcount | scsi3 | useglobal} instance
/usr/cluster/bin/scdidadm -G
/usr/cluster/bin/scdidadm -G {pathcount | prefer3}
/usr/cluster/bin/scdidadm {-l | -L} [-h] [-o fmt]... [path |
    instance]
/usr/cluster/bin/scdidadm -R {path | instance | all}
/usr/cluster/bin/scdidadm -r
/usr/cluster/bin/scdidadm -T remote-nodename -e replication-type
/usr/cluster/bin/scdidadm -t
    source-instance:destination-instance -e replication-type [-g
    replication-device-group]
/usr/cluster/bin/scdidadm [-u] [-i]
/usr/cluster/bin/scdidadm -v
```

注 - Oracle Solaris Cluster 软件包括面向对象的命令集。虽然 Oracle Solaris Cluster 软件仍支持原始的命令集，但是 Oracle Solaris Cluster 过程文档仅使用面向对象的命令集。有关面向对象的命令集的更多信息，请参见 [Intro\(1CL\) \[17\]](#) 手册页。

scdidadm 实用程序管理设备标识符 (device identifier, DID) 伪设备驱动程序 did。

scdidadm 实用程序执行以下主要操作：

- 创建驱动程序配置文件
- 修改文件中的条目
- 将当前配置加载到内核
- 列出设备条目和 did 驱动程序实例号之间的映射

启动脚本 `/etc/init.d/bootcluster` 使用 `scdidadm` 实用程序来初始化 did 驱动程序。您也可以使用 `scdidadm` 来更新或查询现有设备和相应设备标识符以及 did 驱动程序实例号之间的当前设备映射。

[Unresolved link to "devfsadm1M"](#) 命令创建文件系统设备入口点。

仅可以在全局区域中使用此命令。

支持以下选项：

-b

将复制的 DID 实例返回到之前为两个独立 DID 实例时的状态。使用此选项可纠正配置错误或准备影响原始 DID 实例的配置更改。

只能从以群集模式引导的节点使用此选项。仅可以在全局区域中使用此选项。

使用 -b 选项前，从使用复制设备的所有设备组中删除复制设备。然后从其 DID 实例被并入组合 DID 实例的节点之一指定此选项。

您需要具有 `solaris.cluster.device.modify` RBAC 授权才能使用该命令选项。请参见 [Unresolved link to "rbac5"](#)。

-c

删除所有已从当前节点断开的底层设备的 DID 引用。

只能从以群集模式引导的节点使用此选项。仅可以在全局区域中使用此选项。

在已使用 Solaris 设备命令删除对群集节点上不存在设备的引用后指定此选项。

-f 选项不影响已配置的法定设备的隔离协议。

您需要具有 `solaris.cluster.device.modify` RBAC 授权才能使用该命令选项。请参见 [Unresolved link to "rbac5"](#)。

-c

对设备的内核表示和物理设备执行一致性检查。

仅可以在全局区域中使用此选项。

一致性检查失败时，会显示错误消息。该过程会继续，直到已检查完所有设备为止。

您需要具有 `solaris.cluster.device.read` RBAC 授权才能使用该命令选项。请参见 [Unresolved link to "rbac5"](#)。

-e *type*

指定复制类型。指定 SRDF 复制类型时，此选项只能与 -t 选项一起使用。

-F

覆盖各个指定设备的全局默认隔离算法。

设备的默认隔离算法可设置为以下值之一：

- | | |
|------------------------|---|
| <code>pathcount</code> | 根据连接到共享设备的 DID 路径的数目来确定隔离协议。 <ul style="list-style-type: none">■ 对于使用少于 3 个 DID 路径的设备，该命令会设置 SCSI-2 协议。■ 对于使用不少于 3 个 DID 路径的设备，该命令会设置 SCSI-3 协议。 |
|------------------------|---|

`scsi3` 设置 SCSI-3 协议。如果设备不支持 SCSI-3 协议，则隔离协议设置将保持不变。

`useglobal` 设置指定设备的全局默认隔离设置。

默认情况下，全局默认隔离算法将设置为 `pathcount`。有关设置全局隔离默认值的信息，请参见 `-G` 选项的描述。

可以通过设备实例号指定要修改的设备。该命令接受以空格分隔的多个设备的列表。有关 `instance` 格式的设备名称的更多信息，请参见 `-o` 选项的描述。

`-F` 选项不影响已配置的法定设备的隔离协议。

您需要具有 `solaris.cluster.device.modify` RBAC 授权才能使用该命令选项。请参见 [Unresolved link to "rbac5"](#)。

`-G`

为所有共享设备设置或显示当前全局默认隔离算法。

单独指定时，`-G` 选项会显示当前全局默认隔离算法设置。

与设置值一起指定时，`-G` 选项会将所有设备的全局默认隔离设置为该值。全局默认隔离可以设置为以下值之一：

`prefer3` 为所有设备的设备隔离设置 SCSI-3 协议。`pathcount` 设置已指定给不支持 SCSI-3 协议的所有设备。

`pathcount` 根据连接到共享设备的 DID 路径的数目来确定隔离协议。

- 对于使用少于 3 个 DID 路径的设备，该命令会设置 SCSI-2 协议。
- 对于使用不少于 3 个 DID 路径的设备，该命令会设置 SCSI-3 协议。

默认情况下，全局默认隔离算法将设置为 `pathcount`。

`-G` 选项不影响已配置的法定设备的隔离协议。

您需要具有 `solaris.cluster.device.modify` RBAC 授权才能使用该命令选项。请参见 [Unresolved link to "rbac5"](#)。

`-g`

指定复制设备组。

`-h`

列出设备映射时显示标头。

仅当与 `-l` 和 `-L` 选项一起使用时，此选项才有意义。

`-i`

初始化 `did` 驱动程序。

仅可以在全局区域中使用此选项。

如果要启用 did 驱动程序的 I/O 请求，则使用此选项。

您需要具有 `solaris.cluster.device.modify` RBAC 授权才能使用该命令选项。请参见 [Unresolved link to "rbac5"](#)。

-L

列出 DID 配置文件中设备的所有路径，包括远程主机上的路径。

仅可以在全局区域中使用此选项。

可以使用 `-o` 选项定制此命令的输出。如果不指定 `-o` 选项，默认列表将显示 `instance` 号、所有本地和远程 `fullpath` 字符串以及 `fullname`。

您需要具有 `solaris.cluster.device.read` RBAC 授权才能使用该命令选项。请参见 [Unresolved link to "rbac5"](#)。

-L

列出 DID 配置中的本地设备。

仅可以在全局区域中使用此选项。

可以使用 `-o` 选项定制此命令的输出。如果不指定 `-o` 选项，默认列表将显示 `instance` 号、本地 `fullpath` 和 `fullname`。

您需要具有 `solaris.cluster.device.read` RBAC 授权才能使用该命令选项。请参见 [Unresolved link to "rbac5"](#)。

-o *fmt*

根据格式规范 *fmt* 列出 did 驱动程序当前已知的设备。

可指定多个 `-o` 选项。*fmt* 规范体现为以逗号分隔的格式选项参数列表。仅当与 `-l` 和 `-L` 选项一起使用时，此选项才有意义。可用的格式选项参数如下：

<code>instance</code>	显示 did 驱动程序已知的设备的实例号，例如 1。
<code>path</code>	显示与此设备标识符关联的设备的物理路径名称，例如 <code>/dev/rdisk/c0t3d0</code> 。
<code>fullpath</code>	显示与此设备标识符关联的设备的完整物理路径名称。此路径名称包括主机，例如 <code>phys-hostA:/dev/rdisk/c0t3d0</code> 。
<code>host</code>	带有 <code>-L</code> 选项时，显示已连接至指定设备的所有主机名（每行一个）。带有 <code>-l</code> 选项时，显示已连接至指定设备的本地主机的名称。
<code>name</code>	显示与此设备标识符关联的设备的 DID 名称，例如 <code>d1</code> 。
<code>fullname</code>	显示与此设备标识符关联的设备的完整 DID 路径名称，例如 <code>/dev/did/rdisk/d1</code> 。
<code>diskid</code>	显示与列出的设备实例关联的设备标识符的十六进制表示。

`asciidiskid` 显示与列出的设备实例关联的设备标识符的 ASCII 表示。

`defaultfencing` 显示设备的默认隔离算法设置。

`-R {path | instance | all}`

在特定设备实例上执行修复过程。

仅可以在全局区域中使用此选项。

此命令的参数可以是已使用新设备进行替换的特定物理设备 `path`，也可以是刚替换的设备的 `instance`。与 `all` 关键字一起使用时，`scdidadm` 实用程序会更新连接到节点的所有设备的配置数据。

只能从以群集模式引导的节点使用此选项。

您需要具有 `solaris.cluster.device.modify` RBAC 授权才能使用该命令选项。请参见 [Unresolved link to "rbac5"](#)。

`-r`

重新配置数据库。

仅可以在全局区域中使用此选项。

如果指定此选项，将进行对 `rdsk` 和 `rmt` 设备树的全面搜索。将为之前未识别的所有设备标识符分配新的实例号。将为每个新识别的设备添加新路径。

只能从以群集模式引导的节点使用此选项。

您需要具有 `solaris.cluster.device.modify` RBAC 授权才能使用该命令选项。请参见 [Unresolved link to "rbac5"](#)。

`-T remote-nodename`

配置 DID 设备以用于您使用 `-ereplication—type` 参数指定的基于存储的复制类型。

仅可以在全局区域中使用此选项。

仅从配置有复制设备的节点之一运行此选项。使用 `remote-nodename` 选项参数指定远程节点的名称。

本地节点上的 DID 实例将与远程节点上的相应 DID 实例组合到一起，从而将每对复制的设备合并为单个逻辑 DID 设备。

您需要具有 `solaris.cluster.device.modify` RBAC 授权才能使用该命令选项。请参见 [Unresolved link to "rbac5"](#)。

`-t source-instance:destination-instance`

将 DID 实例从原始源移动到您使用 `destination-instance` 选项参数指定的新目标实例。

仅可以在全局区域中使用此选项。

如果意外更改了本地实例，可使用此选项将 DID 实例移回其原始位置。在已连接到共享存储的所有群集节点上运行该命令后，从某一节点运行 `devfsadm` 和 `scgdevs` 命令以使用该配置更改来更新全局设备名称空间。

如果 *destination-instance* 不存在于群集中，则会删除并使用您指定的 *destination-instance* 重新创建对应于 *source-instance* 参数值的 DID 设备路径。

如果 *destination-instance* 已存在于群集中，则 *source-instance* 的一条或多条路径会与 *destination-instance* 的一条或多条路径组合到一起，产生包含这两个实例的所有路径的单个 DID 目标实例。

包括 *-e* 选项以指定复制类型。如果是使用 SRDF 复制设备，则必须使用 *-g* 选项来指定复制设备组。

您需要具有 `solaris.cluster.device.modify` RBAC 授权才能使用该命令选项。请参见 [Unresolved link to "rbac5"](#)。

-u

将设备标识符配置表加载到内核中。

仅可以在全局区域中使用此选项。

此选项可将有关设备路径及其相应实例号的所有当前已知配置信息加载到内核中。

您需要具有 `solaris.cluster.device.modify` RBAC 授权才能使用该命令选项。请参见 [Unresolved link to "rbac5"](#)。

-v

显示此程序的版本号。

仅可以在全局区域中使用此选项。

例 311 将连接到本地主机的设备添加到 CCR

```
# sccdidadm -r
```

例 312 列出设备的物理路径

以下示例列出了与 `did` 驱动程序的实例 2 相对应的设备的物理路径：

```
% sccdidadm -l -o path 2
/dev/dsk/c1t4d0
```

例 313 指定多个格式选项

您可以使用以下任一方法指定多个格式选项参数：

```
% sccdidadm -l -o path -o name 2
```

```
% sccdidadm -l -o path,name 2
```

在任一示例中，输出都可能类似如下：

```
/dev/dsk/c1t4d0 d1
```

例 314 配置 DID 设备以用于基于 EMC SRDF 存储的复制

以下示例配置本地 DID 设备和远程 DID 设备，以用于基于 EMC SRDF 存储的复制。该命令从配置有复制设备的本地源节点上运行。源节点上的 DID 实例会与远程目标节点 phys-schost-1 上的相应 DID 实例组合到一起。

```
# sccidadm -t 15:10 -e srdf -g devgroup1
```

例 315 取消配置复制的 DID 设备

以下示例将复制的 DID 设备 d25 返回至其原始 DID 设备组件。此 DID 设备是通过将 phys-schost--1 上的路径 d15 与 phys-schost--2 上的路径 d25 组合到一起创建的。因为在两个路径进行组合时路径 d15 已合并到路径 d25 中，所以您必须从 phys-schost-1 运行命令以确保恢复路径 d15。

```
phys-schost-1# sccidadm -b 25
```

例 316 移动 DID 实例

以下示例将源实例 15 上的 DID 实例移动到新的 DID 实例 10，然后在全局设备名称空间中更新该配置更改。

```
# sccidadm -t 15:10  
# devfsadm  
# scgdevs
```

例 317 执行修复过程

以下示例为特定设备路径执行修复过程。设备 /dev/dsk/c1t4d0 已替换为关联有新设备标识符的新设备。更新数据库以显示此新设备标识符对应于之前与旧设备标识符相关联的实例号：

```
# sccidadm -R c1t4d0
```

例 318 执行修复过程

执行修复过程的替代方法是使用与设备路径相关联的实例号。例如，如果上一示例中的设备 c1t4d0 的实例号是 2，则以下语法会执行与上一示例相同的操作：

```
# sccidadm -R 2
```

例 319 全局设置 SCSI 协议

以下示例将群集中的所有 SCSI 设备设置为 SCSI-3 协议，已配置的法定设备和不支持 SCSI-3 协议的设备除外。不支持 SCSI-3 协议的所有设备将设置为 pathcount。

```
# sccidadm -G prefer3
```

例 320 显示单个设备的 SCSI 协议

以下示例显示设备 `/dev/rdisk/c0t3d0` 的 SCSI 协议设置。

```
# scdidadm -L -o defaultfencing /dev/rdisk/c0t3d0
```

例 321 设置单个设备的 SCSI 协议

以下示例为设备 11（由实例编号指定）设置 SCSI-3 协议。此设备不是已配置的法定设备，且支持 SCSI-3 协议。

```
# scdidadm -F scsi3 11
```

返回以下退出值：

0 命令已成功完成。

1 发生错误。

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Stable (稳定)

[Intro\(1CL\) \[17\]](#)、[cldevice\(1CL\) \[51\]](#)、[Unresolved link to "devfsadm1M"](#)、[scgdevs\(1M\) \[695\]](#)、[did\(7\) \[1275\]](#)

[Unresolved link to "Oracle Solaris Cluster 系统管理指南"](#)

每次进行物理连接时，每个多端口的磁带驱动器或 CD-ROM 驱动器在名称空间中显示一次。

名称

scdpm — 管理磁盘路径监视守护进程

```
scdpm [-a] {node | all}

scdpm -f filename

scdpm -m {[node | all][:dev/did/rdisk/]dN | [:dev/rdisk/]cNtXdY | all}

scdpm -n {node | all}

scdpm -p [-F] {[node | all][:dev/did/rdisk/]dN | [/dev/rdisk/]cNtXdY | all}

scdpm -u {[node | all][:dev/did/rdisk/]dN | [/dev/rdisk/]cNtXdY | all}
```

注 - Oracle Solaris Cluster 软件包括面向对象的命令集。虽然 Oracle Solaris Cluster 软件仍支持原始的命令集，但是 Oracle Solaris Cluster 过程文档仅使用面向对象的命令集。有关面向对象的命令集的更多信息，请参见 [Intro\(1CL\) \[17\]](#) 手册页。

scdpm 命令用于管理群集中的磁盘路径监视守护进程。可使用此命令监视和取消监视磁盘路径。也可以使用此命令显示磁盘路径或节点的状态。群集中或特定节点上的所有可访问磁盘路径都会显示在标准输出上。必须在处于联机状态且处于群集模式的群集节点上运行此命令。

在监视新的磁盘路径时，可以指定全局磁盘名称或 UNIX 路径名。此外，还可以强制守护进程重新读取整个磁盘配置。

仅可以在全局区域中使用此命令。

支持以下选项：

-a

在所有受监视的磁盘路径均发生故障时启用节点的自动重新引导功能，前提是满足以下条件：

- 节点上所有受监视的磁盘路径均发生故障。
- 至少有一个磁盘可从群集中的其他节点进行访问。

仅可以在全局区域中使用此选项。

重新引导节点会将该节点管理的所有资源和设备组在另一个节点上重新启动。

当节点自动重新引导后，如果该节点上所有受监视的磁盘路径仍不可访问，该节点不会再次自动重新引导。但是，如果节点重新引导后有任何受监视的磁盘路径变为可用，但随后所有受监视磁盘路径又出现故障，则该节点会再次自动重新引导。

您需要具有 `solaris.cluster.device.admin` 基于角色的访问控制 (Role-Based Access Control, RBAC) 授权才能使用此选项。请参见 [Unresolved link to "rbac5"](#)。

-F

如果将 -F 选项与 -p 选项一起指定，`scdpm` 还会显示群集中发生故障的磁盘路径。-p 选项可从已连接到存储的所有节点显示某节点或指定磁盘路径的当前状态。

-f *filename*

读取 *filename* 中要监视或取消监视的磁盘路径列表。

仅可以在全局区域中使用此选项。

以下示例显示 *filename* 的内容。

```
u schost-1:/dev/did/rdisk/d5
m schost-2:all
```

文件中的每一行都必须指定监视或取消监视磁盘路径、节点名称和磁盘路径名。要监视，则指定 `m` 选项；要取消监视，则指定 `u` 选项。必须在命令和节点名称之间插入一个空格。还必须在节点名称和磁盘路径名之间插入一个冒号 (`:`)。

您需要具有 `solaris.cluster.device.admin` RBAC 授权才能使用此选项。请参见 [Unresolved link to "rbac5"](#)。

-m

监视 *node* :*diskpath* 指定的新磁盘路径。

仅可以在全局区域中使用此选项。

您需要具有 `solaris.cluster.device.admin` RBAC 授权才能使用此选项。请参见 [Unresolved link to "rbac5"](#)。

-n

禁用节点在所有受监视的磁盘路径均发生故障时自动重新引导的功能。

仅可以在全局区域中使用此选项。

如果节点上所有受监视的磁盘路径均发生故障，不会重新引导节点。

您需要具有 `solaris.cluster.device.admin` RBAC 授权才能使用此选项。请参见 [Unresolved link to "rbac5"](#)。

-p

从已连接到存储的所有节点显示某节点或指定磁盘路径的当前状态。

仅可以在全局区域中使用此选项。

如果还指定 -F 选项，`scdpm` 将显示群集中发生故障的磁盘路径。

磁盘路径的有效状态值为 `Ok`、`Fail`、`Unmonitored` 或 `Unknown`。

节点的有效状态值为 `Reboot_on_disk_failure`。有关 `Reboot_on_disk_failure` 状态的更多信息，请参见 -a 和 -n 选项的描述。

您需要具有 `solaris.cluster.device.read` RBAC 授权才能使用此选项。请参见 [Unresolved link to "rbac5"](#)。

-u

取消监视磁盘路径。每个节点上的守护进程均停止监视指定的路径。
仅可以在全局区域中使用此选项。

您需要具有 `solaris.cluster.device.admin` RBAC 授权才能使用此选项。请参见 [Unresolved link to "rbac5"](#)。

例 322 监视群集基础结构中的所有磁盘路径

以下命令强制守护进程监视群集基础结构中的所有磁盘路径。

```
# scdpm -m all
```

例 323 监视新磁盘路径

以下命令监视新的磁盘路径。所有节点都监视此路径在其中有效的 `/dev/did/dsk/d3`。

```
# scdpm -m /dev/did/dsk/d3
```

例 324 监视单个节点上的新磁盘路径

以下命令监视单个节点上的新路径。schost-2 节点上的守护进程监视 `/dev/did/dsk/d4` 和 `/dev/did/dsk/d5` 磁盘的路径。

```
# scdpm -m schost-2:d4 -m schost-2:d5
```

例 325 显示所有磁盘路径及其状态

以下命令显示群集中的所有磁盘路径及其状态。

```
# scdpm -p
schost-1:reboot_on_disk_failure enabled
schost-2:reboot_on_disk_failure disabled
schost-1:/dev/did/dsk/d4 Ok
schost-1:/dev/did/dsk/d3 Ok
schost-2:/dev/did/dsk/d4 Fail
schost-2:/dev/did/dsk/d3 Ok
schost-2:/dev/did/dsk/d5 Unmonitored
schost-2:/dev/did/dsk/d6 Ok
```

例 326 显示所有发生故障的磁盘路径

以下命令显示 schost-2 节点上所有发生故障的磁盘路径。

```
# scdpm -p -F all
```

```
schost-2:/dev/did/dsk/d4          Fail
```

例 327 显示单个节点上所有磁盘路径的状态

以下命令显示 schost-2 节点上监视的所有磁盘的磁盘路径和状态。

```
# scdpm -p schost-2:all
schost-2:reboot_on_disk_failure  disabled
schost-2:/dev/did/dsk/d4         Fail
schost-2:/dev/did/dsk/d3         Ok
```

返回以下退出值：

0 命令已成功完成。
1 该命令完全失败。
2 该命令部分失败。

注 - 磁盘路径由节点名称和磁盘名称表示。节点名称必须是主机名或 all。磁盘名称必须是全局磁盘名称、UNIX 路径名或 all。磁盘名称可以是完整的全局路径名或磁盘名称：/dev/did/dsk/d3 或 d3。磁盘名称也可以是完整的 UNIX 路径名：/dev/rdisk/c0t0d0s0。

以 syslogd LOG_INFO 设备级别记录磁盘路径状态的更改。以 LOG_ERR 设备级别记录所有故障。

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
稳定性	Evolving (发展中)

[Intro\(1CL\) \[17\]](#)、[cldevice\(1CL\) \[51\]](#)、[clnode\(1CL\) \[149\]](#)、[Unresolved link to "attributes5"](#)

[Unresolved link to "Oracle Solaris Cluster 系统管理指南"](#)

名称

sceventmib — 管理 Oracle Solaris Cluster 事件 MIB 模块

```
sceventmib -a -c community -h host ...  
sceventmib -a -t auth-type -u username [-f password-file]  
sceventmib -d -s security-level -u username  
sceventmib {-e | -n}  
sceventmib -l protocol  
sceventmib -m -t auth-type -u username  
sceventmib -p {all | hosts | users}  
sceventmib -r -c community -h host...  
sceventmib -r -u username
```

注 - Oracle Solaris Cluster 软件包括面向对象的命令集。虽然 Oracle Solaris Cluster 软件仍支持原始的命令集，但是 Oracle Solaris Cluster 过程文档仅使用面向对象的命令集。有关面向对象的命令集的更多信息，请参见 [Intro\(1CL\) \[17\]](#) 手册页。

sceventmib 命令可启用、禁用和配置 Oracle Solaris Cluster 事件管理信息库 (Management Information Base, MIB) 模块。在某一群集节点上发出该命令时，它只会影响该节点上的 MIB 模块的配置。每个群集节点 MIB 模块独立于群集中的其他模块运行。

可以使用该命令启用或禁用群集节点上的 MIB 模块。还可以使用该命令设置配置属性，例如 SNMP 陷阱通知的版本或要将陷阱通知发送到的 IP 地址对应的主机名。Oracle Solaris Cluster 事件 MIB 在端口 11162 上发送陷阱通知。可在端口 11161 上查看 SNMP 树。

仅可以在全局区域中使用此命令。

基本选项

以下选项控制了该命令的基本格式和功能：

-a

将指定 SNMP 主机和团体的条目或指定用户的条目添加到节点配置文件。
仅可以在全局区域中使用此选项。

-d

设置指定 SNMPv3 协议时要使用的默认安全性级别和用户。

仅可以在全局区域中使用此选项。

指定 SNMPv3 时，必须指定默认用户。使用 SNMPv3，可以为 MIB 模块配置多个用户。任何时候都只能存在一个默认用户。无论该选项的设置如何，添加的第一个用户将自动定义为默认用户。在这种情况下，默认用户不一定与 Oracle Solaris OS 用户相同。

-e

在节点上启用 Oracle Solaris Cluster 事件 MIB 模块。该设置将一直有效，除非您更改它，即使重新引导节点也是如此。

仅可以在全局区域中使用此选项。

-l *protocol*

设置 MIB 将使用的 SNMP 协议的版本。

仅可以在全局区域中使用此选项。

可以为 *protocol* 指定 SNMPv2 或 SNMPv3。除非首先配置了至少一个 SNMPv3 用户，否则无法指定 SNMPv3 协议。

-m

修改 SNMP 用户的验证类型。

仅可以在全局区域中使用此选项。

-n

在节点上禁用 Oracle Solaris Cluster 事件 MIB 模块。该设置将一直有效，除非您更改它，即使重新引导节点也是如此。

仅可以在全局区域中使用此选项。

-p {all | hosts | users}

显示以下类型之一的 MIB 配置信息：

all	所有 MIB 模块配置信息
hosts	仅配置为用于 MIB 模块的 SNMP 主机的配置信息
users	仅配置为使用 MIB 模块的 SNMP 用户的配置信息

仅可以在全局区域中使用此选项。

-r

从节点配置文件中删除指定 SNMP 主机和团体的条目或指定 SNMP 用户的条目。

仅可以在全局区域中使用此选项。

附加选项

您可以将附加选项与基本选项结合使用，以修改每种命令格式的默认行为。有关其中哪些选项能够合法地与哪些 `sceventmib` 格式结合使用的更多详细信息，请参阅“用法概要”部分。

支持下列附加选项：

`-c community`

指定要添加到节点配置文件或从节点配置文件中删除的 SNMP 团体的名称。

`-f password-file`

指定包含一个或多个 SNMP 用户名及其相应密码的密码文件的名称。

对 `password-file` 文件中包括的每一行使用以下语法：

```
user:password
```

例如，为用户 Joseph Bloggs 和 Andrew Smith 指定以下行：

```
jbloggs:fgrxty_0  
asmith:artfli!9
```

`-h host ...`

指定 SNMP 主机的名称。可以为 `host` 指定 IP 地址或主机名。

可以将一个主机添加到多个团体中。但是，如果同一团体中已存在具有相同名称的主机，将会返回错误。

`-s security-level`

为指定的 SNMPv3 用户指定安全性级别。该设置确定了用户可以访问 SNMP MIB 模块的程度。

可以为一个用户分配多个安全性级别。

可以为 `security-level` 指定以下区分大小写设置之一：

<code>authNoPriv</code>	需要验证安全措施，但不需要隐私安全措施。
<code>authPriv</code>	需要验证安全措施和隐私安全措施二者。
<code>noAuthNoPriv</code>	不需要验证安全措施和隐私安全措施。

`-t auth-type`

指定要使用的验证加密机制。可以为 `auth-type` 指定 MD5 或 SHA。

`-u username`

指定 SNMPv3 用户的名称。

如果为用户添加条目，但相同的用户名和安全性级别已存在，则会覆盖该信息。
如果删除默认 SNMPv3 用户，该命令将会自动选择另一个默认用户。

例 328 启用事件 MIB

以下命令启用事件 MIB。

```
# sceventmib -e
```

例 329 将 SNMP 主机添加到团体

以下命令将一个主机添加到 SNMP 团体 public。

- 第一个示例通过主机名 sc-host 指定主机。

```
# sceventmib -a -h sc-host -c public
```

- 第二个示例通过主机的 IP 地址 10.0.0.25 指定主机。

```
# sceventmib -a -h 10.0.0.25 -c public
```

例 330 添加 SNMP 用户而不提供密码文件

以下命令添加用户 jbloggs 并指定 MD5 验证加密机制。由于未指定密码文件，该命令会提示用户提供密码。

```
# sceventmib -a -t MD5 -u jbloggs
Enter password for user jbloggs: *****
```

例 331 添加 SNMP 用户并提供密码文件

以下命令添加用户 jbloggs 并指定 MD5 验证加密机制和密码文件 pfile。由于指定了密码文件，该命令不会提示用户提供密码。

```
# cat pfile
jbloggs:fgrxty_0
# sceventmib -a -f pfile -t MD5 -u jbloggs
```

例 332 显示所有 SNMP 配置信息

以下命令显示所有 SNMP 配置信息。

```
# sceventmib -p all
```

例 333 仅显示有关 SNMP 主机的配置信息

以下命令仅显示有关 SNMP 主机的配置信息。

```
# sceventmib -p hosts
```

例 334 设置 SNMP 协议的版本

以下命令将 SNMP 协议版本设置为 SNMPv3。

```
# sceventmib -l SNMPv3
```

例 335 设置默认 SNMP 用户

以下命令将默认 SNMP 用户设置为用户 jbloggs，并需要验证安全措施和隐私安全措施。

```
# sceventmib -d -s authPriv -u jbloggs
```

例 336 修改用户的验证类型

以下命令将用户 jbloggs 的验证类型更改为 SHA。

```
# sceventmib -m -t SHA -u jbloggs
```

例 337 删除 SNMP 主机

以下命令在团体 public 中删除 IP 地址为 10.0.0.25 的 SNMP 主机。

```
# sceventmib -r -c public -h 10.0.0.25
```

例 338 删除 SNMP 用户

以下命令删除 SNMP 用户 jbloggs。

```
# sceventmib -r -u jbloggs
```

此命令返回下列退出状态代码：

0 命令已成功完成。

非零值 发生错误。

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core

属性类型	属性值
接口稳定性	Evolving (发展中)

`/usr/cluster/lib/mib/sun-cluster-event-mib.mib`

Oracle Solaris Cluster SNMP 事件 MIB 定义文件

[Intro\(1CL\) \[17\]](#)、[clsnmphost\(1CL\) \[409\]](#)、[clsnmpmib\(1CL\) \[417\]](#)、[clsnmpuser\(1CL\) \[425\]](#)、[Unresolved link to "attributes5"](#)

[Unresolved link to "Oracle Solaris Cluster 系统管理指南"](#)

名称

scgdevs — 全局设备名称空间管理脚本

/usr/cluster/bin/scgdevs

注 - Oracle Solaris Cluster 软件包括面向对象的命令集。虽然 Oracle Solaris Cluster 软件仍支持原始的命令集，但是 Oracle Solaris Cluster 过程文档仅使用面向对象的命令集。有关面向对象的命令集的更多信息，请参见 [Intro\(1CL\) \[17\]](#) 手册页。

scgdevs 命令管理全局设备名称空间。全局设备名称空间挂载在 /global 目录下，由一组指向物理设备的逻辑链接组成。由于 /dev/global 目录对群集的每个节点都可见，因此每个物理设备在整个群集中都是可见的。这意味着，可以从群集中的任何节点访问添加到全局设备名称空间中的任何磁盘、磁带或 CD-ROM。

通过 scgdevs 命令，可以将新的全局设备（例如，磁带机、CD-ROM 驱动器和磁盘驱动器）附加到全局设备名称空间，而无需重新引导系统。必须先运行 devfsadm 命令，然后才能运行 scgdevs 命令。

或者，可以执行重新配置重新引导，以重新生成全局名称空间并附加新的全局设备。有关重新配置重新引导的更多信息，请参见 [Unresolved link to "boot1M"](#) 手册页。

必须从属于当前群集成员的节点中运行此命令。如果从不属于群集成员的节点中运行此命令，该命令将退出并显示错误代码，且系统状态保持不变。

仅可以在全局区域中使用此命令。

需要具有 solaris.cluster.system.modify RBAC 授权才能使用该命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

您还必须能够承担已向其分配 Oracle Solaris Cluster 命令权限配置文件的角色，才能使用该命令。经授权的用户可以通过 pfsh、pfcsh 或 pfksh 配置文件 shell 在命令行上发出 Oracle Solaris Cluster 特权命令。配置文件 shell 是一种特殊的 shell，通过该 shell，可以访问分配给 Oracle Solaris Cluster 命令权限配置文件的 Oracle Solaris Cluster 特权命令。当您运行 su 命令以承担某个角色时，即会启动配置文件 shell。还可以使用 pfexec 命令发出 Oracle Solaris Cluster 特权命令。

返回以下退出值：

0	命令已成功完成。
非零值	发生错误。错误消息显示在标准输出上。

/devices 设备节点目录

/global/.devices

全局设备节点目录

/dev/md/shared

Solaris Volume Manager 元集目录

有关下列属性的说明，请参见 [Unresolved link to " attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	过时

[Unresolved link to " pfcsh1"](#)、[Unresolved link to " pfexec1"](#)、[Unresolved link to " pfksh1"](#)、[Unresolved link to " pfsh1"](#)、[Intro\(1CL\) \[17\]](#)、[cldevice\(1CL\) \[51\]](#)、[Unresolved link to " boot1M"](#)、[Unresolved link to " devfsadm1M"](#)、[Unresolved link to " su1M"](#)、[did\(7\) \[1275\]](#)

[Unresolved link to " Oracle Solaris Cluster 系统管理指南 "](#)

从本地节点调用的 `scgdevs` 命令将在远程节点上以异步方式执行其工作。因此，在本地节点上完成命令并不一定意味着该命令已完成其群集中的工作。

本文档不会构成 API。在将来的发行版中，`/global/.devices` 目录和 `/devices` 目录可能不会存在或可能具有不同的内容或解释。存在此通知并不意味着缺少此通知的任何其他文档会构成 API。此接口应被视为不稳定的接口。

名称

scinstall — 初始化 Oracle Solaris Cluster 软件并建立新的群集节点

```
/usr/cluster/bin/scinstall -i -F [-C clustername]  
    [-T authentication-options] [-o] [-A adapter-options]  
    [-B switch-options] [-m cable-options] [-w netaddr-options]  
  
/usr/cluster/bin/scinstall -i -N cluster-member [-C clustername]  
    [-A adapter-options] [-B switch-options] [-m cable-options]  
  
/usr/cluster/bin/scinstall -c net-image-source -U password-file  
    -h nodename -n nodeip-mac-options -W software-specs -F  
    [-C clustername] [-T authentication-options] [-A adapter-options]  
    [-B switch-options] [-m cable-options] [-w netaddr-options]  
  
/usr/cluster/bin/scinstall -c net-image-source -U password-file  
    -h nodename -n nodeip-mac-options -W software-specs  
    -N cluster-member [-C clustername] [-A adapter-options]  
    [-B switch-options] [-m cable-options]  
  
/usr/cluster/bin/scinstall -c archive=archive-location:::cert=cert-file:::  
    key=key-file),action=initial -U password-file -h nodename  
    -n nodeip-mac-options -F [-C clustername] [-f hostnames-map-file]  
    [-T authentication-options] [-A adapter-options]  
    [-B switch-options] [-m cable-options] [-o] [-w netaddr-options]  
  
/usr/cluster/bin/scinstall -c archive=archive-location:::cert=cert-file:::  
    key=key-file),action=initial -U password-file -h nodename  
    -n nodeip-mac-options -N cluster-member [-C clustername] [-f hostnames-map-file]  
    [-T authentication-options] [-A adapter-options]  
    [-B switch-options] [-m cable-options] [-o] [-w netaddr-options]  
  
/usr/cluster/bin/scinstall -c archive=archive-location:::cert=cert-file:::  
    key=key-file),action=restore -h nodename [-F[-o]]  
    -C clustername -n nodeip-mac-options [-T secureAI=yes]  
  
/usr/cluster/bin/scinstall -c archive=archive-location:::cert=cert-file:::  
    key=key-file),action=replicate -h nodename [-F[-o]]  
    -C clustername -n nodeip-mac-options  
    [-T node=archive-source-node::node-to-install[,...] [,secureAI=yes]  
    [-f hostnames-map-file] [-w netaddr-options] -U password-file  
  
/usr/cluster/bin/scinstall -u upgrade-modes [upgrade-options]  
  
/usr/cluster/bin/scinstall -u update upgrade-options [pkg_fmri_pattern ...]  
  
/usr/cluster/bin/scinstall -r [-N cluster-member]  
  
scinstall -p [-v]
```

注 - Oracle Solaris Cluster 软件包括面向对象的命令集。虽然 Oracle Solaris Cluster 软件仍支持原始的命令集，但是 Oracle Solaris Cluster 过程文档仅使用面向对象的命令集。有关面向对象的命令集的更多信息，请参见 [Intro\(1CL\) \[17\]](#) 手册页。

scinstall 命令可执行许多 Oracle Solaris Cluster 节点创建和升级任务，如下所述。

- scinstall 的“初始化”格式 (-i) 将建立一个节点，作为新的 Oracle Solaris Cluster 配置成员。此格式将在新的群集中建立第一个节点 (-F)，或者向现有群集添加一个节点 (-N)。应始终从创建群集的节点或者要添加到群集中的节点运行 scinstall 命令的此格式。
- scinstall 的“添加安装客户机”格式 (-c) 可以建立指定的 *nodename*，将其作为运行该命令的自动化安装程序 (Automated Installer, AI) 安装服务器上的定制 AI 客户机。应始终从 AI 安装服务器运行 scinstall 命令的此格式。
- scinstall 的“删除”格式 (-r) 可以从群集节点中删除群集配置信息并卸载 Oracle Solaris Cluster 软件。
- scinstall 的“升级”格式 (-u) 具有多种模式和选项，可以升级 Oracle Solaris Cluster 节点。应始终从要升级的节点运行 scinstall 命令的此格式。
- scinstall 的“显示发行版”格式 (-p) 可以显示从中运行该命令的节点上安装的 Oracle Solaris Cluster 软件的发行版和软件包版本信息。

如果不指定选项，scinstall 命令将尝试在交互模式下运行。

应该以超级用户的身份运行 scinstall 命令的所有格式（“显示发行版”格式 (-p) 除外）。

ha-cluster/system/install 软件包包括 scinstall 命令的副本。

只能从全局区域运行此命令。

基本选项

下列选项控制了该命令的基本格式和功能。

不能在同一命令行上组合使用下列选项。

-c

指定 scinstall 命令的“添加安装客户机”格式。此选项会建立指定的 *nodename*，作为发出该命令的自动化安装程序 (Automated Installer, AI) 服务器上的定制 AI 客户机。此 -c 选项接受以下两种规范：`-c net-image-source` 和 `-c archive=archive-location[:cert=cert-file::key=key-file],action={initial/restore|replicate}`。

仅可以在全局区域中使用此选项。

在使用 AI 从 IPS 系统信息库安装 Oracle Solaris 和 Oracle Solaris Cluster 软件包并配置新群集时，必须指定 *net-image-source*。它可以是基于群集节点的体系结构

(SPARC 或 i386) ，您可从中检索 `install-image` 或 `solaris-auto-install` IPS 软件包的系统信息库：

```
-c publisher=repo[::cert=cert-file=key-file],arch={sparc|i386}
```

`net-image-source` 也可以是 Oracle Solaris 发行版的 AI ISO 映像文件。该文件必须可从已经建立且配置为安装群集节点的 AI 服务器访问：`-c iso-file`。

在使用统一归档文件自动安装群集或恢复群集节点时，使用 `archive=archive-location,action={initial|restore|replicate}` 命令。此命令指定统一归档文件的位置，并且可以是文件系统中能够从 AI 服务器、HTTP 位置或 HTTPS 位置访问的归档文件的完整路径。如果要访问 HTTPS 位置，必须指定 SSL 密钥和证书文件。您还必须指定归档文件的预定用途：是配置新群集 (`action=initial`)、恢复节点 (`action=restore`) 还是从具有相同硬件配置的现有群集复制一个新群集 (`action=replicate`)。在使用 `restore` 操作时，归档文件必须属于以前在要恢复的同一个节点上创建的归档文件的恢复类型。

此命令格式有助于将每个群集节点（或 `nodename`）建立为已建立的自动化安装程序安装服务器上的定制 AI 客户机，从而可以通过 AI 服务器实现完全自动化的群集安装。

对于 Oracle Solaris Cluster，可以定制 AI 清单文件。请参见 [Unresolved link to "Oracle Solaris Cluster 软件安装指南 中的如何安装和配置 Oracle Solaris 和 Oracle Solaris Cluster 软件 \(IPS 系统信息库\) 和 Unresolved link to "安装 Oracle Solaris 11.2 系统"](#)。

在使用 `scinstall` 命令将某个节点设置为定制 Oracle Solaris Cluster AI 客户机之前，必须先建立 AI 安装服务器。有关设置 AI 安装服务器的更多信息，请参见 [Unresolved link to "安装 Oracle Solaris 11.2 系统 中的第 8 章 设置 AI 服务器"](#)。

-i

指定 `scinstall` 命令的“初始化”格式。此命令格式将会建立一个节点作为新的群集成员。该新节点就是您从中发出 `scinstall` 命令的节点。

仅可以在全局区域中使用此选项。

如果将 `-F` 选项与 `-i` 一起使用，`scinstall` 会将节点建立为新群集中的第一个节点。

如果将 `-o` 选项与 `-F` 选项一起使用，`scinstall` 将会建立单节点群集。

如果将 `-N` 选项与 `-i` 一起使用，`scinstall` 会将节点添加到现有的群集。

-p

显示从中运行该命令的节点上安装的 Oracle Solaris Cluster 软件的发行版和软件包版本信息。只有这种格式的 `scinstall` 才不需要以超级用户的身份运行。

仅可以在全局区域中使用此选项。

-r

删除群集配置信息并从群集节点卸载 Oracle Solaris Cluster 框架和数据服务软件。然后，可以从群集中重新安装节点或删除节点。必须在您要卸载的节点上，从群集软件未占用的目录中运行该命令。该节点必须处于非群集模式。

仅可以在全局区域中使用此选项。

-u

在从中调用 `scinstall` 命令的节点上升级 Oracle Solaris Cluster 软件。`scinstall` 的升级格式有多种操作模式，具体由 `upgrade-mode` 指定。有关特定于您要执行的升级类型的信息，请参见下面的“升级选项”。

仅可以在全局区域中使用此选项。

附加选项

您可以将附加选项与基本选项结合使用，以修改每种命令格式的默认行为。有关其中哪些选项能够合法地与哪些 `scinstall` 命令格式结合使用的更多详细信息，请参阅“用法概要”部分。

支持下列附加选项：

-h *nodename*

指定节点名称。-h 选项只有与命令的“添加安装客户机”(-c) 格式结合使用时才合法。*nodename* 是要为定制 AI 安装设置的群集节点（即 AI 安装客户机）的名称。

-v

以详细模式显示发行版信息。-v 选项只有与命令的“显示发行版”(-p) 格式结合使用时才合法，可用于指定详细模式。

在“显示发行版”的详细模式下，还会显示安装的每个 Oracle Solaris Cluster 软件包的版本字符串。

-F [*config-options*]

在群集中建立第一个节点。-F 选项只有与命令的“初始化”(-i) 或“添加安装客户机”(-c) 格式结合使用时才合法。

在第一个节点已完全实例化为群集成员，并已准备好执行所有与添加新群集节点关联的必要任务之前，将会阻止建立辅助节点。如果将 -F 选项与 -o 选项一起使用，则会在群集创建过程中创建一个单节点群集，且无法添加任何其他节点。

-f *hostnames-map-file*

指定包含旧主机名和新主机名对列表的文本文件，这些主机名对将用来从其他群集来复制群集，或者将一个恢复归档文件与 `initial` 操作一起使用来构建一个全新的群集。该文件可以包含多行，每行包含两列。第一列是在其中创建归档文件的源群集中所使用的主机名或 IP 地址。第二列是新群集的对应主机名或 IP 地址。这些主机名可以用于逻辑主机名、共享地址资源或区域群集。

```
source-cluster-zc-hostname1      target-cluster-zc-hostname1
source-cluster-zc-hostname2      target-cluster-zc-hostname2
source-cluster-lh1               target-cluster-lh1
source-cluster-lh2               target-cluster-lh2
```

仅可以在全局区域中使用此选项。

`-N cluster-member [config-options]`

指定群集成员。`-N` 选项只有与命令的“初始化”(`-i`)、“添加安装客户机”(`-c`)或“删除”(`-r`)格式结合使用时才合法。

与 `-i` 或 `-c` 选项一起使用时，`-N` 选项将用于向现有群集添加其他节点。指定的 `cluster-member` 通常是群集建立的第一个群集节点的名称。但是，`cluster-member` 可以是已作为群集成员参与的任何群集节点的名称。正在初始化的节点将被添加到 `cluster-member` 已是其活动成员的群集。将新节点添加到现有群集的过程包括更新指定的 `cluster-member` 上的配置数据，以及在新节点的本地文件系统上创建配置数据库的副本。

与 `-r` 选项一起使用时，`-N` 选项将会指定 `cluster-member`，这可以是群集中作为活动群集成员的其他任何节点。`scinstall` 命令将会联系指定的 `cluster-member`，以便对群集配置进行更新。如果不指定 `-N` 选项，`scinstall` 将会尽量尝试找到可联系的现有节点。

配置选项

`config-options` 与 `-F` 选项一起使用。

```
/usr/cluster/bin/scinstall{-i | -c net-image-source -U password-file -h nodename -n nodeip-mac-options -W software-spec} -F [-C clustername] [-T authentication-options] [-A adapter-options] [-B switch-options] [-m endpoint=[this-node]:name[@port],endpoint=[node:]name[@port]] [-o] [-w netaddr-options]
```

```
/usr/cluster/bin/scinstall {-i | -c net-image-source -U password-file -h nodename -n nodeip-mac-options -W software-spec} -N cluster-member [-C clustername] [-A adapter-options] [-B switch-options] [-m endpoint=cable-options]
```

`-m cable-options`

指定群集互连连接。仅当同时指定了 `-F` 或 `-N` 选项时，该选项才合法。

`-m` 选项可以通过配置用于连接群集传输适配器和交换机上各个端口的电缆，来帮助建立群集互连拓扑。使用此命令格式配置的每条新电缆将在当前节点上的群集传输适配器与群集传输交换机上的某个端口，或者群集中另一个现有节点上的某个适配器之间建立连接。

如果不指定 `-m` 选项，`scinstall` 命令将尝试配置默认电缆。但是，如果使用 `scinstall` 的某个给定实例配置了多个传输适配器或交换机，`scinstall` 将无法构建默认设置。默认设置是配置一条从单独配置的传输适配器到单独配置（或默认）的传输交换机的电缆。

`-m cable-options` 如下所示。

```
-m endpoint=[this-node]:  
name[@port],endpoint=[  
node:]name[@port]
```

`-m` 选项的语法表明，在两个端点中，必须至少有一个是要配置的节点上的适配器。对于该端点，不需要明确指定 `this-node`。下面是添加电缆的示例：

```
-m endpoint=:net1,endpoint=switch1
```

在本示例中，该节点（即 `scinstall` 要配置的节点）上的 `net1` 传输适配器的端口 0 已通过电缆连接到传输交换机 `switch1` 上的某个端口。`switch1` 上使用的端口号默认为该节点的节点 ID 号。

对于出现的每个 `-m` 选项，必须始终指定两个 `endpoint` 选项。选项参数的 `name` 组成部分用于指定电缆某一个端点上的群集传输适配器或群集传输交换机的名称。

- 如果指定 `node` 组成部分，则 `name` 将是传输适配器的名称。
- 如果不指定 `node` 组成部分，则 `name` 将是传输交换机的名称。

如果不指定 `port` 组成部分，`scinstall` 命令将尝试采用默认端口名称。适配器的默认 `port` 始终为 0。交换机端点的默认端口 `name` 等于要添加到群集的节点的节点 ID。

有关 `port` 分配和其他要求的更多信息，请参阅 [clinterconnect\(1CL\) \[97\]](#) 手册页。

必须先配置电缆两个端点上的适配器和/或交换机（参见 -A 和 -B），然后才能配置电缆。

`-n nodeip-mac-options`

指定节点的 IP 地址和 MAC 地址。仅当同时指定了 `-c` 选项时，该选项才合法。

`-n nodeip-mac-options` 的语法如下所示：

```
-n ip=node-ipaddr/N,mac=
mac-address
```

`-o`

指定单节点群集的配置。仅当同时还指定了 `-i` 和 `-F` 选项时，该选项才合法。

其他 `-F` 选项也受支持，但不是必需的。如果未指定群集名称，则会将节点名称用作群集名称。可以指定传输配置选项，这些选项将存储在 CCR 中。建立单节点群集后，不必配置法定设备或禁用 `installmode`。

`-w netaddr-options`

指定用于专用互连或群集传输的网络地址。仅当同时指定了 `-F` 选项时，该选项才合法。

使用该选项可以指定要在专用互连上使用的专用网络地址。当默认的专用网络地址与企业中已用的某个地址发生冲突时，可以使用该选项。还可以使用该选项来定制预留给专用互连使用的 IP 地址范围的大小。有关更多信息，请参见 [Unresolved link to "networks4"](#) 和 [Unresolved link to "netmasks4"](#) 手册页。

如果不指定该选项，用于专用互连的默认网络地址将为 `172.16.0.0`。默认网络掩码为 `255.255.240.0`。此 IP 地址范围最多支持 62 个节点、10 个专用网络、12 个区域群集以及三个独占 IP 区域群集。

`-w netaddr-options` 如下所示：

```
-w netaddr=netaddr[,netmask=
netmask]
```

```
-w netaddr=netaddr[,maxnodes=nodes,maxprivatenets=maxprivnets,\
numvirtualclusters=zoneclusters, numxipvirtualclusters=xipzoneclusters]
-w netaddr=netaddr[,netmask=netmask,maxnodes=nodes,\maxprivatenets=maxprivnets\
,numvirtualclusters=zoneclusters]
```

netaddr=netaddr

指定专用网络地址。此地址的最后两个八位字节必须始终为零。

[*netmask=netmask*]

指定网络掩码。指定的值必须提供大于或等于默认值的 IP 地址范围。

要分配一个小于默认值的 IP 地址范围，请指定 *maxnodes*、*maxprivatenets* 和 *numvirtualclusters* 操作数。

[,maxnodes=*nodes*,maxprivatenets=*maxprivnets*,numvirtualclusters=*zoneclusters*]

指定群集曾预期包含的最大节点数、专用网络数和区域群集数。该命令使用这些值来计算专用互连支持指定数目的节点、专用网络和区域群集所需的最小网络掩码。*nodes* 的最大值是 62，最小值是 2。*maxprivnets* 的最大值为 128，最小值为 2。可以对 *zoneclusters* 设置 0 值。

[,netmask=*netmask*,maxnodes=*nodes*,maxprivatenets=*maxprivnets*\
,numvirtualclusters=*zoneclusters*]

指定网络掩码以及群集曾预期包含的最大节点数、专用网络数和区域群集数。必须指定一个足以容纳指定数量的 *nodes*、*privnets* 和 *zoneclusters* 的网络掩码。*nodes* 的最大值是 62，最小值是 2。*privnets* 的最大值是 128，最小值是 2。可以为 *zoneclusters* 设置 0 值。

如果仅指定 *netaddr* 子选项，该命令将分配默认网络掩码 255.255.240.0。生成的 IP 地址范围最多可容纳 62 个节点、10 个专用网络和 12 个区域群集。

建立群集后，若要更改专用网络地址或网络掩码，请使用 *cluster* 命令或 *clsetup* 实用程序。

-A adapter-options

指定传输适配器以及（可选）传输类型。仅当同时指定了 *-F* 或 *-N* 选项时，该选项才合法。

每指定一个 *-A* 选项，就会配置一个连接到运行 *scinstall* 命令的节点的群集传输适配器。

如果未指定 *-A* 选项，则会尝试使用默认的适配器和传输类型。默认的传输类型为 *d1pi*。在 SPARC 平台上，默认的适配器为 *hme1*。

如果适配器传输类型为 *d1pi*，则无需指定 *trtype* 子选项。在这种情况下，您可以使用下列两种格式之一来指定 *-A adapter-options*：

```
-A [trtype=type,]name=adaptername[,vlanid=
vlanid][,other-options]
-A adaptername
```

[trtype=type]

指定适配器的传输类型。请将 trtype 选项与每个要指定适配器传输类型的 -A 选项一起使用。dlpi 就是一个传输 type 示例。

默认的传输类型为 dlpi。

name=adaptername

指定适配器名称。要指定 adaptername，必须将 name 子选项与出现的每个 -A 选项一起使用。adaptername 的构造方式为 device name 后面紧跟 physical-unit 编号，例如 hme0。

如果未随 -A 选项指定其他子选项，可以将 adaptername 指定为 -A 选项的独立参数（即 -A adaptername）。

vlanid=vlanid

指定带标记的 VLAN 适配器的 VLAN ID。

[other-options]

指定其他适配器选项。如果某个特定适配器提供其他任何选项，则可以使用 -A 选项指定这些选项。

-B switch-options

指定传输交换机（又称为传输结点）。仅当同时指定了 -F 或 -N 选项时，该选项才合法。

每指定一个 -B 选项，就会配置一个群集传输交换机。这种设备的示例包括但不限于以太网交换机、其他各种类型的交换机和环网。

如果不指定 -B 选项，将第一个节点实例化为群集节点时，scinstall 将会尝试添加默认交换机。默认情况下，当您其他节点添加到群集时，不会添加任何其他交换机。但是，您可以明确添加这些交换机。默认交换机的名称为 switch1，类型为 switch。

如果交换机类型为 switch，则无需指定 type 子选项。在这种情况下，您可以使用下列两种格式之一来指定 -B switch-options。

```
-B [type=type,]name=name[,
other-options]
-B name
```

如果某个群集传输交换机已配置为指定的交换机 name，则 scinstall 会显示消息并忽略 -B 选项。

如果使用直接以电缆连接的传输适配器，则不需要配置任何传输交换机。为避免配置默认传输交换机，请使用以下特殊 -B 选项：

`-B type=direct`

[`type=type`]

指定传输交换机类型。可以将 `type` 选项与出现的每个 `-B` 选项一起使用。以太网交换机就是类型为 `switch` 的群集传输交换机的示例。有关更多信息，请参见 [clinterconnect\(1CL\) \[97\]](#) 手册页。

可以将 `type` 子选项指定为 `direct`，以抑制对任何默认交换机进行配置。仅包括直连传输适配器的传输配置中不存在交换机。如果 `type` 子选项设置为 `direct`，则无需使用 `name` 子选项。

`name=name`

指定传输交换机名称。除非 `type` 为 `direct`，否则，必须将 `name` 子选项与出现的每个 `-B` 选项一起使用才能指定传输交换机 `name`。`name` 的长度最多可为 256 个字符，由字母或数字组成，第一个字符为字母。每个传输交换机名称在群集名称空间中必须唯一。

如果不需要将其他子选项与 `-B` 一起使用，可以将交换机 `name` 指定为 `-B` 的独立参数（即 `-B name`）。

[`other-options`]

指定其他传输交换机选项。如果某个特定交换机类型提供其他选项，则可以使用 `-B` 选项指定这些选项。有关可以用于交换机的任何特殊选项的信息，请参阅 [clinterconnect\(1CL\) \[97\]](#) 手册页。

`-C clustername`

指定群集的名称。仅当同时指定了 `-F` 或 `-N` 选项时，该选项才合法。

- 如果配置的节点是新群集中的第一个节点，则默认的 `clustername` 与您配置的节点的名称相同。
- 如果您配置的节点正被添加到现有群集，则默认的 `clustername` 为 `cluster-member` 所属的群集的名称。

如果指定的 `clustername` 不同于 `cluster-member` 所属的群集的名称，则会出现错误。

`-T authentication-options`

指定群集的节点验证选项。仅当同时指定了 `-F` 选项时，该选项才合法。

使用该选项可为尝试将自身添加到群集配置的节点建立验证策略。具体而言，就是当某台计算机请求将自身作为群集节点添加到群集时，执行检查以确定该节点是否有权加入。如果请求加入的节点具有权限，则该节点将通过验证，并可以加入群集。

设置群集中的第一个节点时，只能将 `-T` 选项与 `scinstall` 命令一起使用。如果需要已在建立的群集上更改验证列表或策略，请使用 `claccess` 命令。

默认设置是允许任何计算机将自身添加到群集。

-T *authentication-options* 如下所示。

```
-T node=nodename[,...][,authtype=authtype][,secureAI=yes]
-T node=archive-source-node::node-to-install[,...][,authtype=authtype][,secureAI=yes]
-T secureAI=yes
```

node=*nodename*[,...]

指定要添加到节点验证列表的节点名称。对于要复制的所有节点必须指定节点对。必须为 -T 选项至少指定一个 *node* 子选项。该选项用于将节点名称添加到可将自身配置为群集中节点的节点的列表。如果验证列表为空，任何节点都可以请求将自身添加到群集配置。但是，如果列表至少包含一个名称，将会使用验证列表验证所有这种请求。您随时可以使用 `claccess` 命令或 `clsetup` 实用程序从某一个活动的群集节点修改或清除此节点列表。

node=*archive-source-node*::*node-to-install*[,...]

node=*archive-source-node*::*node-to-install* 选项指定节点名称对。第一个节点名称是在其中创建归档文件的节点，第二个节点名称是要从该归档文件安装的新群集中的节点。只有当从一个群集上所创建的归档文件复制其他群集时才使用此规范，新群集节点的硬件配置必须与在其中创建归档文件的源群集节点的硬件配置相同（或者是源群集节点的硬件配置的超集）。

[*authtype*=*authtype*]

为 `initialize -i` 或 `add install client -c` 格式（在使用 *net-image-source* 时）格式的 `scinstall` 群集配置指定节点验证类型。当前支持的 *authtype* 只有 `des` 和 `sys`（或 `unix`）。如果不指定 *authtype*，默认值为 `sys`。

您可以根据需要指定 `des` (Diffie-Hellman) 验证，先将条目加入到每个要添加的群集节点的 [Unresolved link to "publickey4"](#) 数据库，然后运行 `scinstall` 命令的 -T 选项。

您随时可以使用 `claccess` 命令或 `clsetup` 实用程序从某一个活动的群集节点更改验证类型。

[*secureAI*=yes]

指定使用 AI 安全安装，仅当使用 AI 安装群集软件时有效。如果没有指定 *secureAI*=yes，默认操作是执行传统 AI 安装。在使用安全安装方法从归档文件恢复节点时，只需指定 -T *secureAI*=yes。不需要指定 *node*=*nodename*[,...] 和 *authtype*=*authtype*。

-U *password-file*

指定包含 root 用户密码的文件的名称。仅当同时指定了 -c 选项时，该选项才合法。在 Oracle Solaris 的初始安装和配置期间，该选项可以实现 root 用户密码的自动设置。用户将创建一个文件，其中包含用作所安装系统的 root 用户密码的文本。通常，*password-file* 位于已经建立的 AI 安装服务器上，或者可以通过该服务器访问。该服务器已配置用于安装 *nodename* 安装客户机。`scinstall` 实用程序将会检索该文件的内容，并将这些内容提供给 Oracle Solaris 配置实用程序。

`-w software-specs`

指定一个或多个发行者和软件包系统信息库的位置。此外，还指定使用 AI 安全安装所需的公钥和 SSL 证书信息。此选项仅在将 `-c` 选项指定为从 IPS 系统信息库安装时才合法。

`-w software-specs` 如下所示：

```
-w publisher=  
repo[:key=  
key-file::cert=certificate-file] \  
::pkg[,...][:::  
publisher=repo[:key=key-file::cert=  
certificate-file]::pkg[,...]...
```

请注意，为方便阅读，`-w` 选项跨多行显示，不过您应该使用一个未换行的字符串指定该选项。

在 `-w` 选项语法中，`publisher` 是发布者名称 `ha-cluster` 或 `solaris`，`repo` 是系统信息库位置，`key-file` 和 `certificate-file` 是通过 HTTPS 系统信息库安全安装所需的公钥和 SSL 证书信息，`pkg` 是软件包名称。

要使用安全的 HTTPS 系统信息库安装 Oracle Solaris 或 Oracle Solaris Cluster，您需要提供公钥和 SSL 证书的信息。您可以从 <http://pkg-register.oracle.com> 站点请求和下载公钥和 SSL 证书。

升级选项

用于标准（非滚动）升级、滚动升级和双分区升级的 `-u upgrade-modes` 和 `upgrade-options` 如下所述。

标准（非滚动）和滚动升级

使用 `-u update` 模式可以在标准（非滚动）升级或滚动升级模式下，将群集节点升级到更高的 Oracle Solaris Cluster 软件发行版。

- 标准（或非滚动）升级过程可以升级某个非活动的引导环境 (boot environment, BE)，同时，您的群集节点可继续为群集请求提供服务。如果未指定现有的非活动 BE，`scinstall` 实用程序将自动创建一个新 BE。完成升级后，`scinstall` 实用程序将激活升级的 BE，并通知用户将节点重新引导至升级后的 BE 中。
- 滚动升级过程一次只会使一个群集节点脱离生产环境。此过程只能用于将 Oracle Solaris 和/或 Oracle Solaris Cluster 软件升级到已安装版本的更新发行版。当您升级某个节点时，群集服务将继续在其余群集节点上运行。升级某个节点后，您可以将它重新加入群集，并针对要升级的下一个节点重复该过程。升级所有节点后，必须对某一个群集节点运行 `scversions` 命令，以将群集提升到升级版本。在运行此命令之前，更新发行版中引入的某些新功能可能不可用。
- （可选）可以指定当前映像中已经安装的软件包 FMRI。

适用于标准和滚动模式的 `-u update` 的 `upgrade-options` 如下所示。

```
/usr/cluster/bin/scinstall -u update [-b be-name] [-L {accept | licenses | accept,licenses  
| licenses,accept}] [pkg_fmri_pattern ...]
```

`-b be-name`

指定要分配给新引导环境 (boot environment, BE) 的名称。如果不指定该选项, `scinstall` 将为新 BE 分配名称。该名称基于当前 BE 的名称, 格式为 `currentBE-N`, 其中后缀 `-N` 是一个递增编号。第一个新 BE 的名称为 `currentBE-1`, 第二个新 BE 的名称为 `currentBE-2`, 依此类推。如果删除了某个 BE, 并且存在一个后缀编号更大的 BE 名称, 那么, 已删除 BE 的名称不会留给下一个新 BE 重新使用。例如, 如果存在 BE `sc4.0`、`sc4.0-1` 和 `sc4.0-2`, 而 `sc4.0-1` 已被删除, 那么, 下一个新 BE 的名称将是 `sc4.0-3`。

如果指定的 BE 名称已存在, 则命令将会退出并显示一个错误。

`-L {accept | licenses | accept,licenses | licenses,accept }`

指定是否接受和/或显示所升级到的软件包的许可证。

`accept` 参数对应于 `pkg` 命令的 `--accept` 选项, `licenses` 参数对应于 `--licenses` 选项。

指定 `-L accept` 选项表示您同意并接受所更新的软件包的许可证。如果有任何软件包要求接受许可证, 而您未提供该选项, 更新操作将失败。

指定 `-L licenses` 显示所更新的软件包的所有许可证。

如果在 `-L` 选项中同时指定了 `accept` 和 `licenses`, 将显示并接受所更新的软件包的许可证。指定 `accept` 和 `licenses` 参数的顺序不会影响该命令的行为。

`scinstall -u update` 命令支持为要更新的软件包指定 `pkg_fmri_patterns` :

`[pkg_fmri_pattern...]`

指定要更新的软件包。当前的映像中必须已安装这些软件包。如果所提供的 `pkg_fmri_pattern` 模式中包含星号 (*), 则会更新当前映像中安装的所有软件包。

双分区升级

用于双分区升级的 `-u upgrade-modes` 和 `upgrade-options` 可以执行双分区升级的多个阶段。在双分区升级过程中, 首先是将群集节点分成两个组或两个分区。接下来, 在升级一个分区的同时, 另一个分区提供群集服务。然后, 将服务切换到已升级的分区, 升级剩余的分区, 将第二个分区中已升级的节点重新加入由升级的第一个分区构成的群集。用于双分区升级的 `upgrade-modes` 还包括在双分区升级期间发生故障后的恢复模式。

双分区升级模式与 `-u update` 升级模式一起使用。有关更多信息, 请参见 [Unresolved link to "Oracle Solaris Cluster Upgrade Guide"](#)。

用于双分区升级的 `-u upgrade-modes` 和 `upgrade-options` 如下所示 :

```
/usr/cluster/bin/scinstall -u begin -h nodelist
```

```
/usr/cluster/bin/scinstall -u plan
```

```
/usr/cluster/bin/scinstall -u recover
```

```
/usr/cluster/bin/scinstall -u status
```

```
/usr/cluster/bin/scinstall -u apply
```

```
/usr/cluster/bin/scinstall -u status
```

apply

指定一个分区的升级已完成。在已升级的分区中的所有节点都升级后，在该分区中的任意节点上运行这种格式的命令。

apply 升级模式执行以下任务：

第一个分区

从第一个分区中的节点上运行时，apply 升级模式会使第一个分区中的所有节点准备好运行新软件。

当第一个分区中的节点准备好支持群集服务时，该命令会在第二个分区中的节点上远程执行脚本 `/etc/cluster/ql/cluster_pre_halt_apps` 和 `/etc/cluster/ql/cluster_post_halt_apps`。这些脚本用于调用用户编写的脚本，这样可以停止不受资源组管理器 (Resource Group Manager, RGM) 控制的应用程序，例如 Oracle Real Application Clusters (Oracle RAC)。

- `cluster_pre_halt_apps` 脚本在受 RGM 控制的应用程序停止之前运行。
- `cluster_post_halt_apps` 脚本在受 RGM 控制的应用程序停止之后、节点停止之前运行。

注 - 运行 apply 升级模式之前，请根据需要修改脚本模板，以调用您编写的其他脚本来停止节点上的某些应用程序。将修改后的脚本以及修改后的脚本调用的用户编写的脚本放在第一个分区中的每个节点上。这些脚本将从第一个分区中的任意一个节点上运行。要停止在第一个分区中多个节点上运行的应用程序，请相应地修改用户编写的脚本。未修改的脚本不执行任何默认操作。

第二个分区上的所有应用程序均停止后，该命令会停止第二个分区中的节点。关闭操作会启动切换，将应用程序和数据服务切换到第一个分区中的节点。然后，该命令将第二个分区中的节点引导到群集模式。

如果某个资源组由于其节点列表仅包含第一个分区的成员而脱机，则该资源组将重新联机。如果某个资源组的节点列表中没有属于第一个分区的节点，则该资源组仍处于脱机状态。

第二个分区

从第二个分区中的节点上运行时，apply 升级模式会使第二个分区中的所有节点准备好运行新软件。然后，该命令将节点引导到群集模式。第二个分区中的节点重新加入由第一个分区中的节点构成的活动群集。

如果某个资源组由于其节点列表仅包含第二个分区的成员而脱机，则该资源组将重新联机。

所有节点均重新加入群集后，该命令将执行最终处理，重新配置法定设备，并恢复法定投票计数。

begin

指定要分配给您升级的第一个分区的节点，并启动双分区升级过程。从群集中的任意节点上运行这种格式的命令。在使用 plan 升级模式后使用此升级模式，可以确定可能的分区方案。

首先，begin 升级模式记录要分配给每个分区的节点。接下来，在一个节点上停止所有应用程序，然后升级模式关闭该节点。关闭操作会启动切换，将该节点上的每个资源组切换到属于第二个分区的节点上，前提是该节点在资源组节点列表中。如果某个资源组的节点列表中没有属于第二个分区的节点，则该资源组仍处于脱机状态。

然后，该命令在第一个分区中剩余的每个节点上重复这一系列操作，一次处理一个节点。

在第一个分区升级期间，第二个分区中的节点一直正常运行。法定设备暂时不配置，法定投票计数在节点上暂时发生变化。

plan

查询群集存储配置，并显示满足共享存储要求的所有可能的分区方案。从群集中的任意节点上运行这种格式的命令。这是您在双分区升级过程中运行的第一个命令。双分区升级要求每个共享存储阵列都必须至少可由每个分区中的一个节点进行物理访问。

plan 升级模式可以返回零个、一个或多个分区解决方案。如果未返回任何解决方案，则群集配置不适合双分区升级。请改用标准升级方法。

对于任何分区解决方案，您都可以选择任一分区组作为要升级的第一个分区。

recover

如果在双分区升级过程中出现致命错误，请恢复节点上的群集配置。在群集中的每个节点上运行这种格式的命令。

您必须在运行该命令之前关闭群集并将所有节点引导到非群集模式。

一旦发生致命错误，即使在运行 recover 升级模式后，也无法恢复或重新启动双分区升级。

recover 升级模式可以将群集配置系统信息库 (Cluster Configuration Repository, CCR) 数据库恢复到双分区升级开始之前的原始状态。

以下列表描述在哪些情况下使用 recover 升级模式，以及在哪些情况下执行其他步骤。

- 如果在 -u begin 处理过程中发生故障，请运行 -u recover 升级模式。
- 如果在 -u begin 处理完成之后、发出第二个分区的关闭警告之前发生故障，请确定出错的位置：
 - 如果在第一个分区中的某个节点上发生故障，请运行 -u recover 升级模式。
 - 如果在第二个分区中的某个节点上发生故障，则无需执行任何恢复操作。
- 如果在发出第二个分区的关闭警告之后、第二个分区上开始 -u apply 处理过程之前发生故障，请确定出错的位置：

-
- 如果在第一个分区中的某个节点上发生故障，请运行 `-u recover` 升级模式。
 - 如果在第二个分区中的某个节点上发生故障，请将发生故障的节点重新引导到非群集模式。
 - 如果在第二个分区上完成 `-u apply` 处理过程之后、升级完成之前发生故障，请确定出错的位置：
 - 如果在第一个分区中的某个节点上发生故障，请运行 `-u recover` 升级模式。
 - 如果在第一个分区中的某个节点上发生故障，但是第一个分区一直在提供服务，请重新引导发生故障的节点。
 - 如果在第二个分区中的某个节点上发生故障，请运行 `-u recover` 升级模式。

在所有情况下，您都可以采用标准升级方法继续手动升级，该操作要求关闭所有群集节点。

`status`

显示双分区升级的状态。可能的状态如下所示：

Upgrade is in progress

`scinstall -u begin` 命令已运行，但是双分区升级尚未完成。

如果在双分区升级过程中出现致命错误，群集也会报告该状态。在这种情况下，即使在执行恢复过程并采用标准升级方法完成群集升级后，该状态也不会清除

Upgrade not in progress

尚未发出 `scinstall -u begin` 命令，或者双分区升级已成功完成。

从群集中的一个节点上运行 `status` 升级模式。节点可以处于群集模式或非群集模式。

无论发出命令的节点处于双分区升级过程中的哪个阶段，报告的状态对于群集中的所有节点都有效。

双分区升级模式支持以下选项：

`-h nodelist`

指定您分配给第一个分区的所有节点的空格分隔列表。您应从 `plan` 升级模式显示的输出中选择这些节点，将其作为您使用的分区方案中某个分区的有效成员。群集中您未指定给 `begin` 升级模式的其余节点将分配给第二个分区。

该选项仅对于 `begin` 升级模式有效。

建立双节点群集

以下示例使用 Oracle Solaris Cluster 软件为基于 SPARC 的平台上的 Oracle Solaris 11 建立一个典型的双节点群集。该示例假定 Oracle Solaris Cluster 软件包已安装在节点上。

在 node1 上，发出以下命令：

```
node1# /usr/cluster/bin/scinstall -i -F
```

在 node2 上，发出以下命令：

```
node2# /usr/cluster/bin/scinstall -i -N node1
```

建立单节点群集

以下命令将在接受所有默认值的情况下，使用 Oracle Solaris Cluster 软件为基于 SPARC 的平台上的 Oracle Solaris 11 建立一个单节点群集。该示例假定 Oracle Solaris Cluster 软件包已安装在节点上。

```
# /usr/cluster/bin/scinstall -i -F -o
```

在 AI 服务器上添加具有网络映像 ISO 文件的安装客户机

以下示例在某个双节点群集中，为基于 SPARC 的平台上的 Oracle Solaris 11 设置一个 AI 安装服务器，以安装并初始化 Oracle Solaris Cluster 软件。

在安装服务器上，发出以下命令。请注意，为方便阅读，-w 选项跨多行显示，不过您应该使用一个未换行的字符串指定该选项。

```
# usr/cluster/bin/scinstall -c /export/home/11-ga-ai-x86.iso -h phys-schost-1 \  
-U /export/pwdfile \  
-C schost \  
-F \  
-W solaris=http://ipkg.us.oracle.com/solaris11/release::  
entire,server_install::  
ha-cluster=cluster-repository::  
ha-cluster-framework-full,ha-cluster-data-services-full,  
ha-cluster-geo-full \  
-n ip=10.255.85.163/24,mac=12:34:56:78:90:ab \  
-T node=phys-schost-1,node=phys-schost-2,authtype=sys \  
-w netaddr=172.16.0.0,netmask=255.255.240.0,maxnodes=62,\  
maxprivatenets=10,numvirtualclusters=12,numxipvirtualclusters=3 \  
-A trtype=dlpi,name=e1000g1 -A trtype=dlpi,name=nxge1 \  
-B type=switch,name=switch1 -B type=switch,name=switch2 \  
-m endpoint=:e1000g1,endpoint=switch1 \  
-m endpoint=:nge1,endpoint=switch2  
  
# usr/cluster/bin/scinstall -c /export/home/11-ga-ai-x86.iso -h phys-schost-2 \  
-U /export/pwdfile \  
-C schost \  
-N phys-schost-1 \  
-W solaris=http://ipkg.us.oracle.com/solaris11/release::  
entire,server_install::  
ha-cluster=cluster-repository::  
ha-cluster-framework-full,ha-cluster-data-services-full,\  
ha-cluster-geo-full \  
-n ip=10.255.85.164/24,mac=12:34:56:78:90:ab \  
-A trtype=dlpi,name=e1000g1 -A trtype=dlpi,name=nxge1 \  

```

```
-m endpoint=:e1000g1,endpoint=switch1 \  
-m endpoint=:nge1,endpoint=switch2
```

执行双分区升级

以下示例使用双分区方法将某个群集的框架和数据服务软件升级到下一个 Oracle Solaris Cluster 发行版。本例在基于 SPARC 的平台上使用适用于 Solaris 11 的 Oracle Solaris Cluster 版本。本例查询群集的有效分区方案，将节点分配给分区，重新引导第一个分区中的节点，在第一个分区升级后使其恢复正常运行并重新引导第二个分区中的节点，在第二个分区升级后使其返回群集。

```
# /usr/cluster/bin/scinstall -u plan  
Option 1  
  First partition  
    phys-schost-1  
  Second partition  
    phys-schost-2  
...  
# /usr/cluster/bin/scinstall -u begin -h phys-schost-1 phys-schost-3
```

```
ok boot -x
```

(Upgrade the node in the first partition)

```
phys-schost-1# /usr/cluster/bin/scinstall -u apply  
ok boot -x
```

(Upgrade the node in the second partition)

```
phys-schost-2# /usr/cluster/bin/scinstall -u apply
```

升级框架和数据服务软件（标准升级或滚动升级）

以下示例将某个群集的框架和数据服务软件升级到下一个 Oracle Solaris Cluster 发行版。请在每个群集节点上执行这些操作。

注 - 对于滚动升级，请在使用 `clnode evacuate` 命令将所有资源组和设备组移到要保留在群集中的其他节点之后，一次在一个节点上执行这些操作。

```
# /usr/cluster/bin/scinstall -u update  
# init 6
```

从归档文件恢复第一个节点

以下示例使用一个安全的 AI 安装从保存在文件系统（可从 AI 服务器访问）上的归档文件恢复第一个节点。

```
# /usr/cluster/bin/scinstall -c archive=file:///net/storagenode/export/archive  
  /phys-schost-1-recovery-archive,action=restore \  
-h phys-schost-1 \  

```

```
-C schost =\  
-F \  
-n ip=10.255.85.163/24,mac=12:34:56:78:90:ab \  
-T secureAI=yes
```

从归档文件恢复其他节点

以下示例使用一个安全的 AI 安装从以前在其他节点上创建的归档文件恢复这些其他节点。

```
# /usr/cluster/bin/scinstall -c archive=file:///net/storagenode/export/archive  
/phys-schost-2-recovery-archive,action=restore \  
-h phys-schost-2 \  
-C schost =\  
-n ip=10.255.85.164/24,mac=12:34:56:78:90:cd \  
-T secureAI=yes
```

执行非安全复制

以下示例执行非安全复制。

```
# /usr/cluster/bin/scinstall -c archive=file:///net/storagenode/export/archive  
/source-node-1-archive,action=replicate \  
-h phys-schost-1 \  
-C schost \  
-F \  
-n ip=10.255.85.163/24,mac=12:34:56:78:90:ab \  
-T node=phys-schost-1,node=phys-schost-2,secureAI=yes \  
-U /export/pwdfile  
  
# /usr/cluster/bin/scinstall -c archive=file:///net/pnass3/export/archive  
/vzonola.clone,action=replicate \  
-h phys-schost-2 \  
-C schost \  
-n ip=10.255.85.164/24,mac=12:34:56:78:90:cd \  
-U /export/pwdfile
```

在 AI 服务器上添加具有 IPS 系统信息库的安装客户机

下面的几个示例使用一个安全的 AI 安装从 IPS 系统信息库安装和配置双节点 x86 群集。

```
# /usr/cluster/bin/scinstall -c solaris=http://ipkg.us.oracle.com/solaris11  
/release::arch=i386 -h phys-schost-1 \  
-C schost \  
-F \  
-W solaris=http://ipkg.us.oracle.com/solaris11/release::entire,server_install::  
ha-cluster=http://ipkg.us.oracle.com/ha-cluster/release::ha-cluster-framework-full \  
-n ip=10.255.85.163/24,mac=12:34:56:78:90:ab \  
-T node=phys-schost-1,node=phys-schost-2,authtype=sys,secureAI=yes \  
-w netaddr=172.16.0.0,netmask=255.255.240.0,maxnodes=32,maxprivatenets=10,  
numvirtualclusters=12,numxipvirtualclusters=3 \  

```

```

-A trtype=dlpi,name=net1 -A trtype=dlpi,name=net3 \
-B type=switch,name=switch1 -B type=switch,name=switch2 \
-m endpoint=:net1,endpoint=switch1 \
-m endpoint=:net3,endpoint=switch2 \
-P task=quorum,state=INIT -P task=security,state=SECURE \
-U /export/pwdfile

# /usr/cluster/bin/scinstall -c solaris=http://ipkg.us.oracle.com/solaris11
  /release::arch=i386 -h phys-schost-2 \
-C schost \
-N phys-schost-1 \
-W solaris=http://ipkg.us.oracle.com/solaris11/release::entire,server_install::
  ha-cluster=http://ipkg.us.oracle.com/ha-cluster/release::ha-cluster-framework-full
\
-n ip=10.255.85.164/24,mac=12:34:56:78:90:ab \
-A trtype=dlpi,name=net1 -A trtype=dlpi,name=net3 \
-m endpoint=:net1,endpoint=switch1 \
-m endpoint=:net3,endpoint=switch2 \
-U /export/pwdfile

```

返回以下退出值：

0 成功完成。

non-zero 发生错误。

/etc/cluster/ql/cluster_post_halt_apps

/etc/cluster/ql/cluster_pre_halt_apps

有关下列属性的说明，请参见 [Unresolved link to " attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/install
接口稳定性	Evolving (发展中)

[Intro\(1CL\) \[17\]](#)、[claccess\(1CL\) \[31\]](#)、[clinterconnect\(1CL\) \[97\]](#)、[clnode\(1CL\) \[149\]](#)、[clsetup\(1CL\) \[407\]](#)、[link to " newfs1M"](#)、[scversions\(1M\) \[767\]](#)、[Unresolved link to " netmasks4"](#)、[Unresolved link to " networks4"](#)、[Unresolved link to " lofi7D"](#)

[Unresolved link to " Oracle Solaris Cluster 软件安装指南 "](#)、[Unresolved link to " Oracle Solaris Cluster 系统管理指南 "](#)、[Unresolved link to " Oracle Solaris Cluster Upgrade Guide "](#)

名称

scnas — 管理 Oracle Solaris Cluster 的网络连接存储 (network-attached storage, NAS) 设备配置数据。

```
scnas [-H]

scnas -a [-H] [-n] -h device-name -t device-type -o
      specific-options [-f input-file]

scnas -c [-H] [-n] -h device-name -o specific-options [-f
      input-file]

scnas -p [-H] [-h device-name] [-t device-type]

scnas -r [-H] -h device-name
```

注 - Oracle Solaris Cluster 软件包括面向对象的命令集。虽然 Oracle Solaris Cluster 软件仍支持原始的命令集，但是 Oracle Solaris Cluster 过程文档仅使用面向对象的命令集。有关面向对象的命令集的更多信息，请参见 [Intro\(1CL\) \[17\]](#) 手册页。

scnas 命令管理 Oracle Solaris Cluster 配置中的 NAS 设备。要管理群集中的 NAS 目录，请使用 scnasdir 命令。

可以使用 scnas 命令创建 NAS 设备配置，更新 NAS 类型特定的属性，以及从 Oracle Solaris Cluster 中删除设备配置。此命令的选项按照在命令行上的键入顺序进行处理。

只能从活动群集节点上运行 scnas 命令。运行命令的结果始终是相同的，而不管使用哪个节点。

scnas 命令的所有格式都接受 -H 选项。指定 -H 将显示帮助信息。所有其他选项都将被忽略。在没有选项的情况下运行 scnas 时也显示帮助信息。

在使用 scnas 命令管理 NAS 设备之前，必须设置 NAS 设备。有关设置设备的过程，请参阅特定 NAS 设备的文档。

仅可以在全局区域中使用此命令。

基本选项

对于 scnas 命令的所有格式，以下选项是通用的：

-H

如果在命令行上的任何位置指定此选项，则命令将显示帮助信息。其他所有选项将被忽略且不执行。如果在没有选项的情况下运行 scnas，则也会显示帮助信息。

仅可以在全局区域中使用此选项。

-n

如果在命令行上的任何位置指定此选项，则 `scnas` 命令仅检查用法而不写入配置数据。如果 -n 选项是与 -f 选项一起指定的，则 `scnas` 命令将检查密码的输入文件。

以下选项用于修饰 `scnas` 命令的基本格式和功能。不能在同一命令行上组合使用这些选项。

-a

指定 `scnas` 命令的 `add` 格式。

仅可以在全局区域中使用此选项。

-a 选项可用于将 NAS 设备添加到 Oracle Solaris Cluster 配置中。

可能还需要设置其他属性，具体取决于您的 NAS 设备类型。“附加选项”部分的 -t 选项描述中也介绍了这些必需属性。

-c

指定 `scnas` 命令的 `change` 格式。-c 选项用于更改特定的 NAS 设备属性。

仅可以在全局区域中使用此选项。

-r

指定 `scnas` 命令的 `remove` 格式。-r 选项用于从 Oracle Solaris Cluster 配置中删除 NAS 设备。

仅可以在全局区域中使用此选项。

在删除设备之前，必须通过使用 `scnasdir` 删除其所有的已导出目录。

-p

指定 `scnas` 命令的 `print` 格式。

仅可以在全局区域中使用此选项。

如果未提供其他选项，则 -p 选项将显示在 Oracle Solaris Cluster 中配置的所有当前 NAS 设备及其关联属性的列表。此选项可与其他选项一起使用以查询特定的设备或特定类型的设备。

附加选项

以下附加选项可与前面描述的一个或多个基本选项组合使用以配置设备的所有属性。设备无需联机即可使用这些选项。请参阅 [用法概要](#) 部分以查看可与 `scnas` 的每种格式一起使用的选项。

附加选项如下：

-h *device-name*

使用此选项可在 Oracle Solaris Cluster 配置中指定 NAS 设备的名称。设备名称标识设备并可用于远程访问设备（通过使用 `rhs` 或 `telnet`）。

必须为 `scnas` 命令的 `add`、`change` 和 `remove` 格式指定此设备名称。

-t device-type

NAS 设备类型。将 NAS 设备添加到 Oracle Solaris Cluster 配置中时，必须指定此选项。NAS 设备类型通过供应商名称进行标识。

您可以为 Oracle Sun ZFS Storage Appliance 指定 `sun_uss`。

不同类型的 NAS 设备具有不同的属性，有时不具有任何属性。

-o specific-options

使用此选项可提供特定于 NAS 设备类型的属性。

`userid` 属性由群集使用以在设备上执行管理职责。将 `userid` 添加到设备配置时，系统会提示您提供其密码。您也可以将密码置于文本文件中并通过指定 `-f` 选项进行使用。

-f input-file

为安全起见，无法在命令行选项中指定密码。为了确保密码安全，请将其置于文本文件中并通过使用 `-f` 选项指定该文件。如果未指定密码的输入文件，此命令会提示您提供密码。

将输入文件的权限设置为可由 `root` 用户读取并禁止组访问或全局访问。

在输入文件中，不能跨多行输入密码。忽略前导空格和制表符。注释以不带引号的磅 (`#`) 符号开头，并继续到下一新行。

解析器会忽略所有注释。在使用输入文件提供设备用户密码时，密码中不能包含 `#` 符号。

例 339 将 NAS 设备添加到群集

以下 `scnas` 命令将 NAS 设备添加到 Oracle Solaris Cluster 配置。

```
# scnas -a -h sun_uss1 -t sun
```

例 340 将 NAS 设备添加到群集

以下 `scnas` 命令将存储系统添加到 Oracle Solaris Cluster 配置。

```
# scnas -a -h sun_uss1 -t sun_uss -o userid=root
Please enter password:
```

例 341 从群集中删除 NAS 设备

以下 `scnas` 命令从 Oracle Solaris Cluster 配置中删除 NAS 设备。

```
# scnas -r -h sun_uss1
```

返回以下退出值：

0 命令已成功执行。

非零值 发生错误。

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
稳定性	过时

[Intro\(1CL\) \[17\]](#)、[clnasdevice\(1CL\) \[137\]](#)、[clquorum\(1CL\) \[195\]](#)、[cluster\(1CL\) \[461\]](#)、[sconf\(1M\) \[637\]](#)

名称

scnasdir — 管理 Oracle Solaris Cluster 配置中的网络连接存储 (network-attached storage, NAS) 设备上的导出目录。

```
scnasdir [-H]

scnasdir [-a] [-H] [-n] -h device-name [-d directory [-d
directory...] [-f input-file]

scnasdir -p [-H] [-h device-name] [-t device-type]

scnasdir -r [-H] [-n] -h device-name [-d all | -d directory [-d
directory...] [-f input-file]
```

注 - Oracle Solaris Cluster 软件包括面向对象的命令集。虽然 Oracle Solaris Cluster 软件仍支持原始的命令集，但是 Oracle Solaris Cluster 过程文档仅使用面向对象的命令集。有关面向对象的命令集的更多信息，请参见 [Intro\(1CL\) \[17\]](#) 手册页。

scnasdir 命令管理 Oracle Solaris Cluster 配置中 NAS 设备上的已导出目录。设备必须已通过使用 scnas 命令在群集中进行了配置。

scnasdir 命令可用于将目录添加到设备的群集配置、从设备的群集配置中删除目录以及显示特定设备或特定设备类型的目录。

此命令中的选项按照在命令行上的键入顺序进行处理。只能从活动群集节点上运行 scnasdir 命令。运行命令的结果始终是相同的，而不管使用哪个节点。

scnasdir 命令的所有格式都接受 -H 选项。指定 -H 将会显示帮助信息，其他所有选项将被忽略且不执行。在没有选项的情况下运行 scnasdir 时也显示帮助信息。

仅可以在全局区域中使用此命令。

基本选项

对于 scnasdir 命令的所有格式，以下选项是通用的：

-H

如果在命令行上的任何位置指定此选项，则命令将显示帮助信息。其他所有选项将被忽略且不执行。如果在没有选项的情况下运行 scnasdir，则也会显示帮助信息。仅可以在全局区域中使用此选项。

-n

如果在命令行上的任何位置指定此选项，则 scnasdir 命令仅检查用法而不写入配置数据。如果 -n 选项是与 -f 选项一起指定的，则 scnasdir 命令显示将为用户审核而处理的数据。

以下选项用于修饰 `scnasdir` 命令的基本格式和功能。不能在同一命令行上组合使用这些选项。

`-a`

指定 `scnasdir` 命令的 `add` 格式。`-a` 选项可用于将目录添加到设备的 Oracle Solaris Cluster 配置中。

仅可以在全局区域中使用此选项。

`-p`

指定 `scnasdir` 命令的 `print` 格式。如果未提供其他选项，则此 `-p` 选项将显示在 Oracle Solaris Cluster 中配置的所有 NAS 设备的所有目录的列表。此选项可与其他选项一起使用以查询特定的设备或特定类型的 NAS 设备。

仅可以在全局区域中使用此选项。

`-r`

指定 `scnasdir` 命令的 `remove` 格式。`-r` 选项用于从 Oracle Solaris Cluster 配置中删除 NAS 设备的所有目录或指定目录。

仅可以在全局区域中使用此选项。

附加选项

以下附加选项可与前面描述的一个或多个基本选项组合使用以管理设备的目录。

附加选项如下：

`-h device-name`

使用此选项可在 Oracle Solaris Cluster 配置中指定 NAS 设备的名称。`-h` 选项标识设备并可用于远程访问设备（通过使用 `rhs` 或 `telnet`）。

必须为 `scnasdir` 命令的 `add`、`change` 和 `remove` 格式指定此设备名称。

`-d all | directory`

使用此选项可列出在 NAS 设备上导出以配置到 Oracle Solaris Cluster 中的目录（或卷）。在使用 `scnasdir` 命令之前，必须在设备上创建和导出这些目录。有关导出目录的过程，请参见 NAS 设备类型的文档。

`-d all` 选项只能由删除选项 `-r` 接受。

必须通过使用 `-d` 选项或 `-f` 选项为 `scnasdir` 命令的 `add` 和 `remove` 格式指定目录。

`-f input-file`

可以将目录放置在纯文本文件中，每行一个目录，并与 `-f` 选项一起使用。忽略前导空格和制表符。注释以不带引号的磅 (#) 符号开头，并继续到下一新行。解析器会忽略所有注释。

例 342 将两个 NAS 存储设备目录添加到群集

以下 `scnasdir` 命令将 NAS 设备的两个目录添加到 Oracle Solaris Cluster 配置。

```
# scnasdir -a -h sunuss1 -d /vol/DB1 -d /vol/DB2
```

例 343 从群集中删除 NAS 存储设备的所有目录

以下 `scnasdir` 命令删除为 NAS 设备配置的所有目录。

```
# scnasdir -r -h sunuss1 -d all
```

返回以下退出值：

0 命令已成功执行。

非零值 发生错误。

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	过时

[Intro\(1CL\) \[17\]](#)、[clnasdevice\(1CL\) \[137\]](#)、[clquorum\(1CL\) \[195\]](#)、[cluster\(1CL\) \[461\]](#)、[sconf\(1M\) \[637\]](#)、

名称

scprivipadm — 管理专用 IP 地址范围

```
scprivipadm -c netaddr=netaddr[,netmask=netmask]  
  
scprivipadm -c netaddr=netaddr[,maxnodes=nodes,maxprivatenets=  
    privnets]  
  
scprivipadm -c netaddr=netaddr[,netmask=netmask,maxnodes=nodes,maxprivatenets=  
    privnets]  
  
scprivipadm -p  
  
scprivipadm -R
```

注 - Oracle Solaris Cluster 软件包括面向对象的命令集。虽然 Oracle Solaris Cluster 软件仍支持原始的命令集，但是 Oracle Solaris Cluster 过程文档仅使用面向对象的命令集。有关面向对象的命令集的更多信息，请参见 [Intro\(1CL\) \[17\]](#) 手册页。

scprivipadm 命令修改为 Oracle Solaris Cluster 专用互连分配的当前 IP 地址范围。

在运行此命令的任何格式之前，群集中的所有节点都必须处于非群集模式。从群集中的一个节点运行此命令。

scprivipadm 命令将专用网络地址作为输入。（可选）该命令还可以带有以下项之一或两者：

- 网络掩码
- 曾预期在群集中的最大节点数和最大专用网络数

然后命令执行物理适配器的 IP 地址分配和每节点 IP 地址。

仅可以在全局区域中使用此命令。

支持以下选项：

-c	修改当前为群集分配的 IP 地址范围。在群集的每个节点上运行 -c 选项。 仅可以在全局区域中使用此选项。 -c 选项支持以下子选项： <i>netaddr=netaddr</i> 指定专用网络地址
----	--

<code>netmask=<i>netmask</i></code>	指定网络掩码
<code>maxnodes=<i>nodes</i></code>	指定群集中节点的最大预期数
<code>maxprivatenets=<i>privnets</i></code>	指定群集中专用网络的最大预期数

对于子选项的每种组合，-c 选项执行以下任务：

- 如果单独指定 `netaddr` 子选项，则该命令会将默认网络掩码 255.255.248.0 分配给专用互连。默认 IP 地址范围最多可以容纳 64 个节点和 10 个专用网络。
- 如果还指定 `netmask` 子选项，则您指定的值必须等于或大于默认网络掩码。如果指定的网络掩码小于默认网络掩码，则该命令会失败然后退出，并显示一条错误消息。如果指定的网络掩码大于或等于默认的网络掩码，则该命令会将指定的网络掩码分配给专用互连。生成的 IP 地址范围最多可以容纳 64 个节点和 10 个专用网络。
要分配比默认范围更小的 IP 地址范围，请指定 `maxnodes` 和 `maxprivatenets` 子选项。
- 如果还指定 `maxnodes` 和 `maxprivatenets` 子选项，则该命令会计算支持指定数量的节点和专用网络的最小网络掩码。该命令随后将计算的网路掩码分配给专用互连。`nodes` 的最大值是 64，最小值是 2。`privnets` 的最大值是 128，最小值是 2。
- 如果还指定了 `netmask` 子选项以及 `maxnodes` 和 `maxprivatenets` 子选项，则该命令会计算可支持指定数量的节点和专用网络的最小网络掩码。该命令会将计算结果与指定的网络掩码进行比较。如果指定的网络掩码小于计算的网路掩码，则该命令会失败然后退出，并显示一条错误消息。如果指定的网络掩码大于或等于计算的网路掩码，则该命令会将指定的网络掩码分配给专用互连。`nodes` 的最大值是 64，最小值是 2。`privnets` 的最大值是 128，最小值是 2。

如果 -c 选项失败，则在重新运行 -c 选项之前，必须在每个节点上运行 -R 选项以修复配置。

超级用户以外的用户需要具有 `solaris.cluster.modify` 基于角色的访问控制 (Role-Based Access Control, RBAC) 授权才能使用该子命令。请参见 [Unresolved link to "rbac5" 手册页](#)。

-R 修复群集配置。如果在群集节点上修改 IP 地址范围时命令失败，且该失败导致节点上出现不一致的群集配置，请使用此选项。

仅可以在全局区域中使用此选项。

在群集的每个节点上运行 -R 选项。

-R 选项修复群集配置并删除由失败导致的任何不一致，以修改所有节点上的 IP 地址范围。

如果在没有首先运行 -R 选项的情况下尝试重新运行 -c 选项，则配置更改可能再次失败。

超级用户以外的用户需要具有 `solaris.cluster.modify` 基于角色的访问控制 (Role-Based Access Control, RBAC) 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

-p 显示分配给专用互连的当前专用网络地址。从任何节点运行 -p 选项。

仅可以在全局区域中使用此选项。

-p 选项显示以下信息：

- 专用网络地址
- 网络掩码形式的 IP 地址范围
- IP 地址范围可以支持的最大节点数和最大专用网络数

超级用户以外的用户需要具有 `solaris.cluster.read` 基于角色的访问控制 (Role-Based Access Control, RBAC) 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

要显示处于群集模式的节点中的当前专用网络地址，请改为运行 `scconf -p` 命令或 `cluster show-netprops` 命令。

例 344 计算定制专用 IP 地址范围

以下命令指定专用网络地址 `172.16.0.0` 并计算网络掩码。该命令指定，计算的网路掩码必须支持群集中多达十六个的节点和多达四个的专用网络。

```
# scprivipadm -c
netaddr=172.16.0.0,maxnodes=16,maxprivatenets=4
```

例 345 指定专用网络地址和网络掩码

以下命令指定专用网络地址 `172.16.0.0` 和网络掩码 `255.255.248.0`。

```
# scprivipadm -c
netaddr=172.16.0.0,netmask=255.255.248.0
```

如果出现以下任一情况，则 `scprivipadm` 命令返回非零值：

- 提供的参数无效。
- 该命令无法成功修改群集的所有节点上的 IP 地址范围。

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	过时

[Intro\(1CL\) \[17\]](#)、[cluster\(1CL\) \[461\]](#)、[scconf\(1M\) \[637\]](#)、[scinstall\(1M\) \[697\]](#)、[Unresolved link to "netmasks4"](#)、[Unresolved link to "networks4"](#)、[Unresolved link to "rbac5"](#)

[Unresolved link to "Oracle Solaris Cluster 软件安装指南"](#)、[Unresolved link to "Oracle Solaris Cluster 系统管理指南"](#)、《配置和管理 Oracle Solaris 网络》

超级用户可以运行此命令的所有格式。除超级用户之外的用户需要具有 RBAC 授权。请参见下表。

选项	RBAC 授权
-c	solaris.cluster.modify
-R	solaris.cluster.modify
-p	solaris.cluster.read

名称

scprivipd — Oracle Solaris Cluster 专用 IP 地址服务守护进程

/usr/cluster/lib/sc/scprivipd

scprivipd 守护进程在系统引导时启动。它用于配置或取消配置在区域引导或关闭时分配的专用 IP 地址或者作为 scconf 操作的结果而分配的专用 IP 地址。

scprivipd 守护进程没有与外部的直接 stdin、stdout 或 stderr 连接。所有诊断消息均是通过 syslog 函数进行记录的。

scprivipd 守护进程必须在超级用户模式下运行。

scprivipd 守护进程是服务管理工具 (Service Management Facility, SMF) 服务，它通过 SMF 来启动。此外，如果 scprivipd 守护进程被信号中止，则 SMF 将自动重新启动它。

SIGTERM 信号可用于正常中止 scprivipd。不应使用其他信号来中止守护进程。

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Private (专用)

[clnode\(1CL\) \[149\]](#)、[scconf\(1M\) \[637\]](#)、[Unresolved link to "syslog3C"](#)、[Unresolved link to "attributes5"](#)

名称

scrgadm — 管理资源类型、资源组和资源的注册和取消注册

```
scrgadm -p[v[v]] [-t resource_type_name] [-g resource_group_name]
        [-j resource_name]

scrgadm -a -t resource_type_name [-h RT_installed_node_list] [-f
        registration_file_path]

scrgadm -a -g RG_name [-h nodelist] [-y property=value...]

scrgadm -a -j resource_name -t resource_type_name -g RG_name [-y
        property=value...] [-x "extension_property[{node_specifier}]=
        value..."]

scrgadm -a -L -g RG_name -l hostnamelist [-j resource_name] [-n
        netiflist] [-y property=value...]

scrgadm -a -S -g RG_name -l hostnamelist [-j resource_name] [-n
        netiflist] [-X auxnodelist] [-y property=value...]

scrgadm -c -t resource_type_name [-h RT_installed_node_list]
        [-y RT_system={TRUE|FALSE}]

scrgadm -c -g RG_name [-h nodelist] -y property=value...

scrgadm -c -j resource_name [-y property...] [-x "
        extension_property[{node_specifier}]=value..."]

scrgadm -r -t resource_type_name

scrgadm -r -g RG_name

scrgadm -r -j resource_name
```

注 - Oracle Solaris Cluster 软件包括面向对象的命令集。虽然 Oracle Solaris Cluster 软件仍支持原始的命令集，但是 Oracle Solaris Cluster 过程文档仅使用面向对象的命令集。有关面向对象的命令集的更多信息，请参见 [Intro\(1CL\) \[17\]](#) 手册页。

资源类型指定该类型的所有资源的通用属性和回调方法。在创建特定类型的资源之前，必须先使用该命令的以下格式注册资源类型：

```
# scrgadm -a -
t resource_type_name
```

资源组包含一组资源，其中的所有资源在给定的一个节点或一组节点上一起联机或脱机。先创建一个空资源组，再向其中放置任何资源。要创建资源组，请使用以下命令：

```
# scrgadm -a -
```

```
g RG_name
```

有以下两种类型的资源组：故障转移资源组和可伸缩资源组。

故障转移资源组每次只能在一个节点上处于联机状态。故障转移资源组可以包含任何类型的资源，尽管在故障转移资源组中配置的可伸缩资源每次只能在一个节点上运行。

要创建名为 MyDatabaseRG 的故障转移资源组，请使用以下命令：

```
# scrgadm -a -  
g MyDatabaseRG
```

一个可伸缩资源组可以同时几个节点上处于联机状态。可伸缩资源组只能包含支持伸缩性的资源，而不能包含由其资源类型定义约束为仅故障转移行为的资源。

要创建名为 MyWebServerRG 的可伸缩资源组，请使用以下命令：

```
# scrgadm -a -  
g MyWebServerRG \  
-y Maximum primaries=integer \  
-y Desired primaries=integer
```

新创建的资源组处于 UNMANAGED 状态。在组中创建资源后，请使用 `scswitch` 命令将资源组置于 MANAGED 状态。

要在资源组中创建给定类型的资源，请使用以下命令：

```
# scrgadm -a -  
j resource_name -  
t resource_type_name -  
g RG_name
```

创建资源可导致底层 RGM 机制执行若干操作。底层 RGM 机制在资源上调用 `VALIDATE` 方法以验证资源的属性设置是否有效。如果 `VALIDATE` 方法成功完成，且资源组已置于 MANAGED 状态，则 RGM 将通过在资源上调用 `INIT` 方法初始化资源。如果已启用资源且其资源组处于联机状态，随后 RGM 会使资源联机。

要删除受管理的资源组，请先从该资源组中删除所有的资源。要删除资源，请先使用 `scswitch` 命令禁用它。通过在资源上调用 `FINI` 方法删除资源可导致 RGM 在该资源之后清理。

仅可以在全局区域中使用此选项。

操作选项

操作选项指定命令执行的操作。在命令行上仅允许一个操作选项。

支持以下操作选项：

-a

添加新配置。与以下选项一起使用：

-g 创建资源组。
仅可以在全局区域中使用此选项。
要将该命令选项与 -a、-c 或 -r 一起使用，需要具有 solaris.cluster.resource.modify RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

-j 创建资源。
仅可以在全局区域中使用此选项。
要将该命令选项与 -a、-c 或 -r 一起使用，需要具有 solaris.cluster.resource.modify RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

-t 添加资源类型。
仅可以在全局区域中使用此选项。
要将该命令选项与 -a、-c 或 -r 一起使用，需要具有 solaris.cluster.resource.modify RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

-c

修改现有的配置。仅设置指定属性的值。其他属性保留其当前值。与以下选项一起使用：

-g 修改资源组。
仅可以在全局区域中使用此选项。
要将该命令选项与 -a、-c 或 -r 一起使用，需要具有 solaris.cluster.resource.modify RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

-j 修改资源。
仅可以在全局区域中使用此选项。
要将该命令选项与 -a、-c 或 -r 一起使用，需要具有 solaris.cluster.resource.modify RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

-t 修改资源类型。
仅可以在全局区域中使用此选项。
要将该命令选项与 -a、-c 或 -r 一起使用，需要具有 solaris.cluster.resource.modify RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

-r

删除配置。与以下选项一起使用：

-g 删除资源组。
仅可以在全局区域中使用此选项。
要将该命令选项与 -a、-c 或 -r 一起使用，需要具有 `solaris.cluster.resource.modify` RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

-j 删除资源。
仅可以在全局区域中使用此选项。
要将该命令选项与 -a、-c 或 -r 一起使用，需要具有 `solaris.cluster.resource.modify` RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

-t 删除资源类型。
仅可以在全局区域中使用此选项。
要将该命令选项与 -a、-c 或 -r 一起使用，需要具有 `solaris.cluster.resource.modify` RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

-p

显示现有的配置信息。

仅可以在全局区域中使用此选项。与以下选项一起使用：

-g *resource_group_name*
显示特定的资源组配置信息。
要将该命令选项与 -p 一起使用，需要具有 `solaris.cluster.resource.read` RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

-j *resource_name*
显示特定的资源配置信息。
要将该命令选项与 -p 一起使用，需要具有 `solaris.cluster.resource.read` RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

-t *resource_type_name*
显示特定的资源类型配置信息。
要将该命令选项与 -p 一起使用，需要具有 `solaris.cluster.resource.read` RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

-v[v]
显示更详细的输出。

要将该命令选项与 `-p` 一起使用，需要具有 `solaris.cluster.resource.read` RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

如果未指定 `-g`、`-j` 或 `-t` 中的任一选项，则默认情况下将提供有关群集中当前配置的所有资源类型、资源组和资源的信息。

支持多个 `-g`、`-j` 和 `-t` 选项，可以将其与 `-v` 选项的任意组合结合使用。

在单个命令行上最多可以使用两个 `-v` 选项。

目标选项

目标选项标识目标对象。支持以下目标选项：

注 - 资源组、资源和资源类型的属性名称不区分大小写。指定属性名称时，可以使用大小写字母的任意组合。

`-g RG_name`

资源组。

`-j resource_name`

资源。与 `-a` 选项一起使用时，必须在命令中指定 `-t` 和 `-g` 目标选项，以指示要实例化的资源的类型以及包含该资源的资源组的名称。

`-t resource_type_name`

资源类型。

资源类型特定的选项

支持以下选项：

`-f registration_file_path`

对 `-a` 有效。指定资源类型注册 (resource type registration, RTR) 文件的路径名称。按照约定，RTR 文件驻留在 `/opt/cluster/lib/rgm/rtreg` 目录下。如果 RTR 文件不在此目录中，则必须指定该选项。

`-h RT_installed_node_list`

对 `-a` 和 `-c` 有效。指定在其上安装此资源类型的节点名称的逗号分隔列表。此类型的资源只能在其节点列表是该列表的子集的资源组中实例化。

`-h` 选项对 `-a` 选项是可选的。如果未指定 `-h`，则表示已在所有节点上安装该资源类型。如果这样做，此类型的资源就可以在任何资源组中进行实例化。

与 `-c` 选项一起使用时，必须与新的已安装节点列表或转义通配符 (`*`) 一起指定 `-h`。通配符指示已在所有节点上安装该资源类型。

注 - 不允许节点名称中包含逗号。

`-t resource_type_name`

对 `-a`、`-c` 和 `-r` 有效。资源类型由用来为该资源类型指定标准和扩展属性值的资源类型注册文件定义。如果将有效的资源类型注册文件放置在通常安装注册文件的已知目录 (`/opt/cluster/lib/rgm/rtreg`) 中，则支持简写表示法：

```
# scrgadm -a -t SUNW.rt:2.0
```

因此，不需要使用以下表示法：

```
# scrgadm -a -t rtn -f full_path_to_SUNW.rt:2.0
```

要查看当前注册的资源类型的名称，请使用以下命令：

```
# scrgadm -p
```

从 Sun Cluster 3.1 开始，资源类型名称的语法如下：

```
vendor_id.resource_type  
:version
```

资源类型名称的三个组成部分是在 RTR 文件中指定为 *Vendor_id*、*Resource_type* 和 *RT_version* 的属性。`scrgadm` 命令用于插入句点和冒号分界符。仅当需要区分由不同供应商提供的两个同名注册文件时，可选的 *Vendor_id* 前缀才是必需的。*RT_version* 用于从一个数据服务版本升级到另一个数据服务版本。

要确保 *Vendor_id* 唯一，请使用要创建资源类型的公司的股票代码。与 `-t` 选项一起使用的 *resource_type_name* 可以是完整的资源类型名称，也可以是省略了 *Vendor_id* 的缩写。例如，`-tSUNW.iws` 和 `-tiws` 都是有效的。如果群集中有两个仅 *Vendor_id* 前缀不同的资源类型名称，则使用缩写名称将发生故障。

如果 *RT_version* 字符串包含空白、制表符、斜杠 (/)、反斜杠 (\)、星号 (*)、问号 (?)、左方括号 (l) 或右方括号 (l) 字符，则 `scrgadm` 命令无法注册资源类型。

将 *resource_type_name* 与 `-t` 选项一起指定时，如果仅注册一个版本，则可以省略版本部分。

在 Sun Cluster 3.1 发行版之前创建的资源类型名称继续符合以下语法：

```
vendor_id.resource_type
```

`-y RT_system={TRUE|FALSE}`

将资源类型的 `RT_system` 属性设置为 `TRUE` 或 `FALSE`。`RT_system` 属性的默认值为 `FALSE`。有关 `RT_system` 属性的描述，请参见 [rt_properties\(5\) \[1143\]](#)。

资源组特定的选项

支持以下选项：

`-h nodelist`

对 `-a` 和 `-c` 有效。此选项是 `-y Nodelist=nodelist` 的快捷方式。

`-y property= value`

对 `-a` 和 `-c` 有效。允许使用多个 `-y property= value` 实例。`value` 的格式由每个 `property` 指定。在以下示例中，`property1` 采用单个字符串作为 `value`，而 `property2` 采用逗号分隔的字符串数组：

```
-y property1=
value1 -y
property2=value2a,value2b
```

要将字符串属性设置为空值，请使用此选项而不指定值，如下所示：

```
-y property=
```

识别 `-y` 属性名称时不区分大小写。

有关资源组属性的描述，请参见 [rg_properties\(5\) \[1129\]](#)。

资源特定的选项

支持以下选项：

```
-x extension_property= value
-x "extension_property {node_specifier}=value "
```

对 `-a` 和 `-c` 有效。允许使用多个 `-xextension_property= value` 或 `-x "extension_property{ node_specifier}=value"` 实例。

`node_specifier` 是可选限定符，它指示仅在指定的一个或多个节点上设置或更改 `extension_property` 的值。在群集中的其他节点上，不会设置或更改指定属性的值。如果不包括 `node_specifier`，则在群集中的所有节点上设置或更改指定属性的值。`node_specifier` 的语法示例如下：

```
-x "myprop{phys-schost-1}=100"
```

使用大括号 (`{}`) 指示要在其上设置属性的一个或多个特定节点。

还可以使用以下的 `node_specifier` 语法同时在两个不同的节点上指定不同的值：

```
-x "myprop{phys-schost-1}=100" -x "myprop{phys-schost-2}=10"
```

或者，可以使用以下语法同时在两个不同的节点上设置或更改一个值：

```
-x "myprop{phys-schost-1,phys-schost-2}=100"
```

`value` 的格式由每个 `extension_property` 指定。在以下示例中，`extension_property1` 采用单个字符串作为 `value`，而 `extension_property2` 采用逗号分隔的字符串：

```
-x "extension_property1{
node_specifier}=value1" \
-x "extension_property2{
node_specifier}=value2a,
value2b"
```

有关可用于特定数据服务的扩展属性的信息，请参阅该数据服务的手册页。

`-y property= value`

对 `-a` 和 `-c` 有效。允许使用多个 `-y property= value` 实例。`value` 的格式由每个 `property` 指定。在以下示例中，`property1` 采用单个字符串作为 `value`，而 `property2` 采用逗号分隔的字符串数组：

```
-y property1=  
value1 -y  
property2=value2a,value2b
```

要将属性设置为空值，请使用此选项而不指定值，如下所示：

```
-y property=
```

识别 `-yproperty` 名称时不区分大小写。

有关资源属性的描述，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

LogicalHostname 特定的选项

这些选项适用于逻辑主机名资源。没有用于删除 LogicalHostname 资源的特殊命令：

```
# scrgadm -r -  
j resource_name
```

`resource_name` 是在创建 LogicalHostname 资源时通过可选的 `-j` 选项提供的名称。如果在创建 LogicalHostname 资源时省略了 `-j` 选项和 `resource_name`，则该名称由 `scrgadm` 生成。

支持以下选项：

`-j resource_name`

使用 IP 地址而不是主机名作为 `-l hostnamelist` 选项的第一个参数时，`-j` 选项是必需的。

在创建资源时将 `-j` 与 `-a` 一起使用可显式命名 LogicalHostname 资源，与 `-r` 一起使用可从资源组中删除资源。如果未使用 `-j` 选项显式命名 `resource`，则 `scrgadm` 命令将创建资源并将 `hostnamelist` 中的第一个主机名分配给该资源。

`-L`

指示在命令行上使用的选项适用于逻辑主机名。如果发出该命令时任一群集节点不是活动群集的成员，则还必须使用 `-n netiflist` 选项。

`-l hostnamelist`

指定要共享的 IPv4 或 IPv6 地址。即使可以指定 IP 地址，也要使用主机名。`hostnamelist` 是要通过此 LogicalHostname 资源使其变为可用的主机名的列表（以逗号分隔）。

`-n netiflist`

指定网络接口的列表。如果在任一群集节点不是活动群集成员时发出命令，则 `-L` 选项需要 `-n` 选项。

`netiflist` 采用以下格式：

```
netif@node[,...]
```

`netif` 可以作为网络适配器名称（如 `le0`）或作为 IP 网络多路径组名称（如 `sc_ipmp`）提供。`node` 可以是节点名称或节点标识符。资源组的 `nodelist` 中的所有节点都必须列在 `netiflist` 中。如果省略了 `-n netiflist`，则将对 `nodelist` 中的每个节点进行尝试以发现通过 `hostnamelist` 标识的子网上的网络适配器。对于尚未在 IP 网络多路径组中的已发现网络适配器，会创建单适配器 IP 网络多路径组。同样，对于指定的适配器，会创建单适配器 IP 网络多路径组（如果该组尚不存在）。

有关更多信息，请参阅 附注 部分。

`-y property= value`

有关详细信息，请参阅“资源特定的选项”部分。

SharedAddress 特定的选项

LogicalHostname 特定的所有选项也适用于 SharedAddress 资源，但更改和添加了以下内容：

`-s`

指示在命令行上使用的选项适用于共享地址。

`-x auxnodelist`

指定节点名称或标识符的逗号分隔列表。此列表上的条目必须是群集的成员。这些节点是可能托管指定的共享地址但在发生故障转移时从未充当主节点的节点。

此列表与 `nodelist` 互斥。请参见“资源组特定的选项”下 `nodelist` 的描述。

返回以下退出值：

0

命令已成功完成。

可能会向标准错误写入警告消息，即使此命令已成功完成也是如此。

非零值

发生错误。

当它以非零状态退出时，将向标准错误写入错误消息。

不允许对 `RT_System` 属性为 `TRUE` 的资源类型执行某些操作。同样，不允许对 `RG_System` 属性为 `TRUE` 的资源组（及其资源）执行某些操作。请参见 [rt_properties\(5\) \[1143\]](#) 和 [rg_properties\(5\) \[1129\]](#)。

有关下列属性的说明，请参见 [Unresolved link to " attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	过时

[Intro\(1CL\) \[17\]](#), [clreslogicalhostname\(1CL\) \[207\]](#), [clresource\(1CL\) \[225\]](#),
[clresourcegroup\(1CL\) \[253\]](#), [clresourcetype\(1CL\) \[277\]](#),
[clressharedaddress\(1CL\) \[289\]](#), [Unresolved link to " ifconfig1M"](#), [scstat\(1M\) \[745\]](#),
[scswitch\(1M\) \[751\]](#), [r_properties\(5\) \[1103\]](#), [Unresolved link to " rbac5"](#),
[rg_properties\(5\) \[1129\]](#), [rt_properties\(5\) \[1143\]](#)

在 LogicalHostname 和 SharedAddress 添加操作执行过程中，无法发现尚未配置为可用的网络适配器或将其放置到 IP 网络多路径组中。请参见 [Unresolved link to " ifconfig1M"](#)。

如果 scrgadm 以非零状态退出并显示错误消息 cluster is reconfiguring，则说明尽管出现错误状态，但请求的操作可能已成功完成。如果对结果有疑问，则可以在重新配置完成后使用相同的参数再次执行 scrgadm。

名称

scsetup — 交互式群集配置工具

scsetup [-f *logfile*]

注 - Oracle Solaris Cluster 软件包括面向对象的命令集。虽然 Oracle Solaris Cluster 软件仍支持原始的命令集，但是 Oracle Solaris Cluster 过程文档仅使用面向对象的命令集。有关面向对象的命令集的更多信息，请参见 [Intro\(1CL\) \[17\]](#) 手册页。

scsetup 命令可提供以下配置功能，具体取决于发出该命令时群集所处的状态。

- 在安装后运行 scsetup 命令时，该命令会执行初始设置任务，例如配置法定设备和重置 `installmode` 属性。如果您在创建群集时没有使用自动法定配置，请在安装该群集后立即运行 scsetup 命令。在运行 scsetup 命令并重置 `installmode` 属性之前，确保所有节点都已加入群集。

如果在创建群集时使用自动法定配置，则无需在群集安装后运行 scsetup 命令。自动法定配置功能还会重置群集的 `installmode` 属性。

- 当您在正常群集操作期间运行 scsetup 命令时，该命令会提供一个菜单驱动式实用程序。可以使用此实用程序执行大多数持续的群集管理任务。
- 从处于非群集模式下的节点发出该命令时，scsetup 命令会提供一个菜单驱动式实用程序，用于更改和显示专用 IP 地址范围。在启动此格式的 scsetup 实用程序之前，必须将所有节点重新引导到非群集模式。

可以从群集中的任何节点发出 scsetup 命令。

仅可以在全局区域中使用此命令。

支持以下选项：

`-f logfile` 指定可以记录命令的日志文件的名称。如果指定此选项，系统会运行并记录或者仅记录（取决于用户响应）scsetup 实用程序生成的大多数命令集。

有关下列属性的描述，请参见 [Unresolved link to "attributes5"](#)。

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	过时

[Intro\(1CL\)](#) [17], [cltelemetryattribute\(1CL\)](#) [447], [cldevicegroup\(1CL\)](#) [65],
[clnode\(1CL\)](#) [149], [clquorum\(1CL\)](#) [195], [clreslogicalhostname\(1CL\)](#) [207],
[clresourcegroup\(1CL\)](#) [253], [clresourcetype\(1CL\)](#) [277],
[clressharedaddress\(1CL\)](#) [289], [cluster\(1CL\)](#) [461],

名称

scshutdown — 关闭群集

```
scshutdown [-y] [-g grace-period] [message]
```

注 - Oracle Solaris Cluster 软件包括面向对象的命令集。虽然 Oracle Solaris Cluster 软件仍支持原始的命令集，但是 Oracle Solaris Cluster 过程文档仅使用面向对象的命令集。有关面向对象的命令集的更多信息，请参见 [Intro\(1CL\) \[17\]](#) 手册页。

scshutdown 实用程序可按顺序关闭整个群集。

在开始关闭之前，scshutdown 先发送一条警告消息，然后发送最终确认消息。

只能从一个节点运行 scshutdown 命令。

scshutdown 命令在关闭群集时执行以下操作：

- 将群集内所有工作中资源组更改为脱机状态。如果发生任何转换失败，scshutdown 不会完成，并显示一条错误消息。
- 卸载所有群集文件系统。如果发生任何卸载失败，scshutdown 不会完成，并显示一条错误消息。
- 关闭所有活动的设备服务。如果设备的任何转换失败，scshutdown 将无法完成任务并显示一条错误消息。
- 在所有节点上运行 `/usr/sbin/init 0`。有关更多信息，请参见 [Unresolved link to "init1M"](#)。

仅可以在全局区域中使用此命令。

需要具有 `solaris.cluster.system.admin` RBAC 授权才能使用该命令。请参见 [Unresolved link to "rbac5"](#)。

支持以下选项：

`-ggrace-period` 将秒数从 60 秒默认值更改为 *grace-period* 指定的时间。

`-y` 预先回答确认问题，使得此命令无需用户介入即可运行。

支持以下操作数：

message 在发出标准警告消息 "The system will be shut down in ..." (系统将在...内关闭) 之后发出的字符串。如果 *message*

包含多个单词，使用单引号 (') 或双引号 (") 进行分隔。警告消息和用户提供的 *message* 在距 `scshutdown` 开始还有 7200、3600、1800、1200、600、300、120、60 和 30 秒时输出。

例 346 关闭群集

```
phys-palindrome-1# scshutdown
```

返回以下退出值：

0 命令已成功完成。

非零值 发生错误。错误消息显示在标准输出上。

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	过时

[Intro\(1CL\) \[17\]](#)、[cluster\(1CL\) \[461\]](#)、[Unresolved link to "shutdown1M"](#)、[Unresolved link to "init1M"](#)、[Unresolved link to "attributes5"](#)

名称

scstat — 监视 Oracle Solaris Cluster 配置的状态

```
scstat [-DWginpqv [v]] [-h node]
```

注 - Oracle Solaris Cluster 软件包括面向对象的命令集。虽然 Oracle Solaris Cluster 软件仍支持原始的命令集，但是 Oracle Solaris Cluster 过程文档仅使用面向对象的命令集。有关面向对象的命令集的更多信息，请参见 [Intro\(1CL\) \[17\]](#) 手册页。

scstat 命令可显示 Oracle Solaris Cluster 组件的当前状态。在 Oracle Solaris Cluster 配置中的任一计算机上，只需运行 scstat 命令的一个实例。

不带选项运行 scstat 命令时，将显示群集的所有组件的状态。将显示以下信息：

- 群集成员列表
- 每个群集成员的状态
- 资源组和资源的状态
- 群集互联的每个路径的状态
- 每个磁盘设备组的状态
- 每个法定设备的状态
- 每个 IP 网络多路径 (IP network multipathing, IPMP) 组和公共网络适配器的状态

要使用不带选项的此命令，您需要具有

solaris.cluster.device.read、solaris.cluster.transport.read、solaris.cluster.resource.read、solaris.cluster.system.read RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

资源和资源组

资源状况、资源组状况和资源状态均是基于每个节点进行维护的。例如，一个给定资源在每个群集节点上可以具有不同的状况和状态。

资源状态是由资源组管理器 (Resource Group Manager, RGM) 仅根据对资源调用的方法在每个节点上设置的。例如，在给定节点上对某个资源成功运行 STOP 方法后，该资源在此节点上的状况将为 OFFLINE。如果 STOP 方法返回非零值或超时，则该资源的状态为 Stop_failed。

可能的资源状态包括：Online、Offline、Start_failed、Stop_failed、Monitor_failed、Online_not_monitored、Starting 和 Stopping。

可能的资源组状态包括：Unmanaged、Online、Offline、Pending_online、Pending_offline、Error_stop_failed、Online_faulted 和 Pending_online_blocked。

除资源状况外，RGM 还可以维护资源状态，资源状态可由资源本身使用 API 设置。Status Message 字段实际上包括两个组成部分：状态关键字和状态消息。状态消息由资源有选择性地设置，它是在状态关键字后面输出的任意文本字符串。

资源状态可能值的描述如下：

DEGRADED	资源已联机，但其性能或可用性可能在某方面出现了下降。
FAULTED	资源遇到错误，无法正常运行。
OFFLINE	资源已脱机。
ONLINE	资源已联机，正在提供服务。
UNKNOWN	当前状态是未知或正在转换。

设备组

设备组状态反映该组中设备的可用性。

设备组状态的可能值及其描述如下：

DEGRADED	设备组处于联机状态，但其潜在主（辅助）节点没有全部联机。对于双节点连接，此状态实质上表明不存在备用主节点，也就是说，如果主节点发生故障，将导致无法访问组中的设备。
OFFLINE	设备组处于脱机状态。不存在主节点。在可以使用设备组中的任何设备之前，必须先使该组联机。
ONLINE	设备组处于联机状态。存在主节点，而且该组中的设备已准备好可以进行 I/O。
WAIT	设备组处于两个状态之间。例如，当设备组正在从脱机进入联机状态时可能发生此状态。

IP 网络多路径组

IP 网络多路径 (IP network multipathing, IPMP) 组状态反映备份组以及该组中适配器的可用性。

IPMP 组状态的可能值及其描述如下：

OFFLINE	备份组故障。该组中的所有适配器都处于脱机状态。
ONLINE	备份组正常工作。该组中至少有一个适配器联机。

UNKNOWN 除上述所列状态以外的任何其他状态。如果适配器断开或由 Solaris 命令（例如 `if_mpadm` 或 `ifconfig`）标记为关闭时，可能导致此状态。

IPMP 适配器状态的可能值及其描述如下：

OFFLINE 适配器发生故障或备份组处于脱机状态。

ONLINE 适配器正常工作。

STANDBY 适配器处于待机状态。

UNKNOWN 除上述所列状态以外的任何其他状态。如果适配器断开或由 Solaris 命令（例如 `if_mpadm` 或 `ifconfig`）标记为关闭时，可能导致此状态。

可以指定命令选项来请求获取特定组件的状态。

如果指定了多个选项，`scstat` 命令将按指定顺序输出状态。

支持以下选项：

`-D` 显示所有磁盘设备组的状态。

您需要具有 `solaris.cluster.device.read` RBAC 授权才能使用该命令选项。请参见 [Unresolved link to "rbac5"](#)。

`-g` 显示所有资源组的状态。

您需要具有 `solaris.cluster.resource.read` RBAC 授权才能使用该命令选项。请参见 [Unresolved link to "rbac5"](#)。

`-h node` 显示指定节点 (*node*) 的状态和以该节点为主节点的磁盘设备组的状态。还显示某些法定设备的状态，即此节点为其是潜在主节点的资源组以及 *node* 所连接到的传输路径保留存储空间的法定设备。

要使用此命令选项，您需要具有 `solaris.cluster.device.read`、`solaris.cluster.transport.read`、`solaris.cluster.r` 和 `solaris.cluster.system.read` RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

`-i` 显示所有 IPMP 组和公共网络适配器的状态。

仅可以在全局区域中使用此选项。

`-n` 显示所有节点的状态。

您需要具有 `solaris.cluster.node.read` RBAC 授权才能使用该命令选项。请参见 [Unresolved link to "rbac5"](#)。

- p 显示群集中所有组件的状态。与 -v 一起使用可显示更详细的输出。
要将 -p 与 -v 一起使用，您需要具有 solaris.cluster.device.read、solaris.cluster.transport.read、solaris.cluster和 solaris.cluster.system.read RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。
- q 显示所有法定设备和法定节点的状态。
您需要具有 solaris.cluster.quorum.read RBAC 授权才能使用该命令选项。请参见 [Unresolved link to "rbac5"](#)。
- v[v] 显示详细输出。
- w 显示群集传输路径的状态。
您需要具有 solaris.cluster.transport.read RBAC 授权才能使用该命令选项。请参见 [Unresolved link to "rbac5"](#)。

例 347 使用 scstat 命令

以下命令可显示所有资源组的状态以及与指定主机相关的所有组件的状态：

```
% scstat -g -h host
```

输出的显示顺序与选项的指定顺序相同。

这些结果与键入以下两个命令的结果相同：

```
% scstat -g
```

和

```
% scstat -h host
```

返回以下退出值：

0 命令已成功完成。

非零值 发生错误。

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core

属性类型	属性值
接口稳定性	过时

[Intro\(1CL\) \[17\]](#)、[cluster\(1CL\) \[461\]](#)、[Unresolved link to "if_mpadm1M"](#)、[Unresolved link to "ifconfig1M"](#)、[scha_resource_setstatus\(1HA\) \[593\]](#)、[scha_resource_setstatus\(3HA\) \[1003\]](#)、[Unresolved link to "attributes5"](#)

联机法定设备表示在上次建立法定设备时该设备可用于形成法定设备。在法定算法上下文中，该设备处于联机状态，因为它主动参与法定设备的形成。但是，联机法定设备可能不一定持续处于足够正常的状态，无法在重新建立法定设备时参与该设备的形成。最新版本的 Oracle Solaris Cluster 不包括磁盘监视设备或对法定设备的定期探测。

名称

scswitch — 在 Oracle Solaris Cluster 配置中执行资源组和设备组的所有权与状态更改

```
scswitch -c -h node[,...] -j resource[,...] -f flag-name
scswitch {-e | -n} [-M] -j resource[,...] [-h node[,...]]
scswitch -F {-g resource-grp[,...] | -D device-group[,...]}
scswitch -m -D device-group[,...]
scswitch -Q [-g resource-grp[,...]] [-k]
scswitch -R -h node[,...] -g resource-grp[,...]
scswitch -r [-g resource-grp[,...]]
scswitch -S -h node[,...] [-K continue_evac]
scswitch -s [-g resource-grp[,...]] [-k]
scswitch {-u | -o} -g resource-grp[,...]
scswitch -Z [-g resource-grp[,...]]
scswitch -z -D device-group[,...] -h node [,...]
scswitch -z [-g resource-grp[,...]] [-h node [,...]]
```

注 - Oracle Solaris Cluster 软件包括面向对象的命令集。虽然 Oracle Solaris Cluster 软件仍支持原始的命令集，但是 Oracle Solaris Cluster 过程文档仅使用面向对象的命令集。有关面向对象的命令集的更多信息，请参见 [Intro\(1CL\) \[17\]](#) 手册页。

scswitch 命令可将资源组或设备组（也称为磁盘设备组）移到新主节点。该命令还提供了各种选项，分别用于通过移动所有权从节点清除所有资源组和设备组、使资源组或设备组脱机和联机、启用或禁用资源、切换资源组的 Unmanaged 状态或者清除资源的错误标志。

可以从 Oracle Solaris Cluster 配置中的任何节点运行 scswitch 命令。如果某个设备组脱机，可使用 scswitch 使该设备组在节点列表中的任何主机上联机。但是，一旦该设备组联机，即不允许切换到备用节点。一次仅允许调用一个 scswitch 命令。

请勿试图中止已经运行的 scswitch 操作。

有关有效使用该命令的更多信息，请参见各个选项的描述。为便于管理，请在全局区域中使用此命令。

基本选项

支持以下基本选项。可以与部分基本选项一起使用的选项在“附加选项”中有所介绍。

- c 清除指定节点中指定资源集的 *-fflag-name* 错误标志。对于 Oracle Solaris Cluster 软件的最新发行版，-c 选项仅对 Stop_failed 资源状态实现。清除 Stop_failed 资源状态会将资源在指定节点中置于脱机状态。
- 仅在全局区域中使用此选项。
- 如果对资源执行 Stop 方法失败，而且该资源的 Failover_mode 属性设置为 Hard，则资源组管理器 (Resource Group Manager, RGM) 将停止或重新引导节点，以强制该资源（以及该节点管理的所有其他资源）脱机。
- 如果对资源执行 Stop 方法失败，而且 Failover_mode 属性设置为 Hard 以外的其他值，则各个资源将进入 Stop_failed 资源状态，而资源组将置于 Error_stop_failed 状态。无法使任何节点上处于 Error_stop_failed 状态的资源组在任何节点上联机，也不能编辑此类资源组（不能添加或删除资源或者更改资源组属性或资源属性）。必须通过执行[Unresolved link to "Oracle Solaris Cluster 数据服务规划和管理指南"](#)中记录的过程来清除 Stop_failed 资源状态。



注意 - 请确保在清除 Stop_failed 资源状态之前，在指定节点中停止了资源及其监视器。清除 Stop_failed 资源状态而不完全中止资源及其监视器可能导致该资源的多个实例在群集中同时执行。如果使用的是共享存储，这种情况可能会导致数据损坏。如有必要，作为最后的解决方法，对关联的进程执行 [Unresolved link to "kill1"](#) 命令。

- e 启用指定的 *resource*。
- 仅在全局区域中使用此选项。
- 启用了资源后，该资源将进入联机或脱机状态，具体取决于其资源组是联机还是脱机。
- 可以同时指定 -h 选项与 -e 选项，仅在指定的节点子集中启用资源。如果省略 -h 选项，则会在所有节点中启用指定资源。
- F 使指定资源组 (-g) 或设备组 (-D) 在所有节点上脱机。
- 指定 -F 选项与 -D 选项时，可以只从全局区域运行 -F 选项。
- 使用 -F 选项使设备组脱机时，主节点会停止或释放关联的 Solaris Volume Manager 磁盘集。将设备组置于脱机状态之前，必须停止对其中设备的所有访问并且卸载所有依赖该组的文件系统。要启动脱机的设备组，可以发出显式 *scswitch* 调用、访问该组中的设备或者挂载依赖该组的文件系统。

-
- m** 使指定的设备组从群集脱机以进行维护。生成的状态在重新引导后保持不变。
- 仅在全局区域中使用此选项。
- 在将设备组置于维护模式之前，必须停止对其中设备的所有访问并且卸载所有依赖该组的文件系统。如果某个设备组当前正被访问，则该操作将失败，指定的设备组不会从群集脱机。
- 可使用 **-z** 选项使设备组恢复联机。只有对 `scswitch` 命令的显式调用才能使设备组脱离维护模式。
- n** 禁用指定的资源。
- 仅在全局区域中使用此选项。
- 在当前主节点上联机的已禁用资源将立即从这些主节点脱机。无论其资源组状态如何，禁用的资源都将保持脱机状态。
- 可以同时指定 **-h** 选项与 **-e** 选项，仅在指定的节点子集中禁用资源。如果省略 **-h** 选项，则会在所有节点中禁用指定资源。
- o** 使不受管理的指定资源组脱离不受管理状态。
- 一旦资源组处于受管理状态，RGM 即会尝试将该资源组联机。
- Q** 将指定的资源组置于停顿状态。
- 如果省略 **-g** 选项，**-Q** 选项将应用于所有资源组。
- 此选项可阻止指定资源组在 `Start` 或 `Stop` 方法失败时不断在节点间切换。在资源组进入停顿状态而不再在任何节点上停止或启动之前，这种 `scswitch` 命令格式不会退出。
- 如果在 `scswitch -Q` 命令执行期间，对组中的任何资源执行 `Monitor_stop`、`Stop`、`Postnet_stop`、`Start` 或 `Prenet_start` 方法失败，则该资源的行为类似于其 `Failover_mode` 属性设置为 `None` 时，无论其实际设置如何。一旦其中某个方法失败，该资源即进入错误状态 (`Start_failed` 或 `Stop_failed` 资源状态)，而不会启动节点的故障转移或重新引导。
- 当 `scswitch -Q` 命令退出时，指定的资源组可能处于联机或脱机状态或者处于 `ONLINE_FAULTED` 或 `ERROR_STOPPED_FAILED` 状态。可通过执行 `clresourcegroup status` 命令确定其当前状态。
- 如果某个节点在 `scswitch -Q` 命令执行期间消亡，则可能会中断其执行，使资源组处于非停顿状态。如果执行中断，`scswitch -Q` 将返回一个非零退出代码，并向标准错误中写入一条错误消息。在此情况下，您可以重新发出 `scswitch -Q` 命令。
- 可以同时指定 **-k** 选项与 **-Q** 选项来加速资源组的停顿。如果指定了 **-k** 选项，该命令会立即中止对受影响资源组中的资源运行的所有方法。如果未指定 **-k** 选项，系统会允许方法继续运行，直到这些方法退出或超过所配置的超时时间。

-
- R** 使指定资源组在指定的主节点中脱机，然后使其恢复联机。
指定的节点必须是资源组的当前主节点。
- r** 继续先前由 **-s** 选项暂停的对指定资源组的自动恢复操作。
如果省略 **-g** 选项，**-r** 选项将应用于所有资源组。
在您没有明确发出命令继续执行自动恢复过程之前，已暂停的资源组不会自动重新启动或进行故障转移。无论是联机还是脱机，已暂停的数据服务都仍将处于其当前状态。您仍可在指定的节点上将资源组手动切换到不同的状态。此外，您依然可以启用或禁用资源组中的单个资源。
有关如何暂停对资源组的自动恢复操作的信息，请参见 **-s** 选项的描述。
- S** 将所有资源组和设备组切换为在指定的 *node* 上脱机。
在全局区域中执行时，此选项可以清除群集中的任何指定节点。
系统尝试根据为每个组配置的首选项选择新的主节点。所有清除的组未必由同一个主节点重新控制。如果无法成功从指定节点清除该节点管理的所有组，此命令将退出并显示错误。
资源组在重定位到新的主节点之前会首先脱机。如果系统无法在新的主节点中启动某个清除的资源组，该资源组可能保持脱机状态。
如果无法将设备组的主所有权更改为其他某个节点，该设备组的主所有权仍将由原始节点持有。
- s** 暂停自动恢复操作并停止指定的资源组。
如果省略 **-g** 选项，**-s** 选项将应用于所有资源组。
除非您使用此选项显式恢复监视暂停的资源组，否则该资源组不会自动启动、重新启动或进行故障转移。即使资源组的监视仍处于暂停状态，数据服务也将保持联机。您仍然可以手动切换该资源组，使其在指定节点中联机或脱机。此外，您依然可以启用或禁用资源组中的单个资源。
为了检查并修复群集中存在的问题，您可能需要暂停资源组的自动恢复过程。也就是说，您可能需要对资源组服务进行维护。
还可以指定 **-k** 选项来立即中止对受影响资源组中的资源运行的所有方法。使用 **-k** 选项可以加速资源组的停顿。如果未指定 **-k** 选项，系统会允许方法继续运行，直到这些方法退出或超过所配置的超时时间。
有关如何使资源组的自动恢复操作继续的信息，请参见 **-r** 选项的描述。
- u** 将指定的受管理资源组置于不受管理状态。
使用 **-u** 选项的一个先决条件是，必须首先禁用属于指定资源组的所有资源。

-z

此选项的作用如下：

- 启用指定资源组中的所有资源
- 将这些资源组更改为受管理状态
- 使这些资源组在所有默认主节点上联机

如果省略 -g 选项，-z 选项将应用于所有资源组。

如果未指定 -g 选项，则除了已暂停的资源组之外，scswitch 命令会尝试将所有资源组联机。

-z

请求更改指定资源组或设备组的主节点。

如果省略 -g 选项，-z 选项将应用于所有资源组。

-z 选项与 -D 选项一起使用时，可将一个或多个指定设备组切换到指定节点。对于一个设备组的切换，只能指定一个主节点名。指定了多个设备组时，-D 选项会按照指定顺序切换这些设备组。如果 -z -D 操作遇到错误，该操作将会停止，不会执行进一步的切换。

-z 选项仅与 -g 选项一起使用时可使指定资源组（必须已经受管理）在其首选的节点中联机。这种 scswitch 命令格式不会违反资源组的强 RG affinities 而使该资源组联机，而且，如果资源组的关联性无法在任何节点中得到满足，该命令将输出一条警告消息。与 -z 选项不同，此选项不会启用任何资源、启用对任何资源的监视或者使任何资源组脱离不受管理状态。

-z 选项与 -g 和 -h 选项一起使用时，可使指定资源组在 -h 选项指定的节点中联机，并使这些资源组在其他所有群集节点中脱机。如果使用 -h 选项指定的节点列表为空 (-h ""), -z 选项会使 -g 选项指定的资源组从其所有当前主节点或区域脱机。-h 选项指定的所有节点必须是群集的当前成员，而且必须是 -g 选项指定的所有资源组的潜在主节点。-h 选项指定的节点的数目不得超过 g 选项指定的任何资源组的 -Maximum primaries 属性设置。

单独使用时 (scswitch -z)，-z 选项会将在其首选节点中未暂停的所有受管理资源组切换为联机。

如果配置一个或多个资源组的 RG_affinities 属性并发出 scswitch -z -g 命令（带有或不带 -h 选项），则可能还会切换除 -g 选项之后指定的资源组以外的其他资源组。[rg_properties\(5\) \[1129\]](#) 中介绍了 RG_affinities。

附加选项

可将以下附加选项与上述基本选项组合使用，如下所述：

-D

指定一个或多个设备组的名称。

此选项仅能与 -F、-m 和 -z 选项一起使用。

要将此命令选项与 -F、-m 或 -z 选项（连同 -h 选项）一起使用，您需要具有 `solaris.cluster.device.admin` 基于角色的访问控制 (role-based access control, RBAC) 授权。请参见 [Unresolved link to "rbac5"](#)。

您还必须能够承担已向其分配 Oracle Solaris Cluster 命令权限配置文件的角色，才能使用该命令。经授权的用户可以通过 `pfsh`、`pfcsch` 或 `pfksh` 配置文件 shell 在命令行上发出 Oracle Solaris Cluster 特权命令。配置文件 shell 是一种特殊的 shell，通过该 shell，可以访问分配给 Oracle Solaris Cluster 命令权限配置文件的 Oracle Solaris Cluster 特权命令。当您运行 `su` 以承担某个角色时，即会启动配置文件 shell。还可以使用 `pfexec` 发出 Oracle Solaris Cluster 特权命令。

-f

指定错误 *flag-name*。

此选项仅能与 -c 选项一起使用。

当前支持的唯一错误标志是 `Stop_failed`。

要将此命令选项与 -c 选项一起使用，您需要具有 `solaris.cluster.resource.admin` RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

您还必须能够承担已向其分配 Oracle Solaris Cluster 命令权限配置文件的角色，才能使用该命令。经授权的用户可以通过 `pfsh`、`pfcsch` 或 `pfksh` 配置文件 shell 在命令行上发出 Oracle Solaris Cluster 特权命令。配置文件 shell 是一种特殊的 shell，通过该 shell，可以访问分配给 Oracle Solaris Cluster 命令权限配置文件的 Oracle Solaris Cluster 特权命令。当您运行 `su` 以承担某个角色时，即会启动配置文件 shell。还可以使用 `pfexec` 发出 Oracle Solaris Cluster 特权命令。

-g

指定一个或多个资源组的名称。

此选项仅能与 -F、-o、-Q、-r、-R、-s、-u、-z 和 -Z 选项一起使用。

要将此命令选项与以下选项一起使用，您需要具有 `solaris.cluster.resource.admin` RBAC 授权：

- -F 选项
- -o 选项
- -Q 选项
- -R 选项连同 -h 选项
- -r 选项
- -s 选项
- -u 选项
- -Z 选项

- -z 选项 -h 选项

请参见 [Unresolved link to "rbac5"](#)。

您还必须能够承担已向其分配 Oracle Solaris Cluster 命令权限配置文件的角色，才能使用该命令。经授权的用户可以通过 `pfsh`、`pfcsch` 或 `pfksh` 配置文件 `shell` 在命令行上发出 Oracle Solaris Cluster 特权命令。配置文件 `shell` 是一种特殊的 shell，通过该 shell，可以访问分配给 Oracle Solaris Cluster 命令权限配置文件的 Oracle Solaris Cluster 特权命令。当您运行 `su` 以承担某个角色时，即会启动配置文件 `shell`。还可以使用 `pfexec` 发出 Oracle Solaris Cluster 特权命令。

-h

指定一个或多个群集节点的名称。

此选项仅能与 `-c`、`-e`、`-n`、`-R`、`-s` 和 `-z` 选项一起使用。

与 `-c`、`-e`、`-n`、`-R` 或 `-z` 选项一起使用时，`-h` 选项可接受以逗号分隔的节点列表。

要为 `-z` 选项指定一个空节点列表，请指定两个双引号 `" "` 作为 `-h` 选项的参数。

对于配置为具有多个主节点的资源组，`-h` 选项列出的节点名称必须全部是 `-g` 选项指定的每个资源组的有效潜在主节点。

如果某个资源组未能在 `-h` 选项指定的节点上成功启动，则该资源组可能会故障转移到其他节点。此行为由 `Failover_mode` 资源属性的设置决定。有关更多信息，请参见 [r_properties\(5\) \[1103\]](#)。

与 `-s` 选项一起使用时，`-h` 选项可指定要清除其资源组和设备组的单个节点的名称。

要将此命令选项与 `-c`、`-R` 选项（连同 `-g` 选项）、`-s` 或 `-z` 选项（连同 `-g` 选项）一起使用，您需要具有 `solaris.cluster.resource.admin` RBAC 授权。此外，要将此命令选项与 `-z` 选项（连同 `-D` 选项）一起使用，您需要具有 `solaris.cluster.device.admin` RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

您还必须能够承担已向其分配 Oracle Solaris Cluster 命令权限配置文件的角色，才能使用该命令。经授权的用户可以通过 `pfsh`、`pfcsch` 或 `pfksh` 配置文件 `shell` 在命令行上发出 Oracle Solaris Cluster 特权命令。配置文件 `shell` 是一种特殊的 shell，通过该 shell，可以访问分配给 Oracle Solaris Cluster 命令权限配置文件的 Oracle Solaris Cluster 特权命令。当您运行 `su` 以承担某个角色时，即会启动配置文件 `shell`。还可以使用 `pfexec` 发出 Oracle Solaris Cluster 特权命令。

-j

指定一个或多个 *resource* 的名称。

此选项仅能与 `-c`、`-e` 和 `-n` 选项一起使用。

要将此命令选项与 `-c`、`-e` 或 `-n` 选项一起使用，您需要具有 `solaris.cluster.resource.admin` RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

您还必须能够承担已向其分配 Oracle Solaris Cluster 命令权限配置文件的角色，才能使用该命令。经授权的用户可以通过 `pfsh`、`pfcsch` 或 `pfksh` 配置文件 `shell` 在命令行上发出 Oracle Solaris Cluster 特权命令。配置文件 `shell` 是一种特殊的 shell，通过该 shell，可以访问分配给 Oracle Solaris Cluster 命令权限配置文件的 Oracle Solaris Cluster 特权命令。当您运行 `su` 以承担某个角色时，即会启动配置文件 `shell`。还可以使用 `pfexec` 发出 Oracle Solaris Cluster 特权命令。

- `-k` 指定从某个节点成功清除资源组后阻止资源组切换回该节点的秒数。
- 如果正从节点清除，则资源组无法故障转移或自动切换到该节点，而且，在清除完成后，在此选项指定的时间（秒）内也无法进行故障转移或自动切换。可以通过在达到 `continue_evac` 指定的秒数之前使用 `scswitch -z -g -h` 命令将资源组切换到清除了资源组的节点中，覆盖 `-k` 计时器。完成此种切换后，`-k` 计时器将立即被视为过期。但是，不带 `-h` 标志的 `scswitch -z -g` 或 `-Z` 命令将继续使用 `-k` 计时器，并阻止将任何资源组切换到清除了资源组的节点。
- 此选项仅能与 `-s` 选项一起使用。必须指定介于 0 和 65535 之间的整数值。如果未指定值，将使用默认值 60 秒。
- 您需要具有 `solaris.cluster.resource.admin` RBAC 授权才能使用该命令选项。请参见 [Unresolved link to "rbac5"](#)。
- 您还必须能够承担已向其分配 Oracle Solaris Cluster 命令权限配置文件的角色，才能使用该命令。经授权的用户可以通过 `pfsh`、`pfcsch` 或 `pfksh` 配置文件 `shell` 在命令行上发出 Oracle Solaris Cluster 特权命令。配置文件 `shell` 是一种特殊的 shell，通过该 shell，可以访问分配给 Oracle Solaris Cluster 命令权限配置文件的 Oracle Solaris Cluster 特权命令。当您运行 `su` 以承担某个角色时，即会启动配置文件 `shell`。还可以使用 `pfexec` 发出 Oracle Solaris Cluster 特权命令。
- `-k` 立即中止对指定资源组中的资源运行的资源组管理器 (Resource Group Manager, RGM) 资源方法。
- 可将此选项与 `-q` 和 `-s` 选项一起使用。如果未指定 `-k` 选项，系统会允许方法继续运行，直到这些方法退出或超过所配置的超时时间。
- `-M` 启用 (`-e`) 或禁用 (`-n`) 对指定资源的监视。如果禁用了资源，则无需禁用其监视，因为该资源及其监视器都会保持脱机。
- 此选项仅能与 `-e` 和 `-n` 选项一起使用。

要将此命令选项与 `-e` 或 `-n` 选项一起使用，您需要具有 `solaris.cluster.resource.admin` RBAC 授权。请参见 [Unresolved link to "rbac5"](#)。

您还必须能够承担已向其分配 Oracle Solaris Cluster 命令权限配置文件的角色，才能使用该命令。经授权的用户可以通过 `pfsh`、`pfcsch` 或 `pfksh` 配置文件 `shell` 在命令行上发出 Oracle Solaris Cluster 特权命令。配置文件 `shell` 是一种特殊的 shell，通过该 shell，可以访问分配给 Oracle Solaris Cluster 命令权限配置文件的 Oracle Solaris Cluster 特权命令。当您运行 `su` 以承担某个角色时，即会启动配置文件 `shell`。还可以使用 `pfexec` 发出 Oracle Solaris Cluster 特权命令。

例 348 切换资源组

以下命令将 `resource-grp-2` 切换为由 `schost-1` 管理。

```
schost-1# scswitch -z -h schost-1 -g resource-grp-2
```

例 349 使受管理资源组联机而不启用监视或资源

以下命令将使 `resource-grp-2` 联机（如果该 `resource-grp-2` 已经受管理），但是不会启用当前禁用的任何资源或对这些资源的监视。

```
schost-1# scswitch -z -g resource-grp-2
```

例 350 切换配置为具有多个主节点的资源组

以下命令将配置为具有多个主节点的资源组 `resource-grp-3` 切换为由 `schost-1,schost-2,schost-3` 管理。

```
schost-1# scswitch -z -h schost-1,schost-2,schost-3 -g resource-grp-3
```

例 351 从节点移出其所有资源组和设备组

以下命令将 `schost-1` 的所有资源组和设备组切换到一个新的主节点集。

```
schost-1# scswitch -S -h schost-1
```

例 352 从节点永久移出其所有资源组和设备组

以下命令将 `schost-1` 的所有资源组和设备组切换到一个新的主节点集。该命令还指定在允许资源组和设备组切换回 `schost-1` 之前需等待 120 秒。

在以下命令中使用 `-K` 选项可阻止资源组在成功从 `schost-1` 清除后自动切换回 `schost-1`。资源组可能尝试切换回 `schost-1` 的一个示例情况是该资源组未能成功在其新主节点上启动。另一个示例情况是资源组具有使用 `RG_affinities` 属性配置的强负关联性。

```
schost-1# scswitch -S -h schost-1 -K 120
```

例 353 禁用资源

```
schost-1# scswitch -n -j resource-1,resource-2
```

例 354 启用资源

```
schost-1# scswitch -e -j resource-1
```

例 355 使资源组进入不受管理状态

```
schost-1# scswitch -u -g resource-grp-1,resource-grp-2
```

例 356 使资源组脱离不受管理状态

```
schost-1# scswitch -o -g resource-grp-1,resource-grp-2
```

例 357 切换设备组

以下命令将 `device-group-1` 切换为由 `schost-2` 管理。

```
schost-1# scswitch -z -h schost-2 -D device-group-1
```

例 358 将设备组置于维护模式

以下命令将 `device-group-1` 置于维护模式。

```
schost-1# scswitch -m -D device-group-1
```

例 359 使资源组停顿

以下命令将资源组 `RG1` 和 `RG2` 置于停顿状态。

```
schost-1# scswitch -Q -g RG1,RG2
```

例 360 通过切换资源组清除 `Start_failed` 资源状态

`Start_failed` 资源状况指示 `Start` 或 `Prenet_start` 方法对某个资源失败或超时，但该资源的资源组仍已联机。即使该资源处于故障状况，可能不提供服务，其资源组仍然会

联机。如果资源的 `Failover_mode` 属性设置为 `None` 或者设置为阻止资源组故障转移的另一个值，则可能会出现此状况。

与 `Stop_failed` 资源状况不同，`Start_failed` 资源状况不会阻止您或 Oracle Solaris Cluster 软件对资源组执行操作。您无需发出 `scswitch -c` 命令来清除 `Start_failed` 资源状态。只需执行一个可重新启动资源的命令。

以下命令清除 `resource-grp-2` 资源组中的资源发生的 `Start_failed` 资源状态。该命令通过将资源组切换到 `schost-2` 节点来清除此状态。

```
schost-1# scswitch -z -h schost-2 -g resource-grp-2
```

例 361 通过重新启动资源组清除 `Start_failed` 资源状态

以下命令清除 `resource-grp-2` 资源组中的资源发生的 `Start_failed` 资源状态。此命令通过在 `schost-1` 节点上重新启动该资源组来清除此状态。

有关 `Start_failed` 资源状态的更多信息，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

```
schost-1# scswitch -R -h schost-1 -g resource-grp-2
```

例 362 通过禁用然后启用资源清除 `Start_failed` 资源状态

以下命令通过禁用然后重新启用 `resource-1` 资源来清除该资源发生的 `Start_failed` 资源状态。

有关 `Start_failed` 资源状态的更多信息，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

```
schost-1# scswitch -n -j resource-1
schost-1# scswitch -e -j resource-1
```

此命令停顿，直到请求的操作全部完成或发生错误为止。

返回以下退出值：

0 命令已成功完成。

非零值 发生错误。scswitch 将错误消息写入标准错误。

如果 `scswitch` 命令退出时退出状态为非零值并且显示错误消息 "cluster is reconfiguring" (群集正在重新配置)，尽管出错，请求的操作仍然可能已经成功完成。如果您对结果不确定，可以在重新配置完成后使用相同参数再次执行 `scswitch` 命令。

如果 `scswitch` 命令退出时退出状态为非零值并且显示错误消息 "Resource group failed to start on chosen node and may fail over to other node(s)" (资源组未能在所选节

点上启动，可能故障转移到其他节点），该资源组在 `scswitch` 命令退出后仍将继续重新配置一段时间。资源组的其他 `scswitch` 或 `clresourcegroup` 操作将失败，直到该资源组在所有节点上达到终端状态，例如 `Online`、`Online_faulted` 或 `Offline` 状态。

如果对多个资源或资源组调用 `scswitch` 命令并且发生多个错误，退出值可能只反映其中一个错误。为了避免这种情况，请一次只对一个资源或资源组调用 `scswitch` 命令。

对于 `RG_system` 属性为 `True` 的资源组（及其资源），不允许执行某些操作。有关更多信息，请参见 [rg_properties\(5\) \[1129\]](#)。

有关下列属性的说明，请参见 [Unresolved link to " attributes5"](#)：

属性类型	属性值
可用性	<code>ha-cluster/system/core</code>
接口稳定性	过时

[Unresolved link to " kill1"](#)、[Unresolved link to " pfsh1"](#)、[Unresolved link to " pfexec1"](#)、[Unresolved link to " pfksh1"](#)、[Unresolved link to " pfsh1"](#)、[Intro\(1CL\) \[17\]](#)、[cldevicegroup\(1CL\) \[65\]](#)、[clresourcegroup\(1CL\) \[253\]](#)、[Unresolved link to " su1M"](#)、[Unresolved link to " attributes5"](#)、[Unresolved link to " rbac5"](#)、[r_properties\(5\) \[1103\]](#)、[rg_properties\(5\) \[1129\]](#)

[Unresolved link to " Oracle Solaris Cluster 数据服务规划和管理指南"](#)

如果您通过将 `-z` 或 `-F` 选项与 `-g` 选项一起使用来使资源组脱机，则该资源组的 `Offline` 状态在节点重新引导后不会保持。即使某个节点消亡或加入群集或者如果正在切换其他资源组，该资源组可能仍会联机。即使您之前将该资源组切换为脱机状态，它也会在节点中联机。即使所有资源都被禁用，该资源组还是会进入联机状态。

要禁止资源组自动联机，可使用 `-s` 选项暂停该资源组的自动恢复操作。要继续自动恢复操作，可使用 `-r` 选项。

名称

sctelemetry — 初始化系统资源监视

```
sctelemetry -d  
sctelemetry -e  
sctelemetry -i -o hasp_rg=rg,hasp_rs=rs [,hasp_mnt_pt=mnt_pt]  
    [,db_rg=rg] [,db_rs=rs] [,telemetry_rg=rg] [,telemetry_rs=rs]  
sctelemetry -i -o hasp_mnt_pt=mnt_pt,hasp_nodelist=node[:...]  
    [,hasp_rs=rs] [,db_rg=rg] [,db_rs=rs]  
    [,telemetry_rg=rg,telemetry_rs=rs]  
sctelemetry -u
```

注 - Oracle Solaris Cluster 软件包括面向对象的命令集。虽然 Oracle Solaris Cluster 软件仍支持原始的命令集，但是 Oracle Solaris Cluster 过程文档仅使用面向对象的命令集。有关面向对象的命令集的更多信息，请参见 [Intro\(1CL\) \[17\]](#) 手册页。

sctelemetry 命令可以初始化系统资源监视、使监视联机，以及使它脱机。初始化时，将 -o 选项与 hasp_rg=*rg*、hasp_rs=*rs* 参数一起使用可依赖于 SUNW.HAStoragePlus 类型的现有资源。将 -o 选项与 hasp_mnt_pt=*mnt_pt*、hasp_nodelist=*node*[:,...] 参数一起使用可使 sctelemetry 命令创建 SUNW.HAStoragePlus 类型的资源。有关资源类型的更多信息，请参见 [SUNW.derby\(5\) \[1175\]](#)、[SUNW.HAStoragePlus\(5\) \[1193\]](#) 和 [SUNW.SCTelemetry\(5\) \[1229\]](#) 手册页。

SUNW.SCTelemetry 是在多主节点资源组中实例化的，该资源组就是在所有群集节点上配置且不使用网络负载均衡的资源组。

仅可以在全局区域中使用此命令。

sctelemetry 的选项如下所述：

-d

禁用对系统资源使用情况的数据和存储遥测数据的数据库的收集。

仅可以在全局区域中使用此选项。

超级用户以外的其他用户需要 solaris.cluster.system.modify RBAC 授权才能使用 sctelemetry 的 -d 选项。有关更多信息，请参见 [Unresolved link to "rbac5"](#) 手册页。

-e

使系统资源使用情况数据的收集联机。默认情况下，当您使用 sctelemetry 命令的 -i 选项时，系统资源监视处于联机状态。

仅可以在全局区域中使用此选项。

超级用户以外的其他用户需要 `solaris.cluster.system.modify` RBAC 授权才能使用 `sctelemetry` 的 `-e` 选项。有关更多信息，请参见 [Unresolved link to "rbac5" 手册页](#)。

`-i`

创建包含 `SUNW.SCTelemetry` 和 `SUNW.derby` 类型的资源的资源组。默认情况下，当您使用 `-i` 选项创建这些资源和资源组时，系统资源监视处于联机状态。

仅可以在全局区域中使用此选项。

超级用户以外的其他用户需要 `solaris.cluster.system.modify` RBAC 授权才能使用 `sctelemetry` 的 `-i` 选项。有关更多信息，请参见 [Unresolved link to "rbac5" 手册页](#)。

`-o hasp_rg=rg,hasp_rs=rs[,hasp_mnt_pt=mnt_pt][,db_rg=rg][,db_rs=rs] [,telemetry_rg=rg][,telemetry_rs=rs]`

当与 `-i` 选项一起使用时，可标识数据库要使用的 `SUNW.HAStoragePlus` 类型的资源以及包含该资源的资源组。数据收集工具必须对 `SUNW.HAStoragePlus` 的文件系统具有访问权限。

参数如下所述：

`hasp_rg=rg`

包含 `SUNW.HAStoragePlus` 类型的资源的资源组，该资源用于系统资源监视。必须指定该资源组的名称 `rg`。

`hasp_rs=rs`

用于系统资源监视的 `SUNW.HAStoragePlus` 类型的资源。必须指定该资源的名称 `rs`。

`hasp_mnt_pt=mnt_pt`

`sctelemetry` 在其上存储数据库文件以进行系统资源监视的挂载点。该挂载点必须是资源 `hasp_rs` 的属性。如果 `hasp_rs` 中有多个挂载点，则必须指定该挂载点。

`db_rg=rg`

`sctelemetry` 在其中配置 `SUNW.derby` 类型的资源的资源组。可以指定该资源组的名称 `rg`。

`db_rs=rs`

`sctelemetry` 配置的 `SUNW.derby` 类型的资源。可以指定该资源的名称 `rs`。

`telemetry_rg=rg`

`sctelemetry` 在其中配置 `SUNW.SCTelemetry` 类型的资源的资源组。可以指定该资源组的名称 `rg`。

telemetry_rs=*rs*

sctelemetry 配置的 SUNW.SCTelemetry 类型的资源。可以指定该资源的名称 *rs*。

-o hasp_mnt_pt=*mnt_pt*,hasp_nodelist=*node*:[...][,hasp_rs=*rs*][,db_rg=*rg*] [,db_rs=*rs*]
[,telemetry_rg=*rg*][,telemetry_rs=*rs*]

与 -i 选项一起使用时，指定可访问其上用于数据收集的 SUNW.HAStoragePlus 文件系统的节点，并指定 Oracle Solaris Cluster 在其中存储系统资源数据的文件系统的挂载点。

参数如下所述：

hasp_mnt_pt=*mnt_pt*

sctelemetry 用来配置 SUNW.HAStoragePlus 类型的资源的挂载点。必须指定挂载点的名称 *mnt_pt*。在创建 HAStoragePlus 资源之前，必须配置共享存储。该挂载点指的是共享存储，必须显示在 /etc/vfstab 中，如下所示：

```
/dev/md/ddg/dsk/d20 /dev/md/ddg/rdisk/d20 /mntpt ufs 2 no logging
```

hasp_nodelist=*node*:[...]

sctelemetry 用来配置 SUNW.HAStoragePlus 类型的资源的节点。必须指定节点的名称 *node*:[...]

hasp_rs=*rs*

sctelemetry 配置的 SUNW.HAStoragePlus 类型的资源。可以指定该资源的名称 *rs*。

db_rg=*rg*

sctelemetry 在其中配置 SUNW.derby 类型的资源的资源组。可以指定该资源组的名称 *rg*。

db_rs=*rs*

sctelemetry 配置的 SUNW.derby 类型的资源。可以指定该资源的名称 *rs*。

telemetry_rg=*rg*

sctelemetry 在其中配置 SUNW.SCTelemetry 类型的资源的资源组。可以指定该资源组的名称 *rg*。

telemetry_rs=*rs*

sctelemetry 配置的 SUNW.SCTelemetry 类型的资源。可以指定该资源的名称 *rs*。

-u

删除之前通过使用 -i 选项创建的资源 and 资源组。

仅可以在全局区域中使用此选项。

超级用户以外的其他用户需要 `solaris.cluster.system.modify` RBAC 授权才能使用 `sctelemetry` 命令的 `-u` 选项。有关更多信息，请参见 [Unresolved link to "rbac5"](#) 手册页。

存在 HAStoragePlus 资源时，初始化系统资源监视

本示例初始化系统资源监视，并验证监视是否已初始化。本示例假定您具有可用于系统资源监视的 `SUNW.HAStoragePlus` 资源。

本示例不指定资源 `db_rs` 和 `telemetry_rs` 或资源组 `db_rg` 和 `telemetry_rg` 的名称。`sctelemetry` 命令提供这些资源和资源组的默认名称。

`scstat -g` 命令的输出显示系统资源监视中所涉及的资源和资源组之间的关系。该输出还显示 `db_rs` 和 `hasp_rs` 资源以及 `db_rg` 资源组在某个节点上各自联机，`telemetry_rg` 和 `telemetry_rs` 在所有群集节点上都联机。

```
# sctelemetry -i \  
-o hasp_mnt_pt=DBDATA,hasp_nodelist=l6-lx-1:l6-lx-4,hasp_rs=anto  
  
# scstat -g
```

禁用系统资源监视

本示例禁用系统资源监视，然后验证监视是否已禁用。禁用监视后，`scstat -g` 命令的输出将显示 `db_rs`、`hasp_rs` 和 `telemetry_rs` 资源以及 `db_rg` 和 `telemetry_rg` 资源组均脱机。

```
# sctelemetry -d  
# scstat -g
```

返回以下退出值：

0 命令已成功完成。

非零值 发生错误。

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

[Intro\(1CL\)](#) [17]、[cltelemetryattribute\(1CL\)](#) [447]、[cluster\(1CL\)](#) [461]、[scstat\(1M\)](#) [745]、[sctelemet](#)

名称

scversions — Oracle Solaris Cluster 版本管理

scversions [-c]

注 - Oracle Solaris Cluster 软件包括面向对象的命令集。虽然 Oracle Solaris Cluster 软件仍支持原始的命令集，但是 Oracle Solaris Cluster 过程文档仅使用面向对象的命令集。有关面向对象的命令集的更多信息，请参见 [Intro\(1CL\) \[17\]](#) 手册页。

scversions 命令可在滚动升级到新 Oracle Solaris Cluster 软件后将群集提升到新功能级别。不带任何参数时，scversions 命令输出一条消息，提示是否需要提升。

支持以下操作数：

-c	将当前是群集活动成员的节点集提升到可能的最高功能级别。 升级节点（通过升级到产品的新发行版或通过应用修补程序）并将其引导回群集中时，该节点上的某些内部协议可能不得不以较低版本运行，以便与群集中的其他节点正确协作。当群集处于此状态时，某些管理操作可能会被禁用，而且在升级中引入的某些新功能可能不可用。 在升级所有节点后从任何节点运行一次此命令时，该群集将会切换到内部协议的可能的最高版本。假设所有节点当时都安装了同一 Oracle Solaris Cluster 软件，则所有新功能都会可用，并且会删除任何管理限制。 如果在运行带 -c 选项的 scversions 命令时，某个尚未升级的节点是群集的活动成员，则该命令不起作用，因为此群集已经在可能的最高功能级别运行。 如果在运行带 -c 选项的 scversions 命令时，某个节点尚未升级而且不是群集的活动成员（例如该节点为进行维护而关闭），则此群集的内部协议将会升级到可能的最高版本。您可能需要升级不是群集活动成员的节点，使其可以重新加入该群集。
0	成功
非零值	失败

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

[scinstall\(1M\) \[697\]](#)

OSC4 3ha

名称

scds_calls — Oracle Solaris Cluster 数据服务开发库 (Data Services Development Library, DSDL) 函数

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l dsdev#include <rgm/libdsdev.h>
```

数据服务开发库 (Data Services Development Library, DSDL) 是一组用于封装和扩展 scha 库函数功能的较高级别的库函数。scha_calls(3HA) [897] 手册页对 scha 库函数进行了说明。

DSDL 函数是在 libdsdev.so 库中实现的。

DSDL 函数通常分为以下几类。

■ 通用函数

通用函数包括初始化函数、检索函数、故障转移与重新启动函数以及执行函数。使用这些函数可以执行以下操作：

- 初始化 DSDL 环境
- 检索资源类型、资源和资源组名称及扩展属性值
- 故障转移和重新启动资源组及重新启动资源
- 将错误字符串转换为错误消息
- 在超时的时间之内执行命令

■ 属性函数

这些函数提供了公用 API，可用来访问相关资源类型、资源和资源组的特定属性，其中包括一些常用的扩展属性。DSDL 提供 scds_initialize () 函数，可用来解析命令行参数。然后该库将缓存相关资源类型、资源和资源组的各种属性。

■ 网络资源访问函数

这些函数可用来管理资源和资源组使用的网络资源。这些函数可用来处理主机名、端口列表和网络地址，并可启用基于 TCP 的监视。

■ 进程监视器工具 (Process Monitor Facility, PMF) 函数

这些函数封装 Process Monitor Facility (PMF) 功能。

■ 故障监视器函数

这些函数通过保留故障历史记录，并与 Retry_count 和 Retry_interval 属性结合使用来对其进行评估，提供了一种预定的故障监视模型。

■ 实用程序函数

使用这些函数可以向系统日志写入消息和调试消息。

初始化函数

以下函数可初始化调用方法：

-
- [scds_initialize\(3HA\) \[847\]](#) – 分配资源和初始化 DSDL 环境。
 - [scds_close\(3HA\) \[777\]](#) – 释放由 `scds_initialize()` 函数分配的资源。

检索函数

以下函数可用于检索有关资源类型、资源、资源组和扩展属性的信息：

- [scds_free_ext_property\(3HA\) \[807\]](#) – 释放由 `scds_get_ext_property()` 分配的内存。
- [scds_get_fullname\(3HA\) \[821\]](#) – 检索区域节点名称。
- [scds_get_fullname_nodeid\(3HA\) \[823\]](#) – 检索带有 ASCII 节点 ID 号的区域节点名称。
- [scds_get_resource_type_name\(3HA\) \[833\]](#) – 检索调用程序的资源类型的名称。
- [scds_get_resource_name\(3HA\) \[831\]](#) – 检索调用程序的资源的名称。
- [scds_get_resource_group_name\(3HA\) \[829\]](#) – 检索调用程序的资源组的名称。
- [scds_get_ext_property\(3HA\) \[817\]](#) – 检索指定扩展属性的值。
- [scds_get_current_method_name\(3HA\) \[815\]](#) – 检索通过其调用数据服务方法的路径名的最后一个元素。请参见 [Unresolved link to "basename3C" 手册页](#)。
- [scds_is_zone_cluster\(3HA\) \[849\]](#) – 返回表示是否已在区域群集中配置资源的布尔值。

以下函数用于检索有关由资源使用的 SUNW.HASStoragePlus 资源的状态信息：

[scds_hasp_check\(3HA\) \[843\]](#) – 检索有关由资源使用的 SUNW.HASStoragePlus 资源的状态信息。使用为该资源定义的 `Resource_dependencies` 或 `Resource_dependencies_weak` 系统属性，从资源依赖的所有 SUNW.HASStoragePlus 资源的状态（联机或其他）中获取此信息。有关更多信息，请参见 [SUNW.HASStoragePlus\(5\) \[1193\]](#) 手册页。

故障转移和重新启动函数

以下函数可用于故障转移或重新启动资源或资源组：

- [scds_failover_rg\(3HA\) \[783\]](#) – 故障转移资源组。
- [scds_restart_rg\(3HA\) \[881\]](#) – 重新启动资源组。
- [scds_restart_resource\(3HA\) \[879\]](#) – 重新启动资源。

执行函数

以下函数用来在超时的时间之内执行命令，并将错误代码转换为错误消息。

- [scds_timerun\(3HA\) \[895\]](#) – 在超时值的时间之内执行命令。
- [scds_error_string\(3HA\) \[779\]](#) 和 [scds_error_string_i18n\(3HA\) \[781\]](#) – 将错误代码转化为错误字符串。由 `scds_error_string()` 返回的字符串以英文显示。由 `scds_error_string_i18n()` 返回的字符串将以 LC_MESSAGES 语言环境类别指定的本机语言显示。

-
- [scds_svc_wait\(3HA\) \[887\]](#) – 等待要停止的受监视进程的指定超时周期。

属性函数

这些函数提供了公用 API，可用来访问相关资源类型、资源和资源组的特定属性，其中包括一些常用的扩展属性。DSDL 提供 `scds_initialize()` 函数，可用来解析命令行参数。然后该库将缓存相关资源类型、资源和资源组的各种属性。

[scds_property_functions\(3HA\) \[871\]](#) 手册页对这些函数进行了说明，其中包括以下函数：

- `scds_get_ext_property-name`
- `scds_get_rg_property-name`
- `scds_get_rs_property-name`
- `scds_get_rt_property-name`

网络资源访问函数

可以使用这些函数来管理网络资源。

以下函数可用于处理主机名：

- [scds_get_rs_hostnames\(3HA\) \[839\]](#) – 检索由资源使用的主机名列表。
- [scds_get_rg_hostnames\(3HA\) \[835\]](#) – 检索资源组中网络资源使用的主机名列表。
- [scds_print_net_list\(3HA\) \[865\]](#) – 向 [Unresolved link to "syslog3C"](#) 写入主机名列表的内容。通常可使用该函数进行调试。
- [scds_free_net_list\(3HA\) \[809\]](#) – 释放由 `scds_get_rs_hostnames()` 或 `scds_get_rg_hostnames()` 分配的内存。

以下函数可用于处理端口列表：

- [scds_get_port_list\(3HA\) \[827\]](#) – 检索由资源使用的端口协议对的列表。
- [scds_print_port_list\(3HA\) \[869\]](#) – 向 [Unresolved link to "syslog3C"](#) 写入端口协议列表的内容。通常可使用该函数进行调试。
- [scds_free_port_list\(3HA\) \[813\]](#) – 释放由 `scds_get_netaddr_list()` 分配的内存。

以下函数可用于处理网络地址：

- [scds_get_netaddr_list\(3HA\) \[825\]](#) – 检索由资源使用的网络地址列表。
- [scds_print_netaddr_list\(3HA\) \[867\]](#) – 向 [Unresolved link to "syslog3C"](#) 写入网络地址列表的内容。通常可使用该函数进行调试。
- [scds_free_netaddr_list\(3HA\) \[811\]](#) – 释放由 `scds_get_netaddr_list()` 分配的内存。

以下函数用于启用基于 TCP 的监视。通常，故障监视器使用这些函数建立与服务的简单套接字连接，向该服务读写数据以确定其状态，然后从该服务断开连接。

这组函数包含以下函数：

- [scds_fm_tcp_connect\(3HA\) \[799\]](#) – 建立与仅使用 IPv4 寻址的进程的 TCP 连接。
- [scds_fm_net_connect\(3HA\) \[789\]](#) – 建立与使用 IPv4 或 IPv6 寻址的进程的 TCP 连接。
- [scds_fm_tcp_read\(3HA\) \[803\]](#) – 使用 TCP 连接从所监视的进程读取数据。
- [scds_fm_tcp_write\(3HA\) \[805\]](#) – 使用 TCP 连接向所监视的进程写入数据。
- [scds_simple_probe\(3HA\) \[885\]](#) – 通过建立和终止与进程的 TCP 连接来探测该进程。该函数仅用于处理 IPv4 地址。
- [scds_simple_net_probe\(3HA\) \[883\]](#) – 通过建立和终止与进程的 TCP 连接来探测该进程。该函数可用于处理 IPv4 或 IPv6 地址。
- [scds_fm_tcp_disconnect\(3HA\) \[801\]](#) – 终止与所监视的进程的连接。该函数仅用于处理 IPv4 地址。
- [scds_fm_net_disconnect\(3HA\) \[793\]](#) – 终止与所监视的进程的连接。该函数可用于处理 IPv4 或 IPv6 地址。

PMF 函数

这些函数封装进程监视器工具 (Process Monitor Facility, PMF) 函数。通过 PMF 进行监视的 DSDL 模型可以创建并使用 `pmfadm` 的隐式 `tag` 值。有关更多信息，请参见 [pmfadm\(1M\) \[625\]](#) 手册页。

PMF 工具还对 `Restart_interval`、`Retry_count` 和 `action_script` (`pmfadm` 的 `-t`、`-n` 和 `-a` 选项) 使用隐式值。最重要的是，DSDL 会将 PMF 确定的进程故障历史记录与故障监视器检测到的应用程序故障历史记录关联起来，以通过计算决定是进行重新启动还是进行故障转移。

这组函数包含以下函数：

- [scds_pmf_get_status\(3HA\) \[851\]](#) – 确定指定的实例是否在 PMF 控制下进行监视。
- [scds_pmf_restart_fm\(3HA\) \[853\]](#) – 使用 PMF 重新启动故障监视器。
- [scds_pmf_signal\(3HA\) \[855\]](#) – 向在 PMF 控制下运行的进程树发送指定信号。
- [scds_pmf_start\(3HA\) \[857\]](#) 和 [scds_pmf_start_env\(3HA\) \[859\]](#) – 在 PMF 控制下执行指定的程序 (包括故障监视器)。除了可执行与 `scds_pmf_start()` 函数一样的操作外，`scds_pmf_start_env()` 函数还可将提供的环境传送给已执行的程序。
- [scds_pmf_stop\(3HA\) \[861\]](#) – 终止在 PMF 控制下运行的进程。
- [scds_pmf_stop_monitoring\(3HA\) \[863\]](#) – 停止对在 PMF 控制下运行的进程的监视。

故障监视器函数

这些函数通过保留故障历史记录，并与 `Retry_count` 和 `Retry_interval` 属性结合使用来对故障历史记录进行评估，提供了一种预定的故障监视模型。

这组函数包含以下函数：

- `scds_fm_sleep(3HA)` [797] – 等待故障监视器控制套接字的消息。
- `scds_fm_action(3HA)` [785] – 在探测完成后执行操作。
- `scds_fm_print_probes(3HA)` [795] – 将探测状态信息写入系统日志。

实用程序函数

使用以下函数可以向系统日志写入消息和调试消息：

- `scds_syslog(3HA)` [891] – 将消息写入系统日志。
- `scds_syslog_debug(3HA)` [893] – 将调试消息写入系统日志。

`/usr/cluster/include/scds.h` 头文件

`/usr/cluster/lib/libdsdev.so` 库

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

`pmfadm(1M)` [625]、`scds_close(3HA)` [777]、`scds_error_string(3HA)` [779]、`scds_error_string_i18n(3HA)` [link to "attributes5"](#)

名称

scds_close — 释放 DSDL 环境资源

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l dsdev#include <rgm/libdsdev.h>
void scds_close(scds_handle_t*handle);
```

scds_close () 函数可回收在通过使用 [scds_initialize\(3HA\) \[847\]](#) 初始化数据服务方法期间分配的资源。在程序终止之前，调用一次此函数。

支持以下参数：

handle 从 [scds_initialize \(\)](#) 返回的句柄。

/usr/cluster/include/rgm/libdsdev.h

头文件

/usr/cluster/lib/libdsdev.so

库

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scds_initialize\(3HA\) \[847\]](#)、[Unresolved link to "attributes5"](#)

名称

scds_error_string, scds_error_string_i18n — 从错误代码生成错误字符串

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l dsdev#include <rgm/libdsdev.h>
const char *scds_error_string(scha_err_t error_code);

const char *scds_error_string_i18n(scha_err_t error_code);
```

scds_error_string () 和 scds_error_string_i18n () 函数可从 DSDL 函数返回的错误代码生成描述错误的简短字符串。由 scds_error_string () 返回的字符串以英文显示。由 scds_error_string_i18n () 返回的字符串将以 LC_MESSAGES 语言环境类别指定的本机语言显示。请参见 [Unresolved link to "setlocale3C"](#)。无效错误代码返回 NULL。

此函数返回的指针指向属于 DSDL 的内存。请勿修改此内存。

支持以下参数：

error_code DSDL 函数返回的错误代码。

/usr/cluster/include/rgm/libdsdev.h

头文件

/usr/cluster/lib/libdsdev.so

库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scha_calls\(3HA\) \[897\]](#)、[Unresolved link to "setlocale3C"](#)、[Unresolved link to "attributes5"](#)

名称

scds_error_string, scds_error_string_i18n — 从错误代码生成错误字符串

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l dsdev#include <rgm/libdsdev.h>
const char *scds_error_string(scha_err_t error_code);

const char *scds_error_string_i18n(scha_err_t error_code);
```

scds_error_string () 和 scds_error_string_i18n () 函数可从 DSDL 函数返回的错误代码生成描述错误的简短字符串。由 scds_error_string () 返回的字符串以英文显示。由 scds_error_string_i18n () 返回的字符串将以 LC_MESSAGES 语言环境类别指定的本机语言显示。请参见 [Unresolved link to "setlocale3C"](#)。无效错误代码返回 NULL。

此函数返回的指针指向属于 DSDL 的内存。请勿修改此内存。

支持以下参数：

error_code DSDL 函数返回的错误代码。

/usr/cluster/include/rgm/libdsdev.h

头文件

/usr/cluster/lib/libdsdev.so

库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scha_calls\(3HA\) \[897\]](#)、[Unresolved link to "setlocale3C"](#)、[Unresolved link to "attributes5"](#)

名称

scds_failover_rg — 对资源组进行故障转移

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l dsdev#include <rgm/libdsdev.h>scha_err_t scds_failover_rg(
scds_handle_t handle);
```

`scds_failover_rg()` 函数在传递给调用程序的资源所属的资源组上执行 [scha_control\(3HA\) \[939\]](#) SCHA_GIVEOVER 操作。

此函数执行成功后不会返回。因此，将此函数作为调用程序中最后执行的代码段。

支持以下参数：

handle 从 [scds_initialize\(3HA\) \[847\]](#) 返回的句柄。

支持以下返回值：

SCHA_ERR_NOERR 表明函数执行成功。

其他值 表明函数执行失败。有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

/usr/cluster/include/rgm/libdsdev.h

头文件

/usr/cluster/lib/libdsdev.so

库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scha_calls\(3HA\) \[897\]](#)、[scha_control\(3HA\) \[939\]](#)、[Unresolved link to "attributes5"](#)

名称

scds_fm_action — 探测完成后采取操作函数

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l dsdev#include <rgm/libdsdev.h> scha_err_t
scds_fm_action(scds_handle_t handle, int probe_status,
long elapsed_milliseconds);
```

scds_fm_action () 函数结合使用数据服务的 probe_status 和过去的故障历史记录，来采取以下操作之一：

- 重新启动应用程序。
- 故障转移资源组。
- 不执行任何操作。

使用输入 probe_status 参数的值指明故障的严重级别。例如，您可能会将连接到应用程序过程中出现的故障视为完全故障，而将断开连接过程中出现的故障视为部分故障。对于后一种情况，您需要为 probe_status 指定一个介于 0 到 SCDS_PROBE_COMPLETE_FAILURE 之间的值。

DSDL 将 SCDS_PROBE_COMPLETE_FAILURE 定义为 100。无论部分探测是成功还是失败，均使用介于 0 与 SCDS_PROBE_COMPLETE_FAILURE 之间的值。

DSDL 将 SCDS_PROBE_IMMEDIATE_FAILOVER 定义为 201。除非 Failover_mode 属性设置为 RESTART_ONLY 或 LOG_ONLY，否则，此探测状态将触发资源组的即时故障转移。要强制实施即时故障转移而不首先尝试重新启动，请使用特殊的 SCDS_PROBE_IMMEDIATE_FAILOVER 值。有关 Failover_mode 属性的更多信息，请参见 [r_properties\(5\) \[1103\]](#)

通过将资源的 Retry_interval 属性定义的时间间隔中的 probe_status 输入参数的值相加，对 scds_fm_action () 的连续调用可以计算故障历史记录。任何早于 Retry_interval 的故障历史记录都会从内存中清除，并且做出重新启动或故障转移决策时不再使用这些记录。

scds_fm_action () 函数使用以下算法选择要采取的操作：

重新启动

如果累积的故障历史记录达到 SCDS_PROBE_COMPLETE_FAILURE，则 scds_fm_action () 会通过先调用资源的 STOP 方法然后调用 START 方法来重新启动资源。它会忽略为资源类型定义的任何 PRENET_START 或 POSTNET_STOP 方法。

除非已设置资源，否则将通过调用 scha_resource_setstatus 将该资源的状态设置为 SCHA_RSSTATUS_DEGRADED ()。

如果尝试重新启动操作因资源的 START 或 STOP 方法发生故障而失败，则会随 GIVEOVER 选项一起调用 scha_control () 以将资源组故障转移到其他节点。

如果 `scha_control()` 调用成功，则资源组将故障转移到其他群集节点，对 `scds_fm_action()` 的调用从不返回。

重新启动成功时，会清除故障历史记录。仅当该故障历史记录再次累积到 `SCDS_PROBE_COMPLETE_FAILURE` 时，才会再次尝试重新启动。

故障转移

当对 `scds_fm_action()` 的连续调用尝试执行的重新启动次数达到为资源定义的 `Retry_count` 值时，会尝试通过一起调用 `scha_control()` 与 `GIVEOVER` 选项来执行故障转移。

除非已设置资源，否则将通过调用 `scha_resource_setstatus` 将该资源的状态设置为 `SCHA_RSSTATUS_FAULTED()`。

如果 `scha_control()` 调用失败，则 `scds_fm_action()` 维护的整个故障历史记录都会被清除。

如果 `scha_control()` 调用成功，则资源组将故障转移到其他群集节点，对 `scds_fm_action()` 的调用从不返回。

通过将 `probe_status` 值指定为 `SCDS_PROBE_IMMEDIATE_FAILOVER`，探测可以触发即时故障转移尝试而不执行任何重新启动。

无操作

如果累积的故障历史记录维持在 `SCDS_PROBE_COMPLETE_FAILURE` 以下，则无须采取任何操作。此外，如果 `probe_status` 的值为 0，则表明无论故障历史记录如何，都将成功执行服务检查并且不采取任何操作。

除非已设置资源，否则将通过调用 `scha_resource_setstatus()` 将该资源的状态设置为 `SCHA_RSSTATUS_OK`。

支持以下参数：

handle

从 [scds_initialize\(3HA\) \[847\]](#) 返回的句柄。

probe_status

您指定的介于 0 与 `SCDS_PROBE_COMPLETE_FAILURE` 或 `SCDS_PROBE_IMMEDIATE_FAILOVER` 之间的表示数据服务状态的一个数字。

- 值 0 表明最近的数据服务检查已成功。
- 值 `SCDS_PROBE_COMPLETE_FAILURE` 意味着完全故障，表明服务已完全出现故障。您还可以提供介于 0 与 `SCDS_PROBE_COMPLETE_FAILURE` 之间的某个值，表明服务只是部分故障。
- 值 `SCDS_PROBE_IMMEDIATE_FAILURE` 将触发资源组的故障转移而不执行任何重新启动，除非 `Failover_mode` 属性设置为 `RESTART_ONLY` 或 `LOG_ONLY`。有关 `Failover_mode` 属性的更多信息，请参见 [r_properties\(5\) \[1103\]](#)

`elapsed_milliseconds`

完成数据服务检查所需的时间（毫秒）。保留该值供将来使用。

`scds_fm_action()` 函数返回以下值：

0 该函数成功。

非零值 该函数失败。

`SCHA_ERR_NOERR`

未采取任何操作或者已成功执行重新启动。

`SCHA_ERR_FAIL`

尝试执行故障转移但没有成功。

`SCHA_ERR_NOMEM`

系统内存不足。

`/usr/cluster/include/rgm/libdsdev.h`

头文件

`/usr/cluster/lib/libdsdev.so`

库

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scds_fm_sleep\(3HA\) \[797\]](#)、[scds_initialize\(3HA\) \[847\]](#)、[scha_calls\(3HA\) \[897\]](#)、[scha_control\(3HA\) \[938\]](#)、[Unresolved link to "attributes5"](#)

名称

scds_fm_net_connect — 建立与应用程序的 TCP 连接

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib -l dsdev#include <rgm/  
libdsdev.h> schar_t scds_fm_net_connect(scds_handle_t handle, scds_socket_t  
*socklist, int count, scds_netaddr_t addr, time_t timeout);
```

scds_fm_net_connect () 函数可建立到受监视进程的一个或多个 TCP 连接（取决于每个地址的 Port_list 协议值，如下所述）。

您可以通过使用 [scds_get_netaddr_list\(3HA\) \[825\]](#) 检索资源的网络地址列表。该调用还会为列表中的每个地址填充协议值。如果 Port_list 中为该地址指定的协议是 tcp6，则协议值将设置为 SCDS_IPPROTO_TCP6。如果 Port_list 中为该地址指定的协议是 tcp，或者未在 Port_list 中指定任何协议，则协议值将设置为 SCDS_IPPROTO_TCP。

此函数还会解析 addr 中提供的 hostname 并连接到：

- hostname 在指定端口的 IPv4 地址（如果 addr 中指定的协议是 SCDS_IPPROTO_TCP）。
- hostname 在指定端口的 IPv4 地址（如果存在一个）和 IPv6 地址（如果存在一个）（如果 addr 中指定的协议是 SCDS_IPPROTO_TCP6）。状态和文件描述符（如果适用）将存储在提供给此函数的 scds_socket_t 数组中。此数组的第一个成员用于 IPv4 映射，第二个成员用于 IPv6 映射。可将状态设置为以下值之一：
 - SCDS_FMSOCK_OK – 操作成功且关联的套接字文件描述符有效。
 - SCDS_FMSOCK_NA – 地址类型（IPv4 或 IPv6）不适用于此 hostname。如果 hostname 仅包含一个或多个 IPv4 映射，则传递给此函数的数组中第二个成员的状态会设置为 SCDS_FMSOCK_NA。关联的套接字文件描述符将设置为未知值，且永不应使用。
 - SCDS_FMSOCK_ERR – 操作失败或超时。关联的套接字文件描述符将设置为未知值，且永不应使用。

支持以下参数：

handle

由 [scds_initialize\(3HA\) \[847\]](#) 返回的句柄。

socklist

scds_socket_t 类型的 SCDS_MAX_IPADDR_TYPES 成员构成的数组。数组中的每个成员为 TCP 连接保存一种状态和一个套接字文件描述符。此参数是由此函数设置的输出参数。

count

socklist 数组中成员的数量。将此参数设置为 SCDS_MAX_IPADDR_TYPES。

addr

指定进程侦听的 hostname、TCP 端口号和协议标识符。

timeout

超时值 (秒)。每个套接字在超时之前等待建立连接的时间相同。因为这些时间间隔以并行方式推进，所以该值实际上是该函数的最大执行时间。

scds_fm_net_connect () 函数返回以下值：

0 该函数成功。至少已连接一个套接字。

SCHA_ERR_INVAL 调用该函数时的参数无效。

非零值 由于超时、连接被拒绝或某些其他错误，无法建立一个连接。您可以检查 socklist 数组中所有成员的设置，将 SCDS_FMSOCK_ERR 的 status 字段来确定确切的错误。

SCHA_ERR_NOERR

表明函数执行成功。

SCHA_ERR_INTERNAL

表明执行函数时发生内部错误。

SCHA_ERR_STATE

表明连接请求被服务器拒绝。

SCHA_ERR_TIMEOUT

表明连接请求超时。

例 363 使用 scds_fm_net_connect () 函数

```
/* this function is called repeatedly,  
   after thorough_probe_interval seconds */  
int probe(scds_handle_t scds_handle, ...)  
{  
    scds_socket_t socklist[SCDS_MAX_IPADDR_TYPES];  
    ...  
  
    /* for each hostname/port/proto */
```

```

for (i = 0; i < netaddr->num_netaddrs, i++) {
    if (scds_fm_net_connect(scds_handle, socklist,
        SCDS_MAX_IPADDR_TYPES, netaddr[i], timeout) !=
        SCHA_ERR_NOERR)
    {
        /* failed completely */
        ...
    } else {
        /* at least one sock connected */
        for (j = 0, j < SCDS_MAX_IPADDR_TYPES, j++) {
            if (socklist[j].status == SCDS_FM_SOCK_NA)
                continue;

            if (socklist[j].status == SCDS_FMSOCK_ERR) {
                /* this particular connection failed */
                scds_syslog(LOG_ERR, "Failed: %s",
                    scds_error_string(socklist[j].err));
                continue;
            }

            /* use socklist[i].fd to perform write/read */
            ...
        }
        (void) scds_fm_net_disconnect(scds_handle, socklist,
            SCDS_MAX_IPADDR_TYPES, remaining_time);
    }
}
...
return (result);
}

```

/usr/cluster/include/rgm/libdsdev.h

头文件

/usr/cluster/lib/libdsdev.so

库

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scds_fm_net_disconnect\(3HA\) \[793\]](#)、[scds_fm_tcp_connect\(3HA\) \[799\]](#)、[scds_get_netaddr_list\(3HA\) \[825\]](#)
[link to "attributes5"](#)

名称

scds_fm_net_disconnect — 终止与应用程序的 TCP 连接

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib -l dsdev#include <rgm/  
libdsdev.h> scha_err_t scds_fm_net_disconnect(scds_handle_t handle, scds_socket_t  
*socklist, int count, time_t timeout);
```

scds_fm_net_disconnect () 函数可终止到受监视进程的一个或多个 TCP 连接。

尝试在指定的 timeout 时间间隔内关闭 socklist 数组中的所有有效套接字连接。返回时，socklist 中的每个成员都包含 SCDS_FMSOCK_NA 值。

支持以下参数：

handle

由 [scds_initialize\(3HA\) \[847\]](#) 返回的句柄。

socklist

[scds_fm_net_connect\(3HA\) \[789\]](#) 返回的套接字列表。此参数是输入/输出参数。

count

socklist 数组中成员的数量。将此参数设置为 SCDS_MAX_IPADDR_TYPES。

timeout

超时值 (秒)。每个套接字在超时之前等待断开连接的时间相同。因为这些时间间隔以并行方式推进，所以该值实际上是该函数的最大执行时间。

scds_fm_net_disconnect () 函数返回以下值：

- | | |
|------------------|---|
| 0 | 该函数成功。 |
| SCHA_ERR_INVALID | 调用该函数时的参数无效。 |
| 其他非零值 | 该函数失败。有关失败代码的含义，请参见 scha_calls(3HA) [897] 。 |

/usr/cluster/include/rgm/libdsdev.h

头文件

/usr/cluster/lib/libdsdev.so

库

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scds_fm_net_connect\(3HA\) \[789\]](#)、[scds_fm_tcp_disconnect\(3HA\) \[801\]](#)、[scds_initialize\(3HA\) \[847\]](#)、[s](#)
[link to "attributes5"](#)

名称

scds_fm_print_probes — 显示探测调试信息

```
cc [flags...] -I /usr/cluster/include/file -L /usr/cluster/lib -l dsdev#include <rgm/libdsdev.h>
void scds_fm_print_probes(scds_handle_t handle, int debug_level);
```

`scds_fm_print_probes()` 函数可将通过 [scds_fm_action\(3HA\) \[785\]](#) 报告的探测状态信息写入系统日志中。此信息包括 DSDL 维护的所有探测状态历史记录和与探测状态相关联的时间戳的列表。

DSDL 将最大调试级别 `SCDS_MAX_DEBUG_LEVEL` 定义为 9。

如果指定比当前所用调试级别更高的 `debug_level`，则不会写入任何信息。

支持以下参数：

`handle` 从 [scds_initialize\(3HA\) \[847\]](#) 返回的句柄。

`debug_level` 写入数据的调试级别。该值是 1 到 `SCDS_MAX_DEBUG_LEVEL`（由 DSDL 定义为 9）之间的一个整数。

`/usr/cluster/include/rgm/libdsdev.h`

头文件

`/usr/cluster/lib/libdsdev.so`

库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving（发展中）

[scds_fm_action\(3HA\) \[785\]](#)、[scds_initialize\(3HA\) \[847\]](#)、[scds_syslog_debug\(3HA\) \[893\]](#)、[Unresolved link to "attributes5"](#)

名称

`scds_fm_sleep` — 等待故障监视器控制套接字上发送的消息

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib -l dsdev#include <rgm/libdsdev.h>
    scha_err_t scds_fm_sleep(scds_handle_t handle, time_t timeout);
```

`scds_fm_sleep()` 函数等待在进程监视器工具控制下运行的数据服务应用程序进程树停止。如果在指定的超时时间段内未发生此类停止，该函数将返回 `SCHA_ERR_NOERR`。

如果发生数据服务应用程序进程树停止，则 `scds_fm_sleep()` 会在失败历史记录中记录 `SCDS_COMPLETE_FAILURE`，并重新启动进程树或根据 [scds_fm_action\(3HA\) \[785\]](#) 手册页中所述的算法对其进行故障转移。如果故障转移尝试不成功，则会尝试重新启动应用程序。

如果尝试重新启动失败，该函数将返回 `SCHA_ERR_INTERNAL`。

请注意，如果失败历史记录引发此函数执行故障转移，且故障转移尝试成功，则 `scds_fm_sleep()` 永不会返回。

支持以下参数：

`handle` 从 [scds_initialize\(3HA\) \[847\]](#) 返回的句柄。

`timeout` 以秒为单位度量的超时时间段。

`scds_fm_sleep()` 函数返回以下值：

0 该函数成功。

非零值 该函数失败。

`SCHA_ERR_NOERR` 表明进程树尚未停止。

`SCHA_ERR_INTERNAL` 表明数据服务应用程序进程树已停止，且无法重新启动。

其他值 表明函数执行失败。有关失败代码的含义，请参见 [scha_calls\(3HA\) \[897\]](#)。

`/usr/cluster/include/rgm/libdsdev.h`

头文件

/usr/cluster/lib/libdsdev.so

库

有关以下属性的描述，请参见 [Unresolved link to " attributes5"](#) :

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scha_calls\(3HA\) \[897\]](#)、[scds_fm_action\(3HA\) \[785\]](#)、[scds_initialize\(3HA\) \[847\]](#)、[Unresolved link to " attributes5"](#)

名称

scds_fm_tcp_connect — 建立与应用程序的 TCP 连接

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l dsdev#include <rgm/libdsdev.h>
scha_err_t scds_fm_tcp_connect(scds_handle_t handle,
int *sock, const char*hostname, int port, time_t timeout);
```

scds_fm_tcp_connect () 函数可与受监视进程建立 TCP 连接。

使用 [scds_get_rs_hostnames\(3HA\) \[839\]](#) 或 [scds_get_rg_hostnames\(3HA\) \[835\]](#) 检索主机名。

考虑使用 [scds_fm_net_connect\(3HA\) \[789\]](#) 代替此函数。

支持以下参数：

handle

由 [scds_initialize\(3HA\) \[847\]](#) 返回的句柄。

sock

指向此函数建立的套接字的句柄。此参数是由此函数设置的输出参数。

hostname

进程侦听的主机的名称。如果 hostname 仅映射到 IPv4 地址，或者同时映射到 IPv4 和 IPv6 地址，此函数将使用 IPv4 映射作为连接地址。如果 hostname 仅映射到 IPv6 地址，此函数将使用该 IPv6 映射作为连接地址。

port

TCP 端口号。

timeout

超时值（秒）。

scds_fm_tcp_connect () 函数返回以下值：

0 该函数成功。

非零值 该函数失败。

SCHA_ERR_NOERR

表明函数执行成功。

SCHA_ERR_STATE

表明尝试在套接字上发起连接时由于超时以外的原因失败。

SCHA_ERR_TIMEOUT

表明函数超时。

其他值

表明函数执行失败。有关失败代码的含义，请参见 [scha_calls\(3HA\) \[897\]](#)。

/usr/cluster/include/rgm/libdsdev.h

头文件

/usr/cluster/lib/libdsdev.so

库

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#) :

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Deprecated (已过时)

[scds_fm_net_connect\(3HA\) \[789\]](#)、[scds_fm_tcp_disconnect\(3HA\) \[801\]](#)、[scds_get_rg_hostnames\(3HA\) \[802\]](#)
[link to "attributes5"](#)

名称

scds_fm_tcp_disconnect — 终止与应用程序的 TCP 连接

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib -l dsdev#include <rgm/libdsdev.h>
    scha_err_t scds_fm_tcp_disconnect(scds_handle_t handle, int sock, time_t timeout);
```

scds_fm_tcp_disconnect () 函数可终止受监视进程与 TCP 的连接。

支持以下参数：

handle 由 [scds_initialize\(3HA\) \[847\]](#) 返回的句柄。

sock 上一 [scds_fm_tcp_connect\(3HA\) \[799\]](#) 调用返回的插槽编号。

timeout 超时值 (秒)。

返回以下退出值：

0 该函数成功。

非零值 该函数失败。

SCHA_ERR_NOERR 表明函数执行成功。

SCHA_ERR_TIMEOUT 表明函数超时。

其他值 表明函数失败。有关失败代码的含义，请参见 [scha_calls\(3HA\) \[897\]](#)。

/usr/cluster/include/rgm/libdsdev.h

头文件

/usr/cluster/lib/libdsdev.so

库

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api

属性类型	属性值
接口稳定性	Deprecated (已过时)

[scds_fm_net_disconnect\(3HA\) \[793\]](#)、[scds_fm_tcp_connect\(3HA\) \[799\]](#)、[scds_initialize\(3HA\) \[847\]](#)、[s](#)
link to "attributes5"

名称

scds_fm_tcp_read — 使用应用程序的 TCP 连接读取数据

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l dsdev#include <rgm/libdsdev.h>
scha_err_t scds_fm_tcp_read(scds_handle_t handle, int sock,
char *buffer, size_t *size, time_t timeout);
```

scds_fm_tcp_read () 函数可从受监视进程与 TCP 的连接读取数据。

size 参数是输入/输出参数。输入时，请指定缓冲区的大小（字节）。完成时，该函数会将数据放置在 buffer 中并指定 size 中读取的实际字节数。如果缓冲区大小不足以存放所读取的字节数，该函数将返回 size 字节的完整缓冲区，您可以为后面的数据再次调用该函数。

如果该函数超时，它会返回 SCHA_ERR_TIMEOUT。在这种情况下，该函数返回的字节数可能会少于请求值，如 size 中返回的值所示。

支持以下参数：

handle

从 [scds_initialize\(3HA\) \[847\]](#) 返回的句柄。

sock

上一 [scds_fm_tcp_connect\(3HA\) \[799\]](#) 调用返回的插槽编号。

buffer

数据缓冲区。

size

数据缓冲区大小。输入时，请指定缓冲区的大小（字节）。输出时，该函数将返回实际读取的字节数。

timeout

超时值（秒）。

scds_fm_tcp_read () 函数返回以下值：

0 该函数成功。

非零值 该函数失败。

SCHA_ERR_NOERR

表明函数执行成功。

SCHA_ERR_TIMEOUT

表明函数超时。

其他值

表明函数失败。有关失败代码的含义，请参见 [scha_calls\(3HA\) \[897\]](#)。

/usr/cluster/include/rgm/libdsdev.h

头文件

/usr/cluster/lib/libdsdev.so

库

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scds_fm_tcp_disconnect\(3HA\) \[801\]](#), [scds_fm_tcp_write\(3HA\) \[805\]](#),
[scds_initialize\(3HA\) \[847\]](#), [scha_calls\(3HA\) \[897\]](#), [Unresolved link to "attributes5"](#)

名称

scds_fm_tcp_write — 使用应用程序的 TCP 连接写入数据

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l dsdev#include <rgm/libdsdev.h>
scha_err_t scds_fm_tcp_write(scds_handle_t handle, int sock, char *buffer,
size_t *size, time_t timeout);
```

scds_fm_tcp_write () 函数可通过与受监视进程的 TCP 连接写入数据。

size 参数是输入和输出参数。输入时，请指定要写入的字节数。输出时，该函数将返回实际写入的字节数。如果 size 的输入和输出值不相等，则会发生错误。如果在写入所有请求数据之前此函数超时，则它会返回 SCHA_ERR_TIMEOUT。

支持以下参数：

handle

从 [scds_initialize\(3HA\) \[847\]](#) 返回的句柄。

sock

上一 [scds_fm_tcp_connect\(3HA\) \[799\]](#) 调用返回的插槽编号。

buffer

数据缓冲区。

size

数据缓冲区大小。输入时，请指定要写入的字节数。输出时，该函数将返回实际写入的字节数。

timeout

超时值（秒）。

scds_fm_tcp_write () 函数返回以下值：

0 该函数成功。

非零值 该函数失败。

SCHA_ERR_NOERR

表明函数执行成功。

SCHA_ERR_TIMEOUT

表明函数超时。

其他值

表明函数失败。有关失败代码的含义，请参见 [scha_calls\(3HA\) \[897\]](#)。

/usr/cluster/include/rgm/libdsdev.h

头文件

/usr/cluster/lib/libdsdev.so

库

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scds_fm_tcp_connect\(3HA\) \[799\]](#), [scds_fm_tcp_read\(3HA\) \[803\]](#),
[scds_initialize\(3HA\) \[847\]](#), [scha_calls\(3HA\) \[897\]](#), [Unresolved link to "attributes5"](#)

名称

scds_free_ext_property — 释放资源扩展属性内存

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib -l dsdev#include <rgm/libdsdev.h>
void scds_free_ext_property(scha_ext_prop_value_t *property_value);
```

scds_free_ext_property () 函数可在调用 [scds_get_ext_property\(3HA\) \[817\]](#) 期间回收已分配的内存。

支持以下参数：

property_value 指向属性值的指针。

/usr/cluster/include/rgm/libdsdev.h

头文件

/usr/cluster/lib/libdsdev.so

库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scds_get_ext_property\(3HA\) \[817\]](#)、[Unresolved link to "attributes5"](#)

名称

scds_free_net_list — 释放网络资源内存

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib -l dsdev#include <rgm/libdsdev.h>
void scds_free_net_list(scds_net_resource_list_t *net_resource_list);
```

scds_free_net_list () 函数可在调用 [scds_get_rg_hostnames\(3HA\) \[835\]](#) 或 [scds_get_rs_hostnames\(3HA\) \[839\]](#) 期间回收已分配的内存。它取消分配由 netresource_list 指向的内存。

支持以下参数：

netresource_list 指向资源组所用的网络资源列表的指针

/usr/cluster/include/rgm/libdsdev.h

头文件

/usr/cluster/lib/libdsdev.so

库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scds_get_rg_hostnames\(3HA\) \[835\]](#)、[scds_get_rs_hostnames\(3HA\) \[839\]](#)、[Unresolved link to "attributes5"](#)

名称

scds_free_netaddr_list — 释放网络地址内存

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib -l dsdev#include <rgm/libdsdev.h>
void scds_free_netaddr_list(scds_netaddr_list_t *netaddr_list);
```

scds_free_netaddr_list () 函数可在调用 [scds_get_netaddr_list\(3HA\) \[825\]](#) 期间回收已分配的内存。它取消分配由 netaddr_list 指向的内存。

支持以下参数：

netaddr_list 指向资源组所用的主机名-端口-协议三元组列表的指针。

/usr/cluster/include/rgm/libdsdev.h

头文件

/usr/cluster/lib/libdsdev.so

库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scds_get_netaddr_list\(3HA\) \[825\]](#)、[Unresolved link to "attributes5"](#)

名称

scds_free_port_list — 释放端口列表内存

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib -l dsdev#include <rgm/libdsdev.h>
void scds_free_port_list(scds_port_list_t *port_list);
```

scds_free_port_list () 函数可在调用 [scds_get_port_list\(3HA\) \[827\]](#) 期间回收已分配的内存。它取消分配由 port_list 指向的内存。

支持以下参数：

port_list 指向资源组所用的端口-协议对列表的指针

/usr/cluster/include/rgm/libdsdev.h

头文件

/usr/cluster/lib/libdsdev.so

库

有关下列属性的说明，请参见 [Unresolved link to " attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scds_get_port_list\(3HA\) \[827\]](#)、[Unresolved link to " attributes5"](#)

名称

`scds_get_current_method_name` — 检索通过其调用数据服务方法的路径名称的最后一个元素

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l dsdev#include <rgm/libdsdev.h>const char *
scds_get_current_method_name(scds_handle_t handle);
```

`scds_get_current_method_name()` 函数可返回指向字符串的指针。此字符串包含通过其调用数据服务方法的路径的最后一个元素。

有关更多信息，请参见 [Unresolved link to "basename3C"](#) 手册页。

字符串的指针指向属于数据服务开发库 (Data Service Development Library, DSDL) 的内存。请勿修改此内存。调用 `scds_close()` 会使此指针失效。

支持以下参数：

`handle` 从 `scds_initialize(3HA)` [847] 返回的句柄。

`SCHA_ERR_NOERR` 该函数成功。

有关其他错误代码的描述，请参见 `scha_calls(3HA)` [897]。

`/usr/cluster/include/rgm/libdsdev.h`

头文件

`/usr/cluster/lib/libdsdev.so`

库

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scds_close\(3HA\)](#) [777]、[scds_initialize\(3HA\)](#) [847]、[scha_calls\(3HA\)](#) [897]、[Unresolved link to "attributes5"](#)

名称

scds_get_ext_property — 检索扩展属性

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l dsdev#include <rgm/libdsdev.h>
scha_err_t scds_get_ext_property(scds_handle_t handle,
const char *property_name, scha_prop_type_t property_type,
scha_extprop_value_t **property_value);
```

scds_get_ext_property () 函数可检索给定扩展属性的值。

将首先在方法参数列表 (argv[], 由 scds_initialize () 解析) 中指定的属性列表中查找属性名称。如果属性名称不在方法参数列表中, 则使用 Oracle Solaris Cluster API 进行检索。请参见 [scha_calls\(3HA\) \[897\]](#)。

成功完成之后, 会将属性值放置在 scha_extprop_value_t 结构联合的适当变量中, 并将指向此结构的指针传递回 property_value 中的调用方。

您负责通过使用 scds_free_ext_property () 来释放内存。

您可以在 [scha_calls\(3HA\) \[897\]](#) 和 scha_types.h 头文件中找到有关数据类型 scha_prop_type_t 和 scha_extprop_value_t 的信息。

DSDL 提供了便利函数以检索某些较常用资源扩展属性的值。请参见 [scds_property_functions\(3HA\) \[871\]](#) 手册页。

支持以下参数 :

handle

从 [scds_initialize\(3HA\) \[847\]](#) 返回的句柄

property_name

要检索的属性的名称

property_type

属性值类型。在 [scha_calls\(3HA\) \[897\]](#) 和 [property_attributes\(5\) \[1087\]](#) 中定义了有效类型。

property_value

指向属性值的指针

scds_get_ext_property () 函数返回以下值 :

0 该函数成功。

非零值 该函数失败。

SCHA_ERR_PROP
RTR 文件不定义指定的属性。

SCHA_ERR_NOERR
该函数成功。

其他值
表明函数失败。有关失败代码的含义，请参见 [scha_calls\(3HA\) \[897\]](#)。

例 364 使用 `scds_get_ext_property()`

```
#include <scha_types.h>
#include <libdsdev.h>
#define INT_EXT_PROP "Int_extension_property"
...
int  retCode;
scha_extprop_value_t *intExtProp;
int  retrievedValue;
...
retCode = scds_get_ext_property(handle,
    INT_EXT_PROP, SCHA_PTYPE_INT, &intExtProp);
if (retCode != SCHA_ERR_NOERR) {
    scds_syslog(LOG_ERR,
        "Failed to retrieve the extension property %s: %s.",
        INT_EXT_PROP, scds_error_string(retCode));
    ...
} else {
    retrievedValue = intExtProp->val.val_int;
    ...
    scds_free_ext_property(intExtProp);
    ...
}
...
```

`/usr/cluster/include/rgm/libdsdev.h`

头文件

`/usr/cluster/lib/libdsdev.so`

库

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scds_free_ext_property\(3HA\) \[807\]](#)、[scds_initialize\(3HA\) \[847\]](#)、[scds_property_functions\(3HA\) \[871\]](#)、[link to "attributes5"](#)、[property_attributes\(5\) \[1087\]](#)

使用此函数仅可检索 RTR 文件中定义的扩展属性的值。请参见 [rt_reg\(4\) \[1067\]](#)。

名称

`scds_get_fullname`, `scds_get_fullname_nodeid` — 检索区域节点名称的指针。

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l dsdev#include <rgm/libdsdev.h>scha_err_t scds_get_fullname(
    const char *zonename, char **fullname,boolean_t is_zone_cluster);

scha_err_t scds_get_fullname_nodeid(const char *zonename,
    char **fullname,boolean_t is_zone_cluster);
```

`scds_get_fullname()` 函数以 `fullname` 形式返回区域节点名称。`scds_get_fullname_nodeid()` 函数以 ASCII 节点 ID 编号形式而非节点名称形式返回区域节点名称。调用方负责释放 `fullname` 指向的内存。

如果 `is_zone_cluster` 为 `true`，则 `zonename` 提供本地主机上区域群集的名称；返回的值是本地主机的区域群集节点名称。

将 `is_zone_cluster` 设置为 `false` 仅适用于 Oracle Solaris Cluster 3.3 发行版版本。该设置当前不适用于 Oracle Solaris Cluster 4.x 发行版。

`zonename` 的值必须为非 `NULL`；否则，将返回 `SCHA_ERR_INVALID` 并且 `fullname` 的值不变。

下例显示了 `scds_get_fullname` 为区域群集节点 (`is_zone_cluster` 设置为 `true`) 返回的 `fullname` 值：

```
"zcnodel"
```

如果 `zcnodel` 的节点 ID 编号为 2，则 `scds_get_fullname_nodeid` 的相应输出为：

```
"2"
```

支持以下参数：

<code>zonename</code>	提供区域群集或全局群集非全局区域的名称。
<code>is_zone_cluster</code>	指示 <code>zonename</code> 是否为区域群集名称。
<code>fullname</code>	将指向返回的节点名称字符串的输出参数。

`SCHA_ERR_NOERR` 该函数成功。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

名称

scds_get_fullname, scds_get_fullname_nodeid — 检索区域节点名称的指针。

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l dsdev#include <rgm/libdsdev.h>scha_err_t scds_get_fullname(
    const char *zonename, char **fullname,boolean_t is_zone_cluster);

scha_err_t scds_get_fullname_nodeid(const char *zonename,
    char **fullname,boolean_t is_zone_cluster);
```

scds_get_fullname () 函数以 fullname 形式返回区域节点名称。scds_get_fullname_nodeid () 函数以 ASCII 节点 ID 编号形式而非节点名称形式返回区域节点名称。调用方负责释放 fullname 指向的内存。

如果 is_zone_cluster 为 true，则 zonename 提供本地主机上区域群集的名称；返回的值是本地主机的区域群集节点名称。

将 is_zone_cluster 设置为 false 仅适用于 Oracle Solaris Cluster 3.3 发行版版本。该设置当前不适用于 Oracle Solaris Cluster 4.x 发行版。

zonename 的值必须为非 NULL；否则，将返回 SCHA_ERR_INVALID 并且 fullname 的值不变。

下例显示了 scds_get_fullname 为区域群集节点 (is_zone_cluster 设置为 true) 返回的 fullname 值：

```
"zcnodel"
```

如果 zcnodel 的节点 ID 编号为 2，则 scds_get_fullname_nodeid 的相应输出为：

```
"2"
```

支持以下参数：

zonename	提供区域群集或全局群集非全局区域的名称。
is_zone_cluster	指示 zonename 是否为区域群集名称。
fullname	将指向返回的节点名称字符串的输出参数。

SCHA_ERR_NOERR 该函数成功。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

名称

scds_get_netaddr_list — 获取资源所用的网络地址

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib -l dsdev#include <rgm/  
libdsdev.h>scha_err_t scds_get_netaddr_list(scds_handle_t handle, scds_netaddr_list_t  
**netaddr_list);
```

scds_get_netaddr_list () 函数可返回资源所使用的所有主机名、端口和协议组合。这些组合是通过将资源上的 Port_list 属性设置与资源所使用的所有主机名组合到一起而派生的，正如 scds_get_rs_hostnames () 函数所返回的那样。

在故障监视器中使用 scds_get_netaddr_list () 监视资源，并派生资源所使用的主机名、端口和协议的列表。

协议类型的值在头文件 rgm/libdsdev.h 中进行定义。

使用 scds_free_netaddr_list () 释放由此函数分配并返回的内存。

支持以下参数：

handle	由 scds_initialize () 返回的句柄
netaddr_list	资源组所用的主机名、端口和协议的列表

scds_get_netaddr_list () 函数返回以下值：

0 该函数成功。

非零值 该函数失败。

SCHA_ERR_NOERR 表明函数执行成功。

其他值 表明函数失败。有关失败代码的含义，请参见 [scha_calls\(3HA\) \[897\]](#)。

/usr/cluster/include/rgm/libdsdev.h
头文件

/usr/cluster/lib/libdsdev.so
库

有关下列属性的描述，请参见 [Unresolved link to "attributes5"](#)。

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scds_free_netaddr_list\(3HA\) \[811\]](#)、[scds_get_rs_hostnames\(3HA\) \[839\]](#)、[scha_calls\(3HA\) \[897\]](#)、[r_p](#)
[link to "attributes5"](#)

名称

scds_get_port_list — 检索资源所用的端口列表

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib -l dsdev#include <rgm/libdsdev.h>
    scha_err_t scds_get_port_list(scds_handle_t handle, scds_port_list_t **port_list);
```

scds_get_port_list () 函数可返回资源所用的端口-协议对列表。协议类型的值在头文件 netinet/in.h 中进行定义。

使用 scds_free_port_list () 释放由此函数分配并返回的内存。

支持以下参数：

handle 从 scds_initialize () 返回的句柄

port_list 资源组所用的端口-协议对的列表

scds_get_port_list () 函数返回以下值：

0 该函数成功。

非零值 该函数失败。

SCHA_ERR_NOERR 表明函数执行成功。

其他值 表明函数执行失败。有关失败代码的含义，请参见 [scha_calls\(3HA\) \[897\]](#)。

/usr/cluster/include/scha.h

头文件

/usr/cluster/lib/libscha.so

库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api

属性类型	属性值
接口稳定性	Evolving (发展中)

[scds_free_port_list\(3HA\) \[813\]](#)、[scha_calls\(3HA\) \[897\]](#)、[Unresolved link to "attributes5"](#)

名称

`scds_get_resource_group_name` — 检索资源组名称

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib -l dsdev#include <rgm/libdsdev.h>
    const char *scds_get_resource_group_name(scds_handle_t handle);
```

`scds_get_resource_group_name()` 函数可返回指向字符串的指针，该字符串是包含传递到调用程序的资源的资源组名称。指针指向属于 DSDL 的内存。请勿修改此内存。调用 `scds_close()` 会使此指针失效。

支持以下参数：

`handle` 从 `scds_initialize()` 返回的句柄

`NULL` 指示错误情形，如先前未调用 [scds_initialize\(3HA\) \[847\]](#)

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

`/usr/cluster/include/scha.h`

头文件

`/usr/cluster/lib/libscha.so`

库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scds_close\(3HA\) \[777\]](#)、[scds_initialize\(3HA\) \[847\]](#)、[scha_calls\(3HA\) \[897\]](#)、[Unresolved link to "attributes5"](#)

名称

scds_get_resource_name — 检索资源名称

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib -l dsdev#include <rgm/libdsdev.h>
    const char *scds_get_resource_name(scds_handle_t handle);
```

scds_get_resource_name () 函数可返回指向字符串的指针，该字符串包含传递到调用程序的资源的名称。指针指向属于 DSDL 的内存。请勿修改此内存。调用 scds_close () 会使此指针失效。

支持以下参数：

handle 从 scds_initialize () 返回的句柄

NULL 指示错误情形，如先前未调用 [scds_initialize\(3HA\) \[847\]](#)

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

/usr/cluster/include/rgm/libdsdev.h

头文件

/usr/cluster/lib/libdsdev.so

库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scds_close\(3HA\) \[777\]](#)、[scds_initialize\(3HA\) \[847\]](#)、[scha_calls\(3HA\) \[897\]](#)、[Unresolved link to "attributes5"](#)

名称

`scds_get_resource_type_name` — 检索资源类型名称

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib -l dsdev#include <rgm/libdsdev.h>
    const char *scds_get_resource_type_name(scds_handle_t handle);
```

`scds_get_resource_type_name()` 函数可返回指向字符串的指针，该字符串包含传递到调用程序的资源的资源类型名称。指针指向属于 DSDL 的内存。因此，请勿修改此内存。调用 `scds_close()` 会使此指针失效。

支持以下参数：

`handle` 从 `scds_initialize()` 返回的句柄

`NULL` 指示错误情形，如先前未调用 `scds_initialize()`

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

`/usr/cluster/include/rgm/libdsdev.h`

头文件

`/usr/cluster/lib/libdsdev.so`

库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scds_close\(3HA\) \[777\]](#)、[scds_initialize\(3HA\) \[847\]](#)、[scha_calls\(3HA\) \[897\]](#)、[Unresolved link to "attributes5"](#)

名称

scds_get_rg_hostnames, scds_get_rg_hostnames_zone — 获取资源组中使用的网络资源

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l dsdev#include <rgm/libdsdev.h>scha_err_t
scds_get_rg_hostnames(char *resourcegroup_name,
scds_net_resource_list_t **netresource_list);

scha_err_t scds_get_rg_hostnames_zone(char *zone_name, char *
resourcegroup_name, scds_net_resource_list_t **netresource_list);
```

#1405

scds_get_rg_hostnames () 和 scds_get_rg_hostnames_zone () 函数可检索资源组中所有网络资源使用的主机名的列表。从全局区域执行时，您可以通过 scds_get_rg_hostnames_zone () 函数从给定区域群集中的资源组检索列表。该函数返回一个指向 netresource_list 中的列表的指针。资源组可能不包含任何网络资源，也可能包含不使用网络资源的资源，所以这些函数可能会返回设置为 NULL 的 netresource_list 参数。

您可以将系统中任意资源组的名称传递给 scds_get_rg_hostnames () 和 scds_get_rg_hostnames_zone ()。使用由 scds_get_rg_hostnames () 和 scds_get_rg_hostnames_zone () 返回的主机名联系在指定资源组中运行的应用程序。

使用 scds_free_net_list () 释放由此函数分配并返回的内存。

支持以下参数

resourcegroup_name 要检索其数据的资源组的名称

netresource_list 资源组所用的网络资源的列表

scds_get_rg_hostnames () 和 scds_get_rg_hostnames_zone () 函数返回的函数可返回以下值：

0 该函数成功。

非零值 该函数失败。

SCHA_ERR_NOERR 函数成功。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

`/usr/cluster/include/rgm/libdsdev.h`

头文件

`/usr/cluster/lib/libdsdev.so`

库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scds_free_net_list\(3HA\) \[809\]](#)、[scds_get_rs_hostnames\(3HA\) \[839\]](#)、[scha_calls\(3HA\) \[897\]](#)、[Unresolved link to "attributes5"](#)

名称

scds_get_rg_hostnames, scds_get_rg_hostnames_zone — 获取资源组中使用的网络资源

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l dsdev#include <rgm/libdsdev.h>scha_err_t
scds_get_rg_hostnames(char *resourcegroup_name,
scds_net_resource_list_t **netresource_list);

scha_err_t scds_get_rg_hostnames_zone(char *zone_name, char *
resourcegroup_name, scds_net_resource_list_t **netresource_list);
```

#1405

scds_get_rg_hostnames () 和 scds_get_rg_hostnames_zone () 函数可检索资源组中所有网络资源使用的主机名的列表。从全局区域执行时，您可以通过 scds_get_rg_hostnames_zone () 函数从给定区域群集中的资源组检索列表。该函数返回一个指向 netresource_list 中的列表的指针。资源组可能不包含任何网络资源，也可能包含不使用网络资源的资源，所以这些函数可能会返回设置为 NULL 的 netresource_list 参数。

您可以将系统中任意资源组的名称传递给 scds_get_rg_hostnames () 和 scds_get_rg_hostnames_zone ()。使用由 scds_get_rg_hostnames () 和 scds_get_rg_hostnames_zone () 返回的主机名联系在指定资源组中运行的应用程序。

使用 scds_free_net_list () 释放由此函数分配并返回的内存。

支持以下参数

resourcegroup_name 要检索其数据的资源组的名称

netresource_list 资源组所用的网络资源的列表

scds_get_rg_hostnames () 和 scds_get_rg_hostnames_zone () 函数返回的函数可返回以下值：

0 该函数成功。

非零值 该函数失败。

SCHA_ERR_NOERR 函数成功。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

`/usr/cluster/include/rgm/libdsdev.h`

头文件

`/usr/cluster/lib/libdsdev.so`

库

有关以下属性的描述，请参见 [Unresolved link to " attributes5"](#) :

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scds_free_net_list\(3HA\) \[809\]](#)、[scds_get_rs_hostnames\(3HA\) \[839\]](#)、[scha_calls\(3HA\) \[897\]](#)、[Unresolved link to " attributes5"](#)

名称

scds_get_rs_hostnames — 获取某一资源所使用的网络资源

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l dsdev#include <rgm/libdsdev.h>
scha_err_t scds_get_rs_hostnames(scds_handle_t
handle,scds_net_resource_list_t **netresource_list);
```

scds_get_rs_hostnames () 函数检索资源所使用的主机名列表。如果设置了资源属性 Network_resources_used ，则主机名与 Network_resources_used 中列出的网络资源相对应。否则，它们对应于包含该资源的资源组中的所有网络资源。

该函数返回一个指向 netresource_list 中的列表的指针。资源组可以不包含任何网络资源，也可以包含不使用网络资源的资源，因此该函数会返回设置为 NULL 的 netresource_list。

使用 [scds_free_net_list\(3HA\) \[809\]](#) 释放由此函数分配并返回的内存。

支持以下参数

handle

从 [scds_initialize\(3HA\) \[847\]](#) 返回的句柄

netresource_list

资源组所使用的网络资源的列表

scds_get_rs_hostnames () 函数返回以下值：

0 该函数成功

non-zero 该函数失败

SCHA_ERR_NOERR

函数成功。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

/usr/cluster/include/rgm/libdsdev.h

头文件

/usr/cluster/lib/libdsdev.so

库

有关下列属性的说明，请参见 [Unresolved link to " attributes5"](#) :

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scds_free_net_list\(3HA\) \[809\]](#), [scds_get_rg_hostnames\(3HA\) \[835\]](#),
[scds_initialize\(3HA\) \[847\]](#), [scha_calls\(3HA\) \[897\]](#), [Unresolved link to " attributes5"](#), [r_properties\(5\) \[1103\]](#)

名称

`scds_get_zone_name` — 检索运行某个方法时所代表的区域的名称

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l dsdev#include <rgm/libdsdev.h>
const char *scds_get_zone_name(scds_handle_t handle);
```

`scds_get_zone_name()` 函数可返回指向字符串的指针。如果满足以下条件，此字符串将包含在其中运行资源组的区域的名称：

- 从全局区域调用 `scds_get_zone_name` 函数。
- 已将 `Global_zone` 资源类型属性设置为 `TRUE`。
有关 `Global_zone` 资源类型属性的信息，请参见 [rt_properties\(5\) \[1143\]](#) 手册页。
- 已在区域群集中配置资源。

在其他所有情况（包括以下情况）下，字符串为 `NULL`：

- 资源组在全局区域中进行配置。
- `Global_zone` 资源类型属性设置为 `FALSE`，或者 `Global_zone_override` 资源属性设置为 `FALSE`。

要获取实际上正在其中执行某个方法的区域的名称，请使用 `zonename` 命令。请参见 [Unresolved link to "zonename1"](#) 手册页。

字符串的指针指向属于数据服务开发库 (Data Service Development Library, DSDL) 的内存。请勿修改此内存。调用 `scds_close()` 会使此指针失效。

支持以下参数：

`handle` 从 [scds_initialize\(3HA\) \[847\]](#) 返回的句柄。

`SCHA_ERR_NOERR` 该函数成功。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

`/usr/cluster/include/rgm/libdsdev.h`

头文件

`/usr/cluster/lib/libdsdev.so`

库

有关下列属性的说明，请参见 [Unresolved link to " attributes5"](#) :

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[Unresolved link to "zonename1"](#)、[scds_close\(3HA\) \[777\]](#)、[scds_initialize\(3HA\) \[847\]](#)、[scha_calls\(3HA\) \[897\]](#)、[Unresolved link to " attributes5"](#)、[rt_properties\(5\) \[1143\]](#)

名称

`scds_hasp_check` — 获取有关由资源使用的 SUNW.HASStoragePlus 资源的状态信息

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l dsdev#include <rgm/libdsdev.h>
scha_err_t scds_hasp_check(scds_handle_t handle,
scds_hasp_status_t *hasp_status);
```

`scds_hasp_check()` () 函数检索有关由资源使用的 SUNW.HASStoragePlus 资源的状态信息。此信息是从资源所依赖的所有 SUNW.HASStoragePlus 资源的状态 (联机或其他) 获得的。此状态是通过使用为资源定义的 `Resource_dependencies`、`Resource_dependencies_weak`、`Resource_dependencies_restart` 或 `Resource_dependencies_offline_restart` 系统属性获得的。

如果 SUNW.HASStoragePlus 资源的 `FileSystemMountPoints` 属性为 `nonempty`，则当且只当 `FileSystemMountPoints` 属性中列出的所有文件系统当前都挂载在某个给定的节点上时，`scds_hasp_check()` () 函数才会认为该资源在该节点上处于联机状态。全局挂载的文件系统可以挂载在其中的 SUNW.HASStoragePlus 资源处于脱机状态的节点上。所有 `FileSystemMountPoints`，无论是在本地挂载的还是全局挂载的，都必须挂载在一个节点上，才会认为 SUNW.HASStoragePlus 资源在该节点上处于联机状态。资源的实际状态可能为联机也可能为脱机。如果这些文件系统中有任何一个没有挂载在该节点上，则会认为该资源处于脱机状态并且其实际状态为脱机。

资源类型实现可以使用 `VALIDATE` 和 `MONITOR_CHECK` 方法回调实现中的 `scds_hasp_check()` () 来确定是否应该执行特定于由 SUNW.HASStoragePlus 资源管理的任何文件系统的检查。

仅在执行该函数的同一群集环境 (全局群集或区域群集) 中检查资源依赖性。忽略 `clustername: resourcename` 形式的依赖性 (群集间依赖性)。例如，如果唯一的 HASStoragePlus 依赖性是集群间依赖性，则函数会返回状态代码 `SCDS_HASP_NO_RESOURCE`。

当函数成功时，状态代码将存储在 `hasp_status` 参数中。此代码可以是以下某一值：

`SCDS_HASP_NO_RESOURCE`

指明该资源不依赖于 SUNW.HASStoragePlus 资源。

`SCDS_HASP_NOT_ONLINE`

指明该资源所依赖的 SUNW.HASStoragePlus 资源在任何潜在主节点上均未处于联机状态。

`SCDS_HASP_ONLINE_NOT_LOCAL`

指明该资源所依赖的至少一个 SUNW.HASStoragePlus 资源在从中调用此函数的节点上未处于联机状态，但在另一节点上处于联机状态。

SCDS_HASP_ONLINE_LOCAL

指明该资源所依赖的所有 SUNW.HAStoragePlus 资源在从中调用此函数的节点上均处于联机状态。

注 - 上述状态代码的显示顺序即为它们相互间的优先级顺序。例如，如果某个 SUNW.HAStoragePlus 资源未处于联机状态，而另一个 SUNW.HAStoragePlus 资源在另一节点上处于联机状态，则状态代码会设置为 SCDS_HASP_NOT_ONLINE 而非 SCDS_HASP_ONLINE_NOT_LOCAL。

scds_hasp_check() () 函数会忽略 FilesystemMountPoints 和 Zpools 属性都设置为空列表（默认设置）的 SUNW.HAStoragePlus 资源，即使 GlobalDevicePaths 属性为 nonempty 也是如此。

支持以下参数：

handle 从 scds_initialize 返回的句柄。

hasp_status 资源使用的 SUNW.HAStoragePlus 资源的状态。

SCHA_ERR_NOERR 该函数成功。
此值还指明 hasp_status 参数中存储的状态代码有效。

SCHA_ERR_INTERNAL 该函数失败。
hasp_status 参数中存储的值未定义。忽略该未定义的值。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#) 手册页。

/usr/cluster/include/rgm/libdsdev.h

头文件

/usr/cluster/lib/libdsdev.so

库

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scds_initialize\(3HA\) \[847\]](#), [scha_calls\(3HA\) \[897\]](#), [Unresolved link to "attributes5"](#), [SUNW.HASStoragePlus\(5\) \[1193\]](#)

名称

scds_initialize — 分配和初始化 DSDL 环境

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib -l dsdev#include <rgm/libdsdev.h>
    scha_err_t scds_initialize(scds_handle_t *handleint argc, char *argv[]);
```

scds_initialize () 函数初始化 DSDL 环境。必须在使用任何其他 DSDL 函数的每个程序或故障监视器的开头调用一次该函数。

scds_initialize () 函数执行以下操作：

- 检查并处理框架传送到调用程序且必须传送到 scds_initialize () 的命令行参数 (argc 和 argv[])。调用程序无需进一步处理命令行参数。请参见 示例。
- 使用 DSDL 中其他函数所需的信息设置内部数据结构。它检索资源、资源类型和资源组属性值，并将其存储在这些数据结构中。在命令行上使用 argv[] 参数提供的任何属性的值优先于从 RGM 中检索到的值。即，如果在传送给数据服务方法的命令行参数 (argv[]) 中已指定属性的新值，那么检索该属性的值的函数将返回此新值。否则，将返回从 RGM 中检索到的现有值。
- 初始化数据服务故障监视信息
- 初始化日志记录环境。所有系统日志消息都带有以下前缀：`SC[<resourceTypeName>,<resourceGroupName>,<resourceName>,<methodName>`
将消息发送到系统日志的函数使用由 `scha_cluster_getlogfacility ()` 返回的设备。这些消息可以转发给相应的日志文件和用户。有关更多信息，请参见 [Unresolved link to "syslog.conf4"](#)。
- 验证故障监视器探测设置。它可验证 `Retry_interval` 是否大于或等于 `(Thorough_probe_interval * Retry_count)`。如果结果不为 true，则会向系统日志设备发送相应的消息。您可以在 `VALIDATE ()` 方法中调用 `scds_initialize ()` 和 `scds_close` 以验证故障监视器探测设置，即使不在 `VALIDATE` 方法中调用其他 DSDL 函数。

如果 `scds_initialize ()` 成功，必须在退出调用程序之前调用 `scds_close ()`。

如果 `scds_initialize ()` 失败，不得调用 `scds_close ()` 进行清理。在 `scds_initialize ()` 失败时，不要调用任何其他 DSDL 函数。否则，它们会返回 `SCHA_ERR_INVALID` 或 `NULL` 值。请改为调用具有非零参数的 `exit ()`。

支持以下参数：

handle 由 `scds_initialize ()` 初始化且由其他 DSDL 函数使用的句柄。

argc 传送给调用程序的参数数目。

`argv` 指向传送给调用程序的参数数组的指针。

`SCHA_ERR_NOERR` 该函数成功。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

例 365 使用 `scds_initialize ()`

```
int main(int argc, char *argv[]){
    scds_handle_t handle;

    if (scds_initialize(&handle, argc, argv) !=
        SCHA_ERR_NOERR)
        exit(1);
    ...
    /* data service code */
    ...
    scds_close(&handle);
}
```

`/usr/cluster/include/rgm/libdsdev.h`

头文件

`/usr/cluster/lib/libdsdev.so`

库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#) :

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scds_close\(3HA\) \[777\]](#)、[scds_property_functions\(3HA\) \[871\]](#)、[scha_calls\(3HA\) \[897\]](#)、[scha_cluster_g link to "syslog.conf4"](#)、[r_properties\(5\) \[1103\]](#)

名称

`scds_is_zone_cluster` — 返回表明是否已在区域群集节点中配置资源的布尔值。

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l dsdev #include <rgm/libdsdev.h>boolean_t
scds_is_zone_cluster(scds_handle_t handle);
```

如果已在区域群集中配置资源，`scds_is_zone_cluster()` 函数将返回 `B_TRUE`；否则，将返回 `B_FALSE`。

支持以下参数：

`handle` 从 [scds_initialize\(3HA\) \[847\]](#) 返回的句柄。

`SCHA_ERR_NOERR` 该函数成功。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

`/usr/cluster/include/rgm/libdsdev.h`

头文件

`/usr/cluster/lib/libdsdev.so`

库

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	SUNWscdev
接口稳定性	Evolving (发展中)

[scds_calls\(3HA\) \[771\]](#)、[scds_close\(3HA\) \[777\]](#)、[scds_initialize\(3HA\) \[847\]](#)、[scha_calls\(3HA\) \[897\]](#)、[Unresolved link to "attributes5"](#)

名称

scds_pmf_get_status — 确定是否存在受 PMF 监视的进程树

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l dsdev#include <rgm/libdsdev.h>
scha_err_t scds_pmf_get_status(scds_handle_t handle,
scds_pmf_type_t program_type, int instance, scds_pmf_status_t*pmf_status);
```

scds_pmf_get_status () 函数可确定是否在 PMF 控制下监视指定实例。此函数等同于带有 -k 选项的 [pmfadm\(1M\) \[625\]](#) 命令。

支持以下参数：

handle

从 scds_initialize () 返回的句柄

program_type

要执行的程序类型。有效类型为：

SCDS_PMF_TYPE_SVC 数据服务应用程序

SCDS_PMF_TYPE_MON 故障监视器

SCDS_PMF_TYPE_OTHER 其他

instance

对于具有多个实例的资源，该从 0 开始的整数可以唯一标识实例。对于单实例资源，请使用 0。

pmf_status

如果 PMF 正在监视指定实例，pmf_status 将设置为 SCDS_PMF_MONITORED。否则将设置为 SCDS_PMF_NOT_MONITORED。

scds_pmf_get_status () 函数返回以下值：

0 该函数成功。

non-zero 该函数失败。

SCHA_ERR_NOERR

函数成功

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

`/usr/cluster/include/rgm/libdsdev.h`

头文件

`/usr/cluster/lib/libdsdev.so`

库

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[pmfadm\(1M\) \[625\]](#), [scds_initialize\(3HA\) \[847\]](#), [scha_calls\(3HA\) \[897\]](#), [Unresolved link to "attributes5"](#)

名称

scds_pmf_restart_fm — 使用 PMF 重新启动故障监视器

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib -l dsdev#include <rgm/libdsdev.h>
    scha_err_t scds_pmf_restart_fm(scds_handle_t handle, int instance);
```

scds_pmf_restart_fm () 函数将 SIGKILL 信号发送到故障监视器进程树以终止故障监视器，然后使用 PMF 重新启动它。此函数将 MONITOR_STOP_TIMEOUT 属性用作其超时值。即，在等待进程树终止时，scds_pmf_restart_fm () 至多等待 MONITOR_STOP_TIMEOUT 属性值的时间。

如果未在 RTR 文件中明确设置 MONITOR_STOP_TIMEOUT 属性，将使用默认超时值。

使用此函数的一个方法是在 UPDATE 方法中调用它以重新启动监视器（可能带有新参数）。

支持以下参数：

handle	从 scds_initialize () 返回的句柄
instance	对于具有多个故障监视器实例的资源，该从 0 开始的整数可以唯一标识故障监视器实例。对于单实例故障监视器，请使用 0。

scds_pmf_restart_fm () 函数返回以下值：

0	该函数成功。
non-zero	该函数失败。

SCHA_ERR_NOERR 函数成功

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

/usr/cluster/include/rgm/libdsdev.h

头文件

/usr/cluster/lib/libdsdev.so

库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[pmfadm\(1M\) \[625\]](#)、[scha_calls\(3HA\) \[897\]](#)、[Unresolved link to "signal3HEAD"](#)、[Unresolved link to "attributes5"](#)、[r_properties\(5\) \[1103\]](#)

名称

scds_pmf_signal — 将信号发送到由 PMF 控制的进程树

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l dsdev#include <rgm/libdsdev.h>
scha_err_t scds_pmf_signal(scds_handle_t handle, scds_pmf_type_t
program_type, int instance, int signal, time_t timeout);
```

scds_pmf_signal () 函数将指定信号发送到在 PMF 控制下运行的进程树。此函数等同于带有 -k 选项的 [pmfadm\(1M\)](#) [625] 命令。

发送信号后，scds_pmf_signal () 函数在返回之前将等候指定的超时时间，以待进程树终止。timeout 的值 0 指示函数立即返回，不等待任何进程退出。值 -1 指示函数无限期地等待进程退出。

支持以下参数：

handle

从 scds_initialize () 返回的句柄

program_type

要执行的程序类型。有效类型为：

SCDS_PMF_TYPE_SVC	数据服务应用程序
SCDS_PMF_TYPE_MON	故障监视器
SCDS_PMF_TYPE_OTHER	其他

instance

对于具有多个实例的资源，该从 0 开始的整数可以唯一标识实例。对于单实例资源，请使用 0。

signal

要发送的 Solaris 信号。请参见 [Unresolved link to "signal3HEAD"](#)。

timeout

超时时间（秒）。

scds_pmf_signal () 函数返回以下值：

0 该函数成功。

非零值 该函数失败。

SCHA_ERR_TIMEOUT

发送信号后，进程树未在指定超时时间内退出。

SCHA_ERR_NOERR

该函数成功。

其他值

表明函数执行失败。有关失败代码的含义，请参见 [scha_calls\(3HA\) \[897\]](#)。

/usr/cluster/include/rgm/libdsdev.h

头文件

/usr/cluster/lib/libdsdev.so

库

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[pmfadm\(1M\) \[625\]](#), [scds_initialize\(3HA\) \[847\]](#), [scha_calls\(3HA\) \[897\]](#), [Unresolved link to "signal3HEAD"](#), [Unresolved link to "attributes5"](#)

名称

scds_pmf_start, scds_pmf_start_env — 执行 PMF 控制下的程序

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l dsdev#include <rgm/libdsdev.h>scha_err_t scds_pmf_start(
    scds_handle_t handle, scds_pmf_type_t program_type, int instance,
    const char *command, int child_monitor_level)scha_err_t
scds_pmf_start_env(scds_handle_t handle, scds_pmf_type_t
    program_type, int instance, const char *command, int
    child_monitor_level, char ** env)
```

scds_pmf_start () 函数可执行由 command 指定的受 PMF 控制的程序。此函数等同于带有 -c 选项的 pmfadm(1M) [625] 命令。

command 参数包含命令行和传递给函数的命令行参数。

使用 scds_pmf_start () 启动 PMF 下的数据服务应用程序、监视器或其他进程 (程序类型 SCDS_PMF_TYPE_SVC、SCDS_PMF_TYPE_MON 或 SCDS_PMF_TYPE_OTHER) 时, 通过使用 child_monitor_level 参数选择要监视的子进程级别。child_monitor_level 参数指定监视 child_monitor_level 级别以上或此级别的子进程。在级别 0 执行原始进程, 在级别 1 执行其子进程, 在级别 2 执行其子进程的子进程, 依此类推。任何新派生操作都会产生新的子进程级别。指定 -1 以监视所有级别的子进程。

例如, 如果要启动的命令是一个守护进程, 则其相应的 child_monitor_level 为 0。如果要启动的命令是一个启动守护进程的脚本, 则 child_monitor_level 的相应值为 1。

如果已运行底层应用程序进程, 则 scds_pmf_start () 将显示 syslog () 错误并返回 SCHA_ERR_INTERNAL, 因为 RGM 保证对节点上的 START 函数的两个调用必需具有介入的 STOP 函数。

支持以下参数:

handle	从 scds_initialize(3HA) [847] 返回的句柄
program_type	要执行的程序类型。有效类型为:
	SCDS_PMF_TYPE_SVC 数据服务应用程序
	SCDS_PMF_TYPE_MON 故障监视器
	SCDS_PMF_TYPE_OTHER 其他
instance	对于具有多个实例的资源, 该从 0 开始的整数可以唯一标识实例。对于单实例资源, 请使用 0。

`command` 要在 PMF 控制下执行的命令（包括命令行参数）。

`child_monitor_level` 指定要监视的子进程的级别（等同于 `pmfadm` 的 `-C` 选项）。使用 `-1` 指定所有级别的子进程。

`env` 指定指向环境字符串的字符指针数组，[Unresolved link to "execve2"](#) 手册页对此进行了说明。执行 `command` 参数指定的程序时，此环境将传递到此程序。

`scds_pmf_start ()` 函数返回以下值：

0 该函数成功。

非零值 该函数失败。

`SCHA_ERR_INTERNAL` 已运行底层应用程序进程。

`SCHA_ERR_NOERR` 该函数成功。

其他值 该函数失败。有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

`/usr/cluster/include/rgm/libdsdev.h`

头文件

`/usr/cluster/lib/libdsdev.so`

库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving（发展中）

[pmfadm\(1M\) \[625\]](#)、[scds_initialize\(3HA\) \[847\]](#)、[scds_pmf_stop\(3HA\) \[861\]](#)、[scds_svc_wait\(3HA\) \[887\]](#)、[link to "execve2"](#)、[Unresolved link to "attributes5"](#)

名称

scds_pmf_start, scds_pmf_start_env — 执行 PMF 控制下的程序

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l dsdev#include <rgm/libdsdev.h>scha_err_t scds_pmf_start(
    scds_handle_t handle, scds_pmf_type_t program_type, int instance,
    const char *command, int child_monitor_level)scha_err_t
scds_pmf_start_env(scds_handle_t handle, scds_pmf_type_t
    program_type, int instance, const char *command, int
    child_monitor_level, char ** env)
```

scds_pmf_start () 函数可执行由 command 指定的受 PMF 控制的程序。此函数等同于带有 -c 选项的 [pmfadm\(1M\) \[625\]](#) 命令。

command 参数包含命令行和传递给函数的命令行参数。

使用 scds_pmf_start () 启动 PMF 下的数据服务应用程序、监视器或其他进程（程序类型 SCDS_PMF_TYPE_SVC、SCDS_PMF_TYPE_MON 或 SCDS_PMF_TYPE_OTHER）时，通过使用 child_monitor_level 参数选择要监视的子进程级别。child_monitor_level 参数指定监视 child_monitor_level 级别以上或此级别的子进程。在级别 0 执行原始进程，在级别 1 执行其子进程，在级别 2 执行其子进程的子进程，依此类推。任何新派生操作都会产生新的子进程级别。指定 -1 以监视所有级别的子进程。

例如，如果要启动的命令是一个守护进程，则其相应的 child_monitor_level 为 0。如果要启动的命令是一个启动守护进程的脚本，则 child_monitor_level 的相应值为 1。

如果已运行底层应用程序进程，则 scds_pmf_start () 将显示 syslog () 错误并返回 SCHA_ERR_INTERNAL，因为 RGM 保证对节点上的 START 函数的两个调用必需具有介入的 STOP 函数。

支持以下参数：

handle	从 scds_initialize(3HA) [847] 返回的句柄						
program_type	要执行的程序类型。有效类型为： <table><tr><td>SCDS_PMF_TYPE_SVC</td><td>数据服务应用程序</td></tr><tr><td>SCDS_PMF_TYPE_MON</td><td>故障监视器</td></tr><tr><td>SCDS_PMF_TYPE_OTHER</td><td>其他</td></tr></table>	SCDS_PMF_TYPE_SVC	数据服务应用程序	SCDS_PMF_TYPE_MON	故障监视器	SCDS_PMF_TYPE_OTHER	其他
SCDS_PMF_TYPE_SVC	数据服务应用程序						
SCDS_PMF_TYPE_MON	故障监视器						
SCDS_PMF_TYPE_OTHER	其他						
instance	对于具有多个实例的资源，该从 0 开始的整数可以唯一标识实例。对于单实例资源，请使用 0。						

`command` 要在 PMF 控制下执行的命令（包括命令行参数）。

`child_monitor_level` 指定要监视的子进程的级别（等同于 `pmfadm` 的 `-C` 选项）。使用 `-1` 指定所有级别的子进程。

`env` 指定指向环境字符串的字符指针数组，[Unresolved link to "execve2"](#) 手册页对此进行了说明。执行 `command` 参数指定的程序时，此环境将传递到此程序。

`scds_pmf_start()` 函数返回以下值：

0 该函数成功。

非零值 该函数失败。

`SCHA_ERR_INTERNAL` 已运行底层应用程序进程。

`SCHA_ERR_NOERR` 该函数成功。

其他值 该函数失败。有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

`/usr/cluster/include/rgm/libdsdev.h`

头文件

`/usr/cluster/lib/libdsdev.so`

库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving（发展中）

[pmfadm\(1M\) \[625\]](#)、[scds_initialize\(3HA\) \[847\]](#)、[scds_pmf_stop\(3HA\) \[861\]](#)、[scds_svc_wait\(3HA\) \[887\]](#)、[link to "execve2"](#)、[Unresolved link to "attributes5"](#)

名称

scds_pmf_stop — 终止在 PMF 控制下运行的进程

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l dsdev#include <rgm/libdsdev.h>
scha_err_t scds_pmf_stop(scds_handle_t handle, scds_pmf_type_t
program_type, int instance, int signal, time_t timeout);
```

scds_pmf_stop () 函数停止在 PMF 控制下运行的程序。它等同于带有 -s 选项的 [pmfadm\(1M\) \[625\]](#) 命令。

如果请求的实例未运行，则 scds_pmf_stop () 返回值 SCHA_ERR_NOERR。

如果请求的实例正在运行，那么指定的信号将发送到该实例。如果实例无法在 80% 的超时值的时间段内终止，将向该实例发送 SIGKILL。之后，如果实例无法在 15% 的超时值的时间段内终止，则该函数将被视为已失败并返回 SCHA_ERR_TIMEOUT。剩余 5% 的超时参数假定已由此函数的开销所用。

支持以下参数：

handle

从 [scds_initialize\(3HA\) \[847\]](#) 返回的句柄

program_type

要执行的程序类型。有效类型为：

SCDS_PMF_TYPE_SVC	数据服务应用程序
SCDS_PMF_TYPE_MON	故障监视器
SCDS_PMF_TYPE_OTHER	其他

instance

对于具有多个实例的资源，该从 0 开始的整数可以唯一标识实例。对于单实例资源，请使用 0。

signal

要发送以终止实例的 Solaris 信号。请参见 [Unresolved link to "signal3HEAD"](#)。如果指定的信号无法终止实例，则使用 SIGKILL。

timeout

超时时间（秒）。

scds_pmf_stop () 函数返回以下值：

0 该函数成功。

non-zero 该函数失败。

SCHA_ERR_TIMEOUT

函数超时。

SCHA_ERR_NOERR

该函数成功。

其他值

表明函数执行失败。有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

/usr/cluster/include/rgm/libdsdev.h

头文件

/usr/cluster/lib/libdsdev.so

库

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[pmfadm\(1M\) \[625\]](#), [scds_initialize\(3HA\) \[847\]](#), [scds_pmf_start\(3HA\) \[857\]](#), [scha_calls\(3HA\) \[897\]](#), [Unresolved link to "signal3HEAD"](#), [Unresolved link to "attributes5"](#)

名称

scds_pmf_stop_monitoring — 停止监视在 PMF 控制下运行的进程

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l dsdev#include <rgm/libdsdev.h>
scha_err_t scds_pmf_stop_monitoring(scds_handle_t handle,
scds_pmf_type_t program_type, int instance);
```

scds_pmf_stop_monitoring () 函数停止监视在 PMF 控制下运行的进程树。PMF 不会发送信号以停止进程。而且，PMF 以后不会尝试重新启动进程。

如果请求的进程不受 PMF 控制，则 scds_pmf_stop_monitoring () 返回值 SCHA_ERR_NOERR。

支持以下参数：

handle

从 [scds_initialize\(3HA\) \[847\]](#) 返回的句柄

program_type

要执行的程序类型。有效类型为：

SCDS_PMF_TYPE_SVC	数据服务应用程序
SCDS_PMF_TYPE_MON	故障监视器
SCDS_PMF_TYPE_OTHER	其他

instance

对于具有多个实例的资源，该从 0 开始的整数可以唯一标识实例。对于单实例资源，请使用 0。

scds_pmf_stop_monitoring () 函数返回以下值：

0 该函数成功。

非零值 该函数失败。

SCHA_ERR_NOERR

该函数成功。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

`/usr/cluster/include/rgm/libdsdev.h`

头文件

`/usr/cluster/lib/libdsdev.so`

库

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[pmfadm\(1M\) \[625\]](#), [scds_initialize\(3HA\) \[847\]](#), [scds_pmf_start\(3HA\) \[857\]](#),
[scds_pmf_stop\(3HA\) \[861\]](#), [scha_calls\(3HA\) \[897\]](#), [Unresolved link to "attributes5"](#)

名称

scds_print_net_list — 显示网络资源列表的内容

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l dsdev#include <rgm/libdsdev.h>
void scds_print_net_list(scds_handle_t handle,int
debug_level,constscds_net_resource_list_t *netresource_list);
```

scds_print_net_list () 函数在由 debug_level 指定的调试级别将 netresource_list 指向的网络资源列表的内容写入系统日志。如果指定的调试级别大于当前正在使用的调试级别，则不会写入信息。

支持以下参数：

handle

从 [scds_initialize\(3HA\) \[847\]](#) 返回的句柄

debug_level

写入数据的调试级别

netresource_list

使用 [scds_get_rg_hostnames\(3HA\) \[835\]](#) 或 [scds_get_rs_hostnames\(3HA\) \[839\]](#) 检索到的初始化网络资源列表的指针

/usr/cluster/include/rgm/libdsdev.h

头文件

/usr/cluster/lib/libdsdev.so

库

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scds_get_rg_hostnames\(3HA\) \[835\]](#), [scds_get_rs_hostnames\(3HA\) \[839\]](#),
[scds_initialize\(3HA\) \[847\]](#), [scds_syslog_debug\(3HA\) \[893\]](#), [Unresolved link to "attributes5"](#)

名称

scds_print_netaddr_list — 显示资源组使用的主机名-端口-协议三元组列表的内容

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l dsdev#include <rgm/libdsdev.h>
void scds_print_netaddr_list(scds_handle_t handle, int
debug_level, constscds_netaddr_list_t *netaddr_list);
```

scds_print_netaddr_list () 函数在由 debug_level 指定的调试级别将 netaddr_list 指向的主机名-端口-协议三元组列表的内容写入系统日志。如果指定的调试级别大于当前正在使用的调试级别，则不会写入信息。

支持以下参数：

handle

从 [scds_initialize\(3HA\) \[847\]](#) 返回的句柄

debug_level

写入数据的调试级别

netaddr_list

使用 [scds_get_netaddr_list\(3HA\) \[825\]](#) 检索到的、指向资源组使用的主机名-端口-协议三元组列表的指针

/usr/cluster/include/rgm/libdsdev.h

头文件

/usr/cluster/lib/libdsdev.so

库

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scds_get_netaddr_list\(3HA\) \[825\]](#), [scds_initialize\(3HA\) \[847\]](#),
[scds_syslog_debug\(3HA\) \[893\]](#), [Unresolved link to "attributes5"](#)

名称

scds_print_port_list — 显示端口列表的内容

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l dsdev#include <rgm/libdsdev.h>
void scds_print_port_list(scds_handle_t handle,int debug_level,
constscds_port_list_t *port_list);
```

scds_print_port_list () 函数在由 debug_level 指定的调试级别将 port_list 指向的端口列表的内容写入系统日志。如果指定的调试级别大于当前正在使用的调试级别，则不会写入信息。

支持以下参数：

handle

从 [scds_initialize\(3HA\) \[847\]](#) 返回的句柄

debug_level

写入数据的调试级别

port_list

使用 [scds_get_port_list \(\)](#) 检索到的、指向资源组使用的端口-协议对列表的指针。

/usr/cluster/include/rgm/libdsdev.h

头文件

/usr/cluster/lib/libdsdev.so

库

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scds_get_port_list\(3HA\) \[827\]](#), [scds_initialize\(3HA\) \[847\]](#),
[scds_syslog_debug\(3HA\) \[893\]](#), [Unresolved link to "attributes5"](#)

名称

`scds_property_functions` — 一组用于检索常用资源属性、资源组属性、资源类型属性和扩展属性的值的便利函数

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l dsdev#include <rgm/libdsdev.h>
return-value-type scds-get-property-name(scds_handle_t handle);
```

数据服务开发库 (Data Service Development Library, DSDL) 提供一组用于检索常用资源属性、资源组属性、资源类型属性和扩展属性的值的便利函数。使用 [scds_get_ext_property\(3HA\) \[817\]](#) 检索用户定义的扩展属性。

所有便利函数均使用以下约定：

- 这些函数仅使用 `handle` 参数。要传递到属性检索函数的 `handle` 参数将通过上一次调用 [scds_initialize\(3HA\) \[847\]](#) 返回。
- 每个函数对应于特定属性。
- 函数的返回值类型与该函数检索到的属性值的类型相匹配。
- 这些函数不会返回错误，因为返回值已在 [scds_initialize\(3HA\) \[847\]](#) 中预计算。对于返回指针的函数，在遇到错误情形时将返回 `NULL` 值，例如，先前未调用 `scds_initialize ()` 的情形。
- 如果属性的新值已在传递到调用程序的命令行参数中指定 (`argv[]`)，将返回此新值（在实施验证方法的情况下）。通过此方法，您可以在实际设置未来的新属性值之前先验证它们。否则，这些函数将返回从 RGM 中检索到的值。
- 其中一些便利函数返回指向属于 DSDL 的内存的指针。请勿修改此内存。调用 [scds_close\(3HA\) \[777\]](#) 会使此指针失效。

有关标准属性的描述，请参见 [r_properties\(5\) \[1103\]](#)、[rg_properties\(5\) \[1129\]](#) 和 [rt_properties\(5\) \[1143\]](#) 手册页。有关扩展属性的描述，请参见各个数据服务手册页。

有关诸如

`scha_prop_type_t`、`scha_extprop_value_t`、`scha_initnodes_flag_t`、`scha_str_array_t`、`scha_failover_m` 和 `scha_rsstatus_t` 等这些函数使用的属性类型的信息，请参见 [scha_calls\(3HA\) \[897\]](#) 手册页和 `scha_types.h` 头文件。

这些函数使用以下命名约定：

资源属性

`scds_get_rs_property-name`

资源组属性

`scds_get_rg_property-name`

资源类型属性

`scds_get_rt_property-name`

常用扩展属性

`scds_get_ext_property-name`

注 - 属性名称不区分大小写。指定属性名称时，可以使用大小写字母的任意组合。

特定于资源的函数

函数将返回特定资源属性的值。某些属性的值在 RTR 文件中或由 `clresource(1CL) [225]` 命令明确设置。其他的由 RGM 动态确定。函数将返回与请求的属性相对应的数据类型。

以下每个资源依赖性查询函数都具有相应的“Q”或“合格”版本：

`scds_get_rs_resource_dependencies`

`scds_get_rs_resource_dependencies_Q`

`scds_get_rs_resource_dependencies_offline_restart`

`scds_get_rs_resource_dependencies_Q_offline_restart`

`scds_get_rs_resource_dependencies_restart`

`scds_get_rs_resource_dependencies_Q_restart`

`scds_get_rs_resource_dependencies_weak`

`scds_get_rs_resource_dependencies_Q_weak`

合格版本将返回为每个资源依赖性声明的范围或限定符（如果有）。`r_properties(5) [1103]` 手册页中介绍了 `{LOCAL_NODE}`、`{ANY_NODE}` 和 `{FROM_RG_AFFINITIES}` 限定符。

Cheap_probe_interval

`int scds_get_rs_cheap_probe_interval(scds_handle_t handle)`

Failover_mode

`scha_failover_mode_t scds_get_rs_failover_mode(scds_handle_t handle)`

Monitor_stop_timeout

`int scds_get_rs_monitor_stop_timeout(scds_handle_t handle)`

Monitored_switch

`scha_switch_t scds_get_rs_monitored_switch(scds_handle_t handle)`

```
Network_resources_used
    scha_str_array_t * scds_get_rs_network_resources_used(scds_handle_t handle)

On_off_switch
    scha_switch_t scds_get_rs_on_off_switch(scds_handle_t handle)

Resource_dependencies
    const scha_str_array_t * scds_get_rs_resource_dependencies(scds_handle_t
    handle)

Resource_dependencies_Q (合格)
    const scha_str_array_t * scds_get_rs_resource_dependencies_Q(scds_handle_t
    handle)

Resource_dependencies_offline_restart
    const scha_str_array_t *
    scds_get_rs_resource_dependencies_offline_restart(scds_handle_t handle)

Resource_dependencies_Q_offline_restart (合格)
    const scha_str_array_t *
    scds_get_rs_resource_dependencies_Q_offline_restart(scds_handle_t handle)

Resource_dependencies_restart
    const scha_str_array_t *
    scds_get_rs_resource_dependencies_restart(scds_handle_t handle)

Resource_dependencies_Q_restart (合格)
    const scha_str_array_t *
    scds_get_rs_resource_dependencies_Q_restart(scds_handle_t handle)

Resource_dependencies_weak
    const scha_str_array_t *
    scds_get_rs_resource_dependencies_weak(scds_handle_t handle)

Resource_dependencies_Q_weak (合格)
    const scha_str_array_t *
    scds_get_rs_resource_dependencies_Q_weak(scds_handle_t handle)

Resource_project_name
    const char * scds_get_rs_resource_project_name(scds_handle_t handle)

Retry_count
    int scds_get_rs_retry_count(scds_handle_t handle)
```

```
Retry_interval
    int scds_get_rs_retry_interval(scds_handle_t handle)

Scalable
    boolean scds_get_rs_scalable(scds_handle_t handle)

Start_timeout
    int scds_get_rs_start_timeout(scds_handle_t handle)

Stop_timeout
    int scds_get_rs_stop_timeout(scds_handle_t handle)

Thorough_probe_interval
    int scds_get_rs_thorough_probe_interval(scds_handle_t handle)
```

特定于资源组的函数

函数将返回特定资源组属性的值。某些属性的值由 `clresourcegroup(1CL) [253]` 命令明确设置。其他的由 RGM 动态确定。函数将返回与请求的属性相对应的数据类型。

```
Desired_primaries
    int scds_get_rg_desired_primaries(scds_handle_t handle)

Global_resources_used
    const scha_str_array_t * scds_get_rg_global_resources_used(scds_handle_t
    handle)

Implicit_network_dependencies
    boolean_t scds_get_rg_implicit_network_dependencies(scds_handle_t handle)

Maximum_primaries
    int scds_get_rg_maximum_primaries(scds_handle_t handle)

Nodelist
    const scha_str_array_t * scds_get_rg_nodelist (scds_handle_t handle)

Pathprefix
    const char * scds_get_rg_pathprefix(scds_handle_t handle)

Pingpong_interval
    int scds_get_rg_pingpong_interval(scds_handle_t handle)

Resource_list
    const scha_str_array_t * scds_get_rg_resource_list(scds_handle_t handle)
```

```
RG_affinities
    const scha_str_array_t * scds_get_rg_rg_affinities(scds_handle_t handle)

RG_mode
    scha_rgmode_t scds_get_rg_rg_mode(scds_handle_t handle)

RG_project_name
    const char * scds_get_rg_rg_project_name(scds_handle_t handle)

RG_slm_cpu_shares
    int scds_get_rg_rg_slm_cpu_shares(scds_handle_t handle)

RG_slm_pset_min
    int scds_get_rg_rg_slm_pset_min(scds_handle_t handle)

RG_slm_pset_type
    const char * scds_get_rg_rg_slm_pset_type(scds_handle_t handle)

RG_slm_type
    const char * scds_get_rg_rg_slm_type(scds_handle_t handle)
```

特定于资源类型的函数

函数将返回特定资源类型属性的值。某些属性的值在 RTR 文件中或由 [clresourcetype\(1CL\) \[277\]](#) 命令明确设置。其他的由 RGM 动态确定。函数将返回与请求的属性相对应的数据类型。

```
API_version
    int scds_get_rt_api_version(scds_handle_t handle)

Failover
    boolean_t scds_get_rt_failover(scds_handle_t handle)

Init_nodes
    scha_initnodes_flag_t scds_get_rt_init_nodes(scds_handle_t handle)

Installed_nodes
    const scha_str_array_t * scds_get_rt_installed_nodes(scds_handle_t handle)

RT_basedir
    const char * scds_get_rt_rt_basedir(scds_handle_t handle)

RT_version
    const char * scds_get_rt_rt_version(scds_handle_t handle)
```

Single_instance

```
boolean_t scds_get_rt_single_instance(scds_handle_t handle)
```

Start_method

```
const char * scds_get_rt_start_method(scds_handle_t handle)
```

Stop_method

```
const char * scds_get_rt_stop_method(scds_handle_t handle)
```

特定于扩展属性的函数

函数将返回特定资源扩展属性的值。属性的值在 RTR 文件中或由 [clresource\(1CL\) \[225\]](#) 命令明确设置。函数将返回与请求的属性相对应的数据类型。

资源类型可以定义除此列出的四个属性之外的扩展属性，但这四个属性具有为其定义的便利函数。可使用这些便利函数或使用 [scds_get_ext_property\(3HA\) \[817\]](#) 函数检索这些属性。必须使用 `scds_get_ext_property()` 才能检索除这四个属性之外的扩展属性。

Confdir_list

```
scha_str_array_t * scds_get_ext_confdir_list(scds_handle_t handle)
```

Monitor_retry_count

```
int scds_get_ext_monitor_retry_count(scds_handle_t handle)
```

Monitor_retry_interval

```
int scds_get_ext_monitor_retry_interval(scds_handle_t handle)
```

Probe_timeout

```
int scds_get_ext_probe_timeout(scds_handle_t handle)
```

所有便利函数都支持以下参数：

handle 从 [scds_initialize\(3HA\) \[847\]](#) 返回的句柄。

函数的返回值类型与该函数检索到的属性值的类型相匹配。

这些函数不会返回错误，因为返回值已在 [scds_initialize\(3HA\) \[847\]](#) 中预计算。对于返回指针的函数，在遇到错误情形时将返回 NULL 值，例如，先前未调用 `scds_initialize()` 的情形。

```
/usr/cluster/include/rgm/libdsdev.h
```

头文件

/usr/cluster/lib/libdsdev.so

库

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#) :

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[clresource\(1CL\) \[225\]](#)、[clresourcegroup\(1CL\) \[253\]](#)、[clresourcetype\(1CL\) \[277\]](#)、[scds_close\(3HA\) \[777\]](#)
[link to "attributes5"](#)、[r_properties\(5\) \[1103\]](#)、[rg_properties\(5\) \[1129\]](#) 和
[rt_properties\(5\) \[1143\]](#)

名称

scds_restart_resource — 重新启动资源

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l dsdev#include <rgm/libdsdev.h>
scha_err_t scds_restart_resource(scds_handle_t handle);
```

scds_restart_resource () 函数可调用带有 SCHA_RESOURCE_RESTART *tag* 参数的 [scha_control\(3HA\) \[939\]](#) 函数以请求重新启动资源。从故障监视器调用此函数。

支持以下参数：

handle 从 [scds_initialize\(3HA\) \[847\]](#) 返回的句柄

scha_restart_resource () 函数返回以下值：

0 该函数成功。

非零值 该函数失败。

SCHA_ERR_NOERR 函数成功。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

/usr/cluster/include/rgm/libdsdev.h

头文件

/usr/cluster/lib/libdsdev.so

库

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[rt_callbacks\(1HA\) \[545\]](#)、[scds_restart_rg\(3HA\) \[881\]](#)、[scha_calls\(3HA\) \[897\]](#)、[scha_control\(3HA\) \[939\]](#)
[link to "attributes5"](#)

名称

scds_restart_rg — 重新启动资源组

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib -l dsdev#include <rgm/libdsdev.h>
    scha_err_t scds_restart_rg(scds_handle_t handle);
```

scds_restart_rg () 函数在传递给调用程序的资源所属的资源组上执行 [scha_control\(3HA\) \[939\]](#) SCHA_RESTART 操作。从故障监视器调用此函数。

此函数执行成功后不会返回。因此，将此函数作为调用程序中最后执行的代码段。

支持以下参数：

handle 从 [scds_initialize\(3HA\) \[847\]](#) 返回的句柄

scds_restart_rg () 函数返回以下值：

0 该函数成功。

非零值 该函数失败。

SCHA_ERR_NOERR 函数成功。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

/usr/cluster/include/rgm/libdsdev.h

头文件

/usr/cluster/lib/libdsdev.so

库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scha_calls\(3HA\) \[897\]](#)、[scha_control\(3HA\) \[939\]](#)、[scds_initialize\(3HA\) \[847\]](#)、[scds_restart_resource\(3HA\) \[847\]](#)、[Unresolved link to "attributes5"](#)

名称

`scds_simple_net_probe` — 通过建立和终止与应用程序的 TCP 连接来进行探测

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib -l dsdev#include <rgm/  
libdsdev.h> scha_err_t scds_simple_net_probe(scds_handle_t handle, scds_netaddr_t addr, time_t timeout, scds_fmsock  
*status, int count);
```

`scds_simple_net_probe()` 函数是 [scds_fm_net_connect\(3HA\) \[789\]](#) 和 [scds_fm_net_disconnect\(3HA\) \[793\]](#) 的包装函数。对于具有多个映射的主机，`scds_simple_net_probe()` 可处理提供的 `hostname` 的 IPv4 地址和 IPv6 地址。

您可以通过使用 [scds_get_netaddr_list\(3HA\) \[825\]](#) 检索资源的网络地址列表。

连接到 IPv4 目标的状态或与 IPv4 目标断开连接的状态将存储在 `scds_fmsock_status_t` 数组的第一个成员中。第二个成员包含 IPv6 目标的状态。如果提供给此函数的 `hostname` 不包含 IPv4 或 IPv6 映射，则相应的状态将设置为 `SCDS_FMSOCK_NA`。

支持以下参数：

`handle`

由 [scds_initialize\(3HA\) \[847\]](#) 返回的句柄。

`addr`

指定进程侦听的 `hostname`、TCP 端口号和协议标识符。

`timeout`

等待成功连接的超时值（秒）。每个套接字（IPv4 或 IPv6）都将获得相同的超时时间，且这些超时将并行进行。

`status`

类型 `scds_fmsock_status_t` 的 `SCDS_MAX_IPADDR_TYPES` 成员的数组。数组中的每个成员都保存一个状态。此参数是由此函数设置的输出参数。

`count`

`socklist` 数组中成员的数量。将此参数设置为 `SCDS_MAX_IPADDR_TYPES`。

`scds_simple_net_probe()` 函数返回以下值：

0 该函数成功。

`SCHA_ERR_INVALID` 调用该函数时的参数无效。

其他非零值 由于超时、拒绝连接或其他错误，至少有一个连接操作失败。检查设置为 `SCDS_FMSOCK_ERR` 的 `socklist` 数组的所有成员的 `err` 字段以确定确切的错误。

非零值 至少有一个连接或断开连接操作失败。可以检查 `scds_fmsock_status_t` 数组以确定该失败是 IPv4 目标、IPv6 目标还是这两者中的失败。

`SCHA_ERR_NOERR`
表明函数执行成功。

`SCHA_ERR_INTERNAL`
表明执行函数时发生内部错误。

`SCHA_ERR_STATE`
表明连接请求被服务器拒绝。

`SCHA_ERR_TIMEOUT`
表明连接请求超时。

`/usr/cluster/include/rgm/libdsdev.h`
头文件

`/usr/cluster/lib/libdsdev.so`
库

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scds_fm_net_connect\(3HA\) \[789\]](#)、[scds_fm_net_disconnect\(3HA\) \[793\]](#)、[scds_get_netaddr_list\(3HA\) \[800\]](#)
[link to "attributes5"](#)

名称

scds_simple_probe — 通过建立和终止与应用程序的 TCP 连接来进行探测

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l dsdev#include <rgm/libdsdev.h>
scha_err_t scds_simple_probe(scds_handle_t handle, const char *hostname,
int port, time_t timeout);
```

scds_simple_probe () 函数是要在超时情况下运行的、位于 [Unresolved link to "connect3SOCKET"](#) 和 [Unresolved link to "close2"](#) 周围的包装函数。

使用 [scds_get_rg_hostnames\(3HA\) \[835\]](#) 或 [scds_get_rs_hostnames\(3HA\) \[839\]](#) 检索 hostname。

考虑使用 [scds_simple_net_probe\(3HA\) \[883\]](#) 代替此函数。

支持以下参数：

handle

由 [scds_initialize\(3HA\) \[847\]](#) 返回的句柄。

hostname

要连接到的计算机的 Internet 主机名。

port

要与其连接的端口号。

timeout

以秒为单位的超时值（等待成功连接）。

scds_simple_probe () 函数返回以下值：

0 该函数成功。

非零值 该函数失败。

SCHA_ERR_NOERR

表明函数执行成功。

SCHA_ERR_TIMEOUT

表明函数超时。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

`/usr/cluster/include/rgm/libdsdev.h`

头文件

`/usr/cluster/lib/libdsdev.so`

库

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Deprecated (已过时)

[Unresolved link to "close2"](#), [Unresolved link to "connect3SOCKET"](#),
[scds_fm_net_connect\(3HA\) \[789\]](#), [scds_fm_net_disconnect\(3HA\) \[793\]](#),
[scds_get_rg_hostnames\(3HA\) \[835\]](#), [scds_get_rs_hostnames\(3HA\) \[839\]](#),
[scds_initialize\(3HA\) \[847\]](#), [scds_simple_net_probe\(3HA\) \[883\]](#),
[scha_calls\(3HA\) \[897\]](#), [Unresolved link to "attributes5"](#)

名称

scds_svc_wait — 等候指定的超时时间，以待受监视进程终止

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib -l dsdev#include <rgm/libdsdev.h>
    scha_err_t scds_svc_wait(scds_handle_t handle, time_t timeout);
```

scds_svc_wait () 函数可等候指定的超时时间，以待受监视进程组终止。它将针对调用 START 方法所传递的资源来等待 [scds_pmf_start\(3HA\) \[857\]](#) 启动的所有进程组。scds_svc_wait () 函数使用资源的 Retry_interval 和 Retry_count 属性限制要等待的进程终止数。如果 Retry_interval 期间的进程终止数达到 Retry_count 值，scds_svc_wait () 将返回 SCHA_ERR_FAIL。

如果进程失败数低于 Retry_count 的值，则将重新启动该进程，scds_svc_wait () 将等候全部超时时间，以待更多进程终止。进程失败的计数包含对 scds_svc_wait () 的连续调用。

支持以下参数：

handle	从 scds_initialize(3HA) [847] 返回的句柄
timeout	超时时间 (秒)

scds_svc_wait () 函数返回以下值：

0 该函数成功。

非零值 该函数失败。

SCHA_ERR_TIMEOUT 函数超时。

SCHA_ERR_NOERR 未发生进程终止，或已成功重新启动进程。

SCHA_ERR_FAIL 失败数达到 Retry_count 属性的值。

SCHA_ERR_STATE 出现系统错误或其他意外错误。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

例 366 在 START 方法中使用 scds_svc_wait ()

以下示例显示如何在 START 方法中使用 scds_svc_wait 以便在服务无法启动时提早返回。使用 scds_pmf_start () 启动应用程序进程后，START 方法必须等待应用程序完成

自身的全部初始化并变得可用，然后才能返回成功。如果应用程序无法启动，START 方法必须等待整个 Start_timeout 时间，然后才能返回失败。通过使用如以下示例中的 scds_svc_wait ()，允许 START 方法最多重新启动应用程序 Retry_count 次，且在服务无法启动时从 START 方法中提早返回失败。

```

/*
 * scds_svc_wait is a subroutine in a START method to
 * check that the service is fully available before returning.
 * Calls svc_probe() to check service availability.
 */
int
svc_wait(scds_handle_t handle)
{
    while (1) {
        /* Wait for 5 seconds */
        if (scds_svc_wait(handle, 5) != SCHA_ERR_NOERR) {
            scds_syslog(LOG_ERR, "Service failed to start.");
            return (1); /* Start Failure */
        }
        /* Check if service is fully up every 5 seconds */
        if (svc_probe(handle) == 0) {
            scds_syslog(LOG_INFO, "Service started successfully.");
            return (0);
        }
    }
    return (0);
}

```

/usr/cluster/include/rgm/libdsdev.h

头文件

/usr/cluster/lib/libdsdev.so

库

有关以下属性的描述，请参见 [Unresolved link to " attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scds_initialize\(3HA\) \[847\]](#)、[scds_pmf_start\(3HA\) \[857\]](#)、[scha_calls\(3HA\) \[897\]](#)、[Unresolved link to " attributes5"](#)、[r_properties\(5\) \[1103\]](#)

- 如果 START 方法超过资源上的 Start_timeout 设置，资源组管理器 (Resource Group Manager, RGM) 将终止 START 方法，即使 START 方法当前正在等待 scds_svc_wait () 返回。

-
- 如果资源上的 `Retry_interval` 大于 `Start_timeout`，则 RGM 可能会使 `START` 方法超时，即使失败数低于 `Retry_count`。
 - 如果 `START` 方法通过对 `scds_pmf_start()` 进行多次调用来启动多个进程组，则 `scds_svc_wait()` 将在它们结束时启动进程组。它不会在进程组之间强制任何依赖性。如果进程组之间存在依赖性（一个进程组失败时需要重新启动其他进程组），请不要使用 `scds_svc_wait()`。请改用 `sleep()` 以在进程组的运行状况检查之间进行等待。

名称

scds_syslog — 将消息写入系统日志

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l dsdev#include <rgm/libdsdev.h>void scds_syslog(int priority,
const char*format...
```

scds_syslog () 函数将消息写入系统日志。它使用由 [scha_cluster_getlogfacility\(3HA\) \[921\]](#) 函数返回的设备。您可以将这些消息转发给相应的日志文件和用户。有关更多信息，请参见 [Unresolved link to "syslog.conf4"](#)。

所有系统日志消息都带有以下前缀：

```
SC[<resourceTypeName>,<resourceGroupName>,<resourceName>,<methodName>
```



注意 - 写入系统日志的消息未国际化。不要与此函数一起使用 `gettext()` 或其他消息转换函数。

支持以下参数：

priority	消息优先级，如 Unresolved link to "syslog3C" 所指定
format	消息格式字符串，如 Unresolved link to "printf3C" 所指定
...	变量，由 format 参数指示，如 <code>printf ()</code> 所指定

/usr/cluster/include/rgm/libdsdev.h

头文件

/usr/cluster/lib/libdsdev.so

库

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[Unresolved link to " printf3C"](#), [scds_syslog_debug\(3HA\) \[893\]](#),
[scha_cluster_getlogfacility\(3HA \) \[921\]](#), [Unresolved link to " syslog3C"](#),
[Unresolved link to " syslog.conf4"](#), [Unresolved link to " attributes5"](#)

名称

scds_syslog_debug — 将调试消息写入系统日志

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l dsdev#include <rgm/libdsdev.h>void scds_syslog_debug(int
debug_level, constchar *format...
```

scds_syslog_debug () 函数将调试消息写入系统日志。它使用由 [scha_cluster_getlogfacility\(3HA\) \[921\]](#) 函数返回的设备。

所有系统日志消息都带有以下前缀：

```
SC[<resourceTypeName>,<resourceGroupName>,<resourceName>,<methodName>
```

如果指定比当前所用调试级别更高的 debug_level，则不会写入任何信息。

DSDL 将最大调试级别 SCDS_MAX_DEBUG_LEVEL 定义为 9。调用程序必须在 scds_syslog_debug () 之前进行调用的 [scds_initialize\(3HA\) \[847\]](#) 函数将从以下文件中检索当前调试级别：/var/cluster/rgm/rt/<resourceTypeName>/loglevel。



注意 - 写入系统日志的消息未国际化。不要与此函数一起使用 gettext () 或其他消息转换函数。

支持以下参数：

debug_level	写入此消息的调试级别。有效的调试级别介于 1 到 SCDS_MAX_DEBUG_LEVEL 之间，此值由 DSDL 定义为 9。如果指定的调试级别大于调用程序设置的调试级别，则消息不会写入到系统日志。
format	消息格式字符串，如 Unresolved link to "printf3C" 所指定
...	变量，由 format 参数指示，如 Unresolved link to "printf3C" 所指定

例 367 显示所有调试消息

要查看资源类型 SUNW.iws 的所有调试消息，请在群集的所有节点上发出以下命令

```
echo 9 > /var/cluster/rgm/rt/SUNW.iws/loglevel
```

例 368 抑制调试消息

要抑制资源类型 `SUNW.iws` 的调试消息，请在群集的所有节点上发出以下命令

```
echo 0 > /var/cluster/rgm/rt/SUNW.iws/loglevel
```

```
/usr/cluster/include/rgm/libdsdev.h
```

头文件

```
/usr/cluster/lib/libdsdev.so
```

库

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[Unresolved link to "printf3C"](#), [scds_syslog\(3HA\) \[891\]](#),
[scha_cluster_getlogfacility\(3HA\) \[921\]](#), [Unresolved link to "syslog3C"](#),
[Unresolved link to "syslog.conf4"](#), [Unresolved link to "attributes5"](#)

名称

scds_timerun, scds_timerun_delay — 在给定时段内执行给定命令

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l dsdev#include <rgm/libdsdev.h>
scha_err_t scds_timerun(scds_handle_t handle, const char *command,
time_t timeout, int signal, int *cmd_exit_code);
scha_err_t scds_timerun_delay(scds_handle_t handle, const char *command,
time_t timeout, int signal, int *cmd_exit_code);
```

scds_timerun () 函数使用 hatimerun 执行指定命令。如果在 timeout 参数指定的分配时段内未完成该命令，scds_timerun () 将发送由 signal 参数指定的信号以终止该命令。

command 参数不支持 I/O 重定向。但是，您可以编写脚本以执行重定向，然后将此脚本加入 command 参数中，作为 scds_timerun () 需要执行的命令。

scds_timerun_delay() () 函数的行为与 scds_timerun () 基本相同，不同的是它在调用 hatimerun 命令时使用 -d (delay, 延迟) 命令行选项。此选项延迟启动超时时钟，直到命令已开始执行。在负载沉重的系统上，从子进程派生到指定的程序开始执行，这之间可能有几秒的延迟。使用 -d 选项避免了将额外的执行前时间计入所分配的超时时间。

支持以下参数：

handle

从 [scds_initialize\(3HA\) \[847\]](#) 返回的句柄

command

包含要运行的命令的字符串

timeout

分配用于运行命令的时间 (秒)

signal

用于终止命令的信号 (如果在超时到期时该命令仍在运行)。如果 signal = -1, 那么将使用 SIGKILL。请参见 [Unresolved link to "signal3HEAD"](#)。

cmd_exit_code

执行命令返回的代码

scds_timerun () 函数返回以下值：

0 该函数成功。

非零值 该函数失败。

SCHA_ERR_NOERR

已执行命令，且 `cmd_exit_code` 包含子程序的退出状态。

SCHA_ERR_INTERNAL

未发生超时，但其他某个错误已被 `scds_timerun()` 检测到，该错误不是子程序检测到的错误。或者 [hatimerun\(1M\) \[623\]](#) 捕获到信号 SIGTERM。

SCHA_ERR_INVALID

存在无效的输入参数。

SCHA_ERR_TIMEOUT

在由 `command` 参数指定的命令完成执行之前发生超时。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

`/usr/cluster/include/rgm/libdsdev.h`

头文件

`/usr/cluster/lib/libdsdev.so`

库

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[hatimerun\(1M\) \[623\]](#), [scds_initialize\(3HA\) \[847\]](#), [scha_calls\(3HA\) \[897\]](#), [Unresolved link to "signal3HEAD"](#), [Unresolved link to "attributes5"](#)

名称

scha_calls — 实现回调方法和监视资源类型过程中所用的 Oracle Solaris Cluster 库函数

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l scha#include <scha.h>scha_err_t scha_get_function(
    handle, const char *tag...);

scha_err_t scha_control(const char *tag...
```

Oracle Solaris Cluster 库函数

[scha_resource_get\(3HA\) \[963\]](#)、[scha_resourcetype_get\(3HA\) \[1041\]](#)、[scha_resourcegroup_get\(3HA\) \[1013\]](#) 和 [scha_resource_setstatus\(3HA\) \[1003\]](#) 提供了实现回调方法和监视资源类型过程中要使用的接口。资源类型表示群集的资源组管理器 (Resource Group Manager, RGM) 设备所控制的服务。

get 函数用于访问群集配置信息。所有这些函数都具有相同的通用签名。这些函数会将从上一调用返回的 *handle* 参数传给 *open* 函数。此 *handle* 指示群集配置中要访问的对象。*tag* 参数指示要访问的对象的属性。*tag* 的值用于确定是否需要其他参数以及最终的 *out* 参数（可用于返回请求的信息）的类型。您可以对同一句柄进行重复的 *get* 调用，直到遇到 *close* 调用为止，*close* 调用可使句柄无效并释放为 *get* 调用返回的值分配的内存。

如果在返回值时需要内存，系统会为每个 *get* 调用分配相应的内存。在一个调用中为返回值而分配的空间不会被后续调用覆盖和重新使用。

[scha_control\(3HA\) \[939\]](#) 函数还具有一个指示控制操作的 *tag* 参数，但不会在输出参数中返回信息。

[scha_resource_setstatus\(1HA\) \[593\]](#) 命令用于设置由 RGM 管理的资源的状态和 *Status_msg* 属性。

有关作为每个函数的 *tag* 参数值接受的宏值以及每个 *tag* 的变量参数类型，应参阅各个函数的手册页。输出参数的类型在下一节中进行说明。

对于 *scha* 函数，存在一组 *scha_err_tenum* 类型的返回值。退出代码的枚举符号、整数值和含义在“返回值”中进行说明。

[scha_strerror\(3HA\) \[1057\]](#) 函数可将 *scha* 函数返回的 *scha_err_t* 代码转换为相应的错误消息。

输出参数数据类型

uint_t

无符号整数类型。此类型在系统头文件 *sys/types.h* 中进行定义。

boolean_t

此类型在系统头文件 `sys/types.h` 中进行定义。

```
typedef enum { B_FALSE, B_TRUE } boolean_t;
```

scha_switch_t

enum 类型，用于指示 `On_Off_switch` 或 `Monitored_switch` 资源属性值。

```
typedef enum scha_switch {  
    SCHA_SWITCH_DISABLED = 0,  
    SCHA_SWITCH_ENABLED  
} scha_switch_t;
```

scha_rg_preemption_mode_t

enum 类型，用于指示 `Preemption_mode` 资源组属性的值。

```
typedef enum scha_rg_preemption_mode {  
    SCHA_HAS_PREEMPTION_COST = 0,  
    SCHA_NO_PREEMPTION_COST,  
    SCHA_NEVER_PREEMPT_RG  
} scha_rg_preemption_mode_t;
```

scha_rsstate_t

enum 类型，用于指示资源状况。

```
typedef enum scha_rsstate {  
    SCHA_RSSTATE_ONLINE = 0,  
    SCHA_RSSTATE_OFFLINE,  
    SCHA_RSSTATE_START_FAILED,  
    SCHA_RSSTATE_STOP_FAILED,  
    SCHA_RSSTATE_MONITOR_FAILED,  
    SCHA_RSSTATE_ONLINE_NOT_MONITORED,  
    SCHA_RSSTATE_STARTING,  
    SCHA_RSSTATE_STOPPING  
} scha_rsstate_t;
```

scha_rgstate_t

enum 类型，用于指示资源组状况。

```
typedef enum scha_rgstate {  
    SCHA_RGSTATE_UNMANAGED = 0,  
    SCHA_RGSTATE_ONLINE,  
    SCHA_RGSTATE_OFFLINE,  
    SCHA_RGSTATE_PENDING_ONLINE,  
    SCHA_RGSTATE_PENDING_OFFLINE,  
    SCHA_RGSTATE_ERROR_STOP_FAILED,  
    SCHA_RGSTATE_ONLINE_FAULTED,  
    SCHA_RGSTATE_PENDING_ONLINE_BLOCKED  
} scha_rgstate_t;
```

scha_rgmode_t

enum 类型，用于指示资源组是处于故障转移模式还是可伸缩模式。

```
typedef enum scha_rgmode {
    RGMODE_NONE = 0,
    RGMODE_FAILOVER,
    RGMODE_SCALABLE
} scha_rgmode_t;
```

scha_failover_mode_t

enum 类型，用于指示 Failover_Mode 资源属性的值。

```
typedef enum scha_failover_mode {
    SCHA_FOMODE_NONE = 0,
    SCHA_FOMODE_HARD,
    SCHA_FOMODE_SOFT,
    SCHA_FOMODE_RESTART_ONLY,
    SCHA_FOMODE_LOG_ONLY
} scha_failover_mode_t;
```

scha_initnodes_flag_t

enum 类型，用于指示 Init_nodes 资源类型属性的值。

```
typedef enum scha_initnodes_flag {
    SCHA_INFLAG_RG_PRIMARYES = 0,
    SCHA_INFLAG_RT_INSTALLED_NODES
} scha_initnodes_flag_t;
```

scha_node_state_t

enum 类型，用于指示节点是处于打开状态还是关闭状态。

```
typedef enum scha_node_state {
    SCHA_NODE_UP = 0,
    SCHA_NODE_DOWN
} scha_node_state_t;
```

scha_str_array_t

用于保存字符串列表值的结构。

```
typedef struct scha_str_array {
    uint_t      array_cnt;
    boolean_t   is_ALL_value;
    char        **str_array;
} scha_str_array_t;
```

array_cnt

提供列表中的数字元素。

is_ALL_value

如果将属性设置为 "all" 值（也称为通配符或星号 (*) 字符），则 is_ALL_value 将设置为 B_TRUE 且 str_array 将为 NULL。因此，将忽略 str_array。

str_array

指向 array_cnt strings 数组的指针。

scha_uint_array_t

用于保存无符号整数列表值的结构。

```
typedef struct scha_uint_array {
    uint_t array_cnt;
    uint_t *int_array;
} scha_uint_array_t;
```

array_cnt 列表中元素的数目。

int_array 指向 array_cnt 无符号整数数组的指针。

scha_status_value_t

用于返回资源的状态和状态消息的结构。

```
typedef struct scha_status_value {
    scha_rsstatus_t status;
    char *status_msg;
} scha_status_value_t;
```

```
typedef enum scha_rsstatus {
    SCHA_RSSTATUS_ONLINE = 0,
    SCHA_RSSTATUS_OFFLINE,
    SCHA_RSSTATUS_FAULTED,
    SCHA_RSSTATUS_DEGRADED,
    SCHA_RSSTATUS_UNKNOWN
} scha_rsstatus_t;
```

status 保存一个 enum 值，该值指示由资源监视器设置的资源状态。

scha_extprop_value_t

用于返回扩展属性值的结构。

prop_type 结构成员用于指示扩展属性的类型并确定将哪个联合元素用于 prop_type 字段和返回值：

```
SCHA_PTYPE_STRING          val_str
SCHA_PTYPE_INT             val_int
SCHA_PTYPE_ENUM            val_enum
SCHA_PTYPE_BOOLEAN        val_boolean
SCHA_PTYPE_STRINGARRAY    val_strarray
```

```
typedef struct scha_extprop_value {
    scha_prop_type_t prop_type;
    union {
        char *val_str;
        int val_int;
        char *val_enum;
    };
};
```

```
        boolean_t      val_boolean;
        scha_str_array_t *val_strarray;
    } val;
} scha_extprop_value_t;
```

下面是 `scha_strerror(3HA)` [1057] 返回的 `scha_err_t` 错误号和错误代码的列表。

0	SCHA_ERR_NOERR	未找到错误。
1	SCHA_ERR_NOMEM	交换空间不足。
2	SCHA_ERR_HANDLE	资源管理句柄无效。
3	SCHA_ERR_INVALID	输入参数无效。
4	SCHA_ERR_TAG	API 标记无效。
5	SCHA_ERR_RECONF	正在重新配置群集。
6	SCHA_ERR_ACCESS	权限被拒绝。
7	SCHA_ERR_SEQID	资源、资源组或资源类型自上次 <code>scha*_open</code> 调用后已进行更新。
8	SCHA_ERR_DEPEND	对象依赖性问题。
9	SCHA_ERR_STATE	对象处于错误状况。
10	SCHA_ERR_METHOD	方法无效。
11	SCHA_ERR_NODE	节点无效。
12	SCHA_ERR_RG	资源组无效。
13	SCHA_ERR_RT	资源类型无效。
14	SCHA_ERR_RSRC	资源无效。
15	SCHA_ERR_PROP	属性无效。
16	SCHA_ERR_CHECKS	健全性检查失败。
17	SCHA_ERR_RSTATUS	错误的资源状态。
18	SCHA_ERR_INTERNAL	遇到内部错误。
19	SCHA_ERR_CLUSTER	无法与其他群集通信。

名称

scha_cluster_open, scha_cluster_open_zone, scha_cluster_get,
scha_cluster_get_zone, scha_cluster_close — 访问并获取有关群集的信息

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l scha#include <scha.h>scha_err_t scha_cluster_open(
    scha_cluster_t *handle);

scha_err_t scha_cluster_open_zone(const char *cluster,
    scha_cluster_t *handlep);

scha_err_t scha_cluster_get(scha_cluster_t handle, const char **
    tag, ...);

scha_err_t scha_cluster_get_zone(const char *cluster,
    scha_cluster_t handlep, const char *cluster_tag, ...);

scha_err_t scha_cluster_close(scha_cluster_t handle);
```

将 `scha_cluster_open()`、`scha_cluster_get()` 和 `scha_cluster_close()` 函数一起使用可以获取有关群集的信息。

`scha_cluster_open()` 初始化群集访问，并返回访问句柄以供 `scha_cluster_get()` 使用。`handle` 参数是变量的地址，用于保存函数调用返回的值。

`scha_cluster_get()` 根据 `tag` 参数的指示访问群集信息。`handle` 参数是从对 `scha_cluster_open()` 的之前调用返回的值。`tag` 参数是字符串值，在 `scha_tags.h` 头文件中由宏定义。`tag` 之后的参数取决于 `tag` 参数的值。

您可能需要提供 `tag` 参数后面的其他参数，以指示将从中检索信息的群集节点。参数列表中的最后一个参数将成为适用于保存 `tag` 参数指示的信息的类型。该参数是群集信息的输出参数。如果函数失败，不会为输出参数返回值。对于 `scha_cluster_get()` 函数的句柄调用 `scha_cluster_close()` 之前，所分配的用于保存 `scha_cluster_get()` 函数返回的信息的内存将保持不变。

`scha_cluster_close()` 将获取从对 `scha_cluster_get()` 函数的之前调用返回的 `handle` 参数。该函数使句柄无效，并将用来返回值而分配的内存释放给使用句柄进行的 `scha_cluster_get()` 调用。请注意，如果需要返回值，则为每个 `get` 调用分配内存。在一个调用中为返回值而分配的空间不会被后续调用覆盖和重新使用。

`scha_cluster_open_zone()` 和 `scha_cluster_get_zone()` 函数的作用分别与 `scha_cluster_open()` 和 `scha_cluster_get()` 相同，且带有一个额外的 `cluster` 参数，用于指定资源组存在于其中且您要在其上执行操作的区域群集的名称。当代码在全局区域中执行，但需要在指定区域群集上操作时，这些函数十分有用。无法在某一区域群集中使用它们来访问其他区域群集。

如果 `scha_cluster_open_zone()` 或 `scha_cluster_get_zone()` 的 `cluster` 参数为 `NULL`，则会对从中执行调用的群集执行查询；换句话说，带有 `NULL` 参数的调用分别等效于 `scha_cluster_open()` 或 `scha_cluster_get()`。

要关闭由 `scha_cluster_open_zone()` 返回的句柄，请使用 `scha_cluster_close()`。不需要使用 `cluster` 参数。

可用于 *tag* 参数的宏

在 `scha_tags.h` 中定义的可用作 *tag* 参数的宏如下所示。指示输出参数和任何其他参数的类型。[scha_calls\(3HA\) \[897\]](#) 介绍了结构和 `enum` 类型。

SCHA_ALL_LOADLIMITS

输出参数类型为 `scha_str_array_t**`。

该宏返回在群集中定义的所有 `loadlimit` 名称。

SCHA_ALL_NODEIDS

输出参数类型为 `scha_uint_array_t**`。

该宏返回群集中所有节点的数字节点标识符。

SCHA_ALL_NODENAMES

输出参数类型为 `scha_str_array_t**`。

该宏返回群集中所有节点的名称。

SCHA_ALL_PRIVATELINK_HOSTNAMES

输出参数类型为 `scha_str_array_t**`。

该宏返回所有群集节点的主机名，根据这些主机名在群集互连上找到节点。

SCHA_ALL_PSTRINGS

输出参数类型为 `scha_str_array_t**`。

该宏返回群集中定义的所有专用字符串的名称。有关专用字符串的更多信息，请参见 [clpstring\(1CL\) \[175\]](#) 手册页。

SCHA_ALL_RESOURCEGROUPS

输出参数类型为 `scha_str_array_t**`。

该宏返回在群集上被管理的所有资源组的名称。

SCHA_ALL_RESOURCETYPES

输出参数类型为 `scha_str_array_t**`。

该宏返回在群集上注册的所有资源类型的名称。

SCHA_CLUSTERNAME

输出参数类型为 `char**`。

该宏返回群集的名称。

SCHA_HARD_LOADLIMIT

输出参数类型为 `scha_str_array_t**`。

该宏为指定的限制名称返回群集中所有节点的硬负载限制值。它需要作为负载限制名称字符串的其他参数，其类型为 `char *`。

字符串数组输出的每个元素的格式均为 `"%s=%d"`，其中，左侧字符串是 *nodename*，右侧整数是该节点上指定的限制名称的硬负载限制值。如果没有指定硬限制，将显示值 `-1` 作为硬限制。

SCHA_LOADLIMIT_PROPS

输出参数类型为 `scha_str_array_t**`。

该宏根据指定的限制名称为群集中的所有节点返回硬负载和软负载限制值（以 / 分隔）。它需要作为负载限制名称字符串的其他参数，其类型为 `char *`。

字符串数组输出的每个元素都是 `"%s=%d/%d"` 格式的字符串，其中，左侧的字符串是 *nodename*，第一个整数是软限制，第二个整数是硬限制。如果没有指定硬限制，将显示值 `-1` 作为硬限制。如果没有指定软限制，将显示值 `0` 作为软限制。

SCHA_LOADLIMITS_NODE

输出参数类型为 `scha_str_array_t**`。

该宏为特定节点返回负载限制（用 / 分隔）和限制名称。它需要类型为 `char *` 的 *nodename* 的附加参数。

字符串数组输出的每个元素都是 `"%s=%d/%d"` 格式的字符串，其中，字符串是指定节点上定义的限制名称，第一个整数是软限制值，第二个整数是硬限制值。如果没有指定硬限制，将显示值 `-1` 作为硬限制。如果没有指定软限制，将显示值 `0` 作为软限制。

SCHA_NODEID_LOCAL

输出参数类型为 `uint_t*`。

该宏返回在其中执行命令的节点的数字节点标识符。

SCHA_NODEID_NODENAME

输出参数类型为 `uint_t*`。附加参数的类型为 `char *`。该宏需要一个作为群集节点名称的附加参数。

该宏返回由名称指示的节点的数字节点标识符。

SCHA_NODENAME_LOCAL

输出参数类型为 `char**`。

该宏返回在其中执行函数的群集节点的名称。

SCHA_NODENAME_NODEID

输出参数类型为 `char**`。附加参数的类型为 `uint_t`。该附加参数是数字群集节点标识符。

该宏返回由数字标识符指示的节点的名称。

SCHA_NODESTATE_LOCAL

输出参数类型为 `scha_node_state_t*`。

该宏返回 `SCHA_NODE_UP` 或 `SCHA_NODE_DOWN`，具体取决于在其中执行命令的节点的状态。

SCHA_NODESTATE_NODE

输出参数类型为 `scha_node_state_t*`。附加参数的类型为 `char*`。该宏需要作为群集节点名称的其他未标记参数。

根据命名节点的状态，该宏返回 `SCHA_NODE_UP` 或 `SCHA_NODE_DOWN`。

SCHA_PRIVATELINK_HOSTNAME_LOCAL

输出参数类型为 `char**`。

该宏返回主机名，根据该主机名在群集互连上找到在其上运行命令的节点。

SCHA_PRIVATELINK_HOSTNAME_NODE

输出参数类型为 `char**`。附加参数的类型为 `char *`。该宏需要作为群集节点名称的其他未标记参数。

该宏返回主机名，根据该主机名在群集互连上找到命名的节点。

SCHA_PSTRING

输出参数类型为 `char**`。

该宏返回专用字符串的明文值。它需要提供专用字符串名称的附加参数，其类型为 `char*`。超级用户以外的用户需要具有 `solaris.cluster.modify` 基于角色的访问控制 (role-based access control, RBAC) 授权才能使用该查询标记。有关专用字符串的更多信息，请参见 [clpstring\(1CL\) \[175\]](#) 手册页。

SCHA_RESOURCE_SECURITY

输出参数类型为 `char**`。

该宏返回 `resource_security` 群集属性的当前设置。

SCHA_SOFT_LOADLIMIT

输出参数类型为 `scha_str_array_t**`。

该宏为指定的限制名称返回群集中所有节点的软负载限制值。它需要作为负载限制名称字符串的其他参数，其类型为 `char *`。

字符串数组输出的每个元素的格式均为 `"%s=%d"`，其中，左侧字符串是 `nodename`，右侧整数是该节点上指定的限制名称的软负载限制值。如果没有指定软限制，将显示值 `0` 作为软限制。

SCHA_SYSLOG_FACILITY

输出参数类型为 `int*`。

该宏返回 RGM 用于日志消息的 [Unresolved link to "syslog3C"](#) 设备的编号。返回的值为 24，它对应于 LOG_DAEMON 设备值。

SCHA_ERR_NOERR

该函数成功。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

例 369 使用 `scha_cluster_get()` 函数

以下示例使用 `scha_cluster_get()` 函数获取所有群集节点的名称。该函数还确定节点是处于打开状态还是关闭状态。

该代码示例还显示为每个群集节点（在其中配置该限制）的 `limitname`（名为 `mylimit`）配置的软负载限制和硬负载限制设置。每个节点的负载限制值按以下格式显示：`nodename=softlimit[/hardlimit]`，其中，如果没有设置硬限制，则 `hardlimit` 值是没有限制的 (-1)。

```
#include <scha.h>
#include <stdio.h>
#include <stdlib.h>

main()
{
    scha_err_t          err;
    scha_node_state_t  node_state;
    scha_str_array_t   *all_nodenames;
    scha_cluster_t     handle;
    int                ix;
    const char         *str;
    scha_str_array_t   *load_limits;

    err = scha_cluster_open(&handle);
    if (err != SCHA_ERR_NOERR) {
        fprintf(stderr, "FAILED: scha_cluster_open()0);
        exit(err);
    }

    err = scha_cluster_get(handle, SCHA_ALL_NODENAMES, &all_nodenames);
    if (err != SCHA_ERR_NOERR) {
        fprintf(stderr, "FAILED: scha_cluster_get()0);
        exit(err);
    }

    for (ix = 0; ix < all_nodenames->array_cnt; ix++) {
        err = scha_cluster_get(handle, SCHA_NODESTATE_NODE,
            all_nodenames->str_array[ix], &node_state);
        if (err != SCHA_ERR_NOERR) {
            fprintf(stderr, "FAILED: scha_cluster_get()"
```

```

        "SCHA_NODESTATE_NODE0);
        exit(err);
    }

    switch (node_state) {
    case SCHA_NODE_UP:
        str = "UP";
        break;
    case SCHA_NODE_DOWN:
        str = "DOWN";
        break;
    }

    printf("State of node: %s value: %s\n",
        all_nodenames->str_array[ix], str);
}
err = scha_cluster_get(handle, SCHA_LOADLIMIT_PROPS, "mylimit",
    &load_limits);

printf("\n\nLoad limits settings for limitname 'mylimit':\n\n");

for (ix = 0; ix < load_limits->array_cnt; ix++) {
    printf("%s\n", load_limits->str_array[ix]);
}
}

```

/usr/cluster/include/scha.h

头文件

/usr/cluster/lib/libscha.so

库

有关以下属性的描述，请参见 [Unresolved link to " attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scha_cluster_get\(1HA\) \[567\]](#)、[scha_calls\(3HA\) \[897\]](#)、[scha_cluster_getlogfacility\(3HA\) \[921\]](#)、[scha_cluster_getlogfacility\(3HA\) \[921\]](#)、[scha_cluster_getlogfacility\(3HA\) \[921\]](#)、[scha_cluster_getlogfacility\(3HA\) \[921\]](#)、[link to " syslog3C"](#)、[Unresolved link to " attributes5"](#)、[rg_properties\(5\) \[1129\]](#)

名称

scha_cluster_open, scha_cluster_open_zone, scha_cluster_get,
scha_cluster_get_zone, scha_cluster_close — 访问并获取有关群集的信息

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l scha#include <scha.h>scha_err_t scha_cluster_open(
    scha_cluster_t *handle);

scha_err_t scha_cluster_open_zone(const char *cluster,
    scha_cluster_t *handlep);

scha_err_t scha_cluster_get(scha_cluster_t handle, const char **
    tag, ...);

scha_err_t scha_cluster_get_zone(const char *cluster,
    scha_cluster_t handlep, const char *cluster_tag, ...);

scha_err_t scha_cluster_close(scha_cluster_t handle);
```

将 `scha_cluster_open()`、`scha_cluster_get()` 和 `scha_cluster_close()` 函数一起使用可以获取有关群集的信息。

`scha_cluster_open()` 初始化群集访问，并返回访问句柄以供 `scha_cluster_get()` 使用。`handle` 参数是变量的地址，用于保存函数调用返回的值。

`scha_cluster_get()` 根据 `tag` 参数的指示访问群集信息。`handle` 参数是从对 `scha_cluster_open()` 的之前调用返回的值。`tag` 参数是字符串值，在 `scha_tags.h` 头文件中由宏定义。`tag` 之后的参数取决于 `tag` 参数的值。

您可能需要提供 `tag` 参数后面的其他参数，以指示将从中检索信息的群集节点。参数列表中的最后一个参数将成为适用于保存 `tag` 参数指示的信息的类型。该参数是群集信息的输出参数。如果函数失败，不会为输出参数返回值。对于 `scha_cluster_get()` 函数的句柄调用 `scha_cluster_close()` 之前，所分配的用于保存 `scha_cluster_get()` 函数返回的信息的内存将保持不变。

`scha_cluster_close()` 将获取从对 `scha_cluster_get()` 函数的之前调用返回的 `handle` 参数。该函数使句柄无效，并将用来返回值而分配的内存释放给使用句柄进行的 `scha_cluster_get()` 调用。请注意，如果需要返回值，则为每个 `get` 调用分配内存。在一个调用中为返回值而分配的空间不会被后续调用覆盖和重新使用。

`scha_cluster_open_zone()` 和 `scha_cluster_get_zone()` 函数的作用分别与 `scha_cluster_open()` 和 `scha_cluster_get()` 相同，且带有一个额外的 `cluster` 参数，用于指定资源组存在于其中且您要在其上执行操作的区域群集的名称。当代码在全局区域中执行，但需要在指定区域群集上操作时，这些函数十分有用。无法在某一区域群集中使用它们来访问其他区域群集。

如果 `scha_cluster_open_zone()` 或 `scha_cluster_get_zone()` 的 `cluster` 参数为 `NULL`，则会对从中执行调用的群集执行查询；换句话说，带有 `NULL` 参数的调用分别等效于 `scha_cluster_open()` 或 `scha_cluster_get()`。

要关闭由 `scha_cluster_open_zone()` 返回的句柄，请使用 `scha_cluster_close()`。不需要使用 `cluster` 参数。

可用于 *tag* 参数的宏

在 `scha_tags.h` 中定义的可用作 *tag* 参数的宏如下所示。指示输出参数和任何其他参数的类型。[scha_calls\(3HA\) \[897\]](#) 介绍了结构和 `enum` 类型。

SCHA_ALL_LOADLIMITS

输出参数类型为 `scha_str_array_t**`。

该宏返回在群集中定义的所有 `loadlimit` 名称。

SCHA_ALL_NODEIDS

输出参数类型为 `scha_uint_array_t**`。

该宏返回群集中所有节点的数字节点标识符。

SCHA_ALL_NODENAMES

输出参数类型为 `scha_str_array_t**`。

该宏返回群集中所有节点的名称。

SCHA_ALL_PRIVATELINK_HOSTNAMES

输出参数类型为 `scha_str_array_t**`。

该宏返回所有群集节点的主机名，根据这些主机名在群集互连上找到节点。

SCHA_ALL_PSTRINGS

输出参数类型为 `scha_str_array_t**`。

该宏返回群集中定义的所有专用字符串的名称。有关专用字符串的更多信息，请参见 [clpstring\(1CL\) \[175\]](#) 手册页。

SCHA_ALL_RESOURCEGROUPS

输出参数类型为 `scha_str_array_t**`。

该宏返回在群集上被管理的所有资源组的名称。

SCHA_ALL_RESOURCETYPES

输出参数类型为 `scha_str_array_t**`。

该宏返回在群集上注册的所有资源类型的名称。

SCHA_CLUSTERNAME

输出参数类型为 `char**`。

该宏返回群集的名称。

SCHA_HARD_LOADLIMIT

输出参数类型为 `scha_str_array_t**`。

该宏为指定的限制名称返回群集中所有节点的硬负载限制值。它需要作为负载限制名称字符串的其他参数，其类型为 `char *`。

字符串数组输出的每个元素的格式均为 `"%s=%d"`，其中，左侧字符串是 *nodename*，右侧整数是该节点上指定的限制名称的硬负载限制值。如果没有指定硬限制，将显示值 `-1` 作为硬限制。

SCHA_LOADLIMIT_PROPS

输出参数类型为 `scha_str_array_t**`。

该宏根据指定的限制名称为群集中的所有节点返回硬负载和软负载限制值（以 / 分隔）。它需要作为负载限制名称字符串的其他参数，其类型为 `char *`。

字符串数组输出的每个元素都是 `"%s=%d/%d"` 格式的字符串，其中，左侧的字符串是 *nodename*，第一个整数是软限制，第二个整数是硬限制。如果没有指定硬限制，将显示值 `-1` 作为硬限制。如果没有指定软限制，将显示值 `0` 作为软限制。

SCHA_LOADLIMITS_NODE

输出参数类型为 `scha_str_array_t**`。

该宏为特定节点返回负载限制（用 / 分隔）和限制名称。它需要类型为 `char *` 的 *nodename* 的附加参数。

字符串数组输出的每个元素都是 `"%s=%d/%d"` 格式的字符串，其中，字符串是指定节点上定义的限制名称，第一个整数是软限制值，第二个整数是硬限制值。如果没有指定硬限制，将显示值 `-1` 作为硬限制。如果没有指定软限制，将显示值 `0` 作为软限制。

SCHA_NODEID_LOCAL

输出参数类型为 `uint_t*`。

该宏返回在其中执行命令的节点的数字节点标识符。

SCHA_NODEID_NODENAME

输出参数类型为 `uint_t*`。附加参数的类型为 `char *`。该宏需要一个作为群集节点名称的附加参数。

该宏返回由名称指示的节点的数字节点标识符。

SCHA_NODENAME_LOCAL

输出参数类型为 `char**`。

该宏返回在其中执行函数的群集节点的名称。

SCHA_NODENAME_NODEID

输出参数类型为 `char**`。附加参数的类型为 `uint_t`。该附加参数是数字群集节点标识符。

该宏返回由数字标识符指示的节点的名称。

SCHA_NODESTATE_LOCAL

输出参数类型为 `scha_node_state_t*`。

该宏返回 `SCHA_NODE_UP` 或 `SCHA_NODE_DOWN`，具体取决于在其中执行命令的节点的状态。

SCHA_NODESTATE_NODE

输出参数类型为 `scha_node_state_t*`。附加参数的类型为 `char*`。该宏需要作为群集节点名称的其他未标记参数。

根据命名节点的状态，该宏返回 `SCHA_NODE_UP` 或 `SCHA_NODE_DOWN`。

SCHA_PRIVATELINK_HOSTNAME_LOCAL

输出参数类型为 `char**`。

该宏返回主机名，根据该主机名在群集互连上找到在其上运行命令的节点。

SCHA_PRIVATELINK_HOSTNAME_NODE

输出参数类型为 `char**`。附加参数的类型为 `char *`。该宏需要作为群集节点名称的其他未标记参数。

该宏返回主机名，根据该主机名在群集互连上找到命名的节点。

SCHA_PSTRING

输出参数类型为 `char**`。

该宏返回专用字符串的明文值。它需要提供专用字符串名称的附加参数，其类型为 `char*`。超级用户以外的用户需要具有 `solaris.cluster.modify` 基于角色的访问控制 (role-based access control, RBAC) 授权才能使用该查询标记。有关专用字符串的更多信息，请参见 [clpstring\(1CL\) \[175\]](#) 手册页。

SCHA_RESOURCE_SECURITY

输出参数类型为 `char**`。

该宏返回 `resource_security` 群集属性的当前设置。

SCHA_SOFT_LOADLIMIT

输出参数类型为 `scha_str_array_t**`。

该宏为指定的限制名称返回群集中所有节点的软负载限制值。它需要作为负载限制名称字符串的其他参数，其类型为 `char *`。

字符串数组输出的每个元素的格式均为 `"%s=%d"`，其中，左侧字符串是 `nodename`，右侧整数是该节点上指定的限制名称的软负载限制值。如果没有指定软限制，将显示值 `0` 作为软限制。

SCHA_SYSLOG_FACILITY

输出参数类型为 `int*`。

该宏返回 RGM 用于日志消息的 [Unresolved link to "syslog3C"](#) 设备的编号。返回的值为 24，它对应于 LOG_DAEMON 设备值。

SCHA_ERR_NOERR

该函数成功。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

例 370 使用 `scha_cluster_get()` 函数

以下示例使用 `scha_cluster_get()` 函数获取所有群集节点的名称。该函数还确定节点是处于打开状态还是关闭状态。

该代码示例还显示为每个群集节点（在其中配置该限制）的 `limitname`（名为 `mylimit`）配置的软负载限制和硬负载限制设置。每个节点的负载限制值按以下格式显示：`nodename=softlimit[/hardlimit]`，其中，如果没有设置硬限制，则 `hardlimit` 值是没有限制的 (-1)。

```
#include <scha.h>
#include <stdio.h>
#include <stdlib.h>

main()
{
    scha_err_t          err;
    scha_node_state_t  node_state;
    scha_str_array_t   *all_nodenames;
    scha_cluster_t     handle;
    int                 ix;
    const char         *str;
    scha_str_array_t   *load_limits;

    err = scha_cluster_open(&handle);
    if (err != SCHA_ERR_NOERR) {
        fprintf(stderr, "FAILED: scha_cluster_open()0);
        exit(err);
    }

    err = scha_cluster_get(handle, SCHA_ALL_NODENAMES, &all_nodenames);
    if (err != SCHA_ERR_NOERR) {
        fprintf(stderr, "FAILED: scha_cluster_get()0);
        exit(err);
    }

    for (ix = 0; ix < all_nodenames->array_cnt; ix++) {
        err = scha_cluster_get(handle, SCHA_NODESTATE_NODE,
            all_nodenames->str_array[ix], &node_state);
        if (err != SCHA_ERR_NOERR) {
            fprintf(stderr, "FAILED: scha_cluster_get()"
```

```

        "SCHA_NODESTATE_NODE0);
        exit(err);
    }

    switch (node_state) {
    case SCHA_NODE_UP:
        str = "UP";
        break;
    case SCHA_NODE_DOWN:
        str = "DOWN";
        break;
    }

    printf("State of node: %s value: %s\n",
        all_nodenames->str_array[ix], str);
}
err = scha_cluster_get(handle, SCHA_LOADLIMIT_PROPS, "mylimit",
    &load_limits);

printf("\n\nLoad limits settings for limitname 'mylimit':\n\n");

for (ix = 0; ix < load_limits->array_cnt; ix++) {
    printf("%s\n", load_limits->str_array[ix]);
}
}

```

/usr/cluster/include/scha.h

头文件

/usr/cluster/lib/libscha.so

库

有关以下属性的描述，请参见 [Unresolved link to " attributes5"](#) :

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scha_cluster_get\(1HA\) \[567\]](#)、[scha_calls\(3HA\) \[897\]](#)、[scha_cluster_getlogfacility\(3HA\) \[921\]](#)、[scha](#)
[link to " syslog3C"](#)、[Unresolved link to " attributes5"](#)、[rg_properties\(5\) \[1129\]](#)

名称

`scha_cluster_open`, `scha_cluster_open_zone`, `scha_cluster_get`,
`scha_cluster_get_zone`, `scha_cluster_close` — 访问并获取有关群集的信息

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l scha#include <scha.h>scha_err_t scha_cluster_open(
    scha_cluster_t *handle);

scha_err_t scha_cluster_open_zone(const char *cluster,
    scha_cluster_t *handlep);

scha_err_t scha_cluster_get(scha_cluster_t handle, const char **
    tag, ...);

scha_err_t scha_cluster_get_zone(const char *cluster,
    scha_cluster_t handlep, const char *cluster_tag, ...);

scha_err_t scha_cluster_close(scha_cluster_t handle);
```

将 `scha_cluster_open()`、`scha_cluster_get()` 和 `scha_cluster_close()` 函数一起使用可以获取有关群集的信息。

`scha_cluster_open()` 初始化群集访问，并返回访问句柄以供 `scha_cluster_get()` 使用。`handle` 参数是变量的地址，用于保存函数调用返回的值。

`scha_cluster_get()` 根据 `tag` 参数的指示访问群集信息。`handle` 参数是从对 `scha_cluster_open()` 的之前调用返回的值。`tag` 参数是字符串值，在 `scha_tags.h` 头文件中由宏定义。`tag` 之后的参数取决于 `tag` 参数的值。

您可能需要提供 `tag` 参数后面的其他参数，以指示将从中检索信息的群集节点。参数列表中的最后一个参数将成为适用于保存 `tag` 参数指示的信息的类型。该参数是群集信息的输出参数。如果函数失败，不会为输出参数返回值。对用于 `scha_cluster_get()` 函数的句柄调用 `scha_cluster_close()` 之前，所分配的用于保存 `scha_cluster_get()` 函数返回的信息的内存将保持不变。

`scha_cluster_close()` 将获取从对 `scha_cluster_get()` 函数的之前调用返回的 `handle` 参数。该函数使句柄无效，并将用来返回值而分配的内存释放给使用句柄进行的 `scha_cluster_get()` 调用。请注意，如果需要返回值，则为每个 `get` 调用分配内存。在一个调用中为返回值而分配的空间不会被后续调用覆盖和重新使用。

`scha_cluster_open_zone()` 和 `scha_cluster_get_zone()` 函数的作用分别与 `scha_cluster_open()` 和 `scha_cluster_get()` 相同，且带有一个额外的 `cluster` 参数，用于指定资源组存在于其中且您要在其上执行操作的区域群集的名称。当代码在全局区域中执行，但需要在指定区域群集上操作时，这些函数十分有用。无法在某一区域群集中使用它们来访问其他区域群集。

如果 `scha_cluster_open_zone()` 或 `scha_cluster_get_zone()` 的 `cluster` 参数为 `NULL`，则会对从中执行调用的群集执行查询；换句话说，带有 `NULL` 参数的调用分别等效于 `scha_cluster_open()` 或 `scha_cluster_get()`。

要关闭由 `scha_cluster_open_zone()` 返回的句柄，请使用 `scha_cluster_close()`。不需要使用 `cluster` 参数。

可用于 *tag* 参数的宏

在 `scha_tags.h` 中定义的可用作 *tag* 参数的宏如下所示。指示输出参数和任何其他参数的类型。[scha_calls\(3HA\) \[897\]](#) 介绍了结构和 `enum` 类型。

SCHA_ALL_LOADLIMITS

输出参数类型为 `scha_str_array_t**`。

该宏返回在群集中定义的所有 `loadlimit` 名称。

SCHA_ALL_NODEIDS

输出参数类型为 `scha_uint_array_t**`。

该宏返回群集中所有节点的数字节点标识符。

SCHA_ALL_NODENAMES

输出参数类型为 `scha_str_array_t**`。

该宏返回群集中所有节点的名称。

SCHA_ALL_PRIVATELINK_HOSTNAMES

输出参数类型为 `scha_str_array_t**`。

该宏返回所有群集节点的主机名，根据这些主机名在群集互连上找到节点。

SCHA_ALL_PSTRINGS

输出参数类型为 `scha_str_array_t**`。

该宏返回群集中定义的所有专用字符串的名称。有关专用字符串的更多信息，请参见 [clpstring\(1CL\) \[175\]](#) 手册页。

SCHA_ALL_RESOURCEGROUPS

输出参数类型为 `scha_str_array_t**`。

该宏返回在群集上被管理的所有资源组的名称。

SCHA_ALL_RESOURCETYPES

输出参数类型为 `scha_str_array_t**`。

该宏返回在群集上注册的所有资源类型的名称。

SCHA_CLUSTERNAME

输出参数类型为 `char**`。

该宏返回群集的名称。

SCHA_HARD_LOADLIMIT

输出参数类型为 `scha_str_array_t**`。

该宏为指定的限制名称返回群集中所有节点的硬负载限制值。它需要作为负载限制名称字符串的其他参数，其类型为 `char *`。

字符串数组输出的每个元素的格式均为 `"%s=%d"`，其中，左侧字符串是 *nodename*，右侧整数是该节点上指定的限制名称的硬负载限制值。如果没有指定硬限制，将显示值 `-1` 作为硬限制。

SCHA_LOADLIMIT_PROPS

输出参数类型为 `scha_str_array_t**`。

该宏根据指定的限制名称为群集中的所有节点返回硬负载和软负载限制值（以 / 分隔）。它需要作为负载限制名称字符串的其他参数，其类型为 `char *`。

字符串数组输出的每个元素都是 `"%s=%d/%d"` 格式的字符串，其中，左侧的字符串是 *nodename*，第一个整数是软限制，第二个整数是硬限制。如果没有指定硬限制，将显示值 `-1` 作为硬限制。如果没有指定软限制，将显示值 `0` 作为软限制。

SCHA_LOADLIMITS_NODE

输出参数类型为 `scha_str_array_t**`。

该宏为特定节点返回负载限制（用 / 分隔）和限制名称。它需要类型为 `char *` 的 *nodename* 的附加参数。

字符串数组输出的每个元素都是 `"%s=%d/%d"` 格式的字符串，其中，字符串是指定节点上定义的限制名称，第一个整数是软限制值，第二个整数是硬限制值。如果没有指定硬限制，将显示值 `-1` 作为硬限制。如果没有指定软限制，将显示值 `0` 作为软限制。

SCHA_NODEID_LOCAL

输出参数类型为 `uint_t*`。

该宏返回在其中执行命令的节点的数字节点标识符。

SCHA_NODEID_NODENAME

输出参数类型为 `uint_t*`。附加参数的类型为 `char *`。该宏需要一个作为群集节点名称的附加参数。

该宏返回由名称指示的节点的数字节点标识符。

SCHA_NODENAME_LOCAL

输出参数类型为 `char**`。

该宏返回在其中执行函数的群集节点的名称。

SCHA_NODENAME_NODEID

输出参数类型为 `char**`。附加参数的类型为 `uint_t`。该附加参数是数字群集节点标识符。

该宏返回由数字标识符指示的节点的名称。

SCHA_NODESTATE_LOCAL

输出参数类型为 `scha_node_state_t*`。

该宏返回 `SCHA_NODE_UP` 或 `SCHA_NODE_DOWN`，具体取决于在其中执行命令的节点的状态。

SCHA_NODESTATE_NODE

输出参数类型为 `scha_node_state_t*`。附加参数的类型为 `char*`。该宏需要作为群集节点名称的其他未标记参数。

根据命名节点的状态，该宏返回 `SCHA_NODE_UP` 或 `SCHA_NODE_DOWN`。

SCHA_PRIVATELINK_HOSTNAME_LOCAL

输出参数类型为 `char**`。

该宏返回主机名，根据该主机名在群集互连上找到在其上运行命令的节点。

SCHA_PRIVATELINK_HOSTNAME_NODE

输出参数类型为 `char**`。附加参数的类型为 `char *`。该宏需要作为群集节点名称的其他未标记参数。

该宏返回主机名，根据该主机名在群集互连上找到命名的节点。

SCHA_PSTRING

输出参数类型为 `char**`。

该宏返回专用字符串的明文值。它需要提供专用字符串名称的附加参数，其类型为 `char*`。超级用户以外的用户需要具有 `solaris.cluster.modify` 基于角色的访问控制 (role-based access control, RBAC) 授权才能使用该查询标记。有关专用字符串的更多信息，请参见 [clpstring\(1CL\) \[175\]](#) 手册页。

SCHA_RESOURCE_SECURITY

输出参数类型为 `char**`。

该宏返回 `resource_security` 群集属性的当前设置。

SCHA_SOFT_LOADLIMIT

输出参数类型为 `scha_str_array_t**`。

该宏为指定的限制名称返回群集中所有节点的软负载限制值。它需要作为负载限制名称字符串的其他参数，其类型为 `char *`。

字符串数组输出的每个元素的格式均为 `"%s=%d"`，其中，左侧字符串是 `nodename`，右侧整数是该节点上指定的限制名称的软负载限制值。如果没有指定软限制，将显示值 `0` 作为软限制。

SCHA_SYSLOG_FACILITY

输出参数类型为 `int*`。

该宏返回 RGM 用于日志消息的 [Unresolved link to "syslog3C"](#) 设备的编号。返回的值为 24，它对应于 LOG_DAEMON 设备值。

SCHA_ERR_NOERR

该函数成功。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

例 371 使用 `scha_cluster_get()` 函数

以下示例使用 `scha_cluster_get()` 函数获取所有群集节点的名称。该函数还确定节点是处于打开状态还是关闭状态。

该代码示例还显示为每个群集节点（在其中配置该限制）的 `limitname`（名为 `mylimit`）配置的软负载限制和硬负载限制设置。每个节点的负载限制值按以下格式显示：`nodename=softlimit[/hardlimit]`，其中，如果没有设置硬限制，则 `hardlimit` 值是没有限制的 (-1)。

```
#include <scha.h>
#include <stdio.h>
#include <stdlib.h>

main()
{
    scha_err_t          err;
    scha_node_state_t  node_state;
    scha_str_array_t   *all_nodenames;
    scha_cluster_t     handle;
    int                 ix;
    const char         *str;
    scha_str_array_t   *load_limits;

    err = scha_cluster_open(&handle);
    if (err != SCHA_ERR_NOERR) {
        fprintf(stderr, "FAILED: scha_cluster_open()0);
        exit(err);
    }

    err = scha_cluster_get(handle, SCHA_ALL_NODENAMES, &all_nodenames);
    if (err != SCHA_ERR_NOERR) {
        fprintf(stderr, "FAILED: scha_cluster_get()0);
        exit(err);
    }

    for (ix = 0; ix < all_nodenames->array_cnt; ix++) {
        err = scha_cluster_get(handle, SCHA_NODESTATE_NODE,
            all_nodenames->str_array[ix], &node_state);
        if (err != SCHA_ERR_NOERR) {
            fprintf(stderr, "FAILED: scha_cluster_get()"
```

```

        "SCHA_NODESTATE_NODE0);
        exit(err);
    }

    switch (node_state) {
    case SCHA_NODE_UP:
        str = "UP";
        break;
    case SCHA_NODE_DOWN:
        str = "DOWN";
        break;
    }

    printf("State of node: %s value: %s\n",
        all_nodenames->str_array[ix], str);
}
err = scha_cluster_get(handle, SCHA_LOADLIMIT_PROPS, "mylimit",
    &load_limits);

printf("\n\nLoad limits settings for limitname 'mylimit':\n\n");

for (ix = 0; ix < load_limits->array_cnt; ix++) {
    printf("%s\n", load_limits->str_array[ix]);
}
}

```

/usr/cluster/include/scha.h

头文件

/usr/cluster/lib/libscha.so

库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scha_cluster_get\(1HA\) \[567\]](#)、[scha_calls\(3HA\) \[897\]](#)、[scha_cluster_getlogfacility\(3HA\) \[921\]](#)、[scha_cluster_getlogfacility\(3HA\) \[921\]](#)、[scha_cluster_getlogfacility\(3HA\) \[921\]](#)、[scha_cluster_getlogfacility\(3HA\) \[921\]](#)、[link to "syslog3C"](#)、[Unresolved link to "attributes5"](#)、[rg_properties\(5\) \[1129\]](#)

名称

scha_cluster_getlogfacility — 群集日志设备访问

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l scha#include <scha.h> scha_err_t
scha_cluster_getlogfacility(int *logfacility);
```

scha_cluster_getlogfacility () 函数可返回要用作群集日志的系统日志设备号。该值将供资源类型实现用于 Solaris [Unresolved link to "syslog3C"](#) 函数，以将事件和状态消息记录到群集日志。

该函数将返回一个错误状态，如果成功，将返回由 *logfacility* 参数指向的位置中的设备号。

scha_cluster_getlogfacility () 函数返回以下值：

0 该函数成功。

非零值 该函数失败。

SCHA_ERR_NOERR 该函数成功。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

例 372 使用 `scha_cluster_getlogfacility ()` 函数

```
main()
{
    scha_err_t err_code;
    int logfacility;

    err_code = scha_cluster_getlogfacility(&logfacility);

    if (err_code == SCHA_ERR_NOERR) {
        openlog("test resource", LOG_CONS, logfacility);
        syslog(LOG_INFO, "Access function call succeeded.");
    }
}
```

/usr/cluster/include/scha.h 头文件

/usr/cluster/lib/libscha.so 库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[Unresolved link to "syslog3C"](#)、[scha_calls\(3HA\) \[897\]](#)、[scha_cluster_get\(3HA\) \[909\]](#)、[scha_strerror\(3HA\) \[1057\]](#)、[Unresolved link to "attributes5"](#)

名称

`scha_cluster_getnodename` — 返回本地群集节点的名称

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l scha#include <scha.h> scha_err_t scha_cluster_getnodename(
char **nodename);
```

`scha_cluster_getnodename` () 函数返回在其上调用函数的群集节点的名称。节点名称不必与 Solaris 系统名称相同。该函数返回错误状态，如果成功，则在由 `nodename` 参数指向的位置上返回包含节点名称的字符串。

如果调用失败，则 `nodename` 设置为 NULL。`scha_cluster_getnodename` () 的调用方通过使用标准 C 库函数 [Unresolved link to "free3C"](#) 负责释放为返回的字符串分配的内存。仅当函数成功时才释放内存。

`scha_cluster_getnodename` () 函数返回以下值：

0 该函数成功。

非零值 该函数失败。

SCHA_ERR_NOERR 函数成功。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

例 373 使用 `scha_cluster_getnodename` () 函数

```
scha_err_t err_code;
char *nodename;
err_code = scha_cluster_getnodename(&nodename);
...
if (nodename != NULL) free(nodename);
```

/usr/cluster/include/scha.h 头文件

/usr/cluster/lib/libscha.so 库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api

属性类型	属性值
接口稳定性	Evolving (发展中)

Unresolved link to "
free3C"、[scha_calls\(3HA\) \[897\]](#)、[scha_cluster_get\(3HA\) \[909\]](#)、[scha_cluster_getzone\(3HA\) \[925\]](#)、[sc](#)
link to "attributes5"

名称

scha_cluster_getzone — 返回区域的名称

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l scha#include <scha.h> scha_err_t scha_cluster_getzone(
char **zonename);
```

scha_cluster_getzone () 函数返回标识从中调用函数的区域的字符串。如果在非全局区域中调用此函数，则以 `nodename:zonename` 格式返回区域名称和节点名称。如果在全局区域中调用此函数，则只返回节点名称。节点名称不必与 Solaris 系统名称相同。此函数返回错误状态。如果成功，此函数还在 `zonename` 参数指向的位置上返回包含节点名称和区域名称的字符串。

如果调用失败，则 `zonename` 参数设置为 NULL。scha_cluster_getzone () 的调用方通过使用标准 C 库函数 [Unresolved link to "free3C"](#) 负责释放为返回的字符串分配的内存。仅当函数成功时才释放内存。

scha_cluster_getzone () 函数返回以下值：

0 该函数成功。

非零值 该函数失败。

SCHA_ERR_NOERR 该函数成功。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

例 374 使用 scha_cluster_getzone () 函数

```
scha_err_t err_code;
char *zonename;
err_code = scha_cluster_getzone(&zonename);
...
if (zonename != NULL) free(zonename);
```

/usr/cluster/include/scha.h 头文件

/usr/cluster/lib/libscha.so 库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

Unresolved link to "[free3C](#)"、[scha_calls\(3HA\) \[897\]](#)、[scha_cluster_get\(3HA\) \[909\]](#)、[scha_cluster_getnodename\(3HA\) \[923\]](#)
link to "[attributes5](#)"

名称

`scha_cluster_open`, `scha_cluster_open_zone`, `scha_cluster_get`,
`scha_cluster_get_zone`, `scha_cluster_close` — 访问并获取有关群集的信息

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l scha#include <scha.h>scha_err_t scha_cluster_open(
    scha_cluster_t *handle);

scha_err_t scha_cluster_open_zone(const char *cluster,
    scha_cluster_t *handlep);

scha_err_t scha_cluster_get(scha_cluster_t handle, const char **
    tag, ...);

scha_err_t scha_cluster_get_zone(const char *cluster,
    scha_cluster_t handlep, const char *cluster_tag, ...);

scha_err_t scha_cluster_close(scha_cluster_t handle);
```

将 `scha_cluster_open()`、`scha_cluster_get()` 和 `scha_cluster_close()` 函数一起使用可以获取有关群集的信息。

`scha_cluster_open()` 初始化群集访问，并返回访问句柄以供 `scha_cluster_get()` 使用。`handle` 参数是变量的地址，用于保存函数调用返回的值。

`scha_cluster_get()` 根据 `tag` 参数的指示访问群集信息。`handle` 参数是从对 `scha_cluster_open()` 的之前调用返回的值。`tag` 参数是字符串值，在 `scha_tags.h` 头文件中由宏定义。`tag` 之后的参数取决于 `tag` 参数的值。

您可能需要提供 `tag` 参数后面的其他参数，以指示将从中检索信息的群集节点。参数列表中的最后一个参数将成为适用于保存 `tag` 参数指示的信息的类型。该参数是群集信息的输出参数。如果函数失败，不会为输出参数返回值。对于 `scha_cluster_get()` 函数的句柄调用 `scha_cluster_close()` 之前，所分配的用于保存 `scha_cluster_get()` 函数返回的信息的内存将保持不变。

`scha_cluster_close()` 将获取从对 `scha_cluster_get()` 函数的之前调用返回的 `handle` 参数。该函数使句柄无效，并将用来返回值而分配的内存释放给使用句柄进行的 `scha_cluster_get()` 调用。请注意，如果需要返回值，则为每个 `get` 调用分配内存。在一个调用中为返回值而分配的空间不会被后续调用覆盖和重新使用。

`scha_cluster_open_zone()` 和 `scha_cluster_get_zone()` 函数的作用分别与 `scha_cluster_open()` 和 `scha_cluster_get()` 相同，且带有一个额外的 `cluster` 参数，用于指定资源组存在于其中且您要在其上执行操作的区域群集的名称。当代码在全局区域中执行，但需要在指定区域群集上操作时，这些函数十分有用。无法在某一区域群集中使用它们来访问其他区域群集。

如果 `scha_cluster_open_zone()` 或 `scha_cluster_get_zone()` 的 `cluster` 参数为 `NULL`，则会对从中执行调用的群集执行查询；换句话说，带有 `NULL` 参数的调用分别等效于 `scha_cluster_open()` 或 `scha_cluster_get()`。

要关闭由 `scha_cluster_open_zone()` 返回的句柄，请使用 `scha_cluster_close()`。不需要使用 `cluster` 参数。

可用于 *tag* 参数的宏

在 `scha_tags.h` 中定义的可用作 *tag* 参数的宏如下所示。指示输出参数和任何其他参数的类型。[scha_calls\(3HA\) \[897\]](#) 介绍了结构和 `enum` 类型。

SCHA_ALL_LOADLIMITS

输出参数类型为 `scha_str_array_t**`。

该宏返回在群集中定义的所有 `loadlimit` 名称。

SCHA_ALL_NODEIDS

输出参数类型为 `scha_uint_array_t**`。

该宏返回群集中所有节点的数字节点标识符。

SCHA_ALL_NODENAMES

输出参数类型为 `scha_str_array_t**`。

该宏返回群集中所有节点的名称。

SCHA_ALL_PRIVATELINK_HOSTNAMES

输出参数类型为 `scha_str_array_t**`。

该宏返回所有群集节点的主机名，根据这些主机名在群集互连上找到节点。

SCHA_ALL_PSTRINGS

输出参数类型为 `scha_str_array_t**`。

该宏返回群集中定义的所有专用字符串的名称。有关专用字符串的更多信息，请参见 [clpstring\(1CL\) \[175\]](#) 手册页。

SCHA_ALL_RESOURCEGROUPS

输出参数类型为 `scha_str_array_t**`。

该宏返回在群集上被管理的所有资源组的名称。

SCHA_ALL_RESOURCETYPES

输出参数类型为 `scha_str_array_t**`。

该宏返回在群集上注册的所有资源类型的名称。

SCHA_CLUSTERNAME

输出参数类型为 `char**`。

该宏返回群集的名称。

SCHA_HARD_LOADLIMIT

输出参数类型为 `scha_str_array_t**`。

该宏为指定的限制名称返回群集中所有节点的硬负载限制值。它需要作为负载限制名称字符串的其他参数，其类型为 `char *`。

字符串数组输出的每个元素的格式均为 `"%s=%d"`，其中，左侧字符串是 *nodename*，右侧整数是该节点上指定的限制名称的硬负载限制值。如果没有指定硬限制，将显示值 `-1` 作为硬限制。

SCHA_LOADLIMIT_PROPS

输出参数类型为 `scha_str_array_t**`。

该宏根据指定的限制名称为群集中的所有节点返回硬负载和软负载限制值（以 / 分隔）。它需要作为负载限制名称字符串的其他参数，其类型为 `char *`。

字符串数组输出的每个元素都是 `"%s=%d/%d"` 格式的字符串，其中，左侧的字符串是 *nodename*，第一个整数是软限制，第二个整数是硬限制。如果没有指定硬限制，将显示值 `-1` 作为硬限制。如果没有指定软限制，将显示值 `0` 作为软限制。

SCHA_LOADLIMITS_NODE

输出参数类型为 `scha_str_array_t**`。

该宏为特定节点返回负载限制（用 / 分隔）和限制名称。它需要类型为 `char *` 的 *nodename* 的附加参数。

字符串数组输出的每个元素都是 `"%s=%d/%d"` 格式的字符串，其中，字符串是指定节点上定义的限制名称，第一个整数是软限制值，第二个整数是硬限制值。如果没有指定硬限制，将显示值 `-1` 作为硬限制。如果没有指定软限制，将显示值 `0` 作为软限制。

SCHA_NODEID_LOCAL

输出参数类型为 `uint_t*`。

该宏返回在其中执行命令的节点的数字节点标识符。

SCHA_NODEID_NODENAME

输出参数类型为 `uint_t*`。附加参数的类型为 `char *`。该宏需要一个作为群集节点名称的附加参数。

该宏返回由名称指示的节点的数字节点标识符。

SCHA_NODENAME_LOCAL

输出参数类型为 `char**`。

该宏返回在其中执行函数的群集节点的名称。

SCHA_NODENAME_NODEID

输出参数类型为 `char**`。附加参数的类型为 `uint_t`。该附加参数是数字群集节点标识符。

该宏返回由数字标识符指示的节点的名称。

SCHA_NODESTATE_LOCAL

输出参数类型为 `scha_node_state_t*`。

该宏返回 `SCHA_NODE_UP` 或 `SCHA_NODE_DOWN`，具体取决于在其中执行命令的节点的状态。

SCHA_NODESTATE_NODE

输出参数类型为 `scha_node_state_t*`。附加参数的类型为 `char*`。该宏需要作为群集节点名称的其他未标记参数。

根据命名节点的状态，该宏返回 `SCHA_NODE_UP` 或 `SCHA_NODE_DOWN`。

SCHA_PRIVATELINK_HOSTNAME_LOCAL

输出参数类型为 `char**`。

该宏返回主机名，根据该主机名在群集互连上找到在其上运行命令的节点。

SCHA_PRIVATELINK_HOSTNAME_NODE

输出参数类型为 `char**`。附加参数的类型为 `char *`。该宏需要作为群集节点名称的其他未标记参数。

该宏返回主机名，根据该主机名在群集互连上找到命名的节点。

SCHA_PSTRING

输出参数类型为 `char**`。

该宏返回专用字符串的明文值。它需要提供专用字符串名称的附加参数，其类型为 `char*`。超级用户以外的用户需要具有 `solaris.cluster.modify` 基于角色的访问控制 (role-based access control, RBAC) 授权才能使用该查询标记。有关专用字符串的更多信息，请参见 [clpstring\(1CL\) \[175\]](#) 手册页。

SCHA_RESOURCE_SECURITY

输出参数类型为 `char**`。

该宏返回 `resource_security` 群集属性的当前设置。

SCHA_SOFT_LOADLIMIT

输出参数类型为 `scha_str_array_t**`。

该宏为指定的限制名称返回群集中所有节点的软负载限制值。它需要作为负载限制名称字符串的其他参数，其类型为 `char *`。

字符串数组输出的每个元素的格式均为 `"%s=%d"`，其中，左侧字符串是 `nodename`，右侧整数是该节点上指定的限制名称的软负载限制值。如果没有指定软限制，将显示值 `0` 作为软限制。

SCHA_SYSLOG_FACILITY

输出参数类型为 `int*`。

该宏返回 RGM 用于日志消息的 [Unresolved link to "syslog3C"](#) 设备的编号。返回的值为 24，它对应于 LOG_DAEMON 设备值。

SCHA_ERR_NOERR

该函数成功。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

例 375 使用 `scha_cluster_get()` 函数

以下示例使用 `scha_cluster_get()` 函数获取所有群集节点的名称。该函数还确定节点是处于打开状态还是关闭状态。

该代码示例还显示为每个群集节点（在其中配置该限制）的 `limitname`（名为 `mylimit`）配置的软负载限制和硬负载限制设置。每个节点的负载限制值按以下格式显示：`nodename=softlimit[/hardlimit]`，其中，如果没有设置硬限制，则 `hardlimit` 值是没有限制的 (-1)。

```
#include <scha.h>
#include <stdio.h>
#include <stdlib.h>

main()
{
    scha_err_t          err;
    scha_node_state_t  node_state;
    scha_str_array_t   *all_nodenames;
    scha_cluster_t     handle;
    int                 ix;
    const char         *str;
    scha_str_array_t   *load_limits;

    err = scha_cluster_open(&handle);
    if (err != SCHA_ERR_NOERR) {
        fprintf(stderr, "FAILED: scha_cluster_open()0);
        exit(err);
    }

    err = scha_cluster_get(handle, SCHA_ALL_NODENAMES, &all_nodenames);
    if (err != SCHA_ERR_NOERR) {
        fprintf(stderr, "FAILED: scha_cluster_get()0);
        exit(err);
    }

    for (ix = 0; ix < all_nodenames->array_cnt; ix++) {
        err = scha_cluster_get(handle, SCHA_NODESTATE_NODE,
            all_nodenames->str_array[ix], &node_state);
        if (err != SCHA_ERR_NOERR) {
            fprintf(stderr, "FAILED: scha_cluster_get()"
```

```

        "SCHA_NODESTATE_NODE0);
        exit(err);
    }

    switch (node_state) {
    case SCHA_NODE_UP:
        str = "UP";
        break;
    case SCHA_NODE_DOWN:
        str = "DOWN";
        break;
    }

    printf("State of node: %s value: %s\n",
        all_nodenames->str_array[ix], str);
}
err = scha_cluster_get(handle, SCHA_LOADLIMIT_PROPS, "mylimit",
    &load_limits);

printf("\n\nLoad limits settings for limitname 'mylimit':\n\n");

for (ix = 0; ix < load_limits->array_cnt; ix++) {
    printf("%s\n", load_limits->str_array[ix]);
}
}

```

/usr/cluster/include/scha.h

头文件

/usr/cluster/lib/libscha.so

库

有关以下属性的描述，请参见 [Unresolved link to " attributes5"](#) :

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scha_cluster_get\(1HA\) \[567\]](#)、[scha_calls\(3HA\) \[897\]](#)、[scha_cluster_getlogfacility\(3HA\) \[921\]](#)、[scha](#)
[link to " syslog3C"](#)、[Unresolved link to " attributes5"](#)、[rg_properties\(5\) \[1129\]](#)

名称

`scha_cluster_open`, `scha_cluster_open_zone`, `scha_cluster_get`,
`scha_cluster_get_zone`, `scha_cluster_close` — 访问并获取有关群集的信息

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l scha#include <scha.h>scha_err_t scha_cluster_open(
    scha_cluster_t *handle);

scha_err_t scha_cluster_open_zone(const char *cluster,
    scha_cluster_t *handlep);

scha_err_t scha_cluster_get(scha_cluster_t handle, const char **
    tag, ...);

scha_err_t scha_cluster_get_zone(const char *cluster,
    scha_cluster_t handlep, const char *cluster_tag, ...);

scha_err_t scha_cluster_close(scha_cluster_t handle);
```

将 `scha_cluster_open()`、`scha_cluster_get()` 和 `scha_cluster_close()` 函数一起使用可以获取有关群集的信息。

`scha_cluster_open()` 初始化群集访问，并返回访问句柄以供 `scha_cluster_get()` 使用。`handle` 参数是变量的地址，用于保存函数调用返回的值。

`scha_cluster_get()` 根据 `tag` 参数的指示访问群集信息。`handle` 参数是从对 `scha_cluster_open()` 的之前调用返回的值。`tag` 参数是字符串值，在 `scha_tags.h` 头文件中由宏定义。`tag` 之后的参数取决于 `tag` 参数的值。

您可能需要提供 `tag` 参数后面的其他参数，以指示将从中检索信息的群集节点。参数列表中的最后一个参数将成为适用于保存 `tag` 参数指示的信息的类型。该参数是群集信息的输出参数。如果函数失败，不会为输出参数返回值。对用于 `scha_cluster_get()` 函数的句柄调用 `scha_cluster_close()` 之前，所分配的用于保存 `scha_cluster_get()` 函数返回的信息的内存将保持不变。

`scha_cluster_close()` 将获取从对 `scha_cluster_get()` 函数的之前调用返回的 `handle` 参数。该函数使句柄无效，并将用来返回值而分配的内存释放给使用句柄进行的 `scha_cluster_get()` 调用。请注意，如果需要返回值，则为每个 `get` 调用分配内存。在一个调用中为返回值而分配的空间不会被后续调用覆盖和重新使用。

`scha_cluster_open_zone()` 和 `scha_cluster_get_zone()` 函数的作用分别与 `scha_cluster_open()` 和 `scha_cluster_get()` 相同，且带有一个额外的 `cluster` 参数，用于指定资源组存在于其中且您要在其上执行操作的区域群集的名称。当代码在全局区域中执行，但需要在指定区域群集上操作时，这些函数十分有用。无法在某一区域群集中使用它们来访问其他区域群集。

如果 `scha_cluster_open_zone()` 或 `scha_cluster_get_zone()` 的 `cluster` 参数为 `NULL`，则会对从中执行调用的群集执行查询；换句话说，带有 `NULL` 参数的调用分别等效于 `scha_cluster_open()` 或 `scha_cluster_get()`。

要关闭由 `scha_cluster_open_zone()` 返回的句柄，请使用 `scha_cluster_close()`。不需要使用 `cluster` 参数。

可用于 *tag* 参数的宏

在 `scha_tags.h` 中定义的可用作 *tag* 参数的宏如下所示。指示输出参数和任何其他参数的类型。[scha_calls\(3HA\) \[897\]](#) 介绍了结构和 `enum` 类型。

SCHA_ALL_LOADLIMITS

输出参数类型为 `scha_str_array_t**`。

该宏返回在群集中定义的所有 `loadlimit` 名称。

SCHA_ALL_NODEIDS

输出参数类型为 `scha_uint_array_t**`。

该宏返回群集中所有节点的数字节点标识符。

SCHA_ALL_NODENAMES

输出参数类型为 `scha_str_array_t**`。

该宏返回群集中所有节点的名称。

SCHA_ALL_PRIVATELINK_HOSTNAMES

输出参数类型为 `scha_str_array_t**`。

该宏返回所有群集节点的主机名，根据这些主机名在群集互连上找到节点。

SCHA_ALL_PSTRINGS

输出参数类型为 `scha_str_array_t**`。

该宏返回群集中定义的所有专用字符串的名称。有关专用字符串的更多信息，请参见 [clpstring\(1CL\) \[175\]](#) 手册页。

SCHA_ALL_RESOURCEGROUPS

输出参数类型为 `scha_str_array_t**`。

该宏返回在群集上被管理的所有资源组的名称。

SCHA_ALL_RESOURCETYPES

输出参数类型为 `scha_str_array_t**`。

该宏返回在群集上注册的所有资源类型的名称。

SCHA_CLUSTERNAME

输出参数类型为 `char**`。

该宏返回群集的名称。

SCHA_HARD_LOADLIMIT

输出参数类型为 `scha_str_array_t**`。

该宏为指定的限制名称返回群集中所有节点的硬负载限制值。它需要作为负载限制名称字符串的其他参数，其类型为 `char *`。

字符串数组输出的每个元素的格式均为 `"%s=%d"`，其中，左侧字符串是 *nodename*，右侧整数是该节点上指定的限制名称的硬负载限制值。如果没有指定硬限制，将显示值 `-1` 作为硬限制。

SCHA_LOADLIMIT_PROPS

输出参数类型为 `scha_str_array_t**`。

该宏根据指定的限制名称为群集中的所有节点返回硬负载和软负载限制值（以 / 分隔）。它需要作为负载限制名称字符串的其他参数，其类型为 `char *`。

字符串数组输出的每个元素都是 `"%s=%d/%d"` 格式的字符串，其中，左侧的字符串是 *nodename*，第一个整数是软限制，第二个整数是硬限制。如果没有指定硬限制，将显示值 `-1` 作为硬限制。如果没有指定软限制，将显示值 `0` 作为软限制。

SCHA_LOADLIMITS_NODE

输出参数类型为 `scha_str_array_t**`。

该宏为特定节点返回负载限制（用 / 分隔）和限制名称。它需要类型为 `char *` 的 *nodename* 的附加参数。

字符串数组输出的每个元素都是 `"%s=%d/%d"` 格式的字符串，其中，字符串是指定节点上定义的限制名称，第一个整数是软限制值，第二个整数是硬限制值。如果没有指定硬限制，将显示值 `-1` 作为硬限制。如果没有指定软限制，将显示值 `0` 作为软限制。

SCHA_NODEID_LOCAL

输出参数类型为 `uint_t*`。

该宏返回在其中执行命令的节点的数字节点标识符。

SCHA_NODEID_NODENAME

输出参数类型为 `uint_t*`。附加参数的类型为 `char *`。该宏需要一个作为群集节点名称的附加参数。

该宏返回由名称指示的节点的数字节点标识符。

SCHA_NODENAME_LOCAL

输出参数类型为 `char**`。

该宏返回在其中执行函数的群集节点的名称。

SCHA_NODENAME_NODEID

输出参数类型为 `char**`。附加参数的类型为 `uint_t`。该附加参数是数字群集节点标识符。

该宏返回由数字标识符指示的节点的名称。

SCHA_NODESTATE_LOCAL

输出参数类型为 `scha_node_state_t*`。

该宏返回 `SCHA_NODE_UP` 或 `SCHA_NODE_DOWN`，具体取决于在其中执行命令的节点的状态。

SCHA_NODESTATE_NODE

输出参数类型为 `scha_node_state_t*`。附加参数的类型为 `char*`。该宏需要作为群集节点名称的其他未标记参数。

根据命名节点的状态，该宏返回 `SCHA_NODE_UP` 或 `SCHA_NODE_DOWN`。

SCHA_PRIVATELINK_HOSTNAME_LOCAL

输出参数类型为 `char**`。

该宏返回主机名，根据该主机名在群集互连上找到在其上运行命令的节点。

SCHA_PRIVATELINK_HOSTNAME_NODE

输出参数类型为 `char**`。附加参数的类型为 `char *`。该宏需要作为群集节点名称的其他未标记参数。

该宏返回主机名，根据该主机名在群集互连上找到命名的节点。

SCHA_PSTRING

输出参数类型为 `char**`。

该宏返回专用字符串的明文值。它需要提供专用字符串名称的附加参数，其类型为 `char*`。超级用户以外的用户需要具有 `solaris.cluster.modify` 基于角色的访问控制 (role-based access control, RBAC) 授权才能使用该查询标记。有关专用字符串的更多信息，请参见 [clpstring\(1CL\) \[175\]](#) 手册页。

SCHA_RESOURCE_SECURITY

输出参数类型为 `char**`。

该宏返回 `resource_security` 群集属性的当前设置。

SCHA_SOFT_LOADLIMIT

输出参数类型为 `scha_str_array_t**`。

该宏为指定的限制名称返回群集中所有节点的软负载限制值。它需要作为负载限制名称字符串的其他参数，其类型为 `char *`。

字符串数组输出的每个元素的格式均为 `"%s=%d"`，其中，左侧字符串是 `nodename`，右侧整数是该节点上指定的限制名称的软负载限制值。如果没有指定软限制，将显示值 `0` 作为软限制。

SCHA_SYSLOG_FACILITY

输出参数类型为 `int*`。

该宏返回 RGM 用于日志消息的 [Unresolved link to "syslog3C"](#) 设备的编号。返回的值为 24，它对应于 LOG_DAEMON 设备值。

SCHA_ERR_NOERR

该函数成功。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

例 376 使用 `scha_cluster_get()` 函数

以下示例使用 `scha_cluster_get()` 函数获取所有群集节点的名称。该函数还确定节点是处于打开状态还是关闭状态。

该代码示例还显示为每个群集节点（在其中配置该限制）的 `limitname`（名为 `mylimit`）配置的软负载限制和硬负载限制设置。每个节点的负载限制值按以下格式显示：`nodename=softlimit[/hardlimit]`，其中，如果没有设置硬限制，则 `hardlimit` 值是没有限制的 (-1)。

```
#include <scha.h>
#include <stdio.h>
#include <stdlib.h>

main()
{
    scha_err_t          err;
    scha_node_state_t  node_state;
    scha_str_array_t   *all_nodenames;
    scha_cluster_t     handle;
    int                 ix;
    const char         *str;
    scha_str_array_t   *load_limits;

    err = scha_cluster_open(&handle);
    if (err != SCHA_ERR_NOERR) {
        fprintf(stderr, "FAILED: scha_cluster_open()0);
        exit(err);
    }

    err = scha_cluster_get(handle, SCHA_ALL_NODENAMES, &all_nodenames);
    if (err != SCHA_ERR_NOERR) {
        fprintf(stderr, "FAILED: scha_cluster_get()0);
        exit(err);
    }

    for (ix = 0; ix < all_nodenames->array_cnt; ix++) {
        err = scha_cluster_get(handle, SCHA_NODESTATE_NODE,
            all_nodenames->str_array[ix], &node_state);
        if (err != SCHA_ERR_NOERR) {
            fprintf(stderr, "FAILED: scha_cluster_get()"
```

```

        "SCHA_NODESTATE_NODE0);
        exit(err);
    }

    switch (node_state) {
    case SCHA_NODE_UP:
        str = "UP";
        break;
    case SCHA_NODE_DOWN:
        str = "DOWN";
        break;
    }

    printf("State of node: %s value: %s\n",
        all_nodenames->str_array[ix], str);
}
err = scha_cluster_get(handle, SCHA_LOADLIMIT_PROPS, "mylimit",
    &load_limits);

printf("\n\nLoad limits settings for limitname 'mylimit':\n\n");

for (ix = 0; ix < load_limits->array_cnt; ix++) {
    printf("%s\n", load_limits->str_array[ix]);
}
}

```

/usr/cluster/include/scha.h

头文件

/usr/cluster/lib/libscha.so

库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scha_cluster_get\(1HA\) \[567\]](#)、[scha_calls\(3HA\) \[897\]](#)、[scha_cluster_getlogfacility\(3HA\) \[921\]](#)、[scha_cluster_getlogfacility\(3HA\) \[921\]](#)、[scha_cluster_getlogfacility\(3HA\) \[921\]](#)、[scha_cluster_getlogfacility\(3HA\) \[921\]](#)、[link to "syslog3C"](#)、[Unresolved link to "attributes5"](#)、[rg_properties\(5\) \[1129\]](#)

名称

scha_control, scha_control_zone — 资源和资源组控制请求函数

```
cc [flags...] -I/usr/cluster/include file -L/usr/cluster/lib
-l scha#include <scha.h> scha_err_t scha_control(const char *
tag, const char *rgname, const char *rname);

scha_err_t scha_control_zone(const char *tag, const char *
rgname, const char *rname, const char *zonename);
```

scha_control () 和 scha_control_zone () 函数均提供一个接口，用于请求重新启动或重定位受资源组管理器 (Resource Group Manager, RGM) 控制的资源或资源组。请在资源监视器中使用这些函数。

请仅对 Global_zone 属性设置为 TRUE 的资源类型使用 scha_control_zone () 函数。如果 Global_zone 属性设置为 FALSE，则不需要此函数。有关更多信息，请参见 [rt_properties\(5\) \[1143\]](#) 手册页。在全局区域调用 scha_control_zone () 函数。zonename 参数指定在其中配置资源组的区域群集的名称。

所指定的资源的 Failover_mode 属性设置可能会抑制所请求的 scha_control () 或 scha_control_zone () 操作。如果 Failover_mode 为 RESTART_ONLY，则只允许 SCHA_RESOURCE_RESTART。其他请求（包括 SCHA_GIVEOVER、SCHA_CHECK_GIVEOVER、SCHA_RESTART 和 SCHA_CHECK_RESTART）将返回 SCHA_ERR_CHECKS 退出代码，并且所请求的移交或重新启动操作不会执行，而只生成 syslog 消息。如果在资源上设置了 Retry_count 和 Retry_interval 属性，则资源重新启动次数限制为 Retry_interval 中的 Retry_count 尝试次数。如果 Failover_mode 为 LOG_ONLY，则任何 scha_control () 或 scha_control_zone () 移交、重新启动或禁用请求将返回 SCHA_ERR_CHECKS 退出代码，并且所请求的移交或重新启动操作不会执行，而只生成 syslog 消息。

tag 参数

tag 参数指明是请求重新启动还是重定位资源或资源组。该参数应当是一个字符串值，该值由 scha_tags.h 中定义的以下宏之一定义：

SCHA_CHANGE_STATE_OFFLINE

请求使 rname 参数指定的代理资源在本地节点上脱机。代理资源是一种 Oracle Solaris Cluster 资源，它可以从另一群集（例如 Oracle Clusterware）导入某个资源的状态。Oracle Clusterware 是一组适用于群集环境的不依赖于平台的系统服务。在 Oracle Solaris Cluster 软件的上下文中，这种状态变化反映了外部资源的状态变化。

当您使用该 tag 参数更改某个代理资源的状态时，不会执行该代理资源的方法。

如果节点上的“被依赖”资源发生故障，并且该资源无法恢复，则监视器将使该资源在该节点上脱机。监视器将会通过随 SCHA_RESOURCE_DISABLE 请求调用

`scha_control ()` 或 `scha_control_zone ()` 函数来使该资源脱机。监视器还会通过触发重新启动操作，使被依赖资源的所有 `offline-restart` 依赖项脱机。当群集管理员解决故障并重新启用被依赖资源时，监视器还会使被依赖资源的 `offline-restart` 依赖资源重新联机。

SCHA_CHANGE_STATE_ONLINE

请求使 `rname` 参数指定的代理资源在本地节点上联机。代理资源是一种 Oracle Solaris Cluster 资源，它可以从另一群集（例如 Oracle Clusterware）导入某个资源的状态。在 Oracle Solaris Cluster 软件的上下文中，这种状态变化反映了外部资源的状态变化。

当您使用该 `tag` 参数更改某个代理资源的状态时，不会执行该代理资源的方法。

SCHA_CHECK_GIVEOVER

对 `rgname` 参数指定的资源组执行与 `SCHA_GIVEOVER` 相同的有效性检查，但不实际重定位该资源组。

SCHA_CHECK_RESTART

对 `rgname` 参数指定的资源组执行与 `SCHA_RESTART` 请求相同的有效性检查，但不实际重新启动该资源组。

`SCHA_CHECK_GIVEOVER` 和 `SCHA_CHECK_RESTART` 请求用于供资源监视器直接对资源采取操作（例如，终止并重新启动进程），而不是通过调用 `scha_control ()` 或 `scha_control_zone ()` 函数来执行移交或重新启动。如果检查失败，监视器应该休眠，然后重新启动其探测器，而不是调用其故障转移操作。请参见 错误。

`rgname` 参数是要重新启动或重定位的资源组的名称。如果该组未在发出请求的节点上联机，则请求会被拒绝。

`rname` 参数是资源组中某个资源的名称。该资源很可能是其监视器正在发出 `scha_control ()` 或 `scha_control_zone ()` 请求的资源。如果指定的资源不在资源组中，则请求会被拒绝。

该命令的退出代码指示是否拒绝了所请求的操作。如果接受了请求，则在资源组或资源已完成脱机并已重新联机之前，该函数不会返回。调用 `scha_control ()` 或 `scha_control_zone ()` 函数的故障监视器可能会由于资源组脱机而停止，因此，可能永远无法收到成功的请求的返回状态。

SCHA_GIVEOVER

请求使 `rgname` 参数指定的资源组在本地节点上脱机，然后再在 RGM 选择的其他节点上重新联机。请注意，如果该资源组当前已在两个或更多个节点上联机，并且没有其他可用节点可供该资源组在其上联机，则可使它在本地节点上脱机且无需再让它在其他位置联机。请求可能会被拒绝，具体取决于各项检查的结果。例如，如果组是由于在 `Pingpong_interval` 属性指定的时间间隔内某个节点上发出的 `SCHA_GIVEOVER` 请求而脱机，则可能会拒绝将该节点用作主机。

如果群集管理员配置了一个或多个资源组的 `RG_affinities` 属性，而您对一个资源组发出了 `scha_control GIVEOVER` 请求，那么，可能会重定位多个资源组。[rg_properties\(5\) \[1129\]](#) 中介绍了 `RG_affinities` 属性。

通过故障监视器调用 `scha_control()` 或 `scha_control_zone()` 函数或者发出 `scha_control` 或 `scha_control_zone()` 命令后，将包含该资源的资源组重定位到新节点之前，将会调用 `MONITOR_CHECK` 方法。请参见 [scha_control\(1HA\) \[579\]](#) 手册页。

您可以在资源组潜在的任何新主节点上调用 `MONITOR_CHECK` 方法。`MONITOR_CHECK` 方法旨在评估某个节点的运行状况是否良好，足以运行某个资源。`MONITOR_CHECK` 方法必须以这样一种方式实现，即它不与同时运行的其他方法相冲突。

如果 `MONITOR_CHECK` 方法失败，则会禁止将资源组重定位到调用了回调的节点。

`SCHA_IGNORE_FAILED_START`

请求在当前执行的 `Prenet_start` 或 `Start` 方法失败时不对资源组进行故障转移，无论 `Failover_mode` 属性的设置如何。

换言之，该请求将覆盖当 `Failover_Mode` 属性设置为 `SOFT` 或 `HARD` 的资源无法启动时通常情况下会采取的恢复操作。通常，资源组会故障转移到另一个节点。而资源的行为却如同 `Failover_Mode` 设置为 `NONE` 一样。如果未发生其他错误，资源将进入 `START_FAILED` 状态，而资源组最终将进入 `ONLINE_FAULTED` 状态。

仅当通过 `Start` 或 `Prenet_start` 方法（该方法随后会退出并返回非零状态，或者超时）调用时，该请求才有意义。该请求仅对 `Start` 或 `Prenet_start` 方法的当前调用有效。在 `Start` 方法确定资源在其他节点上无法成功启动的情况下，应随该请求调用 `scha_control()` 或 `scha_control_zone()` 函数。如果由其他任何方法调用该请求，将返回错误 `SCHA_ERR_INVALID`。该请求可防止资源组发生不使用该请求时会发生的“弹跳”故障转移。有关 `SCHA_ERR_INVALID` 错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#) 手册页。

`SCHA_RESOURCE_DISABLE`

在调用 `scha_control()` 或 `scha_control_zone()` 函数的节点上禁用 `rname` 参数指定的资源。

如果节点上的“被依赖”资源发生故障，并且该资源无法恢复，则监视器将使该资源在该节点上脱机。监视器将会通过随 `SCHA_RESOURCE_DISABLE` 请求调用 `scha_control()` 或 `scha_control_zone()` 函数来使该资源脱机。监视器还会通过触发重新启动操作，使被依赖资源的所有 `offline-restart` 依赖项脱机。当群集管理员解决故障并重新启用被依赖资源时，监视器还会使被依赖资源的 `offline-restart` 依赖资源重新联机。

`SCHA_RESOURCE_IS_RESTARTED`

请求在本地节点上递增 `rname` 参数指定的资源的资源重新启动计数器，但不实际重新启动该资源。

没有在 `SCHA_RESOURCE_RESTART` 请求中调用 `scha_control()` 或 `scha_control_zone()` 函数而直接重新启动资源（例如，使用 [pmfadm\(1M\) \[625\]](#) 命令）的资源监视器可以使用该请求来通知 RGM 该资源已重新启动。随后使用 `NUM_RESOURCE_RESTARTS` 查询调用 `scha_resource_get()` 函数时会反映该事实。

如果资源的类型无法声明 `Retry_interval` 标准属性，则不允许 `scha_control ()` 或 `scha_control_zone ()` 函数的 `SCHA_RESOURCE_IS_RESTARTED` 请求，并且 `scha_control ()` 或 `scha_control_zone ()` 函数将返回错误代码 13 (`SCHA_ERR_RT`)。

SCHA_RESOURCE_RESTART

请求使 `rname` 参数指定的资源在本地节点上脱机然后重新联机，而无需停止该资源组中的任何其他资源。这将通过在本地节点上对该资源应用以下方法序列来使其停止后再启动：

```
MONITOR_STOP
STOP
START
MONITOR_START
```

如果资源类型没有声明 `STOP` 和 `START` 方法，则会改用 `POSTNET_STOP` 和 `PRENET_START` 来重新启动该资源：

```
MONITOR_STOP
POSTNET_STOP
PRENET_START
MONITOR_START
```

如果资源的类型没有声明 `MONITOR_STOP` 和 `MONITOR_START` 方法，则只会调用 `STOP` 和 `START` 方法或者 `POSTNET_STOP` 和 `PRENET_START` 方法来执行重新启动。资源的类型必须声明 `START` 和 `STOP` 方法。有关 `SCHA_ERR_RT` 错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#) 手册页。

如果重新启动资源时某个方法调用失败，RGM 可能会设置一个错误状态、重定位资源组，或者重新引导节点，具体取决于对该资源的 `Failover_mode` 属性的设置。有关其他信息，请参见 [r_properties\(5\) \[1103\]](#) 中的 `Failover_mode` 属性。

使用该请求重新启动资源的资源监视器可以使用 `scha_resource_get ()` 的 `NUM_RESOURCE_RESTARTS` 查询来统计最近的重新启动尝试次数。

具有 `PRENET_START` 或 `POSTNET_STOP` 方法的资源类型需慎用 `SCHA_RESOURCE_RESTART` 请求。只会对此类资源应用 `MONITOR_STOP`、`STOP`、`START` 和 `MONITOR_START` 方法。该资源所依赖的网络地址资源不会重新启动，将保持联机状态。

如果节点上的“被依赖”资源发生故障，并且该资源无法恢复，则监视器将使该资源在该节点上脱机。监视器将会通过随 `SCHA_RESOURCE_DISABLE` 请求调用 `scha_control ()` 或 `scha_control_zone ()` 函数来使该资源脱机。监视器还会通过触发重新启动操作，使被依赖资源的所有 `offline-restart` 依赖项脱机。当群集管理员解决故障并重新启用被依赖资源时，监视器还会使被依赖资源的 `offline-restart` 依赖资源重新联机。

SCHA_RESTART

请求使 `rgname` 参数指定的资源组脱机，然后再联机，而无需强制重定位到其他节点。如果该资源组中的某个资源无法重新启动，该请求最终可能会导致重定位

该组。使用该请求重新启动资源组的资源监视器可以使用 `scha_resource_get` 的 `NUM_RG_RESTARTS` () 查询来统计最近的重新启动尝试次数。

这些函数返回以下值：

0 该函数成功。

非零值 该函数失败。

`SCHA_ERR_NOERR` 该函数成功。

`SCHA_ERR_CHECKS` 请求被拒绝。对重定位执行的检查失败。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#) 手册页。

通常，从 `scha_control` () 或 `scha_control_zone` () 函数接收到错误代码的故障监视器应休眠片刻，然后重新启动其探测器。这些函数必须这样做，因为经过片刻后，某些错误状态可以自行修复。这种错误状况的一个示例是某个全局设备服务发生了故障转移，导致磁盘资源暂时不可用。在错误状况修复后，资源本身的状况将恢复正常。否则，后续的 `scha_control` () 或 `scha_control_zone` () 请求可能会成功。

`/usr/cluster/include/scha.h` 头文件

`/usr/cluster/lib/libscha.so` 库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[rt_callbacks\(1HA\) \[545\]](#)、[scha_control\(1HA\) \[579\]](#)、[pmfadm\(1M\) \[625\]](#)、[scha_calls\(3HA\) \[897\]](#)、[scha_res](#)
[link to "](#)
[attributes5"](#)、[r_properties\(5\) \[1103\]](#)、[rg_properties\(5\) \[1129\]](#)、[rt_properties\(5\) \[1143\]](#)

名称

scha_control, scha_control_zone — 资源和资源组控制请求函数

```
cc [flags...] -I/usr/cluster/include file -L/usr/cluster/lib
-l scha#include <scha.h> scha_err_t scha_control(const char *
tag, const char *rgname, const char *rname);

scha_err_t scha_control_zone(const char *tag, const char *
rgname, const char *rname, const char *zonename);
```

scha_control () 和 scha_control_zone () 函数均提供一个接口，用于请求重新启动或重定位受资源组管理器 (Resource Group Manager, RGM) 控制的资源或资源组。请在资源监视器中使用这些函数。

请仅对 Global_zone 属性设置为 TRUE 的资源类型使用 scha_control_zone () 函数。如果 Global_zone 属性设置为 FALSE，则不需要此函数。有关更多信息，请参见 [rt_properties\(5\) \[1143\]](#) 手册页。在全局区域调用 scha_control_zone () 函数。zonename 参数指定在其中配置资源组的区域群集的名称。

所指定的资源的 Failover_mode 属性设置可能会抑制所请求的 scha_control () 或 scha_control_zone () 操作。如果 Failover_mode 为 RESTART_ONLY，则只允许 SCHA_RESOURCE_RESTART。其他请求（包括 SCHA_GIVEOVER、SCHA_CHECK_GIVEOVER、SCHA_RESTART 和 SCHA_CHECK_RESTART）将返回 SCHA_ERR_CHECKS 退出代码，并且所请求的移交或重新启动操作不会执行，而只生成 syslog 消息。如果在资源上设置了 Retry_count 和 Retry_interval 属性，则资源重新启动次数限制为 Retry_interval 中的 Retry_count 尝试次数。如果 Failover_mode 为 LOG_ONLY，则任何 scha_control () 或 scha_control_zone () 移交、重新启动或禁用请求将返回 SCHA_ERR_CHECKS 退出代码，并且所请求的移交或重新启动操作不会执行，而只生成 syslog 消息。

tag 参数

tag 参数指明是请求重新启动还是重定位资源或资源组。该参数应当是一个字符串值，该值由 scha_tags.h 中定义的以下宏之一定义：

SCHA_CHANGE_STATE_OFFLINE

请求使 rname 参数指定的代理资源在本地节点上脱机。代理资源是一种 Oracle Solaris Cluster 资源，它可以从另一群集（例如 Oracle Clusterware）导入某个资源的状态。Oracle Clusterware 是一组适用于群集环境的不依赖于平台的系统服务。在 Oracle Solaris Cluster 软件的上下文中，这种状态变化反映了外部资源的状态变化。

当您使用该 tag 参数更改某个代理资源的状态时，不会执行该代理资源的方法。

如果节点上的“被依赖”资源发生故障，并且该资源无法恢复，则监视器将使该资源在该节点上脱机。监视器将会通过随 SCHA_RESOURCE_DISABLE 请求调用

`scha_control ()` 或 `scha_control_zone ()` 函数来使该资源脱机。监视器还会通过触发重新启动操作，使被依赖资源的所有 `offline-restart` 依赖项脱机。当群集管理员解决故障并重新启用被依赖资源时，监视器还会使被依赖资源的 `offline-restart` 依赖资源重新联机。

SCHA_CHANGE_STATE_ONLINE

请求使 `rname` 参数指定的代理资源在本地节点上联机。代理资源是一种 Oracle Solaris Cluster 资源，它可以从另一群集（例如 Oracle Clusterware）导入某个资源的状态。在 Oracle Solaris Cluster 软件的上下文中，这种状态变化反映了外部资源的状态变化。

当您使用该 `tag` 参数更改某个代理资源的状态时，不会执行该代理资源的方法。

SCHA_CHECK_GIVEOVER

对 `rgname` 参数指定的资源组执行与 `SCHA_GIVEOVER` 相同的有效性检查，但不实际重定位该资源组。

SCHA_CHECK_RESTART

对 `rgname` 参数指定的资源组执行与 `SCHA_RESTART` 请求相同的有效性检查，但不实际重新启动该资源组。

`SCHA_CHECK_GIVEOVER` 和 `SCHA_CHECK_RESTART` 请求用于供资源监视器直接对资源采取操作（例如，终止并重新启动进程），而不是通过调用 `scha_control ()` 或 `scha_control_zone ()` 函数来执行移交或重新启动。如果检查失败，监视器应该休眠，然后重新启动其探测器，而不是调用其故障转移操作。请参见 错误。

`rgname` 参数是要重新启动或重定位的资源组的名称。如果该组未在发出请求的节点上联机，则请求会被拒绝。

`rname` 参数是资源组中某个资源的名称。该资源很可能是其监视器正在发出 `scha_control ()` 或 `scha_control_zone ()` 请求的资源。如果指定的资源不在资源组中，则请求会被拒绝。

该命令的退出代码指示是否拒绝了所请求的操作。如果接受了请求，则在资源组或资源已完成脱机并已重新联机之前，该函数不会返回。调用 `scha_control ()` 或 `scha_control_zone ()` 函数的故障监视器可能会由于资源组脱机而停止，因此，可能永远无法收到成功的请求的返回状态。

SCHA_GIVEOVER

请求使 `rgname` 参数指定的资源组在本地节点上脱机，然后再在 RGM 选择的其他节点上重新联机。请注意，如果该资源组当前已在两个或更多个节点上联机，并且没有其他可用节点可供该资源组在其上联机，则可使它在本地节点上脱机且无需再让它在其他位置联机。请求可能会被拒绝，具体取决于各项检查的结果。例如，如果组是由于在 `Pingpong_interval` 属性指定的时间间隔内某个节点上发出的 `SCHA_GIVEOVER` 请求而脱机，则可能会拒绝将该节点用作主机。

如果群集管理员配置了一个或多个资源组的 `RG_affinities` 属性，而您对一个资源组发出了 `scha_control GIVEOVER` 请求，那么，可能会重定位多个资源组。[rg_properties\(5\) \[1129\]](#) 中介绍了 `RG_affinities` 属性。

通过故障监视器调用 `scha_control()` 或 `scha_control_zone()` 函数或者发出 `scha_control` 或 `scha_control_zone()` 命令后，将包含该资源的资源组重定位到新节点之前，将会调用 `MONITOR_CHECK` 方法。请参见 [scha_control\(1HA\) \[579\]](#) 手册页。

您可以在资源组潜在的任何新主节点上调用 `MONITOR_CHECK` 方法。`MONITOR_CHECK` 方法旨在评估某个节点的运行状况是否良好，足以运行某个资源。`MONITOR_CHECK` 方法必须以这样一种方式实现，即它不与同时运行的其他方法相冲突。

如果 `MONITOR_CHECK` 方法失败，则会禁止将资源组重定位到调用了回调的节点。

SCHA_IGNORE_FAILED_START

请求在当前执行的 `Prenet_start` 或 `Start` 方法失败时不对资源组进行故障转移，无论 `Failover_mode` 属性的设置如何。

换言之，该请求将覆盖当 `Failover_Mode` 属性设置为 `SOFT` 或 `HARD` 的资源无法启动时通常情况下会采取的恢复操作。通常，资源组会故障转移到另一个节点。而资源的行为却如同 `Failover_Mode` 设置为 `NONE` 一样。如果未发生其他错误，资源将进入 `START_FAILED` 状态，而资源组最终将进入 `ONLINE_FAULTED` 状态。

仅当通过 `Start` 或 `Prenet_start` 方法（该方法随后会退出并返回非零状态，或者超时）调用时，该请求才有意义。该请求仅对 `Start` 或 `Prenet_start` 方法的当前调用有效。在 `Start` 方法确定资源在其他节点上无法成功启动的情况下，应随该请求调用 `scha_control()` 或 `scha_control_zone()` 函数。如果由其他任何方法调用该请求，将返回错误 `SCHA_ERR_INVALID`。该请求可防止资源组发生不使用该请求时会发生的“弹跳”故障转移。有关 `SCHA_ERR_INVALID` 错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#) 手册页。

SCHA_RESOURCE_DISABLE

在调用 `scha_control()` 或 `scha_control_zone()` 函数的节点上禁用 `rname` 参数指定的资源。

如果节点上的“被依赖”资源发生故障，并且该资源无法恢复，则监视器将使该资源在该节点上脱机。监视器将会通过随 `SCHA_RESOURCE_DISABLE` 请求调用 `scha_control()` 或 `scha_control_zone()` 函数来使该资源脱机。监视器还会通过触发重新启动操作，使被依赖资源的所有 `offline-restart` 依赖项脱机。当群集管理员解决故障并重新启用被依赖资源时，监视器还会使被依赖资源的 `offline-restart` 依赖资源重新联机。

SCHA_RESOURCE_IS_RESTARTED

请求在本地节点上递增 `rname` 参数指定的资源的资源重新启动计数器，但不实际重新启动该资源。

没有在 `SCHA_RESOURCE_RESTART` 请求中调用 `scha_control()` 或 `scha_control_zone()` 函数而直接重新启动资源（例如，使用 [pmfadm\(1M\) \[625\]](#) 命令）的资源监视器可以使用该请求来通知 RGM 该资源已重新启动。随后使用 `NUM_RESOURCE_RESTARTS` 查询调用 `scha_resource_get()` 函数时会反映该事实。

如果资源的类型无法声明 `Retry_interval` 标准属性，则不允许 `scha_control ()` 或 `scha_control_zone ()` 函数的 `SCHA_RESOURCE_IS_RESTARTED` 请求，并且 `scha_control ()` 或 `scha_control_zone ()` 函数将返回错误代码 13 (`SCHA_ERR_RT`)。

SCHA_RESOURCE_RESTART

请求使 `rname` 参数指定的资源在本地节点上脱机然后重新联机，而无需停止该资源组中的任何其他资源。这将通过在本地节点上对该资源应用以下方法序列来使其停止后再启动：

```
MONITOR_STOP
STOP
START
MONITOR_START
```

如果资源类型没有声明 `STOP` 和 `START` 方法，则会改用 `POSTNET_STOP` 和 `PRENET_START` 来重新启动该资源：

```
MONITOR_STOP
POSTNET_STOP
PRENET_START
MONITOR_START
```

如果资源的类型没有声明 `MONITOR_STOP` 和 `MONITOR_START` 方法，则只会调用 `STOP` 和 `START` 方法或者 `POSTNET_STOP` 和 `PRENET_START` 方法来执行重新启动。资源的类型必须声明 `START` 和 `STOP` 方法。有关 `SCHA_ERR_RT` 错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#) 手册页。

如果重新启动资源时某个方法调用失败，RGM 可能会设置一个错误状态、重定位资源组，或者重新引导节点，具体取决于对该资源的 `Failover_mode` 属性的设置。有关其他信息，请参见 [r_properties\(5\) \[1103\]](#) 中的 `Failover_mode` 属性。

使用该请求重新启动资源的资源监视器可以使用 `scha_resource_get ()` 的 `NUM_RESOURCE_RESTARTS` 查询来统计最近的重新启动尝试次数。

具有 `PRENET_START` 或 `POSTNET_STOP` 方法的资源类型需慎用 `SCHA_RESOURCE_RESTART` 请求。只会对此类资源应用 `MONITOR_STOP`、`STOP`、`START` 和 `MONITOR_START` 方法。该资源所依赖的网络地址资源不会重新启动，将保持联机状态。

如果节点上的“被依赖”资源发生故障，并且该资源无法恢复，则监视器将使该资源在该节点上脱机。监视器将会通过随 `SCHA_RESOURCE_DISABLE` 请求调用 `scha_control ()` 或 `scha_control_zone ()` 函数来使该资源脱机。监视器还会通过触发重新启动操作，使被依赖资源的所有 `offline-restart` 依赖项脱机。当群集管理员解决故障并重新启用被依赖资源时，监视器还会使被依赖资源的 `offline-restart` 依赖资源重新联机。

SCHA_RESTART

请求使 `rgname` 参数指定的资源组脱机，然后再联机，而无需强制重定位到其他节点。如果该资源组中的某个资源无法重新启动，该请求最终可能会导致重定位

该组。使用该请求重新启动资源组的资源监视器可以使用 `scha_resource_get` 的 `NUM_RG_RESTARTS` () 查询来统计最近的重新启动尝试次数。

这些函数返回以下值：

0 该函数成功。

非零值 该函数失败。

`SCHA_ERR_NOERR` 该函数成功。

`SCHA_ERR_CHECKS` 请求被拒绝。对重定位执行的检查失败。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#) 手册页。

通常，从 `scha_control` () 或 `scha_control_zone` () 函数接收到错误代码的故障监视器应休眠片刻，然后重新启动其探测器。这些函数必须这样做，因为经过片刻后，某些错误状态可以自行修复。这种错误状况的一个示例是某个全局设备服务发生了故障转移，导致磁盘资源暂时不可用。在错误状况修复后，资源本身的状况将恢复正常。否则，后续的 `scha_control` () 或 `scha_control_zone` () 请求可能会成功。

`/usr/cluster/include/scha.h` 头文件

`/usr/cluster/lib/libscha.so` 库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[rt_callbacks\(1HA\) \[545\]](#)、[scha_control\(1HA\) \[579\]](#)、[pmfadm\(1M\) \[625\]](#)、[scha_calls\(3HA\) \[897\]](#)、[scha_res](#)
[link to "](#)
[attributes5"](#)、[r_properties\(5\) \[1103\]](#)、[rg_properties\(5\) \[1129\]](#)、[rt_properties\(5\) \[1143\]](#)

名称

scha_get_fullname — 返回执行资源组的群集节点名称

```
scha_err_tscha_get_fullname(const char *zonename, char **fullname)
```

scha_get_fullname () 函数返回本地节点上区域上下文的逻辑节点名称。输出参数 fullname 设置为包含群集节点名称的字符串。该字符串需要由调用方释放。

如果 zonename 为 NULL，则区域上下文取决于执行该调用的位置。

- 如果在全局区域中执行，则返回本地节点名称。
- 如果在区域群集中执行，则返回区域群集节点名称，而不是全局群集节点名称。

如果 zonename 为非 NULL 值，则意味着要在全局区域中代表 zonename 指定的非全局区域中配置的资源调用该函数。如果 zonename 是区域群集的名称，同时也是底层非全局区域的名称，将返回区域群集节点名称。

如果该函数通过资源回调方法调用，并且 zonename 参数设置为 scds_get_zone_name () 函数的输出，则在所有情况下，生成的 fullname 都将与资源组节点列表中当前的条目匹配：

- 全局群集或区域群集
- global_zone 资源类型或常规资源类型

此函数会返回以下值：

0 该函数成功。

非零值 该函数失败。

SCHA_ERR_NOERR 该函数成功。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

例 377 使用 scha_get_fullname () 函数

以下示例使用 scha_get_fullname() 获取表示执行当前程序的 nodename 上下文的 fullname 字符串，并在资源组的节点列表中搜索该名称：

```
#include <scha.h>
#include <libdsdev.h>
```

```

main(int argc, char *argv[])
{
    scha_err_t      err;
    scds_handle_t   handle;
    char            *myzonename;
    char            *fullname;
    const scha_str_array_t *rgnodelist;
    uint_t          ix;

    if (scds_initialize(&handle) != SCHA_ERR_NOERR) {
        /* handle the error */
        ...
    }
    myzonename = scds_get_zone_name(handle);
    rgnodelist = scds_get_rg_nodelist(handle);
    err = scha_get_fullname(myzonename, &fullname);
    ...
    for (ix = 0; ix < rgnodelist->array_cnt; ix++) {
        if (strcmp(fullname, rgnodelist->str_array[ix]) == 0) {
            /* found this node in the node list */
            ...
        }
    }
    ...
}

```

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scds_calls\(3HA\) \[771\]](#)、[scds_initialize\(3HA\) \[847\]](#)、[scha_cluster_getnodename\(3HA\) \[923\]](#)、[Unresolved link to "attributes5"](#)

名称

scha_resource_open, scha_resource_open_zone, scha_resource_get,
scha_resource_get_zone, scha_resource_close — 资源信息访问函数

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib -l scha#include
<scha.h>scha_err_t scha_resource_open( const char *rname, const char *rgname, scha_resource_t
*handle);

scha_err_t scha_resource_open_zone(const char *cluster, const char *rs_name, const char
*rg_name, scha_resource_t * handlep);

scha_err_t scha_resource_get(scha_resource_t handle, const char *tag,...);

scha_err_t scha_resource_get_zone(const char *cluster,
    scha_resource_t handlep, const char *rs_tag, ...);

scha_err_t scha_resource_close(scha_resource_t handle);
```

将 `scha_resource_open()`、`scha_resource_get()` 和 `scha_resource_close()` 函数一起使用可以访问资源组管理器 (Resource Group Manager, RGM) 群集工具管理的资源的相关信息。

`scha_resource_open()` 初始化资源访问，并返回一个将由 `scha_resource_get()` 使用的句柄。

`scha_resource_open()` 的 *rname* 参数指定要访问的资源名称。*rgname* 参数是其中配置了该资源的资源组的名称。如果组名称未知，*rgname* 参数可以是 NULL。但是，如果提供组名称，可以提高该函数的执行效率。*handle* 参数是用于保存函数调用返回值的某个变量的地址。

`scha_resource_get()` 根据 *tag* 参数的指示访问资源信息。*tag* 参数应该是 `scha_tags.h` 头文件中的某个宏定义的一个字符串值。标记后面的参数取决于 *tag* 的值。可能需要在标记后面额外添加一个参数，用于指示要从中检索信息的群集节点或者指示特定于标记的其他信息。参数列表中最后一个参数的类型应该适合用于保存 *tag* 指定的信息。该参数是资源信息的输出参数。如果函数失败，不会为输出参数返回值。

对用于 `scha_resource_get()` 的句柄调用 `scha_resource_close()` 之前，所分配的用于保存 `scha_resource_get()` 返回的信息的内存将保持不变。请注意，使用相同的句柄和标记重复调用 `scha_resource_get()` 会导致分配新的内存。在一个调用中为返回值而分配的空间不会被后续调用覆盖和重新使用。

`scha_resource_close()` 函数接受上次调用 `scha_resource_open()` 时返回的 *handle* 参数。它将使该句柄失效，并释放为了保存使用该句柄执行的 `scha_resource_get()` 调用的返回值而分配的内存。

下文中列出了 `scha_tags.h` 中定义的可以用作 `scha_resource_get()` 的 *tag* 参数的宏。

指示输出参数和任何其他参数的类型。[scha_calls\(3HA\) \[897\]](#) 介绍了结构和 enum 类型。

`scha_resource_open_zone ()` 和 `scha_resource_get_zone ()` 函数的作用分别与 `scha_resource_open ()` 和 `scha_resource_get ()` 相同，且带有一个额外的 `cluster` 参数，用于指定资源组存在于其中且您要在其上执行操作的区域群集的名称。当代码在全局区域中执行，但需要在指定区域群集上操作时，这些函数十分有用。无法在某一区域群集中使用它们来访问其他区域群集。

如果 `scha_resource_open_zone ()` 或 `scha_resource_get_zone ()` 的 `cluster` 参数为 `NULL`，则会对从中执行调用的群集执行查询；换句话说，带有 `NULL` 参数的调用分别等效于 `scha_resource_open ()` 或 `scha_resource_get ()`。

要关闭由 `scha_resource_open_zone ()` 返回的句柄，请使用 `scha_resource_close ()`。不需要 `cluster` 参数。

Tag 参数

下面列出了用于指定资源属性名称的宏。输出是资源的属性值。`SCHA_RESOURCE_STATE`、`SCHA_STATUS`、`SCHA_NUM_RG_RESTARTS` 和 `SCHA_NUM_RESOURCE_RESTARTS` 属性引用执行命令的节点上的值（请参见 [r_properties\(5\) \[1103\]](#)）。

指示输出参数和任何其他参数的类型。

扩展属性

这些属性是在资源类型的资源类型注册 (Resource Type Registration, RTR) 文件中声明的。资源类型的实现定义了这些属性。

`SCHA_AFFINITY_TIMEOUT`

输出参数类型为 `int*`。

`SCHA_ALL_EXTENSIONS`

输出参数类型为 `scha_str_array_t**`。返回资源的所有扩展属性的名称。

`SCHA_APPLICATION_USER`

输出参数类型为 `char**`。

`SCHA_BOOT_TIMEOUT`

输出参数类型为 `int*`。

`SCHA_CHEAP_PROBE_INTERVAL`

输出参数类型为 `int*`。

`SCHA_EXTENSION`

输出参数类型为 `scha_extprop_value_t**`。需要提供扩展属性名称的附加参数，其类型为 `char*`。返回本地节点的属性类型及其值。

当用户请求未指定明确值的节点上的该属性的值时，将会返回在 RTR 文件中声明的默认值。请参见 [rt_reg\(4\) \[1067\]](#) 手册页。

SCHA_EXTENSION_NODE

输出参数类型为 `scha_extprop_value_t**`。需要类型为 `char*` 的两个附加参数。第一个参数提供扩展属性的名称，第二个参数命名群集节点。返回指定节点的属性类型及其值。

当用户请求未指定明确值的节点上的该属性的值时，将会返回在 RTR 文件中声明的默认值。请参见 [rt_reg\(4\) \[1067\]](#) 手册页。

SCHA_FAILOVER_MODE

输出参数类型为 `scha_failover_mode_t*`。

SCHA_FINI_TIMEOUT

输出参数类型为 `int*`。

SCHA_GLOBAL_ZONE_OVERRIDE

输出参数类型为 `boolean_t*`。

SCHA_GROUP

输出参数类型为 `char**`。返回配置了资源的资源组的名称。

SCHA_INIT_TIMEOUT

输出参数类型为 `int*`。

SCHA_LOAD_BALANCING_POLICY

输出参数类型为 `char**`。

SCHA_LOAD_BALANCING_WEIGHTS

输出参数类型为 `scha_str_array_t**`。

SCHA_MONITOR_CHECK_TIMEOUT

输出参数类型为 `int*`。

SCHA_MONITOR_START_TIMEOUT

输出参数类型为 `int*`。

SCHA_MONITOR_STOP_TIMEOUT

输出参数类型为 `int*`。

SCHA_MONITORED_SWITCH

输出参数类型为 `scha_switch_t*`。返回值指示是否在本节点上监视资源。

SCHA_MONITORED_SWITCH_NODE

输出参数类型为 `scha_switch_t*`。需要命名群集节点的附加参数，其类型为 `char*`。返回值指示是否在指定节点上监视资源。

SCHA_NETWORK_RESOURCES_USED

输出参数类型为 `scha_str_array_t**`。

SCHA_NUM_RESOURCE_RESTARTS

输出参数类型为 `int*`。返回在执行此查询的区域中对该资源发生的资源重新启动请求数。有关更多详细信息，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

SCHA_NUM_RG_RESTARTS

输出参数类型为 `int*`。返回在执行此查询的区域中对该资源发生的资源组重新启动请求数。有关更多详细信息，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

SCHA_ON_OFF_SWITCH

输出参数类型为 `scha_switch_t*`。返回值指示是否在本地节点上启用资源。

SCHA_ON_OFF_SWITCH_NODE

输出参数类型为 `scha_switch_t*`。需要命名群集节点的附加参数，其类型为 `char*`。返回值指示是否在指定节点上启用资源。

SCHA_PORT_LIST

输出参数类型为 `scha_str_array_t**`。

SCHA_POSTNET_STOP_TIMEOUT

输出参数类型为 `int*`。

SCHA_PRENET_START_TIMEOUT

输出参数类型为 `int*`。

SCHA_PRE_EVICT

输出参数类型为 `boolean_t*`。

SCHA_R_DESCRIPTION

输出参数类型为 `char**`。

SCHA_RESOURCE_DEPENDENCIES

输出参数类型为 `scha_str_array_t**`。返回值列出适用于本地节点的依赖性。

SCHA_RESOURCE_DEPENDENCIES_NODE

输出参数类型为 `scha_str_array_t**`。需要命名群集节点的附加参数，其类型为 `char*`。返回值列出适用于指定节点的依赖性。

SCHA_RESOURCE_DEPENDENCIES_OFFLINE_RESTART

输出参数类型为 `scha_str_array_t**`。返回值列出适用于本地节点的依赖性。

SCHA_RESOURCE_DEPENDENCIES_OFFLINE_RESTART_NODE

输出参数类型为 `scha_str_array_t**`。需要命名群集节点的附加参数，其类型为 `char*`。返回值列出适用于指定节点的依赖性。

SCHA_RESOURCE_DEPENDENCIES_Q

输出参数类型为 `scha_str_array_t**`。

SCHA_RESOURCE_DEPENDENCIES 标记的 Q 版本还会返回为资源依赖性声明的作用域或限定符（如果有）。[Unresolved link to "r_properties\(5\)"](#) 手册页中介绍了 `{LOCAL_NODE}`、`{ANY_NODE}`、`{FROM_RG_AFFINITIES}` 和 `@node` 限定符。

SCHA_RESOURCE_DEPENDENCIES_Q_OFFLINE_RESTART

输出参数类型为 `scha_str_array_t**`。

SCHA_RESOURCE_DEPENDENCIES_OFFLINE_RESTART 标记的 Q 版本还会返回为资源依赖性声明的作用域或限定符（如果有）。[Unresolved link to "r_properties\(5\)"](#) 手册页中介绍了 `{LOCAL_NODE}`、`{ANY_NODE}`、`{FROM_RG_AFFINITIES}` 和 `@node` 限定符。

SCHA_RESOURCE_DEPENDENCIES_Q_RESTART

输出参数类型为 `scha_str_array_t**`。

SCHA_RESOURCE_DEPENDENCIES_RESTART 标记的 Q 版本还会返回为资源依赖性声明的作用域或限定符（如果有）。[Unresolved link to "r_properties\(5\)"](#) 手册页中介绍了 `{LOCAL_NODE}`、`{ANY_NODE}`、`{FROM_RG_AFFINITIES}` 和 `@node` 限定符。

SCHA_RESOURCE_DEPENDENCIES_Q_WEAK

输出参数类型为 `scha_str_array_t**`。

SCHA_RESOURCE_DEPENDENCIES_WEAK 标记的 Q 版本还会返回为资源依赖性声明的作用域或限定符（如果有）。[Unresolved link to "r_properties\(5\)"](#) 手册页中介绍了 `{LOCAL_NODE}`、`{ANY_NODE}`、`{FROM_RG_AFFINITIES}` 和 `@node` 限定符。

SCHA_RESOURCE_DEPENDENCIES_RESTART

输出参数类型为 `scha_str_array_t**`。返回值列出适用于本地节点的依赖性。

SCHA_RESOURCE_DEPENDENCIES_RESTART_NODE

输出参数类型为 `scha_str_array_t**`。需要命名群集节点的附加参数，其类型为 `char*`。返回值列出适用于指定节点的依赖性。

SCHA_RESOURCE_DEPENDENCIES_WEAK

输出参数类型为 `scha_str_array_t**`。返回值列出适用于本地节点的依赖性。

SCHA_RESOURCE_DEPENDENCIES_WEAK_NODE

输出参数类型为 `scha_str_array_t**`。需要命名群集节点的附加参数，其类型为 `char*`。返回值列出适用于指定节点的依赖性。

SCHA_RESOURCE_PROJECT_NAME

输出参数类型为 `char**`。

SCHA_RESOURCE_STATE

输出参数类型为 `scha_rsstate_t*`。返回本地节点的资源的 `RESOURCE_STATE` 属性值。

SCHA_RESOURCE_STATE_NODE

输出参数类型为 `scha_rsstate_t*`。需要命名群集节点的附加参数，其类型为 `char*`。返回指定节点的资源的 `RESOURCE_STATE` 属性值。

SCHA_RETRY_COUNT

输出参数类型为 `int*`。

SCHA_RETRY_INTERVAL

输出参数类型为 `int*`。

SCHA_SCALABLE

输出参数类型为 `boolean_t*`。

SCHA_START_TIMEOUT

输出参数类型为 `int*`。

SCHA_STATUS

输出参数类型为 `scha_status_value_t**`。返回本地节点的资源的 `STATUS` 属性值。

SCHA_STATUS_NODE

输出参数类型为 `scha_status_value_t**`。需要命名群集节点的附加参数，其类型为 `char*`。返回指定节点的资源的 `STATUS` 属性值。

SCHA_STOP_TIMEOUT

输出参数类型为 `int*`。

SCHA_THOROUGH_PROBE_INTERVAL

输出参数类型为 `int*`。

SCHA_TYPE

输出参数类型为 `char**`。

SCHA_TYPE_VERSION

输出参数类型为 char**。

SCHA_UDP_AFFINITY

输出参数类型为 boolean_t*。

SCHA_UPDATE_TIMEOUT

输出参数类型为 int*。

SCHA_VALIDATE_TIMEOUT

输出参数类型为 int*。

SCHA_WEAK_AFFINITY

输出参数类型为 boolean_t*。

下面列出了用于指定资源类型属性名称的宏。输出是资源类型的属性值。有关资源类型属性的说明，请参见 [rt_properties\(5\) \[1143\]](#)。

SCHA_API_VERSION

输出参数类型为 int*。

SCHA_BOOT

输出参数类型为 char**。

SCHA_FAILOVER

输出参数类型为 boolean_t*。

SCHA_FINI

输出参数类型为 char**。

SCHA_GLOBAL_ZONE

输出参数类型为 boolean_t*。

SCHA_INIT

输出参数类型为 char**。

SCHA_INIT_NODES

输出参数类型为 scha_initnodes_flag_t*。

SCHA_INSTALLED_NODES

输出参数类型为 scha_str_array_t**。

SCHA_MONITOR_CHECK

输出参数类型为 char**。

SCHA_MONITOR_START
输出参数类型为 char**。

SCHA_MONITOR_STOP
输出参数类型为 char**。

SCHA_PKGLIST
输出参数类型为 scha_str_array_t**。

SCHA_POSTNET_STOP
输出参数类型为 char**。

SCHA_PRENET_START
输出参数类型为 char**。

SCHA_PROXY
输出参数类型为 boolean_t*。

SCHA_RT_BASEDIR
输出参数类型为 char**。

SCHA_RT_DESCRIPTION
输出参数类型为 char**。

SCHA_RT_SYSTEM
输出参数类型为 boolean_t*。

SCHA_RT_VERSION
输出参数类型为 char**。

SCHA_SINGLE_INSTANCE
输出参数类型为 boolean_t*。

SCHA_START
输出参数类型为 char**。

SCHA_STOP
输出参数类型为 char**。

SCHA_UPDATE
输出参数类型为 char**。

SCHA_VALIDATE
输出参数类型为 char**。

如果该资源类型声明了 GLOBAL_ZONE_OVERRIDE 资源属性，则由 SCHA_GLOBAL_ZONE`optag` 检索的值是 GLOBAL_ZONE_OVERRIDE 属性的当前值，而不是 GLOBAL_ZONE 属性的值。有关更多信息，请参见 [rt_properties\(5\) \[1143\]](#) 手册页中的 Global_zone 属性和 [r_properties\(5\) \[1103\]](#) 手册页中的 Global_zone_override 属性的描述。

这些函数返回以下值：

0 该函数成功。

非零值 该函数失败。

SCHA_ERR_NOERR 该函数成功。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

例 378 使用 `scha_resource_get()` 函数

以下示例使用 `scha_resource_get()` 来获取某个资源的 `Retry_count` 属性的值，以及名为 `LogLevel` 的扩展属性的值。

```
main() {
    #include <scha.h>

    scha_err_t err;
    int retry_count_out;
    scha_extprop_value_t *loglevel_out;
    scha_resource_t handle;

    /* a configured resource */
    char * resource_name = "example_R";
    /* resource group containing example_R */
    char * group_name = "example_RG";

    err = scha_resource_open(resource_name, group_name, &handle);

    err = scha_resource_get(handle, SCHA_RETRY_COUNT, &retry_count_out);

    /* Given extension property must be defined in resourcetype RTR file. */
    err = scha_resource_get(handle, SCHA_EXTENSION, "LogLevel", &loglevel_out);

    err = scha_resource_close(handle);

    printf("The retry count for resource %s is %d\n", resource_name,
           retry_count_out);

    printf("The log level for resource %s is %d\n", resource_name,
           loglevel_out->val.val_int);
}
```

/usr/cluster/include/scha.h 头文件
/usr/cluster/lib/libscha.so 库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scha_resource_get\(1HA\) \[585\]](#)、[scha_calls\(3HA\) \[897\]](#)、[scha_strerror\(3HA\) \[1057\]](#)、[Unresolved link to "attributes5"](#)、[r_properties\(5\) \[1103\]](#)、[rt_properties\(5\) \[1143\]](#)、[rt_reg\(4\) \[1067\]](#)

名称

scha_resource_open, scha_resource_open_zone, scha_resource_get,
scha_resource_get_zone, scha_resource_close — 资源信息访问函数

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib -l scha#include
<scha.h>scha_err_t scha_resource_open( const char *rname, const char *rgname, scha_resource_t
*handle);

scha_err_t scha_resource_open_zone(const char *cluster, const char *rs_name, const char
*rg_name, scha_resource_t * handlep);

scha_err_t scha_resource_get(scha_resource_t handle, const char *tag,...);

scha_err_t scha_resource_get_zone(const char *cluster,
    scha_resource_t handlep, const char *rs_tag, ...);

scha_err_t scha_resource_close(scha_resource_t handle);
```

将 `scha_resource_open()`、`scha_resource_get()` 和 `scha_resource_close()` 函数一起使用可以访问资源组管理器 (Resource Group Manager, RGM) 群集工具管理的资源的相关信息。

`scha_resource_open()` 初始化资源访问，并返回一个将由 `scha_resource_get()` 使用的句柄。

`scha_resource_open()` 的 *rname* 参数指定要访问的资源名称。*rgname* 参数是其中配置了该资源的资源组的名称。如果组名称未知，*rgname* 参数可以是 NULL。但是，如果提供组名称，可以提高该函数的执行效率。*handle* 参数是用于保存函数调用返回值的某个变量的地址。

`scha_resource_get()` 根据 *tag* 参数的指示访问资源信息。*tag* 参数应该是 `scha_tags.h` 头文件中的某个宏定义的一个字符串值。标记后面的参数取决于 *tag* 的值。可能需要在标记后面额外添加一个参数，用于指示要从中检索信息的群集节点或者指示特定于标记的其他信息。参数列表中最后一个参数的类型应该适合用于保存 *tag* 指定的信息。该参数是资源信息的输出参数。如果函数失败，不会为输出参数返回值。

对用于 `scha_resource_get()` 的句柄调用 `scha_resource_close()` 之前，所分配的用于保存 `scha_resource_get()` 返回的信息的内存将保持不变。请注意，使用相同的句柄和标记重复调用 `scha_resource_get()` 会导致分配新的内存。在一个调用中为返回值而分配的空间不会被后续调用覆盖和重新使用。

`scha_resource_close()` 函数接受上次调用 `scha_resource_open()` 时返回的 *handle* 参数。它将使该句柄失效，并释放为了保存使用该句柄执行的 `scha_resource_get()` 调用的返回值而分配的内存。

下文中列出了 `scha_tags.h` 中定义的可以用作 `scha_resource_get()` 的 *tag* 参数的宏。

指示输出参数和任何其他参数的类型。[scha_calls\(3HA\) \[897\]](#) 介绍了结构和 enum 类型。

`scha_resource_open_zone ()` 和 `scha_resource_get_zone ()` 函数的作用分别与 `scha_resource_open ()` 和 `scha_resource_get ()` 相同，且带有一个额外的 `cluster` 参数，用于指定资源组存在于其中且您要在其上执行操作的区域群集的名称。当代码在全局区域中执行，但需要在指定区域群集上操作时，这些函数十分有用。无法在某一区域群集中使用它们来访问其他区域群集。

如果 `scha_resource_open_zone ()` 或 `scha_resource_get_zone ()` 的 `cluster` 参数为 `NULL`，则会对从中执行调用的群集执行查询；换句话说，带有 `NULL` 参数的调用分别等效于 `scha_resource_open ()` 或 `scha_resource_get ()`。

要关闭由 `scha_resource_open_zone ()` 返回的句柄，请使用 `scha_resource_close ()`。不需要 `cluster` 参数。

Tag 参数

下面列出了用于指定资源属性名称的宏。输出是资源的属性值。`SCHA_RESOURCE_STATE`、`SCHA_STATUS`、`SCHA_NUM_RG_RESTARTS` 和 `SCHA_NUM_RESOURCE_RESTARTS` 属性引用执行命令的节点上的值（请参见 [r_properties\(5\) \[1103\]](#)）。

指示输出参数和任何其他参数的类型。

扩展属性

这些属性是在资源类型的资源类型注册 (Resource Type Registration, RTR) 文件中声明的。资源类型的实现定义了这些属性。

`SCHA_AFFINITY_TIMEOUT`

输出参数类型为 `int*`。

`SCHA_ALL_EXTENSIONS`

输出参数类型为 `scha_str_array_t**`。返回资源的所有扩展属性的名称。

`SCHA_APPLICATION_USER`

输出参数类型为 `char**`。

`SCHA_BOOT_TIMEOUT`

输出参数类型为 `int*`。

`SCHA_CHEAP_PROBE_INTERVAL`

输出参数类型为 `int*`。

`SCHA_EXTENSION`

输出参数类型为 `scha_extprop_value_t**`。需要提供扩展属性名称的附加参数，其类型为 `char*`。返回本地节点的属性类型及其值。

当用户请求未指定明确值的节点上的该属性的值时，将会返回在 RTR 文件中声明的默认值。请参见 [rt_reg\(4\) \[1067\]](#) 手册页。

SCHA_EXTENSION_NODE

输出参数类型为 `scha_extprop_value_t**`。需要类型为 `char*` 的两个附加参数。第一个参数提供扩展属性的名称，第二个参数命名群集节点。返回指定节点的属性类型及其值。

当用户请求未指定明确值的节点上的该属性的值时，将会返回在 RTR 文件中声明的默认值。请参见 [rt_reg\(4\) \[1067\]](#) 手册页。

SCHA_FAILOVER_MODE

输出参数类型为 `scha_failover_mode_t*`。

SCHA_FINI_TIMEOUT

输出参数类型为 `int*`。

SCHA_GLOBAL_ZONE_OVERRIDE

输出参数类型为 `boolean_t*`。

SCHA_GROUP

输出参数类型为 `char**`。返回配置了资源的资源组的名称。

SCHA_INIT_TIMEOUT

输出参数类型为 `int*`。

SCHA_LOAD_BALANCING_POLICY

输出参数类型为 `char**`。

SCHA_LOAD_BALANCING_WEIGHTS

输出参数类型为 `scha_str_array_t**`。

SCHA_MONITOR_CHECK_TIMEOUT

输出参数类型为 `int*`。

SCHA_MONITOR_START_TIMEOUT

输出参数类型为 `int*`。

SCHA_MONITOR_STOP_TIMEOUT

输出参数类型为 `int*`。

SCHA_MONITORED_SWITCH

输出参数类型为 `scha_switch_t*`。返回值指示是否在本节点上监视资源。

SCHA_MONITORED_SWITCH_NODE

输出参数类型为 `scha_switch_t*`。需要命名群集节点的附加参数，其类型为 `char*`。返回值指示是否在指定节点上监视资源。

SCHA_NETWORK_RESOURCES_USED

输出参数类型为 `scha_str_array_t**`。

SCHA_NUM_RESOURCE_RESTARTS

输出参数类型为 `int*`。返回在执行此查询的区域中对该资源发生的资源重新启动请求数。有关更多详细信息，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

SCHA_NUM_RG_RESTARTS

输出参数类型为 `int*`。返回在执行此查询的区域中对该资源发生的资源组重新启动请求数。有关更多详细信息，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

SCHA_ON_OFF_SWITCH

输出参数类型为 `scha_switch_t*`。返回值指示是否在本地节点上启用资源。

SCHA_ON_OFF_SWITCH_NODE

输出参数类型为 `scha_switch_t*`。需要命名群集节点的附加参数，其类型为 `char*`。返回值指示是否在指定节点上启用资源。

SCHA_PORT_LIST

输出参数类型为 `scha_str_array_t**`。

SCHA_POSTNET_STOP_TIMEOUT

输出参数类型为 `int*`。

SCHA_PRENET_START_TIMEOUT

输出参数类型为 `int*`。

SCHA_PRE_EVICT

输出参数类型为 `boolean_t*`。

SCHA_R_DESCRIPTION

输出参数类型为 `char**`。

SCHA_RESOURCE_DEPENDENCIES

输出参数类型为 `scha_str_array_t**`。返回值列出适用于本地节点的依赖性。

SCHA_RESOURCE_DEPENDENCIES_NODE

输出参数类型为 `scha_str_array_t**`。需要命名群集节点的附加参数，其类型为 `char*`。返回值列出适用于指定节点的依赖性。

SCHA_RESOURCE_DEPENDENCIES_OFFLINE_RESTART

输出参数类型为 `scha_str_array_t**`。返回值列出适用于本地节点的依赖性。

SCHA_RESOURCE_DEPENDENCIES_OFFLINE_RESTART_NODE

输出参数类型为 `scha_str_array_t**`。需要命名群集节点的附加参数，其类型为 `char*`。返回值列出适用于指定节点的依赖性。

SCHA_RESOURCE_DEPENDENCIES_Q

输出参数类型为 `scha_str_array_t**`。

SCHA_RESOURCE_DEPENDENCIES 标记的 Q 版本还会返回为资源依赖性声明的作用域或限定符（如果有）。 [Unresolved link to "r_properties\(5\)"](#) 手册页中介绍了 `{LOCAL_NODE}`、`{ANY_NODE}`、`{FROM_RG_AFFINITIES}` 和 `@node` 限定符。

SCHA_RESOURCE_DEPENDENCIES_Q_OFFLINE_RESTART

输出参数类型为 `scha_str_array_t**`。

SCHA_RESOURCE_DEPENDENCIES_OFFLINE_RESTART 标记的 Q 版本还会返回为资源依赖性声明的作用域或限定符（如果有）。 [Unresolved link to "r_properties\(5\)"](#) 手册页中介绍了 `{LOCAL_NODE}`、`{ANY_NODE}`、`{FROM_RG_AFFINITIES}` 和 `@node` 限定符。

SCHA_RESOURCE_DEPENDENCIES_Q_RESTART

输出参数类型为 `scha_str_array_t**`。

SCHA_RESOURCE_DEPENDENCIES_RESTART 标记的 Q 版本还会返回为资源依赖性声明的作用域或限定符（如果有）。 [Unresolved link to "r_properties\(5\)"](#) 手册页中介绍了 `{LOCAL_NODE}`、`{ANY_NODE}`、`{FROM_RG_AFFINITIES}` 和 `@node` 限定符。

SCHA_RESOURCE_DEPENDENCIES_Q_WEAK

输出参数类型为 `scha_str_array_t**`。

SCHA_RESOURCE_DEPENDENCIES_WEAK 标记的 Q 版本还会返回为资源依赖性声明的作用域或限定符（如果有）。 [Unresolved link to "r_properties\(5\)"](#) 手册页中介绍了 `{LOCAL_NODE}`、`{ANY_NODE}`、`{FROM_RG_AFFINITIES}` 和 `@node` 限定符。

SCHA_RESOURCE_DEPENDENCIES_RESTART

输出参数类型为 `scha_str_array_t**`。返回值列出适用于本地节点的依赖性。

SCHA_RESOURCE_DEPENDENCIES_RESTART_NODE

输出参数类型为 `scha_str_array_t**`。需要命名群集节点的附加参数，其类型为 `char*`。返回值列出适用于指定节点的依赖性。

SCHA_RESOURCE_DEPENDENCIES_WEAK

输出参数类型为 `scha_str_array_t**`。返回值列出适用于本地节点的依赖性。

SCHA_RESOURCE_DEPENDENCIES_WEAK_NODE

输出参数类型为 `scha_str_array_t**`。需要命名群集节点的附加参数，其类型为 `char*`。返回值列出适用于指定节点的依赖性。

SCHA_RESOURCE_PROJECT_NAME

输出参数类型为 `char**`。

SCHA_RESOURCE_STATE

输出参数类型为 `scha_rsstate_t*`。返回本地节点的资源的 `RESOURCE_STATE` 属性值。

SCHA_RESOURCE_STATE_NODE

输出参数类型为 `scha_rsstate_t*`。需要命名群集节点的附加参数，其类型为 `char*`。返回指定节点的资源的 `RESOURCE_STATE` 属性值。

SCHA_RETRY_COUNT

输出参数类型为 `int*`。

SCHA_RETRY_INTERVAL

输出参数类型为 `int*`。

SCHA_SCALABLE

输出参数类型为 `boolean_t*`。

SCHA_START_TIMEOUT

输出参数类型为 `int*`。

SCHA_STATUS

输出参数类型为 `scha_status_value_t**`。返回本地节点的资源的 `STATUS` 属性值。

SCHA_STATUS_NODE

输出参数类型为 `scha_status_value_t**`。需要命名群集节点的附加参数，其类型为 `char*`。返回指定节点的资源的 `STATUS` 属性值。

SCHA_STOP_TIMEOUT

输出参数类型为 `int*`。

SCHA_THOROUGH_PROBE_INTERVAL

输出参数类型为 `int*`。

SCHA_TYPE

输出参数类型为 `char**`。

SCHA_TYPE_VERSION

输出参数类型为 char**。

SCHA_UDP_AFFINITY

输出参数类型为 boolean_t*。

SCHA_UPDATE_TIMEOUT

输出参数类型为 int*。

SCHA_VALIDATE_TIMEOUT

输出参数类型为 int*。

SCHA_WEAK_AFFINITY

输出参数类型为 boolean_t*。

下面列出了用于指定资源类型属性名称的宏。输出是资源类型的属性值。有关资源类型属性的说明，请参见 [rt_properties\(5\) \[1143\]](#)。

SCHA_API_VERSION

输出参数类型为 int*。

SCHA_BOOT

输出参数类型为 char**。

SCHA_FAILOVER

输出参数类型为 boolean_t*。

SCHA_FINI

输出参数类型为 char**。

SCHA_GLOBAL_ZONE

输出参数类型为 boolean_t*。

SCHA_INIT

输出参数类型为 char**。

SCHA_INIT_NODES

输出参数类型为 scha_initnodes_flag_t*。

SCHA_INSTALLED_NODES

输出参数类型为 scha_str_array_t**。

SCHA_MONITOR_CHECK

输出参数类型为 char**。

SCHA_MONITOR_START
输出参数类型为 char**。

SCHA_MONITOR_STOP
输出参数类型为 char**。

SCHA_PKGLIST
输出参数类型为 scha_str_array_t**。

SCHA_POSTNET_STOP
输出参数类型为 char**。

SCHA_PRENET_START
输出参数类型为 char**。

SCHA_PROXY
输出参数类型为 boolean_t*。

SCHA_RT_BASEDIR
输出参数类型为 char**。

SCHA_RT_DESCRIPTION
输出参数类型为 char**。

SCHA_RT_SYSTEM
输出参数类型为 boolean_t*。

SCHA_RT_VERSION
输出参数类型为 char**。

SCHA_SINGLE_INSTANCE
输出参数类型为 boolean_t*。

SCHA_START
输出参数类型为 char**。

SCHA_STOP
输出参数类型为 char**。

SCHA_UPDATE
输出参数类型为 char**。

SCHA_VALIDATE
输出参数类型为 char**。

如果该资源类型声明了 GLOBAL_ZONE_OVERRIDE 资源属性，则由 SCHA_GLOBAL_ZONE`optag` 检索的值是 GLOBAL_ZONE_OVERRIDE 属性的当前值，而不是 GLOBAL_ZONE 属性的值。有关更多信息，请参见 [rt_properties\(5\) \[1143\]](#) 手册页中的 Global_zone 属性和 [r_properties\(5\) \[1103\]](#) 手册页中的 Global_zone_override 属性的描述。

这些函数返回以下值：

0 该函数成功。

非零值 该函数失败。

SCHA_ERR_NOERR 该函数成功。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

例 379 使用 `scha_resource_get()` 函数

以下示例使用 `scha_resource_get()` 来获取某个资源的 `Retry_count` 属性的值，以及名为 `LogLevel` 的扩展属性的值。

```
main() {
    #include <scha.h>

    scha_err_t err;
    int retry_count_out;
    scha_extprop_value_t *loglevel_out;
    scha_resource_t handle;

    /* a configured resource */
    char * resource_name = "example_R";
    /* resource group containing example_R */
    char * group_name = "example_RG";

    err = scha_resource_open(resource_name, group_name, &handle);

    err = scha_resource_get(handle, SCHA_RETRY_COUNT, &retry_count_out);

    /* Given extension property must be defined in resourcetype RTR file. */
    err = scha_resource_get(handle, SCHA_EXTENSION, "LogLevel", &loglevel_out);

    err = scha_resource_close(handle);

    printf("The retry count for resource %s is %d\n", resource_name,
           retry_count_out);

    printf("The log level for resource %s is %d\n", resource_name,
           loglevel_out->val.val_int);
}
```

/usr/cluster/include/scha.h 头文件

/usr/cluster/lib/libscha.so 库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scha_resource_get\(1HA\) \[585\]](#)、[scha_calls\(3HA\) \[897\]](#)、[scha_strerror\(3HA\) \[1057\]](#)、[Unresolved link to "attributes5"](#)、[r_properties\(5\) \[1103\]](#)、[rt_properties\(5\) \[1143\]](#)、[rt_reg\(4\) \[1067\]](#)

名称

scha_resource_open, scha_resource_open_zone, scha_resource_get,
scha_resource_get_zone, scha_resource_close — 资源信息访问函数

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib -l scha#include
<scha.h>scha_err_t scha_resource_open( const char *rname, const char *rgname, scha_resource_t
*handle);

scha_err_t scha_resource_open_zone(const char *cluster, const char *rs_name, const char
*rg_name, scha_resource_t * handlep);

scha_err_t scha_resource_get(scha_resource_t handle, const char *tag,...);

scha_err_t scha_resource_get_zone(const char *cluster,
    scha_resource_t handlep, const char *rs_tag, ...);

scha_err_t scha_resource_close(scha_resource_t handle);
```

将 `scha_resource_open()`、`scha_resource_get()` 和 `scha_resource_close()` 函数一起使用可以访问资源组管理器 (Resource Group Manager, RGM) 群集工具管理的资源的相关信息。

`scha_resource_open()` 初始化资源访问，并返回一个将由 `scha_resource_get()` 使用的句柄。

`scha_resource_open()` 的 *rname* 参数指定要访问的资源名称。*rgname* 参数是其中配置了该资源的资源组的名称。如果组名称未知，*rgname* 参数可以是 NULL。但是，如果提供组名称，可以提高该函数的执行效率。*handle* 参数是用于保存函数调用返回值的某个变量的地址。

`scha_resource_get()` 根据 *tag* 参数的指示访问资源信息。*tag* 参数应该是 `scha_tags.h` 头文件中的某个宏定义的一个字符串值。标记后面的参数取决于 *tag* 的值。可能需要在标记后面额外添加一个参数，用于指示要从中检索信息的群集节点或者指示特定于标记的其他信息。参数列表中最后一个参数的类型应该适合用于保存 *tag* 指定的信息。该参数是资源信息的输出参数。如果函数失败，不会为输出参数返回值。

对用于 `scha_resource_get()` 的句柄调用 `scha_resource_close()` 之前，所分配的用于保存 `scha_resource_get()` 返回的信息的内存将保持不变。请注意，使用相同的句柄和标记重复调用 `scha_resource_get()` 会导致分配新的内存。在一个调用中为返回值而分配的空间不会被后续调用覆盖和重新使用。

`scha_resource_close()` 函数接受上次调用 `scha_resource_open()` 时返回的 *handle* 参数。它将使该句柄失效，并释放为了保存使用该句柄执行的 `scha_resource_get()` 调用的返回值而分配的内存。

下文中列出了 `scha_tags.h` 中定义的可以用作 `scha_resource_get()` 的 *tag* 参数的宏。

指示输出参数和任何其他参数的类型。[scha_calls\(3HA\) \[897\]](#) 介绍了结构和 enum 类型。

`scha_resource_open_zone ()` 和 `scha_resource_get_zone ()` 函数的作用分别与 `scha_resource_open ()` 和 `scha_resource_get ()` 相同，且带有一个额外的 `cluster` 参数，用于指定资源组存在于其中且您要在其上执行操作的区域群集的名称。当代码在全局区域中执行，但需要在指定区域群集上操作时，这些函数十分有用。无法在某一区域群集中使用它们来访问其他区域群集。

如果 `scha_resource_open_zone ()` 或 `scha_resource_get_zone ()` 的 `cluster` 参数为 `NULL`，则会对从中执行调用的群集执行查询；换句话说，带有 `NULL` 参数的调用分别等效于 `scha_resource_open ()` 或 `scha_resource_get ()`。

要关闭由 `scha_resource_open_zone ()` 返回的句柄，请使用 `scha_resource_close ()`。不需要 `cluster` 参数。

Tag 参数

下面列出了用于指定资源属性名称的宏。输出是资源的属性值。`SCHA_RESOURCE_STATE`、`SCHA_STATUS`、`SCHA_NUM_RG_RESTARTS` 和 `SCHA_NUM_RESOURCE_RESTARTS` 属性引用执行命令的节点上的值（请参见 [r_properties\(5\) \[1103\]](#)）。

指示输出参数和任何其他参数的类型。

扩展属性

这些属性是在资源类型的资源类型注册 (Resource Type Registration, RTR) 文件中声明的。资源类型的实现定义了这些属性。

`SCHA_AFFINITY_TIMEOUT`

输出参数类型为 `int*`。

`SCHA_ALL_EXTENSIONS`

输出参数类型为 `scha_str_array_t**`。返回资源的所有扩展属性的名称。

`SCHA_APPLICATION_USER`

输出参数类型为 `char**`。

`SCHA_BOOT_TIMEOUT`

输出参数类型为 `int*`。

`SCHA_CHEAP_PROBE_INTERVAL`

输出参数类型为 `int*`。

`SCHA_EXTENSION`

输出参数类型为 `scha_extprop_value_t**`。需要提供扩展属性名称的附加参数，其类型为 `char*`。返回本地节点的属性类型及其值。

当用户请求未指定明确值的节点上的该属性的值时，将会返回在 RTR 文件中声明的默认值。请参见 [rt_reg\(4\) \[1067\]](#) 手册页。

SCHA_EXTENSION_NODE

输出参数类型为 `scha_extprop_value_t**`。需要类型为 `char*` 的两个附加参数。第一个参数提供扩展属性的名称，第二个参数命名群集节点。返回指定节点的属性类型及其值。

当用户请求未指定明确值的节点上的该属性的值时，将会返回在 RTR 文件中声明的默认值。请参见 [rt_reg\(4\) \[1067\]](#) 手册页。

SCHA_FAILOVER_MODE

输出参数类型为 `scha_failover_mode_t*`。

SCHA_FINI_TIMEOUT

输出参数类型为 `int*`。

SCHA_GLOBAL_ZONE_OVERRIDE

输出参数类型为 `boolean_t*`。

SCHA_GROUP

输出参数类型为 `char**`。返回配置了资源的资源组的名称。

SCHA_INIT_TIMEOUT

输出参数类型为 `int*`。

SCHA_LOAD_BALANCING_POLICY

输出参数类型为 `char**`。

SCHA_LOAD_BALANCING_WEIGHTS

输出参数类型为 `scha_str_array_t**`。

SCHA_MONITOR_CHECK_TIMEOUT

输出参数类型为 `int*`。

SCHA_MONITOR_START_TIMEOUT

输出参数类型为 `int*`。

SCHA_MONITOR_STOP_TIMEOUT

输出参数类型为 `int*`。

SCHA_MONITORED_SWITCH

输出参数类型为 `scha_switch_t*`。返回值指示是否在本节点上监视资源。

SCHA_MONITORED_SWITCH_NODE

输出参数类型为 `scha_switch_t*`。需要命名群集节点的附加参数，其类型为 `char*`。返回值指示是否在指定节点上监视资源。

SCHA_NETWORK_RESOURCES_USED

输出参数类型为 `scha_str_array_t**`。

SCHA_NUM_RESOURCE_RESTARTS

输出参数类型为 `int*`。返回在执行此查询的区域中对该资源发生的资源重新启动请求数。有关更多详细信息，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

SCHA_NUM_RG_RESTARTS

输出参数类型为 `int*`。返回在执行此查询的区域中对该资源发生的资源组重新启动请求数。有关更多详细信息，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

SCHA_ON_OFF_SWITCH

输出参数类型为 `scha_switch_t*`。返回值指示是否在本地节点上启用资源。

SCHA_ON_OFF_SWITCH_NODE

输出参数类型为 `scha_switch_t*`。需要命名群集节点的附加参数，其类型为 `char*`。返回值指示是否在指定节点上启用资源。

SCHA_PORT_LIST

输出参数类型为 `scha_str_array_t**`。

SCHA_POSTNET_STOP_TIMEOUT

输出参数类型为 `int*`。

SCHA_PRENET_START_TIMEOUT

输出参数类型为 `int*`。

SCHA_PRE_EVICT

输出参数类型为 `boolean_t*`。

SCHA_R_DESCRIPTION

输出参数类型为 `char**`。

SCHA_RESOURCE_DEPENDENCIES

输出参数类型为 `scha_str_array_t**`。返回值列出适用于本地节点的依赖性。

SCHA_RESOURCE_DEPENDENCIES_NODE

输出参数类型为 `scha_str_array_t**`。需要命名群集节点的附加参数，其类型为 `char*`。返回值列出适用于指定节点的依赖性。

SCHA_RESOURCE_DEPENDENCIES_OFFLINE_RESTART

输出参数类型为 `scha_str_array_t**`。返回值列出适用于本地节点的依赖性。

SCHA_RESOURCE_DEPENDENCIES_OFFLINE_RESTART_NODE

输出参数类型为 `scha_str_array_t**`。需要命名群集节点的附加参数，其类型为 `char*`。返回值列出适用于指定节点的依赖性。

SCHA_RESOURCE_DEPENDENCIES_Q

输出参数类型为 `scha_str_array_t**`。

SCHA_RESOURCE_DEPENDENCIES 标记的 Q 版本还会返回为资源依赖性声明的作用域或限定符（如果有）。 [Unresolved link to "r_properties\(5\)"](#) 手册页中介绍了 `{LOCAL_NODE}`、`{ANY_NODE}`、`{FROM_RG_AFFINITIES}` 和 `@node` 限定符。

SCHA_RESOURCE_DEPENDENCIES_Q_OFFLINE_RESTART

输出参数类型为 `scha_str_array_t**`。

SCHA_RESOURCE_DEPENDENCIES_OFFLINE_RESTART 标记的 Q 版本还会返回为资源依赖性声明的作用域或限定符（如果有）。 [Unresolved link to "r_properties\(5\)"](#) 手册页中介绍了 `{LOCAL_NODE}`、`{ANY_NODE}`、`{FROM_RG_AFFINITIES}` 和 `@node` 限定符。

SCHA_RESOURCE_DEPENDENCIES_Q_RESTART

输出参数类型为 `scha_str_array_t**`。

SCHA_RESOURCE_DEPENDENCIES_RESTART 标记的 Q 版本还会返回为资源依赖性声明的作用域或限定符（如果有）。 [Unresolved link to "r_properties\(5\)"](#) 手册页中介绍了 `{LOCAL_NODE}`、`{ANY_NODE}`、`{FROM_RG_AFFINITIES}` 和 `@node` 限定符。

SCHA_RESOURCE_DEPENDENCIES_Q_WEAK

输出参数类型为 `scha_str_array_t**`。

SCHA_RESOURCE_DEPENDENCIES_WEAK 标记的 Q 版本还会返回为资源依赖性声明的作用域或限定符（如果有）。 [Unresolved link to "r_properties\(5\)"](#) 手册页中介绍了 `{LOCAL_NODE}`、`{ANY_NODE}`、`{FROM_RG_AFFINITIES}` 和 `@node` 限定符。

SCHA_RESOURCE_DEPENDENCIES_RESTART

输出参数类型为 `scha_str_array_t**`。返回值列出适用于本地节点的依赖性。

SCHA_RESOURCE_DEPENDENCIES_RESTART_NODE

输出参数类型为 `scha_str_array_t**`。需要命名群集节点的附加参数，其类型为 `char*`。返回值列出适用于指定节点的依赖性。

SCHA_RESOURCE_DEPENDENCIES_WEAK

输出参数类型为 `scha_str_array_t**`。返回值列出适用于本地节点的依赖性。

SCHA_RESOURCE_DEPENDENCIES_WEAK_NODE

输出参数类型为 `scha_str_array_t**`。需要命名群集节点的附加参数，其类型为 `char*`。返回值列出适用于指定节点的依赖性。

SCHA_RESOURCE_PROJECT_NAME

输出参数类型为 `char**`。

SCHA_RESOURCE_STATE

输出参数类型为 `scha_rsstate_t*`。返回本地节点的资源的 `RESOURCE_STATE` 属性值。

SCHA_RESOURCE_STATE_NODE

输出参数类型为 `scha_rsstate_t*`。需要命名群集节点的附加参数，其类型为 `char*`。返回指定节点的资源的 `RESOURCE_STATE` 属性值。

SCHA_RETRY_COUNT

输出参数类型为 `int*`。

SCHA_RETRY_INTERVAL

输出参数类型为 `int*`。

SCHA_SCALABLE

输出参数类型为 `boolean_t*`。

SCHA_START_TIMEOUT

输出参数类型为 `int*`。

SCHA_STATUS

输出参数类型为 `scha_status_value_t**`。返回本地节点的资源的 `STATUS` 属性值。

SCHA_STATUS_NODE

输出参数类型为 `scha_status_value_t**`。需要命名群集节点的附加参数，其类型为 `char*`。返回指定节点的资源的 `STATUS` 属性值。

SCHA_STOP_TIMEOUT

输出参数类型为 `int*`。

SCHA_THOROUGH_PROBE_INTERVAL

输出参数类型为 `int*`。

SCHA_TYPE

输出参数类型为 `char**`。

SCHA_TYPE_VERSION

输出参数类型为 char**。

SCHA_UDP_AFFINITY

输出参数类型为 boolean_t*。

SCHA_UPDATE_TIMEOUT

输出参数类型为 int*。

SCHA_VALIDATE_TIMEOUT

输出参数类型为 int*。

SCHA_WEAK_AFFINITY

输出参数类型为 boolean_t*。

下面列出了用于指定资源类型属性名称的宏。输出是资源类型的属性值。有关资源类型属性的说明，请参见 [rt_properties\(5\) \[1143\]](#)。

SCHA_API_VERSION

输出参数类型为 int*。

SCHA_BOOT

输出参数类型为 char**。

SCHA_FAILOVER

输出参数类型为 boolean_t*。

SCHA_FINI

输出参数类型为 char**。

SCHA_GLOBAL_ZONE

输出参数类型为 boolean_t*。

SCHA_INIT

输出参数类型为 char**。

SCHA_INIT_NODES

输出参数类型为 scha_initnodes_flag_t*。

SCHA_INSTALLED_NODES

输出参数类型为 scha_str_array_t**。

SCHA_MONITOR_CHECK

输出参数类型为 char**。

SCHA_MONITOR_START
输出参数类型为 char**。

SCHA_MONITOR_STOP
输出参数类型为 char**。

SCHA_PKGLIST
输出参数类型为 scha_str_array_t**。

SCHA_POSTNET_STOP
输出参数类型为 char**。

SCHA_PRENET_START
输出参数类型为 char**。

SCHA_PROXY
输出参数类型为 boolean_t*。

SCHA_RT_BASEDIR
输出参数类型为 char**。

SCHA_RT_DESCRIPTION
输出参数类型为 char**。

SCHA_RT_SYSTEM
输出参数类型为 boolean_t*。

SCHA_RT_VERSION
输出参数类型为 char**。

SCHA_SINGLE_INSTANCE
输出参数类型为 boolean_t*。

SCHA_START
输出参数类型为 char**。

SCHA_STOP
输出参数类型为 char**。

SCHA_UPDATE
输出参数类型为 char**。

SCHA_VALIDATE
输出参数类型为 char**。

如果该资源类型声明了 GLOBAL_ZONE_OVERRIDE 资源属性，则由 SCHA_GLOBAL_ZONE`optag` 检索的值是 GLOBAL_ZONE_OVERRIDE 属性的当前值，而不是 GLOBAL_ZONE 属性的值。有关更多信息，请参见 [rt_properties\(5\) \[1143\]](#) 手册页中的 Global_zone 属性和 [r_properties\(5\) \[1103\]](#) 手册页中的 Global_zone_override 属性的描述。

这些函数返回以下值：

0 该函数成功。

非零值 该函数失败。

SCHA_ERR_NOERR 该函数成功。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

例 380 使用 `scha_resource_get()` 函数

以下示例使用 `scha_resource_get()` 来获取某个资源的 `Retry_count` 属性的值，以及名为 `LogLevel` 的扩展属性的值。

```
main() {
    #include <scha.h>

    scha_err_t err;
    int retry_count_out;
    scha_extprop_value_t *loglevel_out;
    scha_resource_t handle;

    /* a configured resource */
    char * resource_name = "example_R";
    /* resource group containing example_R */
    char * group_name = "example_RG";

    err = scha_resource_open(resource_name, group_name, &handle);

    err = scha_resource_get(handle, SCHA_RETRY_COUNT, &retry_count_out);

    /* Given extension property must be defined in resourcetype RTR file. */
    err = scha_resource_get(handle, SCHA_EXTENSION, "LogLevel", &loglevel_out);

    err = scha_resource_close(handle);

    printf("The retry count for resource %s is %d\n", resource_name,
        retry_count_out);

    printf("The log level for resource %s is %d\n", resource_name,
        loglevel_out->val.val_int);
}
```

/usr/cluster/include/scha.h 头文件

/usr/cluster/lib/libscha.so 库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scha_resource_get\(1HA\) \[585\]](#)、[scha_calls\(3HA\) \[897\]](#)、[scha_strerror\(3HA\) \[1057\]](#)、[Unresolved link to "attributes5"](#)、[r_properties\(5\) \[1103\]](#)、[rt_properties\(5\) \[1143\]](#)、[rt_reg\(4\) \[1067\]](#)

名称

scha_resource_open, scha_resource_open_zone, scha_resource_get,
scha_resource_get_zone, scha_resource_close — 资源信息访问函数

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib -l scha#include
<scha.h>scha_err_t scha_resource_open( const char *rname, const char *rgname, scha_resource_t
*handle);

scha_err_t scha_resource_open_zone(const char *cluster, const char *rs_name, const char
*rg_name, scha_resource_t * handlep);

scha_err_t scha_resource_get(scha_resource_t handle, const char *tag,...);

scha_err_t scha_resource_get_zone(const char *cluster,
    scha_resource_t handlep, const char *rs_tag, ...);

scha_err_t scha_resource_close(scha_resource_t handle);
```

将 `scha_resource_open()`、`scha_resource_get()` 和 `scha_resource_close()` 函数一起使用可以访问资源组管理器 (Resource Group Manager, RGM) 群集工具管理的资源的相关信息。

`scha_resource_open()` 初始化资源访问，并返回一个将由 `scha_resource_get()` 使用的句柄。

`scha_resource_open()` 的 `rname` 参数指定要访问的资源名称。`rgname` 参数是其中配置了该资源的资源组的名称。如果组名称未知，`rgname` 参数可以是 NULL。但是，如果提供组名称，可以提高该函数的执行效率。`handle` 参数是用于保存函数调用返回值的某个变量的地址。

`scha_resource_get()` 根据 `tag` 参数的指示访问资源信息。`tag` 参数应该是 `scha_tags.h` 头文件中的某个宏定义的一个字符串值。标记后面的参数取决于 `tag` 的值。可能需要在标记后面额外添加一个参数，用于指示要从中检索信息的群集节点或者指示特定于标记的其他信息。参数列表中最后一个参数的类型应该适合用于保存 `tag` 指定的信息。该参数是资源信息的输出参数。如果函数失败，不会为输出参数返回值。

对用于 `scha_resource_get()` 的句柄调用 `scha_resource_close()` 之前，所分配的用于保存 `scha_resource_get()` 返回的信息的内存将保持不变。请注意，使用相同的句柄和标记重复调用 `scha_resource_get()` 会导致分配新的内存。在一个调用中为返回值而分配的空间不会被后续调用覆盖和重新使用。

`scha_resource_close()` 函数接受上次调用 `scha_resource_open()` 时返回的 `handle` 参数。它将使该句柄失效，并释放为了保存使用该句柄执行的 `scha_resource_get()` 调用的返回值而分配的内存。

下文中列出了 `scha_tags.h` 中定义的可以用作 `scha_resource_get()` 的 `tag` 参数的宏。

指示输出参数和任何其他参数的类型。[scha_calls\(3HA\) \[897\]](#) 介绍了结构和 enum 类型。

`scha_resource_open_zone ()` 和 `scha_resource_get_zone ()` 函数的作用分别与 `scha_resource_open ()` 和 `scha_resource_get ()` 相同，且带有一个额外的 `cluster` 参数，用于指定资源组存在于其中且您要在其上执行操作的区域群集的名称。当代码在全局区域中执行，但需要在指定区域群集上操作时，这些函数十分有用。无法在某一区域群集中使用它们来访问其他区域群集。

如果 `scha_resource_open_zone ()` 或 `scha_resource_get_zone ()` 的 `cluster` 参数为 `NULL`，则会对从中执行调用的群集执行查询；换句话说，带有 `NULL` 参数的调用分别等效于 `scha_resource_open ()` 或 `scha_resource_get ()`。

要关闭由 `scha_resource_open_zone ()` 返回的句柄，请使用 `scha_resource_close ()`。不需要 `cluster` 参数。

Tag 参数

下面列出了用于指定资源属性名称的宏。输出是资源的属性值。`SCHA_RESOURCE_STATE`、`SCHA_STATUS`、`SCHA_NUM_RG_RESTARTS` 和 `SCHA_NUM_RESOURCE_RESTARTS` 属性引用执行命令的节点上的值（请参见 [r_properties\(5\) \[1103\]](#)）。

指示输出参数和任何其他参数的类型。

扩展属性

这些属性是在资源类型的资源类型注册 (Resource Type Registration, RTR) 文件中声明的。资源类型的实现定义了这些属性。

`SCHA_AFFINITY_TIMEOUT`

输出参数类型为 `int*`。

`SCHA_ALL_EXTENSIONS`

输出参数类型为 `scha_str_array_t**`。返回资源的所有扩展属性的名称。

`SCHA_APPLICATION_USER`

输出参数类型为 `char**`。

`SCHA_BOOT_TIMEOUT`

输出参数类型为 `int*`。

`SCHA_CHEAP_PROBE_INTERVAL`

输出参数类型为 `int*`。

`SCHA_EXTENSION`

输出参数类型为 `scha_extprop_value_t**`。需要提供扩展属性名称的附加参数，其类型为 `char*`。返回本地节点的属性类型及其值。

当用户请求未指定明确值的节点上的该属性的值时，将会返回在 RTR 文件中声明的默认值。请参见 [rt_reg\(4\) \[1067\]](#) 手册页。

SCHA_EXTENSION_NODE

输出参数类型为 `scha_extprop_value_t**`。需要类型为 `char*` 的两个附加参数。第一个参数提供扩展属性的名称，第二个参数命名群集节点。返回指定节点的属性类型及其值。

当用户请求未指定明确值的节点上的该属性的值时，将会返回在 RTR 文件中声明的默认值。请参见 [rt_reg\(4\) \[1067\]](#) 手册页。

SCHA_FAILOVER_MODE

输出参数类型为 `scha_failover_mode_t*`。

SCHA_FINI_TIMEOUT

输出参数类型为 `int*`。

SCHA_GLOBAL_ZONE_OVERRIDE

输出参数类型为 `boolean_t*`。

SCHA_GROUP

输出参数类型为 `char**`。返回配置了资源的资源组的名称。

SCHA_INIT_TIMEOUT

输出参数类型为 `int*`。

SCHA_LOAD_BALANCING_POLICY

输出参数类型为 `char**`。

SCHA_LOAD_BALANCING_WEIGHTS

输出参数类型为 `scha_str_array_t**`。

SCHA_MONITOR_CHECK_TIMEOUT

输出参数类型为 `int*`。

SCHA_MONITOR_START_TIMEOUT

输出参数类型为 `int*`。

SCHA_MONITOR_STOP_TIMEOUT

输出参数类型为 `int*`。

SCHA_MONITORED_SWITCH

输出参数类型为 `scha_switch_t*`。返回值指示是否在本节点上监视资源。

SCHA_MONITORED_SWITCH_NODE

输出参数类型为 `scha_switch_t*`。需要命名群集节点的附加参数，其类型为 `char*`。返回值指示是否在指定节点上监视资源。

SCHA_NETWORK_RESOURCES_USED

输出参数类型为 `scha_str_array_t**`。

SCHA_NUM_RESOURCE_RESTARTS

输出参数类型为 `int*`。返回在执行此查询的区域中对该资源发生的资源重新启动请求数。有关更多详细信息，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

SCHA_NUM_RG_RESTARTS

输出参数类型为 `int*`。返回在执行此查询的区域中对该资源发生的资源组重新启动请求数。有关更多详细信息，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

SCHA_ON_OFF_SWITCH

输出参数类型为 `scha_switch_t*`。返回值指示是否在本地节点上启用资源。

SCHA_ON_OFF_SWITCH_NODE

输出参数类型为 `scha_switch_t*`。需要命名群集节点的附加参数，其类型为 `char*`。返回值指示是否在指定节点上启用资源。

SCHA_PORT_LIST

输出参数类型为 `scha_str_array_t**`。

SCHA_POSTNET_STOP_TIMEOUT

输出参数类型为 `int*`。

SCHA_PRENET_START_TIMEOUT

输出参数类型为 `int*`。

SCHA_PRE_EVICT

输出参数类型为 `boolean_t*`。

SCHA_R_DESCRIPTION

输出参数类型为 `char**`。

SCHA_RESOURCE_DEPENDENCIES

输出参数类型为 `scha_str_array_t**`。返回值列出适用于本地节点的依赖性。

SCHA_RESOURCE_DEPENDENCIES_NODE

输出参数类型为 `scha_str_array_t**`。需要命名群集节点的附加参数，其类型为 `char*`。返回值列出适用于指定节点的依赖性。

SCHA_RESOURCE_DEPENDENCIES_OFFLINE_RESTART

输出参数类型为 `scha_str_array_t**`。返回值列出适用于本地节点的依赖性。

SCHA_RESOURCE_DEPENDENCIES_OFFLINE_RESTART_NODE

输出参数类型为 `scha_str_array_t**`。需要命名群集节点的附加参数，其类型为 `char*`。返回值列出适用于指定节点的依赖性。

SCHA_RESOURCE_DEPENDENCIES_Q

输出参数类型为 `scha_str_array_t**`。

SCHA_RESOURCE_DEPENDENCIES 标记的 Q 版本还会返回为资源依赖性声明的作用域或限定符（如果有）。 [Unresolved link to "r_properties\(5\)"](#) 手册页中介绍了 `{LOCAL_NODE}`、`{ANY_NODE}`、`{FROM_RG_AFFINITIES}` 和 `@node` 限定符。

SCHA_RESOURCE_DEPENDENCIES_Q_OFFLINE_RESTART

输出参数类型为 `scha_str_array_t**`。

SCHA_RESOURCE_DEPENDENCIES_OFFLINE_RESTART 标记的 Q 版本还会返回为资源依赖性声明的作用域或限定符（如果有）。 [Unresolved link to "r_properties\(5\)"](#) 手册页中介绍了 `{LOCAL_NODE}`、`{ANY_NODE}`、`{FROM_RG_AFFINITIES}` 和 `@node` 限定符。

SCHA_RESOURCE_DEPENDENCIES_Q_RESTART

输出参数类型为 `scha_str_array_t**`。

SCHA_RESOURCE_DEPENDENCIES_RESTART 标记的 Q 版本还会返回为资源依赖性声明的作用域或限定符（如果有）。 [Unresolved link to "r_properties\(5\)"](#) 手册页中介绍了 `{LOCAL_NODE}`、`{ANY_NODE}`、`{FROM_RG_AFFINITIES}` 和 `@node` 限定符。

SCHA_RESOURCE_DEPENDENCIES_Q_WEAK

输出参数类型为 `scha_str_array_t**`。

SCHA_RESOURCE_DEPENDENCIES_WEAK 标记的 Q 版本还会返回为资源依赖性声明的作用域或限定符（如果有）。 [Unresolved link to "r_properties\(5\)"](#) 手册页中介绍了 `{LOCAL_NODE}`、`{ANY_NODE}`、`{FROM_RG_AFFINITIES}` 和 `@node` 限定符。

SCHA_RESOURCE_DEPENDENCIES_RESTART

输出参数类型为 `scha_str_array_t**`。返回值列出适用于本地节点的依赖性。

SCHA_RESOURCE_DEPENDENCIES_RESTART_NODE

输出参数类型为 `scha_str_array_t**`。需要命名群集节点的附加参数，其类型为 `char*`。返回值列出适用于指定节点的依赖性。

SCHA_RESOURCE_DEPENDENCIES_WEAK

输出参数类型为 `scha_str_array_t**`。返回值列出适用于本地节点的依赖性。

SCHA_RESOURCE_DEPENDENCIES_WEAK_NODE

输出参数类型为 `scha_str_array_t**`。需要命名群集节点的附加参数，其类型为 `char*`。返回值列出适用于指定节点的依赖性。

SCHA_RESOURCE_PROJECT_NAME

输出参数类型为 `char**`。

SCHA_RESOURCE_STATE

输出参数类型为 `scha_rsstate_t*`。返回本地节点的资源的 `RESOURCE_STATE` 属性值。

SCHA_RESOURCE_STATE_NODE

输出参数类型为 `scha_rsstate_t*`。需要命名群集节点的附加参数，其类型为 `char*`。返回指定节点的资源的 `RESOURCE_STATE` 属性值。

SCHA_RETRY_COUNT

输出参数类型为 `int*`。

SCHA_RETRY_INTERVAL

输出参数类型为 `int*`。

SCHA_SCALABLE

输出参数类型为 `boolean_t*`。

SCHA_START_TIMEOUT

输出参数类型为 `int*`。

SCHA_STATUS

输出参数类型为 `scha_status_value_t**`。返回本地节点的资源的 `STATUS` 属性值。

SCHA_STATUS_NODE

输出参数类型为 `scha_status_value_t**`。需要命名群集节点的附加参数，其类型为 `char*`。返回指定节点的资源的 `STATUS` 属性值。

SCHA_STOP_TIMEOUT

输出参数类型为 `int*`。

SCHA_THOROUGH_PROBE_INTERVAL

输出参数类型为 `int*`。

SCHA_TYPE

输出参数类型为 `char**`。

SCHA_TYPE_VERSION

输出参数类型为 char**。

SCHA_UDP_AFFINITY

输出参数类型为 boolean_t*。

SCHA_UPDATE_TIMEOUT

输出参数类型为 int*。

SCHA_VALIDATE_TIMEOUT

输出参数类型为 int*。

SCHA_WEAK_AFFINITY

输出参数类型为 boolean_t*。

下面列出了用于指定资源类型属性名称的宏。输出是资源类型的属性值。有关资源类型属性的说明，请参见 [rt_properties\(5\) \[1143\]](#)。

SCHA_API_VERSION

输出参数类型为 int*。

SCHA_BOOT

输出参数类型为 char**。

SCHA_FAILOVER

输出参数类型为 boolean_t*。

SCHA_FINI

输出参数类型为 char**。

SCHA_GLOBAL_ZONE

输出参数类型为 boolean_t*。

SCHA_INIT

输出参数类型为 char**。

SCHA_INIT_NODES

输出参数类型为 scha_initnodes_flag_t*。

SCHA_INSTALLED_NODES

输出参数类型为 scha_str_array_t**。

SCHA_MONITOR_CHECK

输出参数类型为 char**。

SCHA_MONITOR_START
输出参数类型为 char**。

SCHA_MONITOR_STOP
输出参数类型为 char**。

SCHA_PKGLIST
输出参数类型为 scha_str_array_t**。

SCHA_POSTNET_STOP
输出参数类型为 char**。

SCHA_PRENET_START
输出参数类型为 char**。

SCHA_PROXY
输出参数类型为 boolean_t*。

SCHA_RT_BASEDIR
输出参数类型为 char**。

SCHA_RT_DESCRIPTION
输出参数类型为 char**。

SCHA_RT_SYSTEM
输出参数类型为 boolean_t*。

SCHA_RT_VERSION
输出参数类型为 char**。

SCHA_SINGLE_INSTANCE
输出参数类型为 boolean_t*。

SCHA_START
输出参数类型为 char**。

SCHA_STOP
输出参数类型为 char**。

SCHA_UPDATE
输出参数类型为 char**。

SCHA_VALIDATE
输出参数类型为 char**。

如果该资源类型声明了 GLOBAL_ZONE_OVERRIDE 资源属性，则由 SCHA_GLOBAL_ZONE`optag` 检索的值是 GLOBAL_ZONE_OVERRIDE 属性的当前值，而不是 GLOBAL_ZONE 属性的值。有关更多信息，请参见 [rt_properties\(5\) \[1143\]](#) 手册页中的 Global_zone 属性和 [r_properties\(5\) \[1103\]](#) 手册页中的 Global_zone_override 属性的描述。

这些函数返回以下值：

0 该函数成功。

非零值 该函数失败。

SCHA_ERR_NOERR 该函数成功。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

例 381 使用 `scha_resource_get()` 函数

以下示例使用 `scha_resource_get()` 来获取某个资源的 `Retry_count` 属性的值，以及名为 `LogLevel` 的扩展属性的值。

```
main() {
    #include <scha.h>

    scha_err_t err;
    int retry_count_out;
    scha_extprop_value_t *loglevel_out;
    scha_resource_t handle;

    /* a configured resource */
    char * resource_name = "example_R";
    /* resource group containing example_R */
    char * group_name = "example_RG";

    err = scha_resource_open(resource_name, group_name, &handle);

    err = scha_resource_get(handle, SCHA_RETRY_COUNT, &retry_count_out);

    /* Given extension property must be defined in resourcetype RTR file. */
    err = scha_resource_get(handle, SCHA_EXTENSION, "LogLevel", &loglevel_out);

    err = scha_resource_close(handle);

    printf("The retry count for resource %s is %d\n", resource_name,
           retry_count_out);

    printf("The log level for resource %s is %d\n", resource_name,
           loglevel_out->val.val_int);
}
```

/usr/cluster/include/scha.h 头文件

/usr/cluster/lib/libscha.so 库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scha_resource_get\(1HA\) \[585\]](#)、[scha_calls\(3HA\) \[897\]](#)、[scha_strerror\(3HA\) \[1057\]](#)、[Unresolved link to "attributes5"](#)、[r_properties\(5\) \[1103\]](#)、[rt_properties\(5\) \[1143\]](#)、[rt_reg\(4\) \[1067\]](#)

名称

scha_resource_open, scha_resource_open_zone, scha_resource_get,
scha_resource_get_zone, scha_resource_close — 资源信息访问函数

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib -l scha#include
<scha.h>scha_err_t scha_resource_open( const char *rname, const char *rgname, scha_resource_t
*handle);

scha_err_t scha_resource_open_zone(const char *cluster, const char *rs_name, const char
*rg_name, scha_resource_t * handlep);

scha_err_t scha_resource_get(scha_resource_t handle, const char *tag,...);

scha_err_t scha_resource_get_zone(const char *cluster,
    scha_resource_t handlep, const char *rs_tag, ...);

scha_err_t scha_resource_close(scha_resource_t handle);
```

将 `scha_resource_open()`、`scha_resource_get()` 和 `scha_resource_close()` 函数一起使用可以访问资源组管理器 (Resource Group Manager, RGM) 群集工具管理的资源的相关信息。

`scha_resource_open()` 初始化资源访问，并返回一个将由 `scha_resource_get()` 使用的句柄。

`scha_resource_open()` 的 `rname` 参数指定要访问的资源名称。`rgname` 参数是其中配置了该资源的资源组的名称。如果组名称未知，`rgname` 参数可以是 NULL。但是，如果提供组名称，可以提高该函数的执行效率。`handle` 参数是用于保存函数调用返回值的某个变量的地址。

`scha_resource_get()` 根据 `tag` 参数的指示访问资源信息。`tag` 参数应该是 `scha_tags.h` 头文件中的某个宏定义的一个字符串值。标记后面的参数取决于 `tag` 的值。可能需要在标记后面额外添加一个参数，用于指示要从中检索信息的群集节点或者指示特定于标记的其他信息。参数列表中最后一个参数的类型应该适合用于保存 `tag` 指定的信息。该参数是资源信息的输出参数。如果函数失败，不会为输出参数返回值。

对用于 `scha_resource_get()` 的句柄调用 `scha_resource_close()` 之前，所分配的用于保存 `scha_resource_get()` 返回的信息的内存将保持不变。请注意，使用相同的句柄和标记重复调用 `scha_resource_get()` 会导致分配新的内存。在一个调用中为返回值而分配的空间不会被后续调用覆盖和重新使用。

`scha_resource_close()` 函数接受上次调用 `scha_resource_open()` 时返回的 `handle` 参数。它将使该句柄失效，并释放为了保存使用该句柄执行的 `scha_resource_get()` 调用的返回值而分配的内存。

下文中列出了 `scha_tags.h` 中定义的可以用作 `scha_resource_get()` 的 `tag` 参数的宏。

指示输出参数和任何其他参数的类型。[scha_calls\(3HA\) \[897\]](#) 介绍了结构和 enum 类型。

`scha_resource_open_zone ()` 和 `scha_resource_get_zone ()` 函数的作用分别与 `scha_resource_open ()` 和 `scha_resource_get ()` 相同，且带有一个额外的 `cluster` 参数，用于指定资源组存在于其中且您要在其上执行操作的区域群集的名称。当代码在全局区域中执行，但需要在指定区域群集上操作时，这些函数十分有用。无法在某一区域群集中使用它们来访问其他区域群集。

如果 `scha_resource_open_zone ()` 或 `scha_resource_get_zone ()` 的 `cluster` 参数为 `NULL`，则会对从中执行调用的群集执行查询；换句话说，带有 `NULL` 参数的调用分别等效于 `scha_resource_open ()` 或 `scha_resource_get ()`。

要关闭由 `scha_resource_open_zone ()` 返回的句柄，请使用 `scha_resource_close ()`。不需要 `cluster` 参数。

Tag 参数

下面列出了用于指定资源属性名称的宏。输出是资源的属性值。`SCHA_RESOURCE_STATE`、`SCHA_STATUS`、`SCHA_NUM_RG_RESTARTS` 和 `SCHA_NUM_RESOURCE_RESTARTS` 属性引用执行命令的节点上的值（请参见 [r_properties\(5\) \[1103\]](#)）。

指示输出参数和任何其他参数的类型。

扩展属性

这些属性是在资源类型的资源类型注册 (Resource Type Registration, RTR) 文件中声明的。资源类型的实现定义了这些属性。

`SCHA_AFFINITY_TIMEOUT`

输出参数类型为 `int*`。

`SCHA_ALL_EXTENSIONS`

输出参数类型为 `scha_str_array_t**`。返回资源的所有扩展属性的名称。

`SCHA_APPLICATION_USER`

输出参数类型为 `char**`。

`SCHA_BOOT_TIMEOUT`

输出参数类型为 `int*`。

`SCHA_CHEAP_PROBE_INTERVAL`

输出参数类型为 `int*`。

`SCHA_EXTENSION`

输出参数类型为 `scha_extprop_value_t**`。需要提供扩展属性名称的附加参数，其类型为 `char*`。返回本地节点的属性类型及其值。

当用户请求未指定明确值的节点上的该属性的值时，将会返回在 RTR 文件中声明的默认值。请参见 [rt_reg\(4\) \[1067\]](#) 手册页。

SCHA_EXTENSION_NODE

输出参数类型为 `scha_extprop_value_t**`。需要类型为 `char*` 的两个附加参数。第一个参数提供扩展属性的名称，第二个参数命名群集节点。返回指定节点的属性类型及其值。

当用户请求未指定明确值的节点上的该属性的值时，将会返回在 RTR 文件中声明的默认值。请参见 [rt_reg\(4\) \[1067\]](#) 手册页。

SCHA_FAILOVER_MODE

输出参数类型为 `scha_failover_mode_t*`。

SCHA_FINI_TIMEOUT

输出参数类型为 `int*`。

SCHA_GLOBAL_ZONE_OVERRIDE

输出参数类型为 `boolean_t*`。

SCHA_GROUP

输出参数类型为 `char**`。返回配置了资源的资源组的名称。

SCHA_INIT_TIMEOUT

输出参数类型为 `int*`。

SCHA_LOAD_BALANCING_POLICY

输出参数类型为 `char**`。

SCHA_LOAD_BALANCING_WEIGHTS

输出参数类型为 `scha_str_array_t**`。

SCHA_MONITOR_CHECK_TIMEOUT

输出参数类型为 `int*`。

SCHA_MONITOR_START_TIMEOUT

输出参数类型为 `int*`。

SCHA_MONITOR_STOP_TIMEOUT

输出参数类型为 `int*`。

SCHA_MONITORED_SWITCH

输出参数类型为 `scha_switch_t*`。返回值指示是否在本节点上监视资源。

SCHA_MONITORED_SWITCH_NODE

输出参数类型为 `scha_switch_t*`。需要命名群集节点的附加参数，其类型为 `char*`。返回值指示是否在指定节点上监视资源。

SCHA_NETWORK_RESOURCES_USED

输出参数类型为 `scha_str_array_t**`。

SCHA_NUM_RESOURCE_RESTARTS

输出参数类型为 `int*`。返回在执行此查询的区域中对该资源发生的资源重新启动请求数。有关更多详细信息，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

SCHA_NUM_RG_RESTARTS

输出参数类型为 `int*`。返回在执行此查询的区域中对该资源发生的资源组重新启动请求数。有关更多详细信息，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

SCHA_ON_OFF_SWITCH

输出参数类型为 `scha_switch_t*`。返回值指示是否在本地节点上启用资源。

SCHA_ON_OFF_SWITCH_NODE

输出参数类型为 `scha_switch_t*`。需要命名群集节点的附加参数，其类型为 `char*`。返回值指示是否在指定节点上启用资源。

SCHA_PORT_LIST

输出参数类型为 `scha_str_array_t**`。

SCHA_POSTNET_STOP_TIMEOUT

输出参数类型为 `int*`。

SCHA_PRENET_START_TIMEOUT

输出参数类型为 `int*`。

SCHA_PRE_EVICT

输出参数类型为 `boolean_t*`。

SCHA_R_DESCRIPTION

输出参数类型为 `char**`。

SCHA_RESOURCE_DEPENDENCIES

输出参数类型为 `scha_str_array_t**`。返回值列出适用于本地节点的依赖性。

SCHA_RESOURCE_DEPENDENCIES_NODE

输出参数类型为 `scha_str_array_t**`。需要命名群集节点的附加参数，其类型为 `char*`。返回值列出适用于指定节点的依赖性。

SCHA_RESOURCE_DEPENDENCIES_OFFLINE_RESTART

输出参数类型为 `scha_str_array_t**`。返回值列出适用于本地节点的依赖性。

SCHA_RESOURCE_DEPENDENCIES_OFFLINE_RESTART_NODE

输出参数类型为 `scha_str_array_t**`。需要命名群集节点的附加参数，其类型为 `char*`。返回值列出适用于指定节点的依赖性。

SCHA_RESOURCE_DEPENDENCIES_Q

输出参数类型为 `scha_str_array_t**`。

SCHA_RESOURCE_DEPENDENCIES 标记的 Q 版本还会返回为资源依赖性声明的作用域或限定符（如果有）。 [Unresolved link to "r_properties\(5\)"](#) 手册页中介绍了 `{LOCAL_NODE}`、`{ANY_NODE}`、`{FROM_RG_AFFINITIES}` 和 `@node` 限定符。

SCHA_RESOURCE_DEPENDENCIES_Q_OFFLINE_RESTART

输出参数类型为 `scha_str_array_t**`。

SCHA_RESOURCE_DEPENDENCIES_OFFLINE_RESTART 标记的 Q 版本还会返回为资源依赖性声明的作用域或限定符（如果有）。 [Unresolved link to "r_properties\(5\)"](#) 手册页中介绍了 `{LOCAL_NODE}`、`{ANY_NODE}`、`{FROM_RG_AFFINITIES}` 和 `@node` 限定符。

SCHA_RESOURCE_DEPENDENCIES_Q_RESTART

输出参数类型为 `scha_str_array_t**`。

SCHA_RESOURCE_DEPENDENCIES_RESTART 标记的 Q 版本还会返回为资源依赖性声明的作用域或限定符（如果有）。 [Unresolved link to "r_properties\(5\)"](#) 手册页中介绍了 `{LOCAL_NODE}`、`{ANY_NODE}`、`{FROM_RG_AFFINITIES}` 和 `@node` 限定符。

SCHA_RESOURCE_DEPENDENCIES_Q_WEAK

输出参数类型为 `scha_str_array_t**`。

SCHA_RESOURCE_DEPENDENCIES_WEAK 标记的 Q 版本还会返回为资源依赖性声明的作用域或限定符（如果有）。 [Unresolved link to "r_properties\(5\)"](#) 手册页中介绍了 `{LOCAL_NODE}`、`{ANY_NODE}`、`{FROM_RG_AFFINITIES}` 和 `@node` 限定符。

SCHA_RESOURCE_DEPENDENCIES_RESTART

输出参数类型为 `scha_str_array_t**`。返回值列出适用于本地节点的依赖性。

SCHA_RESOURCE_DEPENDENCIES_RESTART_NODE

输出参数类型为 `scha_str_array_t**`。需要命名群集节点的附加参数，其类型为 `char*`。返回值列出适用于指定节点的依赖性。

SCHA_RESOURCE_DEPENDENCIES_WEAK

输出参数类型为 `scha_str_array_t**`。返回值列出适用于本地节点的依赖性。

SCHA_RESOURCE_DEPENDENCIES_WEAK_NODE

输出参数类型为 `scha_str_array_t**`。需要命名群集节点的附加参数，其类型为 `char*`。返回值列出适用于指定节点的依赖性。

SCHA_RESOURCE_PROJECT_NAME

输出参数类型为 `char**`。

SCHA_RESOURCE_STATE

输出参数类型为 `scha_rsstate_t*`。返回本地节点的资源的 `RESOURCE_STATE` 属性值。

SCHA_RESOURCE_STATE_NODE

输出参数类型为 `scha_rsstate_t*`。需要命名群集节点的附加参数，其类型为 `char*`。返回指定节点的资源的 `RESOURCE_STATE` 属性值。

SCHA_RETRY_COUNT

输出参数类型为 `int*`。

SCHA_RETRY_INTERVAL

输出参数类型为 `int*`。

SCHA_SCALABLE

输出参数类型为 `boolean_t*`。

SCHA_START_TIMEOUT

输出参数类型为 `int*`。

SCHA_STATUS

输出参数类型为 `scha_status_value_t**`。返回本地节点的资源的 `STATUS` 属性值。

SCHA_STATUS_NODE

输出参数类型为 `scha_status_value_t**`。需要命名群集节点的附加参数，其类型为 `char*`。返回指定节点的资源的 `STATUS` 属性值。

SCHA_STOP_TIMEOUT

输出参数类型为 `int*`。

SCHA_THOROUGH_PROBE_INTERVAL

输出参数类型为 `int*`。

SCHA_TYPE

输出参数类型为 `char**`。

SCHA_TYPE_VERSION

输出参数类型为 char**。

SCHA_UDP_AFFINITY

输出参数类型为 boolean_t*。

SCHA_UPDATE_TIMEOUT

输出参数类型为 int*。

SCHA_VALIDATE_TIMEOUT

输出参数类型为 int*。

SCHA_WEAK_AFFINITY

输出参数类型为 boolean_t*。

下面列出了用于指定资源类型属性名称的宏。输出是资源类型的属性值。有关资源类型属性的说明，请参见 [rt_properties\(5\) \[1143\]](#)。

SCHA_API_VERSION

输出参数类型为 int*。

SCHA_BOOT

输出参数类型为 char**。

SCHA_FAILOVER

输出参数类型为 boolean_t*。

SCHA_FINI

输出参数类型为 char**。

SCHA_GLOBAL_ZONE

输出参数类型为 boolean_t*。

SCHA_INIT

输出参数类型为 char**。

SCHA_INIT_NODES

输出参数类型为 scha_initnodes_flag_t*。

SCHA_INSTALLED_NODES

输出参数类型为 scha_str_array_t**。

SCHA_MONITOR_CHECK

输出参数类型为 char**。

SCHA_MONITOR_START
输出参数类型为 char**。

SCHA_MONITOR_STOP
输出参数类型为 char**。

SCHA_PKGLIST
输出参数类型为 scha_str_array_t**。

SCHA_POSTNET_STOP
输出参数类型为 char**。

SCHA_PRENET_START
输出参数类型为 char**。

SCHA_PROXY
输出参数类型为 boolean_t*。

SCHA_RT_BASEDIR
输出参数类型为 char**。

SCHA_RT_DESCRIPTION
输出参数类型为 char**。

SCHA_RT_SYSTEM
输出参数类型为 boolean_t*。

SCHA_RT_VERSION
输出参数类型为 char**。

SCHA_SINGLE_INSTANCE
输出参数类型为 boolean_t*。

SCHA_START
输出参数类型为 char**。

SCHA_STOP
输出参数类型为 char**。

SCHA_UPDATE
输出参数类型为 char**。

SCHA_VALIDATE
输出参数类型为 char**。

如果该资源类型声明了 GLOBAL_ZONE_OVERRIDE 资源属性，则由 SCHA_GLOBAL_ZONE`optag` 检索的值是 GLOBAL_ZONE_OVERRIDE 属性的当前值，而不是 GLOBAL_ZONE 属性的值。有关更多信息，请参见 [rt_properties\(5\) \[1143\]](#) 手册页中的 Global_zone 属性和 [r_properties\(5\) \[1103\]](#) 手册页中的 Global_zone_override 属性的描述。

这些函数返回以下值：

0 该函数成功。

非零值 该函数失败。

SCHA_ERR_NOERR 该函数成功。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

例 382 使用 `scha_resource_get()` 函数

以下示例使用 `scha_resource_get()` 来获取某个资源的 `Retry_count` 属性的值，以及名为 `LogLevel` 的扩展属性的值。

```
main() {
    #include <scha.h>

    scha_err_t err;
    int retry_count_out;
    scha_extprop_value_t *loglevel_out;
    scha_resource_t handle;

    /* a configured resource */
    char * resource_name = "example_R";
    /* resource group containing example_R */
    char * group_name = "example_RG";

    err = scha_resource_open(resource_name, group_name, &handle);

    err = scha_resource_get(handle, SCHA_RETRY_COUNT, &retry_count_out);

    /* Given extension property must be defined in resourcetype RTR file. */
    err = scha_resource_get(handle, SCHA_EXTENSION, "LogLevel", &loglevel_out);

    err = scha_resource_close(handle);

    printf("The retry count for resource %s is %d\n", resource_name,
           retry_count_out);

    printf("The log level for resource %s is %d\n", resource_name,
           loglevel_out->val.val_int);
}
```

/usr/cluster/include/scha.h 头文件
/usr/cluster/lib/libscha.so 库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scha_resource_get\(1HA\) \[585\]](#)、[scha_calls\(3HA\) \[897\]](#)、[scha_strerror\(3HA\) \[1057\]](#)、[Unresolved link to "attributes5"](#)、[r_properties\(5\) \[1103\]](#)、[rt_properties\(5\) \[1143\]](#)、[rt_reg\(4\) \[1067\]](#)

名称

`scha_resource_setstatus`, `scha_resource_setstatus_zone` — 设置资源状态函数

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l scha#include <scha.h>scha_err_t scha_resource_setstatus(
    const char *rname, const char *rgname, scha_rsstatus_t status,
    const char *status_msg);

scha_err_t scha_resource_setstatus_zone(const char *rname,
    const char *rgname, const char *zonename, scha_rsstatus_t
    status, const char *status_msg);
```

`scha_resource_setstatus()` 和 `scha_resource_setstatus_zone()` 函数可设置资源组管理器 (Resource Group Manager, RGM) 管理的资源的 `Status` 和 `Status_msg` 属性。资源的监视器使用这些函数来指示它所识别的资源状态。

请仅对 `Global_zone` 属性设置为 `TRUE` 的资源类型使用 `scha_resource_setstatus_zone()` 函数。如果 `Global_zone` 属性设置为 `FALSE`，则不需要此函数。有关更多信息，请参见 [rt_properties\(5\) \[1143\]](#) 手册页。

`rname` 参数指定要设置其状态的资源的名称。

`rgname` 参数是包含该资源的资源组的名称。

`zonename` 参数是资源组配置为在其中运行的区域群集的名称。如果 `Global_zone` 属性设置为 `TRUE`，则即使包含资源的资源组在区域群集中运行，也会在全局区域中执行方法。

`status` 参数是一个 `scha_rsstatus_t` 类型的 enum 值：`SCHA_RSSTATUS_OK`、`SCHA_RSSTATUS_OFFLINE`、`SCHA_RSSTATUS_FAULTED`、`SCHA_RSSTATUS_DEGRADED` 或 `SCHA_RSSTATUS_UNKNOWN`。

`status-msg` 参数是 `Status_msg` 属性的新值。`status-msg` 参数可以是 `NULL`。

`scha_resource_setstatus()` 或 `scha_resource_setstatus_zone()` 函数调用成功会导致使用所提供的值更新资源的 `Status` 和 `Status_msg` 属性。资源状态的更新将记录到群集系统日志中，并可通过群集管理工具进行访问。

`scha_resource_setstatus()` 和 `scha_resource_setstatus_zone()` 函数返回以下值：

0	该函数成功。
非零值	该函数失败。

SCHA_ERR_NOERR

该函数成功。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

例 383 使用 `scha_resource_setstatus()` 函数

```
#include <scha.h>

scha_err_t err_code;
const char *rname = "example_R";
const char *rgname = "example_RG";

err_code = scha_resource_setstatus(rname, rgname,
    SCHA_RSSTATUS_OK, "No problems");
```

`/usr/cluster/include/scha.h`

头文件

`/usr/cluster/lib/libscha.so`

库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scha_resource_setstatus\(1HA\) \[593\]](#)、[scha_calls\(3HA\) \[897\]](#)、[scha_strerror\(3HA\) \[1057\]](#)、[Unresolved link to "attributes5"](#)、[rt_properties\(5\) \[1143\]](#)

名称

`scha_resource_setstatus`, `scha_resource_setstatus_zone` — 设置资源状态函数

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib
-l scha#include <scha.h>scha_err_t scha_resource_setstatus(
    const char *rname, const char *rgname, scha_rsstatus_t status,
    const char *status_msg);

scha_err_t scha_resource_setstatus_zone(const char *rname,
    const char *rgname, const char *zonename, scha_rsstatus_t
    status, const char *status_msg);
```

`scha_resource_setstatus()` 和 `scha_resource_setstatus_zone()` 函数可设置资源组管理器 (Resource Group Manager, RGM) 管理的资源的 `Status` 和 `Status_msg` 属性。资源的监视器使用这些函数来指示它所识别的资源状态。

请仅对 `Global_zone` 属性设置为 `TRUE` 的资源类型使用 `scha_resource_setstatus_zone()` 函数。如果 `Global_zone` 属性设置为 `FALSE`，则不需要此函数。有关更多信息，请参见 [rt_properties\(5\) \[1143\]](#) 手册页。

`rname` 参数指定要设置其状态的资源的名称。

`rgname` 参数是包含该资源的资源组的名称。

`zonename` 参数是资源组配置为在其中运行的区域群集的名称。如果 `Global_zone` 属性设置为 `TRUE`，则即使包含资源的资源组在区域群集中运行，也会在全局区域中执行方法。

`status` 参数是一个 `scha_rsstatus_t` 类型的 enum 值：`SCHA_RSSTATUS_OK`、`SCHA_RSSTATUS_OFFLINE`、`SCHA_RSSTATUS_FAULTED`、`SCHA_RSSTATUS_DEGRADED` 或 `SCHA_RSSTATUS_UNKNOWN`。

`status-msg` 参数是 `Status_msg` 属性的新值。`status-msg` 参数可以是 `NULL`。

`scha_resource_setstatus()` 或 `scha_resource_setstatus_zone()` 函数调用成功会导致使用所提供的值更新资源的 `Status` 和 `Status_msg` 属性。资源状态的更新将记录到群集系统日志中，并可通过群集管理工具进行访问。

`scha_resource_setstatus()` 和 `scha_resource_setstatus_zone()` 函数返回以下值：

0	该函数成功。
非零值	该函数失败。

SCHA_ERR_NOERR

该函数成功。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

例 384 使用 `scha_resource_setstatus()` 函数

```
#include <scha.h>
```

```
scha_err_t err_code;  
const char *rname = "example_R";  
const char *rgname = "example_RG";
```

```
err_code = scha_resource_setstatus(rname, rgname,  
    SCHA_RSSTATUS_OK, "No problems");
```

`/usr/cluster/include/scha.h`

头文件

`/usr/cluster/lib/libscha.so`

库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scha_resource_setstatus\(1HA\) \[593\]](#)、[scha_calls\(3HA\) \[897\]](#)、[scha_strerror\(3HA\) \[1057\]](#)、[Unresolved link to "attributes5"](#)、[rt_properties\(5\) \[1143\]](#)

名称

`scha_resourcegroup_open`, `scha_resourcegroup_open_zone`,
`scha_resourcegroup_get`, `scha_resourcegroup_get_zone`, `scha_resourcegroup_close`
— 资源信息访问函数

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib -l scha#include
<scha.h>scha_err_t scha_resourcegroup_open( const char *rgname, scha_resourcegroup_t
*handle);

scha_err_t scha_resourcegroup_open_zone( const char *cluster, const char
*rg_name, scha_resourcegroup_t * handlep);

scha_err_t scha_resourcegroup_close(scha_resourcegroup_t handle);

scha_err_t scha_resourcegroup_get(scha_resourcegroup_t handle, const char *tag...);

scha_err_t scha_resourcegroup_get_zone(const char *cluster,
    scha_resourcegroup_t handlep, const char *rg_tag, ...);
```

将 `scha_resourcegroup_open` ()、`scha_resourcegroup_get` () 和 `scha_resourcegroup_close` () 函数一起使用可以访问资源组管理器 (Resource Group Manager, RGM) 群集工具管理的资源组的相关信息。

`scha_resourcegroup_open` () 初始化资源组访问，并返回一个将由 `scha_resourcegroup_get` () 使用的句柄。

`rgname` 参数指定要访问的资源组的名称。

`handle` 参数是用于保存函数返回值的某个变量的地址。

`scha_resourcegroup_get` () 根据 `tag` 参数的指示访问资源组信息。`tag` 应该是 `scha_tags.h` 头文件中的某个宏定义的一个字符串值。标记后面的参数取决于 `tag` 的值。可能需要在标记后面额外添加一个参数，用于指定要从中检索信息的群集节点。

参数列表中最后一个参数的类型应该适合用于保存 `tag` 指定的信息。该参数是要检索的资源组信息的输出参数。如果函数失败，不会为输出参数返回值。对用于 `scha_resourcegroup_get` () 的句柄调用 `scha_resourcegroup_close` () 之前，所分配的用于保存 `scha_resourcegroup_get` () 返回的信息的内存将保持不变。

`scha_resourcegroup_close` () 接受上次调用 `scha_resourcegroup_open` () 时返回的 `handle` 参数。它将使句柄失效，并释放为了保存使用该句柄执行的 `scha_resourcegroup_get` () 调用的返回值而分配的内存。请注意，如果需要返回值，则为每个 `get` 调用分配内存。在一个调用中为返回值而分配的空间不会被后续调用覆盖或重新使用。

`scha_resourcegroup_open_zone` () 和 `scha_resourcegroup_get_zone` () 函数的作用分别与 `scha_resourcegroup_open` () 和 `scha_resourcegroup_get` () 相同，但带有一

个额外的 `cluster` 参数，用于指定资源组存在于其中且您要在其上执行操作的区域群集的名称。当代码在全局区域中执行，但需要在指定区域群集上操作时，这些函数十分有用。无法在某一区域群集中使用它们来访问其他区域群集。

如果 `scha_resourcegroup_open_zone ()` 或 `scha_resourcegroup_get_zone ()` 的 `cluster` 参数为 `NULL`，则会对执行调用的群集执行查询；换句话说，带有 `NULL` 参数的调用分别等效于 `scha_resourcegroup_open ()` 或 `scha_resourcegroup_get ()`。

要关闭由 `scha_resourcegroup_open_zone ()` 返回的句柄，请使用 `scha_resourcegroup_close ()`。不需要 `cluster` 参数。

可用于 `tag` 参数的宏

可以将 `scha_tags.h` 中定义的下列宏用作 `scha_resourcegroup_get ()` 函数的 `tag` 参数。这些宏用于指定资源组属性的名称。将生成资源组属性的值。RG_STATE 属性在调用函数的节点上引用该值。

指示输出参数和任何其他参数的类型。[scha_calls\(3HA\) \[897\]](#) 手册页中介绍了结构和 `enum` 类型。

SCHA_ALL_LOAD_FACTORS

输出参数类型为 `scha_str_array_t**`。

SCHA_ALL_LOAD_FACTOR_NAMES

输出参数类型为 `scha_str_array_t**`。

SCHA_DESIRED_PRIMARYES

输出参数类型为 `int*`。

SCHA_FAILBACK

输出参数类型为 `boolean_t*`。

SCHA_LOAD_FACTOR

输出参数类型为 `int*`。

SCHA_GLOBAL_RESOURCES_USED

输出参数类型为 `scha_str_array_t**`。

SCHA_IMPL_NET_DEPEND

输出参数类型为 `boolean_t*`。

SCHA_MAXIMUM_PRIMARYES

输出参数类型为 `int*`。

SCHA_NODELIST

输出参数类型为 `scha_str_array_t**`。

SCHA_PATHPREFIX

输出参数类型为 char**。

SCHA_PINGPONG_INTERVAL

输出参数类型为 int*。

SCHA_PREEMPTION_MODE

输出参数类型为 scha_rg_preemption_mode_t*。

SCHA_PRIORITY

输出参数类型为 int*。

SCHA_RESOURCE_LIST

输出参数类型为 scha_str_array_t**。

SCHA_RG_AFFINITIES

输出参数类型为 char**。

SCHA_RG_AUTO_START

输出参数类型为 boolean_t*。

SCHA_RG_DEPENDENCIES

输出参数类型为 scha_str_array_t**。

SCHA_RG_DESCRIPTION

输出参数类型为 char**。

SCHA_RG_IS_FROZEN

输出参数类型为 boolean_t*。

SCHA_RG_MODE

输出参数类型为 scha_rgmode_t*。

SCHA_RG_PROJECT_NAME

输出参数类型为 char**。

SCHA_RG_SLM_CPU

输出参数类型为 char**。

SCHA_RG_SLM_CPU_MIN

输出参数类型为 char**。

SCHA_RG_SLM_PSET_TYPE

输出参数类型为 char**。

SCHA_RG_SLM_TYPE

输出参数类型为 char**。

SCHA_RG_STATE

输出参数类型为 scha_rgstate_t*。

SCHA_RG_STATE_NODE

输出参数类型为 scha_rgstate_t*。一个附加参数类型为 char*。该附加参数指定群集节点的名称，并返回该节点上的资源组的状态。

SCHA_RG_SUSP_AUTO_RECOVERY

输出参数类型为 boolean_t*。

SCHA_RG_SYSTEM

输出参数类型为 boolean_t*。

SCHA_TARGET_NODES

输出参数类型为 scha_str_array_t**。

这些函数返回下列值：

0 该函数成功。

非零值 该函数失败。

SCHA_ERR_NOERR 该函数成功。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

例 385 使用 `scha_resourcegroup_get ()` 函数

以下示例使用 `scha_resourcegroup_get ()` 来获取资源组 `example_RG` 中的资源的列表。

```
main() {
    #include <scha.h>

    scha_err_t err;
    scha_str_array_t *resource_list;
    scha_resourcegroup_t handle;
    int ix;

    char * rgname = "example_RG";

    err = scha_resourcegroup_open(rgname, &handle);
```

```

err = scha_resourcegroup_get(handle, SCHA_RESOURCE_LIST, \
    &resource_list);

if (err == SCHA_ERR_NOERR) {
    for (ix = 0; ix < resource_list->array_cnt; ix++) {
        printf("Group: %s contains resource %s\n", rgroup,
            resource_list->str_array[ix]);
    }
}

/* resource_list memory freed */
err = scha_resourcegroup_close(handle);
}

```

/usr/cluster/include/scha.h 头文件

/usr/cluster/lib/libscha.so 库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[clnode\(1CL\) \[149\]](#)、[scha_resourcegroup_get\(1HA\) \[597\]](#)、[scha_calls\(3HA\) \[897\]](#)、[Unresolved link to "attributes5"](#)

名称

scha_resourcegroup_open, scha_resourcegroup_open_zone,
scha_resourcegroup_get, scha_resourcegroup_get_zone, scha_resourcegroup_close
— 资源信息访问函数

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib -l scha#include  
<scha.h>scha_err_t scha_resourcegroup_open( const char *rgname, scha_resourcegroup_t  
*handle);  
  
scha_err_t scha_resourcegroup_open_zone( const char *cluster, const char  
*rg_name, scha_resourcegroup_t * handlep);  
  
scha_err_t scha_resourcegroup_close(scha_resourcegroup_t handle);  
  
scha_err_t scha_resourcegroup_get(scha_resourcegroup_t handle, const char *tag...);  
  
scha_err_t scha_resourcegroup_get_zone(const char *cluster,  
scha_resourcegroup_t handlep, const char *rg_tag, ...);
```

将 `scha_resourcegroup_open()`、`scha_resourcegroup_get()` 和 `scha_resourcegroup_close()` 函数一起使用可以访问资源组管理器 (Resource Group Manager, RGM) 群集工具管理的资源组的相关信息。

`scha_resourcegroup_open()` 初始化资源组访问，并返回一个将由 `scha_resourcegroup_get()` 使用的句柄。

`rgname` 参数指定要访问的资源组的名称。

`handle` 参数是用于保存函数返回值的某个变量的地址。

`scha_resourcegroup_get()` 根据 `tag` 参数的指示访问资源组信息。`tag` 应该是 `scha_tags.h` 头文件中的某个宏定义的一个字符串值。标记后面的参数取决于 `tag` 的值。可能需要在标记后面额外添加一个参数，用于指定要从中检索信息的群集节点。

参数列表中最后一个参数的类型应该适合用于保存 `tag` 指定的信息。该参数是要检索的资源组信息的输出参数。如果函数失败，不会为输出参数返回值。对用于 `scha_resourcegroup_get()` 的句柄调用 `scha_resourcegroup_close()` 之前，所分配的用于保存 `scha_resourcegroup_get()` 返回的信息的内存将保持不变。

`scha_resourcegroup_close()` 接受上次调用 `scha_resourcegroup_open()` 时返回的 `handle` 参数。它将使句柄失效，并释放为了保存使用该句柄执行的 `scha_resourcegroup_get()` 调用的返回值而分配的内存。请注意，如果需要返回值，则为每个 `get` 调用分配内存。在一个调用中为返回值而分配的空间不会被后续调用覆盖或重新使用。

`scha_resourcegroup_open_zone()` 和 `scha_resourcegroup_get_zone()` 函数的作用分别与 `scha_resourcegroup_open()` 和 `scha_resourcegroup_get()` 相同，但带有一

个额外的 `cluster` 参数，用于指定资源组存在于其中且您要在其上执行操作的区域群集的名称。当代码在全局区域中执行，但需要在指定区域群集上操作时，这些函数十分有用。无法在某一区域群集中使用它们来访问其他区域群集。

如果 `scha_resourcegroup_open_zone ()` 或 `scha_resourcegroup_get_zone ()` 的 `cluster` 参数为 `NULL`，则会对执行调用的群集执行查询；换句话说，带有 `NULL` 参数的调用分别等效于 `scha_resourcegroup_open ()` 或 `scha_resourcegroup_get ()`。

要关闭由 `scha_resourcegroup_open_zone ()` 返回的句柄，请使用 `scha_resourcegroup_close ()`。不需要 `cluster` 参数。

可用于 *tag* 参数的宏

可以将 `scha_tags.h` 中定义的下列宏用作 `scha_resourcegroup_get ()` 函数的 *tag* 参数。这些宏用于指定资源组属性的名称。将生成资源组属性的值。RG_STATE 属性在调用函数的节点上引用该值。

指示输出参数和任何其他参数的类型。[scha_calls\(3HA\) \[897\]](#) 手册页中介绍了结构和 `enum` 类型。

SCHA_ALL_LOAD_FACTORS

输出参数类型为 `scha_str_array_t**`。

SCHA_ALL_LOAD_FACTOR_NAMES

输出参数类型为 `scha_str_array_t**`。

SCHA_DESIRED_PRIMARYES

输出参数类型为 `int*`。

SCHA_FAILBACK

输出参数类型为 `boolean_t*`。

SCHA_LOAD_FACTOR

输出参数类型为 `int*`。

SCHA_GLOBAL_RESOURCES_USED

输出参数类型为 `scha_str_array_t**`。

SCHA_IMPL_NET_DEPEND

输出参数类型为 `boolean_t*`。

SCHA_MAXIMUM_PRIMARYES

输出参数类型为 `int*`。

SCHA_NODELIST

输出参数类型为 `scha_str_array_t**`。

SCHA_PATHPREFIX

输出参数类型为 char**。

SCHA_PINGPONG_INTERVAL

输出参数类型为 int*。

SCHA_PREEMPTION_MODE

输出参数类型为 scha_rg_preemption_mode_t*。

SCHA_PRIORITY

输出参数类型为 int*。

SCHA_RESOURCE_LIST

输出参数类型为 scha_str_array_t**。

SCHA_RG_AFFINITIES

输出参数类型为 char**。

SCHA_RG_AUTO_START

输出参数类型为 boolean_t*。

SCHA_RG_DEPENDENCIES

输出参数类型为 scha_str_array_t**。

SCHA_RG_DESCRIPTION

输出参数类型为 char**。

SCHA_RG_IS_FROZEN

输出参数类型为 boolean_t*。

SCHA_RG_MODE

输出参数类型为 scha_rgmode_t*。

SCHA_RG_PROJECT_NAME

输出参数类型为 char**。

SCHA_RG_SLM_CPU

输出参数类型为 char**。

SCHA_RG_SLM_CPU_MIN

输出参数类型为 char**。

SCHA_RG_SLM_PSET_TYPE

输出参数类型为 char**。

SCHA_RG_SLM_TYPE

输出参数类型为 char**。

SCHA_RG_STATE

输出参数类型为 scha_rgstate_t*。

SCHA_RG_STATE_NODE

输出参数类型为 scha_rgstate_t*。一个附加参数类型为 char*。该附加参数指定群集节点的名称，并返回该节点上的资源组的状态。

SCHA_RG_SUSP_AUTO_RECOVERY

输出参数类型为 boolean_t*。

SCHA_RG_SYSTEM

输出参数类型为 boolean_t*。

SCHA_TARGET_NODES

输出参数类型为 scha_str_array_t**。

这些函数返回下列值：

0 该函数成功。

非零值 该函数失败。

SCHA_ERR_NOERR 该函数成功。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

例 386 使用 `scha_resourcegroup_get ()` 函数

以下示例使用 `scha_resourcegroup_get ()` 来获取资源组 `example_RG` 中的资源的列表。

```
main() {
    #include <scha.h>

    scha_err_t err;
    scha_str_array_t *resource_list;
    scha_resourcegroup_t handle;
    int ix;

    char * rgname = "example_RG";

    err = scha_resourcegroup_open(rgname, &handle);
```

```

err = scha_resourcegroup_get(handle, SCHA_RESOURCE_LIST, \
    &resource_list);

if (err == SCHA_ERR_NOERR) {
    for (ix = 0; ix < resource_list->array_cnt; ix++) {
        printf("Group: %s contains resource %s\n", rgname,
            resource_list->str_array[ix]);
    }
}

/* resource_list memory freed */
err = scha_resourcegroup_close(handle);
}

```

/usr/cluster/include/scha.h 头文件

/usr/cluster/lib/libscha.so 库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[clnode\(1CL\) \[149\]](#)、[scha_resourcegroup_get\(1HA\) \[597\]](#)、[scha_calls\(3HA\) \[897\]](#)、[Unresolved link to "attributes5"](#)

名称

scha_resourcegroup_open, scha_resourcegroup_open_zone,
scha_resourcegroup_get, scha_resourcegroup_get_zone, scha_resourcegroup_close
— 资源信息访问函数

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib -l scha#include  
<scha.h>scha_err_t scha_resourcegroup_open( const char *rgname, scha_resourcegroup_t  
*handle);  
  
scha_err_t scha_resourcegroup_open_zone( const char *cluster, const char  
*rg_name, scha_resourcegroup_t * handlep);  
  
scha_err_t scha_resourcegroup_close(scha_resourcegroup_t handle);  
  
scha_err_t scha_resourcegroup_get(scha_resourcegroup_t handle, const char *tag...);  
  
scha_err_t scha_resourcegroup_get_zone(const char *cluster,  
scha_resourcegroup_t handlep, const char *rg_tag, ...);
```

将 `scha_resourcegroup_open()`、`scha_resourcegroup_get()` 和 `scha_resourcegroup_close()` 函数一起使用可以访问资源组管理器 (Resource Group Manager, RGM) 群集工具管理的资源组的相关信息。

`scha_resourcegroup_open()` 初始化资源组访问，并返回一个将由 `scha_resourcegroup_get()` 使用的句柄。

`rgname` 参数指定要访问的资源组的名称。

`handle` 参数是用于保存函数返回值的某个变量的地址。

`scha_resourcegroup_get()` 根据 `tag` 参数的指示访问资源组信息。`tag` 应该是 `scha_tags.h` 头文件中的某个宏定义的一个字符串值。标记后面的参数取决于 `tag` 的值。可能需要在标记后面额外添加一个参数，用于指定要从中检索信息的群集节点。

参数列表中最后一个参数的类型应该适合用于保存 `tag` 指定的信息。该参数是要检索的资源组信息的输出参数。如果函数失败，不会为输出参数返回值。对用于 `scha_resourcegroup_get()` 的句柄调用 `scha_resourcegroup_close()` 之前，所分配的用于保存 `scha_resourcegroup_get()` 返回的信息的内存将保持不变。

`scha_resourcegroup_close()` 接受上次调用 `scha_resourcegroup_open()` 时返回的 `handle` 参数。它将使句柄失效，并释放为了保存使用该句柄执行的 `scha_resourcegroup_get()` 调用的返回值而分配的内存。请注意，如果需要返回值，则为每个 `get` 调用分配内存。在一个调用中为返回值而分配的空间不会被后续调用覆盖或重新使用。

`scha_resourcegroup_open_zone()` 和 `scha_resourcegroup_get_zone()` 函数的作用分别与 `scha_resourcegroup_open()` 和 `scha_resourcegroup_get()` 相同，但带有一

个额外的 `cluster` 参数，用于指定资源组存在于其中且您要在其上执行操作的区域群集的名称。当代码在全局区域中执行，但需要在指定区域群集上操作时，这些函数十分有用。无法在某一区域群集中使用它们来访问其他区域群集。

如果 `scha_resourcegroup_open_zone ()` 或 `scha_resourcegroup_get_zone ()` 的 `cluster` 参数为 `NULL`，则会对执行调用的群集执行查询；换句话说，带有 `NULL` 参数的调用分别等效于 `scha_resourcegroup_open ()` 或 `scha_resourcegroup_get ()`。

要关闭由 `scha_resourcegroup_open_zone ()` 返回的句柄，请使用 `scha_resourcegroup_close ()`。不需要 `cluster` 参数。

可用于 *tag* 参数的宏

可以将 `scha_tags.h` 中定义的下列宏用作 `scha_resourcegroup_get ()` 函数的 *tag* 参数。这些宏用于指定资源组属性的名称。将生成资源组属性的值。RG_STATE 属性在调用函数的节点上引用该值。

指示输出参数和任何其他参数的类型。[scha_calls\(3HA\) \[897\]](#) 手册页中介绍了结构和 `enum` 类型。

SCHA_ALL_LOAD_FACTORS

输出参数类型为 `scha_str_array_t**`。

SCHA_ALL_LOAD_FACTOR_NAMES

输出参数类型为 `scha_str_array_t**`。

SCHA_DESIRED_PRIMARYES

输出参数类型为 `int*`。

SCHA_FAILBACK

输出参数类型为 `boolean_t*`。

SCHA_LOAD_FACTOR

输出参数类型为 `int*`。

SCHA_GLOBAL_RESOURCES_USED

输出参数类型为 `scha_str_array_t**`。

SCHA_IMPL_NET_DEPEND

输出参数类型为 `boolean_t*`。

SCHA_MAXIMUM_PRIMARYES

输出参数类型为 `int*`。

SCHA_NODELIST

输出参数类型为 `scha_str_array_t**`。

SCHA_PATHPREFIX

输出参数类型为 char**。

SCHA_PINGPONG_INTERVAL

输出参数类型为 int*。

SCHA_PREEMPTION_MODE

输出参数类型为 scha_rg_preemption_mode_t*。

SCHA_PRIORITY

输出参数类型为 int*。

SCHA_RESOURCE_LIST

输出参数类型为 scha_str_array_t**。

SCHA_RG_AFFINITIES

输出参数类型为 char**。

SCHA_RG_AUTO_START

输出参数类型为 boolean_t*。

SCHA_RG_DEPENDENCIES

输出参数类型为 scha_str_array_t**。

SCHA_RG_DESCRIPTION

输出参数类型为 char**。

SCHA_RG_IS_FROZEN

输出参数类型为 boolean_t*。

SCHA_RG_MODE

输出参数类型为 scha_rgmode_t*。

SCHA_RG_PROJECT_NAME

输出参数类型为 char**。

SCHA_RG_SLM_CPU

输出参数类型为 char**。

SCHA_RG_SLM_CPU_MIN

输出参数类型为 char**。

SCHA_RG_SLM_PSET_TYPE

输出参数类型为 char**。

SCHA_RG_SLM_TYPE

输出参数类型为 char**。

SCHA_RG_STATE

输出参数类型为 scha_rgstate_t*。

SCHA_RG_STATE_NODE

输出参数类型为 scha_rgstate_t*。一个附加参数类型为 char*。该附加参数指定群集节点的名称，并返回该节点上的资源组的状态。

SCHA_RG_SUSP_AUTO_RECOVERY

输出参数类型为 boolean_t*。

SCHA_RG_SYSTEM

输出参数类型为 boolean_t*。

SCHA_TARGET_NODES

输出参数类型为 scha_str_array_t**。

这些函数返回下列值：

0 该函数成功。

非零值 该函数失败。

SCHA_ERR_NOERR 该函数成功。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

例 387 使用 `scha_resourcegroup_get ()` 函数

以下示例使用 `scha_resourcegroup_get ()` 来获取资源组 `example_RG` 中的资源的列表。

```
main() {
    #include <scha.h>

    scha_err_t err;
    scha_str_array_t *resource_list;
    scha_resourcegroup_t handle;
    int ix;

    char * rgname = "example_RG";

    err = scha_resourcegroup_open(rgname, &handle);
```

```

err = scha_resourcegroup_get(handle, SCHA_RESOURCE_LIST, \
    &resource_list);

if (err == SCHA_ERR_NOERR) {
    for (ix = 0; ix < resource_list->array_cnt; ix++) {
        printf("Group: %s contains resource %s\n", rgname,
            resource_list->str_array[ix]);
    }
}

/* resource_list memory freed */
err = scha_resourcegroup_close(handle);
}

```

/usr/cluster/include/scha.h 头文件

/usr/cluster/lib/libscha.so 库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[clnode\(1CL\) \[149\]](#)、[scha_resourcegroup_get\(1HA\) \[597\]](#)、[scha_calls\(3HA\) \[897\]](#)、[Unresolved link to "attributes5"](#)

名称

`scha_resourcegroup_open`, `scha_resourcegroup_open_zone`,
`scha_resourcegroup_get`, `scha_resourcegroup_get_zone`, `scha_resourcegroup_close`
— 资源信息访问函数

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib -l scha#include
<scha.h>scha_err_t scha_resourcegroup_open( const char *rgname, scha_resourcegroup_t
*handle);

scha_err_t scha_resourcegroup_open_zone( const char *cluster, const char
*rg_name, scha_resourcegroup_t * handlep);

scha_err_t scha_resourcegroup_close(scha_resourcegroup_t handle);

scha_err_t scha_resourcegroup_get(scha_resourcegroup_t handle, const char *tag...);

scha_err_t scha_resourcegroup_get_zone(const char *cluster,
scha_resourcegroup_t handlep, const char *rg_tag, ...);
```

将 `scha_resourcegroup_open` ()、`scha_resourcegroup_get` () 和 `scha_resourcegroup_close` () 函数一起使用可以访问资源组管理器 (Resource Group Manager, RGM) 群集工具管理的资源组的相关信息。

`scha_resourcegroup_open` () 初始化资源组访问，并返回一个将由 `scha_resourcegroup_get` () 使用的句柄。

rgname 参数指定要访问的资源组的名称。

handle 参数是用于保存函数返回值的某个变量的地址。

`scha_resourcegroup_get` () 根据 *tag* 参数的指示访问资源组信息。*tag* 应该是 `scha_tags.h` 头文件中的某个宏定义的一个字符串值。标记后面的参数取决于 *tag* 的值。可能需要在标记后面额外添加一个参数，用于指定要从中检索信息的群集节点。

参数列表中最后一个参数的类型应该适合用于保存 *tag* 指定的信息。该参数是要检索的资源组信息的输出参数。如果函数失败，不会为输出参数返回值。对用于 `scha_resourcegroup_get` () 的句柄调用 `scha_resourcegroup_close` () 之前，所分配的用于保存 `scha_resourcegroup_get` () 返回的信息的内存将保持不变。

`scha_resourcegroup_close` () 接受上次调用 `scha_resourcegroup_open` () 时返回的 *handle* 参数。它将使句柄失效，并释放为了保存使用该句柄执行的 `scha_resourcegroup_get` () 调用的返回值而分配的内存。请注意，如果需要返回值，则为每个 `get` 调用分配内存。在一个调用中为返回值而分配的空间不会被后续调用覆盖或重新使用。

`scha_resourcegroup_open_zone` () 和 `scha_resourcegroup_get_zone` () 函数的作用分别与 `scha_resourcegroup_open` () 和 `scha_resourcegroup_get` () 相同，但带有一

个额外的 `cluster` 参数，用于指定资源组存在于其中且您要在其上执行操作的区域群集的名称。当代码在全局区域中执行，但需要在指定区域群集上操作时，这些函数十分有用。无法在某一区域群集中使用它们来访问其他区域群集。

如果 `scha_resourcegroup_open_zone ()` 或 `scha_resourcegroup_get_zone ()` 的 `cluster` 参数为 `NULL`，则会对执行调用的群集执行查询；换句话说，带有 `NULL` 参数的调用分别等效于 `scha_resourcegroup_open ()` 或 `scha_resourcegroup_get ()`。

要关闭由 `scha_resourcegroup_open_zone ()` 返回的句柄，请使用 `scha_resourcegroup_close ()`。不需要 `cluster` 参数。

可用于 `tag` 参数的宏

可以将 `scha_tags.h` 中定义的下列宏用作 `scha_resourcegroup_get ()` 函数的 `tag` 参数。这些宏用于指定资源组属性的名称。将生成资源组属性的值。`RG_STATE` 属性在调用函数的节点上引用该值。

指示输出参数和任何其他参数的类型。[scha_calls\(3HA\) \[897\]](#) 手册页中介绍了结构和 `enum` 类型。

`SCHA_ALL_LOAD_FACTORS`

输出参数类型为 `scha_str_array_t**`。

`SCHA_ALL_LOAD_FACTOR_NAMES`

输出参数类型为 `scha_str_array_t**`。

`SCHA_DESIRED_PRIMARYES`

输出参数类型为 `int*`。

`SCHA_FAILBACK`

输出参数类型为 `boolean_t*`。

`SCHA_LOAD_FACTOR`

输出参数类型为 `int*`。

`SCHA_GLOBAL_RESOURCES_USED`

输出参数类型为 `scha_str_array_t**`。

`SCHA_IMPL_NET_DEPEND`

输出参数类型为 `boolean_t*`。

`SCHA_MAXIMUM_PRIMARYES`

输出参数类型为 `int*`。

`SCHA_NODELIST`

输出参数类型为 `scha_str_array_t**`。

SCHA_PATHPREFIX

输出参数类型为 char**。

SCHA_PINGPONG_INTERVAL

输出参数类型为 int*。

SCHA_PREEMPTION_MODE

输出参数类型为 scha_rg_preemption_mode_t*。

SCHA_PRIORITY

输出参数类型为 int*。

SCHA_RESOURCE_LIST

输出参数类型为 scha_str_array_t**。

SCHA_RG_AFFINITIES

输出参数类型为 char**。

SCHA_RG_AUTO_START

输出参数类型为 boolean_t*。

SCHA_RG_DEPENDENCIES

输出参数类型为 scha_str_array_t**。

SCHA_RG_DESCRIPTION

输出参数类型为 char**。

SCHA_RG_IS_FROZEN

输出参数类型为 boolean_t*。

SCHA_RG_MODE

输出参数类型为 scha_rgmode_t*。

SCHA_RG_PROJECT_NAME

输出参数类型为 char**。

SCHA_RG_SLM_CPU

输出参数类型为 char**。

SCHA_RG_SLM_CPU_MIN

输出参数类型为 char**。

SCHA_RG_SLM_PSET_TYPE

输出参数类型为 char**。

SCHA_RG_SLM_TYPE

输出参数类型为 char**。

SCHA_RG_STATE

输出参数类型为 scha_rgstate_t*。

SCHA_RG_STATE_NODE

输出参数类型为 scha_rgstate_t*。一个附加参数类型为 char*。该附加参数指定群集节点的名称，并返回该节点上的资源组的状态。

SCHA_RG_SUSP_AUTO_RECOVERY

输出参数类型为 boolean_t*。

SCHA_RG_SYSTEM

输出参数类型为 boolean_t*。

SCHA_TARGET_NODES

输出参数类型为 scha_str_array_t**。

这些函数返回下列值：

0 该函数成功。

非零值 该函数失败。

SCHA_ERR_NOERR 该函数成功。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

例 388 使用 `scha_resourcegroup_get ()` 函数

以下示例使用 `scha_resourcegroup_get ()` 来获取资源组 `example_RG` 中的资源的列表。

```
main() {
    #include <scha.h>

    scha_err_t err;
    scha_str_array_t *resource_list;
    scha_resourcegroup_t handle;
    int ix;

    char * rgname = "example_RG";

    err = scha_resourcegroup_open(rgname, &handle);
```

```

err = scha_resourcegroup_get(handle, SCHA_RESOURCE_LIST, \
    &resource_list);

if (err == SCHA_ERR_NOERR) {
    for (ix = 0; ix < resource_list->array_cnt; ix++) {
        printf("Group: %s contains resource %s\n", rgname,
            resource_list->str_array[ix]);
    }
}

/* resource_list memory freed */
err = scha_resourcegroup_close(handle);
}

```

/usr/cluster/include/scha.h 头文件

/usr/cluster/lib/libscha.so 库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[clnode\(1CL\) \[149\]](#)、[scha_resourcegroup_get\(1HA\) \[597\]](#)、[scha_calls\(3HA\) \[897\]](#)、[Unresolved link to "attributes5"](#)

名称

scha_resourcegroup_open, scha_resourcegroup_open_zone,
scha_resourcegroup_get, scha_resourcegroup_get_zone, scha_resourcegroup_close
— 资源信息访问函数

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib -l scha#include  
<scha.h>scha_err_t scha_resourcegroup_open( const char *rgname, scha_resourcegroup_t  
*handle);  
  
scha_err_t scha_resourcegroup_open_zone( const char *cluster, const char  
*rg_name, scha_resourcegroup_t * handlep);  
  
scha_err_t scha_resourcegroup_close(scha_resourcegroup_t handle);  
  
scha_err_t scha_resourcegroup_get(scha_resourcegroup_t handle, const char *tag...);  
  
scha_err_t scha_resourcegroup_get_zone(const char *cluster,  
scha_resourcegroup_t handlep, const char *rg_tag, ...);
```

将 `scha_resourcegroup_open()`、`scha_resourcegroup_get()` 和 `scha_resourcegroup_close()` 函数一起使用可以访问资源组管理器 (Resource Group Manager, RGM) 群集工具管理的资源组的相关信息。

`scha_resourcegroup_open()` 初始化资源组访问，并返回一个将由 `scha_resourcegroup_get()` 使用的句柄。

`rgname` 参数指定要访问的资源组的名称。

`handle` 参数是用于保存函数返回值的某个变量的地址。

`scha_resourcegroup_get()` 根据 `tag` 参数的指示访问资源组信息。`tag` 应该是 `scha_tags.h` 头文件中的某个宏定义的一个字符串值。标记后面的参数取决于 `tag` 的值。可能需要在标记后面额外添加一个参数，用于指定要从中检索信息的群集节点。

参数列表中最后一个参数的类型应该适合用于保存 `tag` 指定的信息。该参数是要检索的资源组信息的输出参数。如果函数失败，不会为输出参数返回值。对用于 `scha_resourcegroup_get()` 的句柄调用 `scha_resourcegroup_close()` 之前，所分配的用于保存 `scha_resourcegroup_get()` 返回的信息的内存将保持不变。

`scha_resourcegroup_close()` 接受上次调用 `scha_resourcegroup_open()` 时返回的 `handle` 参数。它将使句柄失效，并释放为了保存使用该句柄执行的 `scha_resourcegroup_get()` 调用的返回值而分配的内存。请注意，如果需要返回值，则为每个 `get` 调用分配内存。在一个调用中为返回值而分配的空间不会被后续调用覆盖或重新使用。

`scha_resourcegroup_open_zone()` 和 `scha_resourcegroup_get_zone()` 函数的作用分别与 `scha_resourcegroup_open()` 和 `scha_resourcegroup_get()` 相同，但带有一

个额外的 `cluster` 参数，用于指定资源组存在于其中且您要在其上执行操作的区域群集的名称。当代码在全局区域中执行，但需要在指定区域群集上操作时，这些函数十分有用。无法在某一区域群集中使用它们来访问其他区域群集。

如果 `scha_resourcegroup_open_zone ()` 或 `scha_resourcegroup_get_zone ()` 的 `cluster` 参数为 `NULL`，则会对执行调用的群集执行查询；换句话说，带有 `NULL` 参数的调用分别等效于 `scha_resourcegroup_open ()` 或 `scha_resourcegroup_get ()`。

要关闭由 `scha_resourcegroup_open_zone ()` 返回的句柄，请使用 `scha_resourcegroup_close ()`。不需要 `cluster` 参数。

可用于 `tag` 参数的宏

可以将 `scha_tags.h` 中定义的下列宏用作 `scha_resourcegroup_get ()` 函数的 `tag` 参数。这些宏用于指定资源组属性的名称。将生成资源组属性的值。`RG_STATE` 属性在调用函数的节点上引用该值。

指示输出参数和任何其他参数的类型。[scha_calls\(3HA\) \[897\]](#) 手册页中介绍了结构和 `enum` 类型。

`SCHA_ALL_LOAD_FACTORS`

输出参数类型为 `scha_str_array_t**`。

`SCHA_ALL_LOAD_FACTOR_NAMES`

输出参数类型为 `scha_str_array_t**`。

`SCHA_DESIRED_PRIMARYES`

输出参数类型为 `int*`。

`SCHA_FAILBACK`

输出参数类型为 `boolean_t*`。

`SCHA_LOAD_FACTOR`

输出参数类型为 `int*`。

`SCHA_GLOBAL_RESOURCES_USED`

输出参数类型为 `scha_str_array_t**`。

`SCHA_IMPL_NET_DEPEND`

输出参数类型为 `boolean_t*`。

`SCHA_MAXIMUM_PRIMARYES`

输出参数类型为 `int*`。

`SCHA_NODELIST`

输出参数类型为 `scha_str_array_t**`。

SCHA_PATHPREFIX

输出参数类型为 char**。

SCHA_PINGPONG_INTERVAL

输出参数类型为 int*。

SCHA_PREEMPTION_MODE

输出参数类型为 scha_rg_preemption_mode_t*。

SCHA_PRIORITY

输出参数类型为 int*。

SCHA_RESOURCE_LIST

输出参数类型为 scha_str_array_t**。

SCHA_RG_AFFINITIES

输出参数类型为 char**。

SCHA_RG_AUTO_START

输出参数类型为 boolean_t*。

SCHA_RG_DEPENDENCIES

输出参数类型为 scha_str_array_t**。

SCHA_RG_DESCRIPTION

输出参数类型为 char**。

SCHA_RG_IS_FROZEN

输出参数类型为 boolean_t*。

SCHA_RG_MODE

输出参数类型为 scha_rgmode_t*。

SCHA_RG_PROJECT_NAME

输出参数类型为 char**。

SCHA_RG_SLM_CPU

输出参数类型为 char**。

SCHA_RG_SLM_CPU_MIN

输出参数类型为 char**。

SCHA_RG_SLM_PSET_TYPE

输出参数类型为 char**。

SCHA_RG_SLM_TYPE

输出参数类型为 char**。

SCHA_RG_STATE

输出参数类型为 scha_rgstate_t*。

SCHA_RG_STATE_NODE

输出参数类型为 scha_rgstate_t*。一个附加参数类型为 char*。该附加参数指定群集节点的名称，并返回该节点上的资源组的状态。

SCHA_RG_SUSP_AUTO_RECOVERY

输出参数类型为 boolean_t*。

SCHA_RG_SYSTEM

输出参数类型为 boolean_t*。

SCHA_TARGET_NODES

输出参数类型为 scha_str_array_t**。

这些函数返回下列值：

0 该函数成功。

非零值 该函数失败。

SCHA_ERR_NOERR 该函数成功。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#)。

例 389 使用 `scha_resourcegroup_get ()` 函数

以下示例使用 `scha_resourcegroup_get ()` 来获取资源组 `example_RG` 中的资源的列表。

```
main() {
    #include <scha.h>

    scha_err_t err;
    scha_str_array_t *resource_list;
    scha_resourcegroup_t handle;
    int ix;

    char * rgname = "example_RG";

    err = scha_resourcegroup_open(rgname, &handle);
```

```

err = scha_resourcegroup_get(handle, SCHA_RESOURCE_LIST, \
    &resource_list);

if (err == SCHA_ERR_NOERR) {
    for (ix = 0; ix < resource_list->array_cnt; ix++) {
        printf("Group: %s contains resource %s\n", rgname,
            resource_list->str_array[ix]);
    }
}

/* resource_list memory freed */
err = scha_resourcegroup_close(handle);
}

```

/usr/cluster/include/scha.h 头文件

/usr/cluster/lib/libscha.so 库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[clnode\(1CL\) \[149\]](#)、[scha_resourcegroup_get\(1HA\) \[597\]](#)、[scha_calls\(3HA\) \[897\]](#)、[Unresolved link to "attributes5"](#)

名称

scha_resourcetype_open, scha_resourcetype_open_zone, scha_resourcetype_get, scha_resourcetype_get_zone, scha_resourcetype_close — 资源类型信息访问函数

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib -l scha#include
<scha.h>scha_err_t scha_resourcetype_open( const char *rtname, scha_resourcetype_t *handle);

scha_err_t scha_resourcetype_open_zone(const char *cluster, const char
*rt_name, scha_resourcetype_t *handlep);

scha_err_t scha_resourcetype_close(scha_resourcetype_t handle);

scha_err_t scha_resourcetype_get(scha_resourcetype_t handle, const char *tag...);

scha_err_t scha_resourcetype_get_zone(const char *cluster,
    scha_resourcetype_t handlep, const char *rt_tag, ...);
```

使用 `scha_resourcetype_open()`、`scha_resourcetype_get()` 和 `scha_resourcetype_close()` 函数可以访问资源组管理器 (Resource Group Manager, RGM) 群集工具使用的资源类型的相关信息。

`scha_resourcetype_open()` 初始化资源类型访问，并返回一个将由 `scha_resourcetype_get()` 使用的句柄。

`scha_resourcetype_open()` 的 `rtname` 参数指定要访问的资源类型。

`handle` 参数是用于保存函数调用返回值的某个变量的地址。

`scha_resourcetype_get()` 根据 `tag` 参数的指示访问资源类型信息。`tag` 参数应该是 `scha_tags.h` 头文件中的某个宏定义的一个字符串值。标记后面的参数取决于 `tag` 的值。

可能需要在标记后面额外添加一个参数，用于指示要从中检索信息的群集节点或者指示特定于标记的其他信息。参数列表中最后一个参数的类型应该适合用于保存 `tag` 指定的信息。该参数是资源类型信息的“输出”参数。如果该函数失败，则不会返回该“输出”参数的值。对用于 `scha_resourcetype_get()` 的句柄调用 `scha_resourcetype_close()` 之前，所分配的用于保存 `scha_resourcetype_get()` 返回的信息的内存将保持不变。

`scha_resourcetype_close()` 接受上次调用 `scha_resourcetype_open()` 时返回的 `handle` 参数。该函数将使该句柄失效，并释放为了保存使用该句柄执行的 `scha_resourcetype_get()` 调用的返回值而分配的内存。请注意，如果需要返回值，则会为每个 `get` 调用分配内存。在一个调用中为返回值而分配的空间不会被后续调用覆盖和重新使用。

下文中列出了 `scha_tags.h` 中定义的可以用作 `scha_resourcetype_get()` 的 `tag` 参数的宏。指示输出参数和任何其他参数的类型。[scha_calls\(3HA\) \[897\]](#) 中介绍了结构和 `enum` 类型。

`scha_resourcetype_open_zone ()` 和 `scha_resourcetype_get_zone ()` 函数的作用分别与 `scha_resourcetype_open ()` 和 `scha_resourcetype_get ()` 相同，且带有一个额外的 `cluster` 参数，用于指定资源组存在于其中且您要在其上执行操作的区域群集的名称。当代码在全局区域中执行，但需要在指定区域群集上操作时，这些函数十分有用。无法在某一区域群集中使用它们来访问其他区域群集。

如果 `scha_resourcetype_open_zone ()` 或 `scha_resourcetype_get_zone ()` 的 `cluster` 参数为 `NULL`，则会对从中执行调用的群集执行查询；换句话说，带有 `NULL` 参数的调用分别等效于 `scha_resourcetype_open ()` 或 `scha_resourcetype_get ()`。

要关闭由 `scha_resourcetype_open_zone ()` 返回的句柄，请使用 `scha_resourcetype_close ()`。不需要 `cluster` 参数。

optag 参数

下列宏用于指定资源类型属性名称。输出是资源类型的指定属性的值。

注 - `optag` 参数（例如 `SCHA_API_VERSION` 和 `SCHA_BOOT`）不区分大小写。指定 `optag` 参数时，可以使用大小写字母的任意组合。

`SCHA_API_VERSION`

输出参数的类型为 `int*`。

`SCHA_BOOT`

输出参数的类型为 `char **`。

`SCHA_FAILOVER`

输出参数的类型为 `boolean_t *`。

`SCHA_FINI`

输出参数的类型为 `char **`。

`SCHA_GLOBALZONE`

输出参数的类型为 `boolean_t *`。

`SCHA_INIT`

输出参数的类型为 `char **`。

`SCHA_INIT_NODES`

输出参数的类型为 `scha_initnodes_flag_t *`。

`SCHA_INSTALLED_NODES`

输出参数的类型为 `scha_str_array_t **`。

SCHA_IS_LOGICAL_HOSTNAME

输出参数的类型为 `boolean_t *`。

SCHA_IS_SHARED_ADDRESS

输出参数的类型为 `boolean_t *`。

SCHA_MONITOR_CHECK

输出参数的类型为 `char **`。

SCHA_MONITOR_START

输出参数的类型为 `char **`。

SCHA_MONITOR_STOP

输出参数的类型为 `char **`。

SCHA_PER_NODE

输出参数的类型为 `boolean_t *`。

SCHA_PKGLIST

输出参数的类型为 `scha_str_array_t **`。

SCHA_POSTNET_STOP

输出参数的类型为 `char **`。

SCHA_PRENET_START

输出参数的类型为 `char **`。

SCHA_PROXY

输出参数的类型为 `boolean_t *`。

SCHA_RESOURCE_LIST

输出参数的类型为 `scha_str_array_t **`。

SCHA_RT_BASEDIR

输出参数的类型为 `char **`。

SCHA_RT_DESCRIPTION

输出参数的类型为 `char **`。

SCHA_RT_SYSTEM

输出参数的类型为 `boolean_t *`。

SCHA_RT_VERSION

输出参数的类型为 `char **`。

SCHA_SINGLE_INSTANCE

输出参数的类型为 `boolean_t *`。

SCHA_START

输出参数的类型为 `char **`。

SCHA_STOP

输出参数的类型为 `char **`。

SCHA_UPDATE

输出参数的类型为 `char **`。

SCHA_VALIDATE

输出参数的类型为 `char **`。

这些函数返回以下值：

0 该函数成功。

非零值 该函数失败。

SCHA_ERR_NOERR 该函数成功。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#) 手册页。

`/usr/cluster/include/scha.h` 头文件

`/usr/cluster/lib/libscha.so` 库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scha_resource_get\(1HA\) \[585\]](#)、[scha_calls\(3HA\) \[897\]](#)、[scha_strerror\(3HA\) \[1057\]](#)、[scha_strerror_i1 link to "attributes5"](#)、[rt_properties\(5\) \[1143\]](#)

名称

scha_resourcetype_open, scha_resourcetype_open_zone, scha_resourcetype_get, scha_resourcetype_get_zone, scha_resourcetype_close — 资源类型信息访问函数

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib -l scha#include
<scha.h>scha_err_t scha_resourcetype_open( const char *rtname, scha_resourcetype_t *handle);

scha_err_t scha_resourcetype_open_zone(const char *cluster, const char
*rt_name, scha_resourcetype_t *handlep);

scha_err_t scha_resourcetype_close(scha_resourcetype_t handle);

scha_err_t scha_resourcetype_get(scha_resourcetype_t handle, const char *tag...);

scha_err_t scha_resourcetype_get_zone(const char *cluster,
    scha_resourcetype_t handlep, const char *rt_tag, ...);
```

使用 `scha_resourcetype_open()`、`scha_resourcetype_get()` 和 `scha_resourcetype_close()` 函数可以访问资源组管理器 (Resource Group Manager, RGM) 群集工具使用的资源类型的相关信息。

`scha_resourcetype_open()` 初始化资源类型访问，并返回一个将由 `scha_resourcetype_get()` 使用的句柄。

`scha_resourcetype_open()` 的 `rtname` 参数指定要访问的资源类型。

`handle` 参数是用于保存函数调用返回值的某个变量的地址。

`scha_resourcetype_get()` 根据 `tag` 参数的指示访问资源类型信息。`tag` 参数应该是 `scha_tags.h` 头文件中的某个宏定义的一个字符串值。标记后面的参数取决于 `tag` 的值。

可能需要在标记后面额外添加一个参数，用于指示要从中检索信息的群集节点或者指示特定于标记的其他信息。参数列表中最后一个参数的类型应该适合用于保存 `tag` 指定的信息。该参数是资源类型信息的“输出”参数。如果该函数失败，则不会返回该“输出”参数的值。对用于 `scha_resourcetype_get()` 的句柄调用 `scha_resourcetype_close()` 之前，所分配的用于保存 `scha_resourcetype_get()` 返回的信息的内存将保持不变。

`scha_resourcetype_close()` 接受上次调用 `scha_resourcetype_open()` 时返回的 `handle` 参数。该函数将使该句柄失效，并释放为了保存使用该句柄执行的 `scha_resourcetype_get()` 调用的返回值而分配的内存。请注意，如果需要返回值，则会为每个 `get` 调用分配内存。在一个调用中为返回值而分配的空间不会被后续调用覆盖和重新使用。

下文中列出了 `scha_tags.h` 中定义的可以用作 `scha_resourcetype_get()` 的 `tag` 参数的宏。指示输出参数和任何其他参数的类型。[scha_calls\(3HA\) \[897\]](#) 中介绍了结构和 `enum` 类型。

`scha_resourcetype_open_zone ()` 和 `scha_resourcetype_get_zone ()` 函数的作用分别与 `scha_resourcetype_open ()` 和 `scha_resourcetype_get ()` 相同，且带有一个额外的 `cluster` 参数，用于指定资源组存在于其中且您要在其上执行操作的区域群集的名称。当代码在全局区域中执行，但需要在指定区域群集上操作时，这些函数十分有用。无法在某一区域群集中使用它们来访问其他区域群集。

如果 `scha_resourcetype_open_zone ()` 或 `scha_resourcetype_get_zone ()` 的 `cluster` 参数为 `NULL`，则会对从中执行调用的群集执行查询；换句话说，带有 `NULL` 参数的调用分别等效于 `scha_resourcetype_open ()` 或 `scha_resourcetype_get ()`。

要关闭由 `scha_resourcetype_open_zone ()` 返回的句柄，请使用 `scha_resourcetype_close ()`。不需要 `cluster` 参数。

optag 参数

下列宏用于指定资源类型属性名称。输出是资源类型的指定属性的值。

注 - `optag` 参数（例如 `SCHA_API_VERSION` 和 `SCHA_BOOT`）不区分大小写。指定 `optag` 参数时，可以使用大小写字母的任意组合。

`SCHA_API_VERSION`

输出参数的类型为 `int*`。

`SCHA_BOOT`

输出参数的类型为 `char **`。

`SCHA_FAILOVER`

输出参数的类型为 `boolean_t *`。

`SCHA_FINI`

输出参数的类型为 `char **`。

`SCHA_GLOBALZONE`

输出参数的类型为 `boolean_t *`。

`SCHA_INIT`

输出参数的类型为 `char **`。

`SCHA_INIT_NODES`

输出参数的类型为 `scha_initnodes_flag_t *`。

`SCHA_INSTALLED_NODES`

输出参数的类型为 `scha_str_array_t **`。

SCHA_IS_LOGICAL_HOSTNAME

输出参数的类型为 `boolean_t *`。

SCHA_IS_SHARED_ADDRESS

输出参数的类型为 `boolean_t *`。

SCHA_MONITOR_CHECK

输出参数的类型为 `char **`。

SCHA_MONITOR_START

输出参数的类型为 `char **`。

SCHA_MONITOR_STOP

输出参数的类型为 `char **`。

SCHA_PER_NODE

输出参数的类型为 `boolean_t *`。

SCHA_PKGLIST

输出参数的类型为 `scha_str_array_t **`。

SCHA_POSTNET_STOP

输出参数的类型为 `char **`。

SCHA_PRENET_START

输出参数的类型为 `char **`。

SCHA_PROXY

输出参数的类型为 `boolean_t *`。

SCHA_RESOURCE_LIST

输出参数的类型为 `scha_str_array_t **`。

SCHA_RT_BASEDIR

输出参数的类型为 `char **`。

SCHA_RT_DESCRIPTION

输出参数的类型为 `char **`。

SCHA_RT_SYSTEM

输出参数的类型为 `boolean_t *`。

SCHA_RT_VERSION

输出参数的类型为 `char **`。

SCHA_SINGLE_INSTANCE

输出参数的类型为 `boolean_t *`。

SCHA_START

输出参数的类型为 `char **`。

SCHA_STOP

输出参数的类型为 `char **`。

SCHA_UPDATE

输出参数的类型为 `char **`。

SCHA_VALIDATE

输出参数的类型为 `char **`。

这些函数返回以下值：

0 该函数成功。

非零值 该函数失败。

SCHA_ERR_NOERR 该函数成功。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#) 手册页。

`/usr/cluster/include/scha.h` 头文件

`/usr/cluster/lib/libscha.so` 库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scha_resource_get\(1HA\) \[585\]](#)、[scha_calls\(3HA\) \[897\]](#)、[scha_strerror\(3HA\) \[1057\]](#)、[scha_strerror_1 link to "attributes5"](#)、[rt_properties\(5\) \[1143\]](#)

名称

scha_resourcetype_open, scha_resourcetype_open_zone, scha_resourcetype_get, scha_resourcetype_get_zone, scha_resourcetype_close — 资源类型信息访问函数

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib -l scha#include
<scha.h>scha_err_t scha_resourcetype_open( const char *rtname, scha_resourcetype_t *handle);

scha_err_t scha_resourcetype_open_zone(const char *cluster, const char
*rt_name, scha_resourcetype_t *handlep);

scha_err_t scha_resourcetype_close(scha_resourcetype_t handle);

scha_err_t scha_resourcetype_get(scha_resourcetype_t handle, const char *tag...);

scha_err_t scha_resourcetype_get_zone(const char *cluster,
    scha_resourcetype_t handlep, const char *rt_tag, ...);
```

使用 `scha_resourcetype_open()`、`scha_resourcetype_get()` 和 `scha_resourcetype_close()` 函数可以访问资源组管理器 (Resource Group Manager, RGM) 群集工具使用的资源类型的相关信息。

`scha_resourcetype_open()` 初始化资源类型访问，并返回一个将由 `scha_resourcetype_get()` 使用的句柄。

`scha_resourcetype_open()` 的 `rtname` 参数指定要访问的资源类型。

`handle` 参数是用于保存函数调用返回值的某个变量的地址。

`scha_resourcetype_get()` 根据 `tag` 参数的指示访问资源类型信息。`tag` 参数应该是 `scha_tags.h` 头文件中的某个宏定义的一个字符串值。标记后面的参数取决于 `tag` 的值。

可能需要在标记后面额外添加一个参数，用于指示要从中检索信息的群集节点或者指示特定于标记的其他信息。参数列表中最后一个参数的类型应该适合用于保存 `tag` 指定的信息。该参数是资源类型信息的“输出”参数。如果该函数失败，则不会返回该“输出”参数的值。对用于 `scha_resourcetype_get()` 的句柄调用 `scha_resourcetype_close()` 之前，所分配的用于保存 `scha_resourcetype_get()` 返回的信息的内存将保持不变。

`scha_resourcetype_close()` 接受上次调用 `scha_resourcetype_open()` 时返回的 `handle` 参数。该函数将使该句柄失效，并释放为了保存使用该句柄执行的 `scha_resourcetype_get()` 调用的返回值而分配的内存。请注意，如果需要返回值，则会为每个 `get` 调用分配内存。在一个调用中为返回值而分配的空间不会被后续调用覆盖和重新使用。

下文中列出了 `scha_tags.h` 中定义的可以用作 `scha_resourcetype_get()` 的 `tag` 参数的宏。指示输出参数和任何其他参数的类型。[scha_calls\(3HA\) \[897\]](#) 中介绍了结构和 `enum` 类型。

`scha_resourcetype_open_zone ()` 和 `scha_resourcetype_get_zone ()` 函数的作用分别与 `scha_resourcetype_open ()` 和 `scha_resourcetype_get ()` 相同，且带有一个额外的 `cluster` 参数，用于指定资源组存在于其中且您要在其上执行操作的区域群集的名称。当代码在全局区域中执行，但需要在指定区域群集上操作时，这些函数十分有用。无法在某一区域群集中使用它们来访问其他区域群集。

如果 `scha_resourcetype_open_zone ()` 或 `scha_resourcetype_get_zone ()` 的 `cluster` 参数为 `NULL`，则会对从中执行调用的群集执行查询；换句话说，带有 `NULL` 参数的调用分别等效于 `scha_resourcetype_open ()` 或 `scha_resourcetype_get ()`。

要关闭由 `scha_resourcetype_open_zone ()` 返回的句柄，请使用 `scha_resourcetype_close ()`。不需要 `cluster` 参数。

optag 参数

下列宏用于指定资源类型属性名称。输出是资源类型的指定属性的值。

注 - `optag` 参数（例如 `SCHA_API_VERSION` 和 `SCHA_BOOT`）不区分大小写。指定 `optag` 参数时，可以使用大小写字母的任意组合。

`SCHA_API_VERSION`

输出参数的类型为 `int*`。

`SCHA_BOOT`

输出参数的类型为 `char **`。

`SCHA_FAILOVER`

输出参数的类型为 `boolean_t *`。

`SCHA_FINI`

输出参数的类型为 `char **`。

`SCHA_GLOBALZONE`

输出参数的类型为 `boolean_t *`。

`SCHA_INIT`

输出参数的类型为 `char **`。

`SCHA_INIT_NODES`

输出参数的类型为 `scha_initnodes_flag_t *`。

`SCHA_INSTALLED_NODES`

输出参数的类型为 `scha_str_array_t **`。

SCHA_IS_LOGICAL_HOSTNAME

输出参数的类型为 `boolean_t *`。

SCHA_IS_SHARED_ADDRESS

输出参数的类型为 `boolean_t *`。

SCHA_MONITOR_CHECK

输出参数的类型为 `char **`。

SCHA_MONITOR_START

输出参数的类型为 `char **`。

SCHA_MONITOR_STOP

输出参数的类型为 `char **`。

SCHA_PER_NODE

输出参数的类型为 `boolean_t *`。

SCHA_PKGLIST

输出参数的类型为 `scha_str_array_t **`。

SCHA_POSTNET_STOP

输出参数的类型为 `char **`。

SCHA_PRENET_START

输出参数的类型为 `char **`。

SCHA_PROXY

输出参数的类型为 `boolean_t *`。

SCHA_RESOURCE_LIST

输出参数的类型为 `scha_str_array_t **`。

SCHA_RT_BASEDIR

输出参数的类型为 `char **`。

SCHA_RT_DESCRIPTION

输出参数的类型为 `char **`。

SCHA_RT_SYSTEM

输出参数的类型为 `boolean_t *`。

SCHA_RT_VERSION

输出参数的类型为 `char **`。

SCHA_SINGLE_INSTANCE

输出参数的类型为 `boolean_t *`。

SCHA_START

输出参数的类型为 `char **`。

SCHA_STOP

输出参数的类型为 `char **`。

SCHA_UPDATE

输出参数的类型为 `char **`。

SCHA_VALIDATE

输出参数的类型为 `char **`。

这些函数返回以下值：

0 该函数成功。

非零值 该函数失败。

SCHA_ERR_NOERR 该函数成功。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#) 手册页。

`/usr/cluster/include/scha.h` 头文件

`/usr/cluster/lib/libscha.so` 库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scha_resource_get\(1HA\) \[585\]](#)、[scha_calls\(3HA\) \[897\]](#)、[scha_strerror\(3HA\) \[1057\]](#)、[scha_strerror_1 link to "attributes5"](#)、[rt_properties\(5\) \[1143\]](#)

名称

scha_resourcetype_open, scha_resourcetype_open_zone, scha_resourcetype_get, scha_resourcetype_get_zone, scha_resourcetype_close — 资源类型信息访问函数

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib -l scha#include
<scha.h>scha_err_t scha_resourcetype_open( const char *rtname, scha_resourcetype_t *handle);

scha_err_t scha_resourcetype_open_zone(const char *cluster, const char
*rt_name, scha_resourcetype_t *handlep);

scha_err_t scha_resourcetype_close(scha_resourcetype_t handle);

scha_err_t scha_resourcetype_get(scha_resourcetype_t handle, const char *tag...);

scha_err_t scha_resourcetype_get_zone(const char *cluster,
    scha_resourcetype_t handlep, const char *rt_tag, ...);
```

使用 `scha_resourcetype_open()`、`scha_resourcetype_get()` 和 `scha_resourcetype_close()` 函数可以访问资源组管理器 (Resource Group Manager, RGM) 群集工具使用的资源类型的相关信息。

`scha_resourcetype_open()` 初始化资源类型访问，并返回一个将由 `scha_resourcetype_get()` 使用的句柄。

`scha_resourcetype_open()` 的 `rtname` 参数指定要访问的资源类型。

`handle` 参数是用于保存函数调用返回值的某个变量的地址。

`scha_resourcetype_get()` 根据 `tag` 参数的指示访问资源类型信息。`tag` 参数应该是 `scha_tags.h` 头文件中的某个宏定义的一个字符串值。标记后面的参数取决于 `tag` 的值。

可能需要在标记后面额外添加一个参数，用于指示要从中检索信息的群集节点或者指示特定于标记的其他信息。参数列表中最后一个参数的类型应该适合用于保存 `tag` 指定的信息。该参数是资源类型信息的“输出”参数。如果该函数失败，则不会返回该“输出”参数的值。对用于 `scha_resourcetype_get()` 的句柄调用 `scha_resourcetype_close()` 之前，所分配的用于保存 `scha_resourcetype_get()` 返回的信息的内存将保持不变。

`scha_resourcetype_close()` 接受上次调用 `scha_resourcetype_open()` 时返回的 `handle` 参数。该函数将使该句柄失效，并释放为了保存使用该句柄执行的 `scha_resourcetype_get()` 调用的返回值而分配的内存。请注意，如果需要返回值，则会为每个 `get` 调用分配内存。在一个调用中为返回值而分配的空间不会被后续调用覆盖和重新使用。

下文中列出了 `scha_tags.h` 中定义的可以用作 `scha_resourcetype_get()` 的 `tag` 参数的宏。指示输出参数和任何其他参数的类型。[scha_calls\(3HA\) \[897\]](#) 中介绍了结构和 `enum` 类型。

`scha_resourcetype_open_zone ()` 和 `scha_resourcetype_get_zone ()` 函数的作用分别与 `scha_resourcetype_open ()` 和 `scha_resourcetype_get ()` 相同，且带有一个额外的 `cluster` 参数，用于指定资源组存在于其中且您要在其上执行操作的区域群集的名称。当代码在全局区域中执行，但需要在指定区域群集上操作时，这些函数十分有用。无法在某一区域群集中使用它们来访问其他区域群集。

如果 `scha_resourcetype_open_zone ()` 或 `scha_resourcetype_get_zone ()` 的 `cluster` 参数为 `NULL`，则会对从中执行调用的群集执行查询；换句话说，带有 `NULL` 参数的调用分别等效于 `scha_resourcetype_open ()` 或 `scha_resourcetype_get ()`。

要关闭由 `scha_resourcetype_open_zone ()` 返回的句柄，请使用 `scha_resourcetype_close ()`。不需要 `cluster` 参数。

optag 参数

下列宏用于指定资源类型属性名称。输出是资源类型的指定属性的值。

注 - `optag` 参数（例如 `SCHA_API_VERSION` 和 `SCHA_BOOT`）不区分大小写。指定 `optag` 参数时，可以使用大小写字母的任意组合。

`SCHA_API_VERSION`

输出参数的类型为 `int*`。

`SCHA_BOOT`

输出参数的类型为 `char **`。

`SCHA_FAILOVER`

输出参数的类型为 `boolean_t *`。

`SCHA_FINI`

输出参数的类型为 `char **`。

`SCHA_GLOBALZONE`

输出参数的类型为 `boolean_t *`。

`SCHA_INIT`

输出参数的类型为 `char **`。

`SCHA_INIT_NODES`

输出参数的类型为 `scha_initnodes_flag_t *`。

`SCHA_INSTALLED_NODES`

输出参数的类型为 `scha_str_array_t **`。

SCHA_IS_LOGICAL_HOSTNAME

输出参数的类型为 `boolean_t *`。

SCHA_IS_SHARED_ADDRESS

输出参数的类型为 `boolean_t *`。

SCHA_MONITOR_CHECK

输出参数的类型为 `char **`。

SCHA_MONITOR_START

输出参数的类型为 `char **`。

SCHA_MONITOR_STOP

输出参数的类型为 `char **`。

SCHA_PER_NODE

输出参数的类型为 `boolean_t *`。

SCHA_PKGLIST

输出参数的类型为 `scha_str_array_t **`。

SCHA_POSTNET_STOP

输出参数的类型为 `char **`。

SCHA_PRENET_START

输出参数的类型为 `char **`。

SCHA_PROXY

输出参数的类型为 `boolean_t *`。

SCHA_RESOURCE_LIST

输出参数的类型为 `scha_str_array_t **`。

SCHA_RT_BASEDIR

输出参数的类型为 `char **`。

SCHA_RT_DESCRIPTION

输出参数的类型为 `char **`。

SCHA_RT_SYSTEM

输出参数的类型为 `boolean_t *`。

SCHA_RT_VERSION

输出参数的类型为 `char **`。

SCHA_SINGLE_INSTANCE

输出参数的类型为 `boolean_t *`。

SCHA_START

输出参数的类型为 `char **`。

SCHA_STOP

输出参数的类型为 `char **`。

SCHA_UPDATE

输出参数的类型为 `char **`。

SCHA_VALIDATE

输出参数的类型为 `char **`。

这些函数返回以下值：

0 该函数成功。

非零值 该函数失败。

SCHA_ERR_NOERR 该函数成功。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#) 手册页。

`/usr/cluster/include/scha.h` 头文件

`/usr/cluster/lib/libscha.so` 库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scha_resource_get\(1HA\) \[585\]](#)、[scha_calls\(3HA\) \[897\]](#)、[scha_strerror\(3HA\) \[1057\]](#)、[scha_strerror_i1 link to "attributes5"](#)、[rt_properties\(5\) \[1143\]](#)

名称

scha_resourcetype_open, scha_resourcetype_open_zone, scha_resourcetype_get, scha_resourcetype_get_zone, scha_resourcetype_close — 资源类型信息访问函数

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib -l scha#include
<scha.h>scha_err_t scha_resourcetype_open( const char *rtname, scha_resourcetype_t *handle);

scha_err_t scha_resourcetype_open_zone(const char *cluster, const char
*rt_name, scha_resourcetype_t *handlep);

scha_err_t scha_resourcetype_close(scha_resourcetype_t handle);

scha_err_t scha_resourcetype_get(scha_resourcetype_t handle, const char *tag...);

scha_err_t scha_resourcetype_get_zone(const char *cluster,
    scha_resourcetype_t handlep, const char *rt_tag, ...);
```

使用 `scha_resourcetype_open()`、`scha_resourcetype_get()` 和 `scha_resourcetype_close()` 函数可以访问资源组管理器 (Resource Group Manager, RGM) 群集工具使用的资源类型的相关信息。

`scha_resourcetype_open()` 初始化资源类型访问，并返回一个将由 `scha_resourcetype_get()` 使用的句柄。

`scha_resourcetype_open()` 的 `rtname` 参数指定要访问的资源类型。

`handle` 参数是用于保存函数调用返回值的某个变量的地址。

`scha_resourcetype_get()` 根据 `tag` 参数的指示访问资源类型信息。`tag` 参数应该是 `scha_tags.h` 头文件中的某个宏定义的一个字符串值。标记后面的参数取决于 `tag` 的值。

可能需要在标记后面额外添加一个参数，用于指示要从中检索信息的群集节点或者指示特定于标记的其他信息。参数列表中最后一个参数的类型应该适合用于保存 `tag` 指定的信息。该参数是资源类型信息的“输出”参数。如果该函数失败，则不会返回该“输出”参数的值。对用于 `scha_resourcetype_get()` 的句柄调用 `scha_resourcetype_close()` 之前，所分配的用于保存 `scha_resourcetype_get()` 返回的信息的内存将保持不变。

`scha_resourcetype_close()` 接受上次调用 `scha_resourcetype_open()` 时返回的 `handle` 参数。该函数将使该句柄失效，并释放为了保存使用该句柄执行的 `scha_resourcetype_get()` 调用的返回值而分配的内存。请注意，如果需要返回值，则会为每个 `get` 调用分配内存。在一个调用中为返回值而分配的空间不会被后续调用覆盖和重新使用。

下文中列出了 `scha_tags.h` 中定义的可以用作 `scha_resourcetype_get()` 的 `tag` 参数的宏。指示输出参数和任何其他参数的类型。[scha_calls\(3HA\) \[897\]](#) 中介绍了结构和 `enum` 类型。

`scha_resourcetype_open_zone ()` 和 `scha_resourcetype_get_zone ()` 函数的作用分别与 `scha_resourcetype_open ()` 和 `scha_resourcetype_get ()` 相同，且带有一个额外的 `cluster` 参数，用于指定资源组存在于其中且您要在其上执行操作的区域群集的名称。当代码在全局区域中执行，但需要在指定区域群集上操作时，这些函数十分有用。无法在某一区域群集中使用它们来访问其他区域群集。

如果 `scha_resourcetype_open_zone ()` 或 `scha_resourcetype_get_zone ()` 的 `cluster` 参数为 `NULL`，则会对从中执行调用的群集执行查询；换句话说，带有 `NULL` 参数的调用分别等效于 `scha_resourcetype_open ()` 或 `scha_resourcetype_get ()`。

要关闭由 `scha_resourcetype_open_zone ()` 返回的句柄，请使用 `scha_resourcetype_close ()`。不需要 `cluster` 参数。

optag 参数

下列宏用于指定资源类型属性名称。输出是资源类型的指定属性的值。

注 - `optag` 参数（例如 `SCHA_API_VERSION` 和 `SCHA_BOOT`）不区分大小写。指定 `optag` 参数时，可以使用大小写字母的任意组合。

`SCHA_API_VERSION`

输出参数的类型为 `int*`。

`SCHA_BOOT`

输出参数的类型为 `char **`。

`SCHA_FAILOVER`

输出参数的类型为 `boolean_t *`。

`SCHA_FINI`

输出参数的类型为 `char **`。

`SCHA_GLOBALZONE`

输出参数的类型为 `boolean_t *`。

`SCHA_INIT`

输出参数的类型为 `char **`。

`SCHA_INIT_NODES`

输出参数的类型为 `scha_initnodes_flag_t *`。

`SCHA_INSTALLED_NODES`

输出参数的类型为 `scha_str_array_t **`。

SCHA_IS_LOGICAL_HOSTNAME

输出参数的类型为 `boolean_t *`。

SCHA_IS_SHARED_ADDRESS

输出参数的类型为 `boolean_t *`。

SCHA_MONITOR_CHECK

输出参数的类型为 `char **`。

SCHA_MONITOR_START

输出参数的类型为 `char **`。

SCHA_MONITOR_STOP

输出参数的类型为 `char **`。

SCHA_PER_NODE

输出参数的类型为 `boolean_t *`。

SCHA_PKGLIST

输出参数的类型为 `scha_str_array_t **`。

SCHA_POSTNET_STOP

输出参数的类型为 `char **`。

SCHA_PRENET_START

输出参数的类型为 `char **`。

SCHA_PROXY

输出参数的类型为 `boolean_t *`。

SCHA_RESOURCE_LIST

输出参数的类型为 `scha_str_array_t **`。

SCHA_RT_BASEDIR

输出参数的类型为 `char **`。

SCHA_RT_DESCRIPTION

输出参数的类型为 `char **`。

SCHA_RT_SYSTEM

输出参数的类型为 `boolean_t *`。

SCHA_RT_VERSION

输出参数的类型为 `char **`。

SCHA_SINGLE_INSTANCE

输出参数的类型为 `boolean_t *`。

SCHA_START

输出参数的类型为 `char **`。

SCHA_STOP

输出参数的类型为 `char **`。

SCHA_UPDATE

输出参数的类型为 `char **`。

SCHA_VALIDATE

输出参数的类型为 `char **`。

这些函数返回以下值：

0 该函数成功。

非零值 该函数失败。

SCHA_ERR_NOERR 该函数成功。

有关其他错误代码的描述，请参见 [scha_calls\(3HA\) \[897\]](#) 手册页。

`/usr/cluster/include/scha.h` 头文件

`/usr/cluster/lib/libscha.so` 库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scha_resource_get\(1HA\) \[585\]](#)、[scha_calls\(3HA\) \[897\]](#)、[scha_strerror\(3HA\) \[1057\]](#)、[scha_strerror_1 link to "attributes5"](#)、[rt_properties\(5\) \[1143\]](#)

名称

scha_strerror, scha_strerror_i18n — 基于错误代码生成错误消息

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib -l scha#include <scha.h>char
*scha_strerror(scha_err_t error_code);
```

```
char *scha_strerror_i18n(scha_err_t error_code);
```

scha_strerror () 和 scha_strerror_i18n () 函数可以基于给定的 scha_err_t 错误代码生成一个用于描述错误的简短字符串。由 scha_strerror () 返回的字符串以英文显示。由 scha_strerror_i18n () 返回的字符串将以 LC_MESSAGES 语言环境类别指定的本机语言显示。请参见 [Unresolved link to "setlocale3C"](#)。

支持以下参数：

error_code 生成用于描述错误的简短字符串时所依据的错误代码。

例 390 使用 scha_strerror_i18n () 函数

```
sample()
{
    scha_err_t err;

    /* resource group containing example_R */
    char * resource_group = "example_RG";

    /* a configured resource */
    char * resource_name = "example_R";

    err = scha_control(SCHA_GIVEOVER, resource_group, resource_name);

    if (err != SCHA_ERR_NOERR) {
        syslog(LOG_ERR, "scha_control GIVEOVER failed: %s",
            scha_strerror_i18n(err));
    }
}
```

/usr/cluster/include/scha.h 头文件

/usr/cluster/lib/libscha.so 库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scha_calls\(3HA\) \[897\]](#)、[Unresolved link to "setlocale3C"](#)、[Unresolved link to "syslog3C"](#)、[Unresolved link to "attributes5"](#)

名称

scha_strerror, scha_strerror_i18n — 基于错误代码生成错误消息

```
cc [flags...] -I /usr/cluster/include file -L /usr/cluster/lib -l scha#include <scha.h>char
*scha_strerror(scha_err_t error_code);
```

```
char *scha_strerror_i18n(scha_err_t error_code);
```

scha_strerror () 和 scha_strerror_i18n () 函数可以基于给定的 scha_err_t 错误代码生成一个用于描述错误的简短字符串。由 scha_strerror () 返回的字符串以英文显示。由 scha_strerror_i18n () 返回的字符串将以 LC_MESSAGES 语言环境类别指定的本机语言显示。请参见 [Unresolved link to "setlocale3C"](#)。

支持以下参数：

error_code 生成用于描述错误的简短字符串时所依据的错误代码。

例 391 使用 scha_strerror_i18n () 函数

```
sample()
{
    scha_err_t err;

    /* resource group containing example_R */
    char * resource_group = "example_RG";

    /* a configured resource */
    char * resource_name = "example_R";

    err = scha_control(SCHA_GIVEOVER, resource_group, resource_name);

    if (err != SCHA_ERR_NOERR) {
        syslog(LOG_ERR, "scha_control GIVEOVER failed: %s",
            scha_strerror_i18n(err));
    }
}
```

/usr/cluster/include/scha.h 头文件

/usr/cluster/lib/libscha.so 库

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[scha_calls\(3HA\) \[897\]](#)、[Unresolved link to "setlocale3C"](#)、[Unresolved link to "syslog3C"](#)、[Unresolved link to "attributes5"](#)

OSC4 4

名称

clusters — 群集名称数据库

/etc/clusters

clusters 文件包含有关本地命名域中已知群集的信息。对于每个群集，单行应包含以下信息：

```
clustername whitespace-delimited list of hosts
```

如果右侧的名称已使用以下扩展标记进行标记，则扩展是递归过程：`*`。

项目由任意数量的空格和/或 TAB 字符隔开。# 表示注释开头。用于搜索文件的例程并未对整行字符进行解释。

群集名称可以包含除大写字母、字段分界符、换行符或注释字符以外的任何可打印字符。群集名称的最大长度为 32 个字符。

Oracle Solaris Cluster 系统管理工具（如 pconsole 命令）使用此信息指定一组要管理的节点。该数据库中使用的名称必须是主机数据库中使用的主机名。

可从 NIS 或 NIS+ 映射或本地文件使用该数据库。可以在 /etc/nsswitch.conf 文件中指定查找顺序。默认顺序为 nis 文件。

例 392 示例 /etc/clusters 文件

以下是典型的 /etc/clusters 文件：

```
bothclusters *planets *wine
planets mercury venus
wine zinfandel merlot chardonnay riesling
```

以下是典型的 /etc/nsswitch.conf 条目：

```
clusters: nis files
```

/etc/clusters

/etc/nsswitch.conf

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

[serialports\(4\) \[1077\]](#), [Unresolved link to "nsswitch.conf4"](#), [Unresolved link to "attributes5"](#)

名称

commandlog — 命令日志文件

`/var/cluster/logs/commandlog`

commandlog ASCII 文本文件包含在群集中执行的选定 Oracle Solaris Cluster 命令的记录。一旦设置群集，系统会自动启动对命令的日志记录，并在您关闭群集时结束。

不在该日志文件中记录的命令包括那些显示群集配置和当前状态的命令。在该文件中记录的命令包括配置和更改群集当前状态的那些命令，如下所示：

- claccess
- cldevice
- cldevicegroup
- clinterconnect
- clnasdevice
- clnode
- clquorum
- clreslogicalhostname
- clresource
- clresourcegroup
- clresourcetype
- clressharedaddress
- clsnmphost
- clsnmpmib
- clsnmpuser
- cltelemetryattribute
- cluster
- clzonecluster
- scconf
- scdidadm
- scdpm
- scgdevs
- scrgadm
- scshutdown
- scswitch

commandlog 文件中的每个记录都包含以下信息：

- 日期和时间戳
- 从其执行命令的主机名
- 命令的进程 ID
- 执行命令的用户的 ID
- 用户已执行的命令，包括所有选项和操作数

注 - 在 `commandlog` 文件中，命令选项两边已用引号引起来，允许您在 `shell` 中复制、粘贴和执行它们。

- 已执行命令的退出状态或信号

默认情况下，`commandlog` 文件会在每周末定期归档。在任何给定的时刻，Oracle Solaris Cluster 最多可以在每个群集节点上维护八个之前归档的 `commandlog` 文件。

例 393 `/var/cluster/logs/commandlog` 文件

以下示例显示了典型 `/var/cluster/logs/commandlog` 文件的内容：

```
11/11/2011 09:43:36 phys-schost-1 5758 root START - clrg add "app-sa-1"
11/11/2011 09:43:36 phys-schost-1 5758 root END 0
11/11/2011 09:43:36 phys-schost-1 5760 root START - clrg set -y
"RG_description=Department Shared Address RG" "app-sa-1"
11/11/2011 09:43:37 phys-schost-1 5760 root END 0
11/11/2011 09:44:15 phys-schost-1 5810 root START - clrg online "app-sa-1"
11/11/2011 09:44:15 phys-schost-1 5810 root END 0
11/11/2011 09:44:19 phys-schost-1 5222 root END -20988320
12/02/2011 14:37:21 phys-schost-1 5542 jbloggs START - clrg -c -g "app-sa-1"
-y "RG_description=Joe Bloggs Shared Address RG"
12/02/2011 14:37:22 phys-schost-1 5542 jbloggs END 0
```

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core
接口稳定性	Evolving (发展中)

[scha_control\(1HA\) \[579\]](#)、[scha_resource_setstatus\(1HA\) \[593\]](#)、[sconf\(1M\) \[637\]](#)、[scdidadm\(1M\) \[677\]](#)
[link to "attributes5"](#)

名称

rt_reg — 资源类型注册 (resource type registration, RTR) 文件

资源类型注册 (resource type registration, RTR) 文件介绍了资源类型。资源类型表示在资源组管理器 (Resource Group Manager, RGM) 群集设备控制下运行的高可用性或可伸缩服务。该文件是资源类型实现的一部分，用作输入文件以使 `scrgadm(1M)` [731] 命令在群集配置中注册资源类型。注册资源类型是创建要在群集中运行的此类资源的先决条件。按照约定，RTR 文件驻留在 `/opt/cluster/lib/rgm/rtreg` 目录下。

RTR 文件声明资源类型的资源类型属性和资源属性。该文件分为两部分，一部分是资源类型属性的声明，另一部分是资源属性的声明。请注意，属性名称的识别不区分大小写。

资源类型属性的声明提供有关资源类型实现的信息，例如，RGM 为控制此类资源所调用的回调方法的路径。大多数资源类型属性在 `rt_reg` 文件中都具有固定的值集。这些属性由该类型的所有资源继承。

资源类型实现器还可以定制和扩展资源属性的管理视图。可在 `rt_reg` 文件的第二部分中存在条目的资源属性有两种：系统定义的属性和扩展属性。

系统定义的资源属性具有预定的类型和语义。`rt_reg` 文件可用于设置特性，例如，系统定义的资源属性的默认值、最小值和最大值。`rt_reg` 文件还可以用于声明由资源类型实现完全定义的扩展属性。扩展属性为资源类型提供了一种用于向群集系统所维护和管理的资源的配置数据中添加信息的方式。

`rt_reg` 文件可设置资源属性的默认值，但实际值是在各个资源中设置的。`rt_reg` 文件中的属性可以是可由群集管理员设置为不同值并能进行调整的变量。

资源类型属性声明

资源类型属性声明包括多个属性值分配。

```
PROPERTY_NAME = "Value";
```

有关可在 `rt_reg` 文件中声明的资源类型属性的列表，请参见 `rt_properties(5)` [1143] 手册页。由于大多数属性均具有默认值或为可选属性，因此 RTR 文件中所必需的声明仅为类型名称、START 和 STOP 回调方法的路径和 `RT_version`。

请注意，该文件中的第一个属性必须为 `Resource_type` 属性。

资源类型名称的格式为 `vendor-id.RT-name:version`。

资源类型名称的三个组成部分是在 RTR 文件中指定为 `vendor-id`、`resource-type` 和 `RT-version` 的属性。`scrgadm` 命令用于插入句点和冒号分界符。虽然为可选项，但建议使用 `vendor-id` 前缀区分由不同供应商提供的同一名称的两个注册文件。要确保 `vendor-id` 唯一，请使用要创建资源类型的公司的股票代码。

资源属性声明

资源属性声明包括多个条目，每个条目均为特性值分配情况的括号列表。条目中的第一个特性必须为资源属性名称。

系统定义的属性具有预定的类型和描述特性，因此，不能在 `rt_reg` 文件中重新声明这些特性。可以为系统定义的属性声明范围限制、默认值以及对管理员何时可设置值的约束。

可为系统定义的属性设置的特性列在 [property_attributes\(5\) \[1087\]](#) 手册页中。在该表中还说明了不适用于系统定义的属性的特性。

可在 `rt_reg` 文件中具有条目的系统定义的属性列在 [r_properties\(5\) \[1103\]](#) 手册页中。以下是系统定义的 `RETRY_COUNT` 资源属性的样例条目。

```
{
  PROPERTY = RETRY_COUNT;
  MIN=0;
  MAX=10;
  DEFAULT=2;
  TUNABLE = ANYTIME;
}
```

扩展属性的条目必须指出属性的类型。可为扩展属性设置的特性列在 [property_attributes\(5\) \[1087\]](#) 手册页中。

以下是名为 "ConfigDir " (这是字符串类型) 的扩展属性的样例条目。TUNABLE 特性表明群集管理员可在创建资源时设置属性的值。

```
{
  PROPERTY = ConfigDir;
  EXTENSION;
  STRING;
  DEFAULT="/";
  TUNABLE = AT_CREATION;
}
```

用法

`rt_reg` 文件是一个 ASCII 文本文件。它可包含用于说明文件内容的注释。这些内容是上面介绍的两个部分，即资源类型属性列表，后跟资源属性声明。

空格可以是空白、制表符、换行符或注释。标记前后可以存在空格。在带引号的值标记中发现空白和井号 (#) 时，不会将其视为空格。空格用于分隔标记，否则可以忽略。

注释以 # 开始，以遇到的第一个换行符结束，包括第一个换行符。

指令以 # \$ 开始，以遇到的第一个换行符结束，包括第一个换行符。指令必须出现在 RTR 文件中的资源类型属性声明部分和资源属性声明部分之间。在 RTR 文件的任何其他位置插入的指令都将产生解析器错误。有效的指令只有 `#$upgrade` 和 `#$upgrade_from`。任何其他指令都将产生解析器错误。

标记为属性名称、属性值以及以下项：

{ } 括住参数表属性
; 终止属性和特性
= 分隔属性名称和属性值或特性名称和特性值
, 分隔值列表中的值

文件中属性名称的关键字的识别不区分大小写。

属性和特性具有以下三种格式之一。

```
property-name = property-value ;  
property-name;  
property-name = property-value [ , property-value ];
```

在上面的格式中，方括号 [] 括的是可选项。即，属性值可以是一个 *property-value*，也可以是由逗号分隔的两个或多个 *property-value* 的列表。

属性列表中的第一个属性必须是简单资源类型名称。

布尔值属性和特性具有以下语法：

```
boolean-property-name;  
boolean-property-name = TRUE;  
boolean-property-name = FALSE;
```

第一种和第二种格式均将 *boolean-property-name* 设置为 TRUE。

资源类型属性名称列在 [rt_properties\(5\) \[1143\]](#) 手册页中。系统定义的属性列在 [r_properties\(5\) \[1103\]](#) 手册页中。

资源声明包括任意数量的条目，每个条目均为资源属性特性的括号列表。

```
{attribute-value-list}
```

每个 *attribute-value-list* 包括资源属性的特性值，除了两种 *type-attribute* 格式外，这些特性值均采用供属性值使用的同一语法。

```
type-attribute-value;  
enum-type-attribute { enum-value [ , enum-value ] };
```

type-attribute-value 语法将扩展属性的数据类型声明为具有 *type-attribute-value* 值。它不同于 *boolean-property-name* 的第一种格式，后者将名为 *boolean-property-name* 的属性定义为具有 TRUE 值。

例如，TUNABLE 属性可具有以下值之一：FALSE 或 NONE、AT_CREATION、TRUE 或 ANYTIME 以及 WHEN_DISABLED。当 TUNABLE 属性采用语法：

```
TUNABLE;
```

时，它将获取 ANYTIME 值。

语法

下面介绍了 `rt_reg` 文件的语法以及类似 BNF 的语法。尽管对 `rt_reg` 文件中关键字的实际识别不区分大小写，但是非终端关键字是小写的，终端关键字是大写的。在行的开头位于非终端后面的冒号 (:) 表示语法产生式。语法产生式的备用右侧在行中表示为以竖直线 (|) 开头。变量终端标记以尖括号表示，注释括在圆括号内。语法产生式右侧中的其他标点 (例如分号 (;)、等号 (=) 和花括号 ({})) 均为文本。

注释的格式为：

```
COMMENT : # anything but NEWLINE NEWLINE
```

注释可出现在任意标记的后面。注释被视为空格。

```
    rt_reg_file : Resource_type = value ; proplist upgradesect paramtable

    proplist : (NONE: empty)
| proplist rtproperty

    rtproperty : rtboolean_prop ;
| rtvalue_prop ;

    rtboolean_prop : SINGLE_INSTANCE
| FAILOVER | RT_SYSTEM

    rtvalue_prop : rtprop = value
| PKGLIST = valuelist

    rtprop : RT_BASEDIR
| RT_VERSION
| API_VERSION
| INIT_NODES
| START
| STOP
| VALIDATE
| UPDATE
| INIT
| FINI
| BOOT
| MONITOR_START
| MONITOR_STOP
| MONITOR_CHECK
| PRENET_START
| POSTNET_STOP
| RT_DESCRIPTION
| VENDOR_ID
| rtboolean_prop (booleans may have explicit assignments.)

    value : contiguous-non-ws-non-;-characters
| "anything but quote"
| TRUE
| FALSE
```

```

| ANYTIME
| WHEN_DISABLED
| AT_CREATION
| RG_PRIMARYES
| RT_INSTALLED_NODES
| (NONE: Empty value)

    valuelist : value
| valuelist , value

    upgradesect : (empty)
| #UPGRADE upgradelist

    upgradelist : (empty)
| upgradelist #UPGRADE_FROM rt_version upgtunability

    upgtunability : ANYTIME
| AT_CREATION
| WHEN_DISABLED
| WHEN_OFFLINE
| WHEN_UNMANAGED
| WHEN_UNMONITORED

    paramtable : (empty)
| paramtable parameter

    parameter : { pproplist }

    pproplist : PROPERTY = value ; (property name must come first)
| pproplist pproperty

    pproperty : pboolean_prop ;
| pvalue_prop ;
| typespec ;

    pvalue_prop : tunable_prop
| pprop = value
| pprop = (NONE: no value setting)
| DEFAULT = valuelist

    pprop : DESCRIPTION
| MIN
| MAX
| MINLENGTH
| MAXLENGTH
| ARRAY_MINSIZE
| ARRAY_MAXSIZE
| pboolean_prop

    tunable_prop : TUNABLE
| TUNABLE = AT_CREATION
| TUNABLE = ANYTIME
| TUNABLE = WHEN_DISABLED
| TUNABLE = TRUE

```

```

| TUNABLE = FALSE
| TUNABLE = NONE

typespec : INT
| BOOLEAN
| STRING
| STRINGARRAY
| ENUM { valuelist }

```

例 394 注册文件样例

以下是简单示例资源类型的注册文件。

```

#
# Registration information for example resource type
#

Resource_type = example_RT;
Vendor_id = SUNW;
RT_Version = 2.0
RT_Basedir= /opt/SUNWxxx;
START = bin/example_service_start;
STOP = bin/example_service_stop;

#$upgrade
#$upgrade_from "1.0" when_unmonitored

#
# Set range and defaults for method timeouts and Retry_count.
#
{ Property = START_TIMEOUT; Tunable; MIN=60; DEFAULT=300; }
{ Property = STOP_TIMEOUT; Tunable; MIN=60; DEFAULT=300; }
{ Property = Retry_count; Tunable; MIN=1; MAX=20; DEFAULT=10; }

#
# An extension property that can be set at resource creation
#
{ Property = LogLevel;
  Extension;
  enum { OFF, TERSE, VERBOSE };
  Default = TERSE;
  Tunable = AT_CREATION;
  Description = "Controls the detail of example_service logging";
}

```

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api

属性类型	属性值
接口稳定性	Evolving (发展中)

[scrgadm\(1M\) \[731\]](#)、[Unresolved link to "attributes5"](#)、[rt_properties\(5\) \[1143\]](#)、[r_properties\(5\) \[1103\]](#)、[property_attributes\(5\) \[1087\]](#)
[Unresolved link to "Oracle Solaris Cluster Data Services Developer's Guide"](#)

名称

scdpmd.conf — 磁盘路径监视守护进程配置文件

```
/etc/cluster/scdpm/scdpmd.conf
```

scdpmd 守护进程监视磁盘路径并根据路径故障采取适当的操作。您可以通过创建或修改包含可调属性的配置文件 `/etc/cluster/scdpm/scdpmd.conf` 来调节此守护进程，并将 SIGHUP 信号发送到 scdpmd 守护进程以读取配置文件。

```
# pkill -HUP scdpmd
```

您可以在 `scdpmd.conf` 文件中调节以下属性：

Ping_interval

描述

检查磁盘路径状态的时间间隔（秒）

默认值

600

最小值

20

最大值

3600

Ping_retry

描述

出现故障时查询磁盘路径状态的重试次数

默认值

3

最小值

2

最大值

10

Ping_timeout

描述

查询任何磁盘路径状态的超时时间（秒）

默认值
30
最小值
1
最大值
1800

下面是有效 `scdpmd.conf` 文件的示例：

```
Ping_interval = 120  
Ping_retry = 5  
Ping_timeout = 10
```

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
接口稳定性	Evolving (发展中)

[cldevice\(1CL\) \[51\]](#)、[clnode\(1CL\) \[149\]](#)

名称

serialports — 串行端口数据库的名称

```
/etc/serialportsserialports NIS or NIS+ maps
```

serialports 数据库将名称映射到服务器名称与 TCP 端口号（表示与指定的终端服务器主机相连的串行端口）。该数据库通常用于将主机名映射到其控制台，但也可用于提供对打印机、调制解调器之类设备的访问。当服务正由基于网络的终端集中器提供时将会使用映射。对于每个名称，应该有一个单独的行，其中包含以下信息：

```
host-name concentrator-hostname tcp-port-number
```

各项之间由任意数目的空格或 TAB 字符分隔。井号 (#) 表示注释的开始。搜索文件的例程不会解释井号与行末尾之间的字符。

可以从命令行使用并行控制台访问实用程序 (pconsole) 远程登录到群集中。pconsole 实用程序是 Oracle Solaris terminal/pconsole 软件包的组件。可通过执行 `pkg installterminal/pconsole` 安装该软件包。pconsole 实用程序可为您在命令行中指定的每个远程主机创建一个主机终端窗口。该实用程序还将打开一个中央（主）控制台窗口，该窗口可将您在终端窗口中输入的信息传播到打开的每个连接。

```
/etc/serialports
```

```
/etc/nsswitch.conf
```

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/developer/api
接口稳定性	Evolving (发展中)

[clusters\(4\) \[1063\]](#)、[Unresolved link to "nsswitch.conf4"](#)、[Unresolved link to "attributes5"](#)

OSC4 5

名称

SUNW.crs_framework, crs_framework — 协调 Oracle Clusterware 的关闭的资源类型实现

SUNW.crs_framework 资源类型可在 Oracle Solaris Cluster Support for Oracle Real Application Clusters (Oracle RAC) 配置中协调 Oracle Clusterware 和 Oracle Solaris Cluster 资源的关闭。该资源类型通过使 Oracle Solaris Cluster 停止 Oracle Clusterware 来使 Oracle Solaris Cluster 和 Oracle Clusterware 互操作。

注 - 该资源类型不允许使用 Oracle Solaris Cluster 管理命令启动 Oracle Clusterware。只能通过使用 Oracle 命令或引导节点来启动 Oracle Clusterware。

Oracle Clusterware 表决磁盘和 Oracle 群集注册表 (Oracle cluster registry, OCR) 文件可能位于 SUNW.ScalDeviceGroup 或 SUNW.ScalMountPoint 类型的资源表示的存储中。在这种情况下，在使 SUNW.ScalDeviceGroup 或 SUNW.ScalMountPoint 类型的资源脱机之前必须先停止 Oracle Clusterware。SUNW.crs_framework 类型的资源可通过在以下情况下在某个节点上停止 Oracle Clusterware 进程确保满足此要求。

- 当 SUNW.ScalDeviceGroup 或 SUNW.ScalMountPoint 类型的资源在节点上脱机时。出于以下原因，必须停止 Oracle Clusterware 进程：
 - 为了确保正确停止 SUNW.ScalDeviceGroup 或 SUNW.ScalMountPoint 类型的资源
 - 为了避免因在 Oracle Clusterware 或 Oracle RAC 进程访问存储期间使 SUNW.ScalDeviceGroup 或 SUNW.ScalMountPoint 类型的资源脱机而发生数据库或节点故障
- 当某个节点关闭时。如果未停止 Oracle Clusterware 进程，该节点将无法关闭。

SUNW.crs_framework 资源类型是单实例资源类型。在群集中只能创建一个此类型的资源。

为确保 Oracle Solaris Cluster 按正确顺序停止资源，请按如下所述配置一个类型为 SUNW.crs_framework 的资源：

- 确保包含 SUNW.ScalDeviceGroup 或 SUNW.ScalMountPoint 类型的资源的任何资源组为将包含 SUNW.crs_framework 资源的资源组声明强正关联性。
- 设置 SUNW.crs_framework 资源对表示 Oracle Clusterware 表决磁盘和 OCR 文件存储的任何资源的脱机重新启动依赖性。这些资源的类型为 SUNW.ScalDeviceGroup 或 SUNW.ScalMountPoint。将每个依赖性的范围限制为正在运行 SUNW.ScalDeviceGroup 资源或 SUNW.ScalMountPoint 资源的节点。
- 设置 SUNW.crs_framework 类型的资源对 SUNW.rac_framework 类型的资源的强依赖性。

应在为 Oracle Solaris Cluster Support for Oracle RAC 数据服务配置数据库资源时创建这些依赖性和关联性。有关更多信息，请参见[Unresolved link to "适用于 Oracle](#)

Real Application Clusters 的 Oracle Solaris Cluster 数据服务指南 中的配置 [Support for Oracle RAC 数据库实例的资源](#)"。

要注册此资源类型并创建该资源类型的实例，请使用以下方法之一：

- Oracle Solaris Cluster Manager
- 使用 [clsetup\(1CL\) \[407\]](#) 实用程序，并指定用于配置 Oracle Solaris Cluster Support for Oracle Real Application Clusters 的选项
- 以下 Oracle Solaris Cluster 维护命令序列：
 1. 要注册此资源类型，请使用 [clresourcetype\(1CL\) \[277\]](#) 命令。
 2. 要创建此资源类型的实例，请使用 [clresource\(1CL\) \[225\]](#) 命令。

标准属性

有关所有标准资源属性的描述，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

按如下所示覆盖此资源类型的标准资源属性：

Monitor_start_timeout

最小值	60
默认值	300

Monitor_stop_timeout

最小值	60
默认值	300

Start_timeout

最小值	60
默认值	300

Stop_timeout

最小值	60
默认值	1200

Update_timeout

最小值	60
默认值	300

Validate_timeout

最小值 60

默认值 300

扩展属性

SUNW.crs_framework 资源类型没有扩展属性。

例 395 创建 SUNW.crs_framework 资源

此示例注册 SUNW.crs_framework 资源类型并创建 SUNW.crs_framework 资源类型的名为 crs_framework-rs 的实例。本示例做出了以下假定：

- 使用 C shell。
- 存在一个名为 crs-framework-rg 的资源组。
- 存在以下资源：
 - 类型为 SUNW.rac_framework 的名为 rac_framework-rs 的资源，表示 Oracle RAC 框架
 - 类型为 SUNW.ScalDeviceGroup 的名为 db-storage-rs 的资源，表示存储 Oracle Clusterware 表决磁盘和 OCR 文件的可伸缩设备组

```
phys-schost-1# clresourcetype register SUNW.crs_framework
```

```
phys-schost-1# clresource create -g crs-framework-rg \  
-t SUNW.crs_framework \  
-p resource_dependencies=rac_framework-rs \  
-p resource_dependencies_offline_restart=db-storage-rs\{local_node\} \  
crs_framework-rs
```

有关以下属性的描述，请参见 [Unresolved link to " attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/library/ucmm

[clresource\(1CL\) \[225\]](#)、[clresourcetype\(1CL\) \[277\]](#)、[clsetup\(1CL\) \[407\]](#)、[Unresolved link to " attributes5"](#)

[SUNW.rac_framework\(5\) \[1213\]](#)、[SUNWct.ScalDeviceGroup\(5\) \[1215\]](#)、[SUNW.ScalMountPoint\(5\) \[1216\]](#)

[Unresolved link to " 适用于 Oracle Real Application Clusters 的 Oracle Solaris Cluster 数据服务指南 "](#)

名称

SUNW.derby, derby — Java DB 数据库的资源类型实现

SUNW.derby 是故障转移资源类型，允许您结合使用 Java DB 数据库与 Oracle Solaris Cluster。Java DB 数据库以 Derby 数据库为基础。有关数据库的信息，请参见 <http://db.apache.org/derby/>。

与 SUNW.derby 资源类型关联的扩展属性如下所示：

DB_path (字符串)

为 Java DB 数据库指定数据文件的位置。

DB_path 的值是指定路径的字符串。指定的路径应该是由所选存储（例如 HAStoragePlus）控制的路径。

类别	必需
可调	禁用时

DB_port (整数)

指定 Java DB 数据库的端口。

类别	必需
默认值	1527
可调	禁用时

DB_probe_port (整数)

指定 Oracle Solaris Cluster 在测试 Java DB 数据库服务器的运行状况时使用的端口。

类别	必需
默认值	1528
可调	禁用时

Monitor_retry_count (整数)

控制故障监视器的重新启动。该属性指定进程监视器工具重新启动故障监视器的次数。该属性对应于传递给 `pmfadm(1M)` [625] 命令的 `-n` 选项。资源组管理器 (Resource Group Manager, RGM) 在指定时间段（请参见 `Monitor_retry_interval`）中对重新启动次数进行计数。请注

意，Monitor_retry_count 表示故障监视器本身的重新启动，不是 SUNW.derby 资源的重新启动。

类别	可选
默认值	4
可调	随时

Monitor_retry_interval (整数)

以分钟为单位指定时间段，在此期间 RGM 对故障监视器的失败进行计数。该属性对应于传递给 [pmfadm\(1M\) \[625\]](#) 命令的 -t 选项。如果故障监视器的失败次数超出了扩展属性 Monitor_retry_count 的值，则进程监视器工具不会重新启动故障监视器。

类别	可选
默认值	2 分钟
可调	随时

Probe_timeout (整数)

指定探测命令的超时值（以秒为单位）。

类别	可选
默认值	120 秒
默认值	2 秒
可调	随时

[pmfadm\(1M\) \[625\]](#)

名称

property_attributes — 资源属性特性

以下列表说明了可用于更改系统定义属性或创建扩展属性的资源属性特性。

您无法将 NULL 或空字符串 ("") 指定为 Boolean、Enum 或 Int 类型的默认值。

Array_maxsize

对于 Stringarray 类型，表示允许使用的最大数目的数组元素。

Array_minsize

对于 Stringarray 类型，表示允许使用的最小数目的数组元素。

Default

表示属性的默认值。

Description

字符串注释，用于对属性进行简单的说明。说明特性无法在适用于系统定义属性的 RTR 文件中进行设置。

Enumlist

对于 Enum 类型，表示允许用于该属性的一组字符串值。

Extension

如果使用了该属性，则表示 RTR 文件项声明了由资源类型实现定义的扩展属性。否则，该文件项为系统定义的属性。

Max

对于 Int 类型，表示允许用于该属性的最大值。请注意，您无法为方法超时设置最大值。

Maxlength

对于 String 和 Stringarray 类型，表示允许使用的最大字符串长度。

Min

对于 Int 类型，表示允许用于该属性的最小值。请注意，您无法为方法超时指定 Min=0。

Minlength

对于 String 和 Stringarray 类型，表示允许使用的最小字符串长度。

Per_node

如果使用了该属性，则表示扩展属性可以基于每节点进行设置。

如果在类型定义中指定 Per_node 属性特性，则还必须指定默认值和 Default 属性特性。指定默认值可确保用户在对尚未分配显式值的节点请求每节点属性值时，系统会返回一个值。

Property

资源属性的名称。

属性类型

允许的类型有：String、Boolean、Int、Enum 和 Stringarray。您无法在适用于系统定义属性的 RTR 文件项中设置类型特性。类型用于确定可接受的属性值以及 RTR 文件项中允许的特定于类型的特性。Enum 类型是一组字符串值。

Tunable

表示群集管理员何时可在资源中设置该属性的值。可设置为 None 或 False 以阻止管理员设置此属性。允许管理员调节的值有：True 或 Anytime（随时）、At_creation（仅在创建资源时）或 When_disabled（资源处于脱机状态时）。

默认值为 True (Anytime)。

例 396 Int 类型定义

Int 类型定义可能看起来像这样：

```
{
    Property = Probe_timeout;
    Extension;
    Int;
    Default = 30;
    Tunable = Anytime;
    Description = "Time out value for the probe (seconds)";
}
```

例 397 Per_node 类型定义

Per_node 类型定义可能看起来像这样：

```
{
    Property = LogLevel;
    Extension;
    Enum { Off, Terse, Verbose };
    Default = Terse;
    Per_node;
    Tunable = At_creation;
    Description = "Controls the level of detail for logging";
}
```

}

如果在类型定义中指定 `PER_NODE` 属性特性，则还必须指定默认值和 `DEFAULT` 属性特性。

[clresource\(1CL\) \[225\]](#)、[clresourcegroup\(1CL\) \[253\]](#)、[clresourcetype\(1CL\) \[277\]](#)、[r_properties\(5\) \[1103\]](#)
[rg_properties\(5\) \[1129\]](#)、[rt_properties\(5\) \[1143\]](#)

名称

SUNW.Proxy_SMF_failover, Proxy_SMF_failover — 代理故障转移 SMF 服务的资源类型

SUNW.Proxy_SMF_failover 资源类型表示服务管理设备 (Service Management Facility, SMF) 服务的故障转移的代理。

下面的子部分中介绍了为 SUNW.proxysmffailover 资源类型定义的标准属性和扩展属性。要为 SUNW.Proxy_SMF_failover 资源类型的实例设置这些属性，请使用 `clresource` 命令 ([clresource\(1CL\) \[225\]](#))。

标准属性

有关以下资源属性的描述，请参见 [r_properties\(5\) \[1103\]](#)。

Start_timeout

默认值：3600

最小值：60

Stop_timeout

默认值：3600

最小值：60

Init_timeout

默认值：3600

最小值：60

Boot_timeout

默认值：3600

最小值：60

Fini_timeout

默认值：3600

最小值：60

Validate_timeout

默认值：3600

最小值：60

Failover_mode

默认值：SOFT

可调：Anytime

R_description

默认值：""

可调：Anytime

Retry_count

默认值：2

最小值：0

最大值：10

可调：Anytime

Retry_interval

默认值：300

最大值：3600

可调：Anytime

Through_probe_interval

默认值：60

可调：Anytime

扩展属性

Proxied_service_instances

包含关于将由资源代理的 SMF 服务的信息。其值是指向包含所有代理的 SMF 服务的文件的路径。该文件中的每一行专用于一个 SMF 服务，并指定 `svc_fmri` 和指向相应服务清单文件的路径。例如，如果资源需要管理两个服务 `restarter_svc_test_1:default` 和 `restarter_svc_test_2:default`，该文件应该包含以下两行：

```
<svc:/system/cluster/restarter_svc_test_1:default>,  
  svc:/system/cluster/restarter_svc_test_1:default>,  
  </var/svc/manifest/system/cluster/restarter_svc_test_1.xml>  
  
<svc:/system/cluster/restarter_svc_test_2:default>,  
  </var/svc/manifest/system/cluster/restarter_svc_test_2.xml>
```

注 - 以上条目应该位于单独一行中。在此处分成多行是为了便于阅读。

默认值：""

可调：When disabled (禁用时)

示例

此示例说明如何注册 `SUNW.Proxy_SMF_failover` 资源类型、为应用程序创建资源组、创建故障转移应用程序资源、管理资源组，以及如何启用资源。

注册资源类型：

```
# clresourcetype -f <path-to-rtrfile> SUNW.Proxy_SMF_failover
```

为应用程序创建一个名为 `rg1` 的资源组：

```
# clresourcegroup create rg1
```

创建名为 `myfailoverres` 的故障转移应用程序资源：

```
# clresource create -t SUNW.Proxy_SMF_failover -g rg1 \  
-x proxied_service_instances="/usr/local/app/svc myfailoverres"
```

其中 `/usr/local/app/svc` 是文本文件。

管理资源组 `rg1`：

```
# clresourcegroup manage rg1
```

启用 `myfailoverres` 资源：

```
# clresource enable myfailoverres
```

使用以下命令检查应用程序状态：

```
# clresource status
```

[pmfadm\(1M\) \[625\]](#)、[scha_resource_get\(1HA\) \[585\]](#)、[clresourcetype\(1CL\) \[277\]](#)、[clresource\(1CL\) \[225\]](#)、[c](#)
[link to " attributes5"](#)、[r_properties\(5\) \[1103\]](#)

[Unresolved link to " Oracle Solaris Cluster 数据服务规划和管理指南 "](#)

名称

SUNW.Proxy_SMF_multimaster, Proxy_SMF_multimaster — 代理多主节点 SMF 服务的资源类型

SUNW.Proxy_SMF_multimaster 资源类型表示多主节点服务管理设备 (Service Management Facility, SMF) 服务的代理。

下面的子部分中介绍了为 SUNW.proxysmfmultimaster 资源类型定义的标准属性和扩展属性。要为 SUNW.Proxy_SMF_multimaster 资源类型的实例设置这些属性，请使用 `clresource` 命令 ([clresource\(1CL\) \[225\]](#))。

标准属性

有关以下资源属性的描述，请参见 [r_properties\(5\) \[1103\]](#)。

Start_timeout

默认值：3600

最小值：60

Stop_timeout

默认值：3600

最小值：60

Init_timeout

默认值：3600

最小值：60

Boot_timeout

默认值：3600

最小值：60

Fini_timeout

默认值：3600

最小值：60

Validate_timeout

默认值：3600

最小值：60

Failover_mode

默认值：SOFT

可调：Anytime

R_description

默认值：""

可调：Anytime

Retry_count

默认值：2

最小值：0

最大值：3

可调：Anytime

Retry_interval

默认值：300

最大值：3600

可调：Anytime

Through_probe_interval

默认值：60

可调：Anytime

扩展属性

Proxied_service_instances

包含关于将由资源代理的 SMF 服务的信息。其值是指向包含所有代理的 SMF 服务的文件的路径。该文件中的每一行专用于一个 SMF 服务，并指定 `svc_fmri` 和指向相应服务清单文件的路径。例如，如果资源需要管理两个服务 `restarter_svc_test_1:default` 和 `restarter_svc_test_2:default`，该文件应该包含以下两行：

```
<svc:/system/cluster/restarter_svc_test_1:default>,  
  svc:/system/cluster/restarter_svc_test_1:default>,  
  </var/svc/manifest/system/cluster/restarter_svc_test_1.xml>  
  
<svc:/system/cluster/restarter_svc_test_2:default>,  
  </var/svc/manifest/system/cluster/restarter_svc_test_2.xml>
```

注 - 以上条目应该位于单独一行中。在此处分成多行是为了便于阅读。

默认值：""

可调：When disabled (禁用时)

名称

SUNW.Proxy_SMF_scalable, Proxy_SMF_scalable — 代理可伸缩 SMF 服务的资源类型

SUNW.Proxy_SMF_scalable 资源类型表示可伸缩服务管理设备 (Service Management Facility, SMF) 服务的代理。

下面的子部分中介绍了为 SUNW.proxysmfscalable 资源类型定义的标准属性和扩展属性。要设置 SUNW.Proxy_SMF_scalable 资源类型的实例的这些属性，请使用 `clresource` 命令。

标准属性

有关以下资源属性的描述，请参见 [r_properties\(5\) \[1103\]](#)。

Start_timeout

默认值：3600

最小值：60

Stop_timeout

默认值：3600

最小值：60

Init_timeout

默认值：3600

最小值：60

Boot_timeout

默认值：3600

最小值：60

Fini_timeout

默认值：3600

最小值：60

Validate_timeout

默认值：3600

最小值：60

Failover_mode

默认值：SOFT

可调：任何时候

R_description

默认值：""

可调：任何时候

Retry_count

默认值：2

最小值：0

最大值：3

可调：任何时候

Retry_interval

默认值：300

最大值：3600

可调：任何时候

Through_probe_interval

默认值：60

可调：任何时候

扩展属性

Proxied_service_instances

包含关于将由资源代理的 SMF 服务的信息。其值是指向包含所有代理的 SMF 服务的文件的路径。该文件中的每一行专用于一个 SMF 服务，并指定 `svc_fmri` 和指向相应服务清单文件的路径。例如，如果资源需要管理两个服务 `restarter_svc_test_1:default` 和 `restarter_svc_test_2:default`，该文件应该包含以下两行：

```
<svc:/system/cluster/restarter_svc_test_1:default>,  
  svc:/system/cluster/restarter_svc_test_1:default>,  
  </var/svc/manifest/system/cluster/restarter_svc_test_1.xml>  
  
<svc:/system/cluster/restarter_svc_test_2:default>,  
  </var/svc/manifest/system/cluster/restarter_svc_test_2.xml>
```

注 - 以上条目应该位于单独一行中。在此处分成多行是为了便于阅读。

默认值：""

可调：when disabled (禁用时)

名称

r_properties — 资源属性

以下信息描述了由 Oracle Solaris Cluster 软件定义的标准资源属性。标准属性在使用它们的所有资源类型中具有共同的含义。这些描述已针对数据服务开发者进行开发。有关特定数据服务的更多信息，请参见该数据服务的手册页。

除了标准属性之外，每个资源类型还可以定义其自己的特定于类型的资源属性，称为扩展属性。标准属性和扩展属性都是在资源类型注册 (Resource Type Registration, RTR) 文件中声明的。RTR 文件定义群集管理员向 Oracle Solaris Cluster 软件注册数据服务时该数据服务的初始配置。

可以指定群集管理员能够基于每个节点或者为整个群集设置扩展属性。但是，不能（在 RTR 文件中）指定群集管理员可以执行对标准属性执行的相同操作。标准属性可以隐式应用于所有节点或特定节点。标准属性是应用于所有节点还是仅应用于特定节点取决于每个标准属性的具体定义。

有关 RTR 文件的更多信息，请参见 [rt_reg\(4\) \[1067\]](#) 手册页。有关可以为资源属性设置的各个特性的信息，请参见 [property_attributes\(5\) \[1087\]](#) 手册页。

注 - 必须将资源的 Scalable 属性设置为 TRUE，才能使用 Oracle Solaris Cluster 软件的网络负载平衡功能。可伸缩资源可以使用 Affinity_timeout、Generic_affinity、Load_balancing_policy、Load_balancing_weights、Conn_threshold 和 Weak_affinity 属性。

有些资源类型可以运行在多个节点上，而不使用网络负载平衡。此类资源的 Scalable 属性设置为 False，而且此类资源不使用上述其他属性。

标准资源属性类别

必需	在使用管理实用程序创建资源时，群集管理员必须指定值。
可选	如果在创建资源组时群集管理员不指定值，则系统将提供默认值。
有条件的	仅当在资源类型注册 (Resource Type Registration, RTR) 文件中声明属性时，资源组管理器 (Resource Group Manager, RGM) 才创建属性。否则，属性将不存在且对群集管理员不可用。在 RTR 文件中声明的有条件属性是可选的或必需的，具体取决于是否在 RTR 文件中指定了默认值。有关详细信息，请参见每个有条件属性的描述。
仅查询	不能由管理工具直接设置。

群集管理员可以使用以下命令编辑所有的可调属性：

```
# clresource set -p property=  
new-value resource
```

标准资源属性描述

注 - 属性名 (如 `Affinity_timeout` 和 `Cheap_probe_interval`) 不区分大小写。指定属性名称时, 可以使用大小写字母的任意组合。

`Affinity_timeout` (integer)

时长 (秒) , 给定的客户机 IP 地址为获取资源中任何服务而建立的连接在该期间将发送到同一服务器节点。

如果将该属性设置为 -1, 则所有连接都将发送到同一节点。如果将该属性设置为 0, 则所有打开的连接都将发送到同一节点。如果将该属性设置为 *n*, 则在最后一个连接关闭后 *n* 秒, 所有新连接都将发送到与最后一个连接相同的节点。

在所有情况下, 如果服务器节点因故障而离开群集, 则选择新的服务器节点。

仅当 `Load_balancing_policy` 为 `Lb_sticky` 或 `Lb_sticky_wild` 时该属性才相关。此外, `Weak_affinity` 必须设置为 `False` (默认值)。

该属性仅用于可伸缩服务。

类别 有条件的/可选

默认值 0

可调 随时

`Application_user` (string)

执行与资源相关的应用程序执行时使用的 Oracle Solaris 用户名。

资源方法或监视器执行的应用程序可能以 `root` 或非 `root` 用户 (“应用程序用户”) 身份执行, 具体取决于特定代理的实现方式。并非所有资源类型都有 `application_user` 资源属性; 只有声明此属性的那些资源类型才能允许设置该属性。

声明了 `application_user` 资源属性的资源类型通常是一个代理, 该代理使用 [scha_check_app_user\(1HA\) \[563\]](#) 接口对应用程序可执行文件的所有权和权限执行额外的检查。如果应用程序可执行文件不归 `root` 所有, 但要由 `root` 执行, 或者该可执行文件具有组或全局写权限, 则会存在不安全性。在这种情况下, 如果 `resource_security` 属性设置为 `SECURE`, 则执行应用程序会在运行时失败, 并返回错误。如果 `resource_security` 采用任何其他设置, 将允许应用程序在伴有警告消息的情况下执行。

声明 `application_user` 属性的资源类型根据 `resource_security` 群集属性的设置来设置用于执行应用程序的用户 ID。如果 `resource_security` 设置为 `COMPATIBILITY`, 将会忽略 `application_user` 资源属性的设置, 应用程序用户将是调用者的有效用户 ID (通常为 `root`)。此行为与以前的 Oracle Solaris Cluster 发行版兼容。

如果 `resource_security` 设置为 `OVERRIDE`，将会忽略 `application_user` 属性，应用程序用户将是应用程序可执行文件的所有者。

如果 `resource_security` 设置为 `SECURE` 或 `WARN`，应用程序用户将是 `application_user` 资源属性的值；但是，如果未设置 `application_user` 或者该属性为空，则应用程序用户将是应用程序可执行文件的所有者。

如果在 RTR 文件中未指定 `Tunable` 特性，则该属性的可调性是 `When_disabled`。

类别	有条件的/可选
默认值	空字符串
可调	禁用时

`Cheap_probe_interval` (integer)

资源的快速故障探测的两次调用之间的秒数。仅当在 RTR 文件中声明了该属性，该属性才会由 RGM 创建且可供群集管理员使用。

如果在 RTR 文件中指定了默认值，则该属性是可选的。如果在资源类型文件中未指定 `Tunable` 特性，则该属性的 `Tunable` 值是 `When_disabled`。

类别	有条件的
默认值	请参见上文
可调	禁用时

`CheckActivePortInstances`(boolean)

在并非 `Port_list` 属性中指定的所有端口都有活动侦听进程时，决定节点是否参与可伸缩服务以及是否接收来自负载均衡器的客户机请求。该属性仅适用于可伸缩服务。

支持的值包括：

- `FALSE` (默认值) – 当其至少一个端口具有活动侦听进程时，节点参与可伸缩服务。
- `TRUE` – 仅当所有端口都具有活动侦听进程时，节点才参与可伸缩服务。

类别	有条件的/可选
默认值	<code>False</code>
可调	禁用时

`Conn_threshold` (integer)

启用 `Round_robin` 负载分配时支持的活动连接或客户机的最大数量。如果连接端点在服务器节点上保持活动状态，则 TCP 连接将被视为处于活动状态。如果 UDP 会话活动超时窗口设置中有通信流，则 UDP 会话被视为处于活动状态（请参见 `udp_session_timeout` 群集属性）。

类别	可选
默认值	10000
可调	禁用时

Failover_mode (enum)

修改资源无法成功启动或停止时或者资源监视器发现运行状态不佳的资源并进而请求重新启动或故障转移时 RGM 执行的恢复操作。

NONE、SOFT 或 HARD (方法失败)

这些设置仅影响启动或停止方法 (Prenet_start, Start, Monitor_stop, Stop, Postnet_stop) 失败时的故障转移行为。RESTART_ONLY 和 LOG_ONLY 设置还可能影响资源监视器是否可以启动 scha_control 命令或 scha_control () 函数的执行。

NONE 指示在前面提到的启动或停止方法之一失败时 RGM 不执行任何恢复操作。SOFT 或 HARD 指示当 Start 或 Prenet_start 方法失败时, RGM 将资源所属的组重新定位到不同的节点。对于 Start 或 Prenet_start 失败, SOFT 和 HARD 是相同的。

对于停止方法 (Monitor_stop、Stop 或 Postnet_stop) 的失败, SOFT 与 NONE 相同。如果 Failover_mode 设置为 HARD, 则当这些停止方法之一失败时, RGM 将重新引导节点以强制资源组脱机。然后, RGM 可能会尝试在另一个节点上启动该组。但是, 如果资源组被 clresourcegroup quiesce 子命令停顿, 则节点将不重新引导, 即使 Failover_mode 为 HARD 且停止方法失败也是如此。在这种情况下, 资源将转为 STOP_FAILED 状态。

RESTART_ONLY 或 LOG_ONLY

与 NONE、SOFT 和 HARD (它们影响启动或停止方法失败时的故障转移行为) 不同, RESTART_ONLY 和 LOG_ONLY 影响所有的故障转移行为。故障转移行为包括监视器启动的资源和资源组 (scha_control) 重新启动, 以及由资源监视器启动的移交。

RESTART_ONLY 指示监视器可以运行 scha_control 以重新启动资源或资源组。RGM 允许 Retry_count 在 Retry_interval 内重新启动。如果超过 Retry_count, 则不允许进一步的重新启动。

注 - 负的 Retry_count 值 (有些资源类型允许, 并非所有资源类型都允许) 指定无限次数的资源重新启动。指定无限制重新启动的更可靠方法是执行以下操作:

- 将 Retry_interval 设置为小值, 如 1 或 0。
- 将 Retry_count 设置为大值, 如 1000。

如果资源类型未声明 Retry_count 和 Retry_interval 属性, 则允许无限次数的资源重新启动。

如果 Failover_mode 设置为 LOG_ONLY，则不允许资源重新启动或移交。将 Failover_mode 设置为 LOG_ONLY 与在 Retry_count 设置为零时将 Failover_mode 设置为 RESTART_ONLY 相同。

RESTART_ONLY 或 LOG_ONLY (方法失败)

如果 Prenet_start、Start、Monitor_stop、Stop 或 Postnet_stop 方法失败，则 RESTART_ONLY 和 LOG_ONLY 与 NONE 相同。即，数据服务未故障转移或重新引导。

Failover_mode 设置对数据服务的影响

Failover_mode 的每个设置对数据服务具有的影响取决于数据服务是受监视还是未受监视以及它是否基于数据服务开发库 (Data Services Development Library, DSDL)。

- 如果数据服务实现了 Monitor_start 方法且启用了资源的监视，则该数据服务是受监视的。RGM 通过在启动资源本身之后执行 Monitor_start 方法来启动资源监视器。资源监视器可探测资源的运行状况。如果探测失败，则资源监视器可以通过调用 scha_control () 函数来请求重新启动或故障转移。对于基于 DSDL 的资源，探测可能会揭示数据服务的部分故障 (降级) 或完整故障。重复的部分故障累积成完整故障。
- 如果数据服务未提供 Monitor_start 方法或者已禁用对资源的监视，则该数据服务未受监视。
- 基于 DSDL 的数据服务包括使用代理生成器、通过 GDS 或者通过直接使用 DSDL 开发的数据服务。有些数据服务 (例如 HA Oracle) 是在未使用 DSDL 的情况下开发的。

NONE、SOFT 或 HARD (探测失败)

如果将 Failover_mode 设置为 NONE、SOFT 或 HARD，且数据服务是基于 DSDL 的受监视服务，则在探测完全失败时监视器将调用 scha_control () 函数以请求重新启动资源。如果探测继续失败，则在 Retry_interval 内重新启动资源，最大次数为 Retry_count。如果重新启动次数达到 Retry_count 后探测再次失败，则监视器请求将资源所属的组故障转移到其他节点。

如果将 Failover_mode 设置为 NONE、SOFT 或 HARD，且数据服务是基于 DSDL 的未受监视的服务，则检测到的唯一故障就是资源进程树的停止。如果资源的进程树停止，则重新启动该资源。

如果数据服务不是基于 DSDL 的服务，则重新启动或故障转移行为取决于资源监视器的编码方式。例如，Oracle 资源监视器通过重新启动资源或资源组，或者通过故障转移资源组来进行恢复。

RESTART_ONLY (探测失败)

如果将 Failover_mode 设置为 RESTART_ONLY，且数据服务是基于 DSDL 的受监视服务，则在探测完全失败时会在 Retry_interval 内重新启动资源 Retry_count 次。但是，如果超过 Retry_count，资源监视器将退出，将资源状态设置为 FAULTED，并生成状态消息 Application faulted, but not restarted. Probe

quitting.。此时，虽然监视仍处于启用状态，但是在群集管理员修复并重新启动资源之前，资源实际上是未受监视的。

如果将 Failover_mode 设置为 RESTART_ONLY，且数据服务是基于 DSDL 的未受监视的服务，则在进程树停止时不重新启动资源。

如果受监视的数据服务不基于 DSDL，则恢复行为取决于资源监视器的编码方式。如果将 Failover_mode 设置为 RESTART_ONLY，则可以通过在 Retry_interval 内调用 scha_control () 函数 Retry_count 次来重新启动资源或资源组。如果资源监视器超过 Retry_count，则重新启动尝试将失败。如果监视器调用 scha_control () 以请求故障转移，则该请求也将失败。

LOG_ONLY (探测失败)

如果对任何数据服务将 Failover_mode 设置为 LOG_ONLY，则重新启动资源或资源组或者对组进行故障转移的所有 scha_control () 请求都将被阻止。如果数据服务基于 DSDL，则在探测完全失败时记录一条消息，但是不重新启动资源。如果探测在 Retry_interval 内完全失败的次数超过 Retry_count，资源监视器将退出，将资源状态设置为 FAULTED，并生成状态消息 Application faulted, but not restarted. Probe quitting.。此时，虽然监视仍处于启用状态，但是在群集管理员修复并重新启动资源之前，资源实际上是未受监视的。

如果将 Failover_mode 设置为 LOG_ONLY，且数据服务是基于 DSDL 的未受监视的服务，则在进程树停止时记录一条消息，但不重新启动资源。

如果受监视的数据服务不基于 DSDL，则恢复行为取决于资源监视器的编码方式。如果将 Failover_mode 设置为 LOG_ONLY，则重新启动资源或资源组或者对组进行故障转移的所有 scha_control () 请求都将失败。

类别	可选
默认值	NONE
可调	随时

Global_zone_override (boolean)

只有在 RTR 文件中设置 Global_zone=TRUE 属性的资源类型才允许该属性。Global_zone_override 属性的设置将覆盖特定资源的资源类型属性 Global_zone 的值。有关更多信息，请参见 [rt_properties\(5\) \[1143\]](#) 手册页。

将 Global_zone_override 属性设置为 FALSE 将强制资源方法在配置了资源组的非全局区域中执行，而不是始终在全局区域中执行（当 Global_zone 属性设置为 TRUE 时通常会始终在全局区域中执行）。

如果在 RTR 文件中指定了默认值，则该属性是可选的。

如果在 RTR 文件中未指定 Tunable 特性，则该属性的 Tunable 值是 At_creation。可以在 RTR 文件中将 Tunable 属性设置为 At_creation、When_disabled 或 Anytime。

注 - 在 RTR 文件中将 Tunable 特性设置为 Anytime 时，请谨慎操作。对 Global_zone_override 属性的更改将立即生效，即使资源处于联机状态也是如此。例如，假定 Global_zone_override 可调性设置为 ANYTIME，Global_zone_override 属性当前在非全局区域中配置的资源上设置为 FALSE。当资源切换为联机时，将在非全局区域中执行启动方法。如果随后将 Global_zone_override 属性设置为 TRUE 且资源切换为脱机，则将在全局区域中执行停止方法。方法代码必须处理该可能性。如果方法代码不处理该可能性，则必须改为将 Tunable 属性设置为 When_disabled 或 At_creation。

类别	有条件的/可选
默认值	TRUE
可调	创建时

Load_balancing_policy (string)

一个字符串，用于定义使用中的负载平衡策略。该属性仅用于可伸缩服务。如果在 RTR 文件中声明了 Scalable 属性，则 RGM 自动创建该属性。

Load_balancing_policy 可以采用以下值：

- Lb_weighted (默认值)。根据在 Load_balancing_weights 属性中设置的权值，在各个节点之间分配负载。
- Lb_sticky.在配置应用程序资源时，端口集是已知的。可伸缩服务的给定客户机（由客户机的 IP 地址标识）始终发送到群集的另一节点。
- Lb_sticky_wild.预先不知道端口号，但是会动态分配。连接到通配符粘性服务的 IP 地址的给定客户机（由客户机的 IP 地址标识）始终发送到同一群集节点，而不管该 IP 地址来自哪个端口号。

类别	有条件的/可选
默认值	Lb_weighted
可调	创建时

Load_balancing_weights (string_array)

仅用于可伸缩资源。如果在 RTR 文件中声明了 Scalable 属性，则 RGM 自动创建该属性。格式为 *weight@node,weight@node...*，其中 *weight* 是一个整数，反映分配到指定 *node* 的相对负载部分。分配到一个节点的负载在总的负载中占的比例等于该节点的权值除以所有权值的总和。例如，1@1,3@2 指定节点 1 拥有负载的 1/4，节点 2 拥有负载的 3/4。空字符串 ("") (默认值) 设置均匀分配。未分配显式权值的任何节点将拥有默认权值 1。可以指定权值 0 以便不为节点分配负载。

如果在资源类型文件中未指定 Tunable 特性，则该属性的 Tunable 值是 Anytime。更改该属性仅修改新连接的分配。

类别	有条件的/可选
----	---------

默认值	Null
可调	随时

用于每种回调方法的 *method_timeout* (integer)
时间间隔 (秒) , 之后 RGM 断定方法调用已失败。

注 - 不能指定方法超时的最大值 (使用 Max 属性) 。同样, 也不能指定最小值零 (Min=0)。

类别	有条件的/可选
默认值	如果在 RTR 文件中声明了方法本身, 则为 3,600 秒 (1 小时)
可调	随时

Monitored_switch (enum)

不能直接设置该属性。相反, 它在特定的节点上或整个群集中由 RGM 设置为 Enabled 或 Disabled。如果群集管理员使用管理实用程序在特定节点上或者为整个群集启用或禁用监视器, 则 RGM 会这样做。如果已禁用, 则在再次启用监视之前不会在资源上调用 Monitor_start 方法。如果资源没有监视器回调方法, 则该属性的计算结果为 Disabled。

类别	仅查询
默认值	如果资源类型具有监视方法, 则为 Enabled, 否则为 disabled
可调	请参见描述

Network_resources_used (string_array)

该资源对其具有依赖性的逻辑主机名或共享地址资源的列表。该列表包含在属性 Resource_dependencies、Resource_dependencies_weak、Resource_dependencies_restart 或 Resource_dependencies_offline_restart 中出现的所有网络地址资源。

如果在 RTR 文件中声明了 Scalable 属性, 则 RGM 自动创建该属性。如果在 RTR 文件中未声明 Scalable 属性, 则 Network_resources_used 将不可用, 除非在 RTR 文件中显式声明它。

如果没有为 Network_resources_used 属性分配值, 则将基于资源依赖性属性的设置由 RGM 自动更新其值。无需直接设置该属性。而是设置 Resource_dependencies、Resource_dependencies_offline_restart、Resource_dependencies_restart 或 Resource_dependencies_weak 属性。如果指定了每节点依赖性, 则 Network_resources_used 属性的派生值只包含在本地节点上有效的那些依赖性。该值可能在每个节点上有所不同。

要保持与 Oracle Solaris Cluster 软件早期发行版的兼容性, 仍可以直接设置 Network_resources_used 属性的值。如果直接设置 Network_resources_used 属

性的值，则 `Network_resources_used` 属性的值将不再派生自资源依赖性属性的设置。如果向 `Network_resources_used` 属性添加资源名称，则该资源名称也会自动添加到 `Resource_dependencies` 属性。删除该依赖性的唯一方法是将其从 `Network_resources_used` 属性中删除。如果您不确定网络资源依赖性最初添加到 `Resource_dependencies` 属性还是 `Network_resources_used` 属性，请从这两个属性中删除依赖性。例如，以下命令将删除资源 `r1` 对网络资源 `r2` 的依赖性，而不管依赖性是否添加到 `Network_resources_used` 还是 `Resource_dependencies` 属性：

```
# clresource set -p Network_resources_used--r2 -p Resource_dependencies--r2 r1
```

为简单起见，请避免为 `Network_resources_used` 属性设置值。仅设置资源依赖性属性，并将 `Network_resources_used` 属性视为只读属性。

类别	有条件的/可选
默认值	空列表
可调	随时

每个群集节点上的 `Num_resource_restarts` (integer)

在过去的 n 秒内对该资源上发生的重新启动请求数，其中 n 是 `Retry_interval` 属性的值。

重新启动请求是以下任一调用：

- 带有 `RESOURCE_RESTART` 参数的 `scha_control` 命令
- 带有 `SCHA_RESOURCE_RESTART` 参数的 `scha_control ()` 函数
- 带有 `RESOURCE_IS_RESTARTED` 参数的 `scha_control` 命令
- 带有 `SCHA_RESOURCE_IS_RESTARTED` 参数的 `scha_control ()` 函数

只要给定节点上的给定资源执行以下操作之一，RGM 就会为该资源将重新启动计数器重置为零：

- 带有 `GIVEOVER` 参数的 `scha_control` 命令
- 带有 `SCHA_GIVEOVER` 参数的 `scha_control ()` 函数

不管移交尝试成功还是失败，都将重置计数器。

如果资源类型未声明 `Retry_interval` 属性，则 `Num_resource_restarts` 属性不可用于该类型的资源。

类别	仅查询
默认值	无默认值
可调	请参见描述

每个群集节点上的 `Num_rg_restarts` (integer)

在过去的 n 秒内对该资源发生的资源组重新启动请求数，其中 n 是 `Retry_interval` 属性的值。

资源组重新启动请求是以下任一调用：

- 带有 RESTART 参数的 `scha_control` 命令
- 带有 SCHA_RESTART 参数的 `scha_control ()` 函数

如果资源类型未声明 `Retry_interval` 属性，则 `Num_rg_restarts` 属性不可用于该类型的资源。

类别	仅查询
默认值	无默认值
可调	请参见描述

`On_off_switch` (enum)

不能直接设置该属性。相反，它在特定的节点上或整个群集中由 RGM 设置为 `Enabled` 或 `Disabled`。如果群集管理员使用管理实用程序在特定节点上或者为整个群集启用或禁用资源，则 RGM 会这样做。如果已禁用，则在再次启用资源之前它不调用回调。

类别	仅查询
默认值	已禁用
可调	请参见描述

`Outgoing_Connection`(boolean)

指定可伸缩服务是否在发起到群集之外的服务器的传出请求时使用虚拟网络地址（请参见 `Network_resources_used` 属性）。负载均衡器确保将所有传入应答转发至发起节点。

注 - 群集中每次只有一个节点可以发起到给定服务器的请求。

该属性仅适用于 `Generic_Affinity` 设置为 `TRUE` 且 `Load_balancing_policy` 设置为 `LB_STICKY_WILD` 的可伸缩服务。支持的值包括：

- `FALSE`（默认值） - 可伸缩服务在发起到外部服务器的传出请求时不使用 `Network_resources_used` 属性中指定的虚拟网络地址。
- `TRUE` - 可伸缩服务通过使用 `Network_resources_used` 属性中指定的虚拟网络地址发起到外部服务器的传出请求。负载均衡器将传入应答转发到发起节点。

类别	有条件的/可选
默认值	<code>False</code>
可调	创建时

Port_list (string_array)

服务器侦听的端口号的逗号分隔列表。附加到每个端口号的是一个斜杠 (/), 后跟该端口正在使用的协议, 例如 Port_list=80/tcp 或 Port_list=80/tcp6,40/udp6。

可以指定的可能协议包括：

- tcp, 仅适用于 TCP IPv4
- tcp6, 同时适用于 TCP IPv4 和 TCP IPv6
- udp, 仅适用于 UDP IPv4
- udp6, 同时适用于 UDP IPv4 和 UDP IPv6

如果在 RTR 文件中声明了 Scalable 属性, 则 RGM 自动创建 Port_list。否则, 该属性不可用, 除非在 RTR 文件中显式声明它。

[Unresolved link to "Oracle Solaris Cluster Data Service for Apache Guide"](#) 中介绍了有关设置该属性以用于 Oracle Solaris Cluster HA for Apache 的信息。

类别	有条件的/必需
默认值	无默认值
可调	随时

Pre_evict (boolean)

优先级较低的资源组的逐出可以与资源组切换一起执行, 前提是另一个组对要切换的组声明强负关联, 或者在节点上超过了硬负载限制。

该属性可确定 RGM 是否尝试在包含该资源的资源组启动切换之前执行资源组逐出。

支持的值包括：

- FALSE (默认值) - 切换的资源组在目标节点上开始联机时, 执行资源组逐出。
- TRUE - 切换的资源组从当前的主节点上开始脱机之前, 在切换的目标节点上执行资源组逐出。该设置仅对单一主节点的资源组 (即 Maximum primaries 属性设置为 1 的资源组) 有效。

如果在 RTR 文件中未指定 Tunable 特性, 则该属性的 Tunable 值是 When disabled。

类别	可选
默认值	False
可调	禁用时

R_description (string)

资源的简要描述。

类别	可选
默认值	空字符串
可调	随时

Resource_dependencies (string_array)

该资源对其具有强依赖性的资源（位于同一组中或不同组中）的列表。如果列表中任何资源的启动失败，则无法启动该资源。如果该资源和列表中的资源之一同时启动，则 RGM 将等待列表中的资源启动后再启动该资源。如果该资源的 Resource_dependencies 列表中的资源未启动（例如，如果列表中资源的资源组保持脱机状态，或者如果列表中的资源处于 Start_failed 状态），则该资源也保持脱机状态。如果该资源由于对无法启动的其他资源组中资源的依赖性而保持脱机状态，则该资源所属的组将进入 Pending_online_blocked 状态。

如果该资源与列表中的资源同时脱机，则该资源将在列表中的资源之前停止。但是，如果该资源保持联机状态或无法停止，则列表中的资源都会停止。

默认情况下，在资源组中，应用程序资源具有对网络地址资源的隐式强资源依赖性。rg_properties(5) [1129] 手册页中的 Implicit_network_dependencies 提供了更多信息。

在资源组中，Prenet_start 方法在 Start 方法之前按依赖性顺序运行。Postnet_stop 方法在 Stop 方法之后按依赖性顺序运行。在不同的资源组中，依赖资源等待被依赖资源完成 Prenet_start 和 Start 后再运行 Prenet_start。被依赖资源等待依赖资源完成 Stop 和 Postnet_stop 后再运行 Stop。

要指定依赖性的范围，请在指定该属性时将以下限定符（包括花括号 {} 或 at 符号 @）附加到资源名称。

{ANY_NODE}

将指定的依赖性扩展到任何节点。依赖资源的行为受任何节点上的被依赖资源影响。依赖资源等待被依赖资源在任何主节点上启动后再启动自身。停止和重新启动时的情况与此类似。

{FROM_RG_AFFINITIES}

指定资源依赖性的范围派生自资源所属的资源组的 RG_affinities 关系。如果依赖资源所属的组与被依赖资源所属的资源组正关联，且它们在同一节点上启动或停止，则依赖性为 {LOCAL_NODE}。如果不存在此类正关联，或者如果这两个组在不同的节点上启动，则依赖性为 {ANY_NODE}。

{LOCAL_NODE}

将指定的依赖性限制为基于每个节点。依赖资源的行为仅受同一节点上的被依赖资源影响。依赖资源等待被依赖资源在同一节点上启动。停止和重新启动时的情况与此类似。

如果 {LOCAL_NODE} 依赖资源在一个故障转移（即，单一主节点）资源组中，而且在一个节点上未满足 {LOCAL_NODE} 依赖性，则该资源组可能故障转移到

满足 {LOCAL_NODE} 依赖性的其他节点，而不是在未满足依赖性的节点上保持 Pending_online_blocked 状态。

@nodename

指定限制于特定节点且对其他节点无影响的 {LOCAL_NODE} 依赖性。这允许资源对群集中的每个节点具有不同的依赖性。*nodename* 是节点名称或节点 ID。

例如，以下列表指示资源 res1 对节点 node1 的依赖性和资源 res2 对节点 node2 的依赖性：

```
res1@node1, res2@node2
```

如果相同的依赖性适用于多个节点，请对每个节点名称重复资源名称。例如：

```
myres@node1, myres@node2, myres@node3, ...
```

位于同一资源组中的两个资源之间的资源依赖性始终为 {LOCAL_NODE}。

如果不指定限定符，则默认情况下使用 FROM_RG_AFFINITIES。

[scha_resource_get\(1HA\) \[585\]](#)、[scha_resource_get\(3HA\) \[963\]](#) 和 [scds_property_functions\(3HA\) \[871\]](#) 手册页介绍了使用或不使用限定符获取依赖性列表的替代查询表单。

类别	可选
默认值	空列表
可调	随时

Resource_dependencies_offline_restart (string_array)

该资源对其具有脱机重新启动依赖性的资源（位于同一组中或不同组中）的列表。

该属性的工作方式就像 Resource_dependencies 一样，不同之处是，如果停止了脱机重新启动依赖性列表中的任何资源，则该资源也会停止。如果随后重新启动了脱机重新启动依赖性列表中的资源，则该资源也会重新启动。

如果列表中任何资源的启动失败，则无法启动该资源。如果该资源和列表中的资源之一同时启动，则 RGM 将等待列表中的资源启动后再启动该资源。如果该资源的 Resource_dependencies 列表中的资源未启动（例如，如果列表中资源的资源组保持脱机状态，或者如果列表中的资源处于 Start_failed 状态），则该资源也保持脱机状态。如果该资源由于对无法启动的其他资源组中资源的依赖性而保持脱机状态，则该资源所属的组将进入 Pending_online_blocked 状态。

如果该资源与列表中的资源同时脱机，则该资源将在列表中的资源之前停止。但是，如果该资源保持联机状态或无法停止，则列表中的资源都会停止。

如果节点上的“被依赖”资源发生故障，并且该资源无法恢复，则 RGM 将使该资源在该节点上脱机。RGM 还通过在其上触发重新启动使被依赖资源的所有脱机重新启动依赖资源脱机。当群集管理员解决故障并重新启用被依赖资源时，RGM 还会使被依赖资源的 offline-restart 依赖资源重新联机。

要指定依赖性的范围，请在指定该属性时将以下限定符（包括花括号 {} 或 at 符号 (@)）附加到资源名称。

{ANY_NODE}

将指定的依赖性扩展到任何节点。依赖资源的行为受任何节点上的被依赖资源影响。依赖资源等待被依赖资源在任何主节点上启动后再启动自身。停止和重新启动时的情况与此类似。

{FROM_RG_AFFINITIES}

指定资源依赖性的范围派生自资源所属的资源组的 RG affinities 关系。如果依赖资源所属的组与被依赖资源所属的资源组正关联，且它们在同一节点上启动或停止，则依赖性为 {LOCAL_NODE}。如果不存在此类正关联，或者如果这两个组在不同的节点上启动，则依赖性为 {ANY_NODE}。

{LOCAL_NODE}

将指定的依赖性限制为基于每个节点。依赖资源的行为仅受同一节点上的被依赖资源影响。依赖资源等待被依赖资源在同一节点上启动。停止和重新启动时的情况与此类似。

如果 {LOCAL_NODE} 依赖资源在一个故障转移（即，单一主节点）资源组中，而且在一个节点上未满足 {LOCAL_NODE} 依赖性，则该资源组可能故障转移到满足 {LOCAL_NODE} 依赖性的其他节点，而不是在未满足依赖性的节点上保持 Pending_online_blocked 状态。

@nodename

指定限制于特定节点且对其他节点无影响的 {LOCAL_NODE} 依赖性。这允许资源对群集中的每个节点具有不同的依赖性。nodename 是节点名称或节点 ID。

例如，以下列表指示资源 res1 对节点 node1 的依赖性和资源 res2 对节点 node2 的依赖性：

```
res1@node1, res2@node2
```

如果相同的依赖性适用于多个节点，请对每个节点名称重复资源名称。例如：

```
myres@node1, myres@node2, myres@node3, ...
```

位于同一资源组中的两个资源之间的资源依赖性始终为 {LOCAL_NODE}。

如果不指定限定符，则默认情况下使用 FROM_RG_AFFINITIES。

[scha_resource_get\(1HA\) \[585\]](#)、[scha_resource_get\(3HA\) \[963\]](#) 和 [scds_property_functions\(3HA\) \[871\]](#) 手册页介绍了使用或不使用限定符获取依赖性列表的替代查询表单。

类别	可选
默认值	空列表
可调	随时

`Resource_dependencies_restart (string_array)`

该资源对其具有重新启动依赖性的资源（位于同一组中或不同组中）的列表。

该属性的工作方式就像 `Resource_dependencies` 一样，不同之处是，如果重新启动了重新启动依赖性列表中的任何资源，则该资源也会重新启动。在列表中的资源重新联机后再重新启动该资源。

如果列表中任何资源的启动失败，则无法启动该资源。如果该资源和列表中的资源之一同时启动，则 RGM 将等待列表中的资源启动后再启动该资源。

如果该资源的 `Resource_dependencies_restart` 列表中的资源未启动（例如，如果列表中资源的资源组保持脱机状态，或者如果列表中的资源处于 `Start_failed` 状态），则该资源保持脱机状态。如果该资源由于对无法启动的其他资源组中资源的依赖性而保持脱机状态，则该资源所属的组将进入 `Pending_online_blocked` 状态。

如果该资源与列表中的资源同时脱机，则该资源将在列表中的资源之前停止。但是，如果该资源保持联机状态或无法停止，则列表中的资源都会停止。

在资源组中，`Prenet_start` 方法在 `Start` 方法之前按依赖性顺序运行。`Postnet_stop` 方法在 `Stop` 方法之后按依赖性顺序运行。在不同的资源组中，依赖资源等待被依赖资源完成 `Prenet_start` 和 `Start` 后再运行 `Prenet_start`。被依赖资源等待依赖资源完成 `Stop` 和 `Postnet_stop` 后再运行 `Stop`。

要指定依赖性的范围，请在指定该属性时将以下限定符（包括花括号 {} 或 at 符号 @）附加到资源名称。

`{LOCAL_NODE}`

将指定的依赖性限制为基于每个节点。依赖资源的行为仅受同一节点上的被依赖资源影响。依赖资源等待被依赖资源在同一节点上启动。停止和重新启动时的情况与此类似。

如果 `{LOCAL_NODE}` 依赖资源在一个故障转移（即，单一主节点）资源组中，而且在一个节点上未满足 `{LOCAL_NODE}` 依赖性，则该资源组可能故障转移到满足 `{LOCAL_NODE}` 依赖性的其他节点，而不是在未满足依赖性的节点上保持 `Pending_online_blocked` 状态。

`{ANY_NODE}`

将指定的依赖性扩展到任何节点。依赖资源的行为受任何节点上的被依赖资源影响。依赖资源等待被依赖资源在任何主节点上启动后再启动自身。停止和重新启动时的情况与此类似。

`@nodename`

指定限制于特定节点且对其他节点无影响的 `{LOCAL_NODE}` 依赖性。这允许资源对群集中的每个节点具有不同的依赖性。`nodename` 是节点名称或节点 ID。

例如，以下列表指示资源 `res1` 对节点 `node1` 的依赖性和资源 `res2` 对节点 `node2` 的依赖性：

```
res1@node1, res2@node2
```

如果相同的依赖性适用于多个节点，请对每个节点名称重复资源名称。例如：

myres@node1,myres@node2,myres@node3,...

{FROM_RG_AFFINITIES}

指定资源依赖性的范围派生自资源所属的资源组的 `RG_affinities` 关系。如果依赖资源所属的组与被依赖资源所属的资源组正关联，且它们在同一节点上启动或停止，则依赖性为 {LOCAL_NODE}。如果不存在此类正关联，或者如果这两个组在不同的节点上启动，则依赖性为 {ANY_NODE}。

位于同一资源组中的两个资源之间的资源依赖性始终为 {LOCAL_NODE}。

如果不指定限定符，则默认情况下使用 FROM_RG_AFFINITIES。

[scha_resource_get\(1HA\) \[585\]](#)、[scha_resource_get\(3HA\) \[963\]](#) 和 [scds_property_functions\(3HA\) \[871\]](#) 手册页介绍了使用或不使用限定符获取依赖性列表的替代查询表单。

类别	可选
默认值	空列表
可调	随时

`Resource_dependencies_weak (string_array)`

该资源对其具有弱依赖性的资源（位于同一组中或不同组中）的列表。弱依赖性确定组中方法调用的顺序。RGM 先调用该列表中资源的 `Start` 方法再调用该资源的 `Start` 方法。RGM 先调用该资源的 `Stop` 方法再调用列表中资源的 `Stop` 方法。如果列表中的资源无法启动或保持脱机状态，该资源仍然可以启动。

如果该资源及其 `Resource_dependencies_weak` 列表中的资源同时启动，则 RGM 等待列表中的资源启动后再启动该资源。如果列表中的资源未启动（例如，如果列表中资源的资源组保持脱机状态，或者列表中的资源处于 `Start_failed` 状态），则该资源将启动。该资源的 `Resource_dependencies_weak` 列表中的资源启动时，该资源所属的资源组可能暂时进入 `Pending_online_blocked` 状态。列表中的所有资源已启动或无法启动时，该资源将启动且其组重新进入 `Pending_online` 状态。

如果该资源与列表中的资源同时脱机，则该资源将在列表中的资源之前停止。但是，如果该资源保持联机状态或无法停止，则列表中的资源都会停止。

在资源组中，`Prenet_start` 方法在 `Start` 方法之前按依赖性顺序运行。`Postnet_stop` 方法在 `Stop` 方法之后按依赖性顺序运行。在不同的资源组中，依赖资源等待被依赖资源完成 `Prenet_start` 和 `Start` 后再运行 `Prenet_start`。被依赖资源等待依赖资源完成 `Stop` 和 `Postnet_stop` 后再运行 `Stop`。

要指定依赖性的范围，请在指定该属性时将以下限定符（包括花括号 {} 或 at 符号 @）附加到资源名称。

{LOCAL_NODE}

将指定的依赖性限制为基于每个节点。依赖资源的行为仅受同一节点上的被依赖资源影响。依赖资源等待被依赖资源在同一节点上启动。停止和重新启动时的情况与此类似。

如果 {LOCAL_NODE} 依赖资源在一个故障转移（即，单一主节点）资源组中，而且在一个节点上未满足 {LOCAL_NODE} 依赖性，则该资源组可能故障转移到满足 {LOCAL_NODE} 依赖性的其他节点，而不是在未满足依赖性的节点上保持 Pending_online_blocked 状态。

{ANY_NODE}

将指定的依赖性扩展到任何节点。依赖资源的行为受任何节点上的被依赖资源影响。依赖资源等待被依赖资源在任何主节点上启动后再启动自身。停止和重新启动时的情况与此类似。

@nodename

指定限制于特定节点且对其他节点无影响的 {LOCAL_NODE} 依赖性。这允许资源对群集中的每个节点具有不同的依赖性。*nodename* 是节点名称或节点 ID。

例如，以下列表指示资源 res1 对节点 node1 的依赖性和资源 res2 对节点 node2 的依赖性：

```
res1@node1, res2@node2
```

如果相同的依赖性适用于多个节点，请对每个节点名称重复资源名称。例如：

```
myres@node1, myres@node2, myres@node3, ...
```

{FROM_RG_AFFINITIES}

指定资源依赖性的范围派生自资源所属的资源组的 RG_affinities 关系。如果依赖资源所属的组与被依赖资源所属的资源组正关联，且它们在同一节点上启动或停止，则依赖性为 {LOCAL_NODE}。如果不存在此类正关联，或者如果这两个组在不同的节点上启动，则依赖性为 {ANY_NODE}。

位于同一资源组中的两个资源之间的资源依赖性始终为 {LOCAL_NODE}。

如果不指定限定符，则默认情况下使用 FROM_RG_AFFINITIES。

[scha_resource_get\(1HA\) \[585\]](#)、[scha_resource_get\(3HA\) \[963\]](#) 和 [scds_property_functions\(3HA\) \[871\]](#) 手册页介绍了使用或不使用限定符获取依赖性列表的替代查询表单。

类别	可选
默认值	空列表
可调	随时

Resource_name (string)

资源实例的名称。必须在群集配置中是唯一的，且在创建资源后不能进行更改。

类别	必需
默认值	无默认值

可调	从不
----	----

Resource_project_name (string)

与资源关联的 Oracle Solaris 项目名称 (请参见 [Unresolved link to " projects1"](#))。使用该属性可以将 Solaris 资源管理功能 (例如 CPU 份额和资源池) 应用于群集数据服务。RGM 使资源联机时, 它将启动该项目名称下的相关进程。如果未指定该属性, 则将从包含该资源的资源组的 RG_project_name 属性获取项目名称 (请参见 [rg_properties\(5\) \[1129\]](#) 手册页)。如果未指定任一属性, 则 RGM 将使用预定义的项目名称 default。指定的项目名称必须在项目数据库中存在 (请参见 [Unresolved link to " projects1"](#) 手册页和 [Unresolved link to " Oracle Solaris Zones 介绍"](#))。

注 - 对该属性的更改在下次启动资源时生效。

类别	可选
默认值	Null
可调	随时
有效值	任何有效的 Oracle Solaris 项目名称, 或 null

每个群集节点上的 Resource_state (enum)

RGM 为每个群集节点上的资源确定的状态。可能的状态包

括: Online、Offline、Start_failed、Stop_failed、Monitor_failed、Online_not_monitored、Start和 Stopping。

Online

启动方法 (Prenet_start、Start 和 Monitor_start) 已在该节点的资源上成功执行。

Offline

资源尚未在该节点上首次启动, 或者停止方法 (Monitor_stop、Stop 和 Postnet_stop, 适用于特定的资源) 已在该节点的资源上成功执行。

Start_failed

Prenet_start 或 Start 方法在该节点的资源上失败。Start_failed 表示方法以非零退出状态退出或超时。资源表示的服务可能已在也可能未在该节点上实际启动。

Stop_failed

Monitor_stop、Stop 或 Postnet_stop 方法在该节点的资源上失败。Stop_failed 表示方法以非零退出状态退出或超时。资源表示的服务可能已在也可能未在该节点上实际停止。

在资源进入该状态时，资源组状态将变为 `Error_stop_failed` 并要求您干预。 [rg_properties\(5\) \[1129\]](#) 手册页中对 `Error_stop_failed` 进行了更详细的描述。

Monitor_failed

资源已成功执行其 `Prenet_start` 或 `Start` 方法（适用于特定的资源类型）。但是，资源的 `Monitor_start` 方法以非零退出状态退出或超时。资源监视器可能已在也可能未在该节点上实际启动。

Online_not_monitored

资源已成功执行其 `Prenet_start` 或 `Start` 方法（适用于特定的资源类型）。 `Monitor_start` 方法尚未在资源上执行。在资源组进入 `Online` 状态时，未受监视的资源（即，它没有 `Monitor_start` 方法，或者已禁用对它的监视）保持该状态。

Starting

资源正在运行 `Prenet_start` 或 `Start` 方法以尝试联机。

Stopping

资源正在运行 `Start` 或 `Postnet_stop` 方法以尝试脱机。
不能配置该属性。

类别	仅查询
默认值	无默认值
可调	从不

Retry_count (integer)

监视器在启动资源失败时尝试重新启动该资源的次数。如果超过 `Retry_count`，则根据特定的数据服务和 `Failover_mode` 属性的设置，监视器可能执行以下操作之一：

- 允许资源组保留在当前主节点上，即使资源处于故障状态也是如此
- 请求将资源组故障转移到其他节点上

仅当在 `RTR` 文件中声明了该属性，该属性才会由 `RGM` 创建且可供群集管理员使用。如果在 `RTR` 文件中指定了默认值，则该属性是可选的。

如果在资源类型文件中未指定 `Tunable` 特性，则该属性的 `Tunable` 值是 `When_disabled`。

如果为该属性指定一个负值，则监视器将尝试重新启动资源无限次。

注 - 有些资源类型不允许将 `Retry_count` 设置为一个负值。指定无限制重新启动的更可靠方法是执行以下操作：

- 将 `Retry_interval` 设置为小值，如 1 或 0。
- 将 `Retry_count` 设置为大值，如 1000。

类别	有条件的
默认值	请参见上文
可调	禁用时

`Retry_interval (integer)`

对尝试重新启动失败的资源进行计数的秒数。资源监视器将该属性与 `Retry_count` 一起使用。仅当在 RTR 文件中声明了该属性，该属性才会由 RGM 创建且可供群集管理员使用。如果在 RTR 文件中指定了默认值，则该属性是可选的。

如果在资源类型文件中未指定 Tunable 特性，则该属性的 Tunable 值是 `When_disabled`。

注 - 如果未声明 `Retry_interval` 属性，则对 `scha_resource_get (num_*_restarts)` 的调用将失败，退出状态为 13 (`SCHA_ERR_RT`)。

类别	有条件的
默认值	请参见上文
可调	禁用时

`Round_robin (boolean)`

采用循环方式将传入请求分配给特定服务器节点，该方式考虑了分配给每个节点的相对 `load_balancing_weight` 值。对于具有非粘性 `load_balancing_policy` 设置的资源，将基于连接分配请求；否则，将基于每个客户机 IP 地址分配请求。

对于需要对传入请求进行确定性负载分配的资源（期望有少量连接或客户机），应启用 `Round_Robin`。

资源属性 `Conn_threshold` 和群集属性 `udp_session_timeout` 支持循环方案，如果为服务设置 `Round_robin` 资源属性，可以根据需要配置这两个属性。

无需升级现有的资源类型注册 (resource type registration, RTR) 文件即可使用 `Round_robin` 属性。

类别	可选
默认值	FALSE

可调 禁用时

Scalable (boolean)

指示资源是否可伸缩，即，资源是否使用 Oracle Solaris Cluster 软件的网络负载平衡功能。

如果在 RTR 文件中声明了该属性，则 RGM 将为该类型的资源自动创建以下可伸缩服务属性：

Affinity_timeout、Load_balancing_policy、Load_balancing_weights、Network_resources_used、和 Weak_affinity。除非在 RTR 文件中显式声明这些属性，否则这些属性具有其默认值。在 RTR 文件中声明 Scalable 时，其默认值是 True。

如果在 RTR 文件中声明了该属性，则不允许为它分配除 At_creation 之外的 Tunable 特性。

如果在 RTR 文件中未声明该属性，则资源不是可伸缩的，您不能调整该属性，并且 RGM 未设置可伸缩服务属性。但是，如果需要，可以在 RTR 文件中显式声明 Network_resources_used 和 Port_list 属性，因为这些属性在不可伸缩的服务中以及可伸缩服务中可能是很有用的。

将 Scalable 资源属性与 Failover 资源类型属性组合使用，如下所示：

Failover/Scalable	描述
True/True	请勿指定这个不合逻辑的组合。
True/False	可为故障转移服务指定此组合。
False/True	可为使用 SharedAddress 资源进行网络负载平衡的可伸缩服务指定此组合。 Unresolved link to " Oracle Solaris Cluster Concepts Guide " 更详细地介绍了 SharedAddress。
False/False	使用该组合可配置不使用网络负载平衡的多主服务。

[rt_properties\(5\) \[1143\]](#) 手册页中对 Failover 资源类型属性的描述包含相关的附加信息。

类别 可选

默认值 请参见上文

可调 创建时

每个群集节点上的 Status (enum)

由资源监视器设置。可能值如下：Online、Degraded、Faulted、Unknown 和 Offline。启动资源时 RGM 将值设置为 Online（如果 Start（或 Prenet_start）方法尚未设置它）。停止资源时 RGM 将值设置为 Offline（如果 Stop（或 Postnet_stop）方法尚未设置它）。

类别 仅查询

默认值	无默认值
可调	仅通过使用 <code>scha_resource_setstatus</code> 命令

每个群集节点上的 `Status_msg` (string)

由资源监视器在设置 `Status` 属性的同时设置。使资源进入 `Offline` 状态时 RGM 将它设置为空字符串 (如果 `Stop` (或 `Postnet_stop`) 方法尚未设置它)。

类别	仅查询
默认值	无默认值
可调	仅通过使用 <code>scha_resource_setstatus</code>

`Thorough_probe_interval` (integer)

资源的高开销故障探测的两次调用之间的秒数。仅当在 RTR 文件中声明了该属性, 该属性才会由 RGM 创建且可供群集管理员使用。如果在 RTR 文件中指定了默认值, 则该属性是可选的。

如果在资源类型文件中未指定 `Tunable` 特性, 则该属性的 `Tunable` 值是 `When_disabled`。

类别	有条件的
默认值	无默认值
可调	禁用时

`Timeout_delay` (boolean)

确定是否向 `hatimerun` 命令传递 `-d` (`delay`, 延迟) 选项。一个声明 `Timeout_delay` 资源属性在使用 `hatimerun` 命令施加的时间限制下执行命令的资源类型。`-d` 选项会延迟启动超时时钟, 直到命令已开始执行, 这避免了将执行前调度延迟计入所分配的时间期限。

每个声明了此属性的资源类型都将以特定于该资源类型的方式使用此属性。例如, `ORCL.gds` 使用 `hatimerun(1HA)` 调用探测命令。有关详细信息, 请查阅每个数据服务的文档。

`Timeout_delay` 属性的默认值为 `FALSE` 并且默认可调性为 `Any time`。可在 RTR 文件中覆盖这些属性。

类别	有条件的/可选
默认值	<code>False</code>
可调	随时

Type (string)

实例的资源类型。

类别 必需

默认值 无默认值

可调 从不

Type_version (string)

指定当前与该资源关联的资源类型版本。RGM 自动创建该属性，不能在 RTR 文件中声明它。该属性的值与资源类型的 RT_version 属性的相等。创建资源时，不显式指定 Type_version 属性，尽管它可能作为资源类型名称的后缀出现。编辑资源时，Type_version 可能更改为一个新值。

类别 请参见上文

默认值 无

可调 可调性派生自以下各项：

- 资源类型的当前版本。
- 资源类型注册文件中的 `#$upgrade_from` 指令（请参见 [rt_reg\(4\) \[1067\]](#) 手册页）。

UDP_affinity (boolean)

如果为 true，则来自给定客户机的所有 UDP 流量都将发送到当前处理该客户机的所有 TCP 流量的同一服务器节点。

仅当 Load_balancing_policy 为 Lb_sticky 或 Lb_sticky_wild 时该属性才相关。此外，Weak_affinity 必须设置为 FALSE（默认值）。

该属性仅用于可伸缩服务。

类别 有条件的/可选

默认值 False

可调 禁用时

Weak_affinity (boolean)

如果为 true，则启用弱形式的客户机关联。这允许将来自给定客户机的连接发送到同一服务器节点，例外情况是服务器侦听器启动时（例如，由于监视器的重新启动发生故障、资源故障转移或切换或者节点在发生故障后重新加入群集）或者可伸缩资源的 load_balancing_weights 由于管理操作而改变时。

在内存占用和处理器周期这两方面，弱关联提供默认形式的低开销替代选择。

仅当 `Load_balancing_policy` 为 `Lb_sticky` 或 `Lb_sticky_wild` 时该属性才相关。
该属性仅用于可伸缩服务。

类别 有条件的/可选

默认值 `False`

可调 禁用时

[Unresolved link to "](#)

[projects1"](#)、[clresource\(1CL\) \[225\]](#)、[clresourcegroup\(1CL\) \[253\]](#)、[clresourcetype\(1CL\) \[277\]](#)、[scha_co](#)

[Unresolved link to " Oracle Solaris Cluster Concepts Guide "](#)、[Unresolved link to "](#)
[Oracle Solaris Zones 介绍 "](#)。

名称

SUNW.rac_framework, rac_framework — 启用 Oracle Solaris Cluster Support for Oracle Real Application Clusters (Oracle RAC) 的框架的资源类型实现

SUNW.rac_framework 资源类型表示启用 Oracle Solaris Cluster Support for Oracle RAC 的框架。此资源类型允许您监视该框架的状态。

SUNW.rac_framework 资源类型是单实例资源类型。在群集中只能创建一个此类型的资源。

要注册此资源类型并创建该资源类型的实例，请使用以下方法之一：

- Oracle Solaris Cluster Manager
- 使用 `clsetup(1CL)` [407] 实用程序，并指定用于配置 Oracle Solaris Cluster Support for Oracle Real Application Clusters 的选项
- 以下 Oracle Solaris Cluster 维护命令序列：
 1. 要注册此资源类型，请使用 `clresourcetype(1CL)` [277] 命令。
 2. 要创建此资源类型的实例，请使用 `clresource(1CL)` [225] 命令。

当此类资源在某个节点上脱机时，从联机状态转换为脱机状态需要一定的时间才能完成。在转换为脱机状态期间，该资源将继续参与重新配置过程。但是，资源在某个节点上处于脱机状态时，对其属性的更改直到此资源重新联机才会在该节点上生效。Oracle Solaris Cluster 在此类资源被禁用时会针对此效果显示一条警告消息。

将包含此类资源的资源组的状态转换为不受管理需要一定的时间才能完成。在转换为不受管理状态期间，Oracle RAC 框架将继续参与框架重新配置过程。但是，资源组处于不受管理状态时，对资源属性的更改不会在节点上生效。要停止 Oracle RAC 框架，必须重新引导节点。

SUNW.rac_framework 资源类型的扩展属性如下所示：

`reservation_timeout`

键入整数，最小值为 100，最大值为 99999，默认值为 325。此属性为 Oracle Solaris Cluster Support for Oracle RAC 重新配置过程的保留步骤指定超时值（以秒为单位）。您可以随时修改该属性。

例 398 创建 rac_framework 资源

此示例注册 SUNW.rac_framework 资源类型并创建 SUNW.rac_framework 资源类型的名为 rac_framework-rs 的实例。该示例假定已创建了一个名为 rac-framework-rg 的资源组。

```
phys-host-sc1# clresourcetype register SUNW.rac_framework
```

```
phys-host-scl# clresource create -g rac-framework-rg \  
-t SUNW.rac_framework rac_framework-rs
```

例 399 更改 rac_framework 资源的属性

此示例将 Oracle Solaris Cluster Support for Oracle RAC 重新配置过程的保留步骤的超时值设置为 350 秒。示例假定已创建一个类型为 SUNW.rac_framework 的名为 rac_framework-rs 的资源。

```
phys-host-scl# clresource set  
\-p reservation_timeout=350 rac_framework-rs
```

注 - 有关创建卷管理器资源使用的 SUNW.vucmm_framework 资源的示例，请参见 [SUNW.vucmm_framework\(5\) \[1231\]](#)。

有关以下属性的描述，请参见 [Unresolved link to " attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/library/ucmm

[clresource\(1CL\) \[225\]](#)、[clresourcetype\(1CL\) \[277\]](#)、[clsetup\(1CL\) \[407\]](#)、[Unresolved link to " attributes5"](#)、[SUNW.vucmm_framework\(5\) \[1231\]](#)

[Unresolved link to " 适用于 Oracle Real Application Clusters 的 Oracle Solaris Cluster 数据服务指南 "](#)

名称

rg_properties — 资源组属性

以下信息描述由 Oracle Solaris Cluster 定义的资源组属性。

资源组属性和描述

注 - 资源组属性名 (如 `Auto_start_on_new_cluster` 和 `Desired primaries`) 不区分大小写。在指定资源组属性名时, 可以使用大写和小写字母的任何组合。

`Auto_start_on_new_cluster` (boolean)

该属性控制在新群集形成时资源组管理器 (Resource Group Manager, RGM) 是否自动启动资源组。默认值为 `TRUE`。

如果设置为 `TRUE`, 则同时重新引导群集中的所有节点时, RGM 将尝试自动启动资源组以达到 `Desired primaries`。

如果设置为 `FALSE`, 则在重新引导群集时资源组不自动启动。资源组将保持脱机状态, 直到通过使用 `clresourcegroup(1CL)` [253] 命令或等效的图形用户界面命令首次将资源组手动切换为联机。之后, 资源组恢复正常的故障转移行为。

默认值 `TRUE`

可调 随时

`Desired primaries` (integer)

资源组可以在其上同时运行的所需数目的节点。

默认值为 1。 `Desired primaries` 属性的值必须小于或等于 `Maximum primaries` 属性的值。

默认值 1, 请参见上文

可调 随时

`Failback` (boolean)

一个布尔值, 它指示当某个节点加入群集或者某个节点上满足资源组的强正关联时, 是否重新计算资源组联机的一组节点。若重新计算, RGM 可以使组在次优先节点上脱机, 在更为优先的节点上联机。有关更多信息, 请参见 `RG_affinities` 属性。

默认值 `FALSE`

可调 随时

Global_resources_used (string_array)

指示群集文件系统是否由该资源组中的任何资源使用。管理员可以指定的合法值为星号 (*) 和空字符串 ("")，前者指示所有的全局资源，后者指示无全局资源。

默认值 所有的全局资源

可调 随时

Implicit_network_dependencies (boolean)

一个布尔值，它指示为 TRUE 时，RGM 应该在组中强制非网络地址资源对网络地址资源的隐式强依赖性。这表示 RGM 会在启动组中所有其他资源前先启动所有网络地址资源，在停止所有其他资源后再停止网络地址资源。网络地址资源包括逻辑主机名和共享地址资源类型。

在可伸缩的资源组中，该属性不起作用，因为可伸缩的资源组不包含任何网络地址资源。

默认值 TRUE

可调 随时

Load_factors

确定资源组所使用的负载占负载限制的比例。

可以为每个节点配置负载限制，将为资源组分配一组负载因子，这些负载因子与该节点定义的负载限制相对应。RGM 使资源组联机时，会将每个节点上资源组的负载因子累加，以提供与该节点的负载限制相比较的总负载。资源组的负载分配策略还受 Priority 和 Preemption_mode 属性的设置影响。有关更多信息，请参见 Preemption_mode 和 Priority 属性。

可以使用 clresourcegroup set-p 选项设置 load_factors 属性的值。load_factors 属性具有一个复合值，该值由格式为 limitname@value 的零个或多个元素的逗号分隔列表组成，其中 limitname 是标识符字符串，value 是非负整数。每个负载因子的默认值是 0，允许的最大值是 1000。在资源组的节点列表中的任何节点上，如果 limitname 未定义为 loadlimit，则认为它在该节点上是无限限制的。

如果一组资源组使用公共负载因子，则将在节点之间分配这些资源组，即使在这些节点上未指定对应的负载限制（即，无限制）也是如此。存在非零的负载因子可导致 RGM 分配负载。如果要避免基于负载的资源组分配，请删除负载因子或者将它们设置为零。

注 - 更改负载因子或负载限制时，当前处于脱机状态的某些资源组可能会自动联机。可以在资源组上执行 clresourcegroup suspend 命令以防止它自动联机。

有关更多信息，请参见 [clresourcegroup\(1CL\) \[253\]](#) 和 [clnode\(1CL\) \[149\]](#) 手册页。

超级用户以外的用户需要具有 solaris.cluster.admin RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

Maximum primaries (integer)

资源组可能同时在其中处于联机状态的节点的最大数目。

如果 RG_mode 属性为 Failover，则该属性的值不得大于 1。如果 RG_mode 属性为 Scalable，则允许大于 1 的值。

默认值 1，请参见上文

可调 随时

Nodelist (string_array)

可以按优先顺序使其中的组联机的节点的列表。这些节点称为资源组的潜在主节点。

默认值 按任意顺序的所有群集节点的列表

可调 随时

Pathprefix (string)

群集文件系统中，组资源可以编写所需管理文件的目录。某些资源可能需要此属性。使 Pathprefix 对每个资源组都是唯一的。

默认值 空字符串

可调 随时

Pingpong_interval (integer)

RGM 使用的非负整数值（秒），用于确定在重新配置时或者由于执行 scha_control 移交命令或函数而使资源组联机的位置。

对于重新配置情形，如果资源组在过去的 Pingpong_interval 秒内在特定的节点上多次联机失败（由于资源的 Start 或 Prenet_start 方法以非零状态退出或超时），则该节点将被视为没有资格托管资源组，RGM 将寻找其他主节点。

如果资源在给定的节点上执行 [scha_control\(1HA\) \[579\]](#) 命令或 [scha_control\(3HA\) \[939\]](#) 移交而导致其资源组故障转移到其他节点，则在经过 Pingpong_interval 秒之前，第一个节点（在其上调用了 scha_control）不能是同一资源执行的其他 scha_control 移交的目标。

默认值 3600 (1 小时)

可调 随时

Preemption_mode

确定某个资源组由于节点过载而被优先级较高的资源组从节点中抢占的可能性。

可以使用 clresourcegroup set-p 选项设置 preemption_mode 属性的枚举值。preemption_mode 属性的默认设置是 HAS_COST。

资源组的 `preemption_mode` 属性可以具有以下值之一：

- `HAS_COST` – 为满足负载限制，该资源组可能会被优先级较高的资源组从其当前主节点中取代。抢占该资源组具有与之关联的成本，因此 RGM 会通过选择其他节点来控制优先级较高的资源组，尝试尽可能避免该情况。
- `NO_COST` – 为满足负载限制，该资源组可能会被优先级较高的资源组从其当前主节点中取代。抢占该资源组的成本为零。
- `NEVER` – 不能从该资源组的当前主节点中取代它以满足负载限制。

有关更多信息，请参见 [clresourcegroup\(1CL\) \[253\]](#) 和 [clnode\(1CL\) \[149\]](#) 手册页。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

Priority

确定资源组分配给主节点所依照的顺序。优先级越高，表示服务就越重要。

可以使用 `clresourcegroup set-p` 选项设置 `priority` 属性的无符号整数值。优先级值比其他组高的资源组优先，它很可能由其优先节点控制，且不大可能从该节点中被取代。`priority` 属性的默认值为 500。

如果两个资源组具有相等的优先级并通过 `RG_dependencies` 或强 `RG_affinities` 相关，则未指定依赖性 or 关联的资源组将在依赖资源组之前收到其节点分配。如果两个资源组具有相等的优先级但未通过依赖性 or 强关联相关，则按任意顺序为它们分配主节点。

有关更多信息，请参见 [clresourcegroup\(1CL\) \[253\]](#) 和 [clnode\(1CL\) \[149\]](#) 手册页。

超级用户以外的用户需要具有 `solaris.cluster.admin` RBAC 授权才能使用该子命令。请参见 [Unresolved link to "rbac5"](#) 手册页。

Resource_list (string_array)

包含在组中的资源的列表。管理员不直接设置该属性。而是在管理员向资源组中添加或从中删除资源时由 RGM 更新该属性。

默认值 无默认值

可调 从不

RG_affinities (string)

RGM 将尝试：(1) 在为另一个给定资源组的当前主节点的计算机上查找资源组（正关联），或者 (2) 在不是给定资源组的当前主节点的计算机上查找资源组（负关联）。

可以将 `RG_affinities` 设置为以下字符串：

- + 或弱正关联
- ++ 或强正关联
- +++ 或具有故障转移委托的强正关联
- - 或弱负关联

- -- 或强负关联

例如，RG_affinities=+RG2,--RG3 指示该资源组具有与 RG2 的弱正关联和与 RG3 的强负关联。

在[Unresolved link to " Oracle Solaris Cluster 数据服务规划和管理指南 中的第 2 章 数据服务资源管理"](#)中描述了如何使用 RG_affinities。

默认值 空字符串

可调 随时

有时配置单计算机群集以达到原型开发目的。如果将资源组配置为在此类群集的多个节点上运行，则在节点级别而不是计算机级别上解释 RG_affinities。例如，强正关联要求两个资源组运行在同一节点中，而不仅仅运行在同一计算机上。请注意，单计算机群集上的所有节点都是同一计算机上的区域。

RG_dependencies (string_array)

指示使其他组在同一节点上联机或脱机的优先排序的资源组可选列表。所有强 RG_affinities (正和负) 以及 RG_dependencies 的图形不允许包含循环。

例如，假定在资源组 RG1 的 RG_dependencies 列表中列出了资源组 RG2。换句话说，假定 RG1 具有对 RG2 的资源组依赖性。以下列表总结了该资源组依赖性的效果：

- 在节点加入群集时，在该节点上的所有 Boot 方法在 RG2 中的资源上完成之前，不会在 RG1 中的资源上运行该节点上的 Boot 方法。
- 如果 RG1 和 RG2 在同一节点上同时处于 Pending_online 状态，则在 RG2 中的所有资源完成其启动方法之前，不会在 RG1 中的任何资源上运行启动方法 (Preboot_start 或 Start)。
- 如果 RG1 和 RG2 在同一节点上同时处于 Pending_offline 状态，则在 RG1 中的所有资源完成其停止方法之前，不会在 RG2 中的任何资源上运行停止方法 (Stop 或 Postboot_stop)。
- 如果切换主节点使 RG1 在任何节点上保持联机状态而使 RG2 在所有节点上保持脱机状态，则切换 RG1 或 RG2 的主节点的尝试将失败。
- 如果 Desired_primaries 在 RG2 上设置为零，则不允许在 RG1 上将 Desired_primaries 属性设置为大于零的值。
- 如果 Auto_start_on_new_cluster 在 RG2 上设置为 FALSE，则不允许在 RG1 上将 Auto_start_on_new_cluster 属性设置为 TRUE。

默认值 空列表

可调 随时

RG_description (string)

资源组的简要描述。

默认值 空字符串

可调	随时
----	----

RG_is_frozen (boolean)

一个布尔值，它指示资源组所依赖的全局设备是否要进行切换。如果该属性设置为 TRUE，则全局设备要进行切换。如果该属性设置为 FALSE，则全局设备不进行切换。资源组依赖于全局设备的情况由其 Global_resources_used 属性指示。

不直接设置 RG_is_frozen 属性。RGM 在全局设备的状态改变时会更新 RG_is_frozen 属性。

默认值	无默认值
-----	------

可调	从不
----	----

RG_mode (enum)

指示资源组是故障转移组还是可伸缩组。如果值为 Failover，则 RGM 将组的 Maximum primaries 属性设置为 1，并将资源组限制为由单个节点控制。

如果该属性的值为 Scalable，则 RGM 允许将 Maximum primaries 属性设置为大于 1 的值。因此，该组可以同时由多个节点控制。RGM 不允许将其 Failover 属性为 TRUE 的资源添加到其 RG_mode 为 Scalable 的资源组。

如果 Maximum primaries 为 1，则默认值是 Failover。如果 Maximum primaries 大于 1，则默认值是 Scalable。

默认值	依赖于 Maximum primaries 的值
-----	--------------------------

可调	创建时
----	-----

RG_project_name (字符串)

与资源组关联的 Solaris 项目名称（请参见 [Unresolved link to " projects1"](#)）。使用该属性可以将 Solaris 资源管理功能（例如 CPU 份额和资源池）应用于群集数据服务。RGM 使资源组联机时，它将对未设置 Resource_project_name 属性的资源启动该项目名称下的相关进程（请参见 [Unresolved link to " r_properties\(5\)"](#)）。指定的项目名称在项目数据库中必须存在（请参见 [Unresolved link to " projects1"](#) 和 [Unresolved link to " Oracle Solaris Zones 介绍"](#)）。

注 - 对该属性的更改在下次启动资源时生效。

默认值	文本字符串 "default"
-----	-----------------

可调	随时
----	----

有效值	任何有效的 Solaris 项目名称
-----	--------------------

RG_SLM_CPU_SHARES (整数)

与资源组关联的 CPU 份额数。

注 - 仅当 `RG_SLM_TYPE` 设置为 `automated` 时才能设置 `RG_SLM_CPU_SHARES` 属性。有关更多信息，请参见 `RG_SLM_TYPE` 属性。

`RG_SLM_CPU_SHARES` 的最大值是 65535。零不是 `RG_SLM_CPU_SHARES` 的可接受值，因为将份额值设置为零会导致 CPU 重载时不调度进程。在资源组联机时对 `RG_SLM_CPU_SHARES` 进行的更改将动态应用。

由于 `RG_SLM_TYPE` 设置为 `automated`，因此 Oracle Solaris Cluster 将创建一个名为 `SCSLM_resourcegroup-name` 的 [Unresolved link to "project4"](#)，其中 `resourcegroup-name` 是您为资源组提供的名称。在该项目中将执行属于资源组的资源的每个方法。这些项目是在资源组的区域（全局区域）中创建的。

项目 `SCSLM_resourcegroup-name` 具有一个设置为 `RG_SLM_CPU_SHARES` 值的 `project.cpu-shares` 值。如果未设置 `RG_SLM_CPU_SHARES` 属性，则通过 `project.cpu-shares` 值 1 创建该项目。

当 `RG_SLM_PSET_TYPE` 属性设置为 `strong` 或 `weak` 时，`RG_SLM_CPU_SHARES` 属性的值也用于计算所创建的 `pset` 的大小（按照约定，100 个份额相当于一个 CPU）。有关更多信息，请参见 `RG_SLM_PSET_TYPE` 属性。

有关处理器集的信息，请参见 [Unresolved link to "Oracle Solaris Zones 介绍"](#)。

默认值	1
可调	随时

`RG_SLM_PSET_MIN` (integer)

资源组在其中执行的处理器集中处理器的最小数目。仅在符合以下条件时，才可使用该属性：

- 使用的操作系统为 Solaris 11。
- `RG_SLM_TYPE` 设置为 `automated`。
- `RG_SLM_PSET_TYPE` 设置为 `strong` 或 `weak`。（请参见 `RG_SLM_PSET_TYPE` 属性。）
- `RG_SLM_PSET_MIN` 的值必须小于或等于 `RG_SLM_CPU_SHARES` 除以 100 所得到的值。

`RG_SLM_PSET_MIN` 的最大值是 655。Oracle Solaris Cluster 使用 `RG_SLM_PSET_MIN` 属性的值来计算处理器集的最小大小。

在资源组联机时对 `RG_SLM_CPU_SHARES` 和 `RG_SLM_PSET_MIN` 进行的更改将动态应用。但是，如果 `RG_SLM_PSET_TYPE` 设置为 `strong`，则当没有足够的 CPU 可用于适应更改时，不会应用为 `RG_SLM_PSET_MIN` 请求的更改。此时，屏幕显示一条警告消息。在下次切换时，如果没有足够的 CPU 可用于满足为 `RG_SLM_PSET_MIN` 配置的值，则可能会出现因缺少 CPU 而导致的错误。

有关处理器集的信息，请参见 [Unresolved link to "Oracle Solaris Zones 介绍"](#)。

默认值	0
可调	随时

RG_SLM_PSET_TYPE (string)

允许创建专用处理器集。

RG_SLM_PSET_TYPE 的可能值是 default、strong 和 weak。

如果符合以下所有条件，则可以将 RG_SLM_PSET_TYPE 设置为 strong 或 weak：

- 使用的操作系统为 Solaris 11。
- 资源组配置为仅在非全局区域中执行。
- RG_SLM_TYPE 设置为 automated。

RG_SLM_PSET_TYPE 的可能值是 default、strong 和 weak。

为了使资源组在该属性为 strong 或 weak 时执行，必须配置该资源组，以便在其节点列表中只有非全局区域。

不得为除默认池 (pool_default) 之外的池配置非全局区域。有关区域配置的信息，请参见 [Unresolved link to "zonecfg1M"](#)。该非全局区域不得动态绑定到除默认池之外的池。有关池绑定的更多信息，请参见 [Unresolved link to "poolbind1M"](#)。仅当启动资源组中资源的方法时才验证这两种池情况。

对于在其节点列表中具有相同区域的资源组，值 strong 和 weak 是互斥的。不能将同一区域中的资源组配置为如下情形：一些资源组的 RG_SLM_PSET_TYPE 设置为 strong，而其他资源组的该属性设置为 weak。

如果 RG_SLM_PSET_TYPE 设置为 strong 或 weak，且为 RG_SLM_TYPE 列出的操作设置为 automated，则使资源组联机时，Oracle Solaris Cluster 将执行以下操作：

- 创建一个池，并将该池动态绑定到启动资源组的非全局区域。
- 创建一个大小介于最小值和最大值之间的处理器集。
 - 最小值是启动该资源组的区域中所有联机资源组的 RG_SLM_PSET_MIN 值的总和，或者是 1（如果该总和等于零）。
 - 最大值是该区域中所有联机资源组的 RG_SLM_SPU_SHARES 值的总和除以 100 并向上舍入到紧邻的整数，或者是 1（如果计算结果为零）。
- 将处理器集与池关联。
- 将 zone.cpu-shares 设置为在区域中运行的所有资源组中 RG_SLM_CPU_SHARES 的总和。

如果 RG_SLM_PSET_TYPE 设置为 strong 或 weak，则使资源组脱机（更确切地说，在执行资源组的第一个资源的 STOP 或 POSTNET_STOP 方法时），如果区域中不再存在任何联机资源组，则 Oracle Solaris Cluster 将销毁处理器集，销毁池，并将区域绑定到默认池 (pool_default)。

如果 RG_SLM_PSET_TYPE 设置为 strong，则资源组的行为就好像 RG_SLM_PSET_TYPE 设置为 strong 一样。但是，如果没有足够的处理器可用于创建处理器集，则池将与默认的处理集关联。

如果 RG_SLM_PSET_TYPE 设置为 strong，且没有足够的处理器可用于创建处理器集，则一个错误将返回到资源组监视器 (Resource Group Monitor, RGM)，且在该节点或区域上不会启动资源组。

CPU 分配的优先级顺序为：defaultpsetmin 最小大小的优先级高于 strong（其优先级高于 weak）。（有关 defaultpsetmin 属性的信息，请参见 [clnode\(1CL\) \[149\]](#)。）但是，在通过使用 clnode 命令尝试增加默认处理器集的大小但没有足够的可用处理器时，不会保持该优先级。

如果通过使用 clnode 命令将最小数量的 CPU 分配给默认处理器集，则操作是动态完成的。如果指定的 CPU 数不可用，则 Oracle Solaris Cluster 定期地重新尝试为默认处理器集分配该数量的 CPU，随后分配更小数量的 CPU，直到分配最小数量的 CPU。该操作可能会销毁某些 weak 处理器集，但是不会销毁 strong 处理器集。

在其 RG_SLM_PSET_TYPE 配置为 strong 的资源组启动时，它可能会销毁与 weak 处理器集关联的处理器集（如果在这两个处理器集的节点上没有足够的可用 CPU）。在该情况下，weak 处理器集中运行的资源组的处理器将与默认的处理器集关联。

要将处理器集从 weak 更改为 strong 或者从 strong 更改为 weak，必须首先将处理器集更改为将 RG_SLM_PSET_TYPE 设置为 default。

如果将 RG_SLM_PSET_TYPE 设置为 default，则 Oracle Solaris Cluster 将创建池 SCSLM_pool_zone-name，但是不创建处理器集。在这种情况下，SCSLM_pool_zone-name 与默认的处理器集关联。分配给区域的份额由为在该区域中运行的所有资源组的 RG_SLM_CPU_SHARES 设置的值的总和确定。

如果在非全局区域中不再有为 CPU 控制配置的任何联机资源组，则该非全局区域的 CPU 份额值采用在区域配置中找到的 zone.cpu-shares 的值。该参数的默认值为 1。有关区域配置的更多信息，请参见 [Unresolved link to "zonecfg1M"](#)。

有关资源池和处理器集的信息，请参见 [Unresolved link to "Oracle Solaris Zones 介绍"](#)。

默认值 文本字符串 "default"

可调 随时

RG_SLM_TYPE (string)

允许您控制系统资源用法，并自动执行一些步骤以在 Oracle Solaris OS 中配置系统资源管理。RG_SLM_TYPE 的可能值是 automated 和 manual。

如果 RG_SLM_TYPE 设置为 automated，则使资源组联机时，Oracle Solaris Cluster 将执行以下操作：

- 创建一个名为 SCSLM_resourcegroup-name 的项目。该资源组中资源的所有方法都将在该项目中执行。在节点或区域上首次执行该资源组中资源的方法时，会创建该项目。
- 将与项目关联的 project.cpu_shares 的值设置为 RG_SLM_CPU_SHARES 的值。默认情况下，project.cpu_shares 的值是 1。
- 将区域的 zone.cpu-shares 设置为所有资源组（其 RG_SLM_TYPE 设置为 automated）的 RG_SLM_CPU_SHARES 的总和。此区域可以是全局区域。有关专用处理器集的信息，请参见 RG_SLM_PSET_TYPE 属性。

当 RG_SLM_TYPE 设置为 automated 时，执行的任何操作都会导致记录消息。

如果 `RG_SLM_TYPE` 设置为 `manual`，则资源组将在 `RG_project_name` 属性指定的项目中执行。

有关资源池和处理器集的信息，请参见[Unresolved link to " Oracle Solaris Zones 介绍 "](#)。

-
- 不要指定超过 58 个字符的资源组名称。如果资源组名称包含 58 个以上的字符，则不能配置 CPU 控制，即不能将 `RG_SLM_TYPE` 属性设置为 `automated`。
 - 避免在资源组名称中包括短划线 (-)。Oracle Solaris Cluster 软件在创建项目时会将资源组名称中的所有短划线替换为下划线 ()。例如，Oracle Solaris Cluster 对于名为 `rg-dev` 的资源组创建名为 `SCSLM_rg_dev` 的项目。如果名为 `rg_dev` 的资源组已存在，则 Oracle Solaris Cluster 尝试为资源组 `rg-dev` 创建项目时将出现冲突。
-

默认值	<code>manual</code>
-----	---------------------

可调	随时
----	----

每个群集节点上的 `RG_state` (enum)

由 RGM 设置为

`Unmanaged`、`Online`、`Offline`、`Pending_online`、`Pending_offline`、`Error_stop_failed`、`Online_faulted` 或 `Pending_online_blocked` 以描述每个群集节点上资源组的状态。

不能配置该属性。但是，可以通过使用 `clresourcegroup(1CL)` [253] 或通过使用等效的 Oracle Solaris Cluster 图形用户界面命令，间接地设置该属性。当资源组不受 RGM 控制时，该组可以 `Unmanaged` 状态存在。

以下描述对每个状态进行了概述。

注 - 除了 `Unmanaged` 状态（它适用于所有的节点）外，状态仅适用于单个节点。例如，某个资源组在节点 A 上可能是 `Offline`，而在节点 B 上则为 `Pending_online`。

`Error_stop_failed`

资源组内的一个或多个资源无法成功停止且处于 `Stop_failed` 资源状态。组中的其他资源可能保持联机或脱机状态。在清除 `Error_stop_failed` 状态之前，不允许该资源组在任何节点上启动。

必须使用管理命令（如 `clresourcegroup clear`）手动中止 `Stop_failed` 资源并将其状态重置为 `Offline`。

`Offline`

资源组已在节点上停止。换句话说，停止方法（`Monitor_stop`、`Stop` 和 `Postnet_stop`，适用于每个资源）在组中的所有已启用资源上都已成功执行。资源组在节点上首次启动之前，该状态也适用。

Online

资源组已在节点上启动。换句话说，启动方法（`Prenet_start`、`Start` 和 `Monitor_start`，每个资源适用的方法）在组中的所有已启用资源上都已成功执行。

Online_faulted

资源组为 `Pending_online` 并在该节点上已完成启动。但是，一个或多个资源以 `Start_failed` 资源状态或以 `Faulted` 状态结束。

Pending_offline

资源组正在节点上停止。停止方法（`Monitor_stop`、`Stop` 和 `Postnet_stop`，适用于每个资源）正在组中的已启用资源上执行。

Pending_online

资源组正在节点上启动。启动方法（`Prenet_start`、`Start` 和 `Monitor_start`，适用于每个资源）正在组中的已启用资源上执行。

Pending_online_blocked

资源组无法完全启动，因为该资源组中的一个或多个资源对其他资源组中资源的强资源依赖性未得到满足。此类资源将保持 `Offline` 状态。在满足资源依赖性时，资源组将自动恢复为 `Pending_online` 状态。

Unmanaged

新创建的资源组的初始状态，或以前管理的资源组的状态。`Init` 方法尚未在组中的资源上运行，或者 `Fini` 方法已在组中的资源上运行。
组不受 RGM 管理。

默认值 无默认值

可调 从不

RG_system (boolean)

如果资源组的 `RG_system` 属性为 `TRUE`，则只能对该资源组和该资源组包含的资源执行特定的操作。该限制旨在帮助防止意外修改或删除关键的资源组和资源。只有 `clresource(1CL)` [225] 和 `clresourcegroup(1CL)` [253] 命令受该属性影响。`scha_control(1HA)` [579] 和 `scha_control(3HA)` [939] 的操作不受影响。

对资源组（或资源组的资源）执行受限制操作之前，必须先将资源组的 `RG_system` 属性设置为 `FALSE`。修改或删除支持群集服务的资源组时或者修改或删除此类资源组包含的资源时，要小心谨慎。

下表显示了 `RG_system` 设置为 `TRUE` 时对资源组限制的操作。

操作	示例
删除资源组	<code>clresourcegroup delete RG1</code>

操作	示例
编辑资源组属性 (RG_system 除外)	<code>clresourcegroup set -p RG_description=... +</code>
向资源组添加资源	<code>clresource create -g RG1 -t SUNW.nfs R1</code> 创建的资源处于启用状况，并开启了资源监视功能。
从资源组中删除资源	<code>clresource delete R1</code>
编辑属于资源组的资源的属性	<code>clresource set -g RG1 -t SUNW.nfs -p r_description="HA-NFS res" R1</code>
使资源组脱机	<code>clresourcegroup offline RG1</code>
管理资源组	<code>clresourcegroup manage RG1</code>
取消管理资源组	<code>clresourcegroup unmanage RG1</code>
启用资源	<code>clresource enable R1</code>
启用对资源的监视	<code>clresource monitor R1</code>
禁用资源	<code>clresource disable R1</code>
禁用对资源的监视	<code>clresource unmonitor R1</code>

如果资源组的 `RG_system` 属性为 `TRUE`，则唯一可以编辑的资源组属性是 `RG_system` 属性本身。换句话说，从不限编辑 `RG_system` 属性。

默认值 `FALSE`

可调 随时

`Suspend_automatic_recovery (boolean)`

一个布尔值，它指示资源组的自动恢复是否已暂停。在群集管理员没有显式发出命令继续执行自动恢复过程之前，已暂停的资源组不会自动重新启动或进行故障转移。无论是联机还是脱机，已暂停的数据服务都仍将处于其当前状态。

资源组已暂停时，可以通过使用带有子命令（如 `switch`、`online`、`offline`、`disable` 或 `enable`）的 `clresourcegroup(1CL)` 或 `clresource(1CL)` 命令，在特定的节点上将资源组或其资源手动切换到其他状态。无需直接对资源执行操作（如中止应用程序进程或运行应用程序特定的命令），只需使用 `clresourcegroup(1CL)` 或 `clresource(1CL)` 命令即可。这样，群集框架就可以保持资源和资源组的当前状态的准确情形，以便在执行 `resume` 子命令时正确恢复可用性。

如果 `Suspend_automatic_recovery` 属性设置为 `TRUE`，则暂停资源组的自动恢复。如果该属性设置为 `FALSE`，则资源组的自动恢复将继续执行并处于活动状态。

群集管理员不直接设置该属性。当群集管理员暂停或继续执行资源组的自动恢复时，RGM 会更改 `Suspend_automatic_recovery` 属性的值。群集管理员使用 `clresourcegroup suspend` 命令暂停自动恢复。群集管理员使用 `clresourcegroup resume` 命令继续执行自动恢复。可以暂停或继续执行资源组，而不管其 `RG_system` 属性的设置如何。

默认值 `FALSE`

可调 从不

Target_nodes (string_array)

资源组当前通过 `clrg switch`、`clrg remaster`、`scha_control` 移交或故障恢复操作切换到的节点的列表。如果资源组当前未通过这些操作之一切换到新节点，则查询返回空列表。

该查询通常由 `Stop` 或 `Postnet_stop` 停止方法来执行，以搜索是否存在组要切换到的指定的一个或多个目标节点。

默认值 无

可调 从不 (仅查询)

[Unresolved link to "](#)

[projects1"](#)、[clnode\(1CL\) \[149\]](#)、[clresource\(1CL\) \[225\]](#)、[clresourcegroup\(1CL\) \[253\]](#)、[scha_control\(1HA\)](#)
[link to " poolbind1M"](#)、[scha_control\(3HA\) \[939\]](#)、[Unresolved link to "](#)
[project4"](#)、[property_attributes\(5\) \[1087\]](#)、[r_properties\(5\) \[1103\]](#)、[rt_properties\(5\) \[1143\]](#)、[scha_resou](#)
和 [scha_resourcegroup_get\(3HA\) \[1013\]](#)。

[Unresolved link to " Oracle Solaris Cluster Concepts Guide "](#)、[Unresolved link to "](#)
[Oracle Solaris Cluster 数据服务规划和管理指南 "](#)、[Unresolved link to " Oracle Solaris](#)
[Zones 介绍 "](#)。

名称

rt_properties — 资源类型属性

以下信息介绍了 Oracle Solaris Cluster 软件定义的资源类型属性。这些描述已针对数据服务开发者进行开发。有关特定数据服务的信息，请参见该数据服务的手册页。

资源类型属性值

必需	该属性需要在资源类型注册 (Resource Type Registration, RTR) 文件中有显式值。否则，将无法创建该属性所属的对象。该值不允许为空白或空字符串。
有条件的	必须在 RTR 文件中声明该属性，它才会存在。否则，RGM 不会创建该属性，且该属性不能用于管理实用程序。允许使用空白或空字符串。如果该属性在 RTR 文件中已声明但未指定值，RGM 将提供默认值。
有条件的/显式	必须在 RTR 文件中声明该属性且赋予其显式值，它才会存在。否则，RGM 不会创建该属性，且该属性不能用于管理实用程序。不允许使用空白或空字符串。
可选	可在 RTR 文件中声明该属性。如果未在 RTR 文件中声明该属性，RGM 会创建它并提供默认值。如果在 RTR 文件中已声明该属性但未指定值，RGM 会提供相同的默认值，就像未在 RTR 文件中声明该属性一样。
仅查询	该属性不能通过管理实用程序直接进行设置。该属性不是在 RTR 文件中进行设置的。提供该属性的值仅供参考。

注 - 除 `Installed_nodes` 和 `RT_system` 之外，资源类型属性无法通过管理实用程序进行更新。`Installed_nodes` 无法在 RTR 文件中声明且只能由群集管理员设置。`RT_system` 既可以在 RTR 文件中指定有初始值，也可以由群集管理员设置。

资源类型属性和描述

资源类型是由资源类型注册文件定义的，该文件用于为资源类型指定标准属性值和扩展属性值。

注 - 资源类型属性名称 (例如 `API_version` 和 `Boot`) 不区分大小写。指定属性名称时，可以使用大小写字母的任意组合。

`API_version` (integer)

此资源类型实现所使用的资源管理 API 的版本。

以下信息汇总了每个 Oracle Solaris Cluster 软件发行版所支持的最高 API_version。

3.1 及 3.1 之前的版本	2
3.1 10/03	3
3.1 4/04	4
3.1 9/04	5
3.1 8/05	6
3.2	7
3.2 2/08	8
3.2 1/09	9
3.2 11/09	10
3.3	11
3.3 5/11	12
3.3 12/12	13
4.0	20
4.1	21
4.2	22

在 RTR 文件中为 API_version 声明一个大于 2 的值可防止该资源类型安装在支持较低最高版本的 Oracle Solaris Cluster 软件版本上。例如，如果您为某个资源类型声明 API_version=7，则该资源类型将不能安装在 Sun Cluster 3.2 发行版之前发行的任意版本的群集软件上。

类别	可选
默认值	2
可调	从不

Boot (string)

一种可选的回调方法：在发生以下情况时，RGM 对节点调用的程序的路径：

- 节点加入或重新加入群集。
- 包含此类资源的资源组处于受管状态。

预计该方法初始化此类资源的方式与 Init 方法相同。

类别	有条件的/显式
默认值	无
可调	从不

Failover (boolean)

如果将该属性设置为 TRUE，则此类型的资源将无法在可同时在多个节点上联机的任意组中配置。

可以按如下方式将该资源类型属性与 Scalable 资源属性结合使用：

Failover/Scalable	描述
TRUE/TRUE	请勿指定这个不合逻辑的组合。
TRUE/FALSE	可为故障转移服务指定此组合。
FALSE/TRUE	可为使用 SharedAddress 资源进行网络负载均衡的可伸缩服务指定此组合。 Unresolved link to " Oracle Solaris Cluster Concepts Guide " 更详细地介绍了 SharedAddress。
FALSE/FALSE	使用此组合可选择不使用网络负载均衡的多主机服务。

[r_properties\(5\) \[1103\]](#) 中对 Scalable 的描述和 [Unresolved link to " Oracle Solaris Cluster Concepts Guide 中的第 3 章 Key Concepts for System Administrators and Application Developers"](#) 介绍了相关的附加信息。

类别	可选
默认值	FALSE
可调	从不

Fini (string)

一种可选的回调方法：从 RGM 管理中删除此类资源时，RGM 所调用的程序的路径。

类别	有条件的/显式
默认值	无默认值

可调	从不
----	----

Global_zone (boolean)

如果将某个资源类型的该属性设置为 TRUE，在任何情况下，其方法均在全局区域中执行。如果将该属性设置为 TRUE，则即使在区域群集中配置了资源组，也会在全局区域中执行该方法。仅针对只能从全局区域管理的服务，如网络地址和文件系统，将该属性设置为 TRUE。



注意 - 不要注册 Global_zone 属性设置为 TRUE 的资源类型，除非该资源类型来自于已知的可信源。此属性设置为 TRUE 的资源类型会规避区域隔离并引起风险。

请勿在位于区域群集中的 RTR 文件中将 Global_zone 属性设置为 TRUE。该属性设置为 TRUE 的所有资源类型必须位于全局群集的全局区域中。

配置为在非全局区域中启动，并且其 Global_zone 属性设置为 TRUE 的资源的方法始终在全局区域中运行。此类资源配置在非全局区域中时，不会从 CPU 共享和专用处理器集配置中受益。即使将 RG_slm_type 属性设置为 AUTOMATED，该资源也不会受益。Oracle Solaris Cluster 软件将此类资源视为好像位于 RG_slm_type 属性设置为 MANUAL 的资源组中一样。

因为 Global_zone 属性设置为 TRUE 的资源类型的方法在全局区域中运行，所以当非全局区域停止时，RGM 不会立即将这些资源类型视为脱机。实际上，RGM 对这些资源类型（包括 LogicalHostname、SharedAddress 和 HAStoragePlus）运行 Monitor_stop、Stop 和 Postnet_stop 等方法。但是，当非全局区域停止时，RGM 将 Global_zone 属性设置为 FALSE 的资源视为脱机。RGM 无法对此类资源运行停止方法，因为这些方法必须在非全局区域中运行。

对于声明 Global_zone=TRUE 的资源类型，可能还需要声明 Global_zone_override 资源属性。在这种情况下，Global_zone_override 属性的值将代替该资源的 Global_zone 属性值。有关 Global_zone_override 属性的更多信息，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

类别	可选
----	----

默认值	FALSE
-----	-------

可调	从不
----	----

Init (string)

一种可选的回调方法：在此类资源受 RGM 管理时，RGM 所调用的程序的路径。

类别	有条件的/显式
----	---------

默认值	无默认值
-----	------

可调	从不
----	----

Init_nodes (enum)

指示 RGM 要对其调用 Init、Fini、Boot 和 Validate 方法的节点。可将该属性设置为 RG primaries (仅限可管理该资源的节点) 或 RT_installed_nodes (安装该资源类型的所有节点)。

类别	可选
默认值	RG primaries
可调	从不

Installed_nodes (string_array)

允许在其上运行该资源类型的节点的名称列表。RGM 将自动创建该属性。群集管理员可设置该值。无法在 RTR 文件中声明该属性。

类别	可由群集管理员进行配置
默认值	所有群集节点
可调	随时

Is_logical_hostname (boolean)

TRUE 表明该资源类型是管理故障转移 IP 地址的 LogicalHostname 资源类型的某个版本。

类别	仅查询
默认值	无默认值
可调	从不

Is_shared_address (boolean)

TRUE 表明该资源类型是管理共享 IP (Internet Protocol , Internet 协议) 地址的 SharedAddress 资源类型的某个版本。

类别	仅查询
默认值	无默认值
可调	从不

Monitor_check (string)

一种可选的回调方法：在对此类资源执行监视器请求的故障转移之前，RGM 所调用的程序的路径。如果监视器检查程序在某节点上以非零值退出，则会阻止对该节点的任何故障转移尝试操作。

类别	有条件的/显式
----	---------

默认值	无默认值
-----	------

可调	从不
----	----

Monitor_start (string)

一种可选的回调方法：RGM 为启动此类资源的故障监视器所调用的程序的路径。

类别	有条件的/显式
----	---------

默认值	无默认值
-----	------

可调	从不
----	----

Monitor_stop (string)

设置 Monitor_start 后所需的回调方法：RGM 为停止此类资源的故障监视器所调用的程序的路径。

类别	有条件的/显式
----	---------

默认值	无默认值
-----	------

可调	从不
----	----

Pkglist (string_array)

资源类型安装中所包含的软件包的可选列表。

类别	有条件的/显式
----	---------

默认值	无默认值
-----	------

可调	从不
----	----

Postnet_stop (string)

一种可选的回调方法：调用此类资源所依赖的任意网络地址资源的 Stop 方法后，RGM 所调用的程序的路径。该方法预计用于执行在将网络接口配置为关闭之后必须执行的 Stop 操作。

类别	有条件的/显式
----	---------

默认值	无默认值
-----	------

可调	从不
----	----

Prenet_start (string)

一种可选的回调方法：调用此类资源所依赖的任意网络地址资源的 Start 方法之前，RGM 所调用的程序的路径。该方法预计用于执行在将网络接口配置为打开之前必须执行的 Start 操作。

类别	有条件的/显式
默认值	无默认值
可调	从不

Proxy (boolean)

指示此类资源是否为代理资源。

代理资源是一种 Oracle Solaris Cluster 资源，用于从其他群集框架（如 Oracle Clusterware）导入某个资源的状态。Oracle Clusterware 是一组适用于群集环境的不依赖于平台的系统服务。

代理资源类型使用 Prenet_start 方法启动用于监视外部（代理）资源状态的守护进程。Postnet_stop 方法停止正在进行监视的守护进程。正在进行监视的守护进程发出带有 CHANGE_STATE_ONLINE 或 CHANGE_STATE_OFFLINE 标记的 scha_control 命令，从而将代理资源的状态相应地设置为 Online 或 Offline。同样，scha_control () 函数也使用 SCHA_CHANGE_STATE_ONLINE 和 SCHA_CHANGE_STATE_OFFLINE 标记。

如果将该属性设置为 TRUE，资源将为代理资源。

类别	可选
默认值	FALSE
可调	从不

Resource_list (string_array)

该资源类型的所有资源的列表。管理员不直接设置该属性。相反，在管理员向任意资源组添加或从中删除此类资源时，RGM 会更新该属性。

类别	仅查询
默认值	空列表
可调	从不

Resource_type (string)

资源类型的名称 要查看当前已注册的资源类型的名称，请键入：

clresourcetype list

资源类型名称包括版本，这是强制性的：

vendor_id.resource_type:version

资源类型名称的三个组成部分是在 RTR 文件中指定为 *vendor-id*、*resource-type* 和 *RT-version* 的属性。clresourcetype 命令用于插入句点 (.) 和冒号 (:) 分界符。资源类型名称的 RT_version 后缀的值与 RT_version 属性的值相同。要确保 *vendor-id* 唯一，建议的方法是使用要创建资源类型的公司的股票代码。

类别	必需
默认值	空字符串
可调	从不

RT_basedir (string)

用于完成回调方法的相对路径的目录路径。此路径预计设置为资源类型软件包的安装位置。该路径必须是完整路径，即，必须以正斜杠 (/) 开头。如果所有方法路径名称是绝对的，则不需要该属性。

类别	必需的，除非所有方法路径名称是绝对的
默认值	无默认值
可调	从不

RT_description (string)

资源类型的简要描述。

类别	有条件的
默认值	空字符串
可调	从不

RT_system (boolean)

如果将某个资源类型的该属性设置为 TRUE，则无法删除该资源类型 (clresourcetype unregister *resource-type-name*)。该属性旨在帮助防止意外删除用于支持群集基础结构的资源类型，例如 LogicalHostname。但是，您可以将 RT_system 属性应用于任意资源类型。

要删除 RT_system 属性设置为 TRUE 的资源类型，必须先将该属性设置为 FALSE。删除其资源支持群集服务的资源类型时需谨慎。

类别	可选
默认值	FALSE
可调	随时

RT_version (string)

用于标识该资源类型实现的强制性版本字符串。RT_version 是完整资源类型名称的后缀部分。

类别	有条件的/显式或必需的
默认值	无默认值
可调	从不

Single_instance (boolean)

如果将该属性设置为 TRUE，RGM 将只允许在群集中存在一个此类资源。

类别	可选
默认值	FALSE
可调	从不

Start (string)

回调方法：RGM 为启动此类资源所调用的程序的路径。

类别	必需的，除非 RTR 文件声明 Prenet_start 方法
默认值	无默认值
可调	从不

Stop (string)

回调方法：RGM 为停止此类资源所调用的程序的路径。

类别	必需的，除非 RTR 文件声明 Postnet_stop 方法
默认值	无默认值
可调	从不

Update (string)

一种可选的回调方法：在运行的此类资源的属性发生更改时，RGM 所调用的程序的路径。

类别	有条件的/显式
默认值	无默认值
可调	从不

Validate (string)

一种可选的回调方法：RGM 为检查此类资源的属性值所调用的程序的路径。

类别 有条件的/显式

默认值 无默认值

可调 从不

Vendor_ID (string)

请参见 Resource_type 属性。

类别 有条件的

默认值 无默认值

可调 从不

[clresource\(1CL\) \[225\]](#)、[clresourcegroup\(1CL\) \[253\]](#)、[clresourcetype\(1CL\) \[277\]](#)、[rt_reg\(4\) \[1067\]](#)、[su](#)

[Unresolved link to " Oracle Solaris Cluster Concepts Guide "](#)、[Unresolved link to " Oracle Solaris Cluster Data Services Developer's Guide "](#)、[Unresolved link to " Oracle Solaris Cluster 数据服务规划和管理指南 "](#)

名称

scalable_service — 可伸缩资源类型

可伸缩数据服务是一种利用 Oracle Solaris Cluster 网络设备的服务。此类服务是作为资源类型来实现的，可由资源组管理器 (Resource Group Manager, RGM) 进行管理。

标准资源属性

标准资源属性 Scalable、Network_resources_used、Port_list、Load_balancing_policy 和 Load_balancing_weights 通用于所有可伸缩资源类型。有关这些属性的语法和说明，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

某些数据服务可在可伸缩或不可伸缩模式下运行。这些服务允许您在创建资源时为 Scalable 属性指定值 True 或 False。如果此属性在某个资源上设置为 True，则我们说该资源处于“可伸缩模式”。然后，该资源必须包含在可伸缩模式的资源组中，也就是说，必须包含在为其 Maximum primaries 属性设置的值大于 1 的组中。

对于只能在可伸缩模式下运行的数据服务来说，此类资源的 Scalable 属性为隐式 True，且无法由管理员进行更改。

您可以随时更改 Load_balancing_weights 和 Port_list 属性，即使资源正处于联机状态。创建资源时会设置 Network_resources_used 和 Load_balancing_policy，此后将无法编辑这些属性。这些属性可能已经有默认值，也可能需要在用户创建资源时由用户提供相关值，具体取决于资源类型的实现方式。

网络监视

在特定节点上运行的可伸缩服务实例需要能够通过公共网络回复客户机。RGM 会自动监视将运行可伸缩服务的节点上的公共网络的运行状况，如果无法从特定节点访问公共网络，则可能会关闭该节点上的可伸缩服务实例。如果通过使用 clresource unmonitor 命令禁用对可伸缩资源的监视，则会禁用这些网络检查。

资源验证

创建或更新设置为 True 的 Scalable 资源属性时，RGM 将验证各种资源属性，如果这些属性未正确配置，将拒绝用户尝试进行的更新。执行的检查如下：

- Network_resources_used 属性不能为空。该属性必须包含现有 SharedAddress 资源的名称。为包含可伸缩资源的资源组的 NodeList 属性指定的每个节点都必须出现在 SharedAddress 资源之一的 NetIfList 属性或 AuxNodeList 属性中。
- 包含可伸缩资源的资源组必须将其 RG_dependencies 属性设置为包含可伸缩资源的 Network_resources_used 属性中列出的所有 SharedAddress 资源的资源组。
- Port_list 属性不能为空。该属性必须包含端口和协议对的清单，其中的协议为 tcp、tcp6、udp 或 udp6。可指定的可能协议包括仅适用于 TCP IPv4 的 tcp、同时适用于 TCP IPv4 和 TCP IPv6 的 tcp6、仅适用于 UDP IPv4 的 udp 或同时适用于 UDP IPv4 和 UDP IPv6 的 udp6。

例如，可指定 `Port_list=80/tcp,40/udp` 。

关联

IP 关联可以保证将指定客户端 IP 地址发出的连接转发至相同群集节点。将 `Load_balancing_policy` 设置为 `Lb_sticky` 或 `Lb_sticky_wild` 时，`Affinity_timeout`、`UDP_affinity` 和 `Weak_affinity` 才能关联在一起。有关详细信息，请参见 [r_properties\(5\) \[1103\]](#)。

[clresource\(1CL\) \[225\]](#)、[clresourcegroup\(1CL\) \[253\]](#)、[clresourcetype\(1CL\) \[277\]](#)、[rt_callbacks\(1HA\) \[](#)

[Unresolved link to " Oracle Solaris Cluster 软件安装指南 "](#)、[Unresolved link to " Oracle Solaris Cluster Data Services Developer's Guide "](#)

名称

SUNW.ScalDeviceGroup, ScalDeviceGroup — 可伸缩设备组的资源类型实现

SUNW.ScalDeviceGroup 资源类型表示可伸缩设备组。该资源类型的实例表示以下设备组类型：

- Solaris Volume Manager for Sun Cluster 多属主磁盘集

SUNW.ScalDeviceGroup 资源类型是可伸缩资源类型。此资源类型的实例在该资源所属资源组的节点列表中的每个节点上联机。

要注册此资源类型并创建该资源类型的实例，请使用以下方法之一：

- Oracle Solaris Cluster Manager
- 使用 `clsetup` 实用程序，并指定用于配置 Oracle Solaris Cluster Support for Oracle Real Application Clusters 的选项
- 以下 Oracle Solaris Cluster 维护命令序列：
 1. 要注册此资源类型，请使用 `clresourcetype` 命令。
 2. 要创建此资源类型的实例，请使用 `clresource` 命令。

下面的子部分介绍了为 SUNW.ScalDeviceGroup 资源类型定义的标准属性和扩展属性。

标准属性

有关所有标准资源属性的描述，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

按如下所示覆盖此资源类型的标准资源属性：

Monitor_start_timeout

最小值

10

默认值

300

Monitor_stop_timeout

最小值

10

默认值

300

Postnet_stop_timeout

最小值
60

默认值
300

Prenet_start_timeout

最小值
60

默认值
300

Start_timeout

最小值
60

默认值
300

Stop_timeout

最小值
60

默认值
300

Thorough_probe_interval

默认值
300

Update_timeout

最小值
60

默认值
300

Validate_timeout

最小值
60

默认值
300

扩展属性

此资源类型的扩展属性如下所示：

Debug_level

该属性指定来自此类型资源的调试消息的记录级别。调试级别越高，写入日志文件的调试消息越多。

数据类型
整数

默认值
0

范围
0-10

可调
随时

DiskGroupName

该属性指定资源所表示的设备组的名称。必须将该属性设置为以下项目之一：

- 现有 Solaris Volume Manager for Sun Cluster 多属主磁盘集的名称。该名称是在创建磁盘集所用的 [Unresolved link to "metaset1M"](#) 命令中指定的。

所指定设备组的要求如下所示：

- 该设备组必须是有效的现有多属主磁盘集或共享磁盘组。
- 该设备组必须托管于可以控制资源的所有节点上。
- 该设备组必须可从可以控制可伸缩设备组资源的所有节点上访问。
- 该设备组必须至少包含一个卷。

数据类型
字符串

默认值
未定义默认值

范围
不适用

可调
禁用时

IOTimeout

该属性指定 I/O 探测的超时值（以秒为单位）。

默认值
30

范围
1–1800

可调
随时

LogicalDeviceList

该属性以逗号分隔方式指定资源的故障监视器要监视的逻辑卷的列表。此属性是可选的。如果没有为该属性指定值，设备组中的所有逻辑卷均会受到监视。

设备组的状态派生自受监视的各个逻辑卷的状态。如果所有受监视的逻辑卷都运行良好，则说明设备组正常运行。如果有任何受监视的逻辑卷发生故障，则说明设备组存在故障。

通过查询卷的卷管理器可获取各个逻辑卷的状态。如果无法通过查询确定 Solaris Volume Manager for Sun Cluster 卷的状态，故障监视器将执行文件输入/输出 (input/output, I/O) 操作来确定卷状态。

如果发现设备组发生故障，将停止对表示资源组的资源的监视并将该资源置于禁用状态。

注 - 对于镜像磁盘，即使某个子镜像发生故障，也仍认为设备组运行良好。

注 - 使用 Solaris Volume Manager for Sun Cluster 时，如果将指定监视软分区，则还应该指定相应的顶层或底层元设备。

所指定的各逻辑卷的要求如下所示：

- 必须存在逻辑卷。
- 逻辑卷必须包含在 diskgroupname 属性所指定的设备组中。
- 逻辑卷必须可从可以控制可伸缩设备组资源的所有节点上访问。

数据类型
字符串数组

默认值

""

范围

不适用

可调

随时

Monitor_retry_count

该属性指定进程监视器工具 (process monitor facility, PMF) 可以重新启动故障监视器的最大次数。

数据类型

整数

默认值

4

范围

未定义范围

可调

随时

Monitor_retry_interval

该属性指定 PMF 对故障监视器的重新启动进行计数的时间间隔（以分钟为单位）。

数据类型

整数

默认值

2

范围

未定义范围

可调

随时

RebootOnFailure

该属性指定在探测器检测到失败时是否重新引导本地系统。该属性设置为 TRUE 时，资源直接或间接使用的所有设备都必须受磁盘路径监视功能的监视。

如果 RebootOnFailure 设置为 TRUE，且至少有一个设备可用于

GlobalDevicePaths、FileSystemMountPoints 或 Zpools 属性中指定的每个实体，本

地系统将重新引导。本地系统表示资源处于联机状态的全局群集节点或区域群集节点。

默认值

FALSE

可调

随时

例 400 使用 SUNW.vucmm_svm 资源类型创建 ScalDeviceGroup 资源

本示例显示了如何使用 SUNW.vucmm_svm 资源类型创建一个 ScalDeviceGroup 资源来表示名为 datadg 的 Solaris Volume Manager for Sun Cluster 多属主磁盘集。该资源命名为 scaldatadg-rs。此示例假定存在以下 Oracle Solaris Cluster 对象：

- 名为 scaldatadg-rg 的可伸缩资源组。
- 名为 vucmm-svm-rs 的 SUNW.vucmm_svm 资源类型实例。

```
# clresourcetype register SUNW.ScalDeviceGroup
# clresource create -t SUNW.ScalDeviceGroup \
-g scaldatadg-rg \
-p Resource_dependencies=vucmm-svm-rs \
-p DiskGroupName=datadg \
scaldatadg-rs
```

有关以下属性的描述，请参见 [Unresolved link to " attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core

[clresource\(1CL\) \[225\]](#)、[clresourcetype\(1CL\) \[277\]](#)、[clsetup\(1CL\) \[407\]](#)、[Unresolved link to " metaset1M"](#)、[Unresolved link to " attributes5"](#)、[r_properties\(5\) \[1103\]](#)、[SUNW.vucmm_svm\(5\) \[1233\]](#)

[Unresolved link to " 适用于 Oracle Real Application Clusters 的 Oracle Solaris Cluster 数据服务指南 "](#)

名称

SUNW.ScalMountPoint, ScalMountPoint — 可伸缩文件系统挂载点的资源类型实现

SUNW.ScalMountPoint 资源类型表示可伸缩的文件系统挂载点。此资源类型的实例表示以下文件系统类型的挂载点：

- Sun QFS 共享文件系统
- 网络连接的存储 (network-attached storage, NAS) 设备上的文件系统。
NAS 设备和文件系统必须已配置为可用于 Oracle Solaris Cluster。有关更多信息，请参见[Unresolved link to " Oracle Solaris Cluster With Network-Attached Storage Device Manual "](#)。

SUNW.ScalMountPoint 资源类型是可伸缩资源类型。此资源类型的实例在该资源所属资源组的节点列表中的每个节点上联机。

要注册此资源类型并创建该资源类型的实例，请使用以下方法之一：

- Oracle Solaris Cluster Manager
- 使用 `clsetup` 实用程序，并指定用于配置 Oracle Solaris Cluster Support for Oracle Real Application Clusters 的选项
- 以下 Oracle Solaris Cluster 维护命令序列：
 1. 要注册此资源类型，请使用 `clresourcetype` 命令。
 2. 要创建此资源类型的实例，请使用 `clresource` 命令。

下面的子部分中介绍了为 SUNW.ScalMountPoint 资源类型定义的标准属性和扩展属性。

标准属性

有关所有标准资源属性的描述，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

按如下所示覆盖此资源类型的标准资源属性：

Monitor_start_timeout

最小值

10

默认值

300

Monitor_stop_timeout

最小值

10

默认值
300

Postnet_stop_timeout

最小值
60

默认值
300

Prenet_start_timeout

最小值
60

默认值
300

Start_timeout

最小值
60

默认值
300

Stop_timeout

最小值
60

默认值
300

Thorough_probe_interval

默认值
300

Update_timeout

最小值
60

默认值
300

Validate_timeout

最小值
60

默认值
300

扩展属性

此资源类型的扩展属性如下所示：

Debug_level

该属性指定来自文件系统挂载点资源的调试消息的记录级别。调试级别越高，写入日志文件的调试消息越多。

数据类型
整数

默认值
0

范围
0–10

可调
随时

FileSystemType

该属性指定资源表示的挂载点对应的文件系统类型。您必须指定此属性。将该属性设置为以下某一值：

nas
指定文件系统为 NAS 设备上的文件系统。

s-qfs
指定文件系统为 Sun QFS 共享文件系统。

数据类型
字符串

默认值
未定义默认值

范围
不适用

可调

禁用时

IOTimeout

该属性指定故障监视器探测文件输入/输出 (input/output, I/O) 所使用的超时值 (以秒为单位)。为了确定已挂载文件系统是否可用, 故障监视器会对文件系统上的测试文件执行 I/O 操作, 如打开、读取及写入操作。如果在超时时限内未完成 I/O 操作, 故障监视器将报告错误。

数据类型

整数

默认值

300

范围

5-1800

可调

随时

Monitor_retry_count

该属性指定进程监视器工具 (process monitor facility, PMF) 可以重新启动故障监视器的最大次数。

数据类型

整数

默认值

4

范围

未定义范围

可调

随时

Monitor_retry_interval

该属性指定 PMF 对故障监视器的重新启动进行计数的时间间隔 (以分钟为单位)。

数据类型

整数

默认值

2

范围

未定义范围

可调

随时

MountOptions

该属性指定在挂载资源所表示的文件系统时要使用的挂载选项的列表（以逗号分隔）。此属性是可选的。如果没有为该属性指定值，将从文件系统的默认值表中获取挂载选项。

- 对于 Sun QFS 共享文件系统，这些选项从 `/etc/opt/SUNWsamfs/samfs.cmd` 文件获取。
- 对于 NAS 设备上的文件系统，这些选项从 `/etc/vfstab` 文件获取。

通过该属性指定的挂载选项将覆盖文件系统的默认值表中的挂载选项。

数据类型

字符串

默认值

""

范围

不适用

可调

禁用时

MountPointDir

该属性指定资源所表示的文件系统的挂载点。挂载点是当挂载文件系统时此文件系统所附加到的文件系统分层结构的目录的完整路径。您必须指定此属性。

您所指定的目录必须已存在。

数据类型

字符串

默认值

未定义默认值

范围

不适用

可调

禁用时

RebootOnFailure

该属性指定在探测器检测到失败时是否重新引导本地系统。该属性设置为 TRUE 时，资源直接或间接使用的所有设备都必须受磁盘路径监视功能的监视。

如果 RebootOnFailure 设置为 TRUE，且至少有一个设备可用于 GlobalDevicePaths、FileSystemMountPoints 或 Zpools 属性中指定的每个实体，则本地系统将重新引导。本地系统表示资源处于联机状态的全局群集节点或区域群集节点。

默认值

FALSE

可调

随时

TargetFileSystem

该属性指定要在 MountPointDir 扩展属性所指定的挂载点上挂载的文件系统。您必须指定此属性。文件系统的类型必须与 FileSystemType 属性所指定的类型相匹配。该属性的格式取决于文件系统的类型，如下所示：

- 对于 Sun QFS 共享文件系统，将该属性设置为创建文件系统时为该文件系统指定的名称。该文件系统必须已正确配置。有关更多信息，请参见您的 Sun QFS 共享文件系统文档。
- 对于 NAS 设备上的文件系统，将该属性设置为 *nas-device:path*。该格式中的可替换项如下所示：

nas-device

指定正在导出文件系统的 NAS 设备的名称。（可选）您可以使用域对该名称进行限定。

path

指定 NAS 设备正在导出的文件系统的完整路径。

NAS 设备和文件系统必须已配置为可用于 Oracle Solaris Cluster。有关更多信息，请参见 [Unresolved link to " Oracle Solaris Cluster With Network-Attached Storage Device Manual "](#)。

数据类型

字符串

默认值

未定义默认值

范围

不适用

可调
禁用时

例 401 创建 ScalMountPoint 资源

本示例显示如何创建 ScalMountPoint 资源来表示用于 Solaris Volume Manager for Sun Cluster 的 Sun QFS 共享文件系统的挂载点。该资源命名为 scal-db_qfs-Data-rs。该文件系统的特性如下所述：

- 文件系统的挂载点是 /db_qfs/Data。
- 将挂载的文件系统是 Data。
- 挂载选项从该文件系统的默认值表获取，即 /etc/opt/SUNWsamfs/samfs.cmd 文件。

此示例假定存在以下 Oracle Solaris Cluster 对象：

- 名为 scaldatadg-rg 的可伸缩资源组。
- 名为 qfs-db_qfs-Data-rs 的 SUNW.qfs 资源类型实例。
- 名为 scaldatadg-rs 的 SUNW.ScalDeviceGroup 资源类型实例。

```
# clresourcetype register SUNW.ScalMountPoint
# clresource create -t SUNW.ScalMountPoint \
-g scaldatadg-rg \
-p Resource_dependencies=qfs-db_qfs-Data-rs,scaldatadg-rs \
-p MountPointDir=/db_qfs/Data \
-p FileSystemType=s-qfs \
-p TargetFileSystem=Data \
scal-db_qfs-Data-rs
```

有关以下属性的描述，请参见 [Unresolved link to " attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core

[clresource\(1CL\) \[225\]](#)、[clresourcetype\(1CL\) \[277\]](#)、[clsetup\(1CL\) \[407\]](#)、[Unresolved link to " vfstab4"](#)、[Unresolved link to " attributes5"](#)、[r_properties\(5\) \[1103\]](#)、[SUNWct.ScalDeviceGroup\(5\) \[1215\]](#)、[SUNW.vucmm_framework\(5\) \[1231\]](#)

[Unresolved link to " 适用于 Oracle Real Application Clusters 的 Oracle Solaris Cluster 数据服务指南 "](#)、[Unresolved link to " Oracle Solaris Cluster With Network-Attached Storage Device Manual "](#)

名称

SUNW.SCTelemetry, SCTelemetry — 用于收集系统资源使用情况相关数据的资源类型

SUNW.SCTelemetry 是允许您收集系统资源使用情况相关数据的资源类型。SUNW.SCTelemetry 将系统资源使用情况数据在 Java DB 数据库中存储七天。SUNW.SCTelemetry 类型的资源对 SUNW.derby 类型的资源具有依赖性。有关更多信息，请参见 [SUNW.derby\(5\) \[1175\]](#) 手册页。

与 SUNW.SCTelemetry 资源类型关联的扩展属性如下所述：

Extended_accounting_cleanup (布尔值)

指定是否清理扩展记帐日志文件，即是否删除历史数据。Extended_accounting_cleanup 的可能值是 TRUE 和 FALSE。

类别	可选
默认值	TRUE
可调	随时

Monitor_retry_count (整数)

控制故障监视器的重新启动。该属性指定进程监视器工具重新启动故障监视器的次数。该属性对应于传递给 [pmfadm\(1M\) \[625\]](#) 命令的 -n 选项。资源组管理器 (Resource Group Manager, RGM) 在指定时间段内对重新启动次数进行计数。有关更多信息，请参见 Monitor_retry_interval 属性。请注意，Monitor_retry_count 指的是故障监视器自身而非 SUNW.SCTelemetry 类型的资源的重新启动。

类别	可选
默认值	4
可调	随时

Monitor_retry_interval (整数)

以分钟为单位指定时间段，在此期间 RGM 对故障监视器的失败进行计数。该属性对应于传递给 [pmfadm\(1M\) \[625\]](#) 命令的 -t 选项。如果故障监视器的失败次数超出了 Monitor_retry_count 属性的值，则进程监视器工具不会重新启动故障监视器。

类别	可选
默认值	2 分钟
可调	随时

Probe_timeout (整数)

指定探测的超时值 (以秒为单位)。

类别	可选
默认值	60 秒
最小值	2 秒
可调	随时

Sampling_interval (整数)

指定收集监视数据的频率。Telemetry_sampling_interval 属性的值必须介于 30 和 3600 之间。

类别	必需
默认值	60
最小值	30 秒
最大值	3600 秒
可调	随时

[pmfadm\(1M\) \[625\]](#)、[SUNW.derby\(5\) \[1175\]](#)

名称

SUNW.crs_framework, crs_framework — 协调 Oracle Clusterware 的关闭的资源类型实现

SUNW.crs_framework 资源类型可在 Oracle Solaris Cluster Support for Oracle Real Application Clusters (Oracle RAC) 配置中协调 Oracle Clusterware 和 Oracle Solaris Cluster 资源的关闭。该资源类型通过使 Oracle Solaris Cluster 停止 Oracle Clusterware 来使 Oracle Solaris Cluster 和 Oracle Clusterware 互操作。

注 - 该资源类型不允许使用 Oracle Solaris Cluster 管理命令启动 Oracle Clusterware。只能通过使用 Oracle 命令或引导节点来启动 Oracle Clusterware。

Oracle Clusterware 表决磁盘和 Oracle 群集注册表 (Oracle cluster registry, OCR) 文件可能位于 SUNW.ScalDeviceGroup 或 SUNW.ScalMountPoint 类型的资源表示的存储中。在这种情况下，在使 SUNW.ScalDeviceGroup 或 SUNW.ScalMountPoint 类型的资源脱机之前必须先停止 Oracle Clusterware。SUNW.crs_framework 类型的资源可通过在以下情况下在某个节点上停止 Oracle Clusterware 进程确保满足此要求。

- 当 SUNW.ScalDeviceGroup 或 SUNW.ScalMountPoint 类型的资源在节点上脱机时。出于以下原因，必须停止 Oracle Clusterware 进程：
 - 为了确保正确停止 SUNW.ScalDeviceGroup 或 SUNW.ScalMountPoint 类型的资源
 - 为了避免因在 Oracle Clusterware 或 Oracle RAC 进程访问存储期间使 SUNW.ScalDeviceGroup 或 SUNW.ScalMountPoint 类型的资源脱机而发生数据库或节点故障
- 当某个节点关闭时。如果未停止 Oracle Clusterware 进程，该节点将无法关闭。

SUNW.crs_framework 资源类型是单实例资源类型。在群集中只能创建一个此类型的资源。

为确保 Oracle Solaris Cluster 按正确顺序停止资源，请按如下所述配置一个类型为 SUNW.crs_framework 的资源：

- 确保包含 SUNW.ScalDeviceGroup 或 SUNW.ScalMountPoint 类型的资源的任何资源组为将包含 SUNW.crs_framework 资源的资源组声明强正关联性。
- 设置 SUNW.crs_framework 资源对表示 Oracle Clusterware 表决磁盘和 OCR 文件存储的任何资源的脱机重新启动依赖性。这些资源的类型为 SUNW.ScalDeviceGroup 或 SUNW.ScalMountPoint。将每个依赖性的范围限制为正在运行 SUNW.ScalDeviceGroup 资源或 SUNW.ScalMountPoint 资源的节点。
- 设置 SUNW.crs_framework 类型的资源对 SUNW.rac_framework 类型的资源的强依赖性。

应在为 Oracle Solaris Cluster Support for Oracle RAC 数据服务配置数据库资源时创建这些依赖性和关联性。有关更多信息，请参见[Unresolved link to "适用于 Oracle](#)

Real Application Clusters 的 Oracle Solaris Cluster 数据服务指南 中的配置 [Support for Oracle RAC 数据库实例的资源](#)"。

要注册此资源类型并创建该资源类型的实例，请使用以下方法之一：

- Oracle Solaris Cluster Manager
- 使用 [clsetup\(1CL\) \[407\]](#) 实用程序，并指定用于配置 Oracle Solaris Cluster Support for Oracle Real Application Clusters 的选项
- 以下 Oracle Solaris Cluster 维护命令序列：
 1. 要注册此资源类型，请使用 [clresourcetype\(1CL\) \[277\]](#) 命令。
 2. 要创建此资源类型的实例，请使用 [clresource\(1CL\) \[225\]](#) 命令。

标准属性

有关所有标准资源属性的描述，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

按如下所示覆盖此资源类型的标准资源属性：

Monitor_start_timeout

最小值	60
默认值	300

Monitor_stop_timeout

最小值	60
默认值	300

Start_timeout

最小值	60
默认值	300

Stop_timeout

最小值	60
默认值	1200

Update_timeout

最小值	60
默认值	300

Validate_timeout

最小值 60

默认值 300

扩展属性

SUNW.crs_framework 资源类型没有扩展属性。

例 402 创建 SUNW.crs_framework 资源

此示例注册 SUNW.crs_framework 资源类型并创建 SUNW.crs_framework 资源类型的名为 crs_framework-rs 的实例。本示例做出了以下假定：

- 使用 C shell。
- 存在一个名为 crs-framework-rg 的资源组。
- 存在以下资源：
 - 类型为 SUNW.rac_framework 的名为 rac_framework-rs 的资源，表示 Oracle RAC 框架
 - 类型为 SUNW.ScalDeviceGroup 的名为 db-storage-rs 的资源，表示存储 Oracle Clusterware 表决磁盘和 OCR 文件的可伸缩设备组

```
phys-schost-1# clresourcetype register SUNW.crs_framework
```

```
phys-schost-1# clresource create -g crs-framework-rg \  
-t SUNW.crs_framework \  
-p resource_dependencies=rac_framework-rs \  
-p resource_dependencies_offline_restart=db-storage-rs\{local_node\} \  
crs_framework-rs
```

有关以下属性的描述，请参见 [Unresolved link to " attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/library/ucmm

[clresource\(1CL\) \[225\]](#)、[clresourcetype\(1CL\) \[277\]](#)、[clsetup\(1CL\) \[407\]](#)、[Unresolved link to " attributes5"](#)

[SUNW.rac_framework\(5\) \[1213\]](#)、[SUNWct.ScalDeviceGroup\(5\) \[1215\]](#)、[SUNW.ScalMountPoint\(5\) \[1216\]](#)

[Unresolved link to " 适用于 Oracle Real Application Clusters 的 Oracle Solaris Cluster 数据服务指南 "](#)

名称

SUNW.derby, derby — Java DB 数据库的资源类型实现

SUNW.derby 是故障转移资源类型，允许您结合使用 Java DB 数据库与 Oracle Solaris Cluster。Java DB 数据库以 Derby 数据库为基础。有关数据库的信息，请参见 <http://db.apache.org/derby/>。

与 SUNW.derby 资源类型关联的扩展属性如下所示：

DB_path (字符串)

为 Java DB 数据库指定数据文件的位置。

DB_path 的值是指定路径的字符串。指定的路径应该是由所选存储（例如 HAStoragePlus）控制的路径。

类别	必需
可调	禁用时

DB_port (整数)

指定 Java DB 数据库的端口。

类别	必需
默认值	1527
可调	禁用时

DB_probe_port (整数)

指定 Oracle Solaris Cluster 在测试 Java DB 数据库服务器的运行状况时使用的端口。

类别	必需
默认值	1528
可调	禁用时

Monitor_retry_count (整数)

控制故障监视器的重新启动。该属性指定进程监视器工具重新启动故障监视器的次数。该属性对应于传递给 `pmfadm(1M)` [625] 命令的 `-n` 选项。资源组管理器 (Resource Group Manager, RGM) 在指定时间段（请参见 `Monitor_retry_interval`）中对重新启动次数进行计数。请注

意，Monitor_retry_count 表示故障监视器本身的重新启动，不是 SUNW.derby 资源的重新启动。

类别	可选
默认值	4
可调	随时

Monitor_retry_interval (整数)

以分钟为单位指定时间段，在此期间 RGM 对故障监视器的失败进行计数。该属性对应于传递给 [pmfadm\(1M\) \[625\]](#) 命令的 -t 选项。如果故障监视器的失败次数超出了扩展属性 Monitor_retry_count 的值，则进程监视器工具不会重新启动故障监视器。

类别	可选
默认值	2 分钟
可调	随时

Probe_timeout (整数)

指定探测命令的超时值（以秒为单位）。

类别	可选
默认值	120 秒
默认值	2 秒
可调	随时

[pmfadm\(1M\) \[625\]](#)

名称

SUNW.Event — 群集重新配置通知协议 (Cluster Reconfiguration Notification Protocol, CRNP) 的资源类型实现

SUNW.Event 资源类型实现在 Oracle Solaris Cluster 中提供可用性高的 CRNP 服务。此实现通过将通知守护进程 (/usr/cluster/lib/sc/cl_apid) 作为资源在 Oracle Solaris Cluster 资源组管理器 (resource group manager, RGM) 中进行管理, 使该守护进程具有高可用性。对于包含 SUNW.Event 资源的资源组, 必须在该资源组中配置了网络资源。在一个群集中只能存在一个类型为 SUNW.Event 的资源。

只能在全局区域中运行 CRNP。

标准属性

此部分介绍用来控制实现行为的关键标准属性。可使用 `clresource` 命令为 SUNW.Event 资源设置这些属性。[r_properties\(5\) \[1103\]](#) 手册页更详细地介绍了这些资源属性。

`Network_resources_used (string_array)`

该资源依赖的逻辑主机名或共享地址网络资源的列表。该列表包含在属性 `Resource_dependencies`、`Resource_dependencies_weak`、`Resource_dependencies_restart` 或 `Resource_dependencies_offline_restart` 中出现的所有网络地址资源。

该属性由 RGM 基于资源依赖性属性的设置自动更新。不能直接设置此属性。可使用 `Resource_dependencies` 属性替代。

类别	有条件的/可选
----	---------

默认值	空列表
-----	-----

可调	禁用时
----	-----

`Port_list (string_array)`

服务器侦听的端口号的逗号分隔列表。[r_properties\(5\) \[1103\]](#) 手册页更详细地介绍了 `Port_list`。

类别	有条件的/必需
----	---------

默认值	9444/tcp
-----	----------

可调	禁用时
----	-----

`Resource_dependencies (string array)`

某个资源依赖的资源的列表。此列表包含该资源使用的所有逻辑主机名或共享地址网络资源。此属性的默认值为 `null`。如果应用程序需要绑定到一个或多个特定地

址，则必须指定此属性。如果未指定网络资源依赖性，则应用程序将侦听所有地址。

创建事件资源之前，必须已经配置了 LogicalHostname 或 SharedAddress 资源。可以指定一个或多个资源名。每个网络资源可以包含一个或多个逻辑主机名。有关更多信息，请参见 [clreslogicalhostname\(1CL\) \[207\]](#) 和 [clressharedaddress\(1CL\) \[289\]](#) 手册页。

可以使用 Resource_dependencies_weak、Resource_dependencies_restart 或 Resource_dependencies_offline_restart 属性而不是 Resource_dependencies 属性来指定其他依赖性类别。有关更多信息，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

类别	可选
默认值	空列表
可调	随时

Retry_count (integer)

监视器在资源失败时尝试重新启动该资源的次数。 [r_properties\(5\) \[1103\]](#) 手册页更详细地介绍了 Retry_count。

注 - 如果为该属性指定一个负值，则监视器将尝试重新启动资源无限次。

类别	有条件的
默认值	2
最大值	10
可调	随时

Retry_interval (integer)

对失败资源的重新启动尝试进行计数时所基于的秒数。 [r_properties\(5\) \[1103\]](#) 手册页更详细地介绍了 Retry_interval。

类别	有条件的
默认值	300
最大值	3600
可调	随时

Thorough_probe_interval (integer)

对资源进行高开销故障探测的调用之间间隔的秒数。 [r_properties\(5\) \[1103\]](#) 手册页更详细地介绍了 Thorough_probe_interval。

类别	有条件的
默认值	60
最大值	3600
可调	随时

扩展属性

此部分介绍用于控制实现行为的关键扩展属性。

Allow_hosts (string_array)

该属性控制可以向实现注册以接收群集重新配置事件的客户机集。此属性的一般形式是 `ipaddress/masklength`，用于定义哪个子网中的客户机可以注册接收事件。例如，`129.99.77.0/24` 设置允许 `129.99.77` 子网中的客户机注册接收事件。另一个示例是，`192.9.84.231/32` 仅允许客户机 `192.9.84.231` 注册接收事件。

此外，还能够识别以下特殊关键字：

- LOCAL 表示位于与群集直接连接的子网中的所有客户机。
- ALL 允许所有客户机注册。

注 - 如果某个客户机既匹配 Allow_hosts 属性中的条目，也匹配 Deny_hosts 属性中的条目，将禁止该客户机向实现注册。

类别	可选
默认值	LOCAL
可调	随时

Client_retry_count (integer)

该属性控制实现在与外部客户机通信时的尝试次数。如果客户机未能在 `Client_retry_count` 次尝试后响应，该客户机将超时。该客户机随后将从有资格接收群集重新配置事件的已注册客户机列表中删除。要再次开始接收事件，该客户机必须重新注册。关于 `Client_retry_interval` 属性的部分说明了实现进行这些重试的频率。

类别	可选
默认值	3
最小值	1
可调	随时

`Client_retry_interval` (integer)

该属性定义实现在与无响应外部客户机通信时的重试时间间隔（秒）。在此间隔中，最多尝试联系客户机 `Client_retry_count` 次。

可以随时修改此属性的值。

类别	可选
默认值	1800
最小值	30
可调	随时

`Client_timeout` (integer)

该属性定义实现在与外部客户机通信时使用的超时值（秒）。但是，实现会继续尝试联系客户机若干次（次数可调）。关于 `Client_retry_count` 和 `Client_retry_interval` 属性的部分介绍了调整此属性的方法。

类别	可选
默认值	60
最小值	30
可调	随时

`Deny_hosts` (string_array)

该属性控制被禁止注册接收群集重新配置事件的客户机集。确定访问权限时，此属性的设置优先于 `Allow_hosts` 列表中的设置。该属性的格式与 `Allow_hosts` 中定义的格式相同。

类别	可选
默认值	NULL
可调	随时

`Max_clients` (integer)

该属性控制可以向实现注册接收群集事件通知的最大客户机数。实现将会拒绝更多客户机的注册事件尝试。由于每个客户机注册会占用群集中的资源，因此通过调整此属性，用户可以控制外部客户机对群集中的资源用法。

类别	可选
默认值	1000

最小值	1
可调	禁用时

例 403 使用默认属性创建 SUNW.Event 资源

此示例说明如何在名为 events-rg 的现有资源组中创建一个名为 CRNP 的 SUNW.Event 故障转移资源。events-rg 资源组包含用于标识与资源组关联的故障转移主机名的 LogicalHostname 或 SharedAddress 资源。

```
# clresourcetype register SUNW.Event
# clresource create -g events-rg -t SUNW.Event CRNP
```

在本例中，创建的 SUNW.Event 资源命名为 CRNP。该资源侦听端口 9444，并允许直接连接的子网中的所有客户机注册接收事件。

例 404 使用非默认属性创建 SUNW.Event 资源

此示例说明如何在名为 events-rg 的资源组中创建名为 CRNP 的 SUNW.Event 资源。CRNP 资源配置为侦听端口 7000 及特定网络资源 foo-1，该网络资源已经在 events-rg 资源组中配置。此 CRNP 资源允许 192.9.77.0 子网中的客户机以及直接连接的子网中的客户机注册，但是不允许 192.9.77.98 客户机使用实现。

```
# clresource create -g events-rg -t SUNW.Event \
-p Port_list=7000/tcp -p Network_resources_used=foo-1 \
-p Allow_hosts=LOCAL,192.9.77.0/24 \
-p Deny_hosts=192.9.77.98/32 CRNP
```

/usr/cluster/lib/sc/cl_apid

CRNP 守护进程。

/usr/cluster/lib/sc/events/dtds

包含 CRNP 协议的数据类型定义的目录。

有关下列属性的描述，请参见 [Unresolved link to "attributes5"](#)。

属性类型	属性值
可用性	ha-cluster/system/core

[clresource\(1CL\) \[225\]](#)、[clresourcegroup\(1CL\) \[253\]](#)、[clresourcetype\(1CL\) \[277\]](#)、[scha_resource_get\(1HA link to "attributes5"](#)、[r_properties\(5\) \[1103\]](#)

名称

SUNW.gds — 使简单的网络感知和非网络感知应用程序具有高可用性或可伸缩性的资源类型

通用数据服务 (Generic Data Service, GDS) 机制允许您通过将简单的网络感知和非网络感知应用程序插入到 Oracle Solaris Cluster 资源组管理器 (Resource Group Manager, RGM) 框架中，使这些应用程序具有高可用性或可伸缩性。

GDS 包含一个全功能的 Oracle Solaris Cluster 资源类型，通过回调方法 ([rt_callbacks\(1HA\) \[545\]](#)) 和资源类型注册 (Resource Type Registration, RTR) 文件 ([rt_reg\(4\) \[1067\]](#)) 完成。

标准属性

Failover_mode (enum)

修改资源无法成功启动或停止时或者资源监视器发现运行状态不佳的资源并进而请求重新启动或故障转移时 RGM 执行的恢复操作。

有关 Failover_mode 属性的更多信息，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

类别	可选
默认值	SOFT
可调	随时

Network_resources_used (string array)

该资源依赖的逻辑主机名或共享地址网络资源的列表。该列表包含在属性 Resource_dependencies、Resource_dependencies_weak、Resource_dependencies_restart 或 Resource_dependencies_offline_restart 中出现的所有网络地址资源。

该属性由 RGM 基于资源依赖性属性的设置自动更新。不能直接设置此属性。可使用 Resource_dependencies 属性替代。

类别	有条件的/可选
默认值	空列表
可调	禁用时

Port_list (string array)

指定应用程序侦听的端口号的逗号分隔列表。请参见 [r_properties\(5\) \[1103\]](#) 手册页。

必须在代理生成器创建的启动脚本中指定 Port_list 属性，或者，如果使用 Oracle Solaris Cluster 管理命令，使用 clresource 命令指定该属性。

注 - 如果 `Network_aware` 属性设置为 `False`，则不需要 `Port_list` 属性。

附加到每个端口号的是一个斜杠 (/)，后跟该端口正在使用的协议，例如 `Port_list=80/tcp` 或 `Port_list=80/tcp6,40/udp6`。

可以指定以下协议值：

- `tcp`，表示 TCP IPv4
- `tcp6`，表示 TCP IPv6
- `udp`，表示 UDP IPv4
- `udp6`，表示 UDP IPv6

类别	有条件的/必需
默认值	无默认值
可调	随时

`Resource_dependencies` (string array)

指定某个资源依赖的资源的列表。此列表包含该资源使用的所有逻辑主机名或共享地址网络资源。此属性的默认值为 `null`。如果 `Network_aware` 属性设置为真，您必须将该属性设置为应用程序正在侦听的逻辑主机名或共享地址网络资源。

在创建 GDS 资源之前，必须已经配置了 `LogicalHostname` 或 `SharedAddress` 资源。

可以指定一个或多个资源名。每个网络资源可以包含一个或多个逻辑主机名。有关更多信息，请参见 [clreslogicalhostname\(1CL\) \[207\]](#) 和 [clressharedaddress\(1CL\) \[289\]](#) 手册页。

可以使用 `Resource_dependencies_weak`、`Resource_dependencies_restart` 或 `Resource_dependencies_offline_restart` 属性而不是 `Resource_dependencies` 属性来指定其他依赖性类别。有关更多信息，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

类别	可选
默认值	空列表
可调	随时

`Retry_count` (integer)

监视器在启动资源失败时尝试重新启动该资源的次数。

有关 `Retry_count` 属性的更多信息，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

类别	有条件的
默认值	2

可调	随时
----	----

Retry_interval (integer)

对尝试重新启动失败的资源进行计数的秒数。

有关 Retry_interval 属性的更多信息，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

类别	有条件的
----	------

默认值	370 秒
-----	-------

可调	随时
----	----

Scalable (boolean)

指示资源是否可伸缩，即，资源是否使用 Oracle Solaris Cluster 软件的网络负载平衡功能。

如果 Scalable 属性设置为 TRUE，将使用其他属性（例如 Load_balancing_policy 和 Load_balancing_weights）来配置负载平衡行为。

有关 Scalable、Load_balancing_policy 和 Load_balancing_weights 属性的更多信息，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

类别	可选
----	----

默认值	FALSE
-----	-------

可调	创建时
----	-----

Start_timeout (integer)

指定启动命令的超时值（秒）。

类别	可选
----	----

最小值	60 秒
-----	------

默认值	300 秒
-----	-------

可调	随时
----	----

Stop_timeout (integer)

指定停止命令的超时值（秒）。

类别	可选
----	----

最小值	60 秒
-----	------

默认值	300 秒
-----	-------

可调	随时
----	----

Validate_timeout (integer)

指定验证命令的超时值（秒）。

类别	可选
----	----

最小值	60 秒
-----	------

默认值	300 秒
-----	-------

可调	随时
----	----

扩展属性

Child_mon_level (integer)

提供对通过进程监视设备 (Process Monitor Facility, PMF) 监视的进程的控制。该属性表示分支子进程受监视的级别。省略此属性或将该属性设置为默认值与省略 [pmfadm\(1M\) \[625\]](#) 的 -c 选项的结果相同：监视所有子进程（及其后代）。

类别	可选
----	----

默认值	-1
-----	----

可调	创建时
----	-----

Failover_enabled (boolean)

允许资源故障转移。如果将该属性设置为 False，将禁止资源进行故障转移。可以使用该属性阻止应用程序资源启动资源组的故障转移。

类别	可选
----	----

默认值	True
-----	------

可调	禁用时
----	-----

注 - Failover_mode=RESTART_ONLY 设置的行为与 Failover_enabled=False 设置的行为一致。Failover_mode=LOG_ONLY 设置可进一步阻止资源重新启动。要更好地控制故障转移行为，可使用 Failover_mode 属性而不是 Failover_enabled 扩展属性。有关更多信息，请参见 [r_properties\(5\) \[1103\]](#) 中关于 Failover_mode 的 LOG_ONLY 和 RESTART_ONLY 值的描述。

Log_level (enum)

指定由 GDS 记录的诊断消息的级别或类型。可以为该属性指定 None、Info 或 Err。当指定 None 时，GDS 不会记录诊断消息。当指定 Info 时，会记录信息和错误消息。当指定 Err 时，仅会记录错误消息。

类别	可选
默认值	Info
可调	随时

Monitor_retry_count

进程监视器工具 (process monitor facility, PMF) 在 Monitor_retry_interval 属性指定的时间窗口中重新启动故障监视器的次数。该属性表示重新启动故障监视器本身而非资源。系统定义的属性 Retry_interval 和 Retry_count 控制资源的重新启动。

类别	可选
数据类型	整数
默认值	4
范围	0 - 2147483647 -1 表示无限的重试尝试次数。
可调	随时

Monitor_retry_interval

以分钟为单位的时间，在此时间段内对故障监视器的失败进行计数。如果在该时间段内故障监视器的失败次数超出了扩展属性 Monitor_retry_count 指定的值，则 PMF 将不会重新启动故障监视器。

类别	可选
数据类型	整数
默认值	2
范围	0 - 2147483647 -1 表示无限的重试间隔。
可调	随时

Network_aware (boolean)

该属性指定应用程序是否使用网络。

类别	可选
默认值	True
可调	创建时

Probe_command (string)

指定定期检查网络感知或非网络感知应用程序运行状况的命令。该命令必须是完整的命令行，以便可以直接传递给 Shell 来探测应用程序。如果应用程序运行正确，该探测命令将以退出状态 0 返回。

探测命令的退出状态用于确定应用程序故障的严重程度。该退出状态称为探测状态，是一个介于 0（表示成功）与 100（表示完全失败）的整数值。探测状态还可以是 201，该状态会使应用程序进行故障转移，除非 Failover_enabled 设置为 False。

GDS 探测算法使用探测状态来确定是在本地重新启动应用程序，还是将应用程序故障转移到其他节点。如果省略 probe 命令，GDS 将提供其自己的简单探测来连接网络资源中的应用程序。如果连接成功，GDS 将立即断开连接。如果连接和断开连接均成功，则应用程序被认定为正常运行。

GDS 不为非网络感知应用程序提供“默认”探测行为。但是，非网络感知应用程序在 PMF 下启动，PMF 可监视该应用程序并在其失败时重新启动它。pmfadm(1M) [625] 手册页中提供了更多信息。

类别	可选
默认值	Null
可调	禁用时

Probe_timeout (integer)

指定探测命令的超时值（秒）。

类别	可选
最小值	2 秒
默认值	30 秒
可调	随时

Start_command (string)

指定启动应用程序的命令。该命令必须是完整的命令行，以便可以直接传递给 Shell 来启动应用程序。

启动命令或其分支子进程之一应该是长期运行的程序或守护进程，用于将服务实际提供给客户机。启动命令进程树由进程监视设备 (Process Monitor Facility, PMF) 监视，如 Child_mon_level 扩展属性部分中所述。如果受监视的进程退出，它们将根据 Retry_count 和 Retry_interval 资源属性的设置重新启动。如果超过了重试计数，系统会尝试将资源组重定位到其他节点。

启动命令或其子进程返回的退出状态将被忽略。

类别	必需
----	----

最小值	1
默认值	无默认值
可调	禁用时

Stop_command (string)

指定停止应用程序的命令。该命令必须是完整的命令行，以便可以直接传递给 Shell 来停止应用程序。如果省略该属性或者停止命令返回非零值，GDS 将使用信号停止应用程序。

类别	可选
默认值	Null
可调	禁用时

Stop_signal (integer)

指定用于停止应用程序的信号。该属性的值与 [Unresolved link to "signal3HEAD"](#) 中定义的值相同。

类别	可选
最小值	1
最大值	37
默认值	15
可调	禁用时

Validate_command (string)

指定用于验证应用程序的绝对路径。如果未提供绝对路径，则不会验证应用程序。

验证命令的退出状态用于确定是否应允许创建或更新 GDS 资源。在创建或更新资源之前，指定的验证命令将在该资源所属资源组的节点列表中的每个节点上执行。如果验证命令返回非零值，则不允许创建或更新请求的资源。验证命令写入到 stdout 或 stderr 的任何输出将传递回发出管理命令来创建或更新该资源的用户。可根据这些输出解释资源验证失败的原因。

验证命令还会在使 gds 资源联机时、执行 Start_command 扩展属性之前执行。如果验证命令返回非零值，系统会将其视为启动失败。

在执行 scha_control 命令的 GIVEOVER 选项来将资源组重定位到新节点之前也会执行验证命令。如果该命令以非零状态退出，则会阻止移交，该资源组仍在其当前节点上受管理。

类别	可选
默认值	Null
可调	禁用时

以下示例说明如何使用 GDS 使名为 app 的应用程序具有高可用性。也可以使用 Oracle Solaris Cluster 代理生成器 ([scdsbuilder\(1HA\) \[553\]](#)) 创建包含这些命令的脚本。

基本示例

此示例说明如何注册 SUNW.gds 资源类型、如何为应用程序创建资源组、为逻辑主机名 hhead 创建 LogicalHostname 资源、创建网络感知应用程序资源、管理资源组、启用其所有资源，以及如何使其资源联机。

此时，应用程序正在运行，具有高可用性，并且由 GDS 提供的简单探测监视。现在可以检查该应用程序的状态。

```
# clresourcetype register SUNW.gds
# clresourcegroup create rg1
# clreslogicalhostname create -g rg1 -h hhead
# clresource create -g rg1 -t SUNW.gds \
-p Start_command="/usr/local/app/bin/start" \
-p Port_list="1234/tcp" -p Network_aware=True \
-p Resource_dependencies=hhead app-rs
# clresourcegroup online -M rg1
# clresourcegroup status +
```

复杂示例

此示例说明如何注册 SUNW.gds 资源类型、如何为应用程序创建资源组、为逻辑主机名 hhead 创建 LogicalHostname 资源、创建网络感知应用程序资源、仅记录错误消息、管理资源组、启用所有资源，以及如何使资源联机。

此时，应用程序正在运行，具有高可用性，并且由 Probe_command 指定的故障监视器监视。现在可以检查该应用程序的状态。

```
# clresourcetype register SUNW.gds
# clresourcegroup create rg1
# clreslogicalhostname create -g rg1 -h hhead
# clresource create -g rg1 -t SUNW.gds \
-p Start_command="/usr/local/app/bin/start" \
-p Stop_command="/usr/local/app/bin/stop" \
-p Validate_command="/usr/local/app/bin/config" \
-p Probe_command="/usr/local/app/bin/probe" \
-p stop_signal=9 -p failover_enabled=FALSE \
-p Start_timeout=120 -p Stop_timeout=180 \
-p Port_list="1234/tcp" -p Probe_timeout=60 \
-p Network_aware=True \
-p Validate_timeout=120 -p Log_level=Err \
```

```
-p Resource_dependencies=hhead app-rs
# clresourcegroup online -M rg1
# clresourcegroup status +
```

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/ha-service/gds

[clreslogicalhostname\(1CL\)](#) [207], [clresource\(1CL\)](#) [225], [clresourcegroup\(1CL\)](#) [253], [clresourcetype\(1CL\)](#) [277], [clressharedaddress\(1CL\)](#) [289], [rt_callbacks\(1HA\)](#) [545], [scdsbuilder\(1HA\)](#) [553], [scha_control\(1HA\)](#) [579], [scha_resource_get\(1HA\)](#) [585], [hatimerun\(1M\)](#) [623], [pmfadm\(1M\)](#) [625], [Unresolved link to "signal3HEAD"](#), [rt_reg\(4\)](#) [1067], [Unresolved link to "attributes5"](#), [r_properties\(5\)](#) [1103], [scalable_service\(5\)](#) [1153]

名称

SUNW.HAStoragePlus — 强制 Oracle Solaris Cluster 设备服务、文件系统与数据服务之间的依赖性并且监视这些实体的资源类型

SUNW.HAStoragePlus 描述的资源类型允许您指定数据服务资源和设备组、群集文件系统与本地文件系统之间的依赖性。

注 - 本地文件系统包括 UFS、Oracle 的 Sun QFS 和 Oracle Solaris ZFS。

此资源类型允许您使数据服务仅在依赖数据服务的设备组和文件系统肯定可用时联机。SUNW.HAStoragePlus 资源类型为挂载、卸载和检查文件系统提供支持。

资源类型本身并不提供与磁盘设备组、群集文件系统或本地文件系统的直接同步。因此，在群集重新引导或故障转移期间，系统可能会在依赖数据服务的全局设备和文件系统仍不可用的情况下尝试启动该数据服务。后果是，该数据服务的 START 方法可能超时，而数据服务可能失败。

SUNW.HAStoragePlus 资源类型表示将由一个或多个数据服务资源使用的设备组、群集和本地文件系统。可将 SUNW.HAStoragePlus 类型的资源添加到资源组中，并设置其他资源与该 SUNW.HAStoragePlus 之间的依赖性。

如果某个应用程序资源基于 HAStoragePlus 资源进行配置，则该应用程序资源必须定义对该底层 HAStoragePlus 资源的脱机重新启动依赖性。这样可以确保该应用程序资源在相关 HAStoragePlus 资源联机之后联机，而在相关 HAStoragePlus 资源脱机之前脱机。例如：

```
# clresource set \  
-p Resource_dependencies_offline_restart=hasp_rs \  
applicaton_rs
```

这些依赖性可确保数据服务资源在以下情况下联机：

1. 所有指定设备服务均可用（并且在必要时列出）。
2. 已检查并挂载指定的所有文件系统。

SUNW.HAStoragePlus 资源类型还提供了故障监视器，用于监视由 HAStoragePlus 资源管理的实体（包括全局设备、文件系统和 ZFS 存储池）的运行状况。故障监视器定期运行故障探测。如果其中某个实体变得不可用，则该资源将会重新启动或者故障转移到其他节点。

如果监视多个实体，故障监视器会同时探测所有这些实体。要查看针对全局设备、原始设备组、Solaris Volume Manager 设备组、文件系统和 ZFS 存储池监视的内容列表，请参见[Unresolved link to "Oracle Solaris Cluster 数据服务规划和管理指南" 中的第 2 章 数据服务资源管理](#)。

HASStoragePlus 资源故障监视器通过读写文件系统对该资源管理的设备和文件系统进行探测。如果读取操作被 I/O 栈中的任何软件阻止，而且要求 HASStoragePlus 资源联机，则用户必须禁用故障监视器。例如，必须对管理 Oracle Solaris 的 Availability Suite 远程复制卷功能的 HASStoragePlus 资源取消监视，因为 Availability Suite 在 NEED SYNC 状态下会阻止读取任何位图卷或数据卷。管理 Availability Suite 卷的 HASStoragePlus 资源必须始终处于联机状态。

如果 ZFS 文件系统的 mountpoint 属性设置为 none 或 legacy，或者其 canmount 属性设置为 off，则 HASStoragePlus 资源不会监视该文件系统。对于其他所有 ZFS 文件系统，HASStoragePlus 资源故障监视器会检查文件系统是否已挂载。如果文件系统已挂载，HASStoragePlus 资源随后会通过对其文件系统进行读写操作探测其可访问性，具体取决于名为 ReadOnly/ReadWrite 的 IOOption 属性的值。

如果 ZFS 文件系统未挂载或者文件系统探测失败，则资源故障监视器失败，资源将设置为 Faulted。RGM 将尝试重新启动资源，具体取决于资源的 retry_count 和 retry_interval 属性。如果上述特定的 mountpoint 和 canmount 属性设置未起作用，该操作会导致重新挂载文件系统。如果故障监视器操作继续失败，并且超过了 retry_interval 中的 retry_count，则 RGM 将资源故障转移到其他节点。

标准属性

以下标准属性与 SUNW.HASStoragePlus 资源类型关联：

Thorough_probe_interval

定义在调用故障探测与调用资源之间的时间段（秒）。

类别

可选

最小值

5

最大值

3600

默认值

180

可调

随时

扩展属性

以下扩展属性与 SUNW.HASStoragePlus 资源类型关联：

AffinityOn

指定 SUNW.HAStoragePlus 资源是否需要为 GlobalDevicePaths 和 FileSystemMountPoints 扩展属性中定义的所有全局设备执行关联性切换。可以指定 TRUE 或 FALSE。

类别

可选

默认值

TRUE

可调

禁用时

Zpools 扩展属性会忽略 AffinityOn 扩展属性。AffinityOn 扩展属性设计为仅与 GlobalDevicePaths 和 FileSystemMountPoints 扩展属性一起使用。

将 AffinityOn 扩展属性设置为 FALSE 时，SUNW.HAStoragePlus 资源会被动地等待指定的全局服务变为可用。在这种情况下，每个联机的全局设备服务的主节点可能与资源组的主节点不同。

关联性切换的作用是通过确保设备组与资源组共同位于一个特定节点上来增强性能。数据的读写始终通过设备主路径进行。关联性切换要求资源组的潜在主节点列表与设备组的节点列表相同。SUNW.HAStoragePlus 资源对每个设备服务仅执行一次关联性切换，即当该 SUNW.HAStoragePlus 资源联机时执行。

对于可伸缩服务，会忽略 AffinityOn（启用关联性）标志的设置。关联性切换不适用于可伸缩资源组。

FileSystemCheckCommand

覆盖 SUNW.HAStoragePlus 在尝试挂载每个已卸载的文件系统之前对其执行的检查。可以指定替代命令字符串或可执行文件，它对所有已卸载的文件系统进行调用。

类别

可选

默认值

NULL

可调

随时

在可伸缩资源组中配置了 SUNW.HAStoragePlus 资源时，对每个已卸载群集文件系统执行的文件系统检查将被忽略。将此扩展属性设置为 NULL 时，Oracle Solaris Cluster 通过发出 /usr/sbin/fsck -o p 命令检查 UFS。Oracle Solaris Cluster 通过发出 /usr/sbin/fsck 命令检查其他文件系统。

将 FileSystemCheckCommand 扩展属性设置为其他命令字符串时，SUNW.HAStoragePlus 将以文件系统挂载点作为参数调用此命令字符串。您可以

使用此方法指定任意可执行文件。非零返回值被视为在文件系统检查操作期间发生了错误。此错误导致 START 方法失败。

如果不需要执行文件系统检查操作，可将 FileSystemCheckCommand 扩展属性设置为 /bin/true。

FileSystemMountPoints

指定有效文件系统挂载点的列表。可以指定全局或本地文件系统。可从群集中的所有节点访问全局文件系统。只能从一个群集节点访问本地文件系统。SUNW.HAStoragePlus 资源管理的本地文件系统在单个群集节点中挂载。这些本地文件系统要求底层设备是 Oracle Solaris Cluster 全局设备。

类别

可选

默认值

空列表

可调

随时

这些文件系统挂载点以 paths[,...] 格式定义。

在所有群集节点和所有全局区域中，每个文件系统挂载点在 /etc/vfstab 中都应该有等效的条目。在非全局区域中，SUNW.HAStoragePlus 资源类型不检查 /etc/vfstab。

指定本地文件系统的 SUNW.HAStoragePlus 资源只能属于启用了关联性切换的故障转移资源组。因此，这些本地文件系统可以称为故障转移文件系统。可以同时指定本地和全局文件系统挂载点。

在 FileSystemMountPoints 扩展属性中列出其挂载点的任何文件系统被视为本地文件系统，前提是其 /etc/vfstab 条目满足以下两个条件：

1. 指定了非全局挂载选项。
2. 该条目的 "mount at boot" 字段设置为 "no"。

Oracle Solaris ZFS 文件系统始终是本地文件系统。请勿在 /etc/vfstab 中列出 ZFS 文件系统。此外，请勿在 FileSystemMountPoints 属性中包括 ZFS 挂载点。

GlobalDevicePaths

指定有效全局设备组名称或全局设备路径的列表。这些路径以 paths[,...] 格式定义。

类别

可选

默认值

空列表

可调

禁用时

IOOption

定义为了探测文件系统而执行的 I/O 类型。仅支持值 `ReadOnly` 和 `ReadWrite`。`ReadOnly` 值指示允许故障监视器对管理的文件系统执行只读 I/O，包括 `FileSystemMountPoints` 属性中指定的文件系统，以及属于 `Zpools` 属性指定的 ZFS 存储池的 ZFS 文件系统。`ReadWrite` 值指示允许故障监视器对管理的文件系统执行读和写 I/O。

类别

可选

默认值

`ReadOnly`

可调

随时

IOTimeout

定义 I/O 探测的超时值（秒）。

类别

可选

最小值

10

最大值

3600

默认值

300

可调

随时

Monitor_retry_count

控制允许进程监视设备 (Process Monitor Facility, PMF) 重新启动故障监视器的次数。

类别

可选

最小值

1

默认值

4

可调

随时

Monitor_retry_interval

定义重新启动故障监视器间隔的时间（分钟）。

类别

可选

最小值

2

默认值

2

可调

随时

RebootOnFailure

指定在探测器检测到失败时是否重新引导本地系统。设置为 TRUE 时，资源直接或间接使用的所有设备都必须受磁盘路径监视功能的监视。

如果 RebootOnFailure 设置为 TRUE，且至少有一个设备可用于

GlobalDevicePaths、FileSystemMountPoints 或 Zpools 属性中指定的每个实体，则本地系统将重新引导。本地系统表示资源处于联机状态的全局群集节点或区域群集节点。

类别

可选

默认值

FALSE

可调

随时

Zpools

指定有效 ZFS 存储池的列表，每个池至少包含一个 ZFS。这些 ZFS 存储池以 paths[,...] 格式定义。一个 ZFS 存储池中的所有文件系统都一起挂载和卸载。

类别

可选

默认值

空列表

可调

随时

使用 Zpools 扩展属性可指定 ZFS 存储池。组成 ZFS 存储池的设备必须可从 SUNW.HAStoragePlus 资源所属资源组的节点列表中配置的所有节点访问。管理 ZFS 存储池的 SUNW.HAStoragePlus 资源只能属于故障转移资源组。

使管理 ZFS 存储池的 SUNW.HAStoragePlus 资源联机时，该 ZFS 存储池将被导入，而且其中包含的每个文件系统都会挂载。

使该资源在节点上脱机时，对于每个管理的 ZFS 存储池，会卸载所有文件系统并导出该 ZFS 存储池。

注 - SUNW.HAStoragePlus 不支持在 ZFS 卷上创建的文件系统。

ZpoolsSearchDir

指定搜索 Zpools 中设备的位置。ZpoolsSearchDir 扩展属性类似于 zpool 命令的 -d 选项。

类别

可选

默认值

/dev/dsk

可调

禁用时

有关下列属性的说明，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core

[rt_reg\(4\) \[1067\]](#)、[Unresolved link to "attributes5"](#)

使给定资源组中的数据服务资源依赖于某个 SUNW.HAStoragePlus 资源。否则，无法在数据服务与全局设备或文件系统之间实现同步。脱机重新启动资源依赖性可确保

SUNW.HAStoragePlus 资源在其他资源之前联机。SUNW.HAStoragePlus 资源管理的本地文件系统仅在该资源联机后才会挂载。

启用 UFS 系统登录。

应避免在不同资源组中配置多个表示同一个设备组且 AffinityOn 标志设置为 TRUE 的 SUNW.HAStoragePlus 资源。否则可能发生冗余设备切换。此时，资源组和设备组可能发生定位错误。

避免在不同资源组中的多个 SUNW.HAStoragePlus 资源下配置一个 ZFS 存储池。

故障监视器错误

故障监视器可监视 HAStoragePlus 资源管理的实体，包括全局设备、文件系统和 ZFS 存储池。受监视实体的状态包括：

- 联机 – 无部分错误或严重错误。
- 降级 – 部分错误。
- 故障 – 严重错误。资源组管理器 (Resource Group Manager, RGM) 尝试重新启动资源并使其故障转移到其他群集节点。

如果监视多个实体，该资源的状态由所有受监视实体的聚合状态决定。

注 - 如果在故障监视器正在运行时更改了受管理实体的配置，可能导致故障监视器失败而退出，致使资源重新启动。您应该在对任何受管理实体进行配置更改之前禁用故障监视器，然后再重新启用故障监视器。配置更改可能包括删除 ZFS 存储池或池中的 ZFS 文件系统，或删除 Solaris Volume Manager 磁盘集或卷。

SUNW.HAStoragePlus 资源可以挂载任何处于卸载状态的群集文件系统。

所有文件系统以叠加模式挂载。

本地文件系统被强制卸载。

等待所有设备服务和文件系统变为可用的时间由 SUNW.HAStoragePlus 的 Prenet_start_timeout 属性指定。这是一个可调属性。

名称

SUNW.Proxy_SMF_failover, Proxy_SMF_failover — 代理故障转移 SMF 服务的资源类型

SUNW.Proxy_SMF_failover 资源类型表示服务管理设备 (Service Management Facility, SMF) 服务的故障转移的代理。

下面的子部分中介绍了为 SUNW.proxysmffailover 资源类型定义的标准属性和扩展属性。要为 SUNW.Proxy_SMF_failover 资源类型的实例设置这些属性，请使用 `clresource` 命令 ([clresource\(1CL\) \[225\]](#))。

标准属性

有关以下资源属性的描述，请参见 [r_properties\(5\) \[1103\]](#)。

Start_timeout

默认值：3600

最小值：60

Stop_timeout

默认值：3600

最小值：60

Init_timeout

默认值：3600

最小值：60

Boot_timeout

默认值：3600

最小值：60

Fini_timeout

默认值：3600

最小值：60

Validate_timeout

默认值：3600

最小值：60

Failover_mode

默认值：SOFT

可调：Anytime

R_description

默认值：""

可调：Anytime

Retry_count

默认值：2

最小值：0

最大值：10

可调：Anytime

Retry_interval

默认值：300

最大值：3600

可调：Anytime

Through_probe_interval

默认值：60

可调：Anytime

扩展属性

Proxied_service_instances

包含关于将由资源代理的 SMF 服务的信息。其值是指向包含所有代理的 SMF 服务的文件的路径。该文件中的每一行专用于一个 SMF 服务，并指定 `svc_fmri` 和指向相应服务清单文件的路径。例如，如果资源需要管理两个服务 `restarter_svc_test_1:default` 和 `restarter_svc_test_2:default`，该文件应该包含以下两行：

```
<svc:/system/cluster/restarter_svc_test_1:default>,  
  svc:/system/cluster/restarter_svc_test_1:default>,  
  </var/svc/manifest/system/cluster/restarter_svc_test_1.xml>  
  
<svc:/system/cluster/restarter_svc_test_2:default>,  
  </var/svc/manifest/system/cluster/restarter_svc_test_2.xml>
```

注 - 以上条目应该位于单独一行中。在此处分成多行是为了便于阅读。

默认值：""

可调：When disabled (禁用时)

示例

此示例说明如何注册 `SUNW.Proxy_SMF_failover` 资源类型、为应用程序创建资源组、创建故障转移应用程序资源、管理资源组，以及如何启用资源。

注册资源类型：

```
# clresourcetype -f <path-to-rtrfile> SUNW.Proxy_SMF_failover
```

为应用程序创建一个名为 `rg1` 的资源组：

```
# clresourcegroup create rg1
```

创建名为 `myfailoverres` 的故障转移应用程序资源：

```
# clresource create -t SUNW.Proxy_SMF_failover -g rg1 \  
-x proxied_service_instances="/usr/local/app/svc myfailoverres"
```

其中 `/usr/local/app/svc` 是文本文件。

管理资源组 `rg1`：

```
# clresourcegroup manage rg1
```

启用 `myfailoverres` 资源：

```
# clresource enable myfailoverres
```

使用以下命令检查应用程序状态：

```
# clresource status
```

[pmfadm\(1M\) \[625\]](#)、[scha_resource_get\(1HA\) \[585\]](#)、[clresourcetype\(1CL\) \[277\]](#)、[clresource\(1CL\) \[225\]](#)、[c](#)
[link to " attributes5"](#)、[r_properties\(5\) \[1103\]](#)

[Unresolved link to " Oracle Solaris Cluster 数据服务规划和管理指南 "](#)

名称

SUNW.Proxy_SMF_multimaster, Proxy_SMF_multimaster — 代理多主节点 SMF 服务的资源类型

SUNW.Proxy_SMF_multimaster 资源类型表示多主节点服务管理设备 (Service Management Facility, SMF) 服务的代理。

下面的子部分中介绍了为 SUNW.proxysmfmultimaster 资源类型定义的标准属性和扩展属性。要为 SUNW.Proxy_SMF_multimaster 资源类型的实例设置这些属性，请使用 `clresource` 命令 ([clresource\(1CL\) \[225\]](#))。

标准属性

有关以下资源属性的描述，请参见 [r_properties\(5\) \[1103\]](#)。

Start_timeout

默认值：3600

最小值：60

Stop_timeout

默认值：3600

最小值：60

Init_timeout

默认值：3600

最小值：60

Boot_timeout

默认值：3600

最小值：60

Fini_timeout

默认值：3600

最小值：60

Validate_timeout

默认值：3600

最小值：60

Failover_mode

默认值：SOFT

可调：Anytime

R_description

默认值：""

可调：Anytime

Retry_count

默认值：2

最小值：0

最大值：3

可调：Anytime

Retry_interval

默认值：300

最大值：3600

可调：Anytime

Through_probe_interval

默认值：60

可调：Anytime

扩展属性

Proxied_service_instances

包含关于将由资源代理的 SMF 服务的信息。其值是指向包含所有代理的 SMF 服务的文件的路径。该文件中的每一行专用于一个 SMF 服务，并指定 svc_fmri 和指向相应服务清单文件的路径。例如，如果资源需要管理两个服务 restarter_svc_test_1:default 和 restarter_svc_test_2:default，该文件应该包含以下两行：

```
<svc:/system/cluster/restarter_svc_test_1:default>,  
  svc:/system/cluster/restarter_svc_test_1:default>,  
  </var/svc/manifest/system/cluster/restarter_svc_test_1.xml>  
  
<svc:/system/cluster/restarter_svc_test_2:default>,  
  </var/svc/manifest/system/cluster/restarter_svc_test_2.xml>
```

注 - 以上条目应该位于单独一行中。在此处分成多行是为了便于阅读。

默认值：""

可调：When disabled (禁用时)

名称

SUNW.Proxy_SMF_scalable, Proxy_SMF_scalable — 代理可伸缩 SMF 服务的资源类型

SUNW.Proxy_SMF_scalable 资源类型表示可伸缩服务管理设备 (Service Management Facility, SMF) 服务的代理。

下面的子部分中介绍了为 SUNW.proxysmfscalable 资源类型定义的标准属性和扩展属性。要设置 SUNW.Proxy_SMF_scalable 资源类型的实例的这些属性，请使用 `clresource` 命令。

标准属性

有关以下资源属性的描述，请参见 [r_properties\(5\) \[1103\]](#)。

Start_timeout

默认值：3600

最小值：60

Stop_timeout

默认值：3600

最小值：60

Init_timeout

默认值：3600

最小值：60

Boot_timeout

默认值：3600

最小值：60

Fini_timeout

默认值：3600

最小值：60

Validate_timeout

默认值：3600

最小值：60

Failover_mode

默认值：SOFT

可调：任何时候

R_description

默认值：""

可调：任何时候

Retry_count

默认值：2

最小值：0

最大值：3

可调：任何时候

Retry_interval

默认值：300

最大值：3600

可调：任何时候

Through_probe_interval

默认值：60

可调：任何时候

扩展属性

Proxied_service_instances

包含关于将由资源代理的 SMF 服务的信息。其值是指向包含所有代理的 SMF 服务的文件的路径。该文件中的每一行专用于一个 SMF 服务，并指定 `svc_fmri` 和指向相应服务清单文件的路径。例如，如果资源需要管理两个服务 `restarter_svc_test_1:default` 和 `restarter_svc_test_2:default`，该文件应该包含以下两行：

```
<svc:/system/cluster/restarter_svc_test_1:default>,  
  svc:/system/cluster/restarter_svc_test_1:default>,  
  </var/svc/manifest/system/cluster/restarter_svc_test_1.xml>  
  
<svc:/system/cluster/restarter_svc_test_2:default>,  
  </var/svc/manifest/system/cluster/restarter_svc_test_2.xml>
```

注 - 以上条目应该位于单独一行中。在此处分成多行是为了便于阅读。

默认值：""

可调：when disabled (禁用时)

名称

SUNW.rac_framework, rac_framework — 启用 Oracle Solaris Cluster Support for Oracle Real Application Clusters (Oracle RAC) 的框架的资源类型实现

SUNW.rac_framework 资源类型表示启用 Oracle Solaris Cluster Support for Oracle RAC 的框架。此资源类型允许您监视该框架的状态。

SUNW.rac_framework 资源类型是单实例资源类型。在群集中只能创建一个此类型的资源。

要注册此资源类型并创建该资源类型的实例，请使用以下方法之一：

- Oracle Solaris Cluster Manager
- 使用 `clsetup(1CL)` [407] 实用程序，并指定用于配置 Oracle Solaris Cluster Support for Oracle Real Application Clusters 的选项
- 以下 Oracle Solaris Cluster 维护命令序列：
 1. 要注册此资源类型，请使用 `clresourcetype(1CL)` [277] 命令。
 2. 要创建此资源类型的实例，请使用 `clresource(1CL)` [225] 命令。

当此类资源在某个节点上脱机时，从联机状态转换为脱机状态需要一定的时间才能完成。在转换为脱机状态期间，该资源将继续参与重新配置过程。但是，资源在某个节点上处于脱机状态时，对其属性的更改直到此资源重新联机才会在该节点上生效。Oracle Solaris Cluster 在此类资源被禁用时会针对此效果显示一条警告消息。

将包含此类资源的资源组的状态转换为不受管理需要一定的时间才能完成。在转换为不受管理状态期间，Oracle RAC 框架将继续参与框架重新配置过程。但是，资源组处于不受管理状态时，对资源属性的更改不会在节点上生效。要停止 Oracle RAC 框架，必须重新引导节点。

SUNW.rac_framework 资源类型的扩展属性如下所示：

`reservation_timeout`

键入整数，最小值为 100，最大值为 99999，默认值为 325。此属性为 Oracle Solaris Cluster Support for Oracle RAC 重新配置过程的保留步骤指定超时值（以秒为单位）。您可以随时修改该属性。

例 405 创建 rac_framework 资源

此示例注册 SUNW.rac_framework 资源类型并创建 SUNW.rac_framework 资源类型的名为 rac_framework-rs 的实例。该示例假定已创建了一个名为 rac-framework-rg 的资源组。

```
phys-host-sc1# clresourcetype register SUNW.rac_framework
```

```
phys-host-scl# clresource create -g rac-framework-rg \  
-t SUNW.rac_framework rac_framework-rs
```

例 406 更改 rac_framework 资源的属性

此示例将 Oracle Solaris Cluster Support for Oracle RAC 重新配置过程的保留步骤的超时值设置为 350 秒。示例假定已创建一个类型为 SUNW.rac_framework 的名为 rac_framework-rs 的资源。

```
phys-host-scl# clresource set  
\-p reservation_timeout=350 rac_framework-rs
```

注 - 有关创建卷管理器资源使用的 SUNW.vucmm_framework 资源的示例，请参见 [SUNW.vucmm_framework\(5\) \[1231\]](#)。

有关以下属性的描述，请参见 [Unresolved link to " attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/library/ucmm

[clresource\(1CL\) \[225\]](#)、[clresourcetype\(1CL\) \[277\]](#)、[clsetup\(1CL\) \[407\]](#)、[Unresolved link to " attributes5"](#)、[SUNW.vucmm_framework\(5\) \[1231\]](#)

[Unresolved link to " 适用于 Oracle Real Application Clusters 的 Oracle Solaris Cluster 数据服务指南 "](#)

名称

SUNW.ScalDeviceGroup, ScalDeviceGroup — 可伸缩设备组的资源类型实现

SUNW.ScalDeviceGroup 资源类型表示可伸缩设备组。该资源类型的实例表示以下设备组类型：

- Solaris Volume Manager for Sun Cluster 多属主磁盘集

SUNW.ScalDeviceGroup 资源类型是可伸缩资源类型。此资源类型的实例在该资源所属资源组的节点列表中的每个节点上联机。

要注册此资源类型并创建该资源类型的实例，请使用以下方法之一：

- Oracle Solaris Cluster Manager
- 使用 `clsetup` 实用程序，并指定用于配置 Oracle Solaris Cluster Support for Oracle Real Application Clusters 的选项
- 以下 Oracle Solaris Cluster 维护命令序列：
 1. 要注册此资源类型，请使用 `clresourcetype` 命令。
 2. 要创建此资源类型的实例，请使用 `clresource` 命令。

下面的子部分中介绍了为 SUNW.ScalDeviceGroup 资源类型定义的标准属性和扩展属性。

标准属性

有关所有标准资源属性的描述，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

按如下所示覆盖此资源类型的标准资源属性：

Monitor_start_timeout

最小值

10

默认值

300

Monitor_stop_timeout

最小值

10

默认值

300

Postnet_stop_timeout

最小值
60

默认值
300

Prenet_start_timeout

最小值
60

默认值
300

Start_timeout

最小值
60

默认值
300

Stop_timeout

最小值
60

默认值
300

Thorough_probe_interval

默认值
300

Update_timeout

最小值
60

默认值
300

Validate_timeout

最小值
60

默认值
300

扩展属性

此资源类型的扩展属性如下所示：

Debug_level

该属性指定来自此类型资源的调试消息的记录级别。调试级别越高，写入日志文件的调试消息越多。

数据类型
整数

默认值
0

范围
0-10

可调
随时

DiskGroupName

该属性指定资源所表示的设备组的名称。必须将该属性设置为以下项目之一：

- 现有 Solaris Volume Manager for Sun Cluster 多属主磁盘集的名称。该名称是在创建磁盘集所用的 [Unresolved link to "metaset1M"](#) 命令中指定的。

所指定设备组的要求如下所示：

- 该设备组必须是有效的现有多属主磁盘集或共享磁盘组。
- 该设备组必须托管于可以控制资源的所有节点上。
- 该设备组必须可从可以控制可伸缩设备组资源的所有节点上访问。
- 该设备组必须至少包含一个卷。

数据类型
字符串

默认值
未定义默认值

范围
不适用

可调
禁用时

IOTimeout

该属性指定 I/O 探测的超时值（以秒为单位）。

默认值
30

范围
1–1800

可调
随时

LogicalDeviceList

该属性以逗号分隔方式指定资源的故障监视器要监视的逻辑卷的列表。此属性是可选的。如果没有为该属性指定值，设备组中的所有逻辑卷均会受到监视。

设备组的状态派生自受监视的各个逻辑卷的状态。如果所有受监视的逻辑卷都运行良好，则说明设备组正常运行。如果有任何受监视的逻辑卷发生故障，则说明设备组存在故障。

通过查询卷的卷管理器可获取各个逻辑卷的状态。如果无法通过查询确定 Solaris Volume Manager for Sun Cluster 卷的状态，故障监视器将执行文件输入/输出 (input/output, I/O) 操作来确定卷状态。

如果发现设备组发生故障，将停止对表示资源组的资源的监视并将该资源置于禁用状态。

注 - 对于镜像磁盘，即使某个子镜像发生故障，也仍认为设备组运行良好。

注 - 使用 Solaris Volume Manager for Sun Cluster 时，如果将指定监视软分区，则还应该指定相应的顶层或底层元设备。

所指定的各逻辑卷的要求如下所示：

- 必须存在逻辑卷。
- 逻辑卷必须包含在 diskgroupname 属性所指定的设备组中。
- 逻辑卷必须可从可以控制可伸缩设备组资源的所有节点上访问。

数据类型
字符串数组

默认值

""

范围

不适用

可调

随时

Monitor_retry_count

该属性指定进程监视器工具 (process monitor facility, PMF) 可以重新启动故障监视器的最大次数。

数据类型

整数

默认值

4

范围

未定义范围

可调

随时

Monitor_retry_interval

该属性指定 PMF 对故障监视器的重新启动进行计数的时间间隔（以分钟为单位）。

数据类型

整数

默认值

2

范围

未定义范围

可调

随时

RebootOnFailure

该属性指定在探测器检测到失败时是否重新引导本地系统。该属性设置为 TRUE 时，资源直接或间接使用的所有设备都必须受磁盘路径监视功能的监视。

如果 RebootOnFailure 设置为 TRUE，且至少有一个设备可用于

GlobalDevicePaths、FileSystemMountPoints 或 Zpools 属性中指定的每个实体，本

地系统将重新引导。本地系统表示资源处于联机状态的全局群集节点或区域群集节点。

默认值

FALSE

可调

随时

例 407 使用 SUNW.vucmm_svm 资源类型创建 ScalDeviceGroup 资源

本示例显示了如何使用 SUNW.vucmm_svm 资源类型创建一个 ScalDeviceGroup 资源来表示名为 datadg 的 Solaris Volume Manager for Sun Cluster 多属主磁盘集。该资源命名为 scaldatadg-rs。此示例假定存在以下 Oracle Solaris Cluster 对象：

- 名为 scaldatadg-rg 的可伸缩资源组。
- 名为 vucmm-svm-rs 的 SUNW.vucmm_svm 资源类型实例。

```
# clresourcetype register SUNW.ScalDeviceGroup
# clresource create -t SUNW.ScalDeviceGroup \
-g scaldatadg-rg \
-p Resource_dependencies=vucmm-svm-rs \
-p DiskGroupName=datadg \
scaldatadg-rs
```

有关以下属性的描述，请参见 [Unresolved link to " attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core

[clresource\(1CL\) \[225\]](#)、[clresourcetype\(1CL\) \[277\]](#)、[clsetup\(1CL\) \[407\]](#)、[Unresolved link to " metaset1M"](#)、[Unresolved link to " attributes5"](#)、[r_properties\(5\) \[1103\]](#)、[SUNW.vucmm_svm\(5\) \[1233\]](#)

[Unresolved link to " 适用于 Oracle Real Application Clusters 的 Oracle Solaris Cluster 数据服务指南 "](#)

名称

SUNW.ScalMountPoint, ScalMountPoint — 可伸缩文件系统挂载点的资源类型实现

SUNW.ScalMountPoint 资源类型表示可伸缩的文件系统挂载点。此资源类型的实例表示以下文件系统类型的挂载点：

- Sun QFS 共享文件系统
- 网络连接的存储 (network-attached storage, NAS) 设备上的文件系统。
NAS 设备和文件系统必须已配置为可用于 Oracle Solaris Cluster。有关更多信息，请参见[Unresolved link to " Oracle Solaris Cluster With Network-Attached Storage Device Manual "](#)。

SUNW.ScalMountPoint 资源类型是可伸缩资源类型。此资源类型的实例在该资源所属资源组的节点列表中的每个节点上联机。

要注册此资源类型并创建该资源类型的实例，请使用以下方法之一：

- Oracle Solaris Cluster Manager
- 使用 `clsetup` 实用程序，并指定用于配置 Oracle Solaris Cluster Support for Oracle Real Application Clusters 的选项
- 以下 Oracle Solaris Cluster 维护命令序列：
 1. 要注册此资源类型，请使用 `clresourcetype` 命令。
 2. 要创建此资源类型的实例，请使用 `clresource` 命令。

下面的子部分中介绍了为 SUNW.ScalMountPoint 资源类型定义的标准属性和扩展属性。

标准属性

有关所有标准资源属性的描述，请参见 [r_properties\(5\) \[1103\]](#) 手册页。

按如下所示覆盖此资源类型的标准资源属性：

Monitor_start_timeout

最小值

10

默认值

300

Monitor_stop_timeout

最小值

10

默认值
300

Postnet_stop_timeout

最小值
60

默认值
300

Prenet_start_timeout

最小值
60

默认值
300

Start_timeout

最小值
60

默认值
300

Stop_timeout

最小值
60

默认值
300

Thorough_probe_interval

默认值
300

Update_timeout

最小值
60

默认值
300

Validate_timeout

最小值
60

默认值
300

扩展属性

此资源类型的扩展属性如下所示：

Debug_level

该属性指定来自文件系统挂载点资源的调试消息的记录级别。调试级别越高，写入日志文件的调试消息越多。

数据类型
整数

默认值
0

范围
0–10

可调
随时

FileSystemType

该属性指定资源表示的挂载点对应的文件系统类型。您必须指定此属性。将该属性设置为以下某一值：

nas
指定文件系统为 NAS 设备上的文件系统。

s-qfs
指定文件系统为 Sun QFS 共享文件系统。

数据类型
字符串

默认值
未定义默认值

范围
不适用

可调

禁用时

IOTimeout

该属性指定故障监视器探测文件输入/输出 (input/output, I/O) 所使用的超时值 (以秒为单位)。为了确定已挂载文件系统是否可用, 故障监视器会对文件系统上的测试文件执行 I/O 操作, 如打开、读取及写入操作。如果在超时时限内未完成 I/O 操作, 故障监视器将报告错误。

数据类型

整数

默认值

300

范围

5-1800

可调

随时

Monitor_retry_count

该属性指定进程监视器工具 (process monitor facility, PMF) 可以重新启动故障监视器的最大次数。

数据类型

整数

默认值

4

范围

未定义范围

可调

随时

Monitor_retry_interval

该属性指定 PMF 对故障监视器的重新启动进行计数的时间间隔 (以分钟为单位)。

数据类型

整数

默认值

2

范围

未定义范围

可调

随时

MountOptions

该属性指定在挂载资源所表示的文件系统时要使用的挂载选项的列表（以逗号分隔）。此属性是可选的。如果没有为该属性指定值，将从文件系统的默认值表中获取挂载选项。

- 对于 Sun QFS 共享文件系统，这些选项从 `/etc/opt/SUNWsamfs/samfs.cmd` 文件获取。
- 对于 NAS 设备上的文件系统，这些选项从 `/etc/vfstab` 文件获取。

通过该属性指定的挂载选项将覆盖文件系统的默认值表中的挂载选项。

数据类型

字符串

默认值

""

范围

不适用

可调

禁用时

MountPointDir

该属性指定资源所表示的文件系统的挂载点。挂载点是当挂载文件系统时此文件系统所附加到的文件系统分层结构的目录的完整路径。您必须指定此属性。

您所指定的目录必须已存在。

数据类型

字符串

默认值

未定义默认值

范围

不适用

可调

禁用时

RebootOnFailure

该属性指定在探测器检测到失败时是否重新引导本地系统。该属性设置为 TRUE 时，资源直接或间接使用的所有设备都必须受磁盘路径监视功能的监视。

如果 RebootOnFailure 设置为 TRUE，且至少有一个设备可用于 GlobalDevicePaths、FileSystemMountPoints 或 Zpools 属性中指定的每个实体，则本地系统将重新引导。本地系统表示资源处于联机状态的全局群集节点或区域群集节点。

默认值

FALSE

可调

随时

TargetFileSystem

该属性指定要在 MountPointDir 扩展属性所指定的挂载点上挂载的文件系统。您必须指定此属性。文件系统的类型必须与 FileSystemType 属性所指定的类型相匹配。该属性的格式取决于文件系统的类型，如下所示：

- 对于 Sun QFS 共享文件系统，将该属性设置为创建文件系统时为该文件系统指定的名称。该文件系统必须已正确配置。有关更多信息，请参见您的 Sun QFS 共享文件系统文档。
- 对于 NAS 设备上的文件系统，将该属性设置为 *nas-device:path*。该格式中的可替换项如下所示：

nas-device

指定正在导出文件系统的 NAS 设备的名称。（可选）您可以使用域对该名称进行限定。

path

指定 NAS 设备正在导出的文件系统的完整路径。

NAS 设备和文件系统必须已配置为可用于 Oracle Solaris Cluster。有关更多信息，请参见 [Unresolved link to " Oracle Solaris Cluster With Network-Attached Storage Device Manual "](#)。

数据类型

字符串

默认值

未定义默认值

范围

不适用

可调
禁用时

例 408 创建 ScalMountPoint 资源

本示例显示如何创建 ScalMountPoint 资源来表示用于 Solaris Volume Manager for Sun Cluster 的 Sun QFS 共享文件系统的挂载点。该资源命名为 scal-db_qfs-Data-rs。该文件系统的特性如下所述：

- 文件系统的挂载点是 /db_qfs/Data。
- 将挂载的文件系统是 Data。
- 挂载选项从该文件系统的默认值表获取，即 /etc/opt/SUNWsamfs/samfs.cmd 文件。

此示例假定存在以下 Oracle Solaris Cluster 对象：

- 名为 scaldatadg-rg 的可伸缩资源组。
- 名为 qfs-db_qfs-Data-rs 的 SUNW.qfs 资源类型实例。
- 名为 scaldatadg-rs 的 SUNW.ScalDeviceGroup 资源类型实例。

```
# clresourcetype register SUNW.ScalMountPoint
# clresource create -t SUNW.ScalMountPoint \
-g scaldatadg-rg \
-p Resource_dependencies=qfs-db_qfs-Data-rs,scaldatadg-rs \
-p MountPointDir=/db_qfs/Data \
-p FileSystemType=s-qfs \
-p TargetFileSystem=Data \
scal-db_qfs-Data-rs
```

有关以下属性的描述，请参见 [Unresolved link to " attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core

[clresource\(1CL\) \[225\]](#)、[clresourcetype\(1CL\) \[277\]](#)、[clsetup\(1CL\) \[407\]](#)、[Unresolved link to " vfstab4"](#)、[Unresolved link to " attributes5"](#)、[r_properties\(5\) \[1103\]](#)、[SUNWct.ScalDeviceGroup\(5\) \[1215\]](#)、[SUNW.vucmm_framework\(5\) \[1231\]](#)

[Unresolved link to " 适用于 Oracle Real Application Clusters 的 Oracle Solaris Cluster 数据服务指南 "](#)、[Unresolved link to " Oracle Solaris Cluster With Network-Attached Storage Device Manual "](#)

名称

SUNW.SCTelemetry, SCTelemetry — 用于收集系统资源使用情况相关数据的资源类型

SUNW.SCTelemetry 是允许您收集系统资源使用情况相关数据的资源类型。SUNW.SCTelemetry 将系统资源使用情况数据在 Java DB 数据库中存储七天。SUNW.SCTelemetry 类型的资源对 SUNW.derby 类型的资源具有依赖性。有关更多信息，请参见 [SUNW.derby\(5\) \[1175\]](#) 手册页。

与 SUNW.SCTelemetry 资源类型关联的扩展属性如下所述：

Extended_accounting_cleanup (布尔值)

指定是否清理扩展记帐日志文件，即是否删除历史数据。Extended_accounting_cleanup 的可能值是 TRUE 和 FALSE。

类别	可选
默认值	TRUE
可调	随时

Monitor_retry_count (整数)

控制故障监视器的重新启动。该属性指定进程监视器工具重新启动故障监视器的次数。该属性对应于传递给 [pmfadm\(1M\) \[625\]](#) 命令的 -n 选项。资源组管理器 (Resource Group Manager, RGM) 在指定时间段内对重新启动次数进行计数。有关更多信息，请参见 Monitor_retry_interval 属性。请注意，Monitor_retry_count 指的是故障监视器自身而非 SUNW.SCTelemetry 类型的资源的重新启动。

类别	可选
默认值	4
可调	随时

Monitor_retry_interval (整数)

以分钟为单位指定时间段，在此期间 RGM 对故障监视器的失败进行计数。该属性对应于传递给 [pmfadm\(1M\) \[625\]](#) 命令的 -t 选项。如果故障监视器的失败次数超出了 Monitor_retry_count 属性的值，则进程监视器工具不会重新启动故障监视器。

类别	可选
默认值	2 分钟
可调	随时

Probe_timeout (整数)

指定探测的超时值 (以秒为单位)。

类别	可选
默认值	60 秒
最小值	2 秒
可调	随时

Sampling_interval (整数)

指定收集监视数据的频率。Telemetry_sampling_interval 属性的值必须介于 30 和 3600 之间。

类别	必需
默认值	60
最小值	30 秒
最大值	3600 秒
可调	随时

[pmfadm\(1M\) \[625\]](#)、[SUNW.derby\(5\) \[1175\]](#)

名称

SUNW.vucmm_framework, vucmm_framework — Oracle Solaris Cluster 卷管理器重新配置框架的资源类型实现

SUNW.vucmm_framework 资源类型表示可在一个 Oracle Solaris Cluster 配置中启用不同群集卷管理器的框架。此资源类型允许您监视该框架的状态。

SUNW.vucmm_framework 资源类型是单实例资源类型。您只能在群集中创建一个此类型的资源。

要注册此资源类型并创建该资源类型的实例，请使用以下方法：

- 要注册此资源类型，请使用 `clresourcetype(1CL)` [277] 命令。
- 要创建此资源类型的实例，请使用 `clresource(1CL)` [225] 命令。

当此类资源在某个节点上脱机时，从联机状态转换为脱机状态需要一定的时间才能完成。在转换为脱机状态期间，该资源将继续参与重新配置过程。但是，资源在某个节点上处于脱机状态时，对其属性的更改直到此资源重新联机才会在该节点上生效。当禁用了此类型的资源时，Oracle Solaris Cluster 软件会显示一条关于此问题的警告消息。

将包含此类资源的资源组的状态转换为不受管理需要一定的时间才能完成。在转换为不受管理状态期间，该框架继续参与框架重新配置过程。但是，资源组处于不受管理状态时，对资源属性的更改不会在节点上生效。要停止框架，必须重新引导该节点。

SUNW.vucmm_framework 资源类型的扩展属性如下所示：

`reservation_timeout`

键入整数，最小值为 100，最大值为 99999，默认值为 325。该属性为框架重新配置过程的保留步骤指定超时值（以秒为单位）。您可以随时修改该属性。

例 409 创建 vucmm_framework 资源

此示例注册 SUNW.vucmm_framework 资源类型并创建 SUNW.vucmm_framework 资源类型的名为 vucmm_framework-rs 的实例。该示例假定已创建了一个名为 vucmm_framework-rg 的资源组。

```
phys-host-scl1# clresourcetype register SUNW.vucmm_framework
phys-host-scl1# clresource create -g vucmm_framework-rg \
-t SUNW.vucmm_framework vucmm_framework-rs
```

例 410 更改 vucmm_framework 资源的属性

本示例将框架重新配置过程的保留步骤的超时值设置为 350 秒。示例假定已创建一个类型为 SUNW.vucmm_framework 的名为 vucmm_framework-rs 的资源。

```
phys-host-scl# clresource set -p reservation_timeout=350 vucmm_framework-rs
```

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core

[clresource\(1CL\) \[225\]](#)、[clresourcetype\(1CL\) \[277\]](#)、[clsetup\(1CL\) \[407\]](#)、[SUNW.vucmm_svm\(5\) \[1233\]](#)、[Unresolved link to "attributes5"](#)

[Unresolved link to "Oracle Solaris Cluster 数据服务规划和管理指南"](#)

名称

SUNW.vucmm_svm, vucmm_svm — 表示卷管理器重新配置框架的 Solaris Volume Manager for Sun Cluster 组件的资源类型实现

SUNW.vucmm_svm 资源类型表示 Oracle Solaris Cluster 卷管理器重新配置框架的 Solaris Volume Manager for Sun Cluster 组件。

SUNW.vucmm_svm 资源类型的实例保存 Solaris Volume Manager for Sun Cluster 组件配置参数。此类型的实例还可显示 Solaris Volume Manager for Sun Cluster 组件的重新配置状态。

SUNW.vucmm_svm 资源类型是单实例资源类型。在群集中只能创建一个此类型的资源。

要注册此资源类型并创建该资源类型的实例，请使用以下方法：

- 要注册此资源类型，请使用 `clresourcetype(1CL) [277]` 命令。
- 要创建此资源类型的实例，请使用 `clresource(1CL) [225]` 命令。

SUNW.vucmm_svm 资源类型仅在使用 SUNW.vucmm_framework 资源类型的资源组中有效。请勿在 SUNW.rac_framework 资源组中使用此资源类型。

当此类资源在某个节点上脱机时，从联机状态转换为脱机状态需要一定的时间才能完成。在转换为脱机状态期间，该资源将继续参与重新配置过程。但是，资源在某个节点上处于脱机状态时，对其属性的更改直到此资源重新联机才会在该节点上生效。当禁用了此类型的资源时，Oracle Solaris Cluster 软件会显示一条关于此问题的警告消息。

将包含此类资源的资源组的状态转换为不受管理需要一定的时间才能完成。在转换为不受管理状态期间，卷管理器重新配置框架继续参与框架重新配置过程。但是，资源组处于不受管理状态时，对资源属性的更改不会在节点上生效。要停止卷管理器重新配置框架，必须重新引导该节点。

SUNW.vucmm_svm 资源类型的扩展属性如下所示。

`debug_level`

键入整数，最小值为 0，最大值为 10，默认值为 1。该属性指定卷管理器重新配置框架的 Solaris Volume Manager for Sun Cluster 模块的调试级别。调试级别越高，在重新配置期间写入日志文件的调试消息就越多。您可以随时修改该属性。

`svm_abort_step_timeout`

键入整数，最小值为 30，最大值为 99999，默认值为 120。该属性为卷管理器重新配置框架的 Solaris Volume Manager for Sun Cluster 模块重新配置过程的中止步骤指定超时值（以秒为单位）。您可以随时修改该属性。

`svm_return_step_timeout`

键入整数，最小值为 30，最大值为 99999，默认值为 120。该属性为卷管理器重新配置框架的 Solaris Volume Manager for Sun Cluster 模块重新配置过程的返回步骤指定超时值（以秒为单位）。您可以随时修改该属性。

`svm_start_step_timeout`

键入整数，最小值为 30，最大值为 99999，默认值为 120。该属性为卷管理器重新配置框架的 Solaris Volume Manager for Sun Cluster 模块重新配置过程的启动步骤指定超时值（以秒为单位）。您可以随时修改该属性。

`svm_step1_timeout`

键入整数，最小值为 30，最大值为 99999，默认值为 120。该属性为卷管理器重新配置框架的 Solaris Volume Manager for Sun Cluster 模块重新配置过程的步骤 1 指定超时值（以秒为单位）。您可以随时修改该属性。

`svm_step2_timeout`

键入整数；最小值为 30；最大值为 99999；默认值为 120。该属性为卷管理器重新配置框架的 Solaris Volume Manager for Sun Cluster 模块重新配置过程的步骤 2 指定超时值（以秒为单位）。您可以随时修改该属性。

`svm_step3_timeout`

键入整数，最小值为 30，最大值为 99999，默认值为 120。该属性为卷管理器重新配置框架的 Solaris Volume Manager for Sun Cluster 模块重新配置过程的步骤 3 指定超时值（以秒为单位）。您可以随时修改该属性。

`svm_step4_timeout`

键入整数，最小值为 100，最大值为 99999，默认值为 120。该属性为卷管理器重新配置框架的 Solaris Volume Manager for Sun Cluster 模块重新配置过程的步骤 4 指定超时值（以秒为单位）。您可以随时修改该属性。

`svm_stop_step_timeout`

键入整数，最小值为 30，最大值为 99999，默认值为 120。该属性为卷管理器重新配置框架的 Solaris Volume Manager for Sun Cluster 模块重新配置过程的停止步骤指定超时值（以秒为单位）。您可以随时修改该属性。

例 411 创建 `vucmm_svm` 资源

此示例注册 `SUNW.vucmm_svm` 资源类型并创建 `SUNW.vucmm_svm` 资源类型的名为 `vucmm_svm-rs` 的实例。此示例假定已创建以下 Oracle Solaris Cluster 对象：

- 名为 `vucmm_framework-rg` 的资源组。
- 类型为 `SUNW.vucmm_framework` 的名为 `vucmm_framework-rs` 的资源

```
phys-schost-1# clresourcetype register SUNW.vucmm_svm
phys-schost-1# clresource create -g vucmm_framework-rg \
```

```
-t SUNW.vucmm_svm \  
-p resource_dependencies=vucmm_framework-rs vucmm_svm-rs
```

例 412 更改 vucmm_svm 资源的属性

此示例将卷管理器重新配置框架的 Solaris Volume Manager for Sun Cluster 组件重新配置过程的步骤 4 的超时值设置为 300 秒。示例假定已创建一个名为 vucmm_svm-rs 的 SUNW.vucmm_svm 资源类型实例。

```
phys-schost-1# clresource set \  
-p svm_step4_timeout=300 vucmm_svm-rs
```

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core

[clresource\(1CL\) \[225\]](#)、[clresourcetype\(1CL\) \[277\]](#)、[clsetup\(1CL\) \[407\]](#)、[SUNW.vucmm_framework\(5\) \[1231\]](#)
[link to "attributes5"](#)

名称

SUNW.vucmm_framework, vucmm_framework — Oracle Solaris Cluster 卷管理器重新配置框架的资源类型实现

SUNW.vucmm_framework 资源类型表示可在一个 Oracle Solaris Cluster 配置中启用不同群集卷管理器的框架。此资源类型允许您监视该框架的状态。

SUNW.vucmm_framework 资源类型是单实例资源类型。您只能在群集中创建一个此类型的资源。

要注册此资源类型并创建该资源类型的实例，请使用以下方法：

- 要注册此资源类型，请使用 `clresourcetype(1CL)` [277] 命令。
- 要创建此资源类型的实例，请使用 `clresource(1CL)` [225] 命令。

当此类资源在某个节点上脱机时，从联机状态转换为脱机状态需要一定的时间才能完成。在转换为脱机状态期间，该资源将继续参与重新配置过程。但是，资源在某个节点上处于脱机状态时，对其属性的更改直到此资源重新联机才会在该节点上生效。当禁用了此类型的资源时，Oracle Solaris Cluster 软件会显示一条关于此问题的警告消息。

将包含此类资源的资源组的状态转换为不受管理需要一定的时间才能完成。在转换为不受管理状态期间，该框架继续参与框架重新配置过程。但是，资源组处于不受管理状态时，对资源属性的更改不会在节点上生效。要停止框架，必须重新引导该节点。

SUNW.vucmm_framework 资源类型的扩展属性如下所示：

`reservation_timeout`

键入整数，最小值为 100，最大值为 99999，默认值为 325。该属性为框架重新配置过程的保留步骤指定超时值（以秒为单位）。您可以随时修改该属性。

例 413 创建 vucmm_framework 资源

此示例注册 SUNW.vucmm_framework 资源类型并创建 SUNW.vucmm_framework 资源类型的名为 vucmm_framework-rs 的实例。该示例假定已创建了一个名为 vucmm_framework-rg 的资源组。

```
phys-host-scl1# clresourcetype register SUNW.vucmm_framework
phys-host-scl1# clresource create -g vucmm_framework-rg \
-t SUNW.vucmm_framework vucmm_framework-rs
```

例 414 更改 vucmm_framework 资源的属性

本示例将框架重新配置过程的保留步骤的超时值设置为 350 秒。示例假定已创建一个类型为 SUNW.vucmm_framework 的名为 vucmm_framework-rs 的资源。

```
phys-host-scl# clresource set -p reservation_timeout=350 vucmm_framework-rs
```

有关以下属性的描述，请参见 [Unresolved link to " attributes5"](#) :

属性类型	属性值
可用性	ha-cluster/system/core

[clresource\(1CL\) \[225\]](#)、[clresourcetype\(1CL\) \[277\]](#)、[clsetup\(1CL\) \[407\]](#)、[SUNW.vucmm_svm\(5\) \[1233\]](#)、[Unresolved link to " attributes5"](#)

[Unresolved link to " Oracle Solaris Cluster 数据服务规划和管理指南 "](#)

名称

SUNW.vucmm_svm, vucmm_svm — 表示卷管理器重新配置框架的 Solaris Volume Manager for Sun Cluster 组件的资源类型实现

SUNW.vucmm_svm 资源类型表示 Oracle Solaris Cluster 卷管理器重新配置框架的 Solaris Volume Manager for Sun Cluster 组件。

SUNW.vucmm_svm 资源类型的实例保存 Solaris Volume Manager for Sun Cluster 组件配置参数。此类型的实例还可显示 Solaris Volume Manager for Sun Cluster 组件的重新配置状态。

SUNW.vucmm_svm 资源类型是单实例资源类型。在群集中只能创建一个此类型的资源。

要注册此资源类型并创建该资源类型的实例，请使用以下方法：

- 要注册此资源类型，请使用 `clresourcetype(1CL) [277]` 命令。
- 要创建此资源类型的实例，请使用 `clresource(1CL) [225]` 命令。

SUNW.vucmm_svm 资源类型仅在使用 SUNW.vucmm_framework 资源类型的资源组中有效。请勿在 SUNW.rac_framework 资源组中使用此资源类型。

当此类资源在某个节点上脱机时，从联机状态转换为脱机状态需要一定的时间才能完成。在转换为脱机状态期间，该资源将继续参与重新配置过程。但是，资源在某个节点上处于脱机状态时，对其属性的更改直到此资源重新联机才会在该节点上生效。当禁用了此类型的资源时，Oracle Solaris Cluster 软件会显示一条关于此问题的警告消息。

将包含此类资源的资源组的状态转换为不受管理需要一定的时间才能完成。在转换为不受管理状态期间，卷管理器重新配置框架继续参与框架重新配置过程。但是，资源组处于不受管理状态时，对资源属性的更改不会在节点上生效。要停止卷管理器重新配置框架，必须重新引导该节点。

SUNW.vucmm_svm 资源类型的扩展属性如下所示。

`debug_level`

键入整数，最小值为 0，最大值为 10，默认值为 1。该属性指定卷管理器重新配置框架的 Solaris Volume Manager for Sun Cluster 模块的调试级别。调试级别越高，在重新配置期间写入日志文件的调试消息就越多。您可以随时修改该属性。

`svm_abort_step_timeout`

键入整数，最小值为 30，最大值为 99999，默认值为 120。该属性为卷管理器重新配置框架的 Solaris Volume Manager for Sun Cluster 模块重新配置过程的中止步骤指定超时值（以秒为单位）。您可以随时修改该属性。

`svm_return_step_timeout`

键入整数，最小值为 30，最大值为 99999，默认值为 120。该属性为卷管理器重新配置框架的 Solaris Volume Manager for Sun Cluster 模块重新配置过程的返回步骤指定超时值（以秒为单位）。您可以随时修改该属性。

`svm_start_step_timeout`

键入整数，最小值为 30，最大值为 99999，默认值为 120。该属性为卷管理器重新配置框架的 Solaris Volume Manager for Sun Cluster 模块重新配置过程的启动步骤指定超时值（以秒为单位）。您可以随时修改该属性。

`svm_step1_timeout`

键入整数，最小值为 30，最大值为 99999，默认值为 120。该属性为卷管理器重新配置框架的 Solaris Volume Manager for Sun Cluster 模块重新配置过程的步骤 1 指定超时值（以秒为单位）。您可以随时修改该属性。

`svm_step2_timeout`

键入整数；最小值为 30；最大值为 99999；默认值为 120。该属性为卷管理器重新配置框架的 Solaris Volume Manager for Sun Cluster 模块重新配置过程的步骤 2 指定超时值（以秒为单位）。您可以随时修改该属性。

`svm_step3_timeout`

键入整数，最小值为 30，最大值为 99999，默认值为 120。该属性为卷管理器重新配置框架的 Solaris Volume Manager for Sun Cluster 模块重新配置过程的步骤 3 指定超时值（以秒为单位）。您可以随时修改该属性。

`svm_step4_timeout`

键入整数，最小值为 100，最大值为 99999，默认值为 120。该属性为卷管理器重新配置框架的 Solaris Volume Manager for Sun Cluster 模块重新配置过程的步骤 4 指定超时值（以秒为单位）。您可以随时修改该属性。

`svm_stop_step_timeout`

键入整数，最小值为 30，最大值为 99999，默认值为 120。该属性为卷管理器重新配置框架的 Solaris Volume Manager for Sun Cluster 模块重新配置过程的停止步骤指定超时值（以秒为单位）。您可以随时修改该属性。

例 415 创建 `vucmm_svm` 资源

此示例注册 `SUNW.vucmm_svm` 资源类型并创建 `SUNW.vucmm_svm` 资源类型的名为 `vucmm_svm-rs` 的实例。此示例假定已创建以下 Oracle Solaris Cluster 对象：

- 名为 `vucmm_framework-rg` 的资源组。
- 类型为 `SUNW.vucmm_framework` 的名为 `vucmm_framework-rs` 的资源

```
phys-schost-1# clresourcetype register SUNW.vucmm_svm
phys-schost-1# clresource create -g vucmm_framework-rg \
```

```
-t SUNW.vucmm_svm \  
-p resource_dependencies=vucmm_framework-rs vucmm_svm-rs
```

例 416 更改 vucmm_svm 资源的属性

此示例将卷管理器重新配置框架的 Solaris Volume Manager for Sun Cluster 组件重新配置过程的步骤 4 的超时值设置为 300 秒。示例假定已创建一个名为 vucmm_svm-rs 的 SUNW.vucmm_svm 资源类型实例。

```
phys-schost-1# clresource set \  
-p svm_step4_timeout=300 vucmm_svm-rs
```

有关以下属性的描述，请参见 [Unresolved link to "attributes5"](#)：

属性类型	属性值
可用性	ha-cluster/system/core

[clresource\(1CL\) \[225\]](#)、[clresourcetype\(1CL\) \[277\]](#)、[clsetup\(1CL\) \[407\]](#)、[SUNW.vucmm_framework\(5\) \[1231\]](#)
[link to "attributes5"](#)

OSC4 5cl

```
        <endpoint>

<-- Cluster Global Devices -->

<deviceList>
  <device>
    <devicePath>

<-- Cluster Quorum -->

<clusterQuorum>
  <quorumNodeList>
    <quorumNode>
  <quorumDeviceList>
    <quorumDevice>
      <quorumDevicePathList>
        <quorumDevicePath>

<-- Cluster Device Groups -->

<devicegroupList>
  <devicegroup>
    <memberDeviceList>
      <memberDevice>
    <devicegroupNodeList>
      <devicegroupNode>

<-- Cluster Resource Types -->

<resourcetypeList>
  <resourcetype>
    <resourcetypeRTRFile>
    <resourcetypeNodeList>
      <resourcetypeNode>
    <methodList>
      <method>
    <parameterList>
      <parameter>

<-- Cluster Resources -->

<resourceList>
  <resource>
    <resourceNodeList>
      <resourceNode>
    <monitoredState>

<-- Cluster Resource Groups -->

<resourcegroupList>
  <resourcegroup>
    <failoverMode>
    <managedState>
    <resourcegroupNodeList>
```

```

    <resourcegroupNode>
      <resourcegroupResourceList>
        <resourcegroupResource>

<-- Cluster NAS Devices -->

<nasdeviceList>
  <nasdevice>
    <nasdir>

<-- Cluster SNMP -->

<snmpmibList>
  <snmpmib>
<snmphostList>
  <snmphost>
<snmpuserList>
  <snmpuser>
<-- Cluster Telemetrics -->

<telemetrics>
  <telemetryObjectType>
    <telemetryAttribute>

```

元素

此部分列出并描述了群集 DTD 中定义的所有元素。如果元素具有必需的子项或属性，则必需项默认为一个。可选元素默认为零或一个。

<allNodes>

群集中所有成员节点的列表。<allNodes> 元素为通用元素。

<allNodes> 元素用于表示群集的所有节点。

父项： <resourcetypeNodeList>

子项： 无

属性： 无

<cluster>

完整群集配置 XML 文件的根元素。每个群集配置 XML 文件都必须以此元素（作为根元素）开始。DTD 只能接受一个 <cluster> 元素。群集配置 XML 文件中的后续 <cluster> 元素将被忽略。

父项： 无

子项： 可选：

- <propertyList>
- <nodeList>
- <clusterTransport>
- <deviceList>
- <clusterQuorum>
- <deviceGroupList>
- <resourcetypeList>
- <resourcegroupList>
- <resourceList>
- <nasdeviceList>
- <snmpmibList>
- <snmpHostList>
- <snmpuserList>

属性：

必需：

- name
群集的名称。

<clusterQuorum>

群集法定配置的根元素。所有群集法定信息在 <clusterQuorum> 元素的子元素中进行定义。

父项： <cluster>

子项： 可选：

- <quorumDeviceList>
- <quorumNodeList>

属性： 无

<clusterTransport>

群集传输配置的根元素。所有群集传输信息都显示在 <clusterTransport> 元素的子级别。

父项： <cluster>。

子项： 可选：

- <transportNodeList>
- <transportSwitchList>
- <transportCableList>

属性： 无

<device>

群集设备 ID 伪驱动程序 (DID) 设备。

父项： <deviceList>

子项： 可选：

- <devicePath> (零个或多个)

属性： 必需：

- ctd
UNIX 磁盘名称。
- name
设备的实例编号。

<devicegroup>

群集设备组实例的根元素。单个设备组的所有方面都在 <devicegroup> 元素的子元素中进行定义。

父项： <devicegroupList>

子项： 可选：

- <devicegroupNodeList>
- <memberDeviceList>
- <propertyList>

属性： 必需：

- name
设备组的名称。name 属性可以是任何有效的字符序列。
- type
设备组的类型。type 属性的值可以为 rawdisk、vxvm、svm 或 sds。

<devicegroupList>

所有群集设备组的列表。

父项： <cluster>

子项： 可选：

	<ul style="list-style-type: none"> ▪ <code><devicegroup></code> 一个 <code><devicegroup></code> 元素可用于群集中的每个设备组。
属性 :	无
<code><devicegroupNode></code>	设备组所在的节点。
父项 :	<code><devicegroupNodeList></code>
子项 :	无
属性 :	必需 : <ul style="list-style-type: none"> ▪ <code>nodeRef</code> 指定群集节点的名称。
<code><devicegroupNodeList></code>	设备组所在的节点的列表。
父项 :	<code><devicegroup></code> (1 个或多个)
子项 :	必需 : <ul style="list-style-type: none"> ▪ <code><devicegroupNode></code> (1 个或多个)
属性 :	无
<code><deviceList></code>	群集 DID 设备列表。
父项 :	<code><cluster></code>
子项 :	可选 : <ul style="list-style-type: none"> ▪ <code><device></code>
属性 :	固定 : <ul style="list-style-type: none"> ▪ <code>readonly</code> <code>readonly</code> 属性具有固定值 <code>true</code>。
<code><deviceNode></code>	存在特定 <code><device></code> 的节点和磁盘设备。
父项 :	<code><device></code>

子项：	无
属性：	必需： <ul style="list-style-type: none"> ■ nodeRef 存在实例的节点的名称。
<endpoint> 传输端点之一。	
父项：	<transportCable>
子项：	无
属性：	必需： <ul style="list-style-type: none"> ■ name 适配器或交换机的名称。 ■ nodeRef 托管指定适配器的节点的名称。仅当 type 属性设置为 adapter 时，才需要 nodeRef 属性。 ■ type type 属性可以设置为 adapter 或 switch。 如果 type 属性设置为 adapter，则必须指定 nodeRef 属性。 如果 type 属性设置为 switch，则可以指定 port 属性。但是，port 属性不是必需的。 可选： <ul style="list-style-type: none"> ■ port 交换机的端口号。仅当 type 属性设置为 switch 时，才指定 port 属性。
<failoverMode> 资源组的故障转移模式。	
父项：	<resourcegroup>
子项：	无
属性：	必需： <ul style="list-style-type: none"> ■ value value 属性可以设置为 failover 或 scalable。

`<managedState>`

指示资源组受管理还是不受管理。

父项： `<resourcegroup>`

子项： 无

属性： 必需：

- `value`

`value` 属性可以为 `true` 或 `false`。

`<memberDevice>`

特定设备组的成员名称。如果 `<devicegroup>` 是一组 `rawdisk` 类型，则必须指定一个或多个 `<member>` 元素，其中每个元素都带有原始磁盘路径的名称。

父项： `<memberDeviceList>`

子项： 无

属性： 必需：

- `name`

成员的名称。

`<memberDeviceList>`

设备组成员列表。

父项： `<devicegroup>` (一个或多个)

子项： 必需：

- `<memberDevice>`

属性： 无

`<method>`

通用方法类型与特定资源类型的实际方法名称之间的映射。

父项： `<methodList>`

子项： 无

属性： 必需：

- `name`

资源类型的方法的实际名称。

-
- type
资源类型的方法的类型。可以指定以下类型：
 - MONITOR_CHECK
 - MONITOR_START
 - MONITOR_STOP
 - PRENET_START
 - START
 - STOP
 - VALIDATE
 - UPDATE

<methodList>

适用于特定 <resourcetype> 的所有 <method> 元素的列表。

父项： <resourcetype>

子项： 可选：

- <method>

属性： 固定：

- readonly
readonly 属性的固定值为 true。

<monitoredState>

指示群集中元素状态的一部分的布尔值。例如，资源的 <monitoredState> 指定资源是否受监视，但不指定资源是否可用。

父项： <resource>

子项： 无

属性： 必需：

- value
value 属性可以设置为 true 或 false。

<nasdevice>

群集上 NAS 设备的单个实例。

父项： <nasdeviceList>

子项： 可选：

属性：

- 必需：
 - `<nasdir>`
- `name`
NAS 设备的主机名。
- `type`
NAS 设备的类型。必须为 Oracle ZFS Storage Appliance 指定 `sun_uss`。

可选：

- `userid`
访问 NAS 设备所必需的用户名。

`<nasdeviceList>`

群集中所有 NAS 设备的列表。

父项： `<cluster>`

子项： 可选：

- `<nasdevice>`

属性： 无

`<nasdir>`

NAS 设备的一个目录。每个 NAS 设备可以具有多个 NAS 目录。

父项： `<nasdevice>`

子项： 无

属性： 必需：

- `path`
NAS 目录的路径。

`<node>`

群集节点。为群集中的每个节点指定一个 `<node>` 元素。

父项： `<nodeList>`

子项： 可选：

- `<propertyList>`

属性： 必需：

- name
必须等同于节点名称。

可选：

- id
群集节点 ID。如果未指定，则为群集节点 ID 属性提供空字符串默认值。

<nodeList>

群集中所有节点的列表。

父项： <cluster>

子项： 可选：

- <node>
必须为群集上的每个节点至少提供一个 node 属性。

属性： 无

<parameter>

描述 <method> 元素超时值和群集资源类型的其他参数的一组属性。

父项： <parameterList>

子项： 无

属性： 必需：

- extension
extension 属性可以设置为 true 或 false。
- name
参数的名称。
- tunability
参数可调性的值。tunability 属性可以设置为以下值之一：atCreation、anyTime 或 whenDisabled。
- type
参数类型。type 属性可以设置为以下值之一：boolean、enum、int、string 或 stringArray。

可选：

- default
如果未显式指定一个值，则为该参数的默认值。例如，method 元素超时的默认值为 START。
- description
参数的描述。如果未定义，则该属性默认为空字符串。
- enumList
对象的枚举列表。例如，属性可能为按照优先级顺序排列的故障转移模式的列表。
- maxLength
string 或 stringArray 类型参数的最大长度。
- minArrayLength
stringArray 类型参数的最小大小。
- minLength
string 或 stringArray 类型参数的最小长度。

<parameterList>

描述资源类型的 <parameter> 元素的列表。

父项： <resourcetype>

子项： 可选：
▪ <parameter>

属性： 固定：
▪ readonly
readonly 属性具有固定值 true。

<property>

描述一个属性的通用元素。该属性不特定于与群集相关的配置的任何子集。

父项： <propertyList>

子项： 无

属性： 必需：
▪ name
属性的名称。
▪ value
属性的值。

可选：

- readonly
readonly 属性可以设置为 true 或 false。如果未指定该值，则属性默认为值 false。
- type
属性类型。

<propertyList>

<property> 元素的列表。 <propertyList> 元素为通用元素。

父项： <cluster>, <deviceGroup>, <node>, <quorumDevice>, <quorumNode>, <resource>, <resourceNode> , <resourcegroup>, <resourceType>, <transportAdapter>, <transportType>

子项： 可选：

- <property>

属性： 可选：

- extension
该属性可以具有以下值之一：
— true、 false、 mixed 或 doesNotApply。
如果未指定值，则 extension 属性具有默认值 doesNotApply。
- readonly
该属性可以具有值 true 或 false。如果未指定值，则 readonly 属性具有默认值 false。

<quorumDevice>

单个群集法定设备。

父项： <quorumDeviceList>

子项： 可选：

- <propertyList>
<quorumDevice> 元素只能有一个 <propertyList> 子项。
- <quorumDevicePathList>
<quorumDevice> 元素只能有一个 <quorumDevicePathList> 子项。

<quorumNode>

参与群集法定的群集中的节点。

- 父项： <quorumNodeList>
- 子项： 可选：
- <propertyList>
- 属性： 必需：
- <nodeRef>
节点名称。

<quorumNodeList>

参与群集法定的所有节点的列表。在不处于 `installmode` 的功能群集中，该列表通常包含群集中的所有节点。在仍处于 `installmode` 的群集中，该列表可能只包含群集节点之一。

- 父项： <clusterQuorum>
- 子项： 必需：
- <quorumNode>
- 属性： 固定：
- `readonly`
`readonly` 属性设置为 `true`。

<resource>

群集资源。

- 父项： <resourceList>
- 子项： 可选：
- <resourceNodeList>
 - <propertyList>
- 属性： 必需：
- `name`
资源的名称。
 - `resourcegroupRef`
资源所属的资源组。
 - `resourcetypeRef`

该元素描述的资源的类型。

`<resourceList>`

在配置中定义的群集资源的根节点列表。

父项：`<cluster>`

子项：可选：
■ `<resource>`

属性：无

`<resourcegroup>`

群集资源组。

父项：`<resourcegroupList>`

子项：必需：
■ `<failoverMode>`
■ `<managedState>`
■ `<resourcegroupNodeList>`
■ `<resourcegroupResourceList>`
■ `<propertyList>`

属性：必需：
■ `name`
资源的名称。

`<resourcegroupList>`

在配置中定义的群集资源组的根节点。

父项：`<cluster>`

子项：可选：
■ `<resourcegroup>`

属性：无

`<resourcegroupNode>`

在其上定义资源组的节点。

父项：`<resourcegroupNodeList>`

子项： 无

属性： 必需：

- nodeRef
群集节点的名称。

可选：

- Zone
区域的名称。

<resourcegroupNodeList>
运行特定资源组的群集节点。

父项： <resourcegroup>

子项： 必需：

- <resourcegroupNode>

属性： 无

<resourcegroupResource>
属于特定资源组的群集资源。

父项： <resourcegroupResourceList>

子项： 无

属性： 必需：

- resourceRef
资源的名称。

<resourcegroupResourceList>
在资源组中定义的资源的列表。

父项： <resourcegroup>

子项： 可选：

- <resourcegroupResource>

属性： 无

<resourceNode>
在其上定义资源的节点。

父项：	<resourceNodeList>
子项：	必需： <ul style="list-style-type: none"> ■ <state> ■ <monitoredState> 可选： <ul style="list-style-type: none"> ■ <propertyList>
属性：	必需： <ul style="list-style-type: none"> ■ nodeRef 资源类型的名称 可选： <ul style="list-style-type: none"> ■ zone 区域的名称。

<resourcetype>

在群集中可用的群集资源类型。

父项：	<resourcetypeList>
子项：	可选： <ul style="list-style-type: none"> ■ <resourcetypeRTRFile> ■ <resourcetypeNodeList> ■ <methodList> ■ <parameterList> ■ <propertyList>
属性：	必需： <ul style="list-style-type: none"> ■ name 资源类型的名称

<resourcetypeList>

在配置中定义的群集资源类型的根节点。

父项：	<cluster>
子项：	可选： <ul style="list-style-type: none"> ■ <resourcetype>
属性：	无

<resourcetypeNode>

在其上定义资源类型的节点。

父项： <resourcetypeNodeList>

子项： 无

属性： 必需：
■ nodeRef
群集节点的名称。

<resourcetypeNodeList>

存在特定资源类型的群集节点的列表。

父项： <resourcetype>

子项： 必需：<resourcetypeNodeList> 元素必须包含一个或多个 <resourcetypeNode> 元素，或只包含一个 <allNodes> 元素。

- <resourcetypeNode>
- <allNodes>

属性： 无

<resourcetypeRTRFile>

描述特定资源类型的资源类型注册 (resource type registration, RTR) 文件的名称。

父项： <resourcetype>

子项： 无

属性： 必需：
■ name
RTR 文件的名称。

<snmpHost>

在群集节点上配置的 SNMP 主机和团体。

父项： <snmpHostList>

子项： 无

属性： 必需：

- community
SNMP 团体名称。
- name
实例名称。
- nodeRef
SNMP 主机和团体所在的节点。

`<snmpHostList>`

在某个群集节点上配置的 SNMP 主机和团体的列表。

父项：`cluster>`

子项：可选：
▪ `<snmpHost>`

属性：无

`<snmpMib>`

位于群集节点上的 SNMP MIB。

父项：`<snmpMibList>`

子项：可选：
▪ `state`

属性：必需：
▪ `name`
MIB 名称。
▪ `nodeRef`
SNMP MIB 所在的节点。

可选：
▪ `protocol`
MIB 将使用的 SNMP 协议。该属性默认为 SNMPv2。
▪ `value`
SNMPv3 或 SNMPv2

`snmpMibList`

位于某个群集节点上的 SNMP MIB 的列表。

父项 : <cluster>

子项 : 可选 :
▪ <snmpmib>

属性 : 无

<snmpuser>

在群集节点上配置的 SNMPv3 用户。

父项 : <snmpuserList>

子项 : 无

属性 : 必需 :
▪ name
 用户的名称。
▪ nodeRef
 SNMPv3 用户所在的节点。
▪ auth
 auth 属性可设置为 MD5 或 SHA。

可选 :

- defaultUser
 defaultUser 属性可以设置为 yes 或 no。如果未指定值，该属性默认为任何合适的值，具体取决于节点配置。
- defaultSecurityLevel
 用户的安全性级别。security 属性可设置为以下值之一：
 - authPriv
 - authNoPriv
 - noAuthNoPriv

<snmpuserList>

在某个群集节点上配置的 SNMPv3 用户的列表。

父项 : <cluster>

子项 : <snmpuser>

属性 : 无

<state>

群集配置中各种对象的状态。<state> 元素为通用元素。

父项： <quorumDevicePath>, <resourceNode>, snmpmib,
telemetryAttribute, <transportAdapter>,
<transportCable>, <transportSwitch>

子项： 无

属性： 必需：
■ value
value 属性可以设置为 enabled 或 disabled。

<telemetrics>

群集监视阈值

父项： <cluster>

子项： 可选：
■ <telemetryObjectType>

属性： 无

<telemetryAttribute>

您可以监视的系统资源的属性。

父项： <telemetryObjectType>

子项： 必需：
■ <state> (1 个或多个)

属性： 必需：
■ name
属性的名称。

<telemetryObjectType>

您可以监视的对象的类型。

父项： <telemetrics>

子项： 必需：
■ <telemetryAttribute>

-
- <transportCable>

属性： 无

<transportNode>

用于专用群集传输的群集节点之一。为群集的每个节点指定一个 <transportNode> 元素。

父项： <transportNodeList>

子项： 可选：

- <nodeRef>

属性： 必需：

- transportAdapterList
群集节点的名称。

<transportNodeList>

用于专用群集传输的节点的列表。此节点列表始终包含与群集成员相同的一组节点。

父项： <clusterTransport>

子项： 可选：

- <transportNode>

属性： 无

<transportSwitch>

群集传输交换机。

父项： <transportSwitchList>

子项： 可选：

- <state>

属性： 必需：

- name
传输交换机的名称。

可选：

- port

交换机的端口号。

<transportSwitchList>

专用群集传输系统使用的网络交换机的列表。

父项 : <clusterTransport>

子项 : 可选 :
▪ <transportSwitch>

属性 : 无

<transportType>

用于 <transportAdapter> 元素的网络传输的类型。

父项 : <transportAdapter>

子项 : 可选 :
▪ <propertyList>

属性 : 必需 :
▪ value
value 属性可以设置为 dlpi 或 rsm。

/usr/cluster/lib/xml/cluster.dtd

定义 Oracle Solaris Cluster 配置 XML 文件结构的文档类型定义 (document type definition, DTD) 文件。

[Intro\(1CL\) \[17\]](#)、[cluster\(1CL\) \[461\]](#)

<http://www.w3.org/XML/>

OSC4 7

名称

clprivnet — SUNW, clprivnet Oracle Solaris Cluster 专用网络驱动程序

/dev/clprivnet

注 - Oracle Solaris Cluster 软件包括面向对象的命令集。虽然 Oracle Solaris Cluster 软件仍支持原始的命令集，但是 Oracle Solaris Cluster 过程文档仅使用面向对象的命令集。有关面向对象的命令集的更多信息，请参见 [Intro\(1CL\) \[17\]](#) 手册页。

SUNW,clprivnet Oracle Solaris Cluster 专用网络驱动程序是一个 STREAMS 伪驱动程序，用于支持使用标准 Solaris 接口通过 Oracle Solaris Cluster 专用网络进行通信的 Oracle Solaris Cluster 驻留应用程序。通过将数据流量分流到所有链路上，该驱动程序可以在支持具有高可用性的软件容错通信的同时最大限度地利用专用网络的带宽。

应用程序编程接口

该驱动程序受特殊字符设备 /dev/clprivnet 支持，但保留用于 Oracle Solaris Cluster 内部操作和标准 Solaris 网络实用程序。不得将该设备接口直接用于一般的应用程序通信。

管理

将该驱动程序作为网络接口进行管理和配置的工作完全由 Oracle Solaris Cluster 基础架构内部操作完成。

/dev/clprivnet

clprivnet 特殊字符设备

/usr/kernel/drv/clprivnet.conf

系统范围内的默认设备驱动程序属性

名称

did — 用户可配置的 DID 伪驱动程序

注 - Oracle Solaris Cluster 软件包括面向对象的命令集。虽然 Oracle Solaris Cluster 软件仍支持原始的命令集，但是 Oracle Solaris Cluster 过程文档仅使用面向对象的命令集。有关面向对象的命令集的更多信息，请参见 [Intro\(1CL\) \[17\]](#) 手册页。

DID 是用户可配置的伪设备驱动程序，提供对基础磁盘、磁带和 CDROM 设备的访问权限。设备支持唯一的设备 ID 时，到设备的多个路径是根据该设备的设备 ID 确定的。即使多个路径可用于同一个设备 ID，也只能将一个 DID 名称指定给实际设备。

在群集环境中，无论特定物理设备与多个主机或控制器的连接如何，它都具有相同的 DID 名称。但这仅适用于支持全局唯一设备标识符的设备（如物理磁盘）。

对于 DID 在 `/dev/did` 下管理的每种类型的设备，DID 都为其维护并行目录。这些目录中的设备的行为与相应的非 DID 对象的行为相同。其中包括维护磁盘和 CD-ROM 设备的分片以及不同磁带设备行为的名称。通过 `/dev/did/dsk` 和 `/dev/did/rdsk` 还支持磁盘的原始设备和块设备访问权限。

在任何时间点，I/O 仅支持一个设备路径。当前不通过 DID 提供任何多路径支持。

在可以使用 DID 设备之前，该设备必须先通过 [scdidadm\(1M\) \[677\]](#) 命令进行初始化。

DID 驱动程序维护管理节点以及每个 DID 次要设备的节点。

管理节点不支持任何用户 IOCTL。

针对 DID 设备节点（如 `/dev/did/rdsk/d0s2`）调用 `DKIOCINFO` ioctl 时才支持该 ioctl。

所有其他 IOCTL 已直接传递给以下驱动程序。

`/dev/did/dsk/dns m`

块磁盘或 CD-ROM 设备，其中 *n* 是设备编号，*m* 是分片编号

`/dev/did/rdsk/dns m`

原始磁盘或 CD-ROM 设备，其中 *n* 是设备编号，*m* 是分片编号

`/dev/did/rmt/n`

磁带设备，其中 *n* 是设备编号

`/dev/did/admin`

管理设备

/kernel/drv/did

驱动程序模块

/kernel/drv/did.conf

驱动程序配置文件

/etc/did.conf

非群集系统的 cldevice 配置文件

Cluster Configuration Repository (CCR) 文件

[cldevice\(1CL\) \[51\]](#) 维护群集系统 CCR 中的配置

[Intro\(1CL\) \[17\]](#)、[cldevice\(1CL\) \[51\]](#)、[Unresolved link to " devfsadm1M"](#)

为了能够减少在设备热插拔期间的开销，DID 为组中的设备创建名称。对于磁盘，设备名称是在 /dev/did/dsk 和 /dev/did/rdisk 中创建的，一次 100 个磁盘（作为一组）。对于磁带，设备名称是在 /dev/did/rmt 中创建的，一次 10 个磁盘（作为一组）。如果添加到群集中的设备比当前名称处理的设备多，则将创建另一个组。

OSC4 7p

名称

sctransp_dlp_i — 配置 dlp_i 群集互连

注 - Oracle Solaris Cluster 软件包括面向对象的命令集。虽然 Oracle Solaris Cluster 软件仍支持原始的命令集，但是 Oracle Solaris Cluster 过程文档仅使用面向对象的命令集。有关面向对象的命令集的更多信息，请参见 [Intro\(1CL\) \[17\]](#) 手册页。

dlp_i 是支持的群集传输类型。

[Intro\(1CL\) \[17\]](#)、[cluster\(1CL\) \[461\]](#)、[sconf\(1M\) \[637\]](#)、[scinstall\(1M\) \[697\]](#)

索引

B

保留步骤超时, 1127, 1213, 1231, 1237
本地群集节点名称访问函数, 923

C

查询 DCS, 617
超时
 Solaris Volume Manager for Sun
 Cluster, 1233, 1239
 保留步骤, 1127, 1213, 1231, 1237
超时时运行子程序, 623
初始化 Oracle Solaris Cluster 软件并建立新的群集节点, 697
初始化系统资源监视, 763
创建 Oracle Solaris Cluster 资源类型模板, 559
磁盘路径监视管理命令, 685
磁盘路径监视守护进程配置文件, 1075
从错误代码生成错误字符串, 779, 779, 781, 781
CCR 表文件管理命令, 607
ccradm, 607
cl_eventd, 613
cl_pnmd, 615
claccess, 31
clconfiguration, 1245
cldev, 37, 51
cldevice, 37, 51
cldevicegroup, 65, 81
clidg, 65, 81
clinterconnect, 97, 107
clintr, 97, 107
clmib, 117, 417
clnas, 125, 137
clnasdevice, 125, 137
clnode, 149

clprivnet, 1273
clpstring 命令, 167, 175
clq, 183, 195
clquorum, 183, 195
clreslogicalhostname, 207, 359
clresource 命令, 225, 331
clresourcegroup, 253, 307
clresourcetype 命令, 277, 395
clressharedaddress, 289, 377
clrg, 253, 307
clrs 命令, 167, 175, 225, 331
clrslh, 207, 359
clrssa, 289, 377
clrt 命令, 277, 395
clsetup, 407
clsnmphot, 409
clsnmpmib, 117, 417
clsnmpuser, 425
clta, 433, 447
cltelemetryattribute, 433, 447
cluster, 461
clzc, 491, 517
clzonecluster, 491, 517
commandlog, 1065
crs_framework 资源类型, 1081, 1171

D

代理多主节点 SMF 服务的资源类型, 1095, 1205
代理故障转移 SMF 服务的资源类型, 1091, 1201
代理可伸缩 SMF 服务的资源类型, 1099, 1209
等待故障监视器控制套接字上发送的消息, 797
等候指定的超时时间, 以待受监视进程终止, 887
对资源组进行故障转移, 783
dcs_config, 617
Debug_level 扩展属性

- ScalDeviceGroup 资源类型, 1157, 1217
 - ScalMountPoint 资源类型, 1163, 1223
 - SUNW.ScalDeviceGroup 资源类型, 1157, 1217
 - SUNW.ScalMountPoint 资源类型, 1163, 1223
 - debug_level 扩展属性
 - SUNW.vucmm_svm 资源类型, 1233, 1239
 - vucmm_svm 资源类型, 1233, 1239
 - derby
 - Java DB 数据库的资源类型实现, 1085, 1175
 - did, 1275
 - DiskGroupName 扩展属性, 1157, 1217
 - DSDL (Data Service Development Library, 数据服务开发库)
 - 实用程序函数, 775
 - 属性函数, 773
 - 故障监视, 773
 - 故障监视器函数, 774
 - 网络资源访问函数, 773
 - 进程监视器工具 (Process Monitor Facility, PMF) 函数, 774
- F**
- 返回表明是否已在区域群集节点中配置资源的 TRUE 或 FALSE, 849
 - 返回步骤超时
 - Solaris Volume Manager for Sun Cluster, 1234, 1240
 - 返回执行资源组的群集节点名称, 951
 - 访问群集信息, 567
 - 访问资源信息, 585
 - 访问资源组信息, 597
 - 分配和初始化 DSDL 环境, 847
 - FileSystemType 扩展属性, 1163, 1223
- G**
- 更改 Solaris Volume Manager 设备组配置。 , 659
 - 更新 Oracle Solaris Cluster 软件配置, 637
 - 公共网络管理 (Public Network Management, PNM) 服务守护进程, 615
 - 共享对象提供逻辑主机名而不是物理主机名, 25
 - 故障监视器
 - 函数, DSDL, 774
 - 挂载点
 - 可伸缩
 - 资源类型, 1161, 1221
- H**
- 关闭群集, 743
 - 管理
 - 资源, 167, 175, 225, 331
 - 资源类型, 277, 395
 - 管理 Oracle Solaris Cluster 法定, 183, 183, 195, 195
 - 管理 Oracle Solaris Cluster 共享地址的资源, 289, 289, 377, 377
 - 管理 Oracle Solaris Cluster 互连, 97, 97, 107, 107
 - 管理 Oracle Solaris Cluster 节点, 149
 - 管理 Oracle Solaris Cluster 逻辑主机名的资源, 207, 207, 359, 359
 - 管理 Oracle Solaris Cluster 区域群集节点, 491, 491, 517, 517
 - 管理 Oracle Solaris Cluster 设备, 37, 37, 51, 51
 - 管理 Oracle Solaris Cluster 设备组, 65, 65, 81, 81
 - 管理 Oracle Solaris Cluster 事件管理信息库 (Management Information Base, MIB) 模块, 689
 - 管理 Oracle Solaris Cluster 数据服务的资源组, 253, 253, 307, 307
 - 管理 Oracle Solaris Cluster SNMP 用户, 425
 - 管理 Oracle Solaris Cluster SNMP 主机, 409
 - 管理 Oracle Solaris Cluster SNMP MIB, 117, 117, 417, 417
 - 管理群集的全局配置和状态, 461
 - 管理添加节点的 Oracle Solaris Cluster 访问策略, 31
 - 管理专用 IP 地址范围, 725
 - 管理资源类型、资源组和资源的注册和取消注册。 , 731
- H**
- 函数
 - DSDL 实用程序, 775
 - DSDL 属性, 773
 - DSDL 故障监视器, 774
 - DSDL 网络资源访问, 773
 - DSDL 进程监视器工具 (Process Monitor Facility, PMF), 774
 - 获取某一资源所使用的网络资源, 839

获取有关由资源使用的 SUNW.HAStoragePlus 资源的状态信息, 843
 获取资源所用的网络地址, 825
 获取资源组中使用的网络资源, 835, 837
 halockrun, 621
 hatimerun, 623

I

intro, 17
 Intro, 17
 IOTimeout 扩展属性, 1158, 1164, 1218, 1224

J

基于 RPC 的进程监视器服务器, 631, 631, 633, 633
 基于错误代码生成错误消息, 1057, 1059
 基于错误代码生成错误字符串, 1057, 1059
 监视
 Oracle Solaris Cluster Support for Oracle RAC, 1127, 1213
 卷管理器重新配置框架, 1231, 1237
 监视 Oracle Solaris Cluster 配置的状态, 745
 检索扩展属性, 817
 检索区域的全名, 821, 823
 检索区域全名的节点 ID, 821, 823
 检索通过其调用数据服务方法的路径名称的最后一个元素, 815
 检索运行某个方法时所代表的区域的名称, 841
 检索资源类型名称, 833
 检索资源名称, 831
 检索资源所用的端口列表, 827
 检索资源组名称, 829
 建立与应用程序的 TCP 连接, 789, 799
 将调试消息写入系统日志, 893
 将消息写入系统日志, 891
 将信号发送到由 PMF 控制的进程树, 855
 交互式群集配置工具, 741
 进程监视器工具管理, 625
 卷, 1158, 1218
 卷管理器重新配置
 框架, 1231, 1237
 卷管理器重新配置框架
 Solaris Volume Manager for Sun Cluster, 1233, 1239

状态信息, 1231, 1237
 监视, 1231, 1237
 资源类型
 SUNW.vucmm_framework, 1231, 1237
 SUNW.vucmm_svm, 1233, 1239
 vucmm_framework, 1231, 1237
 vucmm_svm, 1233, 1239

Java DB 数据库

资源类型实现, 1085, 1175

K

可伸缩挂载点
 资源类型, 1161, 1221
 可伸缩设备组
 资源类型, 1155, 1215
 可伸缩文件系统
 资源类型, 1161, 1221
 框架
 Oracle Clusterware, 1081, 1171
 Oracle Solaris Cluster Support for Oracle RAC, 1127, 1213
 卷管理器重新配置, 1231, 1237
 扩展属性
 crs_framework 资源类型, 1083, 1173
 rac_framework 资源类型, 1127, 1213
 SUNW.crs_framework 资源类型, 1083, 1173
 SUNW.rac_framework 资源类型, 1127, 1213
 SUNW.vucmm_framework 资源类型, 1231, 1237
 SUNW.vucmm_svm 资源类型, 1233, 1239
 vucmm_framework 资源类型, 1231, 1237
 vucmm_svm 资源类型, 1233, 1239

L

libscho.st.so.1, 25
 LogicalDeviceList 扩展属性, 1158, 1218

M

命令日志文件 commandlog, 1065
 Monitor_retry_count 扩展属性
 ScalDeviceGroup 资源类型, 1159, 1219
 ScalMountPoint 资源类型, 1164, 1224

SUNW.ScalDeviceGroup 资源类型, 1159, 1219
 SUNW.ScalMountPoint 资源类型, 1164, 1224
 Monitor_retry_interval 扩展属性
 ScalDeviceGroup 资源类型, 1159, 1219
 ScalMountPoint 资源类型, 1164, 1224
 SUNW.ScalDeviceGroup 资源类型, 1159, 1219
 SUNW.ScalMountPoint 资源类型, 1164, 1224
 MountOptions 扩展属性, 1165, 1225
 MountPointDir 扩展属性, 1165, 1225

N

NAS (network-attached storage, 网络连接的存储) 设备, 1161, 1221

O

Oracle Clusterware
 框架, 1081, 1081, 1171, 1171
 Oracle RAC (Real Application Clusters) 见 Sun Cluster Support for Oracle RAC
 Oracle Real Application Clusters (RAC) 见 Sun Cluster Support for Oracle RAC
 Oracle Solaris Cluster 版本管理, 767
 Oracle Solaris Cluster 数据服务开发库 (Data Services Development Library, DSDL) 函数, 771
 Oracle Solaris Cluster 维护命令的简介, 17, 17
 Oracle Solaris Cluster 系统配置文件, 1245
 Oracle Solaris Cluster 专用 IP 地址服务守护进程, 635, 729
 Oracle Solaris Cluster Support for Oracle RAC
 Oracle Clusterware 框架, 1081, 1171
 框架, 1127, 1213
 状态信息, 1127, 1213
 监视, 1127, 1213
 资源类型
 crs_framework, 1081, 1171
 rac_framework, 1127, 1213
 SUNW.crs_framework, 1081, 1171
 SUNW.rac_framework, 1127, 1213

P

配置 dlpi 群集互连, 1279
 配置 InfiniBand 群集传输交换机, 675

配置 Intel PRO/1000 网络适配器, 671
 配置系统资源监视, 433, 433, 447, 447
 配置以太网群集传输交换机, 673
 配置资源类型模板, 555
 PMF (Process Monitor Facility, 进程监视器工具)
 函数, DSDL, 774
 pmfadm, 625
 pmfd, 631, 633

Q

启动 GUI 版本的 Oracle Solaris Cluster Agent Builder, 553
 启动步骤超时
 Solaris Volume Manager for Sun Cluster, 1234, 1234, 1240, 1240
 强制 Oracle Solaris Cluster 设备服务、文件系统与数据服务之间依赖性的资源类型, 1193
 请求资源和资源组控制, 579
 区域名称访问函数, 925
 区域群集
 管理, 491, 517
 全局设备名称空间管理脚本, 695
 确定是否存在受 PMF 监视的进程树, 851
 群集, 1063
 群集名称数据库, 1063
 群集日志设备访问, 921
 群集事件守护进程, 613
 群集信息访问函数。
 , 903, 903, 903, 903, 903, 909, 909, 909, 909, 909, 909
 群集重新配置通知协议 (Cluster Reconfiguration Notification Protocol, CRNP) 的资源类型实现, 1177, 1177
 QFS 共享文件系统 见 Sun QFS 共享文件系统

R

r_properties, 1103
 RAC (Real Application Clusters) 见 Oracle Solaris Cluster Support for Oracle RAC
 rac_framework 资源类型, 1127, 1213
 Real Application Clusters (RAC) 见 Oracle Solaris Cluster Support for Oracle RAC
 RebootOnFailure 扩展属性, 1159, 1166, 1219, 1226

reservation_timeout 扩展属性, 1127, 1213, 1231, 1237
 rg_properties, 1129
 rpc.pmf, 631, 633
 rt_callbacks, 545
 rt_properties, 1143
 rt_reg, 1067

S

设备标识符配置和管理实用程序包装, 677
 设备组
 可伸缩
 资源类型, 1155, 1215
 设置资源状态, 593
 设置资源状态函数, 1003, 1003, 1005, 1005
 实现回调方法和监视资源类型过程中所用的 Oracle Solaris Cluster 库函数, 897, 897, 897
 实用程序函数
 DSDL, 775
 使用 PMF 重新启动故障监视器, 853
 使用应用程序的 TCP 连接读取数据, 803
 使用应用程序的 TCP 连接写入数据, 805
 事件, 1177
 释放 DSDL 环境资源, 777
 释放端口列表内存, 813
 释放网络地址内存, 811
 释放网络资源内存, 809
 释放资源扩展属性内存, 807
 sc_zonesd, 635
 sc_zonesd - Oracle Solaris Cluster 区域管理守护进程, 635
 ScalDeviceGroup 资源类型, 1155, 1215
 ScalMountPoint 资源类型, 1161, 1221
 scconf, 637
 scconf_dg_rawdisk, 655
 scconf_dg_svm, 659
 scconf_transp_adap_e1000g, 671
 scconf_transp_jct_etherswitch, 673
 scconf_transp_jct_ibswitch, 675
 scdidadm, 677
 scdpm, 685
 scdpmd.conf, 1075
 scds_calls, 771
 scds_close, 777
 scds_error_string, 779, 781

scds_error_string_i18n, 779, 781
 scds_failover_rg, 783
 scds_fm_action, 785
 scds_fm_net_connect, 789
 scds_fm_net_disconnect, 793
 scds_fm_sleep, 797
 scds_fm_tcp_connect, 799
 scds_fm_tcp_disconnect, 801
 scds_fm_tcp_read, 803
 scds_fm_tcp_write, 805
 scds_free_ext_property, 807
 scds_free_net_list, 809
 scds_free_netaddr_list, 811
 scds_free_port_list, 813
 scds_get_current_method_name, 815
 scds_get_ext_property, 817
 scds_get_fullname, 821, 823
 scds_get_fullname_nodeid, 821, 823
 scds_get_netaddr_list, 825
 scds_get_port_list, 827
 scds_get_resource_group_name, 829
 scds_get_resource_name, 831
 scds_get_resource_type_name, 833
 scds_get_rg_hostnames, 835, 837
 scds_get_rs_hostnames, 839
 scds_get_zone_name, 841
 scds_hasp_check, 843
 scds_initialize, 847
 scds_is_zone_cluster, 849
 scds_pmf_get_status, 851
 scds_pmf_restart_fm, 853
 scds_pmf_signal, 855
 scds_pmf_start, 857, 859
 scds_pmf_stop, 861
 scds_pmf_stop_monitoring, 863
 scds_print_net_list, 865
 scds_print_netaddr_list, 867
 scds_print_port_list, 869
 scds_restart_resource, 879
 scds_restart_rg, 881
 scds_simple_net_probe, 883
 scds_simple_probe, 885
 scds_svc_wait, 887
 scds_syslog, 891
 scds_syslog_debug, 893
 scds_timerun, 895

T

探测完成后采取操作函数, 785
 添加、更改或更新原始磁盘设备组配置, 655
 停止监视在 PMF 控制下运行的进程, 863
 通过建立和终止与应用程序的 TCP 连接来进行探测, 883, 885
 TargetFileSystem 扩展属性, 1166, 1226
 TCP 连接
 使用 DSDL 故障监视, 773

V

vucmm_framework 资源类型, 1231, 1237
 vucmm_svm 资源类型, 1233, 1239

W

网络连接的存储 (network-attached storage, NAS) 设备, 1161, 1221
 网络资源访问函数
 DSDL, 773
 为 Oracle Solaris Cluster 管理对 NAS 设备的访问权限, 125, 125, 137, 137
 文件系统
 可伸缩
 资源类型, 1161, 1221

X

系统资源使用情况
 用于数据收集的资源类型, 1169, 1229
 显示端口列表的内容, 869
 显示网络资源列表的内容, 865
 显示资源组使用的主机名-端口-协议三元组列表的内容, 867

Y

以交互方式配置 Oracle Solaris Cluster, 407
 用户可配置磁盘 ID 驱动程序, 1275
 用户权限, 563
 用于管理作为 Oracle Solaris Cluster 资源的服务的回调接口, 545
 用于数据收集的资源类型

系统资源使用情况, 1169, 1229

Z

在 Oracle Solaris Cluster 配置中执行资源组和设备组的所有权/状态更改, 751
 在持有文件锁时运行子程序, 621
 在给定时段内执行给定命令, 895
 执行 PMF 控制下的程序, 857, 859
 中止步骤超时
 Solaris Volume Manager for Sun Cluster, 1233, 1239
 终止与应用程序的 TCP 连接, 793, 801
 终止在 PMF 控制下运行的进程, 861
 重新配置超时
 Solaris Volume Manager for Sun Cluster, 1233, 1239
 保留步骤, 1127, 1213, 1231, 1237
 重新启动资源, 879
 重新启动资源组, 881
 状态信息
 Oracle Solaris Cluster Support for Oracle RAC, 1127, 1213
 卷管理器重新配置框架, 1231, 1237
 资源
 管理, 167, 175, 225, 331
 资源和资源组控制请求函数, 939, 939, 945, 945
 资源类型, 1183
 crs_framework, 1081, 1171
 rac_framework, 1127, 1213
 ScalDeviceGroup, 1155, 1215
 ScalMountPoint, 1161, 1221
 SUNW.crs_framework, 1081, 1171
 SUNW.rac_framework, 1127, 1213
 SUNW.ScalDeviceGroup, 1155, 1215
 SUNW.ScalMountPoint, 1161, 1221
 SUNW.vucmm_framework, 1231, 1237
 SUNW.vucmm_svm, 1233, 1239
 vucmm_framework, 1231, 1237
 vucmm_svm, 1233, 1239
 管理, 277, 395
 资源类型属性, 1143
 资源类型信息访问函数, 1037, 1037, 1037, 1037, 1037, 1041, 1041, 1041, 1041,

