



APUSIC  
固若长城  
睿比世界

# 命令行操作指南

金蝶Apusic应用服务器V10

版权所有 © 深圳市金蝶天燕云计算股份有限公司2026。保留所有权利。

## 版权声明

本档所涉及的软件著作权、版权等知识产权已依法进行了注册，由金蝶天燕云计算股份有限公司合法拥有。受《中华人民共和国著作权法》《计算机软件保护条例》《知识产权保护条例》和相关国际版权条约、法律、法规以及其它知识产权法律和条约的保护。未经授权许可，不得非法使用。

## 免责声明

本档包含的版权信息由金蝶天燕云计算股份有限公司合法拥有，受法律的保护，金蝶天燕云计算股份有限公司对本档可能涉及到的非金蝶天燕云计算股份有限公司的信息不承担任何责任。在法律允许的范围内，您可以查阅并仅能够在《中华人民共和国著作权法》规定的合法范围内复制和打印本档。任何单位和个人未经金蝶天燕云计算股份有限公司书面授权许可，不得使用、修改、再发布本档的任何部分和内容，否则将被视为侵权，金蝶天燕云计算股份有限公司有依法追究其责任的权利。

本档如有更新，不另行通知。对本档中的问题您可向金蝶天燕云计算股份有限公司告知或查询。未经本公司明确授予的任何权利均予保留。

## 商标声明

 是深圳市金蝶天燕云计算股份有限公司向中华人民共和国国家商标局申请注册的注册商标，注册商标专用权由金蝶天燕合法拥有，受法律保护。未经金蝶天燕的书面许可，任何单位及个人不得以任何方式或理由对该商标的任何部分进行使用、复制、修改、传播、抄录或与其它产品捆绑使用销售。凡侵犯金蝶天燕商标权的，金蝶天燕将依法追究其法律责任。本档提及的其他所有商标或注册商标，由各自的所有人拥有。

# 目录

- .1 域
  - .1.1 启动域
    - .1.1.1 概要
    - .1.1.2 选项
    - .1.1.3 示例
  - .1.2 停止域
    - .1.2.1 概要
    - .1.2.2 选项
    - .1.2.3 示例
  - .1.3 创建域
    - .1.3.1 概要
    - .1.3.2 选项
    - .1.3.3 示例
  - .1.4 删除域
    - .1.4.1 概要
    - .1.4.2 选项
    - .1.4.3 示例
  - .1.5 备份域
    - .1.5.1 概要
    - .1.5.2 选项
- .2 节点
  - .2.1 查看节点
    - .2.1.1 概要
    - .2.1.2 选项
    - .2.1.3 示例
  - .2.2 安装节点
    - .2.2.1 概要
    - .2.2.2 选项
    - .2.2.3 示例
  - .2.3 安装节点-SSH
    - .2.3.1 概要
    - .2.3.2 选项
    - .2.3.3 示例
  - .2.4 创建节点-SSH
    - .2.4.1 概要
    - .2.4.2 选项
    - .2.4.3 示例
  - .2.5 创建节点-config
    - .2.5.1 概要
    - .2.5.2 选项
    - .2.5.3 示例
  - .2.6 删除节点-config
    - .2.6.1 概要
    - .2.6.2 选项
    - .2.6.3 示例
  - .2.7 删除节点-SSH
    - .2.7.1 概要
    - .2.7.2 选项
    - .2.7.3 示例

- .2.8 卸载节点
  - .2.8.1 概要
  - .2.8.2 选项
  - .2.8.3 示例
- .2.9 卸载节点-SSH
  - .2.9.1 概要
  - .2.9.2 选项
  - .2.9.3 示例
- .2.10 ping节点
  - .2.10.1 概要
  - .2.10.2 选项
  - .2.10.3 示例
- .3 实例
  - .3.1 查看实例
    - .3.1.1 概要
    - .3.1.2 选项
    - .3.1.3 示例
  - .3.2 创建实例
    - .3.2.1 概要
    - .3.2.2 选项
    - .3.2.3 示例
  - .3.3 创建本地实例
    - .3.3.1 概要
    - .3.3.2 选项
    - .3.3.3 示例
  - .3.4 启动实例
    - .3.4.1 概要
    - .3.4.2 选项
    - .3.4.3 示例
  - .3.5 启动本地实例
    - .3.5.1 概要
    - .3.5.2 选项
    - .3.5.3 示例
  - .3.6 停止实例
    - .3.6.1 概要
    - .3.6.2 选项
    - .3.6.3 示例
  - .3.7 停止本地实例
    - .3.7.1 概要
    - .3.7.2 选项
    - .3.7.3 示例
  - .3.8 删除实例
    - .3.8.1 概要
    - .3.8.2 选项
    - .3.8.3 示例
  - .3.9 删除本地实例
    - .3.9.1 概要
    - .3.9.2 选项
    - .3.9.3 示例
- .4 应用程序
  - .4.1 部署应用程序

- 4.1.1 概要
- 4.1.2 选项
- 4.1.3 示例
- 4.2 重新部署
  - 4.2.1 概要
  - 4.2.2 选项
  - 4.2.3 示例
- 4.3 卸载应用
  - 4.3.1 概要
  - 4.3.2 选项
  - 4.3.3 示例
- 4.4 引用应用程序
  - 4.4.1 概要
  - 4.4.2 选项
  - 4.4.3 示例
- 4.5 查询应用程序
  - 4.5.1 概要
  - 4.5.2 选项
  - 4.5.3 示例
- 4.6 列出应用程序引用
  - 4.6.1 概要
  - 4.6.2 选项
  - 4.6.3 示例
- 4.7 删除应用程序引用
  - 4.7.1 概要
  - 4.7.2 选项
  - 4.7.3 示例
- 4.8 启动应用程序
  - 4.8.1 概要
  - 4.8.2 选项
  - 4.8.3 示例
- 4.9 停止应用程序
  - 4.9.1 概要
  - 4.9.2 选项
  - 4.9.3 示例
- 4.10 查看组件状态
  - 4.10.1 概要
  - 4.10.2 选项
  - 4.10.3 示例
- 4.11 编辑应用或组件属性
  - 4.11.1 概要
  - 4.11.2 选项
  - 4.11.3 示例
- 5 JDBC连接池
  - 5.1 创建JDBC连接池
    - 5.1.1 概要
    - 5.1.2 选项
    - 5.1.3 示例
  - 5.2 删除JDBC连接池
    - 5.2.1 概要
    - 5.2.2 选项

- .5.2.3 示例
- .5.3 列出JDBC连接池
  - .5.3.1 概要
  - .5.3.2 选项
  - .5.3.3 示例
- .5.4 更新JDBC连接池信息
  - .5.4.1 概要
  - .5.4.2 选项
  - .5.4.3 示例
- .6 JDBC资源
  - .6.1 查看JDBC资源
    - .6.1.1 概要
    - .6.1.2 选项
    - .6.1.3 示例
  - .6.2 创建JDBC资源
    - .6.2.1 概要
    - .6.2.2 选项
    - .6.2.3 示例
  - .6.3 删除JDBC资源
    - .6.3.1 概要
    - .6.3.2 选项
    - .6.3.3 示例
  - .6.4 创建资源引用
    - .6.4.1 概要
    - .6.4.2 选项
    - .6.4.3 示例
  - .6.5 删除资源引用
    - .6.5.1 概要
    - .6.5.2 选项
    - .6.5.3 示例
  - .6.6 创建JDBC多数据源资源
    - .6.6.1 概要
    - .6.6.2 选项
    - .6.6.3 示例
  - .6.7 删除JDBC多数据源资源
    - .6.7.1 概要
    - .6.7.2 选项
    - .6.7.3 示例
  - .6.8 更新JDBC资源
    - .6.8.1 概要
    - .6.8.2 选项
    - .6.8.3 示例
- .7 服务器集群
  - .7.1 查询服务器集群
    - .7.1.1 概要
    - .7.1.2 选项
    - .7.1.3 示例
  - .7.2 创建服务器集群
    - .7.2.1 概要
    - .7.2.2 选项
    - .7.2.3 示例

- .7.3 启动服务器集群
  - .7.3.1 概要
  - .7.3.2 选项
  - .7.3.3 示例
- .7.4 停止服务器集群
  - .7.4.1 概要
  - .7.4.2 选项
  - .7.4.3 示例
- .7.5 删除服务器集群
  - .7.5.1 概要
  - .7.5.2 选项
  - .7.5.3 示例
- .8 负载均衡器
  - .8.1 查询负载均衡器
    - .8.1.1 概要
    - .8.1.2 选项
    - .8.1.3 示例
  - .8.2 创建负载均衡器
    - .8.2.1 概要
    - .8.2.2 选项
    - .8.2.3 示例
  - .8.3 启动负载均衡器
    - .8.3.1 概要
    - .8.3.2 选项
    - .8.3.3 示例
  - .8.4 停止负载均衡器
    - .8.4.1 概要
    - .8.4.2 选项
    - .8.4.3 示例
  - .8.5 更新负载均衡器
    - .8.5.1 概要
    - .8.5.2 选项
    - .8.5.3 示例
  - .8.6 删除负载均衡器
    - .8.6.1 概要
    - .8.6.2 选项
    - .8.6.3 示例
- .9 负载均衡器集群
  - .9.1 查看负载均衡器集群
    - .9.1.1 概要
    - .9.1.2 选项
    - .9.1.3 示例
  - .9.2 查看负载均衡器集群实例
    - .9.2.1 概要
    - .9.2.2 选项
    - .9.2.3 示例
  - .9.3 创建负载均衡器集群
    - .9.3.1 概要
    - .9.3.2 选项
    - .9.3.3 示例
  - .9.4 新建负载均衡器集群实例

- .9.4.1 概要
- .9.4.2 选项
- .9.4.3 示例
- .9.5 更新负载均衡器集群属性
  - .9.5.1 概要
  - .9.5.2 选项
  - .9.5.3 示例
- .9.6 更新负载均衡器集群中的负载均衡器实例
  - .9.6.1 概要
  - .9.6.2 选项
  - .9.6.3 示例
- .9.7 更新ApusicHA实例信息
  - .9.7.1 概要
  - .9.7.2 选项
  - .9.7.3 示例
- .9.8 启动负载均衡器集群
  - .9.8.1 概要
  - .9.8.2 选项
  - .9.8.3 示例
- .9.9 启动负载均衡器集群实例
  - .9.9.1 概要
  - .9.9.2 选项
  - .9.9.3 示例
- .9.10 停止负载均衡器集群
  - .9.10.1 概要
  - .9.10.2 选项
  - .9.10.3 示例
- .9.11 停止负载均衡器集群实例
  - .9.11.1 概要
  - .9.11.2 选项
  - .9.11.3 示例
- .10 监视管理
  - .10.1 开启监控服务
    - .10.1.1 概要
    - .10.1.2 选项
    - .10.1.3 示例
  - .10.2 关闭监控服务
    - .10.2.1 概要
    - .10.2.2 选项
    - .10.2.3 示例
  - .10.3 监视管理
    - .10.3.1 概要
    - .10.3.2 选项
    - .10.3.3 示例
  - .10.4 监控阻塞线程
    - .10.4.1 概要
    - .10.4.2 选项
    - .10.4.3 示例
  - .10.5 监控繁忙线程
    - .10.5.1 概要
    - .10.5.2 选项

- 10.5.3 示例
- 10.6 监控长线程
  - 10.6.1 概要
  - 10.6.2 选项
  - 10.6.3 示例
- 10.7 生成火焰图
  - 10.7.1 概要
  - 10.7.2 选项
  - 10.7.3 示例
- 11 JMS资源
  - 11.1 查看JMS资源
    - 11.1.1 概要
    - 11.1.2 选项
    - 11.1.3 示例
  - 11.2 创建JMS资源
    - 11.2.1 概要
    - 11.2.2 选项
    - 11.2.3 示例
  - 11.3 删除JMS资源
    - 11.3.1 概要
    - 11.3.2 选项
    - 11.3.3 示例
  - 11.4 查看JMS物理目标
    - 11.4.1 概要
    - 11.4.2 选项
    - 11.4.3 示例
  - 11.5 创建JMS物理目标
    - 11.5.1 概要
    - 11.5.2 选项
    - 11.5.3 示例
  - 11.6 删除JMS物理目标
    - 11.6.1 概要
    - 11.6.2 选项
    - 11.6.3 示例
- 12 共享类库
  - 12.1 新建共享类库
    - 12.1.1 概要
    - 12.1.2 选项
    - 12.1.3 示例
  - 12.2 删除共享类库资源
    - 12.2.1 概要
    - 12.2.2 选项
    - 12.2.3 示例
  - 12.3 删除共享类库
    - 12.3.1 概要
    - 12.3.2 选项
    - 12.3.3 示例
- 13 JavaMail会话
  - 13.1 查看JavaMail会话
    - 13.1.1 概要
    - 13.1.2 选项

- 13.1.3 示例
- 13.2 新建JavaMail会话
  - 13.2.1 概要
  - 13.2.2 选项
  - 13.2.3 示例
- 13.3 删除JavaMail会话
  - 13.3.1 概要
  - 13.3.2 选项
  - 13.3.3 示例
- 14 资源适配器配置
  - 14.1 查看资源适配器配置
    - 14.1.1 概要
    - 14.1.2 选项
    - 14.1.3 示例
  - 14.2 新建资源适配器配置
    - 14.2.1 概要
    - 14.2.2 选项
    - 14.2.3 示例
  - 14.3 删除资源适配器配置
    - 14.3.1 概要
    - 14.3.2 选项
    - 14.3.3 示例
- 15 上下文服务
  - 15.1 查看上下文服务
    - 15.1.1 概要
    - 15.1.2 选项
    - 15.1.3 示例
  - 15.2 新建上下文服务
    - 15.2.1 概要
    - 15.2.2 选项
    - 15.2.3 示例
  - 15.3 删除上下文服务
    - 15.3.1 概要
    - 15.3.2 选项
    - 15.3.3 示例
- 16 托管线程工厂
  - 16.1 查看托管线程工厂
    - 16.1.1 概要
    - 16.1.2 选项
    - 16.1.3 示例
  - 16.2 新建托管线程工厂
    - 16.2.1 概要
    - 16.2.2 选项
    - 16.2.3 示例
  - 16.3 删除托管线程工厂
    - 16.3.1 概要
    - 16.3.2 选项
    - 16.3.3 示例
- 17 执行者服务
  - 17.1 查看执行者服务资源
    - 17.1.1 概要

- 17.1.2 选项
- 17.1.3 示例
- 17.2 新建执行者服务资源
  - 17.2.1 概要
  - 17.2.2 选项
  - 17.2.3 示例
- 17.3 删除执行者服务资源
  - 17.3.1 概要
  - 17.3.2 选项
  - 17.3.3 示例
- 18 周期性执行者服务
  - 18.1 查看周期性执行者服务资源
    - 18.1.1 概要
    - 18.1.2 选项
    - 18.1.3 示例
  - 18.2 新建周期性执行者服务资源
    - 18.2.1 概要
    - 18.2.2 选项
    - 18.2.3 示例
  - 18.3 删除周期性执行者服务资源
    - 18.3.1 概要
    - 18.3.2 选项
    - 18.3.3 示例
- 19 连接器连接池
  - 19.1 查看连接器连接池
    - 19.1.1 概要
    - 19.1.2 选项
    - 19.1.3 示例
  - 19.2 新建连接器连接池
    - 19.2.1 概要
    - 19.2.2 选项
    - 19.2.3 示例
  - 19.3 删除连接器连接池
    - 19.3.1 概要
    - 19.3.2 选项
    - 19.3.3 示例
- 20 连接器资源
  - 20.1 查看连接器资源
    - 20.1.1 概要
    - 20.1.2 选项
    - 20.1.3 示例
  - 20.2 新建连接器资源
    - 20.2.1 概要
    - 20.2.2 选项
    - 20.2.3 示例
  - 20.3 删除连接器资源
    - 20.3.1 概要
    - 20.3.2 选项
    - 20.3.3 示例
- 21 管理对象资源
  - 21.1 查看管理对象资源

- 21.1.1 概要
- 21.1.2 选项
- 21.1.3 示例
- 21.2 新建管理对象资源
  - 21.2.1 概要
  - 21.2.2 选项
  - 21.2.3 示例
- 21.3 删除管理对象资源
  - 21.3.1 概要
  - 21.3.2 选项
  - 21.3.3 示例
- 22 安全映射
  - 22.1 查看安全映射
    - 22.1.1 概要
    - 22.1.2 选项
    - 22.1.3 示例
  - 22.2 新建安全映射
    - 22.2.1 概要
    - 22.2.2 选项
    - 22.2.3 示例
  - 22.3 删除安全映射
    - 22.3.1 概要
    - 22.3.2 选项
    - 22.3.3 示例
- 23 OSGI Bundle储存库
  - 23.1 上传OSGI Bundle资源
    - 23.1.1 概要
    - 23.1.2 选项
    - 23.1.3 示例
  - 23.2 删除OSGI Bundle资源
    - 23.2.1 概要
    - 23.2.2 选项
    - 23.2.3 示例
- 24 线程池管理
  - 24.1 查看线程池
    - 24.1.1 概要
    - 24.1.2 选项
    - 24.1.3 示例
  - 24.2 新建线程池
    - 24.2.1 概要
    - 24.2.2 选项
    - 24.2.3 示例
  - 24.3 删除线程池
    - 24.3.1 概要
    - 24.3.2 选项
    - 24.3.3 示例
- 25 协议配置
  - 25.1 查看协议信息
    - 25.1.1 概要
    - 25.1.2 选项
    - 25.1.3 示例

- .25.2 新建协议信息
  - .25.2.1 概要
  - .25.2.2 选项
  - .25.2.3 示例
- .25.3 删除协议
  - .25.3.1 概要
  - .25.3.2 选项
  - .25.3.3 示例
- .26 网络监听程序
  - .26.1 查看网络监听程序
    - .26.1.1 概要
    - .26.1.2 选项
    - .26.1.3 示例
  - .26.2 新建网络监听程序
    - .26.2.1 概要
    - .26.2.2 选项
    - .26.2.3 示例
  - .26.3 删除网络监听程序
    - .26.3.1 概要
    - .26.3.2 选项
    - .26.3.3 示例
- .27 HTTP参数
  - .27.1 新建HTTP参数
    - .27.1.1 概要
    - .27.1.2 选项
    - .27.1.3 示例
- .28 传输
  - .28.1 查看传输
    - .28.1.1 概要
    - .28.1.2 选项
    - .28.1.3 示例
  - .28.2 新建传输
    - .28.2.1 概要
    - .28.2.2 选项
    - .28.2.3 示例
  - .28.3 删除传输
    - .28.3.1 概要
    - .28.3.2 选项
    - .28.3.3 示例
- .29 虚拟服务器
  - .29.1 查看虚拟服务器
    - .29.1.1 概要
    - .29.1.2 选项
    - .29.1.3 示例
  - .29.2 新建虚拟服务器
    - .29.2.1 概要
    - .29.2.2 选项
    - .29.2.3 示例
  - .29.3 删除虚拟服务器
    - .29.3.1 概要
    - .29.3.2 选项

- 29.3.3 示例
- 30 日志配置
  - 30.1 设置日志属性
    - 30.1.1 概要
    - 30.1.2 选项
    - 30.1.3 示例
  - 30.2 设置日志级别
    - 30.2.1 概要
    - 30.2.2 选项
    - 30.2.3 示例
  - 30.3 滚动日志
    - 30.3.1 概要
    - 30.3.2 选项
    - 30.3.3 示例
  - 30.4 收集日志
    - 30.4.1 概要
    - 30.4.2 选项
    - 30.4.3 示例
- 31 安全管理
  - 31.1 开启安全管理
    - 31.1.1 概要
    - 31.1.2 选项
    - 31.1.3 示例
  - 31.2 禁用安全管理
    - 31.2.1 概要
    - 31.2.2 选项
    - 31.2.3 示例
- 32 list子命令
  - 32.1 list子命令
    - 32.1.1 概要
    - 32.1.2 选项
    - 32.1.3 示例
- 33 get子命令
  - 33.1 get子命令
    - 33.1.1 概要
    - 33.1.2 选项
    - 33.1.3 示例
- 34 set子命令
  - 34.1 set子命令
    - 34.1.1 概要
    - 34.1.2 选项
    - 34.1.3 示例
- 35 export子命令
  - 35.1 export子命令
    - 35.1.1 概要
    - 35.1.2 选项
    - 35.1.3 示例
- 36 unset子命令
  - 36.1 unset子命令
    - 36.1.1 概要
    - 36.1.2 选项

- 36.1.3 示例
- 37 enable子命令
  - 37.1 enable子命令
    - 37.1.1 概要
    - 37.1.2 选项
    - 37.1.3 示例
- 38 disable子命令
  - 38.1 disable子命令
    - 38.1.1 概要
    - 38.1.2 选项
    - 38.1.3 示例
- 39 角色管理
  - 39.1 管理角色子命令
    - 39.1.1 概要
    - 39.1.2 选项
    - 39.1.3 示例
- 40 用户管理
  - 40.1 管理用户子命令
    - 40.1.1 概要
    - 40.1.2 选项
    - 40.1.3 示例
- 41 附件
  - 41.1 dotted-names

# 1 域

## 1.1 启动域

`start-domain` 子命令启动指定域的域管理服务器 (DAS)。如果未指定域, 则表示为默认域。如果 `domain` 目录包含两个或多个域, 则必须指定\* `domain-name` \*属性值。

### 1.1.1 概要

```
start-domain [--help]
[--debug={true|false}] [--domaindir domain-dir]
[--dry-run={true|false}] [--upgrade={true|false}]
[--verbose={true|false}] [--watchdog={true|false}]
[domain-name]
```

### 1.1.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--debug
```

指定域是否在启用Java 平台调试器体系结构 (JPDA)调试的情况下启动。

可能的值如下:

- `true`  
实例在启用 JPDA 调试的情况下启动, 并显示用于 JPDA 调试的端口号, 默认调试端口为8000。
- `false`  
该实例在禁用 JPDA 调试的情况下启动 (默认)。

```
--domaindir
```

域根目录, 包含要重启的域的目录。如果指定, 则该路径必须在文件系统中可访问。域根目录的默认位置是 `aas-install/domains`。

```
--upgrade
```

指定域管理服务器 (DAS) 的配置是否升级到当前版本。通常, 如果子命令检测到配置为旧版本, 则配置会在启动之前自动升级。一般情况下不需要显式使用此选项。

可能的值如下:

- `true`  
当域启动时, 配置被修改为与此版本的AAS兼容, 并且AAS进程停止。
- `false`  
DAS 的配置未更新 (默认)。

```
--verbose
```

```
-v
```

指定是否在运行子命令的控制台窗口中显示有关域的详细信息。

可能的值如下:

- `true`

有关域的详细启动消息和日志消息显示在运行子命令的控制台窗口中。如果稍后通过从不同的控制台窗口运行 `restart-domain` 子命令重新启动域，则消息将继续显示在原始控制台窗口中。可以通过 `CTRL-C` 在控制台窗口中键入来终止 AAS 进程。可以通过在控制台窗口中键入以下组合键之一来终止 AAS 进程并获取服务器的线程转储：`CTRL-\` 在 UNIX 系统上，`CTRL-Break` 在 Windows 系统上。

- `false`

不显示详细信息（默认）。

```
domain-name
```

要启动的域的唯一名称。如果只有一个默认域，可以忽略该属性。

### 1.1.3 示例

示例1: 启动默认域mydomain

```
[apusic@AAS-168 bin]# ./asadmin start-domain
Waiting for mydomain to start ...
Successfully started the domain : mydomain
domain Location: /opt/ApusicAS/aas/domains/mydomain
Log File: /opt/ApusicAS/aas/domains/mydomain/logs/server.log
Admin Port: 6848
Command start-domain executed successfully.
```

示例2: 启动指定域mydomain1

```
[apusic@AAS-168 bin]# ./asadmin start-domain mydomain1
Waiting for mydomain1 to start ...
Successfully started the domain : mydomain1
domain Location: /opt/ApusicAS/aas/domains/mydomain1
Log File: /opt/ApusicAS/aas/domains/mydomain1/logs/server.log
Admin Port: 6848
Command start-domain executed successfully.
```

## 1.2 停止域

`stop-domain` 子命令停止指定域的域管理服务器 (DAS)。如果未指定域目录，则停止默认域目录中的域。如果 `domain` 目录中有两个或多个域，则必须指定 `* domain-name *` 属性值。

在本地或远程模式下支持此子命令。如果指定主机名，则子命令表示在远程模式下操作，这意味着必须正确地向远程服务器进行身份验证。在本地模式下，只要以启动服务器的同一用户身份运行子命令，通常不需要对服务器进行身份验证。

### 1.2.1 概要

```
stop-domain [--help] [--domaindir domaindir]
[--force={true|false}] [--kill={false|true}]
[domain-name]
```

### 1.2.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--domaindir
```

指定要停止的域的目录。如果指定，则该路径必须在文件系统中可访问。如果未指定，默认位置 `aas-install/domains` 中的域将停止。

```
--force
```

指定是否立即强制停止域。

可能的值如下：

- `true`

域被立即强制停止（默认）。

- `false`

该子命令会一直等待，直到与该域关联的所有线程都退出，然后再停止该域。

```
--kill
```

指定是否通过使用操作系统的功能终止域进程来终止域。

可能的值如下：

- `false`

域没有被杀死。该子命令使用 Java 平台的功能来终止域进程（默认）。

- `true`

域被杀死。该子命令使用操作系统的功能来终止域进程。

### 1.2.3 示例

示例1: 停止默认域mydomain

```
[apusic@AAS-168 bin]# ./asadmin stop-domain
Waiting for the domain to stop ...
Command stop-domain executed successfully.
```

示例2: 停止指定域mydomain1

```
[apusic@AAS-168 bin]# ./asadmin stop-domain mydomain1
Waiting for the domain to stop ...
Command stop-domain executed successfully.
```

## 1.3 创建域

`create-domain` 子命令创建一个 AAS 域。AAS 中的域是符合 Java Platform, Enterprise Edition (Java EE) 标准的管理名称空间。每个域都有一个配置，该配置存储在一组文件中。可以在给定的 AAS 安装中创建任意数量的域，每个域都有不同的管理身份。一个域可以独立于其他域而存在。

任何有权访问 `asadmin` 实用程序的用户都可以创建域并将其配置存储在用户选择的文件夹中。默认情况下，域配置是在域默认目录中创建的。可以覆盖此位置以将配置存储在其他位置。

如果在运行子命令时在 `aas-install/modules` 目录的 JAR 文件中找到域定制器 `create-domain`，则处理定制器。域定制器是实现 `DomainInitializer` 接口的类。

该 `create-domain` 子命令使用 `asadmin` 实用程序选项指定的单个管理用户创建域 `--user`。如果 `--user` 未指定该 `--nopassword` 选项，并且该选项设置为 `true`，则使用默认管理用户，`admin`。如果该 `--nopassword` 选项设置为 `false`（默认值），则需要用户名。在这种情况下，如果没有使用该 `--user` 选项指定用户名，系统会显示提示

此子命令仅在本地模式下受支持。

### 1.3.1 概要

```
create-domain [--help]
              [--adminport adminport]
              [--instanceport instanceport]
              [--portbase portbase]
```

```

[--template template-name]
[--domaindir domaindir]
[--savemasterpassword={false|true}]
[--usemasterpassword={false|true}]
[--domainproperties (name=value)[:name=value]*]
[--keytooloptions (name=value)[:name=value]*]
[--savelogin={false|true}]
[--checkports={true|false}]
[--nopassword={false|true}]
domain-name

```

### 1.3.2 选项

```

--help
-?

```

显示子命令的帮助文本。

```

--adminport

```

用于管理的 HTTP 端口或 HTTPS 端口。此端口是在 Web 浏览器中指定用于管理域的 URL 中的端口，例如 `http://localhost:6949`。该 `--adminport` 选项不能与该 `--portbase` 选项一起使用。默认值为 6848。

该 `--adminport` 选项覆盖选项的 `domain.adminPort` 属性 `--domainproperties`。

```

--instanceport

```

域提供服务，以便应用程序在部署时可以运行。此 HTTP 端口指定 Web 应用程序上下文根可用于 Web 浏览器连接的位置。此端口是一个正整数，并且必须在创建域时可用。该 `--instanceport` 选项不能与该 `--portbase` 选项一起使用。默认值为 8080。

该 `--instanceport` 选项覆盖选项的 `domain.instancePort` 属性 `--domainproperties`。

```

--portbase

```

确定端口分配应开始的编号。域使用一定数量的静态分配的端口。该 `*portbase*` 值决定了分配应该开始。端口的值计算如下：

- Administration port: `portbase + 48`
- HTTP listener port: `portbase + 80`
- HTTPS listener port: `portbase + 81`
- JMS port: `portbase + 76`
- IIOP listener port: `portbase + 37`
- Secure IIOP listener port: `portbase + 38`
- Secure IIOP with mutual authentication port: `portbase + 39`
- JMX port: `portbase + 86`
- JPDA debugger port: `portbase + 9`
- Felix shell service port for OSGi module management: `portbase + 66`

当 `--portbase` 指定选项，该子命令的输出包括使用的端口的完整列表。

该 `--portbase` 选项不能与使用 `--adminport`，`--instanceport` 或 `--domainproperties` 选择。

```

--template

```

用于创建域的域配置模板的文件名，包括相对或绝对路径。如果指定了相对路径，则子命令会将路径附加到 `as-install/lib/templates` 目录以定位文件。如果它是绝对路径名，则子命令在指定路径中定位文件。

此选项允许创建不同类型的域并定义自定义域模板。

```

--domaindir

```

要创建域的目录。如果指定，则该路径必须在文件系统中可访问。如果未指定，则在默认域目录 `as-install` 中创建域 `/domains`。

### `--savemasterpassword`

将此选项设置为 `true` 允许将主密码写入文件系统。无论在命令行上指定的值如何，如果此选项为 `true`，则 `--usemasterpassword` 选项也为 `true`。默认值为 `false`。

主密码实际上是安全密钥存储的密码。域旨在将自己的证书（在域创建时创建）保存在配置位置的安全位置。此证书称为域的SSL服务器证书。当Web浏览器通过安全通道（HTTPS）联系域时，此证书由域提供。主密码应该保护包含此证书的存储（一个文件）。此文件称为 `keystore.jks`，在创建的域的配置目录中创建。但是，如果选择此选项，则主密码将保存在域的配置位置的磁盘上。主密码存储在一个名为 `master-password` 的文件中，该文件是一个JavaJCEKS类型的密钥库。使用 `--savemasterpassword` 选项的原因是用于无人值守系统引导。在这种情况下，域启动时不会提示输入主密码，因为密码将从该文件中提取出来。

创建域时最好创建一个主密码，因为主密码由 `start-domain` 子命令使用。出于安全目的，默认设置应该为 `false`，因为将主密码保存在磁盘上是一种不安全的做法，除非正确设置了文件系统权限。如果保存了主密码，则 `start-domain` 不会提示输入它。主密码为环境提供了额外的安全级别。

### `--usemasterpassword`

指定是使用系统内置的主密码还是用户定义的主密码对密钥库进行加密。

如果 `false`（默认），则使用系统内置的众所周知的密码对密钥库进行加密。使用系统内置的密码加密密钥库不会提供额外的安全性。

如果 `true`，则子命令从 `AS_ADMIN_MASTERPASSWORD` 密码文件中的条目中获取主密码或提示输入主密码。密码文件 `--passwordfile` 在 `asadmin` 实用程序的选项中指定。

如果 `--savemasterpassword` 选项为 `true`，则此选项也为真，无论命令行上指定的值如何。

### `--domainproperties`

设置可选的 `name/value` 会覆盖要创建的域属性的默认值。列表必须用冒号（:）字符分隔。该 `--portbase` 选项不能与使用 `--domainproperties` 选项。以下属性可用：

- `domain.adminPort`

此属性指定用于管理的 HTTP 端口或 HTTPS 端口的端口号。此端口是在 Web 浏览器中指定用于管理实例的 URL 中的端口，例如 `http://localhost:6949`。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。该 `domain.adminPort` 属性被 `--adminport` 选项覆盖。

- `domain.instancePort`

此属性指定用于侦听 HTTP 请求的端口号。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。该 `domain.instancePort` 属性被 `--instanceport` 选项覆盖。

- `domain.jmxPort`

此属性指定 JMX 连接器侦听的端口号。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。

- `http.ssl.port`

此属性指定用于侦听 HTTPS 请求的端口的端口号。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。

- `java.debugger.port`

此属性指定用于连接到Java平台调试器体系结构（JPDA）调试器的端口号。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。

- `jms.port`

此属性指定 Java 消息服务提供者的端口号。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。

- `orb.listener.port`

此属性指定用于 IIOP 连接的端口号。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。

- `orb.mutualauth.port`

此属性指定用于通过客户端身份验证进行安全 IIOP 连接的端口的端口号。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。

- `orb.ssl.port`

此属性指定用于安全 IIOP 连接的端口号。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。

- `osgi.shell.telnet.port`

此属性指定用于连接到Apache Felix Remote Shell的端口号。这个shell 使用Felix shell 服务与OSGi 模块管理子系统进行交互。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。

#### --keytooloptions

为自签名服务器证书指定 keytool 选项的名称-值对的可选列表。证书是在创建域期间生成的。列表中的每一对都必须用冒号 (:) 字符分隔。

允许的选项如下：

- CN  
指定要用于自签名证书的主机的公用名。此选项名称不区分大小写。默认情况下，该名称 `create-domain` 是运行子命令的主机的完全限定名称。

#### --savelogin

如果设置为 true，此选项将保存管理用户名和密码。默认值为假。用户名和密码存储 `.asadminpass` 在用户主目录的文件中。域只能在本地创建。因此，使用该 `--savelogin` 选项时，保存的主机名 `.asadminpass` 始终为 `localhost`。如果用户指定的默认管理端口，同时创建域，没有必要指定 `--user`，`--passwordfile`，`--host`，或 `--port` 在任何后续的 `asadmin` 远程命令。这些值将自动获得。

#### --checkports

指定是否检查管理、HTTP、JMS、JMX 和 IIOP 端口的可用性。默认值是true。

#### --nopassword

指定管理用户是否有密码。如果为 false（默认值），则密码由密码文件中的 `AS_ADMIN_PASSWORD` 条目指定 `asadmin`（使用 `--passwordfile` 选项设置）。如果为 false `AS_ADMIN_PASSWORD` 且未设置，系统将提示您输入密码。

如果为 true，则无需密码即可创建管理用户。如果未使用该 `--user` 选项指定域的用户名，并且该 `--nopassword` 选项设置为 true，则使用默认用户名，`admin`。

#### domain-name

定义该域的名称。

### 1.3.3 示例

示例 1：创建域

此示例创建一个名为domain4的域。

```
asadmin>create-domain --adminport 4848 domain4
Enter admin user name [Enter to accept default "admin" / no password]>
Using port 4848 for Admin.
Using default port 8080 for HTTP Instance.
Using default port 7676 for JMS.
Using default port 3700 for IIOP.
Using default port 8181 for HTTP_SSL.
Using default port 3820 for IIOP_SSL.
Using default port 3920 for IIOP_MUTUALAUTH.
Using default port 8686 for JMX_ADMIN.
Using default port 6666 for OSGI_SHELL.
Distinguished Name of the self-signed X.509 Server Certificate is:
[CN=sr1-usca-22,OU=APUSIC,O=Oracle Corp.,L=Redwood Shores,ST=California,C=US]
No domain initializers found, bypassing customization step
Domain domain4 created.
Domain domain4 admin port is 4848.
Domain domain4 allows admin login as user "admin" with no password.
Command create-domain executed successfully.
```

## 示例 2：在指定目录中创建域

此示例创建一个在 /home/someuser/domains 目录中命名sampleDomain的域。

```

asadmin> create-domain --domaindir /home/someuser/domains --adminport 7070
--instanceport 7071 sampleDomain
Enter admin user name [Enter to accept default "admin" / no password]>
Using port 7070 for Admin.
Using port 7071 for HTTP Instance.
Using default port 7676 for JMS.
Using default port 3700 for IIOP.
Using default port 8181 for HTTP_SSL.
Using default port 3820 for IIOP_SSL.
Using default port 3920 for IIOP_MUTUALAUTH.
Using default port 8686 for JMX_ADMIN.
Using default port 6666 for OSGI_SHELL.
Enterprise ServiceDistinguished Name of the self-signed X.509 Server Certificate is:
[CN=sr1-usca-22,OU=APUSIC,O=Oracle Corp.,L=Redwood Shores,ST=California,C=US]
No domain initializers found, bypassing customization step
Domain sampleDomain created.
Domain sampleDomain admin port is 7070.
Domain sampleDomain allows admin login as user "admin" with no password.
Command create-domain executed successfully.

```

## 示例 3：创建域并保存管理用户名和密码

此示例创建一个名为myDomain的域并保存管理用户名和密码。

```

asadmin> create-domain --adminport 8282 --savelogin=true myDomain
Enter the admin password [Enter to accept default of no password]>
Enter the master password [Enter to accept default password "changeit"]>
Using port 8282 for Admin.
Using default port 8080 for HTTP Instance.
Using default port 7676 for JMS.
Using default port 3700 for IIOP.
Using default port 8181 for HTTP_SSL.
Using default port 3820 for IIOP_SSL.
Using default port 3920 for IIOP_MUTUALAUTH.
Using default port 8686 for JMX_ADMIN.
Using default port 6666 for OSGI_SHELL.
Enterprise ServiceDistinguished Name of the self-signed X.509 Server Certificate is:
[CN=sr1-usca-22,OU=APUSIC,O=Oracle Corp.,L=Redwood Shores,ST=California,C=US]
No domain initializers found, bypassing customization step
Domain myDomain created.
Domain myDomain admin port is 8282.
Domain myDomain allows admin login as user "admin" with no password.
Login information relevant to admin user name [admin]
for this domain [myDomain] stored at
[/home/someuser/.asadminpass] successfully.
Make sure that this file remains protected.
Information stored in this file will be used by
asadmin commands to manage this domain.
Command create-domain executed successfully.

```

## 示例 4：创建域并指定证书主机

此示例创建一个名为 domain5 的域。要用于自签名证书的主机的通用名称是 trio。

```
asadmin> create-domain --adminport 9898 --keytooloptions CN=trio domain5
Enter the admin password [Enter to accept default of no password]>
Enter the master password [Enter to accept default password "changeit"]>
Using port 9898 for Admin.
Using default port 8080 for HTTP Instance.
Using default port 7676 for JMS.
Using default port 3700 for IIOP.
Using default port 8181 for HTTP_SSL.
Using default port 3820 for IIOP_SSL.
Using default port 3920 for IIOP_MUTUALAUTH.
Using default port 8686 for JMX_ADMIN.
Using default port 6666 for OSGI_SHELL.
Distinguished Name of the self-signed X.509 Server Certificate is:
[CN=trio,OU=APUSIC,O=Oracle Corp.,L=Redwood Shores,ST=California,C=US]
No domain initializers found, bypassing customization step
Domain domain5 created.
Domain domain5 admin port is 9898.
Domain domain5 allows admin login as user "admin" with no password.
Command create-domain executed successfully.
```

## 1.4 删除域

`delete-domain` 子命令删除指定的域。该域必须已经存在并且必须停止。

此子命令仅在本地模式下受支持。

### 1.4.1 概要

```
delete-domain [--help] [--domaindir domaindir] domain-name
```

### 1.4.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--domaindir
```

要删除的域所在的目录。如果指定，则该路径必须在文件系统中可访问。如果不指定，则删除域根目录下的域，默认为 `aas-install/domains`。

```
domain-name
```

指定需要删除的域名称。

### 1.4.3 示例

示例 1：删除域

此示例 `mydomain1` 从默认域目录中删除一个名为的域。

```
asadmin> delete-domain mydomain1
Domain mydomain1 deleted.
Command delete-domain executed successfully.
```

示例 2：从指定位置删除域

此示例从 `/home/someuser/domains` 目录中删除一个名为sampleDomain的域。

```
asadmin> delete-domain --domaindir /home/someuser/domains sampleDomain
Domain sampleDomain deleted
Command delete-domain executed successfully.
```

## 1.5 备份域

`backup-domain` 子命令备份指定的域。该域必须已经存在并且必须停止。

### 1.5.1 概要

```
backup-domain [--help]
[--long[={false| true}]]
[--description description-text]
[--domaindir domain-root-dir]
[--backupdir backup-directory]
[--backupconfig backup-config-name]
[domain_name]
```

### 1.5.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--long
```

是否展示详细信息，值为true或false。默认为false。

```
--description
```

指定要存储在备份文件中的描述。该描述显示为有关备份文件的信息的一部分。默认值的形式如下：`domain-name backup created on YYYY**_**MM**_**DD by user user-name`

```
--domaindir
```

指定域根目录，即要备份的域的父目录。默认值为 `aas-install/domains`。

```
--backupdir
```

指定存储备份文件的目录。默认值为 `aas-install/domains/domain-dir/backups`。如果域不在默认位置，则位置为 `domain-dir/backups`。

```
--backupconfig
```

存储备份文件的备份目录中的域备份配置的名称。

```
domain_name
```

指定要备份的域的名称。如果Apusic应用服务器只存在一个域，则此属性值可忽略。

## 2 节点

### 2.1 查看节点

`list-nodes` 子命令列出域中的所有节点。默认情况下，子命令为列出的每个节点显示以下信息：■ 节点的名称 ■ 节点的类型，可以是以下类型之一：

- CONFIG

该节点不支持远程通信。

- SSH

该节点支持通过安全shell (SSH) 进行通信。

■ 节点所代表的主机的名称 子命令的 `--long` 选项指定节点是否以长格式列出。在长格式中，关于每个节点的以下附加信息显示有：■ 节点所代表的主机上Apusic应用服务器的基本安装目录的父目录的路径 ■ 以逗号分隔的节点上Apusic应用服务器实例的名称列表 如果asadmin实用程序的 `--terse` 选项为true，子命令的 `--long` 选项为false，则子命令仅列出每个节点的名称。此子命令仅支持在远程模式下执行。

#### 2.1.1 概要

```
list-nodes [--help]
[--long={false| true}]
```

#### 2.1.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--long
-1
```

指定是否以详细格式显示。

可能的值有：

- true

节点以详细格式显示

- false

节点以简短格式显示，默认值

#### 2.1.3 示例

示例1: 显示当前所有节点。

```
asadmin> list-nodes
localhost-mydomain CONFIG localhost
mynode CONFIG 127.0.0.1
Command list-nodes executed successfully.
```

## 2.2 安装节点

`install-node` 子命令在指定为该子命令属性值的主机上安装 Apusic Server 软件。此子命令需要在运行子命令的主机和安装 Apusic Server 软件的每台主机上配置安全 shell (SSH)。

注意：此子命令等效于 `install-node-ssh` 子命令。

如有必要，该子命令会从运行该子命令的安装中创建 Apusic Server 软件的 ZIP 存档。存档不包含 `domains` 目录或 `nodes` 目录。当创建和启动代表主机的节点上的实例时，这些目录与域管理服务器 (DAS) 同步。从指定主机上的存档安装 Apusic Server 软件后，该子命令不会删除存档。

如果指定多个主机，则以下各项的配置在所有主机上都相同：

- 基础安装目录
- SSH 端口
- SSH 用户
- SSH 密钥文件

如果要安装 Apusic Server 软件的主机上不存在 SSH 密钥文件，则子命令以交互方式运行并提示输入密码。要使子命令以非交互方式运行，必须满足以下条件：

- `asadmin`实用程序的 `--interactive` 选项必须是 `false`
- 该实用程序的 `--passwordfile` 选项 `asadmin` 必须指定一个密码文件。
- 密码文件必须包含该 `AS_ADMIN_SSHPASSWORD` 条目。

该子命令不会修改 DAS 的配置。

此子命令仅在本地模式下受支持。

### 2.2.1 概要

```
install-node [--help]
[--archive archive]
[--create={false|true}] [--save[={false|true}]]
[--installdir as-install-parent]
[--sshport ssh-port] [--sshuser ssh-user]
[--sshkeyfile ssh-keyfile]
[--force={false|true}]
host-list
```

### 2.2.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--archive
```

要安装的 Apusic Server 软件的存档文件的绝对路径。如果未指定存档文件，则子命令会从运行子命令的 Apusic Server 软件的安装中创建存档。此存档在运行该命令的用户的主目录中创建。

```
--create
```

指定子命令是否应创建要安装的 Apusic Server 软件的存档文件。

- `false`  
不创建存档文件。该子命令从 `--archive` 选项指定的现有存档文件（默认）安装软件。
- `true`  
该子命令从运行该子命令的 Apusic Server 软件的安装中创建一个存档文件。

```
--save
```

指定安装后是否保存安装软件的存档文件。

- `false`  
存档文件未保存。该子命令在安装软件后删除文件（默认）。

- `true`

存档文件被保存。

```
--installdir
```

Apusic Server 软件将在每个主机上安装的基本安装目录的父级的绝对路径，例如 `/home/apusic/`。如果该目录不存在，则子命令将创建该目录。

运行此子命令的用户必须对指定目录具有写访问权。否则，会发生错误。

要覆盖 Apusic Server 软件的现有安装，请将 `--force` 选项设置为 `true`。如果目录已包含安装且 `--force` 选项为 `false`，则会发生错误。

默认值是运行此子命令的主机上 Apusic Server 软件的基本安装目录的父目录。

```
--sshport
```

用于 SSH 连接到要安装 Apusic Server 软件的主机的端口。默认值为 22。

```
--sshuser
```

要安装 Apusic Server 软件的主机上的用户，该用户将运行通过 SSH 连接到主机的过程。默认值是运行此子命令的用户。要确保 DAS 可以读取此用户的 SSH 私钥文件，请指定运行 DAS 进程的用户。

```
--sshkeyfile
```

`--sshuser` 选项指定的用户的 SSH 私钥文件的绝对路径。此文件用于对 `sshd` 主机上的守护程序进行身份验证。

运行此子命令的用户必须能够到达密钥文件的路径并读取密钥文件。

默认是用户 `.ssh` 目录中的密钥文件。如果找到多个密钥文件，则子命令使用以下优先顺序：

1. `id_rsa`
2. `id_dsa`
3. `identity`

```
--force
```

是否强制安装 Apusic Server 软件。`--installdir` 指定子命令是否覆盖选项指定目录中现有的 Apusic Server 软件安装。可能的值如下：

- `false`
- `true`

现有安装不会被覆盖（默认）。

现有安装将被覆盖。

### 2.2.3 示例

示例 1：在服务器 172.2.1.1 的目录 `/opt/testz/mytestin` 中安装 AAS

```
asadmin> install-node --installdir=/opt/testz/mytestin 172.2.1.1
Enter remote password for root@172.2.1.1>
使用密码进行授权 <concealed>
Created installation zip /opt/testz/0216/ApusicAS/aas/bin/aas1412650340676358384.zip
使用密码进行授权 <concealed>
Copying /opt/testz/0216/ApusicAS/aas/bin/aas1412650340676358384.zip (369402454 bytes) to
172.2.1.1:/opt/testz/mytestin
Installing aas1412650340676358384.zip into 172.2.1.1:/opt/testz/mytestin
Removing 172.2.1.1:/opt/testz/mytestin/aas1412650340676358384.zip
使用密码进行授权 <concealed>
```

```
Fixing file permissions of all bin files under 172.2.1.1:/opt/testz/mytestin
Fixing file permissions for nadmin file under 172.2.1.1:/opt/testz/mytestin/aas/lib
Command install-node executed successfully.
```

## 2.3 安装节点-SSH

`install-node-ssh` 子命令在指定为该子命令属性值的主机上安装 Apusic Server 软件。此子命令需要在运行子命令的主机和安装 Apusic Server 软件的每台主机上配置安全 shell (SSH)。

注意：此子命令等效于 `install-node` 命令。

如有必要，该子命令会从运行该子命令的安装中创建 Apusic Server 软件的 ZIP 存档。存档不包含 `domains` 目录或 `nodes` 目录。当创建和启动代表主机的节点上的实例时，这些目录与域管理服务器 (DAS) 同步。从指定主机上的存档安装 Apusic Server 软件后，该子命令不会删除存档。

如果指定多个主机，则以下各项的配置在所有主机上都相同：

- 基础安装目录
- SSH 端口
- SSH 用户
- SSH 密钥文件

如果要安装 Apusic Server 软件的主机上不存在 SSH 密钥文件，则子命令以交互方式运行并提示输入密码。要使子命令以非交互方式运行，必须满足以下条件：

- `asadmin`实用程序的 `--interactive` 选项必须是 `false`
- 该实用程序的 `--passwordfile` 选项 `asadmin` 必须指定一个密码文件。
- 密码文件必须包含该 `AS_ADMIN_SSHPASSWORD` 条目。

该子命令不会修改 DAS 的配置。

此子命令仅在本地模式下受支持。

### 2.3.1 概要

```
install-node-ssh [--help]
[--archive archive]
[--create={false|true}] [--save[={false|true}]]
[--installdir as-install-parent]
[--sshport ssh-port] [--sshuser ssh-user]
[--sshkeyfile ssh-keyfile]
[--force={false|true}]
host-list
```

### 2.3.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--archive
```

要安装的 Apusic Server 软件的存档文件的绝对路径。如果未指定存档文件，则子命令会从运行子命令的 Apusic Server 软件的安装中创建存档。此存档在运行该命令的用户的主目录中创建。

```
--create
```

指定子命令是否应创建要安装的 Apusic Server 软件的存档文件。

- `false`

不创建存档文件。该子命令从 `--archive` 选项指定的现有存档文件（默认）安装软件。

- `true`

该子命令从运行该子命令的 Apusic Server 软件的安装中创建一个存档文件。

```
--save
```

指定安装后是否保存安装软件的存档文件。

- `false`

存档文件未保存。该子命令在安装软件后删除文件（默认）。

- `true`

存档文件被保存。

```
--installdir
```

Apusic Server 软件将在每个主机上安装的基本安装目录的父级的绝对路径，例如 `/home/apusic/`。如果该目录不存在，则子命令将创建该目录。

运行此子命令的用户必须对指定目录具有写访问权。否则，会发生错误。

要覆盖 Apusic Server 软件的现有安装，请将 `--force` 选项设置为 `true`。如果目录已包含安装且 `--force` 选项为 `false`，则会发生错误。

默认值是运行此子命令的主机上 Apusic Server 软件的基本安装目录的父目录。

```
--sshport
```

用于 SSH 连接到要安装 Apusic Server 软件的主机的端口。默认值为 22。

```
--sshuser
```

要安装 Apusic Server 软件的主机上的用户，该用户将运行通过 SSH 连接到主机的过程。默认值是运行此子命令的用户。要确保 DAS 可以读取此用户的 SSH 私钥文件，请指定运行 DAS 进程的用户。

```
--sshkeyfile
```

`--sshuser` 选项指定的用户的 SSH 私钥文件的绝对路径。此文件用于对 `sshd` 主机上的守护程序进行身份验证。

运行此子命令的用户必须能够到达密钥文件的路径并读取密钥文件。

默认是用户 `.ssh` 目录中的密钥文件。如果找到多个密钥文件，则子命令使用以下优先顺序：

1. `id_rsa`
2. `id_dsa`
3. `identity`

```
--force
```

`--installdir` 指定子命令是否覆盖选项指定目录中现有的 Apusic Server 软件安装。可能的值如下：

- `false`

现有安装不会被覆盖（默认）。

- `true`

现有安装将被覆盖。

### 2.3.3 示例

示例 1：安装使用SSH连接的节点

此示例安装远程节点至服务器172.20.1.1。

```

asadmin> install-node-ssh 172.20.1.1
Enter remote password for root@172.20.140.17>
使用密码进行授权 <concealed>
Created installation zip /opt/ApusicAS/aas/bin/aas5352729830419748759.zip
使用密码进行授权 <concealed>
Copying /opt/ApusicAS/aas/bin/aas5352729830419748759.zip (417097301 bytes) to 172.20.1.1:/opt/ApusicAS
Installing aas5352729830419748759.zip into 172.20.1.1:/opt/ApusicAS
Removing 172.20.1.1:/opt/ApusicAS/aas5352729830419748759.zip
使用密码进行授权 <concealed>
Fixing file permissions of all bin files under 172.20.1.1:/opt/ApusicAS
Fixing file permissions for nadmin file under 172.20.1.1:/opt/ApusicAS/aas/lib
Fixing file permissions for async-profiler file under 172.20.1.1:/opt/ApusicAS/tools
Command install-node-ssh executed successfully.

```

## 2.4 创建节点-SSH

`create-node-ssh` 子命令创建一个节点，该节点可用于通过Secure Shell (SSH) 进行通信。

节点表示安装了 Apusic应用服务器的主机。Apusic应用服务器实例所在的每台主机都必须存在一个节点。

域管理服务器 (DAS) 通过 SSH 连接器联系 SSH 节点的主机，以管理驻留在节点上的 Apusic应用服务器实例。

默认情况下，如果 DAS 无法通过 SSH 联系节点的主机，则子命令将失败并且不会创建节点。要强制在 DAS 配置中创建节点，即使无法通过 SSH 联系主机，请将 `--force` 选项设置为true。

此子命令仅支持在远程模式下执行。

### 2.4.1 概要

```

create-node-ssh [--help]
                --nodehost node-host
                [--installdir as-install-parent] [--nodedir node-dir]
                [--sshport ssh-port] [--sshuser ssh-user] [--sshpassword ssh-password]
                [--sshkeyfile ssh-keyfile]
                [--force={false|true}]
                [--install={false|true}] [--archive archive]
                [--port<port (default:1099)>]
                node-name

```

### 2.4.2 选项

```

--help
--?

```

显示子命令的帮助文本。

```

--nodehost

```

节点代表的主机名。必须指定主机名。否则，会发生错误。

```

--installdir

```

主机上Apusic应用服务器基本安装目录的父目录的完整路径，例如 `/home/apusic/`。默认值是 DAS 的Apusic应用服务器默认基本安装目录的父目录，`${com.apusic.aas.productRoot}`。仅当Apusic应用服务器安装在所有主机上的相同位置时，此默认值才有用。

```

--noededir

```

在节点上创建的Apusic应用服务器实例的目录的路径。默认为 ``${com.apusic.aas.productRoot}/aas/nodes``，其中 ``${com.apusic.aas.productRoot}`` 是主机上 Apusic应用服务器的基本安装目录。

```
--sshport
```

用于与此节点主机的 SSH 连接的端口。默认值为 22。如果该 `--nodehost` 选项设置为 `localhost-mydomain`，`--sshport` 则忽略该选项。

```
--sshuser
```

此节点主机上的用户，该用户将运行通过 SSH 连接到主机的过程。默认值是运行 DAS 进程的用户。要确保 DAS 可以读取该用户的 SSH 私钥文件，请指定运行 DAS 进程的用户。如果该 `--nodehost` 选项设置为 `localhost-mydomain`，`--sshport` 则忽略该选项。

```
--sshkeyfile
```

`--sshuser` 选项指定的用户 SSH 私钥文件的绝对路径。此文件用于对节点主机上的 sshd 守护程序进行身份验证。

注意：

1. 应用服务器支持通过 SSH 用户密码连接，密码通过 `--sshpassword` 选项指定，二者选其一即可。
2. 应用服务器支持通过密码文件中的 `AS_ADMIN_SSHPASSWORD` 条目进行密码身份验证。密码文件在 `asadmin` 实用程序的 `--passwordfile` 选项中指定。

如果 SSH 私钥文件受密码保护，则密码文件必须包含 `AS_ADMIN_SSHKEYPASSPHRASE` 条目。

DAS 必须能够访问密钥文件的路径，并且密钥文件必须可由 DAS 读取。

默认情况下，这是用户 `.ssh` 目录中的密钥文件。如果找到多个密钥文件，子命令将使用以下优先顺序

1. `id_rsa`
2. `id_dsa`
3. `identity`

```
--sshpassword
```

使用 SSH 用户密码连接，与 `--sshkeyfile` 二者选其一连接即可。

```
--force
```

指定节点是否在 DAS 配置中创建，即使节点参数验证失败。要验证节点的参数，DAS 必须能够通过 SSH 联系节点的主机。可能的值如下：

- `false`  
如果节点参数验证失败（默认），则不会创建节点。
- `true`  
即使节点参数的验证失败，也会创建节点。

```
--install
```

指定子命令是否应在节点代表的主机上安装 Apusic Server 软件。

可能的值如下：

- `false`  
该子命令表示不需要在执行远程安装 Apusic Server 软件。设置为 `false` 的时候需要确保该节点的安装目录是否已经安装了 Apusic 应用服务器。默认为 `false`。
- `true`  
该子命令表示需要远程执行安装 Apusic Server 软件命令。

```
--archive
```

要安装的Apusic应用服务器的存档文件的绝对路径。如果省略此选项并且 `--install` 为true，则子命令将从运行此子命令的安装中创建Apusic应用服务器的ZIP存档。存档不包含域目录或节点目录。

```
node-name
```

节点名称。名称必须符合以下要求：

- 名称只能包含ASCII字符。
- 名称必须以字母、数字或下划线开头。
- 名称只能包含以下字符
- 该名称在域中必须是唯一的，并且不能是另一个节点、群集、命名配置或Apusic应用服务器实例的名称。
- 名称不能是域、服务器或Apusic应用服务器保留的任何其他关键字。

### 2.4.3 示例

示例 1：创建远程节点，使用SSH用户密码连接方法

此示例创建远程节点node128，节点主机为172.24.1.128，SSH用户为Apusicuser，SSH用户密码为apu，需要在节点上安装 Apusic Server 软件。

```
asadmin> create-node-ssh --nodehost 172.24.1.128 --sshuser Apusicuser --sshpassword apu --install true node128
```

示例2：使用SSH免密连接方式创建远程节点

此示例使用SSH免密方式创建远程节点node127，节点主机为172.24.1.127，远程节点的Apusic应用服务器安装目录为 `/opt/AAS`。

1. 设置 SSH 免密登录。

设置从 DAS 通过 SSH 免密登录到各节点所在服务器。

在 DAS 上执行以下命令：

```
asadmin> setup-ssh 172.24.1.127
```

2. 在节点上安装Apusic应用服务器。

在 DAS 上执行以下命令：

```
asadmin> install-node-ssh --installdir /opt/AAS 172.24.1.127
```

3. 创建节点node127。

在DAS上执行以下命令：

```
asadmin> create-node-ssh --nodehost 172.24.1.127 --installdir /opt/AAS node127
```

## 2.5 创建节点-config

`create-node-config` 子命令创建一个未启用远程通信的节点。通常用于创建本地节点。

节点表示安装了Apusic应用服务器软件的主机。Apusic应用服务器实例所在的每台主机都必须存在一个节点。

### 2.5.1 概要

```
create-node-config [--help]
[--nodehost node-host]
[--installdir as-install-parent] [--nodedir node-dir] node-name
```

### 2.5.2 选项

```
--help
--?
```

显示子命令的帮助文本。

```
--nodehost
```

节点代表的主机名。如果省略此选项，则不会为节点指定主机，建议必填。

```
--installdir
```

主机上Apusic应用服务器基本安装目录的父目录的完整路径，例如 `/home/apusic/`。默认值是 DAS 的Apusic应用服务器默认基本安装目录的父目录，`$(com.apusic.aas.productRoot)`。仅当Apusic应用服务器安装在所有主机上的相同位置时，此默认值才有用。

```
--nodedir
```

在节点上创建的Apusic应用服务器实例的目录的路径。默认为 `$(com.apusic.aas.productRoot)/aas/nodes`，其中 `$(com.apusic.aas.productRoot)` 是主机上Apusic应用服务器的基本安装目录。

```
node-name
```

节点名称。名称必须符合以下要求：

- 名称只能包含ASCII字符。
- 名称必须以字母、数字或下划线开头。
- 名称只能包含以下字符
- 该名称在域中必须是唯一的，并且不能是另一个节点、群集、命名配置或Apusic应用服务器实例的名称。
- 名称不能是域、服务器或Apusic应用服务器保留的任何其他关键字。

### 2.5.3 示例

示例 1：创建config通讯方式的节点mynode。

```
asadmin> create-node-config --nodehost 127.0.0.1 mynode
Enter admin password for user "admin">
Command create-node-config executed successfully.
```

## 2.6 删除节点-config

`delete-node-config` 子命令从域中删除未启用远程通信的节点。此子命令不需要配置分布式组件对象模型 (DCOM) 远程协议或安全外壳 (SSH)。

此子命令只能删除未启用远程通信的节点。删除其他类型节点的子命令取决于节点的类型：

- 已启用通过DCOM进行远程通信的节点必须使用 `delete-node-dcom` 子命令删除。
- 已启用通过SSH进行远程通信的节点必须使用 `delete-node-ssh` 子命令删除。

要确定是否已启用节点进行远程通信，请使用 `list-nodes` 子命令。

任何Apusic Server实例都不能驻留在正在删除的节点上。否则，子命令将失败。在运行此子命令之前，请使用 `delete-example` 子命令或 `delete-local-instance` 子命令删除驻留在节点上的所有实例。

### 2.6.1 概要

```
delete-node-config [--help] node-name
```

### 2.6.2 选项

```
--help
```

```
/?
```

显示子命令的帮助文本。

### 2.6.3 示例

示例 1：删除通讯方式为config的节点mynode。

```
asadmin> delete-node-config mynode
Command delete-node-config executed successfully.
```

## 2.7 删除节点-SSH

`delete-node-ssh` 子命令从域中删除一个启用了通过安全shell (SSH) 进行通信的节点。此子命令不需要配置SSH。

此子命令只能删除一个启用了通过SSH进行通信的节点。删除其他类型节点的子命令取决于节点的类型：

- 启用了通过分布式组件对象模型 (DCOM) 远程协议进行通信的节点必须使用 `delete-node-dcom` 子命令删除。
- 未启用远程通信的节点必须使用 `delete-node-config` 子命令删除。

要确定一个节点是否启用了通过SSH进行通信，请使用 `list-nodes` 子命令。

任何Apusic Server实例都不能驻留在正在删除的节点上。否则，子命令将失败。在运行此子命令之前，使用 `delete-example` 子命令或 `delete-local-instance` 子命令删除节点上的所有实例。

此子命令仅在远程模式下受支持。

### 2.7.1 概要

```
delete-node-ssh [--help]
[--uninstall={false|true}] [--force={false|true}]
node-name
```

### 2.7.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--uninstall
```

指定是否从节点所代表的主机上卸载 Apusic Server 软件。

可能的值如下：

- `false`

Apusic Server 软件不会从主机中卸载（默认）。

- `true`

Apusic Server 软件将从主机中卸载。默认情况下，如果预定义节点 `localhost-mydomain` 以外的任何节点驻留在要卸载 Apusic Server 软件的任何主机上，则子命令将失败。要从用户定义节点所在的主机卸载 Apusic Server 软件，请将 `--force` 选项设置为 `true`。如果 `--force` 选项是 `true`，则子命令将删除基本安装目录的父目录的全部内容。

```
--force
```

如果 `--uninstall` 为 `true`，则指定子命令是否从主机卸载 Apusic Server 软件，即用户定义的节点驻留在主机上也是如此。可能的值如下：

- `false`

如果用户定义的节点驻留在主机上，则不会卸载软件并且子命令会失败（默认）。如果 `--force` 选项是 `false`，则子命令仅删除 Apusic Server 软件文件。如果基本安装目录的父目录（例如配置文件）未删除，则其他内容。

- `true`

即使用户定义的节点驻留在主机上，该子命令也会从主机卸载 Apusic Server 软件。如果 `--force` 选项是 `true`，则子命令将删除基本安装目录的父目录的全部内容。

### 2.7.3 示例

示例1:删除已启用SSH通信的节点此示例

删除已启用SSH通信的节点egnode1。

```
asadmin> delete-node-ssh egnode1
```

```
Command delete-node-ssh executed successfully.
```

## 2.8 卸载节点

`uninstall-node` 子命令从指定为该子命令属性值的主机中卸载 Apusic Server 软件。此子命令需要在运行子命令的主机和正在卸载 Apusic Server 软件的每个主机上配置安全 shell (SSH)。

注意：此子命令等效于 `uninstall-node-ssh` 子命令。要在使用分布式组件对象模型 (DCOM) 远程协议的主机上卸载 Apusic Server 软件，请使用 `uninstall-node-dcom` 子命令。

默认情况下，如果预定义节点 `localhost-mydomain` 以外的任何节点驻留在要卸载 Apusic Server 软件的任何主机上，则子命令将失败。要从用户定义节点所在的主机卸载 Apusic Server 软件，请将 `--force` 选项设置为 `true`。如果 `--force` 选项是 `true`，则子命令将删除基本安装目录的父目录的全部内容。

如果基本安装目录的父目录下的文件已打开，则子命令将失败。

如果指定多个主机，则以下各项的配置在所有主机上必须相同：

- Apusic Server 软件的基本安装目录的父级
- SSH 端口
- SSH 用户
- SSH 密钥文件

该子命令不会修改域管理服务器 (DAS) 的配置。

此子命令仅在本地模式下受支持。

### 2.8.1 概要

```
uninstall-node [--help]
[--installdir as-install-parent]
[--sshport ssh-port] [--sshuser ssh-user]
[--sshkeyfile ssh-keyfile]
[--force={false|true}]
host-list
```

### 2.8.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--installdir
```

每个主机上安装 Apusic Server 软件的基本安装目录的父目录的绝对路径，例如 `/home/apusic`。

运行此子命令的用户必须对指定目录具有写访问权。否则，会发生错误。

指定目录必须包含主机上 Apusic Server 软件的安装。否则，会发生错误。

默认值是运行此子命令的主机上 Apusic Server 软件的基本安装目录的父目录。

```
--sshport
```

用于 SSH 连接到要卸载 Apusic Server 软件的主机的端口。默认值为 22。

```
--sshuser
```

要卸载 Apusic Server 软件的主机上的用户，该用户将运行通过 SSH 连接到主机的进程。默认值是运行此子命令的用户。要确保 DAS 可以读取此用户的 SSH 私钥文件，请指定运行 DAS 进程的用户。

```
--sshkeyfile
```

`--sshuser` 选项指定的用户的 SSH 私钥文件的绝对路径。此文件用于对 `sshd` 主机上的守护程序进行身份验证。

运行此子命令的用户必须能够到达密钥文件的路径并读取密钥文件。

默认是用户 `.ssh` 目录中的密钥文件。如果找到多个密钥文件，则子命令使用以下优先顺序：

1. `id_rsa`
2. `id_dsa`
3. `identity`

```
--force
```

指定子命令是否从主机卸载 Apusic Server 软件，即使用户定义的节点驻留在主机上也是如此。可能的值如下：

- `false`

如果用户定义的节点驻留在主机上，则不会卸载软件并且子命令会失败（默认）。如果 `--force` 选项是 `false`，则子命令仅删除 Apusic Server 软件文件。如果基本安装目录的父目录（例如配置文件）未删除，则其他内容。

- `true`

即使用户定义的节点驻留在主机上，该子命令也会从主机卸载 Apusic Server 软件。如果 `--force` 选项是 `true`，则子命令将删除基本安装目录的父目录的全部内容。

```
host-list
```

需要卸载的节点主机名；多个以空格分隔。

### 2.8.3 示例

示例 1：卸载节点

卸载主机 172.2.1.2 上的 Apusic 应用服务器。

```
asadmin> uninstall-node 172.2.1.2
Command uninstall-node executed successfully.
```

## 2.9 卸载节点-SSH

`uninstall-node-ssh` 从指定为子命令属性值的主机上卸载 Apusic 应用服务器软件。此子命令需要在运行子命令的主机和正在卸载 Apusic Server 软件的每个主机上配置安全 shell (SSH)。

注意：

- 此子命令等效于 `uninstall-node` 子命令。

默认情况下，如果除预定义节点 `localhost-mydomain` 之外的任何节点驻留在卸载 Apusic 应用服务器软件的任何主机上，则子命令将失败。要从用户定义节点所在的主机上卸载 Apusic 应用服务器软件，请将 `--force` 选项设置为 `true`。如果 `--force` 选项为真子命令删除基本安装目录父目录的全部内容。如果基本安装目录父目录下的文件已打开，则子命令将失败。如果指定了多个主机，则以下项目的配置必须在所有主机上相同：■ `{product-name}` 软件的基本安装目录的父目录 ■ SSH 端口 ■ SSH 用户 ■ SSH 密钥文件 该子命令不会修改域管理服务器 (DAS) 的配置。此子命令仅在本地模式下受支持。

### 2.9.1 概要

```
uninstall-node-ssh [--help]
[--installdir as-install-parent]
[--sshport ssh-port] [--sshuser ssh-user]
[--sshkeyfile ssh-keyfile]
[--force={false|true}]
host-list
```

### 2.9.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--installdir
```

每个主机上安装 Apusic Server 软件的基本安装目录的父目录的绝对路径，例如 `/home/apusic/`。

运行此子命令的用户必须对指定目录具有写访问权。否则，会发生错误。

指定目录必须包含主机上 Apusic Server 软件的安装。否则，会发生错误。

默认值是运行此子命令的主机上 Apusic Server 软件的基本安装目录的父目录。

```
--sshport
```

用于 SSH 连接到要卸载 Apusic Server 软件的主机的端口。默认值为 22。

```
--sshuser
```

要卸载 Apusic Server 软件的主机上的用户，该用户将运行通过 SSH 连接到主机的进程。默认值是运行此子命令的用户。要确保 DAS 可以读取此用户的 SSH 私钥文件，请指定运行 DAS 进程的用户。

```
--sshkeyfile
```

`--sshuser` 选项指定的用户的 SSH 私钥文件的绝对路径。此文件用于对 `sshd` 主机上的守护程序进行身份验证。

运行此子命令的用户必须能够到达密钥文件的路径并读取密钥文件。

默认是用户 `.ssh` 目录中的密钥文件。如果找到多个密钥文件，则子命令使用以下优先顺序：

1. `id_rsa`
2. `id_dsa`
3. `identity`

```
--force
```

指定子命令是否从主机卸载 Apusic Server 软件，即使用户定义的节点驻留在主机上也是如此。可能的值如下：

- `false`

如果用户定义的节点驻留在主机上，则不会卸载软件并且子命令会失败（默认）。如果 `--force` 选项是 `false`，则子命令仅删除 Apusic Server 软件文件。如果基本安装目录的父目录（例如配置文件）未删除，则其他内容。

- `true`

即使用户定义的节点驻留在主机上，该子命令也会从主机卸载 Apusic Server 软件。如果 `--force` 选项是 `true`，则子命令将删除基本安装目录的父目录的全部内容。

```
host-list
```

需要卸载的节点主机名；多个以空格分隔。

### 2.9.3 示例

示例 1：卸载远程节点

卸载远程节点

```
asadmin> uninstall-node-ssh 172.2.1.2
Command uninstall-node-ssh executed successfully.
```

## 2.10 ping节点

`ping-node-ssh` 子命令测试通过安全shell (SSH) 启用通信的节点是否可用。此子命令要求在运行域管理服务器 (DAS) 的机器和节点所在的机器上配置安全shell (SSH)。

此子命令仅在远程模式下受支持。

### 2.10.1 概要

```
ping-node-ssh [--help]
[--validate={false|true}] node-name
```

### 2.10.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--validate
```

指定子命令是否验证节点。默认为false。

```
node-name
```

要测试的节点名称。该节点必须启用通过SSH进行通信的功能。否则，将会出现错误。

### 2.10.3 示例

示例1: 测试SSH节点是否可以连接

该示例测试节点node1是否可以连接。

```
asadmin> ping-node-ssh node1
Successfully made SSH connection to node node1
Command ping-node-ssh executed successfully.
```

示例2: 验证SSH节点是否可连接

该示例验证节点node2是否可以连接。

```
asadmin> ping-node-ssh --validate=true node17
成功地创建 SSH 连接到 node17 (172.1.1.1)
在 /opt/tApusicAS:
Using locally retrieved version string from version class.
Version = Apusic Application Server 10.0.8 EE.SP8.ST. (build 202503191441)
Command version executed successfully. 发现AAS版本
Command ping-node-ssh executed successfully.
```

## 3 实例

### 3.1 查看实例

`list-instances` 子命令列出域中的Apusic应用服务器实例。该列表可以按集群、实例、节点或配置进行筛选。默认情况下，子命令指示列出的每个实例是否正在运行。此子命令的选项控制每个实例显示的信息。

此子命令仅在远程模式下受支持。

#### 3.1.1 概要

```
list-instances [--help]
[--timeoutmsec timeout]
[--long={false| true} | --nostatus={false| true}]
[--standaloneonly={false| true} | target]
```

#### 3.1.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--timeoutmsec
```

确定实例状态的时间限制（毫秒）。默认值为2000，相当于2秒。

```
--long
```

指定是否为列出的每个实例显示详细信息。 `--long` 选项和 `--nostatus` 选项是互斥的。如果在同一命令中指定了这两个选项，则会发生错误。有效值如下：

- true

列出的每个实例都显示了以下详细信息：

- 实例的名称
- 实例节点所在的主机名
- 实例监听管理请求的HTTP端口
- 实例进程的进程标识符（PID），如果实例未运行，则为-1
- 实例所属集群的名称（如果有的话）
- 实例的状态，正在运行或未运行 当一个实例被列出时，该实例在域管理服务器（DAS）中的一些配置更改可能尚未应用于实例本身。在这种情况下，应用这些更改所需的命令会列在实例状态旁边。为一个实例列出的命令最大数量是10。

- false

仅显示实例的名称和实例是否正在运行的指示（默认）。不显示实例运行的时间长度。

```
--nostatus
```

指定是否控制有关实例是否正在运行的信息。 `--long` 选项和 `--nostatus` 选项是互斥的。如果在同一命令中指定了这两个选项，则会发生错误。有效值如下：

- true

有关实例是否正在运行的信息被抑制。仅显示每个实例的名称。

- false

显示有关实例是否正在运行的信息（默认）。

**--standalonly**

指定是否仅列出独立实例。 `--standalonly` 选项和目标属性值是互斥的。如果在同一命令中同时指定了 `--standalonly` 选项和目标属性值，则会发生错误。有效值如下：

- true  
仅列出独立实例。
- false  
将列出指定目标中的所有实例（默认）。

**--target**

通过指定列出实例的目标来筛选 {product--name} 实例列表。目标属性值和 `--standalonly` 选项是互斥的。如果在同一命令中同时指定了目标属性值和 `--standalonly` 选项，则会发生错误。有效值如下：

- domain  
列出域中的所有实例（默认）。
- cluster-name  
列出属于指定集群的实例。
- instance-name  
仅列出指定的实例。
- node-name  
列出驻留在指定节点上的实例。
- configuration-name  
列出其配置由指定命名配置定义的所有实例。

**3.1.3 示例**

示例1: 查询域中所有实例

```
asadmin> list-instances
pmd-i-sj02 running
yml-i-sj01 running
pmdsa1 not running
Command list-instances executed successfully.
```

**3.2 创建实例**

`create-instance` 子命令用于创建一个 AAS实例。此子命令要求在运行域管理服务器（DAS）的主机和实例所在节点所代表的主机上配置分布式组件对象模型（DCOM）远程协议或安全外壳（SSH）。

注意：如果实例驻留在代表本地主机的CONFIG类型的节点上，则不需要DCOM或SSH。CONFIG类型的节点未启用通过DCOM或SSH进行远程通信。

可以从任何可以联系DAS的主机上运行此命令。Apusic Server实例是运行Apusic Server的单个节点上的Java平台（Java虚拟机或JVM机器）的单个虚拟机。节点定义Apusic Server实例所在的主机。JVM机器必须与Java平台，企业版（JavaEE）兼容。

Apusic Server实例需要对以下项目的引用：

- 定义实例所在主机的节点。必须在创建实例的命令中指定节点。
- 定义实例配置的命名配置。可以在创建实例的命令中指定配置，但不是必需的。如果没有为未加入集群的实例指定配置，则子命令为实例创建配置。加入集群的实例从其父集群接收其配置。

每个Apusic Server实例都是以下类型的实例之一：

## 独立实例

独立实例不会与任何其他实例或集群共享其配置。如果满足以下任一条件，则创建独立实例：

- 在创建实例的命令中没有指定配置或集群。
- 在创建实例的命令中指定了一个未被任何其他实例或集群引用的配置。

当未指定配置或集群时，将为实例创建 `default-config` 配置的副本。此配置的名称 `instance-name-config`，其中 `instance-name` 表示未集群服务器实例的名称。。

**共享实例** 共享实例与其他实例或集群共享其配置。如果在创建实例的命令中指定了其他实例或集群引用的配置，则创建共享实例。

## 集群实例

集群实例从实例所属的集群继承其配置，并与集群中的其他实例共享其配置。如果在创建实例的命令中指定了集群，则创建集群实例。任何不属于集群的实例都被视为非集群服务器实例。因此，独立实例和共享实例是非集群服务器实例。

默认情况下，此子命令会尝试为正在创建的实例解决可能的端口冲突。该子命令还会分配当前未使用且尚未分配给同一节点上其他实例的端口。该子命令根据子命令内部的算法分配这些端口。使用 `--systemproperties` 选项解决同一节点上其他实例的端口冲突。可以使用 `create-system-properties` 子命令和 `delete-system-property` 子命令来操作实例的系统属性。仅在远程模式下支持此子命令。

### 3.2.1 概要

```
create-instance [--help] --node node-name
                [--config config-name | --cluster cluster-name]
                [--lbenabled={true|false}]
                [--portbase=port-number] [--checkports={true|false}]
                [--systemproperties (name=value) [:name=value]* ]
                instance-name
```

### 3.2.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--node
```

定义要在其中创建实例的主机的节点的名称。该节点必须已经存在。如果要在运行域管理服务器 (DAS) 的主机上创建实例，请使用预定义的节点 `localhost- domain`。

```
--config
```

指定实例引用的命名配置。配置必须存在且不得命名为 `default-config` 或 `server-config`。指定该 `--config` 选项会创建一个共享实例。

该 `--config` 选项与 `--cluster` 选项是互斥的。如果省略这两个选项，则会创建一个独立实例。

```
--cluster
```

指定实例从中继承其配置的集群。指定该 `--cluster` 选项会创建一个集群实例。

该 `--config` 选项与 `--cluster` 选项是互斥的。如果省略这两个选项，则会创建一个独立实例。

```
--lbenabled
```

指定是否为实例启用负载均衡。可能的值如下：

- `true`  
该实例已启用负载均衡（默认）。当实例启用负载均衡时，负载均衡器会向实例发送请求。
- `false`  
该实例被禁用以进行负载均衡。当实例被禁用负载均衡时，负载均衡器不会向实例发送请求。

**--portbase**

确定端口分配应开始的编号。一个实例使用一定数量的静态分配的端口。该`portbase`值决定了分配应该开始。端口的值计算如下：

- 管理端口:  $portbase + 48$
- HTTP 监听端口:  $portbase + 80$
- HTTPS监听端口:  $portbase + 81$
- JMS 端口:  $portbase + 76$
- IIOP 侦听器端口:  $portbase + 37$
- 安全 IIOP 侦听器端口:  $portbase + 38$
- 具有相互身份验证端口的安全 IIOP:  $portbase + 39$
- JMX 端口:  $portbase + 86$
- JPA 调试器端口:  $portbase + 9$
- 用于 OSGi 模块管理的 Felix shell 服务端点:  $portbase + 66$

当 `--portbase` 指定选项，该子命令的输出包括使用的端口的完整列表。

**--checkports**

指定是否检查管理、HTTP、JMS、JMX 和 IIOP 端口的可用性。默认值为 `true`。

**--systemproperties**

定义实例的系统属性。这些属性会覆盖实例配置中端口设置的属性定义。例如，如果两个集群实例驻留在同一主机上，则必须覆盖预定义的端口设置。在这种情况下，必须覆盖一个实例的端口设置，因为两个实例共享相同的配置。

以下属性可用：

- `ASADMIN_LISTENER_PORT`

该属性指定DAS连接到实例以管理实例的HTTP端口或HTTPS端口的端口号。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。

- `HTTP_LISTENER_PORT`

此属性指定用于侦听 HTTP 请求的端口号。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。

- `HTTP_SSL_LISTENER_PORT`

此属性指定用于侦听 HTTPS 请求的端口的端口号。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。

- `IIOP_LISTENER_PORT`

此属性指定用于 IIOP 连接的端口号。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。

- `IIOP_SSL_LISTENER_PORT`

此属性指定用于安全 IIOP 连接的端口号。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。

- `IIOP_SSL_MUTUALAUTH_PORT`

此属性指定用于通过客户端身份验证进行安全 IIOP 连接的端口的端口号。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。

- `JAVA_DEBUGGER_PORT`

此属性指定用于连接到 Java 平台调试器体系结构 (JPDA) 调试器的端口号。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。

- `JMS_PROVIDER_PORT`

此属性指定 Java 消息服务提供者的端口号。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。

- `JMX_SYSTEM_CONNECTOR_PORT`

此属性指定 JMX 连接器侦听的端口号。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。

- `OSGI_SHELL_TELNET_PORT`

此属性指定用于连接到 Apache Felix Remote Shell 的端口号。这个shell 使用Felix shell 服务与OSGi 模块管理子系统进行交互。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限

```
instance-name
```

定义创建的实例的名称。

### 3.2.3 示例

示例 1：创建服务器独立实例

此示例在节点 localhost-mydomain 创建独立实例 insloc。

```
asadmin> create-instance --node localhost-mydomain insloc
Port Assignments for server instance pmdsainst:
JMX_SYSTEM_CONNECTOR_PORT=28688
JMS_PROVIDER_PORT=27678
ASADMIN_LISTENER_PORT=24850
HTTP_LISTENER_PORT=28082
IIOP_LISTENER_PORT=23702
IIOP_SSL_LISTENER_PORT=23822
HTTP_SSL_LISTENER_PORT=28183
IIOP_SSL_MUTUALAUTH_PORT=23922

Command create-instance executed successfully.
```

示例 2：创建具有自定义端口分配的独立服务器实例

此示例在节点 localhost-mydomain 创建独立实例 myinsloc。自定义端口号分配给以下端口：

- HTTP 侦听器端口
- HTTPS 监听端口
- IIOP 连接端口
- 安全 IIOP 连接端口
- 具有相互身份验证的安全 IIOP 连接端口
- JMX 连接器端口

```
asadmin> create-instance --node localhost-mydomain
--systemproperties HTTP_LISTENER_PORT=58294:
HTTP_SSL_LISTENER_PORT=58297:
IIOP_LISTENER_PORT=58300:
IIOP_SSL_LISTENER_PORT=58303:
IIOP_SSL_MUTUALAUTH_PORT=58306:
JMX_SYSTEM_CONNECTOR_PORT=58309 myinsloc
Port Assignments for server instance pmdcpinst:
JMS_PROVIDER_PORT=27679
ASADMIN_LISTENER_PORT=24851

Command create-instance executed successfully.
```

示例 3：创建共享服务器实例

此示例在节点 localhost-mydomain 创建共享实例 insshare。此实例的共享配置是 insshareconfig。

```
asadmin create-instance --node localhost-mydomain --config insshareconfig
insshare
Port Assignments for server instance insshare:
JMX_SYSTEM_CONNECTOR_PORT=28687
```

```
JMS_PROVIDER_PORT=27677
ASADMIN_LISTENER_PORT=24849
HTTP_LISTENER_PORT=28081
IIOP_LISTENER_PORT=23701
IIOP_SSL_LISTENER_PORT=23821
HTTP_SSL_LISTENER_PORT=28182
IIOP_SSL_MUTUALAUTH_PORT=23921
```

```
Command create-instance executed successfully.
```

#### 示例 4：创建服务器集群实例

此示例在节点 `localhost-mydomain` 创建集群实例 `insclus`。该实例属于集群 `mycluster`。

```
asadmin> create-instance --node localhost-mydomain --cluster mycluster insclus
Port Assignments for server instance insclus:
JMX_SYSTEM_CONNECTOR_PORT=28686
JMS_PROVIDER_PORT=27676
HTTP_LISTENER_PORT=28080
ASADMIN_LISTENER_PORT=24848
IIOP_SSL_LISTENER_PORT=23820
IIOP_LISTENER_PORT=23700
HTTP_SSL_LISTENER_PORT=28181
IIOP_SSL_MUTUALAUTH_PORT=23920
```

```
Command create-instance executed successfully.
```

### 3.3 创建本地实例

`create-local-instance` 子命令在代表运行该子命令的主机的节点上创建一个 `{product--name}` 实例。此子命令不需要配置 secure shell (SSH)。

需要从实例将驻留的节点所代表的主机运行此子命令。要联系域管理服务器 (DAS)，此子命令需要运行 DAS 的主机的名称。如果使用非默认端口进行管理，则此子命令还需要端口号。

#### 3.3.1 概要

```
create-local-instance [--help]
[--node node-name] [--nodedir node-dir]
[--config config-name | --cluster cluster-name]
[--lbenabled={true|false}]
[--portbase port-number] [--checkports={true|false}]
[--savemasterpassword={false|true}]
[--usemasterpassword={false|true}]
[--systemproperties (name=value) [:name=value]* ]
instance-name
```

#### 3.3.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--node
```

定义要在其中创建实例的主机的节点的名称。仅当存储节点的目录中存在多个节点时，才必须指定节点。否则，可以省略节点。如果指定了节点，则该节点必须存在。

如果未指定节点，则子命令的行为取决于存储节点的目录中现有节点的数量：

- 如果不存在节点，则子命令会为实例创建一个节点。节点的名称是运行子命令的主机的名称。
- 如果仅存在一个节点，则子命令会为实例创建对现有节点的引用。
- 如果存在两个或更多节点，则会发生错误。

```
--nodedir
```

要存储实例节点文件的目录的路径。默认为`\${aas-install}/nodes`。

```
--config
```

指定实例引用的命名配置。配置必须存在，不得命名为 `default-config` 或 `server-config`。指定 `--config` 选项会创建一个共享实例。

`--config` 期权和 `--cluster` 期权是互斥的。如果省略这两个选项，则会创建一个独立实例。

```
--cluster
```

指定实例从中继承其配置的集群。指定该 `--cluster` 选项会创建一个集群实例。

`--config` 期权和 `--cluster` 期权是互斥的。如果省略这两个选项，则会创建一个独立实例。

```
--lbenabled
```

指定是否为实例启用负载均衡。可能的值如下：

- `true`  
该实例已启用负载均衡（默认）。当实例启用负载均衡时，负载均衡器会向实例发送请求。
- `false`  
该实例被禁用以进行负载均衡。当实例被禁用负载均衡时，负载均衡器不会向实例发送请求。

```
--portbase
```

确定端口分配应开始的编号。一个实例使用一定数量的静态分配的端口。\* `portbase` \*值确定分配应该从哪里开始。端口的值计算如下：

- 管理端口: `portbase + 48`
- HTTP 监听端口: `portbase + 80`
- HTTPS监听端口: `portbase + 81`
- JMS 端口: `portbase + 76`
- IIOP 侦听器端口: `portbase + 37`
- 安全 IIOP 侦听器端口: `portbase + 38`
- 具有相互身份验证端口的安全 IIOP: `portbase + 39`
- JMX 端口: `portbase + 86`
- JPA 调试器端口: `portbase + 9`
- 用于 OSGi 模块管理的 Felix shell 服务端口: `portbase + 66`

指定选项时 `--portbase`，此子命令的输出包括已用端口的完整列表。

```
--checkports
```

指定是否检查管理、HTTP、JMS、JMX 和 IIOP 端口的可用性。默认值为 `true`。

```
--savemasterpassword
```

将此选项设置为 `true` 允许将主密码写入文件系统。如果将主密码写入文件系统，则无需提示输入密码即可启动实例。如果此选项为 `true`，则该 `--usemasterpassword` 选项也为真，无论命令行上指定的值如何。因为将主密码写入文件系统是不安全的做法，所以默认为 `false`。

实例的主密码文件保存在节点目录中，而不是域目录中。因此，只有为域中的每个节点创建的第一个实例才需要此选项。

**--usemasterpassword**

指定是使用系统内置的主密码还是用户定义的主密码对密钥库进行加密。

如果 `false`（默认），则使用系统内置的众所周知的密码对密钥库进行加密。使用系统内置的密码加密密钥库不会提供额外的安全性。

如果 `true`，则子命令从 `AS_ADMIN_MASTERPASSWORD` 密码文件中的条目中获取主密码或提示输入主密码。密码文件在 [asadmin\(1M\)](#) 实用程序的 `--passwordfile` 选项中指定。

如果 `--savemasterpassword` 选项为 `true`，则此选项也为真，无论命令行上指定的值如何。

域中所有实例的主密码必须相同。

**--systemproperties**

定义实例的系统属性。这些属性会覆盖实例配置中端口设置的属性定义。例如，如果两个集群实例驻留在同一主机上，则必须覆盖预定义的端口设置。在这种情况下，必须覆盖一个实例的端口设置，因为两个实例共享相同的配置。

以下属性可用：

- `ASADMIN_LISTENER_PORT`

该属性指定DAS连接到实例以管理实例的HTTP端口或HTTPS端口的端口号。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。

- `HTTP_LISTENER_PORT`

此属性指定用于侦听 HTTP 请求的端口号。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。

- `HTTP_SSL_LISTENER_PORT`

此属性指定用于侦听 HTTPS 请求的端口的端口号。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。

- `IIOP_LISTENER_PORT`

此属性指定用于 IIOP 连接的端口号。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。

- `IIOP_SSL_LISTENER_PORT`

此属性指定用于安全 IIOP 连接的端口号。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。

- `IIOP_SSL_MUTUALAUTH_PORT`

此属性指定用于通过客户端身份验证进行安全 IIOP 连接的端口的端口号。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。

- `JAVA_DEBUGGER_PORT`

此属性指定用于连接到 Java 平台调试器体系结构 (JPDA) 调试器的端口号。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。

- `JMS_PROVIDER_PORT`

此属性指定 Java 消息服务提供者的端口号。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。

- `JMX_SYSTEM_CONNECTOR_PORT`

此属性指定 JMX 连接器侦听的端口号。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。

- `OSGI_SHELL_TELNET_PORT`

此属性指定用于连接到 Apache Felix Remote Shell 的端口号。这个 shell 使用 Felix shell 服务与 OSGi 模块管理系统进行交互。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。

**instance-name**

定义创建的实例的名称。

**3.3.3 示例**

示例 1 创建独立 Apusic 应用服务器实例

此示例在运行命令的主机上创建独立实例 `ins21`。DAS 在同一主机上运行。该实例引用唯一的现有节点。

```
asadmin> create-local-instance ins21
Rendezvoused with DAS on localhost:4848. Port Assignments for server instance ins21:
JMX SYSTEM CONNECTOR PORT=28686
JMS PROVIDER PORT=27676
HTTP LISTENER PORT=28080
ASADMIN LISTENER PORT=24848
JAVA DEBUGGER PORT=29009
IIOP SSL LISTENER PORT=23820
IIOP LISTENER PORT=23700
OSGI SHELL TELNET PORT=26666
HTTP SSL LISTENER PORT=28181
IIOP SSL MUTUALAUTH PORT=23920
Command create-local-instance executed successfully.
```

### 3.4 启动实例

`start-instance` 子命令启动Apusic应用服务器实例。该子命令要求在运行域管理服务器（DAS）的机器和实例所在的机器上配置分布式组件对象模型（DCOM）远程协议或安全外壳（SSH）。

#### 3.4.1 概要

```
start-instance [--help]
               [--debug={false|true}] [--sync={normal|full|none}]
               instance-name
```

#### 3.4.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--debug
```

指定实例是否在启用Java 平台调试器体系结构 (JPDA)调试的情况下启动。

可能的值如下：

- `true`  
实例在启用 JPDA 调试的情况下启动，并显示用于 JPDA 调试的端口号。
- `false`  
该实例在禁用 JPDA 调试的情况下启动（默认）。

```
--sync
```

实例启动时 DAS 与实例文件之间的同步类型。

可能的值如下：

- `none`  
DAS 不会将实例的文件与任何更改同步。这种类型的同步最大限度地减少了启动实例所需的时间。
- `normal`

DAS 将实例与自上次同步以来的更改同步，如下所示：对于 `config` 目录，DAS 将实例与所有更改同步。对于 `applications` 目录和 `docroot` 目录，只有对顶级子目录的更改会导致 DAS 同步该子目录下的所有文件。如果更改了顶级子目录下的文件而没有更改顶级子目录中的文件，则需要完全同步。在正常操作中，这些目录的顶级子目录下的文件不会更改。如果应用程序已部署和取消部署，则无需完全同步即可使用更改更新实例。此值为默认值。

- `full`

DAS 将实例与实例的所有文件同步，无论文件自上次同步后是否已更改。当 DAS 更新实例目录中的所有文件时，这种类型的同步可能会延迟实例的启动。

```
instance-name
```

需要启动的实例的名称，该实例名称需要存在才能启动成功。

### 3.4.3 示例

示例 1：启动实例ins21

此示例启动实例ins21。

```
asadmin> start-instance ins21
Waiting for the server to start .....
Successfully started the instance: ins21
instance Location: /export/Apusic3/Apusic/nodes/localhost/ins21
Log File: /export/Apusic3/Apusic/nodes/localhost/ins21/logs/server.log
Admin Port: 26848
Command start-local-instance executed successfully.
The instance, ins21, was started on host localhost
Command start-instance executed successfully.
```

示例 2：Debug模式下启动insloc

此示例在启用JPA调试的情况下启动[product--name]实例insloc

```
asadmin> start-instance --debug=true insloc
Waiting for the server to start .....
Successfully started the instance: insloc
instance Location: /export/Apusic3/Apusic/nodes/localhost/insloc
Log File: /export/Apusic3/Apusic/nodes/localhost/insloc/logs/server.log
Admin Port: 26849
Debugging is enabled. The debugging port is: 28000
Command start-local-instance executed successfully.
The instance, insloc, was started on host localhost
Command start-instance executed successfully.
```

## 3.5 启动本地实例

`start-local-instance` 子命令在运行该子命令的主机上启动一个 Apusic 应用服务器 实例。此子命令不需要配置secure shell (SSH)。必须从实例所在的主机运行此命令。

该子命令可以启动所有 Apusic Server 实例。

在本地模式下支持此子命令。但是，要将实例与 DAS 同步，此子命令必须在远程模式下运行。

### 3.5.1 概要

```
start-local-instance [--help]
                    [--nodedir node-dir] [--node node]
                    [--debug={false|true}] [--dry-run={true|false}]
                    [--sync={normal|full|none}]
                    [--verbose={false|true}] [--watchdog={true|false}]
                    [instance-name]
```

### 3.5.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--nodedir
```

指定包含实例节点目录的目录。实例的文件存储在实例的节点目录中。默认为 `as-install/nodes`。

```
--node
```

指定实例所在的节点。

```
--debug
```

指定实例是否在启用[Java 平台调试器体系结构 (JPDA)调试的情况下启动。

可能的值如下：

- `true`  
实例在启用 JPDA 调试的情况下启动，并显示用于 JPDA 调试的端口号。
- `false`  
该实例在禁用 JPDA 调试的情况下启动（默认）。

```
--sync
```

实例启动时 DAS 与实例文件之间的同步类型。

可能的值如下：

- `none`  
DAS 不会将实例的文件与任何更改同步。这种类型的同步最大限度地减少了启动实例所需的时间。
- `normal`  
DAS 将实例与自上次同步以来的更改同步，如下所示：对于 `config` 目录，DAS 将实例与所有更改同步。对于 `applications` 目录和 `docroot` 目录，只有对顶级子目录的更改会导致 DAS 同步该子目录下的所有文件。如果更改了顶级子目录下的文件而没有更改顶级子目录中的文件，则需要完全同步。在正常操作中，这些目录的顶级子目录下的文件不会更改。如果应用程序已部署和取消部署，则无需完全同步即可使用更改更新实例。此值为默认值。
- `full`  
DAS 将实例与实例的所有文件同步，无论文件自上次同步后是否已更改。当 DAS 更新实例目录中的所有文件时，这种类型的同步可能会延迟实例的启动。警告：如果 DAS 未运行或无法从运行此子命令的主机访问，请不要将 `--sync` 选项设置为 `full`。要执行完全同步，子命令会删除实例的缓存。如果无法联系 DAS 以替换缓存，则子命令将失败并且实例无法重新启动，直到它与 DAS 重新同步。

```
--verbose
```

```
-v
```

指定是否在运行子命令的控制台窗口中显示有关实例的详细信息。

可能的值如下：

- `true`  
有关实例的详细启动消息和日志消息显示在运行子命令的控制台窗口中。如果稍后通过从不同的控制台窗口运行 `restart-local-instance` 子命令重新启动实例，则消息将继续显示在原始控制台窗口中。

可以通过在控制台窗口中键入 `CTRL-C` 来终止Apusic Server进程。

可以通过在控制台窗口中键入以下组合键之一来终止Apsuic Server进程并获取服务器的线程转储：

1. UNIX系统上使用 `CTRL-\`
2. Windows系统上使用 `CTRL-Break`

- `false`

不显示详细信息（默认）。

```
instance-name
```

需要启动的实例的名称，该实例名称需要存在才能启动成功。

### 3.5.3 示例

示例 1：启动节点的实例

此示例启动在节点node21上的实例myins21。

```
asadmin> start-local-instance --node node21 myins21
Waiting for the server to start .....
Successfully started the instance: myins21
instance Location: /export/Apusic3/Apusic/nodes/node21/myins21
Log File: /export/Apusic3/Apusic/nodes/node21/myins21/logs/server.log
Admin Port: 26849
Command start-local-instance executed successfully.
```

## 3.6 停止实例

`stop-instance` 子命令停止正在运行的 Apusic Server 实例。

此命令仅在远程模式下受支持。

### 3.6.1 概要

```
stop-instance [--help]
[--force={false|true}] [--kill={false|true}]
instance-name delete-instance [--help] instance-name
```

### 3.6.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--force
```

指定是否立即强制停止实例。

可能的值如下：

- `true`

立即强制停止实例（默认）。

- `false`

该子命令会一直等待，直到与该实例关联的所有线程都退出，然后再停止该实例。

```
--kill
```

指定是否通过使用操作系统的功能终止实例进程来终止实例。

可能的值如下：

- `false`

实例没有被杀死。该子命令使用 Java 平台的功能来终止实例进程（默认）。

- `true`

实例被杀死。该子命令使用操作系统的功能来终止实例进程。

```
instance-name
```

需要停止的实例名称。

### 3.6.3 示例

示例 1：停止实例myins21。

```
asadmin> stop-instance myins21
The instance, myins21, was stopped.
Command stop-instance executed successfully.
```

## 3.7 停止本地实例

`stop-local-instance` 子命令停止正在运行的 Apusic Server 实例。

`stop-local-instance` 子命令不与 DAS 联系以确定实例所在的节点。要确定实例所在的节点，子命令搜索包含节点目录的目录。如果存在多个节点目录，则必须将节点指定为子命令的选项。

本地模式支持此子命令。

### 3.7.1 概要

```
stop-local-instance [--help]
[--nodedir node-dir] [--node node]
[--force={true|false}] [--kill={false|true}]
[instance-name]
```

### 3.7.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--force
```

指定是否立即强制停止实例。

可能的值如下：

- `true`

立即强制停止实例（默认）。

- `false`

该子命令会一直等待，直到与该实例关联的所有线程都退出，然后再停止该实例。

```
--kill
```

指定是否通过使用操作系统的功能终止实例进程来终止实例。

可能的值如下：

- `false`

实例没有被杀死。该子命令使用 Java 平台的功能来终止实例进程（默认）。

- `true`

实例被杀死。该子命令使用操作系统的功能来终止实例进程。

```
--nodedir
```

指定包含实例节点目录的目录。实例的文件存储在实例的节点目录中。默认值为 `${aas-install}/nodes`。

```
--node
```

指定实例所在的节点。只有当 `--nodedir` 选项指定的目录仅包含一个节点目录时，才能省略此选项。否则，此选项是必需的。

```
instance-name
```

需要停止的实例名称。

### 3.7.3 示例

示例 1：停止实例 locins21。

```
asadmin> stop-instance locins21
The instance, locins21, was stopped.
Command stop-local-instance executed successfully.
```

## 3.8 删除实例

`delete-instance` 子命令删除 Apusic Server 实例。此子命令要求在运行域管理服务器 (DAS) 的主机和由实例所在节点表示的主机上配置 secure shell (SSH)。

可以从相互通讯的 DAS 的任何主机运行此子命令。

不能删除正在运行中的实例。否则，会发生错误。

该子命令将会执行以下操作：

- 从域管理服务器 (DAS) 的配置中删除实例
- 从文件系统中删除实例的文件

如果要删除的实例是使用节点目录的唯一实例，则该目录也会被删除。

如果删除了独立实例，则该独立实例的配置也会被删除。独立实例是指名为 `instance-name-config` 的配置，没有其他集群或非集群实例引用该配置。

此子命令仅在远程模式下受支持。

### 3.8.1 概要

```
delete-instance [--help] instance-name
```

### 3.8.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
instance-name
```

需要删除的实例名称。

### 3.8.3 示例

示例 1：删除实例myins21

```
asadmin> delete-instance myins21
Command delete-instance executed successfully.
```

## 3.9 删除本地实例

`delete-local-instance` 子命令删除 Apusic Server 实例。

不能删除正在运行中的实例。否则，会发生错误。

该子命令将会执行以下操作：

- 从域管理服务器 (DAS) 的配置中删除实例
- 从文件系统中删除实例的文件

如果要删除的实例是使用节点目录的唯一实例，则该目录也会被删除。

如果删除了独立实例，则该独立实例的配置也会被删除。独立实例是指名为 `instance-name-config` 的配置，没有其他集群或非集群实例引用该配置。

此子命令仅在远程模式下受支持。

### 3.9.1 概要

```
delete-local-instance [--help]
[--nodedir node-dir] [--node node-name]
[instance-name]
```

### 3.9.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--nodedir
```

指定实例所属的节点的目录。默认为 `aas-install/nodes`。

```
--node
```

指定实例所在的节点。仅当 `--nodedir` 选项指定的目录仅包含一个节点目录时，才可以省略此选项。否则，需要此选项。

```
instance-name
```

需要删除的实例名称。

### 3.9.3 示例

示例 1：删除实例myinsloc

```
asadmin> delete-local-instance myinsloc
Command delete-local-instance executed successfully.
```

## 4 应用程序

### 4.1 部署应用程序

deploy 子命令将应用程序部署到服务器。应用程序可以是企业应用程序、Web 应用程序、Enterprise JavaBeans (EJB) 模块、EBA应用程序、连接器模块和应用程序客户端模块。如果组件已经部署或已经存在，如果 --force 选项设置为 true（默认为 false），则强制重新部署。

--createtables 和 --dropandcreatetables 选项是布尔标志，因此可以取 true 或 false 的值。这些选项仅在部署尚未映射到数据库的CMP bean时使用（即，模块的META-INF目录中没有提供 sun-CMP-mappings.xml 描述符）。否则，它们将被忽略。--createtables 和 --dropandcreatetables 选项是互斥的；应该只使用一个。如果删除或创建表失败，部署不会失败；日志文件中提供警告消息。此子命令仅在远程模式下受支持。

#### 4.1.1 概要

```

deploy [--help]
[--force={false|true}]
[--virtualservers virtual_servers]
[--contextroot context_root]
[--precompilejsp={false|true}]
[--verify={false|true}]
[--name component_name]
[--upload={true|false}]
[--retrieve local_dirpath]
[--dbvendorname dbvendorname]
[--createtables={true|false}|--dropandcreatetables={true|false}]
[--uniquetablenames={true|false}]
[--deploymentplan deployment_plan]
[--altdtd alternate_deploymentdescriptor]
[--runtimealtdtd runtime_alternate_deploymentdescriptor]
[--deploymentorder deployment_order]
[--enabled={true|false}]
[--generateterminstubs={false|true}]
[--availabilityenabled={false|true}]
[--asyncreplication={true|false}]
[--lbenabled={true|false}]
[--keepstate={false|true}]
[--libraries jar_file[,jar_file]*]
[--target target]
[--type pkg-type]
[--BuiltinWebserviceEnabled={false|true}]
[--BuiltinCDIEnabled={false|true}]
[--BuiltinJPAEnabled={false|true}]
[--BuiltinJSFEnabled={false|true}]
[--BuiltinBeanValidationEnabled={false|true}]
[--BuiltinJSONPEnabled={false|true}]
[--BuiltinRESTFULEnabled={false|true}]
[--extra-path sharelibname]
[--share-lib-priority={appFirst|classesFirst|shareLibFirst}]
[--hot-loading-enabled={false|true}]
[--hot-loading-delay-time hot-loading-delay-time]
[--properties (name=value) [:name=value]*]
[file_archive|filepath]

```

#### 4.1.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--force
```

如果设置为 `true`，即使指定的组件已经部署或已经存在，也会重新部署该组件。默认为 `false`。

```
--virtualservers
```

指定一个或多个虚拟服务器 ID。多个 ID 用英文逗号分隔。

```
--contextroot
```

当应用为 Web 模块时才有效。对于其他类型，它会被忽略；默认为不带扩展名的文件名。

```
--precompilejsp
```

默认情况下，此选项不允许在部署期间预编译 JSP。相反，JSP 在运行时编译。默认为 `false`。

```
--verify
```

如果设置为 `true` 并且从更新工具安装了所需的验证程序包，则会验证部署描述符的语法和语义。默认为 `false`。

```
--name
```

所部署组件的名称。

```
--upload
```

指定子命令是否将文件上载到 DAS。在大多数情况下，可以省略此选项。

有效值如下：

- `false`

该子命令不上传文件并尝试通过指定的文件名访问该文件。如果 DAS 无法访问该文件，则子命令将失败。例如，DAS 可能以与管理用户不同的用户身份运行，并且没有对文件的读取权限。在这种情况下，如果 `--upload` 选项为 `false`，则子命令将失败。

- `true`

该子命令通过网络连接将文件上载到 DAS。

默认值取决于 DAS 是在运行子命令的主机上还是在远程主机上。

- 如果 DAS 位于运行子命令的主机上，则默认值为 `false`。
- 如果 DAS 位于远程主机上，则默认值为 `true`。

如果指定了目录文件路径，则忽略此选项。

```
--retrieve
```

将客户端存根 JAR 文件从服务器机器检索到本地目录。

```
--dbvendorname
```

指定为其创建表的数据库供应商的名称。支持的值包括 `db2`、`mssql`、`mysql`、`oracle`、`derby`、`javadb`、`postgresql`、`sybase` 等。这些值不区分大小写。如果未指定，则使用 `apusic-ejb-jar.xml` 中的数据库供应商名称属性的值。如果没有指定值，则连接到 `apusic-ejb-jar.xml` 文件中 `cmp-resource` 元素的 `jndi-name` 子元素指定的资源，并读取数据库供应商名称。如果无法建立连接，或者无法识别该值，则表示符合 SQL-92。

**--createtables**

如果指定为 `true`，则在部署具有未映射CMP beans的应用程序时创建表。如果指定为 `false`，则不创建表。如果未指定，`apusic-ejb-jar.xml` 文件中 `cmp-resource` 元素的 `create-tables-at-deploy` 实体的值，决定是否创建表。

**--dropandcreatetables**

如果在重新部署组件时指定为`true`，则在创建新表之前，将删除之前部署创建的表。适用于具有未映射CMP bean的已部署应用程序。在应用程序的初始部署或显式取消部署后的部署中，不会删除先前存在的表。如果指定为`false`，则不会删除或创建表。如果未指定，如果 `apusic-ejb-jar.xml` 文件中 `cmp-resource` 元素的 `drop-tables-at-deploy` 实体的值设置为`true`，将会删除表；如果 `apusic-ejb-jar.xml` 文件中 `cmp-resource` 元素的 `create-tables-at-deploy` 实体的值设置为`true`将会新建表。

**--uniquetablenames**

保证所有 bean 的唯一表名，并导致将哈希码添加到表名中。如果您的应用程序具有区分大小写的 bean 名称，这将非常有用。适用于具有未映射 CMP bean 的应用程序。

**--deploymentplan**

部署计划，这是一个包含 Apusic Server 描述符的 JAR 文件。部署纯 EAR 文件时指定此选项。纯 EAR 文件是没有 Apusic Server 描述符的 EAR。

**--enabled**

允许用户访问应用程序。如果设置为 `false`，用户将无法访问应用程序。此选项启用指定目标实例或集群上的应用程序。默认值为 `true`。

**--generatermistubs**

如果设置为 `true`，则生成静态 RMI-IIOP 存根并将其放入 `client.jar`。如果设置为 `false`，则不生成存根。默认为 `false`。

**--availabilityenabled**

此选项控制是否为 Web 会话和有状态会话 bean (SFSB) 检查点和潜在的钝化启用高可用性。如果设置为 `false`（默认值），则为指定的应用程序、Web 应用程序或 EJB 模块禁用所有 Web 会话保存和 SFSB 检查点。如果设置为 `true`，则启用指定的应用程序或模块以实现高可用性。仅当在更高级别（例如服务器和容器级别）配置和启用高可用性时，才将此选项设置为 `true`。

**--asyncreplication**

此选项控制是否首先缓冲启用高可用性的 Web 会话和 SFSB 状态，然后使用单独的异步线程进行复制。如果设置为 `true`（默认），则性能会提高，但可用性会降低。如果状态被缓冲但尚未复制的实例失败，则状态将丢失。如果设置为 `false`，则会降低性能，但保证可用性。状态不会被缓冲，而是立即传输到集群中的其他实例。

**--lbenabled**

此选项控制已部署的应用程序是否可用于负载均衡。默认值为真。

**--keepstate**

此选项控制是否在重新部署之间保留 Web 会话、SFSB 实例和持久创建的 EJB 计时器。

默认值为`false`。此选项仅在默认服务器实例`server`上支持。

重新部署之间对应用程序的某些更改会阻止此功能正常工作。例如，不要更改SFSB bean类中的实例变量集。对于web应用程序，只有在 `apusic-web-app.xml` 文件中 `session-manager` 元素的 `persistence-type` 属性为 `file` 时，此功能才适用。

对于有状态的会话bean实例，在服务器中设置了不具有高可用性的持久性类型（`sfsb持久性类型属性`），并且必须将其设置为`file`，这是默认值和推荐值。如果任何活动的web会话、SFSB实例或EJB计时器未能保留或恢复，则重新部署完成后，这些都不可用。然而，重新部署仍在继续，并记录了一条警告。为了保留活动状态数据，`{product---name}`对数据进行序列化并将其保存在内存中。为了恢复数据，重新部署的应用程序的类加载器对之前保存的数据进行反序列化。

**--libraries**

类库JAR文件，多个以逗号分隔。通过它们的相对或绝对路径指定库 JAR 文件。指定相对于 `domain-dir` 的相对路径 `/lib/aplibs`。

**--target**

指定要部署到的目标。有效值为：

- `server`

将组件部署到默认服务器实例 `server` 并且是默认值。

- `domain`

将组件部署到域。如果 `domain` 是初始部署的目标，则应用程序被部署到域中，但没有服务器实例或集群引用该应用程序。如果 `domain` 是重新部署的目标（该 `--force` 选项设置为 `true`），并且为引用应用程序的集群或服务器实例启用了动态重新配置，则引用集群或服务器实例会自动获取应用程序的新版本。如果重新部署并禁用动态配置，则在重新启动集群或独立服务器实例之前，引用集群或服务器实例不会获得应用程序的新版本。

- `cluster_name`

将组件部署到集群中的每个服务器实例。

- `instance_name`

将组件部署到特定的独立服务器实例。

## `--type`

正在部署的组件的打包归档类型。可能的值如下：

- `car`

打包成CAR文件的组件。

- `ear`

打包成EAR文件的组件。

- `osgi`

组件为OSGI bundle。

- `rar`

打包成RAR文件的组件。

- `war`

打包成WAR文件的组件。

该 `--type` 选项是可选的。如果组件被打包为常规存档，请忽略此选项。

## `--BuiltinWebserviceEnabled`

配置是否开启AAS内置webservice组件，值为`true`表示启用，值为`false`表示禁用。

## `--BuiltinCDIEnabled`

配置是否开启AAS内置CDI组件，值为`true`表示启用，值为`false`表示禁用。

## `--BuiltinJPAEnabled`

配置是否开启AAS内置JPA组件，值为`true`表示启用，值为`false`表示禁用。

## `--BuiltinJSFEnabled`

配置是否开启AAS内置JSF组件，值为`true`表示启用，值为`false`表示禁用。

## `--BuiltinBeanValidationEnabled`

配置是否开启AAS内置BeanValidation组件，值为`true`表示启用，值为`false`表示禁用。

```
--BuiltinJSONPEnabled
```

配置是否开启AAS内置JSONP组件，值为true表示启用，值为false表示禁用。

```
--BuiltinRESTFULEnabled
```

配置是否开启AAS内置RESTFUL组件，值为true表示启用，值为false表示禁用。

```
extra-path
```

配置引用共享类库，值为共享类库的名称。需要先创建共享类库才能配置，参考子命令 `create-shared-lib`。

```
share-lib-priority
```

设置共享类库优先级，appFirst表示应用优先（默认），classesFirst表示classes目录优先，shareLibFirst表示共享类库优先。

```
--hot-loading-enabled
```

是否开启热加载功能，开启后应用文件有更新时无需重启应用即可重新加载。默认为false。

```
--hot-loading-delay-time
```

设置应用热加载延迟时间，默认为60秒。

```
--properties 或 --property
```

可选的 `keyword-value` 对，用于指定部署的其他属性。可用属性由正在部署或重新部署的组件的实现决定。`--properties` 选项和 `--property` 选项是等效的。无论指定多少属性，都可以使用任一选项。

您可以为部署指定以下属性：

- `jar-signing-alias`

指定用于签名应用程序客户端容器JAR文件的安全证书的别名。Java Web Start不会运行需要提升权限的代码，除非它位于用用户系统信任的证书签名的JAR文件中。为了您的方便，`{product---name}`使用域密钥库中具有此别名的证书自动对JAR文件进行签名。Java Web Start随后询问用户是否信任该代码，并显示`{product---name}`证书信息。要使用其他证书对此JAR文件进行签名，请将证书添加到域密钥库中，然后使用此属性。例如，您可以使用来自受信任机构的证书，这避免了Java Web Start提示，也可以使用来自您自己的公司的证书，用户知道他们可以信任该公司。默认值为`kaas`，这是为每个域创建的自签名证书的别名。

- `java-web-start-enabled`

指定是否允许对应用程序客户端模块进行Java Web Start访问。默认为 `true`。

- `compatibility`

指定在应用程序的JAR可见性要求方面向后兼容的`{product---name}`版本。唯一允许的值是`v2`，它指的是Sun Apusic Enterprise Server版本2或Sun Java System Application Server版本9.1或9.1.1。从Java EE 6开始，Java EE平台规范对EAR文件中的各个模块可以看到JAR文件提出了比Java EE 5更严格的要求。特别是，应用程序客户端不得访问EAR文件中的EJB JAR文件或其他JAR文件，除非引用使用标准的Java SE机制（例如扩展）或Java EE库目录机制。将此属性设置为`v2`将删除这些限制。

- `keepSessions={false|true}`

`--keepstate` 被选项取代。如果`--force`选项设置为`true`，则此属性可用于指定是否保留正在重新部署的应用程序的活动会话，然后在重新部署完成时恢复。适用于web容器中的HTTP会话。默认值为`false`。

- `false`

应用程序的活动会话不会被保留和还原（默认）。

- `true`

应用程序的活动会话将被保留和还原。如果应用程序的任何活动会话未能保留或恢复，则重新部署完成后，所有会话都将不可用。然而，重新部署仍在继续，并记录了一条警告。为了保留活动会话，`{product---name}`对会话进行序列化并将其保存在内存中。为了恢复会话，新重新部署的应用程序的类加载器会反序列化以前保存的任何会话

- `preserveAppScopedResources`

如果设置为 `true`，则保留任何应用程序范围的资源并在重新部署期间恢复它们。默认为 `false`。

其他可用属性由正在重新部署的组件实现确定。

```
[file_archive|filepath]
```

包含需要部署的已打包的应用程序的路径。此路径可以是相对路径或绝对路径。打包格式可以是以下格式之一：

- 一个存档文件，例如， `/home/samples/hello.war`

如果 `--upload` 选项设置为 `true`，则这是本地客户端计算机上可部署文件的路径。如果 `--upload` 选项设置为 `false`，则这是服务器上文件的路径。

- 包含可部署存档的分解格式的目录。这是服务器上目录的路径。

如果指定目录，则忽略 `--upload` 选项。

```
dirpath
```

包含可部署存档的分解格式的目录的路径。这是服务器上目录的路径。

#### 4.1.3 示例

示例 1：部署 EAR 应用

部署文件名为 `jsp.ear` 的应用，部署成功之后，应用名称为 `jps`。

```
asadmin> deploy jps.ear
Application deployed successfully with name jps.
Command deploy executed successfully
```

示例 2：部署 WEB 应用

部署存放在 `/home/samples` 下的 WEB 应用 `hello.war`。部署成功后，应用名称为 `hello`。

```
asadmin> deploy /home/samples/hello.war
Application deployed successfully with name hello.
Command deploy executed successfully
```

示例 3：强制部署应用文件，并且指定应用上下文

强制部署应用文件 `hello.war`，并且指定应用上下文路径为 `greetings`。部署之后应用名称为 `hello`，访问时，上下文路径为 `greetings`。

```
asadmin> deploy --force=true --contextroot greetings hello.war
Application deployed successfully with name hello.
Command deploy executed successfully
```

示例 4：部署连接器模块，并且指定部署目标

部署连接器模块 `jdbcra.rar`，并且指定部署目标为独立实例 `myins`。

```
asadmin> deploy --target myins jdbcra.rar
Application deployed successfully with name jdbcra.
Command deploy executed successfully
```

示例 5：部署打包成目录的应用文件

部署打包成目录形式的应用 `default`。

```
asadmin> deploy /home/apusic/samples/example/default/
应用 default 部署结束.
deploy command deprecated. Please use deploy command instead.
Command deploy executed successfully.
```

## 4.2 重新部署

`redeploy` 子命令重新部署企业应用程序、Web 应用程序、基于 Enterprise JavaBeans (EJB) 规范的模块 (EJB 模块)、EBA应用、连接器模块或已部署或已经存在的应用程序客户端模块。该 `redeploy` 子命令保留应用程序最初部署时使用的设置和其他选项。应用程序必须已经部署。否则，会发生错误。

此子命令仅在远程模式下受支持。

### 4.2.1 概要

```
redeploy [--help]
--name component_name
[--upload={true|false}]
[--retrieve local_dirpath]
[--dbvendorname dbvendorname]
[--createtables={true|false}|--dropandcreatetables={true|false}]
[--uniquetablenames={true|false}]
[--deploymentplan deployment_plan]
[--enabled={true|false}]
[--generateterminstubs={false|true}]
[--contextroot context_root]
[--precompilejsp={true|false}]
[--verify={false|true}]
[--virtualservers virtual_servers]
[--availabilityenabled={false|true}]
[--asyncreplication={true|false}]
[--lbenabled={true|false}]
[--keepstate={false|true}]
[--libraries jar_file[,jar_file]*]
[--target target]
[--type pkg-type]
[--properties (name=value) [:name=value]*]
[file_archive|filepath]
```

### 4.2.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--virtualservers
```

一个或多个虚拟服务器 ID。多个 ID 用英文逗号分隔。

```
--contextroot
```

仅当存档是 Web 模块时才有效。对于其他归档类型，它会被忽略；默认为不带扩展名的文件名。

```
--precompilejsp
```

默认情况下，此选项不允许在部署期间预编译 JSP。相反，JSP 在运行时编译。默认为 `false`。

```
--verify
```

如果设置为 `true` 并且从更新工具安装了所需的验证程序包，则会验证部署描述符的语法和语义。默认为 `false`。

```
--name
```

可部署组件的名称。

```
--upload
```

指定子命令是否将文件上传到 DAS。在大多数情况下，可以省略此选项。

有效值如下：

- `false`

该子命令不上传文件并尝试通过指定的文件名访问该文件。如果 DAS 无法访问该文件，则子命令将失败。例如，DAS 可能以不同于管理用户的用户身份运行，并且没有对该文件的读取访问权限。在这种情况下，如果 `--upload` 选项为 `false`，则子命令将失败。

- `true`

该子命令通过网络连接将文件上传到 DAS。

默认值取决于 DAS 是在运行子命令的主机上还是在远程主机上。

- 如果 DAS 位于运行子命令的主机上，则默认值为 `false`。
- 如果 DAS 位于远程主机上，则默认值为 `true`。

如果指定了目录文件路径，则忽略此选项。

```
--retrieve
```

将客户端存根 JAR 文件从服务器机器检索到本地目录。

```
--dbvendorname
```

指定为其创建表的数据库供应商的名称。支持的值包括 `db2`、`mssql`、`mysql`、`oracle`、`derby`、`javadb`、`postgresql`、`sybase` 等。这些值不区分大小写。如果未指定，则使用 `apusic-ejb-jar.xml` 中的数据库供应商名称属性的值。如果没有指定值，则连接到 `apusic-ejb-jar.xml` 文件中 `cmp-resource` 元素的 `jndi-name` 子元素指定的资源，并读取数据库供应商名称。如果无法建立连接，或者无法识别该值，则表示符合 SQL-92。

```
--createtables
```

如果指定为 `true`，则在部署具有未映射 CMP beans 的应用程序时创建表。如果指定为 `false`，则不创建表。如果未指定，`apusic-ejb-jar.xml` 文件中 `cmp-resource` 元素的 `create-tables-at-deploy` 实体的值，决定是否创建表。

```
--dropandcreatetables
```

如果在重新部署组件时指定为 `true`，则在创建新表之前，将删除之前部署创建的表。适用于具有未映射 CMP bean 的已部署应用程序。在应用程序的初始部署或显式取消部署后的部署中，不会删除先前存在的表。如果指定为 `false`，则不会删除或创建表。如果未指定，如果 `apusic-ejb-jar.xml` 文件中 `cmp-resource` 元素的 `drop-tables-at-deploy` 实体的值设置为 `true`，将会删除表；如果 `apusic-ejb-jar.xml` 文件中 `cmp-resource` 元素的 `create-tables-at-deploy` 实体的值设置为 `true` 将会新建表。

```
--uniquetablenames
```

保证所有 bean 的唯一表名，并导致将哈希码添加到表名中。如果您的应用程序具有区分大小写的 bean 名称，这将非常有用。适用于具有未映射 CMP bean 的应用程序。

```
--deploymentplan
```

部署计划，这是一个包含 Apusic Server 描述符的 JAR 文件。部署纯 EAR 文件时指定此选项。纯 EAR 文件是没有 Apusic Server 描述符的 EAR。

```
--enabled
```

允许用户访问应用程序。如果设置为 `false`，用户将无法访问应用程序。此选项启用指定目标实例或集群上的应用程序。如果部署到目标 `domain`，则忽略此选项，因为部署到域不会部署到特定实例或集群。默认值为 `true`。

```
--generatermistubs
```

如果设置为 `true`，则生成静态 RMI-IIOP 存根并将其放入 `client.jar`。如果设置为 `false`，则不生成存根。默认为 `false`。

### `--availabilityenabled`

此选项控制是否为 Web 会话和有状态会话 bean (SFSB) 检查点和潜在的钝化启用高可用性。如果设置为 `false` (默认值)，则为指定的应用程序、Web 应用程序或 EJB 模块禁用所有 Web 会话保存和 SFSB 检查点。如果设置为 `true`，则启用指定的应用程序或模块以实现高可用性。仅当在更高级别 (例如服务器和容器级别) 配置和启用高可用性时，才将此选项设置为 `true`。

### `--asyncreplication`

此选项控制是否首先缓冲启用高可用性的 Web 会话和 SFSB 状态，然后使用单独的异步线程进行复制。如果设置为 `true` (默认)，则性能会提高，但可用性会降低。如果状态被缓冲但尚未复制的实例失败，则状态将丢失。如果设置为 `false`，则会降低性能，但保证可用性。状态不会被缓冲，而是立即传输到集群中的其他实例。

### `--lbenabled`

此选项控制已部署的应用程序是否可用于负载均衡。默认值为真。

### `--keepstate`

此选项控制是否在重新部署之间保留 Web 会话、SFSB 实例和持久创建的 EJB 计时器。

默认值为 `false`。此选项仅在默认服务器实例 `server` 上支持。

重新部署之间对应用程序的某些更改会阻止此功能正常工作。例如，不要更改 SFSB bean 类中的实例变量集。对于 web 应用程序，只有在 `apusic-web-app.xml` 文件中 `session-manager` 元素的 `persistence-type` 属性为 `file` 时，此功能才适用。

对于有状态的会话 bean 实例，在服务器中设置了不具有高可用性的持久性类型 (`sfsb`持久性类型属性)，并且必须将其设置为 `file`，这是默认值和推荐值。如果任何活动的 web 会话、SFSB 实例或 EJB 计时器未能保留或恢复，则重新部署完成后，这些都不可用。然而，重新部署仍在继续，并记录了一条警告。为了保留活动状态数据，`{product---name}` 对数据进行序列化并将其保存在内存中。为了恢复数据，重新部署的应用程序的类加载器对之前保存的数据进行反序列化。

### `--libraries`

以逗号分隔的库 JAR 文件列表。通过它们的相对或绝对路径指定库 JAR 文件。指定相对于 `domain-dir` 的相对路径 `/lib/applibs`。这些库按指定的顺序提供给应用程序。

### `--target`

指定要部署到的目标。有效值为：

- `server`

将组件部署到默认服务器实例 `server` 并且是默认值。

- `domain`

将组件部署到域。如果 `domain` 是初始部署的目标，则应用程序被部署到域中，但没有服务器实例或集群引用该应用程序。如果 `domain` 是重新部署的目标，并且为引用应用程序的集群或服务器实例启用了动态重新配置，则引用集群或服务器实例会自动获取应用程序的新版本。如果重新部署并禁用动态配置，则在重新启动集群或独立服务器实例之前，引用集群或服务器实例不会获得应用程序的新版本。

- **集群名称**

将组件部署到集群中的每个服务器实例。

- **实例名称**

将组件部署到特定的独立服务器实例。

### `--type`

正在部署的组件的打包归档类型。可能的值如下：

- `car`

打包成 CAR 文件的组件。

- ear

打包成EAR文件的组件。

- osgi

组件为OSGI bundle。

- rar

打包成RAR文件的组件。

- war

打包成WAR文件的组件。

该 `--type` 选项是可选的。如果组件被打包为常规存档，请忽略此选项。

`--properties` 或者 `--property`

可选的 `keyword-value` 对，用于指定部署的其他属性。可用属性由正在部署或重新部署的组件的实现决定。`--properties` 选项和 `--property` 选项是等效的。无论指定多少属性，都可以使用任一选项。

可以为部署指定以下属性：

- `jar-signing-alias`

指定用于签名应用程序客户端容器JAR文件的安全证书的别名。Java Web Start不会运行需要提升权限的代码，除非它位于用用户系统信任的证书签名的JAR文件中。为了您的方便，`{product---name}`使用域密钥库中具有此别名的证书自动对JAR文件进行签名。Java Web Start随后询问用户是否信任该代码，并显示`{product---name}`证书信息。要使用其他证书对此JAR文件进行签名，请将证书添加到域密钥库中，然后使用此属性。例如，您可以使用来自受信任机构的证书，这避免了Java Web Start提示，也可以使用来自您自己的公司的证书，用户知道他们可以信任该公司。默认值为`kaas`，这是为每个域创建的自签名证书的别名。

- `java-web-start-enabled`

指定是否允许对应用程序客户端模块进行Java Web Start访问。默认为 `true`。

- `compatibility`

指定在应用程序的JAR可见性要求方面向后兼容的`{product---name}`版本。唯一允许的值是`v2`，它指的是Sun Apusic Enterprise Server版本2或Sun Java System Application Server版本9.1或9.1.1。从Java EE 6开始，Java EE平台规范对EAR文件中的各个模块可以看到JAR文件提出了比Java EE 5更严格的要求。特别是，应用程序客户端不得访问EAR文件中的EJB JAR文件或其他JAR文件，除非引用使用标准的Java SE机制（例如扩展）或Java EE库目录机制。将此属性设置为`v2`将删除这些限制。

- `keepSessions={false|true}`

`--keepstate` 被选项取代。如果`--force`选项设置为`true`，则此属性可用于指定是否保留正在重新部署的应用程序的活动会话，然后在重新部署完成时恢复。适用于web容器中的HTTP会话。默认值为`false`。

- `false`

应用程序的活动会话不会被保留和还原（默认）。

- `true`

应用程序的活动会话将被保留和还原。如果应用程序的任何活动会话未能保留或恢复，则重新部署完成后，所有会话都将不可用。然而，重新部署仍在继续，并记录了一条警告。为了保留活动会话，`{product---name}`对会话进行序列化并将其保存在内存中。为了恢复会话，新重新部署的应用程序的类加载器会反序列化以前保存的任何会话

- `preserveAppScopedResources`

如果设置为 `true`，则保留任何应用程序范围的资源并在重新部署期间恢复它们。默认为 `false`。

其他可用属性由正在重新部署的组件实现确定。

`[file_archive|filepath]`

包含需要部署的已打包的应用程序的路径。此路径可以是相对路径或绝对路径。打包格式可以是以下格式之一：

- 一个存档文件，例如， `/home/samples/hello.war`

如果 `--upload` 选项设置为 `true`，则这是本地客户端计算机上可部署文件的路径。如果 `--upload` 选项设置为 `false`，则这是服务器上文件的路径。

- 包含可部署存档的分解格式的目录。这是服务器上目录的路径。

如果指定目录，则忽略 `--upload` 选项。

### 4.2.3 示例

示例1: 重新部署应用程序

重新部署应用程序hello。

```
asadmin> redeploy --name hello /home/apusic/samples/hello.war
Application deployed successfully with name hello.
Command redeploy executed successfully
```

## 4.3 卸载应用

`undeploy` 子命令卸载已部署的应用程序或模块并将其从存储库中删除。

### 4.3.1 概要

```
undeploy [--help] [--target target] [--droptables={true|false}]
[--cascade={false|true}] name
```

### 4.3.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--cascade
```

如果设置为 `true`，则删除与正在取消部署的资源适配器关联的所有连接池和连接器资源。如果设置为 `false`，则如果任何池和资源仍与资源适配器关联，则取消部署将失败。然后，要么必须明确删除这些池和资源，要么必须将选项设置为 `true`。如果选项设置为 `false`，并且没有池和资源仍与资源适配器关联，则取消部署资源适配器。此选项适用于连接器（资源适配器）和应用程序。默认值为 `false`。

```
--droptables
```

如果设置为 `true`，则删除应用程序在部署期间使用CMP bean创建的表。如果设置为 `false`，则不会删除表。如果未指定，则 `apusic-ejb-jar.xml` 文件的 `cmp-resource` 元素中的 `drop-tables-at-deploy` 条目的值将决定是否删除表。默认值为 `true`。

```
--target
```

指定要从中取消部署的目标。有效值为：

- `server`

从默认服务器实例取消部署组件 `server` 并且是默认值。

- `domain`

从域中取消部署组件。

- `cluster_name`

从集群中的每个服务器实例中取消部署组件。

- `instance_name`

从特定的独立服务器实例中取消部署组件。

```
name
```

需要卸载的应用的名称。

### 4.3.3 示例

示例1: 卸载应用

卸载名为hello的应用程序。

```
asadmin> undeploy hello
Command undeploy executed successfully.
```

示例2: 卸载目标中的应用

卸载服务器集群mycluster中的应用test。

```
asadmin> undeploy --target mycluster hello
Command undeploy executed successfully.
```

## 4.4 引用应用程序

`create-application-ref` 子命令创建已部署的应用程序元素（例如，JavaEE应用程序、Web模块或企业bean模块）的引用。可以让应用元素部署到集群或实例。此子命令仅在远程模式下受支持。

### 4.4.1 概要

```
create-application-ref [--help] [--target target]
[--virtualservers virtual_servers] [--enabled=true]
[--lbenabled=true] reference_name
```

### 4.4.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--enabled
```

启用该应用程序元素，默认为true。

```
--target
```

指定实例的目标。有效值为：

- `server`

指定默认服务器实例 `server` 为应用的引用目标，默认值。

- `cluster_name`

列出集群服务器为应用的引用目标。

- `instance_name`

列出独立服务器为应用的引用目标。

```
--virtualservers
```

指定要部署的虚拟服务器ID的逗号分隔列表。此选项仅适用于Web模块（独立或在JavaEE应用程序中）。如果未指定此选项，则应用程序将部署到 `s_erver` 和 `__asadmin`。

```
reference_name
```

应用程序或模块的名称，可以是JavaEE应用程序、Web模块、EJB模块、连接器模块、应用程序客户端模块或生命周期模块。

#### 4.4.3 示例

示例 1：创建应用引用

该示例创建应用 `myapp` 被服务器集群 `cluster1` 引用。

```
asadmin> create-application-ref --target cluster1 myapp
Command create-application-ref executed successfully.
```

## 4.5 查询应用程序

`list-applications` 子命令列出已部署的 Java EE 应用程序以及列出的每个应用程序的类型。

如果 `--type` 未指定该选项，则会列出所有应用程序。如果指定了 `type` 选项，则必须指定类型。此帮助页面的“选项”部分列出了可能的类型。

如果部署了多个版本的模块或应用程序，则此子命令会列出所有版本。要列出启用的版本，请将 `--long` 选项设置为 `true`。

此子命令仅在远程模式下支持。

#### 4.5.1 概要

```
list-applications [--help]
[--long={false|true}] [--resources] [--subcomponents]
[--type type] [target]
```

#### 4.5.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--long
```

如果 `true`，则显示列出的每个模块或应用程序是否已启用。默认值为 `false`。

```
--resources
```

列出每个应用程序的应用程序范围的资源。如果 `--subcomponents` 还使用了该选项，则列出应用程序中每个组件的应用程序范围资源。

```
--subcomponents
```

列出每个应用程序的子组件。列出的子组件取决于应用程序类型。例如，对于 Java EE 应用程序（EAR 文件），列出了模块。对于 Web 应用程序，列出了 `servlet` 和 `JSP` 页面。对于 EJB 模块，列出了 EJB 子组件。

```
--type
```

指定要列出的应用程序的类型。选项如下：

- `application`
- `appclient`
- `connector`
- `ejb`
- `web`

- `webservice`

如果未指定类型，则列出所有应用程序。

```
--target
```

这是子命令操作的目标的名称。有效值如下：

- `server`

列出默认服务器实例的应用程序 `server` 并且是默认值。

- `domain`

列出域的应用程序。

- `cluster_name`

列出集群的应用程序。

- `instance_name`

列出特定独立服务器实例的应用程序。

### 4.5.3 示例

示例1: 列出所有应用程序

```
asadmin> list-applications
jdbc-test <web>
default <ear, web>
Command list-applications executed successfully.
```

## 4.6 列出应用程序引用

`list-application-refs` 子命令列出独立实例或集群服务器实例中的所有应用程序引用。这有效地列出了部署在指定目标上的所有模块（例如，J2EE应用程序、Web模块和企业bean模块）。如果部署了模块或应用程序的多个版本，此子命令将列出所有版本。要列出启用的版本，请将 `--long` 选项设置为 `true`。此子命令仅在远程模式下受支持。

### 4.6.1 概要

```
list-application-refs [--help]
[--long={false| true}] [target]
```

### 4.6.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--long
```

如果为`true`，则显示列出的每个模块或应用程序是否已启用。默认值为`false`。

```
--target
```

指定实例的目标。有效值为：

- `server`

列出默认服务器实例 `server` 的应用的引用情况，默认值。

- `cluster_name`

列出集群服务器的应用的引用情况。

- `instance_name`

列出独立服务器的应用的引用情况。

### 4.6.3 示例

示例 1：列出实例中的应用程序的引用情况

列出实例myserver的应用程序的应用情况

```
asadmin> list-application-refs myserver
ClientSessionMDBApp
MEjbApp
__ejb_container_timer_app
Command list-application-refs executed successfully.
```

## 4.7 删除应用程序引用

`delete-application-refs` 子命令删除应用程序的引用。删除之后，服务器集群或者独立实例下次启动时将不会加载该应用。

删除引用不会导致将应用程序从域中删除。这些位仅由取消部署子命令删除

此子命令仅在远程模式下受支持。

### 4.7.1 概要

```
delete-application-ref [--help] [--target target]
[--cascade=false] reference_name
```

### 4.7.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--target
```

指定实例的目标。有效值为 `server`、`cluster_name`、`instance_name`。

```
reference_name
```

应用程序或模块的名称，可以是JavaEE应用程序、Web模块、EJB模块、连接器模块、应用程序客户端模块或生命周期模块。

### 4.7.3 示例

示例 1：删除应用程序引用

删除独立实例 `myins` 中的应用程序 `MyWebApp` 的引用。

```
asadmin> delete-application-ref --target myins MyWebApp
Command delete-application-ref executed successfully.
```

## 4.8 启动应用程序

`enable` 子命令启用指定的已部署组件。如果组件已启用，则重新启用。如果尚未部署，则返回错误消息。此子命令仅在远程模式下受支持。

如果需要重启组件，应先执行 `disable` 子命令停止组件，再执行 `enable` 子命令启动组件。

### 4.8.1 概要

```
enable [--help] [--target target_name] component_name
```

### 4.8.2 选项

```
--help  
-?
```

显示子命令的帮助文本。

```
--target
```

指定实例的目标。有效值为：

- server

目标为默认服务器实例 `server`，默认值。

- cluster\_name

目标为服务器集群。

- instance\_name

目标为独立实例。

```
component_name
```

要启用的组件的组件名称。

应用多版本同时启动时，如果浏览器已经在访问应用程序，则再次部署其他版本时，该浏览器访问的还是原始版本；如果把浏览器关闭后再打开，再次访问应用程序时，访问的就是最新的版本。

如果应用程序有多个版本同时运行，且浏览器没有处于已经访问该应用程序的状态，此时访问应用程序会访问最新部署的版本。

如果应用程序有多个版本同时运行，且设置了“部署顺序”，重启实例后再访问应用程序，会优先访问“部署顺序”较低数字的版本。

注：应用多版本仅限于类型为Web应用程序。

### 4.8.3 示例

示例 1：启动应用程序

启动应用程序test1.

```
asadmin> enable test1  
Command enable executed successfully
```

示例 2：启动部署在服务器集群中的应用程序

启动部署在服务器集群mycluster的应用程序jpa.

```
asadmin> enable --target mycluster jpa  
Command enable executed successfully
```

## 4.9 停止应用程序

`disable` 子命令会立即禁用指定的已部署组件。如果尚未部署，则返回错误消息。此子命令仅在远程模式下受支持。

### 4.9.1 概要

```
disable [--help] [--target target_name] component_name
```

#### 4.9.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--target
```

指定实例的目标。有效值为：

- server

目标为默认服务器实例 server，默认值。

- cluster\_name

目标为服务器集群。

- instance\_name

目标为独立实例。

```
component_name
```

要停止的组件的组件名称。

#### 4.9.3 示例

示例 1：停止应用程序

停止应用程序test1.

```
asadmin> disable test1
Command disable executed successfully
```

示例 12：停止部署在服务器集群中的应用程序

停止部署在服务器集群mycluster的应用程序jpa.

```
asadmin> disable --target mycluster jpa
Command disable executed successfully
```

## 4.10 查看组件状态

show-component-status 子命令用于查看已部署的组件的状态。此子命令仅在远程模式下受支持。

#### 4.10.1 概要

```
show-component-status [--help] [--target target] component-name
```

#### 4.10.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--target
```

指定实例的目标。有效值为：

- `server`

目标为默认服务器实例 `server`，默认值。

- `cluster_name`

目标为服务器集群。

- `instance_name`

目标为独立实例。

```
component-name
```

要查看的组件名称。

#### 4.10.3 示例

示例 1：查看已部署的组件状态

该示例查看已部署的组件，`testapp`，的状态。

```
asadmin> show-component-status testapp
Status of testapp is enabled
Command show-component-status executed successfully.
```

## 4.11 编辑应用或组件属性

通常情况下，使用 `set` 子命令编辑组件的属性。

可以使用 `get` 子命令查看已部署的应用或组件的属性，再使用 `set` 子命令设置。

### 4.11.1 概要

```
get dotted-attribute-name
```

```
set dotted-attribute-name
```

### 4.11.2 选项

```
dotted-attribute-name
```

以虚线表示法标识属性名称。至少需要一个点名称属性。虚线符号是用于访问可配置实体属性的语法。

### 4.11.3 示例

示例 1：更新已部署的应用或组件的属性

不确定当前配置时，可以使用`get`子命令查看属性配置情况。示例查看应用程序`hello`的配置情况。

```
asadmin> get *.hello.*
applications.application.hello.property.deploymentModules=war
applications.application.hello.property.precompilejsp=false
applications.application.hello.property.secretLevel=SECRET
applications.application.hello.module.hello.name=hello
applications.application.hello.property.BuiltinCDIEnabled=false
applications.application.hello.property.hot-loading-delay-time=60
```

```

applications.application.hello.property.appLocation=${com.apusic.aas.instanceRootURI}/applications/___internal
applications.application.hello.property.hot-loading-enabled=false
applications.application.hello.property.BuiltinJPAEnabled=false
applications.application.hello.async-replication=true
applications.application.hello.availability-enabled=false
applications.application.hello.context-root=/hello
applications.application.hello.deployment-order=100
applications.application.hello.directory-deployed=false
applications.application.hello.enabled=true
applications.application.hello.extra-path=mylib
applications.application.hello.hot-loading-delay-time=60
applications.application.hello.hot-loading-enabled=false
applications.application.hello.location=${com.apusic.aas.instanceRootURI}/applications/hello/
applications.application.hello.name=hello
applications.application.hello.object-type=user
applications.application.hello.shared-lib-priority=sharedLibFirst
applications.application.hello.tamper-black-list=.png,.js
applications.application.hello.tamper-black-select=list
applications.application.hello.tamper-enabled=false
applications.application.hello.tamper-interval=10
applications.application.hello.tamper-white-list-pattern=(.)*\/hello\/WEB-INF\/(.)*$
applications.application.hello.tamper-white-select=pattern
applications.application.hello.module.hello.engine.web.sniffer=web
applications.application.hello.property.inheritedSession=false
applications.application.hello.property.archiveType=war
applications.application.hello.property.implicitCdiEnabled=true
applications.application.hello.property.BuiltinBeanValidationEnabled=false
servers.server.server.application-ref.hello.disable-timeout-in-minutes=30
servers.server.server.application-ref.hello.enabled=true
servers.server.server.application-ref.hello.lb-enabled=true
servers.server.server.application-ref.hello.ref=hello
servers.server.server.application-ref.hello.virtual-servers=server
applications.application.hello.property.java-web-start-enabled=true
applications.application.hello.property.webserviceDisabled=true
applications.application.hello.property.BuiltinRESTFULEnabled=false
applications.application.hello.property.BuiltinJSFEnabled=false
applications.application.hello.module.hello.engine.security.sniffer=security
applications.application.hello.property.BuiltinJSONPEnabled=false
applications.application.hello.property.defaultAppName=hello
applications.application.hello.property.delegate=false
Command get executed successfully.

```

再使用set子命令设置属性。示例设置应用程序hello的上下文路径。

```
asadmin>set applications.application.hello.context-root=/myhello
```

#### 示例2 设置应用文件防篡改策略

该示例对应用hello设置防篡改策略，属性包括：

- tamper-enabled：是否开启防篡改功能。
- tamper-interval：设置应用文件检测时间间隔。
- tamper-white-select：设置防篡改文件的格式，list表示通过常规格式，pattern表示通过正则表达式。
- tamper-white-list：配置防篡改的文件，支持配置文件夹、文件名、文件前缀、文件后缀(如.js)；多个用逗号分隔。与tamper-white-list-pattern互斥。
- tamper-white-list-pattern：通过正则表达式格式配置防篡改的文件，与tamper-white-list互斥。
- tamper-black-select：设置排除文件的格式，list表示通过常规格式，pattern表示通过正则表达式。

- tamper-black-list: 配置排除的文件, 支持配置文件夹、文件名、文件前缀、文件后缀(如.js); 多个用逗号分隔。与tamper-black-list-pattern互斥。
- tamper-black-list-pattern: 输入通过正则表达式格式配置排除文件, 与tamper-black-list互斥。

```
asadmin>set applications.application.hello.tamper-black-list=.png,.js
applications.application.hello.tamper-black-select=list
applications.application.hello.tamper-enabled=false
applications.application.hello.tamper-interval=10
applications.application.hello.tamper-white-list-pattern=(.)*hello/WEB-INF/(.)*$
applications.application.hello.tamper-white-select=pattern
```

## 5 JDBC连接池

### 5.1 创建JDBC连接池

`create-jdbc-connection-pool` 子命令使用指定的 JDBC 连接池名称注册一个新的 Java 数据库连接 (“JDBC”) 软件连接池。

`--property` 可以通过使用指定用户、密码或其他连接信息的选项或通过 XML 描述符文件中指定连接信息来创建具有身份验证的 JDBC 连接池。

此子命令仅在远程模式下受支持。

#### 5.1.1 概要

```
create-jdbc-connection-pool [--help]
[--datasourceclassname=datasourceclassname]
[--restype=resourcetype]
[--steadypoolsize=poolsize]
[--maxpoolsize=maxpoolsize]
[--maxwait=maxwaittime]
[--poolresize=poolresizelimit]
[--idletimeout=idletimeout]
[--initsql=initsqlstring]
[--isolationlevel=isolationlevel]
[--isisolationguaranteed={true|false}]
[--isconnectvalidatereq={false|true}]
[--validationmethod=validationmethod]
[--validationtable=validationtable]
[--failconnection={false|true}]
[--allownoncomponentcallers={false|true}]
[--nontransactionalconnections={false|true}]
[--validateatmostonceperiod=validationinterval]
[--leaktimeout=leaktimeout]
[--leakreclaim={false|true}]
[--statementleaktimeout=satementleaktimeout]
[--statmentleakreclaim={false|true}]
[--creationretryattempts=creationretryattempts]
[--creationretryinterval=creationretryinterval]
[--sqltracelisteners=sqltracelisteners[,sqltracelisteners]]
[--statementtimeout=statementtimeout]
[--lazyconnectionenlistment={false|true}]
[--lazyconnectionassociation={false|true}]
[--associatewiththread={false|true}]
[--driverclassname=jdbcdriverclassname]
[--matchconnections={false|true}]
[--maxconnectionusagecount=maxconnectionusagecount]
[--ping={false|true}]
[--pooling={false|true}]
[--statementcachesize=statementcachesize]
[--validationclassname=validationclassname]
[--wrapjdbcobjects={false|true}]
[--description description]
[--property name=value) [:name=value]*]
[--target=target]
connectionpoolid
```

#### 5.1.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--datasourceclassname
```

供应商提供的 JDBC 数据源资源管理器的名称。支持 XA 或全局事务的数据源类将实现该 `javax.sql.XADataSource` 接口。非 XA 或独占本地事务数据源将实现该 `javax.sql.DataSource` 接口。

```
--restype
```

当数据源类实现两个或更多接口（`javax.sql.DataSource`、`javax.sql.XADataSource` 或 `javax.sql.ConnectionPoolDataSource`）时，或者必须提供驱动程序类名时需要。

- 如果 `--restype = java.sql.Driver`，则该 `--driverclassname` 选项是必需的。
- 如果 `--restype = javax.sql.DataSource`、`javax.sql.XADataSource` 或 `javax.sql.ConnectionPoolDataSource`，则该 `--datasourceclassname` 选项是必需的。
- 如果 `--restype` 未指定，则必须指定 `--driverclassname` 或 `--datasourceclassname`，但不能同时指定两者。

```
--steadypoolsize
```

池中维护的最小和初始连接数。默认值为 8。

```
--maxpoolsize
```

可以创建的最大连接数。默认值为 32。

```
--maxwait
```

在发送连接超时之前调用者将等待的时间量（以毫秒为单位）。默认值为 60000（60 秒）。值 0 会强制调用者无限期地等待。

```
--poolresize
```

`idle-timeout-in-seconds` 计时器到期时要删除的连接数。这是池将扩大或缩小连接数的数量。纵向扩展：当池没有空闲连接时，池将按此数量纵向扩展。Scale down：删除所有无效和空闲的连接，有时会导致删除数量大于此值的连接。空闲时间超过超时的连接是删除的候选对象。Steadypoolsize 将得到保证。可能的值是从 0 到 `MAX_INTEGER`。默认值为 2。

```
--idletimeout
```

连接可以在池中保持空闲的最长时间（以秒为单位）。在这段时间之后，实现可以关闭这个连接。此超时值必须保持比数据库服务器端超时值短，以防止在应用程序中累积不可用的连接。默认值为 300。

```
--initsql
```

每当从池中创建连接时执行的 SQL 字符串。如果重用现有连接，则不执行此字符串。空闲时间超过超时的连接是删除的候选对象。此选项没有默认值。

```
--isolationlevel
```

池化数据库连接上的事务隔离级别。此选项没有默认值。如果未指定，则池以 JDBC 驱动程序提供的默认隔离级别运行。您可以使用以下标准事务隔离级别之一设置所需的隔离级别：`read-uncommitted`、`read-committed`、`repeatable-read`、`serializable`。以编程方式更改池连接上的隔离级别的应用程序可能会污染池。这可能导致程序错误。

```
--isisolationguaranteed
```

这仅适用于为 `transaction-isolation-level` 指定特定隔离级别时。默认值是 true。

此选项可确保每次从池中获得连接时，隔离级别都会设置为所需的值。这可能会对某些 JDBC 驱动程序产生一些性能影响。`--isolationlevel` 当应用程序在返回连接之前没有更改时，管理员可以将其设置为 false。

```
--isconnectvalidatereq
```

如果设置为 true，则在发送给应用程序之前验证或检查连接以查看它们是否可用。默认值为假。

```
--validationmethod
```

is-connection-validation-required 当为真时要执行的验证类型。有效设置为：`auto-commit`、`meta-data`、`table` 或 `custom-validation`。默认值为 `table`。

```
--validationtable
```

用于执行查询以验证连接的验证表的名称。如果 `is-connection-validation-required` 设置为 true 并且 `connection-validation-type` 设置为 `table`，则此选项是必需的。

```
--failconnection
```

如果设置为 true，则在单个验证检查失败时必须关闭池中的所有连接。默认值为假。尝试重新建立失败的连接。

```
--allownoncomponentcallers
```

将此属性设置为 true 的池可供非 Java EE 组件使用，即除 EJB 或 Servlet 之外的组件。返回的连接与从事务管理器获得的事务上下文一起自动登记。由非组件调用者获得的连接不会在事务结束时由容器自动清理。这些连接需要由调用者显式关闭。

```
--nontransactionalconnections
```

将此属性设置为 true 的池返回非事务性连接。此连接不会自动加入事务管理器。

```
--validateatmostonceperiod
```

指定最多一次验证连接的连续请求之间的时间间隔（以秒为单位）。将此属性设置为适当的值可以最大限度地减少连接的验证请求数。默认值为 0，表示该属性未启用。

```
--leaktimeout
```

指定要跟踪连接池中的连接泄漏的时间量（以秒为单位）。当应用程序在指定期限内没有将连接返回到池中时，就认为是潜在的泄漏，调用者的堆栈跟踪将被记录下来。此选项仅检测是否存在连接泄漏。只有 `connection-leak-reclaim` 设置为 true 时才能回收连接。

如果启用了连接泄漏跟踪，您可以使用管理控制台启用对 JDBC 连接池的监控，以获取有关连接泄漏数量的统计信息。默认值为 0，即禁用连接泄漏跟踪。

```
--leakreclaim
```

指定泄漏连接跟踪完成后是否将泄漏连接恢复到连接池。可能的值如下：

- `false`

泄漏的连接不会恢复到连接池（默认）。

- `true`

泄漏的连接恢复到连接池。

```
--statementleaktimeout
```

指定时间量（以秒为单位），在此之后将检测到任何尚未被应用程序关闭的语句。如果语句对象没有正确关闭，应用程序可能会用完游标。此选项仅检测是否存在语句泄漏。

`statement-leak-reclaim` 仅当设置为 true 时才能回收该语句。泄漏的语句在回收时关闭。

当检测到语句泄漏时，将记录创建语句的调用者的堆栈跟踪。如果启用了语句泄漏跟踪，您可以使用管理控制台启用对 JDBC 连接池的监控，以获取有关语句泄漏数量的统计信息。默认值为 0，即禁用语句泄漏跟踪。

以下限制适用于语句泄漏超时值：

- 该值必须小于为连接设置的值 `leak-timeout`。
- 该值必须大于为设置的值 `statement-timeout`。

`--statementleakreclaim`

指定是否在语句泄漏后回收泄漏的语句。可能的值如下：

- `false`

泄露的语句不会被回收（默认）。

- `true`

泄露的语句被回收。

`--creationretryattempts`

指定如果初始尝试失败，Apusic Server 重试创建连接的最大次数。默认值为 0，它指定 Apusic Server 不会重试创建连接。

`--creationretryinterval`

指定连续尝试创建连接的时间间隔（以秒为单位）。

如果 `--creationretryattempts` 为 0，`--creationretryinterval` 则忽略该选项。默认值为 10。

`--sqltracelisteners`

提供数据库活动自定义日志记录的一个或多个自定义模块的列表。每个模块都必须实现 `org.APUSIC.api.jdbc.SQLTraceListener` 公共接口。当设置为适当的值时，将跟踪应用程序执行的 SQL 语句。此选项没有默认值。

`--statementtimeout`

指定在多长时间后终止未完成的查询。

长时间保持不完整的查询可能会导致提交查询的应用程序挂起。为防止发生这种情况，请使用此选项为将从您正在创建的连接池中创建的所有语句设置超时。创建语句时，Apusic Server 将 `QueryTimeout` 语句的属性设置为指定的时间长度。默认值为 -1，它指定不完整的查询永远不会终止。

`--lazyconnectionenlistment`

指定是否仅在方法实际使用资源时才征用事务的资源。可能的值如下：

- `false`

事务的资源总是被征用，而不仅仅是在方法实际使用资源时（默认）。

- `true`

仅当方法实际使用资源时，才会征用事务的资源。

`--lazyconnectionassociation`

指定是否仅在使用物理连接时才将物理连接与逻辑连接关联，并在事务完成时解除关联。这种关联和分离使物理连接的重用成为可能。可能的值如下：

- `false`

物理连接甚至在使用物理连接之前就与逻辑连接相关联，并且在事务完成时不会取消关联（默认）。

- `true`

物理连接仅在使用物理连接时才与逻辑连接关联，并在事务完成时解除关联。该 `--lazyconnectionenlistment` 选项还必须设置为 `true`。

`--associatewiththread`

指定连接是否与线程关联以使线程能够重用连接。如果连接不与线程关联，则每次线程需要连接时，线程都必须从池中获取连接。可能的值如下：

- `false`

连接不与线程关联（默认）。

- `true`

连接与线程相关联。

`--driverclassname`

供应商提供的 JDBC 驱动程序类的名称。该驱动程序应实现该 `java.sql.Driver` 接口。

`--matchconnections`

指定从池中选择的连接是否应由资源适配器匹配。如果池中的所有连接都是同构的，则从池中挑选的连接不需要由资源适配器匹配，这意味着可以将此选项设置为 `false`。可能的值如下：

- `false`

资源适配器不应匹配连接（默认）。

- `true`

连接应该由资源适配器匹配。

`--maxconnectionusagecount`

指定可以重用连接的最大次数。达到此限制时，连接将关闭。通过限制可以重用连接的最大次数，您可以避免语句泄漏。

默认值为 0，表示对连接可以重复使用的次数没有限制。

`--ping`

指定在池创建或重新配置期间是否对池进行 ping 操作，以识别并警告其属性的任何错误值。默认值为假。

`--pooling`

指定是否为池启用连接池。默认值是 `true`。

`--statementcachesize`

使用默认缓存机制（最近最少使用）缓存的 SQL 语句数。默认值为 0，表示未启用语句缓存。

`--validationclassname`

当值为 `time` 时提供自定义验证的类的 `validationmethod` 名称 `custom-validation`。此类必须实现该 `org.APUSIC.api.jdbc.ConnectionValidation` 接口，并且 Apusic Server 必须可以访问该接口。如果连接验证类型设置为自定义验证，则此选项是必需的。

`--wrapjdbcobjects`

指定池基础结构是否向应用程序提供包装的 JDBC 对象。通过提供包装的 JDBC 对象，池化基础设施通过确保应用程序使用连接池中的逻辑连接而不是物理连接来防止连接泄漏。使用逻辑连接可确保连接在关闭时返回到连接池。但是，提供包装的 JDBC 对象会损害应用程序的性能。默认值是 `true`。

池化基础设施为 JDBC API 中以下接口的实现提供包装对象：

- `java.sql.CallableStatement`
- `java.sql.DatabaseMetaData`
- `java.sql.PreparedStatement`
- `java.sql.ResultSet`
- `java.sql.Statement`

`--wrapjdbcobjects` 的可能值如下：

- `false`

池化基础架构不向应用程序提供包装的 JDBC 对象。（默认）。

- `true`

池化基础架构为应用程序提供包装的 JDBC 对象。

## --description

提供有关指定 JDBC 连接池的详细信息文本。

## --property

用于配置池的可选属性名称/值对。以下属性可用：

- `user`

指定连接数据库的用户名。

- `password`

指定连接数据库的密码。

- `databaseName`

指定此连接池的数据库。

- `serverName`

指定此连接池的数据库服务器。

- `port`

指定数据库服务器侦听请求的端口。

- `networkProtocol`

指定通信协议。

- `roleName`

指定初始 SQL 角色名称。

- `datasourceName`

指定一个基础 `XADataSource`， 或一个 `ConnectionPoolDataSource` 如果连接池完成。

- `description`

指定文本描述。

- `url`

指定此连接池的 URL。虽然这不是标准属性，但它是常用的。

- `dynamic-reconfiguration-wait-timeout-in-seconds`

用于启用对连接池的动态重新配置，对使用该池的应用程序透明，因此无需重新启用应用程序即可对池的属性或属性更改生效。只要连接请求在超时期限内，任何进行中事务的连接请求都将被允许以旧池配置完成，从而完成事务。新的连接请求将等待池重新配置完成，并且将使用修改后的池配置获取连接。

- `LazyConnectionEnlistment`

已弃用。使用等效属性。默认值为假。

- `LazyConnectionAssociation`

已弃用。使用等效属性。默认值为假。

- `AssociateWithThread`

已弃用。使用等效属性。默认值为假。

- `MatchConnections`

已弃用。使用等效属性。默认值是true。

- `Prefer-Validate-Over-Recreate`

指定池大小调整器是否应在销毁和重新创建空闲连接之前验证它们。默认值是true。

- `time-to-keep-queries-in-minutes`

指定将被缓存以用于计算常用查询的分钟数。JDBC连接池开启SQL跟踪和监控后生效。默认值为 5 分钟。

- `number-of-top-queries-to-report`

指定在报告最常用和最常用的查询时要列出的查询数。JDBC连接池开启SQL跟踪和监控后生效。默认值为 10 个查询。

#### 注意:

如果属性名称或属性值包含冒号, 则 \ 必须使用反斜杠 ( \ ) 对名称或值中的冒号进行转义。其他字符可能还需要转义字符。

```
--target
```

不要指定此选项。保留此选项是为了与早期版本兼容。如果您指定此选项, 则不会发生语法错误。相反, 子命令会成功运行并显示一条警告消息, 指出该选项被忽略。

```
connectionpoolid
```

要创建的 JDBC 连接池的名称。

### 5.1.3 示例

示例 1: 创建资源类型为 `java.sql.Driver`, 名为 `mydata` 的达梦数据库连接池

先将数据库驱动文件, 如 `dm-jdbc-driver-7-jdk18.jar` 放置 `${DOMAIN_HOME}/lib/ext` 中。创建指定资源类型为 `java.sql.Driver`, 驱动程序类名称 `dm.jdbc.driver.DmDriver`, 数据库为达梦数据库, 地址为 `127.0.0.1`, 端口为 `5236`, 库名为 `DATATEST`, 命名为 `mydata` 的JDBC连接池。

```
asadmin>create-jdbc-connection-pool --driverclassname dm.jdbc.driver.DmDriver --restype java.sql.Driver --
property
user=DATATEST:password=DATATEST:databaseName=DATATEST:port=5236:url=\ "jdbc\ :dm\ ://127.0.0.1\ :5236/DATATE
mydata
```

示例 2: 创建一个资源类型为名为`javax.sql.XADataSource`, 名为`sample_derby_pool`的连接池。

```
asadmin> create-jdbc-connection-pool
--datasourceclassname org.apache.derby.jdbc.ClientDataSource
--restype javax.sql.XADataSource
--property portNumber=1527:password=APP:user=APP:serverName=
localhost:databaseName=sun-appserv-samples:connectionAttributes=\;
create\ \=true sample_derby_pool
Command create-jdbc-connection-pool executed successfully
```

转义字符反斜杠 ( \ ) 用于 `--property` 区分分号 ( ; ) 的选项。两个反斜杠 ( \ \ ) 用于区分等号 ( = )。

## 5.2 删除JDBC连接池

`delete-jdbc-connection-pool` 子命令删除 JDBC 连接池。在运行此子命令之前, 必须删除与 JDBC 连接池的所有关联。

此子命令仅在远程模式下受支持。

### 5.2.1 概要

```
delete-jdbc-connection-pool [--help]
[--cascade={false|true}]
```

```
jdbc_connection_pool_id
```

### 5.2.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--cascade
```

如果该选项设置为 true，则与池关联的所有 JDBC 资源（除了池本身）都将被删除。设置为 false 时，如果有任何资源与池相关联，则删除池将失败。资源必须被显式删除或该选项必须设置为 true。默认值为假。

```
jdbc_connection_pool_id
```

要删除的 JDBC 资源的名称。

### 5.2.3 示例

示例 1 删除 JDBC 连接池

此示例删除 sample\_derby\_pool JDBC 连接池。

```
asadmin> delete-jdbc-connection-pool --cascade=false sample_derby_pool
Command delete-jdbc-connection-pool executed correctly.
```

## 5.3 列出JDBC连接池

list-jdbc-connection-pool 子命令列出 JDBC连接池。

此子命令仅在远程模式下受支持。

### 5.3.1 概要

```
list-jdbc-connection-pools [--help]
```

### 5.3.2 选项

```
--help
-?
```

显示子命令的帮助文本。

### 5.3.3 示例

示例 1 列出 JDBC 连接池

```
asadmin> list-jdbc-connection-pools
sample_derby_pool
    TimerPool
Command list-jdbc-connection-pools executed successfully.
```

## 5.4 更新JDBC连接池信息

set 子命令可用于更新JDBC连接池信息。

set 子命令使用带点名称修改一个或多个可配置属性的值。

监视层次结构中的属性是只读的，但可以修改配置属性。可以使用 `list` 子命令来显示表示单个服务器组件和子系统的带点名称。例如，带点名称可能是 `server.application.web-module`。在发现特定组件或子系统后，可以使用 `get` 子命令访问这些属性。有关带点名称的更多详细信息，请参阅带点名称帮助页面。

注意：对shell或命令解释器具有特殊含义的字符，例如\*（星号），应根据shell适当引用或转义，例如，将参数括在引号中。在多模中，仅包含空格、引号或反斜杠的参数需要引号。

设置属性值时，建议单个属性执行。

#### 5.4.1 概要

```
set resources.jdbc-connection-pool.[JDBCpool_name].[attribute]
```

#### 5.4.2 选项

```
[JDBCpool_name]
```

需要更新的JDBC连接池名称。

```
[attribute]
```

需要更新的属性。

#### 5.4.3 示例

示例1: 查看JDBC连接池的属性

通过get子命令获取当前的JDBC连接池信息，在不确定哪个连接池时，可以使用\*(星号)。

```
asadmin> set resources.jdbc-connection-pool.*
```

示例 2: 设置JDBC连接池的最小和初始池大小

```
asadmin> set resources.jdbc-connection-pool.DerbyPool.steady-pool-size=9
Command set executed successfully.
```

示例 3: 设置JDBC连接池的最大池大小

```
asadmin> set resources.jdbc-connection-pool.mydata.max-pool-size=50
Command set executed successfully.
```

示例4: 设置数据库用户名为mydata

```
asadmin> set resources.jdbc-connection-pool.mydata.property.user=mydaya
Enter admin password for user "admin">
resources.jdbc-connection-pool.mydata.property.user | =mydaya |
Command set executed successfully.
```

## 6 JDBC资源

### 6.1 查看JDBC资源

`list-jdbc-resources` 子命令查询 JDBC 资源。

此子命令仅在远程模式下受支持。

#### 6.1.1 概要

```
list-jdbc-resources [--help] [--target target]
```

#### 6.1.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--target
```

指定要从中查询 JDBC 资源的目标。

有效值为：

`server` 查询默认服务器实例。这是默认值。

`domain` 查询域。

`cluster_name` 查询集群。

`instance_name` 查询实例。

#### 6.1.3 示例

示例 1：查询 JDBC 资源

```
asadmin> list-jdbc-resources
```

### 6.2 创建JDBC资源

`create-jdbc-resource` 子命令创建一个新的 JDBC 资源。

此子命令仅在远程模式下受支持。

#### 6.2.1 概要

```
create-jdbc-resource [--help]
--connectionpoolid connectionpoolid
[--enabled={false|true}]
[--description description]
[--property (property=value)[:name=value]*]
[--target target]
jndi_name
```

#### 6.2.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--connectionpoolid
```

JDBC 连接池的名称。如果两个或多个 JDBC 资源元素指向同一个连接池元素，它们在运行时使用同一个池连接。

```
--enabled
```

确定是否在运行时启用 JDBC 资源。默认值是true。

```
--description
```

提供有关 JDBC 资源的描述性详细信息的文本。

```
--property
```

用于配置资源的可选属性名称/值对。

```
--target
```

选择目标。

有效值为：

`server` 默认服务器实例。这是默认值。

`domain` 域。

`cluster_name` 集群。

`instance_name` 服务器实例。

**注意：**

资源是为整个域创建的，但资源的资源引用仅为指定的 `--target` 创建。这意味着，尽管资源是在域级别定义的，但它仅在指定的目标级别可用。如果需要，使用 `create-resource-ref` 子命令引用多个目标中的资源。

```
jndi_name
```

此 JDBC 资源的 JNDI 名称。

### 6.2.3 示例

示例 1：创建名为jdbc/DerbyPool的JDBC资源关联连接池sample\_derby\_pool

```
asadmin> create-jdbc-resource --connectionpoolid sample_derby_pool jdbc/DerbyPool
Command create-jdbc-resource executed successfully.
```

## 6.3 删除JDBC资源

`delete-jdbc-resource` 子命令删除 JDBC 资源。确保在运行此子命令之前删除与 JDBC 资源的所有关联。

此子命令仅在远程模式下受支持。

### 6.3.1 概要

```
delete-jdbc-resource [--help] [--target target] jndi_name
```

### 6.3.2 选项

```
--help
```

```
-?
```

显示子命令的帮助文本。

```
--target
```

选择目标。

有效值为：

`server` 默认服务器实例。这是默认值。

`domain` 域。

`cluster_name` 集群。

`instance_name` 服务器实例。

```
jndi_name
```

需要删除的 JDBC 资源的 JNDI 名称。

### 6.3.3 示例

示例 1：删除 JDBC 资源 jdbc/DerbyPool

```
asadmin> delete-jdbc-resource jdbc/DerbyPool
Command delete-jdbc-resource executed successfully.
```

## 6.4 创建资源引用

`create resource-ref` 子命令创建独立实例或集群的 JNDI 资源的引用，例如，使用 `create-jdbc-resource` 子命令创建的 JDBC 资源通过资源引用命令使其在实例或集群中可用。组成集群的一个或多个目标实例不需要运行或可用，此子命令才能成功。如果一个或多个实例不可用，它们将在下次启动时收到新资源。

注意：

只能为可绑定资源创建资源引用，例如 jdbc-resource, connector-resource, admin-object-resource, mail-resource, custom-resource, 或 jndi-resource。应用程序不直接引用 jdbc-connection-pool 或 connector-connection-pool。相反，它们分别通过 JDBC 资源 jdbc-resource 或 连接器资源 connector-resource 引用。

此子命令仅在远程模式下支持。

### 6.4.1 概要

```
create-resource-ref [--help] [--target target]
[--enabled={false|true}] reference_name
```

### 6.4.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--target
```

选择目标。

有效值为：

`server` 默认服务器实例。这是默认值。

`domain` 域。

`cluster_name` 集群。

`instance_name` 服务器实例。

```
--enabled
```

是否启用资源。只有在全局级别启用资源时，此值才会生效。默认值为 `true`。

```
reference_name
```

资源的名称或 JNDI 名称。

### 6.4.3 示例

示例 1 创建对 JMS 目标资源的引用

此示例创建 `jms/Topic` 对集群 `Cluster1` 上 JMS 目标资源的引用。

```
asadmin> create-resource-ref --target Cluster1 jms/Topic
resource-ref jms/Topic created successfully.
Command create-resource-ref executed successfully.
```

示例 2 创建 JDBC 资源引用

此示例创建 JDBC 资源 `jdbc/test` 对独立实例 `myins` 的引用，且设置状态为 `true`。

```
asadmin> create-resource-ref --enabled=true --target myins jdbc/test
resource-ref jdbc/test created successfully.
myins:
resource-ref jdbc/test created successfully.

Command create-resource-ref executed successfully.
```

## 6.5 删除资源引用

`delete-resource-ref` 子命令从集群或非集群服务器实例中删除对资源（例如，JDBC 资源）的引用。这有效地导致从目标实例或集群的 JNDI 树中删除资源。

组成集群的一个或多个目标实例不需要运行或可用，此子命令才能成功。如果一个或多个实例不可用，它们将在下次启动时不再加载 JNDI 树中的资源。

删除引用不会导致从域中删除资源。仅通过该资源的 `delete` 子命令（例如，`delete-jdbc-resource`）才会删除该资源。

此子命令仅在远程模式下受支持。

### 6.5.1 概要

```
delete-resource-ref [--help] [--target target] reference_name
```

### 6.5.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--target
```

选择目标。

有效值为：

`server` 默认服务器实例。这是默认值。

`domain` 域。

`cluster_name` 集群。

`instance_name` 服务器实例。

`reference_name`

需要删除的资源的名称或 JNDI 名称。

### 6.5.3 示例

示例 1 删除对资源的引用

此示例删除 `jms/Topic` 对集群 `cluster1` 上 JMS 目标资源的引用。

```
asadmin> delete-resource-ref --target cluster1 jms/Topic
resource-ref jms/Topic deleted successfully.
Command delete-resource-ref executed successfully.
```

## 6.6 创建JDBC多数据源资源

`create-jdbc-multi-resource` 子命令创建一个新的 JDBC 多数据源资源。

此子命令仅在远程模式下受支持。

### 6.6.1 概要

```
create-jdbc-multi-resource --poolnames=poolnames [--enabled=true] [--description=description] [--
target=target]
[--secretlevel=SECRET] --algotype=algotype [--failoverifbusy=false] --testsql=select 1
[--testfrequency=5] jndiname
```

### 6.6.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--poolnames
```

JDBC 连接池的名称。多个用英文逗号分开。

```
--enabled
```

确定是否在运行时启用 JDBC 资源。默认值是 true。

```
--description
```

提供有关 JDBC 资源的描述性详细信息的文本。

```
--secretlevel
```

设置该资源的级别。

```
--target
```

选择目标。

有效值为：

`server` 默认服务器实例。这是默认值。

`domain` 域。

`cluster_name` 集群。

`instance_name` 服务器实例。

```
--algotype fail-over load-balancing
```

此 JDBC 资源的算法类型。可设置为失效转移 `fail-over` 或负载均衡 `load-balancing`。

```
--failoverifbusy
```

是否开启失效转移, true或false。 `--algotype` 值为 `fail-over` 时设置; 默认值为false。

```
--testsql
```

测试 语句, 默认值为select 1。

```
--testfrequency
```

检测间隔, 默认值为5秒。

```
jndi_name
```

此 JDBC 资源的 JNDI 名称。

### 6.6.3 示例

示例 1: 创建名为jdbc/test的JDBC资源关联连接池mydata、mydatapool

```
asadmin> create-jdbc-multi-resource --poolnames mydata,mydatapool --algotype load-balancing --testSql
"select 1" jdbc/test
Command create-jdbc-multi-resource executed successfully.
```

## 6.7 删除JDBC多数据源资源

`delete-jdbc-multi-resource` 子命令删除 JDBC多数据源 资源。确保在运行此子命令之前删除与 JDBC 资源的所有关联。

此子命令仅在远程模式下受支持。

### 6.7.1 概要

```
delete-jdbc-multi-resource [--help] [--target target] jndi_name
```

### 6.7.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--target
```

选择目标。

有效值为:

`server` 默认服务器实例。这是默认值。

`domain` 域。

`cluster_name` 集群。

`instance_name` 服务器实例。

```
jndi_name
```

需要删除的 JDBC 资源的 JNDI 名称。

### 6.7.3 示例

示例 1：删除 JDBC 资源 jdbc/test

```
asadmin> delete-jdbc-multi-resource jdbc/test
Command delete-jdbc-multi-resource executed successfully.
```

## 6.8 更新 JDBC 资源

set 子命令可用于更新 JDBC 资源信息。

set 子命令使用带点名称修改一个或多个可配置属性的值。

监视层次结构中的属性是只读的，但可以修改配置属性。可以使用 list 子命令来显示表示单个服务器组件和子系统的带点名称。例如，带点名称可能是 server.application.web-module。在发现特定组件或子系统后，可以使用 get 子命令访问这些属性。有关带点名称的更多详细信息，请参阅带点名称帮助页面。

注意：对 shell 或命令解释器具有特殊含义的字符，例如 \*（星号），应根据 shell 适当引用或转义，例如，将参数括在引号中。在多模中，仅包含空格、引号或反斜杠的参数需要引号。

设置属性值时，建议单个属性执行。

### 6.8.1 概要

```
set resources.jdbc-resources.[JNDI_name].[attribute]
```

### 6.8.2 选项

```
[JNDI_name]
```

需要更新的 JDBC 资源的 JNDI 名称。

```
[attribute]
```

需要更新的属性。

### 6.8.3 示例

示例 1：查看 JDBC 资源的属性

通过 get 子命令获取当前的 JDBC 资源信息，在不确定哪个连接池时，可以使用 \*（星号）。

```
asadmin> get resources.jdbc-resource.*
```

示例 2：修改 JDBC 资源关联的 JDBC 连接池

修改 JDBC 资源 jdbc/test 的连接池为 mydata。

```
asadmin> set resources.jdbc-resource.jdbc/test.pool-name=mydata
Command set executed successfully.
```

示例 3：修改 JDBC 多数据源资源关联的 JDBC 连接池

修改 JDBC 多数据源资源 jdbc/mytest 的连接池为 mydata、mydatapool。

```
asadmin> set resources.jdbc-multi-resource.jdbc/jpa.pool-names=mydata,mydatapool
Command set executed successfully.
```

## 7 服务器集群

查询服务器集群使用 `list-clusters` 子命令，创建服务器集群使用 `create-cluster` 子命令，删除服务器集群使用 `delete-cluster` 子命令，创建服务器集群实例使用 `create-instance` 子命令，删除服务器集群实例使用 `delete-instance` 子命令，启用服务器集群使用 `start-cluster` 子命令，停止服务器集群使用 `stop-cluster` 子命令，部署应用至服务器集群使用 `deploy` 子命令。

### 7.1 查询服务器集群

`list-clusters` 子命令查询 服务器集群。

此子命令仅在远程模式下受支持。

#### 7.1.1 概要

```
list-clusters [--help] [--target target]
```

#### 7.1.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--target
```

指定要从中查询 JDBC 资源的目标。

有效值为：

`domain` 查询域，默认值。

`cluster_name` 查询集群。

`instance_name` 查询实例。

`node-name` 查询节点。

`configuration-name` 查询配置。

#### 7.1.3 示例

示例 1：查询集群

```
asadmin>list-clusters
```

## 7.2 创建服务器集群

`create-cluster` 子命令创建一个 Apusic Server 集群。

集群需要对命名配置的引用，该配置定义了添加到集群中的所有实例的配置。可以在命令中指定配置来创建集群，但不是必需的。如果未指定配置，则子命令会为集群创建一个名为 `cluster-name-config` 的配置。创建的集群是独立集群，因为集群的配置不与任何其他集群或独立实例共享。

#### 7.2.1 概要

```
create-cluster [--help] [--config config-name]
[--systemproperties (name=value)[:name=value]*]
[--properties (name=value)[:name=value]*]
[--gmsenabled={true|false}]
[--multicastport multicast-port]
[--multicastaddress multicast-address]
[--bindaddress bind-address]
```

```
[--hosts hadb-host-list]
[--haagentport port-number]
[--haadminpassword password]
[--haadminpasswordfile file-name] [--devicesize devicesize ]
[--haproperty (name=value)[:name=value]*]
[--autohadb=false] [--portbase port-number]
cluster-name
```

### 7.2.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--config
```

指定集群引用的命名配置。配置必须存在且不得命名为 `default-config` 或 `server-config`。指定该 `--config` 选项会创建一个共享集群。如果省略此选项，则会创建一个独立集群。

```
--systemproperties
```

定义为集群创建的配置的系统属性。这些属性会覆盖 `default-config` 配置中的属性值。以下属性可用：

- `ASADMIN_LISTENER_PORT` 该属性指定DAS连接到实例以管理实例的HTTP端口或HTTPS端口的端口号。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。
- `HTTP_LISTENER_PORT` 此属性指定用于侦听 HTTP 请求的端口号。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。
- `HTTP_SSL_LISTENER_PORT` 此属性指定用于侦听 HTTPS 请求的端口的端口号。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。
- `IIOP_LISTENER_PORT` 此属性指定用于 IIOP 连接的端口号。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。
- `IIOP_SSL_LISTENER_PORT` 此属性指定用于安全 IIOP 连接的端口号。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。
- `IIOP_SSL_MUTUALAUTH_PORT` 此属性指定用于通过客户端身份验证进行安全 IIOP 连接的端口的端口号。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。
- `JAVA_DEBUGGER_PORT` 此属性指定用于连接到Java 平台调试器体系结构 (JPDA)调试器的端口号。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。
- `JMS_PROVIDER_PORT` 此属性指定 Java 消息服务提供者的端口号。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。
- `JMX_SYSTEM_CONNECTOR_PORT` 此属性指定 JMX 连接器侦听的端口号。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。
- `OSGI_SHELL_TELNET_PORT` 此属性指定用于连接到Apache Felix Remote Shell的端口号。这个shell 使用Felix shell 服务与OSGi 模块管理子系统进行交互。有效值为 1-65535。在 UNIX 上，创建侦听端口 1-1024 的套接字需要超级用户权限。

```
--properties
```

定义集群的属性。以下属性可用：

- `GMS_DISCOVERY_URI_LIST`

用于发现集群的 {product---name} 实例在集群中的位置。仅当组管理服务 (GMS) 未使用多播广播消息时，才需要此属性。此属性的有效值如下：

- 以逗号分隔的统一资源标识符 (URI) 列表。每个URI必须定位一个{product---name}实例或DAS。如果在同一主机上运行多个{product---name}实例，则需要此格式。列表中每个URI的格式如下：

```
scheme://host-name-or-IP-address:port
```

- `scheme` 是URI方案，即tcp。
- `host-name-or-IP-address:port` 是运行实例的主机名或IP地址。
- `port` 是实例监听来自GMS消息的端口的端口号。实例的系统属性 `GMS_LISTENER_PORT-clustername` 需要设置。
- 运行DAS或实例的IP地址或主机名的逗号分隔列表。该列表可以包含IP地址和主机名的混合。仅当每个主机上运行一个群集实例时，才能使用此格式。`GMS_LISTENER_PORT` 属性的值对于域中的每个群集都必须是唯一的。

- 关键字生成。只有当群集中的一个实例在每个主机上运行，并且DAS在单独的主机上运行时，才能使用此格式。同一主机上的多个实例不能是同一群集的成员。

`GMS_LISTENER_PORT` 属性的值对于域中的每个群集都必须是唯一的

- `GMS_LISTENER_PORT`

集群监听来自GMS的消息的端口号。默认值是对 `GMS_LISTENER_PORT-cluster-name` 系统属性的引用。默认情况下，未设置此系统属性。在这种情况下，GMS从属性 `GMS_TCPSTARTPORT` 和 `GMS_TCPENDPORT` 定义的范围内选择一个空闲端口。默认情况下，此范围为9090-9200。在大多数情况下，默认值就行。

但是，如果GMS不使用多播来广播消息，则 `GMS_LISTENER_PORT` 属性必须指定一个对集群中的所有(product--name)实例有效的端口号。要使用默认值满足此要求，请使用系统属性为每个实例单独设置端口号。例如，使用 `create-system-properties` 子命令为DAS创建系统属性 `GMS_LISTENER_PORT-cluster-name`。然后，对于集群中的每个 `GMS_LISTENER_PORT-cluster-name` 实例，将 `GMS_LISTENER_PORT-cluster-name` 系统属性设置为该实例监听GMS消息的端口号。集群的 `GMS_LISTENER_PORT` 属性的默认值引用此系统属性。

- `GMS_LOOPBACK`

指定实例是否可以从自身接收该实例广播到集群的应用程序级消息。可能的值如下：

- `false`

实例可能不会从自身接收消息（默认）。

- `true`

实例可以从自身接收消息。当实例是集群中唯一的实例时，使用此设置测试实例。

- `GMS_MULTICAST_TIME_TO_LIVE`

在丢弃消息之前，以下类型事件的多播消息可以经历的最大迭代或传输次数：

- 组发现
- 成员心跳
- 成员资格变更

要匹配部署DAS和群集实例的网络配置，请将此值设置得尽可能低。要确定系统的最低可能值，请使用 `validate-multicast` 子命令 值为0可确保多播消息永远不会离开广播它们的主机。值为1可能会阻止在由交换机或路由器连接的同一子网上的主机之间广播消息。

默认值为4，可确保消息成功广播到主机通过交换机或路由器连接的网络中的所有集群成员。

- `GMS_TCPENDPORT`

如果未设置 `GMS_LISTENER_PORT-cluster-name` 系统属性，则GMS选择空闲端口的范围内的最高端口号。默认值为9200。

- `GMS_STARTPORT`

如果未设置 `GMS_LISTENER_PORT-cluster-name` 系统属性，则GMS选择空闲端口的范围内的最低端口号。默认值为9090。

```
--gmsenabled
```

指定是否为集群启用 GMS。可能的值如下：

- `true`

为集群启用 GMS（默认）。当为集群启用GMS时，集群和DAS中的每个服务器实例都会启动GMS。DAS参与了此选项设置为true的每个集群。

- `false`

集群禁用 GMS。

```
--multicastaddress
```

GMS监听组事件的地址。此选项必须指定224.0.0.0到239.255.255.255范围内的多播地址。默认值为228.9.XX.YY；其中XX和YY是自动生成的0到255之间的独立值。

```
--multicastport
```

GMS 侦听器事件的通信端口的端口号。此选项必须指定 2048-49151 范围内的有效端口号。默认值是该范围内自动生成的值。

```
--bindaddress
```

GMS绑定到的网络接口的互联网协议 (IP) 地址。此选项必须指定本地网络接口的IP地址。默认值是所有公共网络接口地址。在多主机上, 此选项配置用于GMS的网络接口。多主机具有两个或多个网络接口。要指定一个对群集中的所有(product--name)实例有效的地址, 请使用系统属性为每个实例单独设置地址。例如, 使用 `create-system-properties` 子命令创建系统属性 `GMS-BIND-INTERFACE-ADDRESS-cluster-name`。然后将此子命令的 `--bindaddress` 选项设置为 `$(GMS-BIND-INTERFACE-ADDRESS-cluster-name)` 指定系统属性。最后, 对于集群中的每个实例, 设置 `GMS-BIND-INTERFACE-ADDRESS-cluster-name` 系统属性到实例计算机上所需的网络接口地址。

```
--hosts
```

不要指定此选项。保留此选项是为了与早期版本兼容。如果您指定此选项, 则不会发生语法错误。相反, 子命令会成功运行并显示一条警告消息, 指出该选项被忽略。

```
--haagentport
```

不要指定此选项。保留此选项是为了与早期版本兼容。如果您指定此选项, 则不会发生语法错误。相反, 子命令会成功运行并显示一条警告消息, 指出该选项被忽略。

```
--haadminpassword
```

不要指定此选项。保留此选项是为了与早期版本兼容。如果您指定此选项, 则不会发生语法错误。相反, 子命令会成功运行并显示一条警告消息, 指出该选项被忽略。

```
--haadminpasswordfile
```

不要指定此选项。保留此选项是为了与早期版本兼容。如果您指定此选项, 则不会发生语法错误。相反, 子命令会成功运行并显示一条警告消息, 指出该选项被忽略。

```
--devicesize
```

不要指定此选项。保留此选项是为了与早期版本兼容。如果您指定此选项, 则不会发生语法错误。相反, 子命令会成功运行并显示一条警告消息, 指出该选项被忽略。

```
--haproperty
```

不要指定此选项。保留此选项是为了与早期版本兼容。如果您指定此选项, 则不会发生语法错误。相反, 子命令会成功运行并显示一条警告消息, 指出该选项被忽略。

```
--autohadb
```

不要指定此选项。保留此选项是为了与早期版本兼容。如果您指定此选项, 则不会发生语法错误。相反, 子命令会成功运行并显示一条警告消息, 指出该选项被忽略。

```
--portbase
```

不要指定此选项。保留此选项是为了与早期版本兼容。如果您指定此选项, 则不会发生语法错误。相反, 子命令会成功运行并显示一条警告消息, 指出该选项被忽略。

```
cluster-name
```

集群的名称。名称必须符合以下要求: 该名称只能包含 ASCII 字符。名称必须以字母、数字或下划线开头。名称只能包含以下字符: 小写字母大写字母数字连字符时期下划线。该名称在域中必须是唯一的, 并且不能是另一个集群、命名配置、Apusic Server 实例或节点的名称。名称不得为 `domain`、`server` 或 Apusic Server 保留的任何其他关键字。

### 7.2.3 示例

#### 示例 1 创建集群

此示例定义集群名称为 `mycluster`, 其他配置都是默认项,

```
asadmin> create-cluster mycluster
Command create-cluster executed successfully.
```

#### 示例2 创建指定IIOP安全连接的端口的集群

此示例创建一个集群, 该集群的名称 `ltscluster` 将用于安全 IIOP 连接的端口 1169。由于 `--config` 未指定该选项, 因此集群引用了默认配置 `default-config`, 并命名为 `ltscluster-config`。

```
asadmin> create-cluster
--systemproperties IIOP_SSL_LISTENER_PORT=1169
```

```
ltscluster
Command create-cluster executed successfully.
```

## 7.3 启动服务器集群

`start-cluster` 子命令通过启动服务器集群中的所有实例。此子命令要求在运行域管理服务器 (DAS) 的主机和集群中的实例所在的所有主机上配置secure shell (SSH)。

在启动服务器集群前，需要确保服务器集群已存在一个或多个集群实例。实例可以通过 `create-instance` 子命令创建，创建时设置 `--target`，`--target` 的属性为服务器集群名称。

此子命令仅在远程模式下支持。

### 7.3.1 概要

```
start-cluster [--help] [--autohadboverride={true|false}]
[--verbose={false|true}] cluster-name
```

### 7.3.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--autohadboverride
```

不要指定此选项。保留此选项是为了与早期版本兼容。如果您指定此选项，则不会发生语法错误。相反，子命令会成功运行并显示一条警告消息，指出该选项被忽略。

```
--verbose
```

指定集群启动时是否显示附加状态信息。有效值如下：

- `true`  
显示启动集群中每个实例的命令以及启动每个实例的尝试是否成功。
- `false`  
不显示其他状态信息（默认）。

```
cluster-name
```

要启动的集群的名称。

### 7.3.3 示例

#### 示例 1 启动集群

此示例启动集群 `ymlcluster`。启动集群时会显示其他状态信息。

```
asadmin> start-cluster --verbose ymlcluster
start-instance yml-i-sr1-usca-02
start-instance yml-i-sr1-usca-01

The command start-instance executed successfully for:
yml-i-sr1-usca-02 yml-i-sr1-usca-01

Command start-cluster executed successfully.
```

## 7.4 停止服务器集群

`stop-cluster` 子命令通过停止服务器集群中所有服务器集群实例。

此子命令仅在远程模式下支持。

### 7.4.1 概要

```
stop-cluster [--help]
[--verbose={false|true}]
[--kill={false|true}]
cluster-name
```

### 7.4.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--verbose
```

指定在停止集群时是否显示有关集群中实例状态的其他进度消息。可能的值如下：

- `true`  
将显示有关集群中实例状态的其他进度消息。
- `false`  
不会显示有关集群中实例状态的消息。

```
--kill
```

指定是否通过使用操作系统的功能终止实例进程来终止集群中的每个实例。可能的值如下：

- `false`  
没有实例被杀死。该子命令使用 Java 平台的功能来终止每个实例进程（默认）。
- `true`  
每个实例都被杀死。该子命令使用操作系统的功能来终止每个实例进程。

```
cluster-name
```

要停止的集群的名称。

### 7.4.3 示例

#### 示例 1 停止集群

此示例停止集群 `pmdcluster`。停止集群时，会显示有关集群中实例状态的其他进度消息。

```
asadmin> stop-cluster --verbose pmdcluster
stop-instance pmd-i-sj01
stop-instance pmd-i-sj02

The command stop-instance executed successfully for: pmd-i-sj01 pmd-i-sj02
```

```
Command stop-cluster executed successfully.
```

## 7.5 删除服务器集群

`delete-cluster` 子命令删除 Apusic Server 集群。只有当集群不包含实例时，才能删除集群。如果您要删除的集群包含任何实例，请在删除集群之前停止并删除这些实例。

此命令仅在远程模式下受支持。

### 7.5.1 概要

```
delete-cluster [--help]  
cluster-name
```

### 7.5.2 选项

```
--help  
-?
```

显示子命令的帮助文本。

```
cluster-name
```

要删除的集群的名称。

### 7.5.3 示例

示例 1 删除服务器集群

此示例删除服务器集群 `mycluster`。

```
asadmin> delete-cluster mycluster  
Command delete-cluster executed successfully.
```

## 8 负载均衡器

### 8.1 查询负载均衡器

`list-balancer` 子命令查询负载均衡器。

此子命令仅在远程模式下受支持。

#### 8.1.1 概要

```
list-balancer [--help] [--clustername=clustername] [whichtarget]
```

#### 8.1.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--clustername
```

显示该负载均衡器集群下的负载均衡器。

```
whichtarget
```

显示没有被负载均衡器集群引用的负载均衡器，[whichtarget]为newCluster。

#### 8.1.3 示例

示例 1：查询负载均衡器

```
asadmin> list-balancer
alb1
<runningStatus=started,weight=1,type=alb,version=alb/2.0,selected=false,status=0>
myalb7
<runningStatus=stopped,weight=1,type=alb,version=alb/2.0,selected=false,status=0>

Command list-balancer executed successfully.
```

## 8.2 创建负载均衡器

`create-balancer` 子命令创建负载均衡器。

此子命令仅在远程模式下受支持。

#### 8.2.1 概要

```
create-balancer --type=type [--agentip=agentip] [--agentport=agentport]
--balancerport=balancerport [--executefilepath=executefilepath] [--configfilepath=configfilepath]
--algorithm=algorithm [--failover=failover] [--receivetimeout=receivetimeout]
--connecttimeout=connecttimeout [--sockettimeout=sockettimeout]
[--clientheadertimeout=clientheadertimeout] [--clientbodytimeout=clientbodytimeout]
--maxconnections=maxconnections [--maxprocesses=maxprocesses] [--sessionsticky=sessionsticky]
--staticcache=staticcache --ssl=ssl [--sslport=sslport] [--gm=false] --isexternal=isexternal [--
node=node]
[--sslauthmethod=sslauthmethod] [--crtfilepath=crtfilepath] [--crtfilepath2=crtfilepath2]
[--keyfilepath=keyfilepath] [--keyfilepath2=keyfilepath2] [--chainfilepath=chainfilepath]
```

```
[--clustername=clustername] [--licensefilepath=licensefilepath]
  balancename
```

### 8.2.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--type
```

负载均衡器类型，可选择apache/nginx/alb。

```
--isexternal
```

是否引用外部负载均衡器，可选择项有：

- true  
引用外部已安装的负载均衡器，需要设置负载均衡器的IP、端口、执行文件目录、配置文件目录。
- false  
通过节点安装负载均衡器。需要先创建SSH通讯方式的节点，可通过 `create-node-ssh` 子命令创建节点；同时节点需要启动节点代理nodemanager。

```
--agentip
```

节点代理监听IP。

```
--agentport
```

节点代理程序监听端口。

```
--balancerport
```

负载均衡器监听端口，不管是否引用外部负载均衡器都需要指定，访问时访问此端口。

```
--executefilepath
```

负载均衡器执行文件目录，引用外部负载均衡器时需要设置。

```
--configfilepath
```

负载均衡器配置文件目录，引用外部负载均衡器时需要设置。

```
--algorithm
```

设置负载均衡算法。根据负载均衡器的类型 `--type` 指定。

如果 `--type` 为apache，可设置为：

- Requests: 按权重，将根据服务器实例的权重由高至低访问。
- Traffic: 按流量，将根据服务器实例所在的服务器流量由大至小访问。
- Busyness: 按繁忙程度，将根据服务器实例所在的服务器 CPU 使用率由小至大访问。

如果 `--type` 为nginx，可设置为：

- weight: 按权重，将根据服务器实例的权重由高至低访问。
- ip\_hash: 按访问 IP 的 HASH (会话粘滞)；将根据用户请求过来的 IP，然后映射成 hash 值，然后分配到一个特定的服务器里面。实现会话粘滞需要使用该算法。
- fair: 按响应时间，将根据服务器实例的响应时间由快至慢访问。
- url\_hash: 按访问 URL 的 HASH，将根据 URL 的 HASH 访问。

- sticky: 按浏览器cookie (会话粘滞sticky), 将添加sticky模块。节点安装nginx时, 即 `--isexternal` 值为false, 才可以设置该项。

如果 `--type` 为alb, 可设置为:

- round\_robin: 轮询, 按时间顺序将请求轮流分配给后端服务器。
- ip\_hash: 按访问IP的HASH (会话粘滞), 将根据用户请求过来的 IP, 然后映射成 hash 值, 然后分配到一个特定的服务器里面。实现会话粘滞需要使用该算法。
- weight: 按权重, 根据服务器实例的权重由高至低访问。
- least\_conn: 按活跃连接数最少。

```
--failover
```

是否开启失效转移。 `--type` 为apache时可设置。

```
--receivetimeout
```

设置接收响应超时时间。

```
--connecttimeout
```

设置连接后端服务器超时时间。

```
--sockettimeout
```

设置请求后端处理超时时间。

```
--clientheadertimeout
```

设置请求头读取超时时间。

```
--clientbody_timeout
```

设置请求体读取超时时间。

```
--maxconnections
```

设置最大并发数。

```
--maxprocesses
```

最大进程数。

```
--sessionsticky
```

是否开启会话粘滞, 值为true/false。 `--type` 值为 apache 时可设置。

```
--staticcache
```

是否开启静态缓存, 值为true/false。

```
--ssl
```

是否开启SSL, 值为true/false。

```
--sslauthmethod
```

设置SSL认证方式。可设置为:

- one-way: 单向认证。需要配置 `--keyfilepath`, `--certfilepath`。
- two-way: 双向认证。需要配置 `--keyfilepath`, `--certfilepath`、`--chainfilepath`。

```
--crtfilepath
```

设置SSL证书文件目录，如 server.crt。开启SSL认证时需要设置。

```
--chainfilepath
```

设置SSL根证书文件目录，如 ca.crt。开启SSL认证时需要设置。

```
--keyfilepath
```

设置SSL私钥证书文件，如server.key。开启SSL认证时需要设置。

```
--gm
```

是否开启GM/T，值为true/false。开启后，需要配置对应的国密证书。当 --type 值为 alb 时可设置。

--sslauthmethod 的值为one-way时，需要配置 --keyfilepath， --crtfilepath、 --keyfilepath2， --crtfilepath2。

--sslauthmethod 的值为two-way时，需要配置 --keyfilepath， --crtfilepath、 --keyfilepath2， --crtfilepath2、 --chainfilepath。

```
--keyfilepath2
```

设置SSL私钥证书文件，如server.key。开启国密认证时需要设置。

```
--crtfilepath2
```

设置SSL证书文件目录，如 server.crt。开启国密认证时需要设置。

```
--node
```

负载均衡器安装的节点，节点安装负载均衡器时配置，即 --isexternal 值为 false 时设置。

```
--licenseFilePath
```

设置ALB授权文件的路径，当 --type 的值为alb 时可设置。

```
--clustername
```

需要关联的服务器集群名称。

```
balancename
```

定义负载均衡器名称。

### 8.2.3 示例

示例1: 新建引用外部负载均衡器nginx

示例为创建负载均衡器nginx，节点代理IP为 192.20.1.1，节点代理端口为 1099，nginx安装地址为 192.20.1.1，安装目录为 /program/nginx/，定义负载均衡端口为 8085，负载算法为 weight，负载均衡器名称为 nginx43。

```
asadmin>create-balancer --type=nginx --agentip=192.20.1.1 --agentport=1399 --balancerport=8085 --
executefilepath=/program/nginx/sbin/nginx --configfilepath=/program/nginx/conf/nginx.conf --
algorithm=weight --connect_timeout 5 --receivetimeout 60 --sockettimeout 300 --clientheadertimeout 30 --
clientbodytimeout 30 --maxconnections 500 --maxprocesses 5 --staticcache=true --ssl=false --
isexternal=true nginx43
Command create-balancer executed successfully.
```

示例 2: 新建节点安装负载均衡器alb

该示例新建类型为alb的负载均衡器，命名为myalb12，选择节点安装，负载均衡监听端口为8085，设置负载均衡算法为按访问的IPHash（会话粘滞），ALB的授权文件路径为 /opt/alb/license.lic，安装节点为node12，设置连接后端服务器超时时间为5秒，请求后端处理超时时间为300秒，请求体读取超时时间为30秒，请求头读取超时时间为30秒，最大进程数为5，最大并发数为300，开启静态缓存，禁用SSL。

```
asadmin> create-balancer --type alb --isexternal=false --balancerport 8085 --algorithm ip_hash --
licensefilepath "/opt/alb/license.lic" --node node12 --connecttimeout 5 --maxconnections 300 --
clientheadertimeout 30 --clientbodytimeout 30 --staticcache=true --ssl=false --receivetimeout 6 --
maxprocesses 5 --sockettimeout 300 myalb12
```

示例3：新建引用外部已安装负载均衡器alb，且启用国密认证

该示例新建在服务器172.2.1.5已安装的负载均衡器ALB，已上传国密证书至安装目录 /opt/alb/ALB-V2.0.2-AE-arm64 的 conf/ 目录下。负载均衡IP为172.2.1.5，负载均衡监听端口为8086，节点代理IP为172.2.1.5，节点代理端口为1099，负载均衡器命令文件目录为 /opt/alb/ALB-V2.0.2-AE-arm64/sbin/alb，负载均衡器配置文件目录为 /opt/alb/ALB-V2.0.2-AE-arm64/conf/alb.conf，开启国密认证，SSL认证方法为单向认证，证书文件为 /opt/alb/ALB-V2.0.2-AE-arm64/conf/sm2.apusic.com.sig.crt.pem，私钥文件为 /opt/alb/ALB-V2.0.2-AE-arm64/conf/sm2.apusic.com.sig.key.pem，加密证书文件为 /opt/alb/ALB-V2.0.2-AE-arm64/conf/sm2.apusic.com.enc.crt.pem，加密私钥文件为 /opt/alb/ALB-V2.0.2-AE-arm64/conf/sm2.apusic.com.enc.key.pem。开启静态缓存，设置连接后端服务器超时时间为5秒，请求后端处理超时时间为300秒，请求体读取超时时间为30秒，请求头读取超时时间为30秒，最大进程数为5，最大并发数为500，自定义该负载均衡器名称为alb5。

```
asadmin> create-balancer --type alb --isexternal=true --balancerport 8086 --algorithm weight --agentip
172.2.1.5 --agentport 1099 --executefilepath /opt/alb/ALB-V2.0.2-AE-arm64/sbin/alb --configfilepath
/opt/alb/ALB-V2.0.2-AE-arm64/conf/alb.conf --ssl=true --gm=true --sslauthmethod=one-way --staticcache=true
--crtfilepath /opt/alb/ALB-V2.0.2-AE-arm64/conf/sm2.apusic.com.sig.crt.pem --keyfilepath /opt/alb/ALB-
V2.0.2-AE-arm64/conf/sm2.apusic.com.sig.key.pem --keyfilepath2 /opt/alb/ALB-V2.0.2-AE-
arm64/conf/sm2.apusic.com.enc.key.pem --crtfilepath2 /opt/alb/ALB-V2.0.2-AE-
arm64/conf/sm2.apusic.com.enc.crt.pem --connecttimeout 5 --receivetimeout 60 --sockettimeout 300 --
clientheadertimeout 30 --clientbodytimeout 30 --maxconnections 500 --maxprocesses 5 alb5
Enter admin password for user "admin">
Command create-balancer executed successfully.
```

## 8.3 启动负载均衡器

start-balancer 子命令启动负载均衡器。

此命令仅在远程模式下支持。

### 8.3.1 概要

```
start-balancer [--help]
balancer-name
```

### 8.3.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
balancer-name
```

要启动的负载均衡器的名称。

### 8.3.3 示例

示例 1 启动负载均衡器

此示例启动负载均衡器 nginx43

```
asadmin> start-balancer nginx43
Command start-balancer executed successfully.
```

## 8.4 停止负载均衡器

`stop-balancer` 子命令停止负载均衡器。

此命令仅在远程模式下支持。

### 8.4.1 概要

```
stop-balancer [--help]
balancer-name
```

### 8.4.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
balancer-name
```

要停止的负载均衡器的名称。

### 8.4.3 示例

示例 1 停止负载均衡器

此示例停止负载均衡器nginx43。

```
asadmin> stop-balancer nginx43
Command stop-balancer executed successfully.
```

## 8.5 更新负载均衡器

`update-balancer` 子命令更新负载均衡器配置。

此命令仅在远程模式下支持。

注意：更新负载均衡器信息也可以使用 `set` 子命令，使用方法参考 `set` 子命令 部分。

### 8.5.1 概要

```
update-balancer [--agentip=agentip] [--agentport=agentport] [--balancerip=balancerip]
[--balancerport=balancerport] [--algorithm=algorithm] [--failover=failover]
[--sessionsticky=sessionsticky] [--connecttimeout=connecttimeout] [--sockettimeout=sockettimeout]
[--clientheadertimeout=clientheadertimeout] [--clientbodytimeout=clientbodytimeout] [--
maxconnections=maxconnections]
[--maxprocesses=maxprocesses] [--staticcache=staticcache] [--ssl=ssl] [--sslport=sslport] [--gm=false]
[--isexternal=isexternal] [--receivetimeout=receivetimeout] [--sendtimeout=sendtimeout]
[--sslauthmethod=sslauthmethod] [--crtfilepath=crtfilepath] [--keyfilepath=keyfilepath] [--
crtfilepath2=crtfilepath2]
[--keyfilepath2=keyfilepath2] [--chainfilepath=chainfilepath] [--
loadbalancerclustername=loadbalancerclustername]
[--clustername=clustername] [--weight=weight] [--action=action] [--licensefilepath=licensefilepath]
[--authcode=authcode] [--type=type] [--executefilepath=executefilepath] [--configfilepath=configfilepath]
[--os=os] [--version=version] [--status=status] [--algorithmlist=algorithmlist] [--
```

```
algorithmnamelist=algorithmnamelist]
[--ssl_auth_methodlist=ssl_auth_methodlist] [--ssl_auth_methodnamelist=ssl_auth_methodnamelist] [--enablegzip=enablegzip]
[--noderef=noderef] name
```

### 8.5.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--type
```

负载均衡器类型，可选择apache/nginx/alb。显示项，不能修改。

```
--isexternal
```

是否引用外部负载均衡器，显示项，不能修改。选择项有：

- true  
引用外部已安装的负载均衡器，需要设置负载均衡器的IP、端口、执行文件目录、配置文件目录。
- false  
通过节点安装负载均衡器。需要先创建SSH通讯方式的节点，可通过 `create-node-ssh` 子命令创建节点；同时节点需要启动节点代理nodemanager。

```
--agentip
```

节点代理监听IP。

```
--agentport
```

节点代理程序监听端口。

```
--balancerip
```

负载均衡器监听IP，不管是否引用外部负载均衡器都需要指定，访问时访问此IP。

```
--balancerport
```

负载均衡器监听端口，不管是否引用外部负载均衡器都需要指定，访问时访问此端口。

```
--executefilepath
```

负载均衡器执行文件目录，引用外部负载均衡器时需要设置。

```
--configfilepath
```

负载均衡器配置文件目录，引用外部负载均衡器时需要设置。

```
--algorithm
```

设置负载均衡算法。根据负载均衡器的类型 `--type` 指定。

如果 `--type` 为apache，可设置为：

- Requests：按权重，将根据服务器实例的权重由高至低访问。
- Traffic：按流量，将根据服务器实例所在的服务器流量由大至小访问。

- Busyness: 按繁忙程度, 将根据服务器实例所在的服务器 CPU 使用率由小至大访问。

如果 `--type` 为nginx, 可设置为:

- weight: 按权重, 将根据服务器实例的权重由高至低访问。
- ip\_hash: 按访问 IP 的 HASH (会话粘滞); 将根据用户请求过来的 IP, 然后映射成 hash 值, 然后分配到一个特定的服务器里面。实现会话粘滞需要使用该算法。
- fair: 按响应时间, 将根据服务器实例的响应时间由快至慢访问。
- url\_hash: 按访问 URL 的 HASH, 将根据 URL 的 HASH 访问。
- sticky: 按浏览器cookie (会话粘滞sticky), 将添加sticky模块。节点安装nginx时, 即 `--isexternal` 值为false, 才可以设置该项。

如果 `--type` 为alb, 可设置为:

- round\_robin: 轮询, 按时间顺序将请求轮流分配给后端服务器。
- ip\_hash: 按访问IP的HASH (会话粘滞), 将根据用户请求过来的 IP, 映射成 hash 值, 然后分配到一个特定的服务器里面。实现会话粘滞需要使用该算法。
- weight: 按权重, 根据服务器实例的权重由高至低访问。
- least\_conn: 按活跃连接数最少。

```
--failover
```

是否开启失效转移。 `--type` 为apache时可设置。

```
--receivetimeout
```

设置接收响应超时时间。

```
--connecttimeout
```

设置连接后端服务器超时时间。

```
--sockettimeout
```

设置请求后端处理超时时间。

```
--clientheadertimeout
```

设置请求头读取超时时间。

```
--clientbodytimeout
```

设置请求体读取超时时间。

```
--maxconnections
```

设置最大并发数。

```
--maxprocesses
```

最大进程数。

```
--sessionsticky
```

是否开启会话粘滞, 值为true/false。 `--type` 值为 apache 时可设置。

```
--staticcache
```

是否开启静态缓存, 值为true/false。

```
--ssl
```

是否开启SSL, 值为true/false。显示项, 不能修改。

```
--sslauthmethod
```

设置SSL认证方式。显示项，不能修改。可设置为：

- one-way: 单向认证。需要配置 `--keyfilepath` , `--crtfilepath` 。
- two-way: 双向认证。需要配置 `--keyfilepath` , `--crtfilepath` 、 `--chainfilepath` 。

```
--crtfilepath
```

设置SSL证书文件目录，如 server.crt。开启SSL认证时需要设置。显示项，不能修改。

```
--chainfilepath
```

设置SSL根证书文件目录，如 ca.crt。开启SSL认证时需要设置。显示项，不能修改。

```
--keyfilepath
```

设置SSL私钥证书文件，如server.key。开启SSL认证时需要设置。显示项，不能修改。

```
--gm
```

是否开启GM/T，值为true/false。开启后，需要配置对应的国密证书。当 `--type` 值为 `alb` 时可设置。显示项，不能修改。

`--sslauthmethod` 的值为one-way时，需要配置 `--keyfilepath` , `--crtfilepath` 、 `--keyfilepath2` , `--crtfilepath2` 。

`--sslauthmethod` 的值为two-way时，需要配置 `--keyfilepath` , `--crtfilepath` 、 `--keyfilepath2` , `--crtfilepath2` 、 `--chainfilepath` 。

```
--keyfilepath2
```

设置SSL私钥证书文件，如server.key。开启国密认证时需要设置。显示项，不能修改。

```
--crtfilepath2
```

设置SSL证书文件目录，如 server.crt。开启国密认证时需要设置。显示项，不能修改。

```
--noderef
```

负载均衡器安装的节点，节点安装负载均衡器时配置，即 `--isexternal` 值为 `false` 时设置。显示项，不能修改。

```
--licenseFilePath
```

设置ALB授权文件的路径，当 `--type` 的值为`alb` 时可设置。

```
--clustername
```

需要关联的服务器集群名称。

```
name
```

要更新的负载均衡器的名称。

### 8.5.3 示例

示例 1 更新负载均衡器信息

此示例更新负载均衡器nginx43的负载均衡算法为ip\_hash。

```
asadmin> update-balancer --algorithm ip_hash nginx43
Command update-balancer executed successfully.
```

示例 2 使用set子目录更新负载均衡器信息

此示例使用set子命令更新负载均衡器myalb12的负载均衡监听端口为8082。

```
asadmin> set lbs.load-balancers.load-balancer.myalb12.port=8082
lbs.load-balancers.load-balancer.myalb12.port | =8082 |
Command set executed successfully.
```

## 8.6 删除负载均衡器

`delete-balancer` 子命令删除负载均衡器。在执行之前，需要先停止运行该负载均衡器，且需要取消关联目标。

此命令仅在远程模式下受支持。

### 8.6.1 概要

```
delete-balancer [--help]
balancername
```

### 8.6.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
balancername
```

要删除的负载均衡器的名称。

### 8.6.3 示例

示例1 删除负载均衡器

此示例删除负载均衡器nginx43。

```
asadmin> delete-balancer nginx43
Command delete-balancer executed successfully.
```

## 9 负载均衡器集群

### 9.1 查看负载均衡器集群

`list-HALoadBalancer-cluster` 子命令查看负载均衡器集群。

此命令仅在远程模式下受支持。

#### 9.1.1 概要

```
list-HALoadBalancer-cluster [--whichtarget=domain]
```

#### 9.1.2 选项

```
--help  
-?
```

显示子命令的帮助文本。

```
--whichtarget
```

指定查询的域。

#### 9.1.3 示例

示例1 查询负载均衡器

该示例列出当前所有的负载均衡器集群。

```
asadmin> list-HALoadBalancer-cluster  
myaha started  
  
Command list-HALoadBalancer-cluster executed successfully.
```

### 9.2 查看负载均衡器集群实例

`list-HALoadBalancer-cluster` 子命令查看负载均衡器集群实例。

此命令仅在远程模式下受支持。

#### 9.2.1 概要

```
list-HALoadBalancer-instances [whichtarget]
```

#### 9.2.2 选项

```
--help  
-?
```

显示子命令的帮助文本。

```
whichtarget
```

指定查询负载均衡器集群名称。

#### 9.2.3 示例

示例1 查询负载均衡器集群实例

该示例列出负载均衡器集群myaha下的所有实例。

```
asadmin> list-HALoadBalancer-instances myaha
aha-1
<type=backup,version=v1.0.0,status=RUNNING>
aha-2
<type=master,version=v1.0.0,status=RUNNING>

Command list-HALoadBalancer-instances executed successfully.
```

## 9.3 创建负载均衡器集群

`register-HALoadBalancer-cluster` 子命令可以创建负载均衡器集群。在执行之前，需要先创建有负载均衡器，可使用 `create-balancer` 子命令创建，详见负载均衡器部分。

注意：

- 如果类型为keepalived，需要先安装有可用的keepalived。
- 如果类型为keepalived，负载均衡器安装服务器需要配置虚拟IP。

此命令仅在远程模式下受支持。

### 9.3.1 概要

```
register-HALoadBalancer-cluster --delayloop=delayloop
[--citedby=citedby] --advertint=advertint --delaybeforeretry=delaybeforeretry
--lbalgo=lbalgo --nbgetretry=nbgetretry
--type=type --persistencetimeout=persistencetimeout --targets=targets
--virtualip=virtualip [--virtualip2=virtualip2] [--domainname=domainname] [--tcpcheck=tcpcheck]
--connecttimeout=connecttimeout [--lbkind=lbkind]
name
```

### 9.3.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--delayloop
```

健康检查时间间隔。单位为秒。

```
--citedby
```

引用该负载均衡器集群的服务器集群。

```
--advertint
```

主备同步检查时间间隔。单位为秒。

```
--delaybeforeretry
```

失败重连间隔时间。单位为秒。

```
--lbalgo
```

负载均衡器策略，类型 `--keepalivedType` 为keepalived时设置，可设置为：

- rr

轮询

- wrr

加权轮询

- sh

源地址hash

- lc

最少连接

- wlc

加权最少连接

```
--nbgetretry
```

检测失败后重连次数。

```
--type
```

集群模式。可设置为 `master_backup` 或 `master_master`。

```
--persistencetimeout
```

会话保持超时时间。单位为秒。当 `--keepalivedType` 为 `keepalived` 时可设置。

```
--targets
```

关联的负载均衡器实例。格式为 负载均衡器名称(负载均衡器类型/负载均衡监听端口)，多个用英文逗号，分隔。例如 `--targets nginx1(nginx/8085),nginx2(nginx/8085)`。

```
--virtualip
```

虚拟IP，必须配置。

```
--virtualip2
```

虚拟IP2，当 `--keepalivedType` 为 `keepalived`，且集群模式为双主集群 `--type` 为 `master_master` 的时候需要设置。

```
--domainname
```

域名，当 `--keepalivedType` 为 `keepalived`，且集群模式为双主集群 `--type` 为 `master_master` 的时候需要设置，访问时通过该域名访问。

```
--tcpcheck
```

健康检测方法。

```
--connecttimeout
```

连接超时时间。单位为秒。当 `--keepalivedType` 为 `keepalived` 时可设置。

```
--lbkind
```

路由方法，必填，值为DR。

```
--keepalivedType
```

设置该集群的keepalived类型，可设置项为 `keepalived` 和 `aha`。

```
name
```

定义负载均衡器集群的名称。

### 9.3.3 示例

示例1 新建keepalived实例类型为keepalived的负载均衡器集群

该示例创建keepalived实例为keepalived的负载均衡器集群；引用的负载均衡器实例为nginx，nginx名称为nginx1和nginx2，监听端口为8082；集群模式为主备模式；虚拟IP为192.2.1.1；定义该集群名称为mynginxcluster。

```
asadmin> register-HALoadBalancer-cluster --type master_backup --lbkind DR --virtualIp 192.2.1.1 --
keepalivedType keepalived --delayloop 6 --advertint 3 --delaybeforeretry 3 --nbgetretry 3 --targets
nginx1(nginx/8082),nginx1(nginx/8082) mynginxcluster
Command register-HALoadBalancer-cluster executed successfully.
```

示例2 新建keepalived实例类型为Apusic HA的负载均衡器集群

该示例创建keepalived实例为Apusic HA的负载均衡器集群；引用的负载均衡器实例为alb，alb名称为alb1和alb2，监听端口为8083；集群模式为主备模式；虚拟IP为192.2.1.2；定义该集群名称为myahacluster。

```
asadmin> register-HALoadBalancer-cluster --type master_backup --lbkind DR --virtualIp 192.2.1.2 --
keepalivedType aha --delayloop 6 --advertint 3 --delaybeforeretry 3 --nbgetretry 3 --targets
myalb1(alb/8083),alb2(alb/8083) myahacluster
Command register-HALoadBalancer-cluster executed successfully.
```

## 9.4 新建负载均衡器集群实例

`create-HALoadBalancer-instance` 子命令新建负载均衡器集群实例。

此命令仅在远程模式下受支持。

### 9.4.1 概要

```
create-HALoadBalancer-instance --cluster=cluster --master=master [--action=action]
[--agentip=agentip] [--agentport=agentport] [--executefile=executefile]
[--configfile=configfile] [--type=keepalived] [--node=node] [--installdir=installdir]
[--healthurl=healthurl] [--priority=priority] name
```

### 9.4.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--cluster
```

所属的负载均衡器集群名称。

```
--master
```

设置为主实例，没有设置的表示为从实例。true或false。

```
--action
```

新建操作，必须设置为 `newInstance`。

```
--agentip
```

节点代理的IP。

```
--agentport
```

节点代理的端口。

```
--executefile
```

配置keepalived的执行文件，如 `${keepalived_root}/keepalived/sbin/keepalived`。不配置时，默认为 `/usr/sbin/keepalived`。实例的类型 `--type` 为 `keepalived`时可设置。

```
--configfile
```

配置keepalived的配置文件，如 `${keepalived_root}/keepalived/etc/keepalived/keepalived.conf`。不配置时，默认为 `/etc/keepalived/keepalived.conf`。实例的类型 `--type` 为 `keepalived`时可设置。

```
--type
```

实例的类型，`keepalived`或`aha`。

```
--node
```

节点的名称，表示在该节点安装Apusic HA实例。实例的类型 `--type` 为`aha`时可设置。

```
--installDir
```

指定Apusic HA安装目录。实例的类型 `--type` 为`aha`时可设置。

```
--priority
```

设置ApusicHA实例的优先级。当负载均衡器故障时，虚拟IP优先迁移到优先级高的实例上，限制范围0-255。主实例的优先级需要高于备实例。实例的类型 `--type` 为`aha`时可设置。

```
--healthUrl
```

设置当前负载均衡器状态的检查url，用于检查负载均衡器的健康情况。实例的类型 `--type` 为`aha`时可设置。

```
name
```

定义新建的实例名称。

### 9.4.3 示例

#### 示例 1 新建Apusic HA集群实例

该示例新建Apusic HA实例，命名为`myaha1`；所属负载均衡器集群为`myahacluster`，所属节点为`node1`，为主实例，优先级为100。

```
asadmin> create-HALoadBalancer-instance --cluster myahacluster --master true --node node1 --action
newInstance --type aha --priority 100 myaha1
```

实例 `myaha1`在主机 `192.2.1.1` 上创建

```
Command create-HALoadBalancer-instance executed successfully.
```

#### 示例2 新建keepalived集群实例

该示例新建keepalived实例，命名为`mykeepalived1`，所属负载均衡器集群为`mynginxcluster`，节点代理IP为`192.2.1.2`，节点代理端口为`1099`，为主实例。

```
asadmin> create-HALoadBalancer-instance --cluster mynginxcluster --master true --action newInstance --type
keepalived --agentip 192.2.1.2 --agentport 1099 mykeepalived1
```

```
Command create-HALoadBalancer-instance executed successfully.
```

## 9.5 更新负载均衡器集群属性

`update-loadBalancer-cluster` 子命令更新负载均衡器集群的属性。

此命令仅在远程模式下受支持。

### 9.5.1 概要

```
update-loadBalancer-cluster [--lbalgo=lbalgo] [--advertint=advertint] [--connecttimeout=connecttimeout]
[--delaybeforeretry=delaybeforeretry] [--nbgetretry=nbgetretry] [--delayloop=delayloop]
[--persistencetimeout=persistencetimeout]
name
```

### 9.5.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--delayloop
```

健康检查时间间隔。单位为秒。

```
--citedby
```

引用该负载均衡器集群的服务器集群。

```
--advertint
```

主备同步检查时间间隔。单位为秒。

```
--delaybeforeretry
```

失败重连间隔时间。单位为秒。

```
--lbalgo
```

负载均衡器策略，类型 `--keepalivedType` 为 `keepalived` 时设置，可设置为：

- rr  
轮询
- wrr  
加权轮询
- sh  
源地址hash
- lc  
最少连接
- wlc  
加权最少连接

```
--nbgetretry
```

检测失败后重连次数。

```
--persistencetimeout
```

会话保持超时时间。单位为秒。当 `--keepalivedType` 为 `keepalived` 时可设置。

```
--connecttimeout
```

连接超时时间。单位为秒。当 `--keepalivedType` 为 `keepalived` 时可设置。

```
name
```

负载均衡器集群的名称。

### 9.5.3 示例

示例1 更新负载均衡器集群信息

该示例更新负载均衡器集群 `myahacluster` 的检测失败后重连次数为3。

```
asadmin> update-loadBalancer-cluster --nbgetretry 3 myahacluster
Command update-loadBalancer-cluster executed successfully.
```

## 9.6 更新负载均衡器集群中的负载均衡器实例

`update-loadBalancer-cluster-targets` 子命令更新负载均衡器集群中的负载均衡器实例。更新负载均衡器实例时，需要注意一个集群中的负载均衡器类型和端口需要一致。更新后需要重启负载均衡器集群才能生效。

此命令仅在远程模式下受支持。

### 9.6.1 概要

```
update-loadBalancer-cluster-targets [--targets=targets] name
```

### 9.6.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--targets
```

指定负载均衡器实例，多个用英文逗号，分隔。

```
name
```

负载均衡器集群的名称。

### 9.6.3 示例

示例1 更新负载均衡器实例

该示例更新负载均衡器集群 `myahacluster` 中的负载均衡器实例为 `myalb1,myalb2`。

```
asadmin> update-loadBalancer-cluster-targets --targets myalb1,myalb2 myahacluster
Command update-loadBalancer-cluster-targets executed successfully.
```

## 9.7 更新ApusicHA实例信息

`update-aha-instance` 子命令更新Apusic HA负载均衡器集群中的Apusic HA实例信息，当负载均衡器的类型为 `aha` 时可以使用该子命令。

此命令仅在远程模式下受支持。

### 9.7.1 概要

```
update-aha-instance [--healthurl=healthurl] [--priority=priority] name
```

### 9.7.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--priority
```

设置ApusicHA实例的优先级。当负载均衡器故障时，虚拟IP优先迁移到优先级高的实例上，限制范围0-255。主实例的优先级需要高于备实例。实例的类型 `--type` 为aha时可设置。

```
--healthUrl
```

设置当前负载均衡器状态的检查url，用于检查负载均衡器的健康情况。实例的类型 `--type` 为aha时可设置。

```
name
```

AHA实例的名称。

### 9.7.3 示例

示例1 更新AHA实例信息

该示例更新AHA实例myaha1的优先级为99。

```
asadmin> update-aha-instance --priority 99 myaha1
Command update-aha-instance executed successfully.
```

## 9.8 启动负载均衡器集群

`start-loadBalancer-cluster` 子命令启动负载均衡器集群。在执行之前，集群需要有集群实例。

此命令仅在远程模式下支持。

### 9.8.1 概要

```
start-HALoadBalancer-cluster name
```

### 9.8.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
name
```

要启动的负载均衡器集群的名称。

### 9.8.3 示例

示例1 启动负载均衡器集群

该示例启动负载均衡器集群myahacluster

```
asadmin> start-HALoadBalancer-cluster myahacluster
启动负载均衡器集群myahacluster成功
Command start-HALoadBalancer-cluster executed successfully.
```

## 9.9 启动负载均衡器集群实例

`start-HALoadBalancer-instance` 子命令启动负载均衡器集群实例。

此命令仅在远程模式下支持。

### 9.9.1 概要

```
start-HALoadBalancer-instance name
```

### 9.9.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
name
```

要启动的负载均衡器集群实例的名称。

### 9.9.3 示例

示例1 启动负载均衡器集群实例

该示例启动负载均衡器集群实例myaha1。

```
asadmin> start-HALoadBalancer-instance myaha1
实例 myaha1 成功启动
Command start-HALoadBalancer-instance executed successfully.
```

## 9.10 停止负载均衡器集群

`stop-HALoadBalancer-cluster` 子命令停止运行负载均衡器集群。

此命令仅在远程模式下受支持。

### 9.10.1 概要

```
stop-HALoadBalancer-cluster name
```

### 9.10.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
name
```

要停止的负载均衡器集群的名称。

### 9.10.3 示例

示例1 停止运行负载均衡器集群

该示例停止运行负载均衡器集群myahacluster。

```
asadmin> stop-HALoadBalancer-cluster myahacluster
停止负载均衡器集群myahacluster成功
Command stop-HALoadBalancer-cluster executed successfully.
```

## 9.11 停止负载均衡器集群实例

`stop-HALoadBalancer-instance` 子命令删除负载均衡器。

此命令仅在远程模式下受支持。

### 9.11.1 概要

```
stop-HALoadBalancer-instance name
```

### 9.11.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
name
```

要停止运行的负载均衡器集群实例的名称。

### 9.11.3 示例

示例1 删除负载均衡器

```
asadmin> stop-HALoadBalancer-instance myaha1
实例 myaha1 成功停止
Command stop-HALoadBalancer-instance executed successfully.
```

## 10 监视管理

### 10.1 开启监控服务

`enable-monitoring` 子命令用于在运行时打开对Apusic Server或特定模块的监控。更改不需要重新启动服务器实时生效。

默认情况下，监控服务是启用的，即 `monitoring-service` 的值为true。但是，单个模块的默认监控级别为OFF。需要使用 `--modules` 选项开启监控组件，将监控组件监视级别设置为HIGH或LOW。

#### 10.1.1 概要

```
enable-monitoring [--help]
[--mbean={false|true}]
[--dtrace={true|false}]
[--modules modules[=level][:module[=level]]*]
[--pid pid]
[--options options={true|false}]
```

#### 10.1.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--mbean
```

启用Mbean监控。默认值为false。

```
--dtrace
```

仅在存在DTrace监控模块时可用。启用Oracle Solaris DTrace监控。默认值为false。

```
--modules
```

通过设置监视级别来启用指定的一个或多个模块。有效级别为OFF、HIGH、LOW。如果未指定级别，则默认设置为OFF。多个模块之间用：（冒号）分隔。可监视模块包括 `connector-connection-pool`、`connector-service`、`ejb-container`、`http-service`、`jdbc-connection-pool`、`jersey`、`jpa`、`jms-service`、`jvm`、`security`、`thread-pool`、`transaction-service`、`web-container`、`web-services-container`。可以使用 `get` 子命令列出其他模块。

```
--pid
```

指定Apusic Server JVM进程识别号（PID）。启用监视时，根据指定的PID附加btrace-agent。仅在系统无法确定PID的特殊情况下才需要指定。在这种情况下，子命令会提示输入相应Apusic Serverprocess的PID。

```
--options
```

设置以下 `btrace-agent` 选项：

`debug`：为BTrace启用调试。默认值为false。

#### 10.1.3 示例

示例1：开启监控服务

```
asadmin> enable-monitoring
Command enable-monitoring executed successfully
```

示例2：为Web和EJB容器启用监控

此示例通过设置特定容器的监控级别来启用对特定容器的监控。

```
asadmin> enable-monitoring --modules web-container=LOW:ejb-container=HIGH
Command enable-monitoring executed successfully.
```

示例3: 为监控打开调试

此示例打开调试。

```
asadmin> enable-monitoring --options debug=true
Command enable-monitoring executed successfully
```

## 10.2 关闭监控服务

`disable-monitoring` 子命令用于在运行时关闭对Apusic Server或特定模块的监控。更改不需要重新启动服务器实时生效。

运行不带 `--module` 选项的 `disable-monitoring` 子命令会将 `monitoring-service` 元素的 `monitoring-enabled` 属性设置为false, 从而禁用监控服务。各个模块保留其监控级别, 但由于整个监控服务被禁用, 因此不会生成监控数据。

此子命令与 `--modules` 选项一起使用, 通过将监控级别设置为OFF来禁用对模块的监控。

禁用监视的另一种方法是使用set子命令。在这种情况下, 必须重新启动服务器才能使更改生效。

此子命令仅在远程模式下受支持。

### 10.2.1 概要

```
disable-monitoring [--help] [--modules module-name][:module-name]*
```

### 10.2.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--modules
```

通过设置监视级别来启用指定的一个或多个模块。有效级别为OFF、HIGH、LOW。如果未指定级别, 则默认设置为OFF。多个模块之间用: (冒号) 分隔。可监视模块包括 `connector-connection-pool`、`connector-service`、`ejb-container`、`http-service`、`jdbc-connection-pool`、`jersey`、`jpa`、`jms-service`、`jvm`、`security`、`thread-pool`、`transaction-service`、`web-container`、`web-services-container`。可以使用 `get` 子命令列出其他模块。

### 10.2.3 示例

示例1: 禁用Apusic Server的监控服务

此示例通过将enable-监控标志设置为false (默认为true) 来禁用Apusic Server的监控。

```
asadmin> disable-monitoring
Command disable-monitoring executed successfully
```

示例2: 禁用对Web和EJB容器的监控

此示例禁用对特定容器的监控。它们的监控级别将设置为OFF。

```
asadmin> disable-monitoring --modules web-container:ejb-container
Command disable-monitoring executed successfully
```

## 10.3 监视管理

`monitor` 子命令常用监视的Apusic Server组件和服务的统计信息。必须使用 `--type` 选项来指定要为其显示统计信息的对象。数据以表格形式连续显示，或者可以使用 `--interval` 选项以特定时间间隔显示数据。在监视组件或服务之前，需要通过管理控制台，或 `enable-monitoring` 子命令，或 `set` 子命令为组件或服务启用监视（设置为HIGH或LOW）。`monitor` 子命令具有筛选结果和在逗号分隔值（CSV）文件中捕获输出的选项。输出以表格形式显示。要查看表头的图例，请键入 `h`。此子命令仅在本地模式下受支持。

### 10.3.1 概要

```
monitor [--help]
--type type
[--interval interval]
instance-name
```

### 10.3.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--type
```

要监视的组件或服务。此选项是必需的。未定义默认值。

- `httplistener`

对于这种类型，属性 `server.monitor-service.module-monitor-levels.http-service` 必须设置为LOW或HIGH。

显示HTTP侦听器服务的以下统计信息：

- `ec`  
处理HTTP请求时的错误总数。
- `mt`  
处理单个HTTP请求的最长响应时间（以毫秒为单位）。
- `pt`  
HTTP侦听器服务处理HTTP请求所花费的总时间（以毫秒为单位）。
- `rc`  
HTTP侦听器服务已处理的请求总数。

- `jvm`

对于这种类型，属性 `server.server-config.monitoring-service.module-monitoring-levels.jvm` 必须设置为LOW或HIGH。

显示Java平台虚拟机（Java虚拟机或JVM机器）的以下统计信息：

- `Uptime`  
JVM机器自上次启动以来运行的毫秒数。
- `min`  
JVM机器在启动期间从操作系统请求用于内存管理的初始内存量（以字节为单位）。
- `max`  
可用于内存管理的最大内存量。
- `low`  
保留以与其他版本兼容。

- hogh

保留以与其他版本兼容。

- count 保证JVM机器可用的内存量（以字节为单位）。

- webmodule

对于这种类型，属性 `server.server-config.monitoring-service.module-monitoring-levels.web-container` 必须设置为LOW或HIGH。

显示所有已部署web模块的以下统计信息：

- asc

当前活动会话的数量。

- ast

当前处于活动状态或以前处于活动状态的会话总数。

- rst

被拒绝的会话总数。

- st

已创建的会话总数。

- ajlc

加载的当前活动的JavaServer Pages (JSP) 技术页面的数量。

- mjlc

任何时候同时活动的JSP技术页面的最大数量。

- tjlc

已加载的JSP技术页面总数。

- aslc

加载的当前活动的Java servlet的数量。

- mslc

任何时候同时处于活动状态的Java servlet的最大数量。

- tslc 已加载的Java Servlet总数

```
--interval
```

捕获监视属性之前的间隔（秒）。间隔必须大于0。在您键入Control-C或q之前，监控属性将显示在stdout上。默认值为30。

```
instance-name
```

指定需要查看的实例，默认查看server。

### 10.3.3 示例

示例1 监控JVM信息

```
asadmin> monitor --type=jvm --interval 2000 server
                                     NCLS-SRVRMGMT-00005
NCLS-SRVRMGMT-00006                 NCLS-SRVRMGMT-00007
current          min          max          low          high          coun
611791404        260505600  1908932607  0             0             1265278976
```

示例2 监控httplistener信息

```
asadmin> monitor --type httplistener --filter http-listener-1 server
ec    mt    pt    rc
0     627  118.00  7
0     627  38.00  22
```

## 10.4 监控阻塞线程

`monitor-blockingthread` 子命令监控阻塞线程。

此命令仅在本地模式下受支持。

### 10.4.1 概要

```
monitor-blockingthread [--target=server] [--method=get]
```

### 10.4.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--target
```

指定监控的实例，默认server。

```
--method
```

指定监控方法。

### 10.4.3 示例

示例1 监控阻塞线程

```
asadmin> monitor-blockingthread --method=get
Command monitor-blockingthread executed successfully.
```

## 10.5 监控繁忙线程

`monitor-busythread` 子命令监控繁忙线程。

此命令仅在本地模式下受支持。

### 10.5.1 概要

```
monitor-busythread [--target=server] [--method=get]
```

### 10.5.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--target
```

指定监控的实例，默认server。

```
--method
```

指定监控方法。

### 10.5.3 示例

示例1 监控繁忙线程

```
asadmin> monitor-busythread --method=get
Command monitor-busythread executed successfully.
```

## 10.6 监控长线程

`monitor-longthread` 子命令监控超长线程。

此命令仅在本地模式下受支持。

### 10.6.1 概要

```
monitor-longthread [--target=server]
```

### 10.6.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--target
```

指定监控的实例，默认server。

### 10.6.3 示例

示例1 监控长线程

```
asadmin> monitor-longthread
Command monitor-longthread executed successfully.
```

## 10.7 生成火焰图

`monitor-flamegraph` 子命令监控阻塞线程。

此命令仅在本地模式下受支持。

### 10.7.1 概要

```
monitor-flamegraph [--target=server] [--method=process] [--duration=10] [--format=flamegraph]
```

### 10.7.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--target
```

指定监控的实例，默认server。

```
--method
```

指定监控方法, process。

```
--duration
```

监控持续时间。

```
--format
```

指定生成报告格式, flamegraph或jfr。

### 10.7.3 示例

示例1 生成火焰图

此示例设置监控持续时长为10秒, 生成报告格式为火焰图。

```
asadmin> monitor-flamegraph --target server --method process --duration 10 --format flamegraph  
  
Command monitor-flamegraph executed successfully.
```

## 11 JMS资源

### 11.1 查看JMS资源

`list-jms-resources` 子命令列出了现有的Java消息服务 (JMS) 资源 (目标和连接工厂资源)。

此子命令仅在远程模式下受支持。远程`asadmin`子命令需要正在运行的域管理服务器 (DAS)。

#### 11.1.1 概要

```
list-jms-resources [--help]
[--restype type]
[target]
```

#### 11.1.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--target
```

指定JMS资源所在的实例，可选项有`server`、`domain`、`cluster-name`、`instance-name`。

```
--restype
```

需要查看的JMS资源类型，可以是 `javax.jms.Topic`、`javax.jms.Queue`、`javax.jms.ConnectionFactory`、`javax.jms.TopicConnectionFactory`，或 `javax.jms.QueueConnectionFactory`。

#### 11.1.3 示例

示例1 查看JMS资源

此示例查看所有的JMS资源。

```
asadmin> list-jms-resources
jms/Queue
jms/ConnectionFactory
jms/DurableConnectionFactory
jms/Topic
Command list-jms-resources executed successfully.
```

示例2 根据资源类型查看JMS资源

此示例查询JMS资源类型为 `javax.jms.ConnectionFactory` 的资源。

```
asadmin> list-jms-resources --restype javax.jms.ConnectionFactory
jms/ConnectionFactory
jms/DurableConnectionFactory
Command list-jms-resources executed successfully.
```

## 11.2 创建JMS资源

`create-jms-resource` 子命令创建Java消息服务 (jms) 连接工厂资源或jms目标资源。

此命令在远程模式下支持。

#### 11.2.1 概要

```
create-jms-resource [--help]
--restype type
[--target target]
[--enabled={true| false}]
[--description text]
[--property (name=value)[:name=value]*]
[--force={false| true}]
jndi_name
```

### 11.2.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--target
```

指定所属的实例，默认server。

```
--restype
```

设置JMS资源类型，可设置为 `javax.jms.Topic`、`javax.jms.Queue`、`javax.jms.ConnectionFactory`、`javax.jms.TopicConnectionFactory`，或 `javax.jms.QueueConnectionFactory`。

```
--enabled
```

设置启用状态，可选项为 `true` 或 `false`。

```
--description
```

对JMS资源的描述。

```
--property
```

用于配置JMS资源的可选属性名称/值对。可以为连接工厂资源指定以下属性：

- ClientID  
持久订户将使用的连接工厂的客户端ID。
- AddressList  
以逗号分隔的消息队列地址列表，指定应用程序将与之通信的一个或多个消息代理实例的主机名（以及可选的端口号）。例如，该值可以是地球或地球:7677。如果消息代理在默认端口（7676）以外的端口上运行，请指定端口号。默认值是由服务器JMS服务配置中定义的JMS主机组成的地址列表。默认值为localhost，默认端口号为7676。客户端将尝试连接到本地主机的端口7676上的代理。
- UserName  
连接工厂的用户名称，默认为guest。
- Password  
连接工厂的密码，默认值为guest。
- ReconnectEnabled  
值true表示当连接丢失时，客户端运行时尝试重新连接到消息服务器（或地址列表中的地址列表）。默认值为false。
- ReconnectAttempts  
在客户端运行时尝试列表中的下一个地址之前，对AddressList中的每个地址进行连接（或重新连接）的尝试次数。值-1表示重新连接尝试的次数不受限制（客户端运行时尝试连接到第一个地址，直到成功为止）。默认值为6。

- ReconnectInterval

重新连接尝试之间的间隔（毫秒）。这适用于对地址列表中的每个地址以及列表中连续地址的尝试。如果间隔太短，则代理没有时间恢复。如果时间太长，重新连接可能代表不可接受的延迟。默认值为30000毫秒。

- AddressListBehavior

指定连接尝试是否按地址顺序排列。地址列表（优先级）或随机顺序（随机）。优先级意味着重新连接将始终尝试连接到地址列表中的第一个服务器地址，并且只有在第一个代理不可用时才会使用另一个地址。如果有许多客户端尝试使用同一连接工厂进行连接，请指定RANDOM以防止它们都连接到同一地址。默认值是服务器JMS服务配置的AddressListBehavior值。

- AddressListIterations

客户端运行时迭代AddressList以建立（或重新建立）连接的次数。值-1表示尝试次数不受限制。默认值为-1。

此外，您可以将连接器连接池属性指定为连接器资源属性。您可以为目标资源指定以下属性：

- Name

资源将引用的物理目标的名称。当您运行使用目标资源的应用程序时，会自动创建物理目标。您还可以使用 `create-jmsdest` 子命令创建物理目标。如果不指定此属性，JMS服务将创建一个与目标资源同名的物理目标（将JNDI名称中的任何正斜杠替换为下划线）。

- Description

物理目的地的描述。

```
--force
```

指定子命令是否覆盖同名的现有JMS资源。默认值为false。

```
jndi_name
```

定义的JMS连接工厂的JNDI名称。

### 11.2.3 示例

示例1: 创建JMS连接工厂

示例创建资源类型为 `javax.jms.ConnectionFactory` 连接工厂。JNDI名称为 `jms/DurableConnectionFactory`。`ClientId` 属性在连接工厂上设置客户端ID，以用于持久订阅。JMS资源的JNDI名称通常包括 `jms/` 命名子上下文。

```
asadmin> create-jms-resource --restype javax.jms.ConnectionFactory --description "connection factory for durable subscriptions"
--property ClientId=MyID jms/DurableConnectionFactory
Connector resource jms/DurableConnectionFactory created.
Command create-jms-resource executed successfully.
```

示例2: 创建JMS目标资源 以下子命令创建JNDI名称为 `jms/MyQueue`，资源类型为 `javax.jms.Queue` 的目标资源。`Name` 属性指定资源引用的物理目标。

```
asadmin> create-jms-resource --restype javax.jms.Queue --property Name=PhysicalQueue jms/MyQueue
Administered object jms/MyQueue created.
Command create-jms-resource executed successfully
```

## 11.3 删除JMS资源

`delete-jms-resource` 子命令删除指定的Java消息服务（JMS）资源。请确保在执行此子命令之前删除对此资源的所有引用。

此命令仅在远程模式下支持。

### 11.3.1 概要

```
delete-jms-resource [--help]
[--target target]
jndi_name
```

### 11.3.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--target
```

指定需要删除的JMS资源所在的实例，可选项server、domain、cluster-name、instance-name。

```
jndi_name
```

需要删除的JMS资源的JNDI名称。

### 11.3.3 示例

示例1 删除JMS资源

```
asadmin> delete-jms-resource jms/MyQueue
Administered object jms/MyQueue deleted.
Command delete-jms-resource executed successfully.
```

## 11.4 查看JMS物理目标

`list-jmsdest` 子命令列出Java消息服务（JMS）物理目标。

此子命令仅在远程模式下受支持。远程asadmin子命令需要正在运行的域管理服务器（DAS）。

### 11.4.1 概要

```
list-jmsdest [--help]
[--desttype type]
[target]
```

### 11.4.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--target
```

指定JMS资源所在的实例，可选项有server、domain、cluster-name、instance-name。

```
--desttype
```

需要查看的JMS物理资源类型，可以是 topic 和 queue 。

### 11.4.3 示例

示例1：查看JMS物理目标

此示例查看所有的JMS物理目标。

```
asadmin> list-jmsdest
PhysicalQueue
PhysicalTopic
Command list-jmsdest executed successfully.
```

## 11.5 创建JMS物理目标

`create-jmsdest` 子命令子命令创建Java消息服务 (JMS) 物理目标。通常, 使用 `create-jms-Resource` 子命令创建JMS目标资源, 该资源具有指定物理目标的 `Name` 属性。当运行使用目标资源的应用程序时, 会自动创建物理目标。如果要使用非默认属性设置创建物理目标, 请使用 `create-jmsdest` 子命令。

仅在远程模式下支持此子命令。远程asadmin子命令需要正在运行的域管理服务器 (DAS)。

### 11.5.1 概要

```
create-jmsdest [--help]
--desttype dest_type
[--property (name=value)[:name=value]*]
[--target target]
dest_name
```

### 11.5.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--desttype
```

JMS目标的类型。有效值是 `topic` 和 `queue`。

```
--property
```

用于配置物理目的地的可选属性 `name/value`。您可以为物理目的地指定以下属性。

- `maxNumMsgs`

目标允许的最大未使用消息数。值-1表示消息数量不受限制。默认值为-1。对于死消息队列, 默认值为1000。

如果限制行为属性设置为 `FLOW_CONTROL`, 则可能会超过指定的消息限制, 因为代理无法足够快地做出反应以停止传入消息的流。在这种情况下, 为 `maxNumMsgs` 指定的值仅作为代理的提示, 而不是严格强制的限制。

- `maxBytesPerMsg`

任何单个消息的最大大小 (以字节为单位)。拒绝持久消息会报告给生成客户端, 但有例外; 不会为非持久消息发送通知。该值可以用字节、千字节或兆字节表示, 使用以下后缀:

- `b`: 字节
- `k`: 千字节 (1024 字节)
- `m`: 兆字节 (1024 x 1024=1,048,576字节)

没有后缀的值以字节表示; 值-1表示无限的消息大小。默认值为-1。

- `maxTotalMsgBytes`

未使用消息的最大总内存, 以字节为单位。默认值为-1。语法与 `maxBytesPerMsg` 相同。对于死消息队列, 默认值为10m。

- `limitBehavior`

当消息队列代理达到内存限制阈值时的行为。有效取值如下。

- `REJECT_NEWEST`: 拒绝最新消息, 只有当消息是持久的时才通知生产客户端异常。这是默认值。

- FLOW\_CONTROL: 减慢消息生产者发送消息的速率。
- REMOVE\_OLDEST: 丢弃最旧的消息。
- REMOVE\_LOW\_PRIORITY: 根据年龄丢弃优先级最低的消息, 不通知生产客户端。

如果该值 REMOVE\_OLDEST 或 REMOVE\_LOW\_PRIORITY 并且 useDMQ 属性设置为true, 则将多余的消息移动到死消息队列。对于死消息队列本身, REMOVE\_OLDEST 默认限制行为, 并且该值不能设置为 FLOW\_CONTROL。

- maxNumProducers

目标的最高消息生产者数量。达到此限制时, 不能创建新的生产者。值-1表示生产者数量不受限制。默认值为100。此属性不适用于死消息队列。

- consumerFlowLimit

单个批处理中可以传递给使用者的最大消息数。值-1表示消息数量不受限制。默认值为1000。客户端运行时可以通过在连接工厂对象上指定较低的值来覆盖此限制。在负载均衡队列传递中, 这是负载均衡开始之前路由到活动使用者的排队消息的初始数量。

- useDMQ

如果设置为true, 则死消息进入死消息队列。如果设置为false, 则丢弃死消息。默认值为真。

- validateXMLSchemaEnabled

如果设置为true, 则为目标启用XML模式验证。默认值为false。启用XML验证后, 消息队列客户端运行时将尝试根据指定的XSD (如果未指定XSD, 则根据DTD) 验证XML消息, 然后再将其发送到代理。如果无法找到指定的模式或无法验证消息, 则不发送消息, 并抛出异常。当目标处于非活动状态时, 即当目标中没有消费者或生产者以及目标中没有消息时, 应设置此属性。否则生产者必须重新连接。

XMLSchemaURLList 以空格分隔的XML模式文档 (XSD) URI字符串列表。如果 validateXMLSchemaEnabled 设置为true, 则URI指向用于XML模式验证的一个或多个XSD的位置。默认值为null。如果指定了多个URI, 则在此值周围使用双引号, 如下例所示:

```
"http://foo/flap.xsd http://test.com/test.xsd"
```

如果未设置此属性或为null并且启用了XML验证, 则使用XML文档中指定的DTD执行XML验证。如果XSD因应用程序需求更改而更改, 则所有基于更改后的XSD生成XML消息的客户端应用程序都必须重新连接到代理。

要修改这些属性的值, 您可以使用 `aas-install/mq/bin/imqcmd` 命令。

```
--target
```

仅为指定目标创建物理目标。尽管 `create-jmsdest` 子命令与资源相关, 但使用JMS Service (JMS Broker) 创建物理目标, 这是配置的一部分。在 `domain.xml` 的config部分中配置了JMS Broker。有可选项server、domain、cluster-name、instance-name。

```
dest_name
```

要创建的JMS目标的唯一标识符。

### 11.5.3 示例

示例1: 创建JMS物理目标

以下子命令创建一个名为物理队列的JMS物理队列, 其中包含非默认属性值。

```
asadmin> create-jmsdest --desttype queue
--property maxNumMsgs=1000:maxBytesPerMsg=5k PhysicalQueue
Command create-jmsdest executed successfully.
```

## 11.6 删除JMS物理目标

`delete-jms-resource` 子命令删除指定的Java消息服务 (JMS) 物理目标。

此命令仅在远程模式下支持。

### 11.6.1 概要

```
delete-jmsdest [--help]
--desttype type
[--target target]
dest_name
```

### 11.6.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--desttype
```

物理目标的类型，可以为 topic 或 queue 。

```
--target
```

指定需要删除的JMS物理目标所在的实例，可选项server、domain、cluster-name、instance-name。

```
jndi_name
```

需要删除的JMS资源的物理目标名称。

### 11.6.3 示例

示例1 删除JMS物理目标

```
asadmin> delete-jmsdest --desttype queue PhysicalQueue
Command delete-jmsdest executed successfully.
```

## 12 共享类库

### 12.1 新建共享类库

`create-shared-lib` 子命令用于创建共享类库。

此命令仅在远程模式下支持。

#### 12.1.1 概要

```
create-shared-lib --libname=libname [--path=path]
[--target=target] [--secretlevel=SECERT]
```

#### 12.1.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--target
```

指定需要删除的JMS资源所在的实例，可选项server、domain、cluster-name、instance-name。

```
--libname
```

需要删除的JMS资源的JNDI名称。

```
--path
```

该共享类库的路径。

```
--secretlevel
```

该共享类库的级别，可以为 SECRET (秘密)、CONFIDENTIAL (机密)、TOPSECRET (绝密)。

#### 12.1.3 示例

示例1: 新建共享类库

该示例新建名为mylib的共享类库，共享类库路径为 /opt/testz/mylib。

```
asadmin> create-shared-lib --libname mylib --path "/opt/testz/mylib"
create mylib successfully.
Command create-shared-lib executed successfully.
```

## 12.2 删除共享类库资源

`delete-shared-lib-jars` 子命令用于删除类库中的类。

此命令仅在远程模式下支持。

#### 12.2.1 概要

```
delete-shared-lib-jars --libname=libname [--jarnames=jarnames] [--deleteall=false]
```

#### 12.2.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--libname
```

指定需要删除的共享类库名称

```
--jarnames
```

需要删除的共享类库的类名称。

```
--deleteall
```

是否删除所有的类，默认值为false。

### 12.2.3 示例

示例1: 删除共享类库中的某个类资源

该示例删除共享类库mylib，目录下的shared.jar被删除。

```
asadmin> delete-shared-lib-jars --libname mylib --jarnames shared.jar
Command delete-shared-lib-jars executed successfully.
```

示例2: 删除共享类库中的所有类资源

该示例删除共享类库mylib，目录下的所有类资源。

```
asadmin> delete-shared-lib-jars --libname mylib --deleteall true
Command delete-shared-lib-jars executed successfully.
```

## 12.3 删除共享类库

delete-shared-lib 子命令用于删除类库。

此命令仅在远程模式下支持。

### 12.3.1 概要

```
delete-shared-lib [--target=target] jndi_name
```

### 12.3.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--target
```

指定需要删除的共享类库所在的实例，可选项server、domain、cluster-name、instance-name。

```
jndi_name
```

需要删除的共享类库名称。

### 12.3.3 示例

示例1 删除共享类库

该示例删除共享类库mylib，目录下所有的文件被删除。

```
asadmin> delete-shared-lib mylib  
Command delete-shared-lib executed successfully.
```

## 13 JavaMail会话

### 13.1 查看JavaMail会话

`list-javamail-resources` 子命令查看JavaMail会话资源。

此命令仅在远程模式下支持。

#### 13.1.1 概要

```
list-javamail-resources [--help]
[target]
```

#### 13.1.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--target
```

指定JMS资源所在的实例，可选项有server、domain、cluster-name、instance-name。

#### 13.1.3 示例

示例1 查看JavaMail会话资源

此示例查看所有的JavaMail会话资源。

```
asadmin> list-javamail-resources
mail/MyMailSession
Command list-javamail-resources executed successfully.
```

## 13.2 新建JavaMail会话

`create-javamail-resource` 子命令用于新建JavaMail会话资源。

此命令仅在远程模式下支持。

#### 13.2.1 概要

```
create-javamail-resource [--help]
[--target target] --mailhost hostname
--mailuser username --fromaddress address [--storeprotocol storeprotocol]
[--storeprotocolclass storeprotocolclass] [--transprotocol transprotocol]
[--transprotocolclass transprotocolclass] [--debug={false| true}]
[--enabled={true| false}]
[--description resource-description] [--property (name=value)[:name=value]*]
jndi-name
```

#### 13.2.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--target
```

指定所属的实例，可选项server、domain、cluster-name、instance-name。

```
--mailhost
```

默认邮件服务器的 DNS 名称。如果未提供特定于协议的主机属性，存储对象和传输对象的连接方法将使用此值。该名称必须可解析为实际的主机名。

```
--mailuser
```

连接到邮件服务器时提供的邮件帐户用户的名称。如果未提供特定于协议的用户名属性，则存储和传输对象的连接方法使用此值。

```
--fromaddress
```

默认用户的电子邮件地址，格式为 `username@host.domain`。

```
--storeprotocol
```

邮件服务器存储协议。默认值为 `imap`。仅当已将Apusic Server的邮件提供程序重新配置为使用非默认存储协议时才更改此值。。

```
--storeprotocolclass
```

邮件服务器存储协议类名。默认值为 `com.sun.mail.imap.IMAPStore`。仅当已将Apusic Server的邮件提供程序重新配置为使用非默认存储协议时才更改此值。。

```
--transprotocol
```

邮件服务器传输协议。默认值为 `smtp`。仅当已将Apusic Server的邮件提供程序重新配置为使用非默认传输协议时，才更改此值。

```
--transprotocolclass
```

邮件服务器传输协议类名。默认值为 `com.sun.mail.smtp.SMTPTransport`。仅当已将Apusic Server的邮件提供程序重新配置为使用非默认传输协议时，才更改此值。

```
--debug
```

如果设置为true，服务器将以调试模式启动此资源。如果JavaMail日志级别设置为 `FINE` 或 `FINER`，则将生成调试输出并将其包含在服务器日志文件中。默认值为false。

```
--enabled
```

如果设置为true，则资源将在运行时启用。默认值为true。

```
--description
```

提供JavaMail资源的一些详细信息的文本。

```
--property
```

用于配置JavaMail资源的可选属性名称/值对。Apusic Server特定的邮件前缀被转换为标准邮件前缀 `mail-`。

```
jndi_name
```

要创建的JavaMail资源的JNDI名称。建议对JavaMail资源使用命名子上下文前缀 `mail/`。

### 13.2.3 示例

示例1 新建JavaMail会话

此示例创建了一个名为mail/MyMailSession的JavaMail会话资源。

```
asadmin> create-javamail-resource --mailhost localhost --mailuser sample --fromaddress sample@sun.com
mail/MyMailSession
```

```
Mail Resource mail/MyMailSession created.  
Command create-javamail-resource executed successfully.
```

### 13.3 删除JavaMail会话

`delete-javamail-resource` 子命令删除指定的JavaMail会话资源。请确保在执行此子命令之前删除对此资源的所有引用。

此命令仅在远程模式下支持。

#### 13.3.1 概要

```
delete-javamail-resource [--help]  
[--target target]  
jndi_name
```

#### 13.3.2 选项

```
--help  
-?
```

显示子命令的帮助文本。

```
--target
```

指定需要删除的JavaMail资源所在的实例，可选项server、domain、cluster-name、instance-name。

```
jndi_name
```

需要删除的JavaMail资源的JNDI名称。

#### 13.3.3 示例

示例1 删除JavaMail资源

```
asadmin> delete-javamail-resource mail/MyMailSession  
Command delete-javamail-resource executed successfully.
```

## 14 资源适配器配置

### 14.1 查看资源适配器配置

`list-resource-adapter-configs` 子命令查看资源适配器配置。

此命令仅在远程模式下支持。

#### 14.1.1 概要

```
list-resource-adapter-configs [--help] [--raname raname] [--verbose {false|true}]
```

#### 14.1.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--raname
```

指定连接器模块名称。

```
--verbose
```

是否需要列出已配置的属性。可设置true或false；默认为false。

#### 14.1.3 示例

示例1 查看资源适配器配置

```
asadmin> list-resource-adapter-configs
ra1
ra2
Command list-resource-adapter-configs executed successfully
```

### 14.2 新建资源适配器配置

`create-resource-adapter-config` 子命令为连接器模块创建配置信息。此子命令可以在部署资源适配器之前运行，以便在部署时可以获得配置信息。资源适配器配置也可以在部署资源适配器后创建。在这种情况下，资源适配器将使用新配置重新启动。您必须首先使用 `create-thread-pool` 子命令创建一个线程池，然后在 `--threadpoolid` 选项中将该线程池值标识为ID。

此命令仅在远程模式下支持。

#### 14.2.1 概要

```
create-resource-adapter-config [--help]
[--threadpoolid threadpool]
[--objecttype object-type]
[--property (property-name=value)[:name=value]*]
raname
```

#### 14.2.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--threadpoolid
```

工作管理器从中获取线程的线程池ID。此选项只需要一个线程池ID。

```
--objecttype
```

默认值为user。

```
--property
```

指定资源适配器Javabeans的附加配置属性的关键字-值对。关键字-值对由冒号(:)分隔。属性是 ra.xml 文件中的 resourceadapter-class 元素引用的类的setter方法的名称。

```
raname
```

指示连接器模块名称。它是domain.xml文件中资源适配器名称的值。

### 14.2.3 示例

示例1：新建资源适配器配置

此示例为新建名称为 ra1 的资源适配器配置。

```
asadmin> create-resource-adapter-config --property foo=bar --threadpoolid
mycustomerthreadpool ra1
Command create-resource-adapter-config executed successfully
```

## 14.3 删除资源适配器配置

delete-resource-adapter-config 子命令删除连接器模块的配置信息。

此命令仅在远程模式下支持。

### 14.3.1 概要

```
delete-resource-adapter-config [--help]
raname
```

### 14.3.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
raname
```

需要删除的资源适配器模块名称。

### 14.3.3 示例

示例1 删除资源适配器配置

此示例删除ra1的配置信息。

```
asadmin> delete-resource-adapter-config ra1
Command delete-resource-adapter-config executed successfully
```

## 15 上下文服务

### 15.1 查看上下文服务

`list-context-services` 子命令查看资源适配器配置。

此命令仅在远程模式下支持。

#### 15.1.1 概要

```
list-context-services [--help]
[target]
```

#### 15.1.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
target
```

指定目标实例进行查询。

#### 15.1.3 示例

示例1 查看上下文服务

```
asadmin> list-context-services
concurrent/ defaultContextService
concurrent/myContextService
Command list-context-services executed successfully.
```

### 15.2 新建上下文服务

`create-context-service` 子命令新建上下文服务资源。

此命令仅在远程模式下支持。

#### 15.2.1 概要

```
create-context-service [--help]
[--enabled={false| true}]
[--contextinfoenabled={false| true}]
[--contextinfo={ClassLoader|JNDI|Security|WorkArea}]]
[--description description]
[--property property]
[--target target]
jndi_name
```

#### 15.2.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--enabled
```

确定是否在运行时启用资源。默认值为true。

```
--contextinfoenabled
```

确定是否将容器上下文传播到线程。如果设置为true，则传播 `--contextinfo` 选项中指定的上下文。如果设置为false，则不会传播任何上下文，并且忽略 `--contextinfo` 选项。默认值为true。

```
--contextinfo
```

指定要传播到线程的单个容器上下文。有效值是 `ClassLoader`、`JNDI`、`Security` 和 `WorkArea`。值以逗号分隔的列表指定，不区分大小写。默认情况下，所有上下文都会传播。

```
--description
```

上下文服务的描述。

```
--property
```

用于配置资源的可选属性名称/值对。Apusic Server没有为此资源定义任何其他属性。此外，此资源当前不使用任何其他属性。

```
--target
```

指定上下文服务的目标，可选项有 `server`、`domain`、`cluster_name`、`instance_name`。

```
jndi_name
```

该资源的JNDI名称。

### 15.2.3 示例

示例1 新建上下文服务资源

此示例为新建JNDI名称为 `concurrent/myContextService` 的上下文服务资源。

```
asadmin> create-context-service concurrent/myContextService
Context service concurrent/myContextService created successfully. Command create-context-service executed successfully.
```

## 15.3 删除上下文服务

`delete-context-service` 子命令删除上下文服务资源信息。

此命令仅在远程模式下支持。

### 15.3.1 概要

```
delete-context-service [--help]
[--target target]
context-service-name
```

### 15.3.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--target
```

指定上下文服务所在的实例。

```
context-service-name
```

需要删除的上下文服务资源名称。

### 15.3.3 示例

示例1 删除上下文服务

```
asadmin> delete-context-service concurrent/myContextService  
Context service concurrent/myContextService deleted successfully.  
Command delete-context-service executed successfully.
```

## 16 托管线程工厂

### 16.1 查看托管线程工厂

`list-managed-thread-factories` 子命令查看托管线程工厂。

此命令仅在远程模式下支持。

#### 16.1.1 概要

```
list-managed-thread-factories [--help]
[target]
```

#### 16.1.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
target
```

指定目标实例进行查询。

#### 16.1.3 示例

示例1 查看托管线程工厂

```
asadmin> list-managed-thread-factories
concurrent/__defaultManagedThreadFactory
concurrent/myThreadFactory
Command list-managed-thread-factories executed successfully.
```

### 16.2 新建托管线程工厂

`create-managed-thread-factory` 子命令新建托管线程工厂资源。

此命令仅在远程模式下支持。

#### 16.2.1 概要

```
create-managed-thread-factory [--help]
[--enabled={false|true}]
[--contextinfoenabled={false|true}]
[--contextinfo={ClassLoader|JNDI|Security|WorkArea}]
[--threadpriority threadpriority]
[--description description]
[--property property]
[--target target]
jndi_name
```

#### 16.2.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--target
```

指定托管线程工厂的目标，可选项有 `server`、`domain`、`cluster_name`、`instance_name`。

```
--enabled
```

确定是否在运行时启用托管线程工厂。默认值为true。

```
--contextinfoenabled
```

确定是否将容器上下文传播到线程。如果设置为true，则传播 `--contextinfo` 选项中指定的上下文。如果设置为false，则不会传播任何上下文，并且忽略 `--contextinfo` 选项。默认值为true。

```
--contextinfo
```

指定要传播到线程的单个容器上下文。有效值是 `ClassLoader`、`JNDI`、`Security` 和 `WorkArea`。值以逗号分隔的列表指定，不区分大小写。默认情况下，所有上下文都会传播。

```
--threadpriority
```

指定要分配给已创建线程的优先级。默认值为5。

```
--description
```

托管线程工厂的描述。

```
--property
```

用于配置资源的可选属性名称/值对。Apusic Server没有为此资源定义任何其他属性。此外，此资源当前不使用任何其他属性。

```
jndi_name
```

该托管线程工厂的JNDI名称。

### 16.2.3 示例

示例1 新建托管线程工厂

此示例新建JNDI名称为 `concurrent/myThreadFactory` 的托管线程工厂。

```
asadmin> create-managed-thread-factory concurrent/myThreadFactory
Managed thread factory concurrent/myThreadFactory created successfully.
Command create-managed-thread-factory executed successfully.
```

## 16.3 删除托管线程工厂

`delete-managed-thread-factory` 子命令删除托管线程工厂资源信息。

此命令仅在远程模式下支持。

### 16.3.1 概要

```
delete-managed-thread-factory [--help]
[--target target]
managed_thread_factory_name
```

### 16.3.2 选项

```
--help  
-?
```

显示子命令的帮助文本。

```
--target
```

指定托管线程工厂所在的实例。

```
managed_thread_factory_name
```

需要删除的托管线程工厂资源名称。

### 16.3.3 示例

示例1 删除托管线程工厂资源

```
asadmin> delete-managed-thread-factory concurrent/myThreadFactory  
Managed thread factory concurrent/myThreadFactory deleted successfully.  
Command delete-managed-thread-factory executed successfully.
```

## 17 执行者服务

### 17.1 查看执行者服务资源

`list-managed-executor-service` 子命令查看执行者服务资源。

此命令仅在远程模式下支持。

#### 17.1.1 概要

```
list-managed-executor-services [--help]
[target]
```

#### 17.1.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
target
```

指定目标实例进行查询。

#### 17.1.3 示例

示例1 查看执行者服务资源

```
asadmin> list-managed-executor-services
concurrent/ defaultManagedExecutorService
concurrent/myExecutor
Command list-managed-executor-services executed successfully.
```

### 17.2 新建执行者服务资源

`create-managed-executor-service` 子命令新建执行者服务资源。

此命令仅在远程模式下支持。

#### 17.2.1 概要

```
create-managed-executor-service [--help]
[--enabled={false|true}]
[--contextinfoenabled={false|true}]
[--contextinfo={ClassLoader|JNDI|Security|WorkArea}]
[--threadpriority threadpriority]
[--longrunningtasks={false|true}]
[--hungafterseconds hungafterseconds]
[--corepoolsize corepoolsize]
[--maximumpoolsize maximumpoolsize]
[--keepalivesseconds keepalivesseconds]
[--threadlifetimesseconds threadlifetimesseconds]
[--taskqueuecapacity taskqueuecapacity]
[--description description]
[--property property]
[--target target]
jndi_name
```

## 17.2.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--target
```

指定该执行者服务所在的目标实例。

```
--enabled
```

确定是否在运行时启用执行者服务。默认值为true。

```
--contextinfoenabled
```

确定是否将容器上下文传播到线程。如果设置为true，则传播--contextinfo选项中指定的上下文。如果设置为false，则不会传播任何上下文，并且忽略--contextinfo选项。默认值为true。

```
--contextinfo
```

指定要传播到线程的单个容器上下文。有效值是ClassLoader、JNDI、Security和WorkArea。值以逗号分隔的列表指定，不区分大小写。默认情况下，所有上下文都会传播。

```
--threadpriority
```

指定要分配给已创建线程的优先级。默认值为5。

```
--longrunningtasks
```

指定资源是否应用于长时间运行的任务。默认值为false。如果设置为true，则长时间运行的任务不会被报告为卡住。

```
--hungafterseconds
```

指定任务在被视为无响应之前可以执行的秒数。默认值为0，这意味着任务永远不会被视为无响应。

```
--corepoolsize
```

指定线程池中要保留的线程数，即使它们处于空闲状态。默认值为0。当提交了一个新任务并且正在运行的线程数量小于corepoolsize大小时，会创建一个新线程来处理该请求。当corepoolsize大小的值为0（默认值）时，除非任务队列已满或资源正在使用直接排队，否则永远不会创建新线程。直接排队发生在taskqueuecapacity为0时，或者taskqueuecapacity为2147483647且corepoolsize为0时。。

```
--maximumpoolsize
```

指定线程池可以包含的最大线程数。默认值为2147483647，这意味着线程池基本上是无限制的，可以包含任意数量的线程。

```
--keepaliveseconds
```

指定当线程数大于核心池大小时，线程可以保持空闲的秒数。默认值为60。

```
--threadlifetimeseconds
```

指定线程在清除之前可以在线程池中保留的秒数，无论线程数是否大于核心池大小或线程是否处于空闲状态。默认值为0，这意味着线程永远不会被清除。

```
--taskqueuecapacity
```

指定可以存储在任务队列中等待执行的已提交任务的数量。默认值为2147483647，这意味着任务队列基本上是无限制的，可以存储任意数量的提交任务。

```
--description
```

该执行者服务的描述。

```
--property
```

用于配置资源的可选属性名称/值对。 {product---name}没有为此资源定义任何其他属性。此外，此资源当前不使用任何其他属性。

```
jndi_name
```

该执行者服务资源的JNDI名称。

### 17.2.3 示例

示例1新建执行者服务资源

此示例为新建JNDI名称为concurrent/myExecutor的执行者服务资源。

```
asadmin> create-managed-executor-service concurrent/myExecutor
Managed executor service concurrent/myExecutor created successfully. Command create-managed-executor-
service executed successfully.
```

## 17.3 删除执行者服务资源

delete-managed-thread-factory 子命令删除托管线程工厂资源信息。

此命令仅在远程模式下支持。

### 17.3.1 概要

```
delete-managed-executor-service [--help]
[--target target]
managed_executor_service_name
```

### 17.3.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--target
```

指定执行者服务资源所在的实例。

```
managed_executor_service_name
```

需要删除的执行者服务资源名称。

### 17.3.3 示例

示例1 删除执行者服务资源

```
asadmin> delete-managed-executor-service concurrent/myExecutor
Managed executor service concurrent/myExecutor deleted successfully.
Command delete-managed-executor-service executed successfully.
```

## 18 周期性执行者服务

### 18.1 查看周期性执行者服务资源

`list-managed-scheduled-executor-services` 子命令查看周期性执行者服务资源。

此命令仅在远程模式下支持。

#### 18.1.1 概要

```
list-managed-scheduled-executor-services [--help]
[target]
```

#### 18.1.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
target
```

指定目标实例进行查询。

#### 18.1.3 示例

示例1 查看周期性执行者服务资源

```
asadmin> list-managed-scheduled-executor-services
concurrent/ defaultManagedScheduledExecutorService
concurrent/myScheduledExecutor
Command list-managed-scheduled-executor-services executed successfully.
```

### 18.2 新建周期性执行者服务资源

`create-managed-scheduled-executor-service` 子命令新建周期性执行者服务资源。

此命令仅在远程模式下支持。

#### 18.2.1 概要

```
create-managed-scheduled-executor-service [--help]
[--enabled={false|true}]
[--contextinfoenabled={false|true}]
[--contextinfo={ClassLoader|JNDI|Security|WorkArea}]
[--threadpriority threadpriority]
[--longrunningtasks={false|true}]
[--hungafterseconds hungafterseconds]
[--corepoolsize corepoolsize]
[--keepaliveseconds keepaliveseconds]
[--threadlifetimeseconds threadlifetimeseconds]
[--description description]
[--property property]
[--target target]
jndi_name
```

#### 18.2.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--enabled
```

确定是否在运行时启用执行者服务。默认值为true。

```
--contextinfoenabled
```

确定是否将容器上下文传播到线程。如果设置为true，则传播--contextinfo选项中指定的上下文。如果设置为false，则不会传播任何上下文，并且忽略--contextinfo选项。默认值为true。

```
--contextinfo
```

指定要传播到线程的单个容器上下文。有效值是Classloader、JNDI、Security和WorkArea。值以逗号分隔的列表指定，不区分大小写。默认情况下，所有上下文都会传播。

```
--threadpriority
```

指定要分配给已创建线程的优先级。默认值为5。

```
--longrunningtasks
```

指定资源是否应用于长时间运行的任务。默认值为false。如果设置为true，则长时间运行的任务不会被报告为卡住。

```
--hungafterseconds
```

指定任务在被视为无响应之前可以执行的秒数。默认值为0，这意味着任务永远不会被视为无响应。

```
--corepoolsize
```

指定线程池中要保留的线程数，即使它们处于空闲状态。默认值为0。当提交了一个新任务并且正在运行的线程数量小于corepoolsize大小时，会创建一个新线程来处理该请求。当corepoolsize大小的值为0（默认值）时，除非任务队列已满或资源正在使用直接排队，否则永远不会创建新线程。直接排队发生在taskqueuecapacity为0时，或者taskqueuecapacity为2147483647且corepoolsize为0时。。

```
--maximumpoolsize
```

指定线程池可以包含的最大线程数。默认值为2147483647，这意味着线程池基本上是无限制的，可以包含任意数量的线程。

```
--keepaliveseconds
```

指定当线程数大于核心池大小时，线程可以保持空闲的秒数。默认值为60。

```
--threadlifetimeseconds
```

指定线程在清除之前可以在线程池中保留的秒数，无论线程数是否大于核心池大小或线程是否处于空闲状态。默认值为0，这意味着线程永远不会被清除。

```
--taskqueuecapacity
```

指定可以存储在任务队列中等待执行的已提交任务的数量。默认值为2147483647，这意味着任务队列基本上是无限制的，可以存储任意数量的提交任务。

```
--description
```

该执行者服务的描述。

```
--property
```

用于配置资源的可选属性名称/值对。{product--name}没有为此资源定义任何其他属性。此外，此资源当前不使用任何其他属性。

```
--target
```

指定该执行者服务所在的目标实例。

```
jndi_name
```

该周期性执行者服务资源的JNDI名称。

### 18.2.3 示例

示例1 新建周期执行者服务资源

```
asadmin> create-managed-scheduled-executor-service concurrent/myScheduledExecutor
Managed scheduled executor service concurrent/myScheduledExecutor created successfully.
Command create-managed-scheduled-executor-service executed successfully.
```

## 18.3 删除周期性执行者服务资源

`delete-managed-scheduled-executor-service` 子命令删除周期性托管线程工厂资源信息。

此命令仅在远程模式下支持。

### 18.3.1 概要

```
delete-managed-scheduled-executor-service [--help]
[--target target]
managed_scheduled_executor_service_name
```

### 18.3.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--target
```

指定周期性执行者服务资源所在的实例。

```
managed_scheduledexecutor_service_name
```

需要删除的周期性执行者服务资源名称。

### 18.3.3 示例

示例1 删除周期性执行者服务资源

```
asadmin> delete-managed-scheduled-executor-service concurrent/myScheduledExecutor
Managed scheduled executor service concurrent/myScheduledExecutor deleted successfully.
Command delete-managed-scheduled-executor-service executed successfully.
```

## 19 连接器连接池

### 19.1 查看连接器连接池

`list-connector-connection-pools` 子命令查看连接器连接池。

此命令仅在远程模式下支持。

#### 19.1.1 概要

```
list-connector-connection-pools [--help]
```

#### 19.1.2 选项

```
--help
-?
```

显示子命令的帮助文本。

#### 19.1.3 示例

示例1 查看连接器连接池

```
asadmin> list-connector-connection-pools
jms/myConnPool
Command list-connector-connection-pools executed successfully
```

## 19.2 新建连接器连接池

`create-connector-connection-pool` 子命令定义了到企业信息系统 (EIS) 的连接池。命名池可以被多个连接器资源引用。每个定义的池在服务器启动时实例化，并在首次访问时填充。如果两个或多个连接器资源指向同一个连接器连接池，则它们在运行时使用相同的连接池。单个资源适配器中的连接定义可以有多个池。可以通过使用 `--property` 选项指定用户、密码或其他连接信息，或者通过在XML描述符文件中指定连接信息来创建具有身份验证的连接器连接池。

此命令仅在远程模式下支持。

#### 19.2.1 概要

```
create-connector-connection-pool [--help] [--target=target]
--rename rename
--connectiondefinition connectiondefinitionname
[--steadypoolsize steadypoolsize]
[--maxpoolsize maxpoolsize]
[--maxwait maxwait]
[--poolresize poolresize]
[--idletimeout idletimeout]
[--isconnectvalidatereq={false|true}]
[--failconnection={false|true}]
  [--leaktimeout=timeout]
  [--leakreclaim={false|true}]
  [--creationretryattempts=attempts]
  [--creationretryinterval=interval]
  [--lazyconnectionenlistment={false|true}]
  [--lazyconnectionassociation={false|true}]
  [--associatewiththread={false|true}]
  [--matchconnections={true|false}]
  [--maxconnectionusagecount=count]
  [--validateatmostonceperiod=interval]
```

```
[--transactionsupport transactionsupport]
[--descrip[ti]on description]
[--ping {false|true}]
[--pooling {true|false}]
[--property (name=value)[:name=value]*]
poolname
```

### 19.2.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--target
```

指定连接器连接池所属目标实例。

```
--rname
```

连接器名称。

```
--associatewiththread
```

与线程关联，指定连接是否与线程相关联，以使线程能够重用该连接。如果连接未与线程关联，则线程每次需要连接时都必须从池中获取连接。可能的值如下：

- false  
连接与线程不关联（默认）。
- true  
连接与线程相关联。

```
--connectiondefinition
```

连接定义的名称。

```
--creationretryattempts
```

创建重试尝试次数，指定如果初始尝试失败，服务器重试创建连接的最大次数。

默认值为0，表示服务器不会重试创建连接。

```
--creationretryinterval
```

重试时间间隔，指定连续尝试创建连接之间的间隔（秒）。如果 `--creationretryattempts` 为0，则忽略 `--creationretryinterval` 选项。默认值为10。

```
--description
```

该连接器连接池的描述。

```
--failconnection
```

一旦失败，如果设置为true，则如果单个验证检查失败，池中的所有连接都将关闭。如果 `--isconnectvalidatereq` 选项设置为true，则此参数是必需的。默认值为false。

```
--idletimeout
```

空闲时间，连接在池中保持空闲的最长时间。经过这段时间后，池可以关闭此连接。默认值为300。

```
--isconnectvalidatereq
```

连接验证, 如果该值设置为true, 则在将连接提供给应用程序之前, 将检查连接是否可用。默认值为false。

```
--lazyconnectionenlistment
```

延迟连接登记, 指定是否仅在方法实际使用资源时才登记事务的资源。默认值为false。

```
--lazyconnectionassociation
```

延迟关联, 指定物理连接是否应仅在使用物理连接时与逻辑连接相关联, 并在事务完成时解除关联。这种关联和分离使得物理连接的重用成为可能。可能的值如下:

- false  
即使在使用物理连接之前, 物理连接也与逻辑连接相关联, 并且在事务完成时不会解除关联 (默认)。
- true  
物理连接仅在使用物理连接时与逻辑连接相关联, 并在事务完成时解除关联。 `--lazyconnectionenrollment` 选项也必须设置为true。

```
--leakreclaim
```

泄露回收, 指定泄露连接跟踪完成后是否将泄露连接恢复到连接池。可能的值如下:

- false  
泄露的连接不会恢复到连接池 (默认)。
- true  
泄露的连接将恢复到连接池。

```
--leaktimeout
```

泄露超时, 指定要跟踪连接池中的连接泄露的时间量 (秒)。如果启用了连接泄露跟踪, 则可以使用管理控制台启用对JDBC连接池的监视, 以获取连接泄露数量的统计信息。默认值为0, 这将禁用连接泄露跟踪。

```
--matchconnections
```

匹配连接, 指定从池中选择的连接是否应与资源适配器匹配。如果池中的所有连接都相同, 则不需要连接和资源适配器之间的匹配。可能的值如下:

- true  
连接应与资源适配器匹配 (默认)。
- false  
连接不应与资源适配器匹配

```
--maxconnectionusagecount
```

最大连接使用次数, 指定连接可以重复使用的最大次数。当达到此限制时, 连接将关闭。默认值为0, 表示对连接可重用的次数没有限制。

```
--maxpoolsize
```

最大池大小, 可以创建以满足客户端请求的最大连接数。默认值为32。

```
--maxwait
```

最长等待时间, 如果连接不可用, 则调用者在创建连接之前必须等待的时间量 (以毫秒为单位)。如果设置为0, 则调用方将被无限期阻止, 直到资源可用或发生错误为止。默认值为60000。

```
--ping
```

在创建（或重新配置）过程中，会联系将此属性设置为true的池，以识别其属性的任何错误值并发出警告。默认值为false。

```
--pooling
```

池化，当设置为false时，此属性将禁止池的连接共享。默认值为true。

```
--poolresize
```

池调整大小数量，池将按比例增加或减少连接数量的数量。扩展：当池没有空闲连接时，池将按此数量扩展。按比例缩小：删除所有无效和空闲的连接，有时会导致删除数量大于此值的连接。将确保由 `--steadypoolsize` 指定的连接数。可能的值从0到 `MAX_INTEGER`。默认值为2。

```
--property
```

用于配置池的可选属性名称/值对。

```
--steadypoolsize
```

池中维护的最小和初始连接数。默认值为8。

```
--transactionsupport
```

事务处理支持，指示此池将具有的事务支持级别。可能的值包括 `XATransaction`、`LocalTransaction` 和 `NoTransaction`。此属性的值可以小于或等于但不高于资源适配器的 `transactionsupport` 属性。资源适配器的事务支持属性具有值的顺序，其中 `XATransaction` 最高，`NoTransaction` 最低。

```
--validateatmostonceperiod
```

最多验证一次，指定连续请求之间最多验证一次连接的时间间隔（秒）。将此属性设置为适当的值可以最大限度地减少连接的验证请求数量。默认值为0，这意味着该属性未启用。

```
poolname
```

定义新建的连接器连接池名称。

### 19.2.3 示例

示例1 新建连接器连接池

此示例新建命名为 `jms/myConnPool` 的连接器连接池，资源适配器为 `jmsra`，连接定义的接口为 `javax.jms.QueueConnectionFactory`。

```
asadmin> create-connector-connection-pool --raname jmsra --connectiondefinition
javax.jms.QueueConnectionFactory --steadypoolsize 20 --maxpoolsize 100 --poolresize 2 --maxwait 60000
jms/myConnPool
Command create-connector-connection-pool executed successfully
```

## 19.3 删除连接器连接池

`delete-connector-connection-pool` 子命令删除连接器连接池。

此命令仅在远程模式下支持。

### 19.3.1 概要

```
delete-connector-connection-pool [--help]
[--target target]
[--cascade={false| true}] poolname
```

### 19.3.2 选项

```
--help  
-?
```

显示子命令的帮助文本。

```
--target
```

指定连接器连接池所在的实例。

```
--cascade
```

设置为true时，将删除与池关联的所有连接器资源以及池本身。当设置为false时，如果有任何资源与池相关联，则删除池失败。必须明确删除资源，或者必须将选项设置为true。默认值为false。

```
poolname
```

需要删除的连接器连接池名称。

### 19.3.3 示例

示例1 删除连接器连接池

```
asadmin> delete-connector-connection-pool  
--cascade=false jms/qConnPool  
Command delete-connector-connection-pool executed successfully
```

## 20 连接器资源

### 20.1 查看连接器资源

`list-connector-resources` 子命令查看连接器资源。

此命令仅在远程模式下支持。

#### 20.1.1 概要

```
list-connector-resources [--help]
[target]
```

#### 20.1.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
target
```

选择需要查询的目标实例。

#### 20.1.3 示例

示例1 查看连接器资源

```
asadmin> list-connector-resources
jms/myConnFactory
Command list-connector-resources executed successfully.
```

## 20.2 新建连接器资源

`create-connector-resource` 子命令新建连接器资源。

此命令仅在远程模式下支持。

#### 20.2.1 概要

```
create-connector-resource [--help]
--poolname connectorConnectionPoolName
[--enabled={true|false}]
[--description description]
[--objecttype ojecttype]
[--property (name=value)[:name=value]*]
[--target target]
jndi_name
```

#### 20.2.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--poolname
```

连接池的名称。当两个或多个资源元素指向同一个连接池元素时，它们在运行时使用相同的池连接。

```
--enabled
```

此选项决定是否在运行时启用资源。默认值为true。

```
--objecttype
```

定义连接器资源的类型。默认值为用户。允许的值为：

- system-all  
所有服务器实例和域管理服务器（DAS）的系统资源。
- system-admin  
仅用于DAS的系统资源。
- system-instance  
仅适用于所有服务器实例的系统资源。
- user  
资源用户名。

```
--description
```

该连接器资源的描述。

```
--property
```

该连接器资源的属性。

```
--target
```

该连接器资源所在的目录，可选项有server、domain、cluster\_name、instance\_name。

```
jndi_name
```

该连接器资源JNDI名称。

### 20.2.3 示例

示例1 新建连接器资源

```
asadmin> create-connector-resource --poolname jms/qConnPool --description "sample connector resource"
jms/qConnFactory
Command create-connector-resource executed successfully
```

## 20.3 删除连接器资源

`delete-connector-resource` 子命令删除连接器资源。

此命令仅在远程模式下支持。

### 20.3.1 概要

```
delete-connector-resource [--help] [--target target] jndi_name
```

### 20.3.2 选项

```
--help  
-?
```

显示子命令的帮助文本。

```
--target
```

指定连接器资源所在的实例。

```
jndi_name
```

需要删除的连接器资源名称。

### 20.3.3 示例

示例1 删除连接器资源

```
asadmin> delete-connector-resource jms/myConnFactory  
Command delete-connector-resource executed successfully
```

## 21 管理对象资源

### 21.1 查看管理对象资源

`list-admin-objects` 子命令查看管理对象资源。

此命令仅在远程模式下支持。

#### 21.1.1 概要

```
list-admin-objects [--help]
[target]
```

#### 21.1.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
target
```

选择需要查询的目标实例。

#### 21.1.3 示例

示例1 查看管理对象资源

```
asadmin> list-admin-objects
jms/samplequeue
Command list-admin-objects executed successfully
```

### 21.2 新建管理对象资源

`create-connector-resources` 子命令新建管理对象资源。

此命令仅在远程模式下支持。

#### 21.2.1 概要

```
create-admin-object [--help] [--target target]
--restype restype
[--classname classname]
--rename rename
[--enabled={true|false}]
[--description description]
[--property name=value[:name=value]*]
jndi_name
```

#### 21.2.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
target
```

该管理对象资源所在的目录，可选项有server、domain、cluster\_name、instance\_name。

```
--restype
```

指定管理对象的接口定义。资源类型必须是在资源适配器的thera.xml文件中指定的接口定义。

```
--classname
```

指定管理对象的类名。如果多个管理对象使用相同的接口定义，则需要此项。

```
--raname
```

指定与此受管对象关联的资源适配器的名称。

```
--enabled
```

指定是否启用此对象。默认值为true。

```
--description
```

描述管理对象的文本字符串。

```
--property
```

配置资源的名称/值对的描述。取决于资源适配器。

```
jndi_name
```

该管理对象资源的JNDI名称。

### 21.2.3 示例

示例1 新建管理对象资源

```
asadmin> create-admin-object --restype javax.jms.Queue --raname jmsra --description "sample administered
object" --property Name=sample_jmsqueue jms/samplequeue
Command create-admin-object executed successfully
```

## 21.3 删除管理对象资源

`delete-connector-resource` 子命令删除管理对象资源。

此命令仅在远程模式下支持。

### 21.3.1 概要

```
delete-admin-object [--help]
[--target target] jndi_name
```

### 21.3.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--target
```

指定管理对象资源所在的实例。

```
jndi_name
```

需要删除的管理对象资源名称。

### 21.3.3 示例

示例1 删除管理对象资源

```
asadmin> delete-admin-object jms/samplequeue  
Command delete-admin-object executed successfully
```

## 22 安全映射

### 22.1 查看安全映射

`list-connector-work-security-maps` 子命令查看工作安全映射。

此命令仅在远程模式下支持。

#### 22.1.1 概要

```
list-connector-work-security-maps [--help]
[--securitymap securitymap]
resource_adapter_name
```

#### 22.1.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--securitymap
```

指定资源适配器中包含的安全映射的名称，应从中列出标识和主体。

```
resource_adapter_name
```

要列出其安全映射的资源适配器的名称。

#### 22.1.3 示例

示例1 查看安全映射的资源适配器

```
asadmin> list-connector-work-security-maps my_resource_adapter
workSecurityMap1: EIS principal=eis-principal-2, mapped
principal=server-principal-2
workSecurityMap1: EIS principal=eis-principal-1, mapped
principal=server-principal-1
workSecurityMap2: EIS principal=eis-principal-2, mapped
principal=server-principal-2
workSecurityMap2: EIS principal=eis-principal-1, mapped
principal=server-principal-1
Command list-connector-work-security-maps executed successfully.
```

### 22.2 新建安全映射

`create-connector-work-security-map` 子命令将资源适配器EIS主体或EIS用户组提交的工作的调用者标识映射到{product---name}安全域中的合适主体或用户组。一个或多个工作安全映射可能与资源适配器相关联。连接器工作安全映射配置支持使用通配符星号 (\*) 来指示所有用户或所有用户组。企业信息系统 (EIS) 是保存组织数据的任何系统。它可以是大型机、消息传递系统、数据库系统或应用程序。

此命令仅在远程模式下支持。

#### 22.2.1 概要

```
create-connector-work-security-map [--help]
--rename raname
[--principalsmap eis-principal1=principal_name1[, eis-principal2=principal_name2]*
| --groupsmap eis-group1=server-group1[, eis-group2=server-group2]*]
```

```
[--description description]
mapname
```

### 22.2.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--raname
```

指示连接器模块名称，即资源适配器的名称。

```
--description
```

该工作安全映射的相关描述。

```
--groupsmap
```

指定后端EIS用户组到 {product---name} 用户组的映射。使用逗号分隔的列表指定多个映射。使用 `--principalsmap` 选项或 `--groupsmap` 选项，但不要同时使用。

```
--principalsmap
```

指定后端EIS主体到{product---name}主体的映射。使用逗号分隔的列表指定多个映射。使用 `--principalsmap` 选项或 `--groupsmap` 选项，但不要同时使用。

```
mapname
```

定义新建的安全映射的名称。

### 22.2.3 示例

示例1 新建安全映射

```
asadmin> create-connector-work-security-map --raname my-resource-adapter --principalsmap eis-principal-1=server-principal-1,eis-principal-2=server-principal-2,eis-principal-3=server-principal-1
workSecurityMap1
Command create-connector-work-security-map executed successfully.
```

## 22.3 删除安全映射

`deleteconnector-work-security-map` 子命令删除与指定资源适配器关联的安全映射。要使此子命令成功，您必须首先创建并部署指定的资源适配器。企业信息系统 (EIS) 是保存组织数据的任何系统。它可以是大型机、消息传递系统、数据库系统或应用程序。

此命令仅在远程模式下支持。

### 22.3.1 概要

```
delete-connector-work-security-map [--help]
--raname raname
mapname
```

### 22.3.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--raname
```

指定与工作安全映射关联的连接器模块名称。

```
mapname
```

需要删除的工作安全映射名称。

### 22.3.3 示例

示例1 删除工作安全映射

```
asadmin> delete-connector-work-security-map --raname ra_name work_security_map_name  
Command delete-connector-work-security-map executed successfully.
```

## 23 OSGI Bundle储存库

### 23.1 上传OSGI Bundle资源

`create-osgi-bundle-module` 子命令上传OSGI Bundle资源。

此命令仅在远程模式下支持。

#### 23.1.1 概要

```
create-osgi-bundle-module --filepath=filepath
```

#### 23.1.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--filepath
```

选择需要上传的jar文件。

#### 23.1.3 示例

示例1 上传OSGI Bundle资源

```
asadmin> create-osgi-bundle-module --
filepath="/opt/testz/demo/com.ibm.samples.websphere.osgi.colors.blender_1.0.1.jar"
Command create-osgi-bundle-module executed successfully.
```

## 23.2 删除OSGI Bundle资源

`delete-osgi-bundle-module` 子命令删除OSGI Bundle资源。

此命令仅在远程模式下支持。

#### 23.2.1 概要

```
delete-osgi-bundle-module [--bundlenames=bundlenames]
[--deleteall=false]
```

#### 23.2.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--deleteall
```

设置为true时，将会删除所有 `--bundlenames` 相同名称的Bundle资源。默认为false。

```
--bundlenames
```

需要删除的OSGI Bundle资源名称。

#### 23.2.3 示例

## 示例1 删除OSGI Bundle资源

此示例删除名称为 `com.ibm.samples.websphere.osgi.colors.blender_1.0.1.jar` 的bundle资源。

```
asadmin> delete-osgi-bundle-module --bundlenames=com.ibm.samples.websphere.osgi.colors.blender_1.0.1.jar  
Command delete-osgi-bundle-module executed successfully.
```

## 示例2 统一删除相同名称的bundle资源

此示例统一删除名称有 `com.ibm.samples` 的bundle资源。

```
asadmin> delete-osgi-bundle-module --bundlenames=com.ibm.samples --deleteall=true  
Command delete-osgi-bundle-module executed successfully.
```

## 24 线程池管理

### 24.1 查看线程池

`list-threadpools` 子命令查看线程池信息。

此命令仅在远程模式下支持。

#### 24.1.1 概要

```
list-threadpools [--help] target
```

#### 24.1.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
target
```

需要查询的目标实例。

#### 24.1.3 示例

示例1 查看线程池

此示例查询实例server中所有的线程池。

```
asadmin> list-threadpools server
admin-thread-pool
http-thread-pool
thread-pool-1
Command list-threadpools executed successfully.
```

### 24.2 新建线程池

`create-threadpool` 子命令创建具有指定名称的线程池。您可以指定池中线程的最大和最小数量、消息数量和线程的空闲超时。创建的线程池可用于为IIOP请求提供服务，也可用于资源适配器为工作管理请求提供服务。一个线程池可以在多个资源适配器中使用。

此命令仅在远程模式下支持。

#### 24.2.1 概要

```
create-threadpool [--help] [--target target]
[--maxthreadpoolsize maxthreadpoolsize]
[--minthreadpoolsize minthreadpoolsize]
[--idletimeout idletimeout] [--maxqueue size maxqueue size]
[--workqueues workqueues] threadpool-id
```

#### 24.2.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--target
```

指定线程池所属的目标实例，可选项有server、configuration-name、cluster-name、instance-name。

```
--maxthreadpoolsize
```

指定最大线程池大小。

```
--minthreadpoolsize
```

指定最小线程池大小。

```
--idletimeout
```

指定空闲超时时间，单位为秒。

```
--workqueues
```

指定最大队列大小。

```
threadpool-id
```

定义线程池名称。

### 24.2.3 示例

示例1 新建线程池

```
asadmin> create-threadpool --maxthreadpoolsize 100 --minthreadpoolsize 20 --idletimeout 2 threadpool-1
Command create-threadpool executed successfully
```

## 24.3 删除线程池

delete-threadpool 子命令删除线程池信息。

此命令仅在远程模式下支持。

### 24.3.1 概要

```
delete-threadpool [--help]
[--target target] threadpool-id
```

### 24.3.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--target
```

指定线程池所在的目标。

```
threadpool-id
```

需要删除的线程池名称。

### 24.3.3 示例

示例1 删除线程池

```
asadmin> delete-threadpool threadpool-1
```

```
Command delete-threadpool executed successfully.
```

## 25 协议配置

### 25.1 查看协议信息

`list-protocols` 子命令查看协议信息。

此命令仅在远程模式下支持。

#### 25.1.1 概要

```
list-protocol [--help] target
```

#### 25.1.2 选项

```
--help  
-?
```

显示子命令的帮助文本。

```
target
```

需要查询的目标实例。

#### 25.1.3 示例

示例1 查看协议

```
asadmin> list-protocols  
admin-listener  
http-listener-1  
http-listener-2  
Command list-protocols executed successfully.
```

### 25.2 新建协议信息

`create-protocol` 子命令新建协议信息。

此命令仅在远程模式下支持。

#### 25.2.1 概要

```
create-protocol [--help]  
[--securityenabled={false|true}]  
[--target target]  
protocol-name
```

#### 25.2.2 选项

```
--help  
-?
```

显示子命令的帮助文本。

```
--securityenabled
```

是否启用安全性，设置为true时，HTTP。默认为false。

```
--target
```

指定协议所属的目标实例。

```
protocol-name
```

定义协议名称。

### 25.2.3 示例

示例1 新建协议

此示例新建名称为myprotocol的协议。

```
asadmin> create-protocol myprotocal
Command create-protocol executed successfully.
```

## 25.3 删除协议

`delete-protocol` 子命令删除协议信息。

此命令仅在远程模式下支持。

### 25.3.1 概要

```
delete-protocol [--help]
[--target target]
protocol-name
```

### 25.3.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--target
```

协议所属的目标。

```
protocol-id
```

需要删除的协议名称。

### 25.3.3 示例

示例1 删除协议

```
asadmin> delete-protocol http-1
Command delete-protocol executed successfully.
```

## 26 网络监听程序

### 26.1 查看网络监听程序

`list-network-listeners` 子命令查看网络监听程序信息。

此命令仅在远程模式下支持。

#### 26.1.1 概要

```
list-network-listeners [--help] target
```

#### 26.1.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
target
```

需要查询的目标实例。

#### 26.1.3 示例

示例1 查看网络监听程序

```
asadmin> list-network-listeners
admin-listener
http-listener-1
https-listener-2
Command list-network-listeners executed successfully.
```

### 26.2 新建网络监听程序

`create-network-listener` 子命令新建网络监听程序。

您可以使用 `create-http-listener` 子命令创建使用HTTP协议的网络侦听器，而无需首先创建协议、传输或http配置。此子命令是一个方便的快捷方式，但它只能访问有限数量的选项。

此命令仅在远程模式下支持。

#### 26.2.1 概要

```
create-network-listener [--help]
[--address address]
--listenerport listener-port
[--threadpool thread-pool]
--protocol protocol
[--transport transport]
[--enabled={true| false}]
[--jkenabled={false| true}]
[--target target]
listener-name
```

#### 26.2.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--address
```

设置监听地址，可通过DNS解析。

```
--listenerport
```

用于创建监听套接字的端口号。合法值为1-65535。

```
--threadpool
```

此监听器的线程池名称。指定线程池是可选的。默认设置为 `http-thread-pool`。

```
--protocol
```

该监听器的协议名称。

```
--transport
```

此监听器的传输名称。指定传输是可选的。默认值为tcp。

```
--enabled
```

该监听器的状态，为true时表示启用。

```
--jkenabled
```

如果设置为true，启用mod\_jk。默认为false。

```
--target
```

指定该网络监听程序的目标实例。

```
listener-name
```

定义该监听程序的名称。

### 26.2.3 示例

示例1 新建网络监听程序

此示例新建名为sampleListener的网络监听程序，监听端口为7270，协议为http-1。

```
asadmin> create-network-listener --listenerport 7270 --protocol http-1 --enabled=false sampleListener
Command create-network-listener executed successfully.
```

## 26.3 删除网络监听程序

`delete-network-listener` 子命令删除网络监听程序信息。

此命令仅在远程模式下支持。

### 26.3.1 概要

```
delete-network-listener [--help]
[--target target]
```

```
listener-name
```

### 26.3.2 选项

```
--help  
-?
```

显示子命令的帮助文本。

```
--target
```

协议所属的目标。

```
listener-id
```

需要删除的网络监听程序名称。

### 26.3.3 示例

示例1 删除网络监听程序

```
asadmin> delete-network-listener sampleListener  
Command delete-network-listener executed successfully.
```

## 27 HTTP参数

### 27.1 新建HTTP参数

`create-http` 子命令为协议创建一组http参数，进而配置一个或多个网络侦听器。

此命令仅在远程模式下支持。

#### 27.1.1 概要

```
create-http [--help]
--default-virtual-server virtual-server
[--request-timeout-seconds timeout]
[--timeout-seconds timeout]
[--max-connection max-keepalive]
[--dns-lookup-enabled={false| true}]
[--servername server-name]
[--target target]
protocol-name
```

#### 27.1.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--default-virtual-server
```

关联网络监听器的默认虚拟服务器的ID属性。

```
--request-timeout-seconds
```

请求超时的时间（秒）。如果不设置此选项，请求将在30秒后超时。

```
--timeout-seconds
```

保持活动连接的最长时间（秒），可以将连接视为空闲并保持为保持活动状态的最长时间。值为0或更小意味着保持活动连接将无限期保持打开状态。默认值为30。

```
--max-connection
```

在服务器关闭连接之前，可以管道化的HTTP请求的最大数量。将此属性设置为1可以禁用HTTP/1.0的持久连接，以及HTTP/1.1的持久连接和管道化。默认值是256。

```
--dns-lookup-enabled
```

如果设置为true，则查找客户端的DNS条目。默认值为false。

```
--servername
```

服务器名。告诉服务器在发送给客户端的任何URL的主机名部分中应该填写什么。这会影响到服务器自动生成的URL；但它不会影响服务器上存储的目录和文件的URL。如果你的服务器使用了别名，那么这个名称应该是别名。如果后面附加了冒号和端口号，服务器发送给客户端的URL中将使用这个端口号。

```
--target
```

指定所属的目标实例。

```
protocol-name
```

此HTTP参数集所应用的协议的名称。

### 27.1.3 示例

示例1 设置HTTP参数

以下命令为名为HTTP-1的协议创建HTTP参数集：

```
asadmin> create-http --timeout-seconds 60 --default-virtual-server server http-1  
Command create-http executed successfully.
```

## 28 传输

### 28.1 查看传输

`list-transport` 子命令查看传输信息。

此命令仅在远程模式下支持。

#### 28.1.1 概要

```
list-transport [--help] target
```

#### 28.1.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
target
```

需要查询的目标实例。

#### 28.1.3 示例

示例1 查看传输

```
asadmin> list-transport
http1-trans
tcp
Command list-transport executed successfully.
```

## 28.2 新建传输

`create-transport` 子命令新建传输配置。

此命令仅在远程模式下支持。

#### 28.2.1 概要

```
create-transport [--help]
  [--acceptorthreads acceptor-threads]
  [--buffersizebytes buffer-size]
  [--bytebuffertype byte-buffer-type]
  [--classname class-name]
  [--displayconfiguration={false|true}]
  [--enablesnoop={false|true}]
  [--idlekeytimeoutseconds idle-key-timeout]
  [--maxconnectionscount max-connections]
  [--readtimeoutmillis read-timeout]
  [--writetimeoutmillis write-timeout]
  [--selectionkeyhandler selection-key-handler]
  [--selectorpolltimeoutmillis selector-poll-timeout]
  [--tcpnodelay={false|true}]
  [--target target]
transport-name
```

### 28.2.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--acceptorthreads
```

传输的接收器线程数。建议值是机器中的处理器数量。默认值为1。

```
--buffersizebytes
```

缓冲器大小。为引用此传输的网络侦听器创建的输入流提供的缓冲区的大小（以字节为单位）。默认值为8192。

```
--bytebuffertype
```

字节缓冲器类型。为网络侦听器创建的输入流提供的缓冲区类型。允许的值是HEAP和DIRECT。默认值为HEAP。

```
--classname
```

实现传输的Java类的完全限定名。默认设置为 `org.Apusic.grizzly.TCPSelectorHandler`。

```
--displayconfiguration
```

显示配置。如果为true，则将内部网络配置刷新到服务器日志中。可用于调试，但会降低性能。默认值为false。

```
--enablesnoop
```

启动探测。如果为true，则将请求/响应信息写入服务器日志。可用于调试，但会降低性能。默认值为false。

```
--idlekeytimeoutseconds
```

空闲键超时。取消空闲键并关闭通道之前的时间。默认值为30秒。

```
--maxconnectionscount
```

最大连接数。引用此传输的网络侦听器的最大连接数。值-1表示没有限制。默认值为4096。

```
--readtimeoutmillis
```

读取超时。服务器在头和正文解析阶段等待的时间。默认值为30000毫秒或30秒。

```
--writetimeoutillis
```

写入超时。在写入响应时，服务器在考虑远程客户端已断开连接之前等待的时间量。默认值为30000毫秒或30秒。

```
--tcpnodelay
```

如果为true，则默认启用TCP\_NODELAY（也称为Nagle算法）。默认值为false。

```
--target
```

指定的目标实例，可选项有server、configuration-name、cluster-name、standalone-instance-name。

```
transport-name
```

定义的传输名称。

### 28.2.3 示例

示例1 新建传输

```
asadmin> create-transport --acceptorthreads 100 http1-trans  
Command create-transport executed successfully.
```

## 28.3 删除传输

`delete-transport` 子命令删除传输。

此命令仅在远程模式下支持。

### 28.3.1 概要

```
delete-transport [--help]  
[--target target]  
transport-name
```

### 28.3.2 选项

```
--help  
-?
```

显示子命令的帮助文本。

```
--target
```

所属的目标。

```
transport-name
```

需要删除的传输名称。

### 28.3.3 示例

示例1 删除传输

```
asadmin> delete-transport http1-trans  
Command delete-transport executed successfully.
```

## 29 虚拟服务器

### 29.1 查看虚拟服务器

`list-virtual-servers` 子命令查看虚拟服务器信息。

此命令仅在远程模式下支持。

#### 29.1.1 概要

```
list-virtual-servers [--help]
[target]
```

#### 29.1.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
target
```

需要查询的目标实例。

#### 29.1.3 示例

示例1 查看虚拟服务器

```
asadmin> list-virtual-servers
server on
asadmin on
Command list-virtual-servers executed successfully.
```

### 29.2 新建虚拟服务器

`create-virtual-server` 子命令用于创建指定的虚拟服务器。 {product---name} 中的虚拟化允许多个URL域由监听多个主机地址的单个HTTP服务器进程提供服务。如果应用程序在两个虚拟服务器上可用，它们仍然共享相同的物理资源池。。

此命令仅在远程模式下支持。

#### 29.2.1 概要

```
create-virtual-server [--help]
--hosts hosts
[--httplisteners http-listeners]
[--networklisteners network-listeners]
[--defaultwebmodule default-web-module]
[--state={on|off}]
[--logfile log-file]
[--property (name=value)[:name=value]*]
[--target target]
virtual-server-id
```

#### 29.2.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--host
```

主机请求标头中允许的逗号分隔的 ( , ) 值列表，用于选择当前虚拟服务器。配置为同一连接组的每个虚拟服务器都必须具有该组的唯一主机。

```
--httplisteners
```

以逗号分隔的 ( , ) HTTP侦听器ID列表。仅对非默认虚拟服务器的虚拟服务器是必需的。HTTP侦听器转换为网络侦听器。此选项已弃用，但为了向后兼容性而保留。请改用 `--networklisteners`。如果使用 `--networklisteners`，则忽略此选项。

```
--networklisteners
```

网络监听程序。以逗号分隔的 ( , ) 网络侦听器ID列表。仅对非默认虚拟服务器的虚拟服务器是必需的。

```
--defaultwebmodule
```

默认情况下，与此虚拟服务器关联的独立web模块。

```
--state
```

确定虚拟服务器是处于活动状态（打开）还是非活动状态（关闭或禁用）。默认设置为启用。当处于非活动状态时，虚拟服务器不会为请求提供服务。

```
--logfile
```

要写入此虚拟服务器日志条目的文件的名称。默认情况下，这是服务器日志。保存访问日志的文件和目录必须由运行服务器的用户帐户写入。

```
--property
```

用于配置虚拟服务器的可选属性名称/值对。以下属性可用：

- sso-max-inactive-seconds

指定在没有收到客户端活动的情况下，用户的单点登录记录有资格被清除的秒数。由于单点登录适用于同一虚拟服务器上的多个应用程序，因此对任何应用程序的访问都会使单点登录记录保持活动状态。默认值为300秒（5分钟）。更高的值为用户提供了更长的单点登录持久性，但代价是服务器上更多的内存使用。

- sso-reap-interval-seconds

指定清除过期单点登录记录之间的秒数。默认值为60。

- setCacheControl

指定以逗号分隔的缓存控制响应指令列表。

- allowLinking

如果此属性的值为true，则将为部署在此虚拟服务器上的所有web应用程序提供符号链接资源。默认为true。

- accessLogWriteInterval

指示日志写入磁盘之前的秒数。当缓冲区已满或间隔到期时，会写入访问日志。如果该值为0（零），则即使缓冲区未满，也始终会写入缓冲区。这意味着每次访问服务器时，日志消息都会直接存储到文件中。

- accessLogBufferSize

指定存储访问日志调用的缓冲区的大小（以字节为单位）。

- allowRemoteAddress

- 访问地址白名单。这是一个逗号分隔的正则表达式模式列表，与远程客户端的IP地址进行比较。如果指定了此属性，则远程地址必须匹配才能接受此请求。如果未指定此属性，则将接受所有请求，除非远程地址与 `denyRemoteAddress` 模式匹配。此属性的默认值为null。

- denyRemoteAddress

访问地址黑名单。这是一个逗号分隔的正则表达式模式列表，与远程客户端的IP地址进行比较。如果指定了此属性，则远程地址必须不匹配才能接受此请求。如果未指定此属性，则请求接受仅由 `allowRemoteAddress` 属性控制。此属性的默认值为null。

- `allowRemoteHost` 访问域名白名单。这是一个逗号分隔的正则表达式模式列表，与远程客户端的主机名（由 `java.net.Socket.getInetAddress().getHostName()` 返回）进行比较。如果指定了此属性，则远程主机名必须匹配才能接受此请求。如果未指定此属性，则将接受所有请求，除非远程主机名与 `denyRemoteHost` 模式匹配。此属性的默认值为 `null`。
- `denyRemoteHost` 访问域名黑名单。这是一个逗号分隔的正则表达式模式列表，与远程客户端的主机名（由 `java.net.Socket.getInetAddress().getHostName()` 返回）进行比较。如果指定了此属性，则远程主机名必须不匹配才能接受此请求。如果未指定此属性，则请求接受仅由 `allowRemoteHost` 属性控制。此属性的默认值为 `null`。
- `authRealm` 指定身份验证领域的名称属性，该属性覆盖部署到此虚拟服务器的独立web应用程序的服务器实例的默认领域。在独立Web应用程序的 `web.xml` 文件中定义的领域会覆盖虚拟服务器的领域。
- `securePagesWithPragma` 将此属性设置为 `false`，以确保对于此虚拟服务器上的所有web应用程序，使用SSL下载的文件在Internet Explorer中正常工作。可以为特定的web应用程序设置此属性。
- `contextXmlDefault` 指定此虚拟服务器的 `context.xml` 文件相对于域目录的位置（如果使用）。
- `alternatedocroot_n` 指定一个替代的文档根目录（`docroot`），其中 `n` 是一个正整数，允许指定多个。替代的 `docroot` 允许Web应用程序根据请求是否与其替代 `docroot` 的URI模式之一（或多个）匹配，而从其自身的 `docroot` 外部为某些资源提供服务。如果请求与替代 `docroot` 的URI模式匹配，则通过将请求URI（去掉Web应用程序的上下文根）附加到替代 `docroot` 的物理位置（目录），将其映射到替代 `docroot`。如果请求与多个URI模式匹配，则根据以下优先级顺序确定替代 `docroot`：
  - 完全匹配
  - 最长路径匹配
  - 扩展匹配

例如，以下属性指定了三个备选的文档根。第一个备选 `docroot` 的URI模式使用完全匹配，而第二个和第三个备选 `docroots` 的URI模式分别使用扩展和最长路径前缀匹配。

```
<property name="alternatedocroot_1"
value="from=/my.jpg dir=/srv/images/jpg"/>
<property name="alternatedocroot_2"
value="from=*.jpg dir=/srv/images/jpg"/>
<property name="alternatedocroot_3"
value="from=/jpg/* dir=/src/images"/>
```

每个备用 `docroot` 的值都有两个组件：第一个组件 `from` 指定备用 `docroot` 的URI模式，第二个组件 `dir` 指定备用 `docroot` 的物理位置（目录）。`dir` 组件中允许有空格。可以为特定的web应用程序设置此属性。

- `send-error_n` 指定虚拟服务器的自定义错误页面映射，这些映射由部署在虚拟服务器上的所有web应用程序继承。web应用程序可以在其 `web.xml` 部署描述符中覆盖这些自定义错误页面映射。每个 `send-error_n` 属性的值都有三个组件，可以按任何顺序指定：第一个组件 `code` 指定了三位HTTP响应状态代码，应在响应中返回自定义错误页面。第二个组件 `path` 指定自定义错误页面的绝对或相对文件系统路径。相对文件系统路径被解释为相对于 `domain-dir/config` 目录。第三个组件 `reason` 是可选的，指定要返回的原因字符串的文本（如 `Unauthorized` 或 `Forbidden`）。例如：

```
<property name="send-error_1"
value="code=401 path=/myhost/401.html reason=MY-401-REASON"/>
```

此示例属性定义导致 `/myhost/401.html` 的内容返回401响应，以及以下响应行：`HTTP/1.1 401 MY-401-REASON`

- `redirect_n` 指定将对旧URL的请求视为对新URL的请求。这些属性由部署在虚拟服务器上的所有web应用程序继承。每个 `redirect_n` 属性的值都有两个组件，可以按任何顺序指定：第一个组件 `from` 指定要匹配的请求URI的前缀。第二个组件 `url-prefix` 指定要返回给客户端的新URL前缀。`from` 前缀仅被此URL前缀替换。例如：

```
<property name="redirect_1"
value="from=/dummy url-prefix=http://etude"/>
```

- value\_n

指定自定义阀的完全限定类名，其中 n 为正整数，允许指定多个。

- listener\_n

指定自定义Catalina侦听器的完全限定类名，其中n是一个正整数，允许指定多个。

- errorReportValue

指定自定义阀的完全限定类名，该自定义阀为此虚拟服务器上的应用程序生成默认错误页。指定一个空字符串以禁用此虚拟服务器的默认错误页机制。

```
--target
```

指定所属的目标实例。

```
virtual-server-id
```

标识要创建的虚拟服务器的唯一ID。此ID不能以数字开头。

### 29.2.3 示例

示例1 新建虚拟服务器

此示例新建命名为 testserver 的虚拟服务器，其中主机为 m.testhost.com，网络监听程序为 http-listener-1，访问地址黑名单为 172.20.140.14，访问地址白名单为 172.20.140.15 和 172.20.140.16。

```
asadmin> create-virtual-server --hosts m.testhost.com --networklisteners http-listener-1 --property
denyRemoteAddress=172.20.140.14:allowRemoteAddress=172.20.140.15,172.20.140.16 testserver
Command create-virtual-server executed successfully.
```

## 29.3 删除虚拟服务器

delete-virtual-server 子命令删除虚拟服务器。

此命令仅在远程模式下支持。

### 29.3.1 概要

```
delete-virtual-server [--help]
[--target target] virtual-server-id
```

### 29.3.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--target
```

所属的目标。

```
virtual-server-id
```

需要删除的虚拟服务器名称。

### 29.3.3 示例

## 示例1 删除虚拟服务器

```
asadmin> delete-virtual-server sample_vs1  
Command delete-virtual-server executed successfully.
```

## 30 日志配置

### 30.1 设置日志属性

`set-log-attributes` 子命令为一个或多个记录器设置日志属性。您可以设置的属性对应于域的logging.properties文件中可用的属性。根据您的属性，可能需要重新启动服务器。

此命令仅在远程模式下支持。

#### 30.1.1 概要

```
set-log-attributes [--help]
[--target=target]
attribute-name=attribute-value[:attribute-name=attribute-value]*
```

#### 30.1.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--target
```

指定目标。server、instance、cluster。如果省略此选项，将为默认server设置属性。

```
attribute-name
```

日志记录属性的全范围名称。 `list-log-attributes` 子命令可用于列出当前定义的所有属性的名称。

```
attribute-value
```

应用于指定属性的值。可以指定的属性值包括以下内容：

配置项	说明	默认值
配置名称	所属的配置文件	
写入系统日志	com.apusic.enterprise.server.logging.SyslogHandler.useSystemLogging=false; 如果选中此选项，则除了服务器日志以外，系统还会将日志记录输出发送到 syslog 工具。此功能仅在 Solaris 和 Linux 系统上可用	不启用
记录到控制台	com.apusic.enterprise.server.logging.ApusicFileHandler.logtoConsole=false; 将日志消息写入到系统控制台	不启用
按日生成日志文件	com.apusic.enterprise.server.logging.ApusicFileHandler.rotationOnDateChange=false; 每天凌晨生成服务器日志文件	不启用
多行模式	com.apusic.enterprise.server.logging.ApusicFileHandler.multiLineMode=true; 在消息头之后的新行上开始日志消息体	启用
控制台日志格式	java.util.logging.ConsoleHandler.formatter=com.apusic.enterprise.server.logging.UniformLogFormatter; 控制台日志格式，包含 ULF 和 ODL	ULF
日志文件格式	com.apusic.enterprise.server.logging.ApusicFileHandler.formatter=com.apusic.enterprise.server.logging.ODLLogFormatter; 日志文件格式，包含 ULF 和 ODL	ODL
排除的字段	com.apusic.enterprise.server.logging.ApusicFileHandler.excludeFields; 日志文件排除选中的字段，可选项为线程 ID、用户 ID、执行上下文 ID、毫秒时间戳、等级值、日期格式化、日志记录器名称、消息 ID	
文件滚动限制	com.apusic.enterprise.server.logging.ApusicFileHandler.rotationLimitInBytes=2000000; 在滚动之前，服务器日志文件必须达到的大小(字节)	2MB

文件滚动时间限制	com.apusic.enterprise.server.logging.ApusicFileHandler.rotationTimelimitInMinutes=0; 服务器日志必须滚动之前的分钟数, 如果设置为 0, 将根据在日志滚动限制中指定的大小滚动文件	0
刷新频率	com.apusic.enterprise.server.logging.ApusicFileHandler.flushFrequency=100; 一次从队列写入到服务器日志的最大消息数	100
最大历史文件数	com.apusic.enterprise.server.logging.ApusicFileHandler.maxHistoryFiles=0; 要保留的最大日志文件数。输入 0 保留所有的日志文件	0
日志压缩时间	com.apusic.enterprise.server.logging.ApusicFileHandler.compressTime; 每天进行日志压缩的时间点	00:03
最大压缩文件数	com.apusic.enterprise.server.logging.ApusicFileHandler.compressMaxRemain; 要保留的访问日志压缩文件的最大数目, 输入0保留所有的压缩日志文件	10
日志压缩路径	com.apusic.enterprise.server.logging.ApusicFileHandler.compressPath; 指定每天进行日志压缩生成的压缩文件路径	\${com.apusic.aas.instanceRoot}/logs
日志文件	com.apusic.enterprise.server.logging.ApusicFileHandler.file=\${com.apusic.aas.instanceRoot}/logs/server.log; 服务器日志文件的替代名称或位置 注意, 如果修改日志文件, 需要同时修改【虚拟服务器】中的日志文件路径为同一个	\${com.apusic.aas.instanceRoot}/logs/server.log
日志处理程序	handlers; 定制日志处理程序的绝对类名	com.apusic.aas.logging.ApusicConsoleHandler
推送到Kafka	com.apusic.enterprise.server.logging.ApusicKafkaHandler.enable=false; 是否开启推送到Kafka功能	不启用
Kafka日志格式	com.apusic.enterprise.server.logging.ApusicKafkaHandler.formatter=com.apusic.enterprise.server.logging.ODLLogFormatter;设置Kafka日志格式, ODL/ULF	ODL
压缩数据类型	com.apusic.enterprise.server.logging.ApusicKafkaHandler.compressionType=none;Producer用于压缩数据的压缩类型, 包括: none, gzip, snappy, lz4, zstd	none
需要确认的服务数量	消息推送完成需要收到的服务确认数量, 取值为all、0、1	0
Kafka集群服务	com.apusic.enterprise.server.logging.ApusicKafkaHandler.bootstrapServers=localhost:9092; 推送的Kafka集群地址, 可配置多个, 格式:host1:port1,host2:port2	localhost:9092
主题	com.apusic.enterprise.server.logging.ApusicKafkaHandler.topic=log;Kafka订阅的topic	log
生产者id	com.apusic.enterprise.server.logging.ApusicKafkaHandler.clientId=aas;用于追踪请求源的字符串id	aas
阻塞时长	com.apusic.enterprise.server.logging.ApusicKafkaHandler.maxBlockMs=10000;当 buffer 空间不够或者metadata 丢失时产生的阻塞时长, 取值为0-3600000	10000
重试次数	com.apusic.enterprise.server.logging.ApusicKafkaHandler.retries=1;推送失败进行重试的次数, 取值为0-65535	1
消息延迟发送时间	com.apusic.enterprise.server.logging.ApusicKafkaHandler.lingerMs=0;分批数据最大空闲时间, 超过该时间的batch也会被发送到broker端, 取值为0-3600000	0
缓存内存	com.apusic.enterprise.server.logging.ApusicKafkaHandler.bufferMemory=33554432;用于缓存推送数据的内存大小, 取值为0-33554432, 即最大为32M。单位字节	33554432
分批容量大小	com.apusic.enterprise.server.logging.ApusicKafkaHandler.batchSize=16384;分批发送到broker端的批容量大小, 取值为0-33554432, 即最大为32M, 单位字节	16384

### 30.1.3 示例

示例1 设置日志属性

```
asadmin> set-log-attributes --target=server
com.apusic.enterprise.server.logging.ApusicFileHandler.maxHistoryFiles=12

com.apusic.enterprise.server.logging.ApusicFileHandler.maxHistoryFiles logging attribute set with value
12.
These logging attributes are set for server.
```

```
Command set-log-attributes executed successfully.
```

## 30.2 设置日志级别

`set-log-level` 子命令设置一个或多个记录器的日志级别。更改会动态生效。根据您的设置的日志级别，可能需要重新启动服务器。

此命令仅在远程模式下支持。

### 30.2.1 概要

```
set-log-levels [--help]
[--target=target]
logger-name=logger-level[:logger-name=logger-level]*
```

### 30.2.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--target
```

指定目标。server、instance、cluster。如果省略此选项，将为默认server设置属性。

```
logger-name
```

日志记录器的名称。 `list-log-levels` 子命令可用于列出当前日志记录器的名称。

```
logger-level
```

设置日志的等级，可选项有SEVERE, WARNING, INFO, CONFIG, FINE, FINER, 和FINEST。默认值为INFO。

### 30.2.3 示例

示例1 设置日志级别

此示例先查询当前记录器的日志级别，然后设置 `javax.enterprise.system.container.web` 的日志级别 `WARNING`，`javax.enterprise.system.container.ejb` 的日志级别为 `SEVERE`。

```
# 查看当前日志级别
asadmin> list-log-levels
ShoalLogger          <CONFIG>
com.apusic.aas.admingui <INFO>
com.apusic.aas.jdbc.pools.trace <OFF>
com.apusic.aas.logging.ApusicConsoleHandler <INFO>
com.apusic.aas.naming <INFO>
com.apusic.aas.thread.pools.trace <OFF>
com.apusic.enterprise.server.logging.ApusicFileHandler <ALL>
java.util.logging.ConsoleHandler <INFO>
javax.com.apusic.aas.persistence <INFO>
javax.enterprise.resource.corba <INFO>
javax.enterprise.resource.javamail <INFO>
javax.enterprise.resource.jdo <INFO>
javax.enterprise.resource.jms <INFO>
javax.enterprise.resource.jta <INFO>
javax.enterprise.resource.resourceadapter <INFO>
```

```

javax.enterprise.resource.sqltrace      <FINE>
javax.enterprise.resource.webcontainer.jsf.application <INFO>
javax.enterprise.resource.webcontainer.jsf.config    <INFO>
javax.enterprise.resource.webcontainer.jsf.context   <INFO>
javax.enterprise.resource.webcontainer.jsf.facelets  <INFO>
javax.enterprise.resource.webcontainer.jsf.lifecycle <INFO>
javax.enterprise.resource.webcontainer.jsf.managedbean <INFO>
javax.enterprise.resource.webcontainer.jsf.renderkit <INFO>
javax.enterprise.resource.webcontainer.jsf.resource  <INFO>
javax.enterprise.resource.webcontainer.jsf.taglib    <INFO>
javax.enterprise.resource.webcontainer.jsf.timing    <INFO>
javax.enterprise.system.container.cmp    <INFO>
javax.enterprise.system.container.ejb    <INFO>
javax.enterprise.system.container.ejb.mdb <INFO>
javax.enterprise.system.container.web    <INFO>
javax.enterprise.system.core.classloading <INFO>
javax.enterprise.system.core.config     <INFO>
javax.enterprise.system.core            <INFO>
javax.enterprise.system.core.security   <INFO>
javax.enterprise.system.core.selfmanagement <INFO>
javax.enterprise.system.core.transaction <INFO>
javax.enterprise.system <INFO>
javax.enterprise.system.ssl.security     <INFO>
javax.enterprise.system.tools.admin      <INFO>
javax.enterprise.system.tools.backup     <INFO>
javax.enterprise.system.tools.deployment <INFO>
javax.enterprise.system.util            <INFO>
javax.enterprise.system.webservices.registry <INFO>
javax.enterprise.system.webservices.rpc  <INFO>
javax.enterprise.system.webservices.saaj <INFO>
javax.enterprise.web.core                <INFO>
javax <INFO>
javax.mail                               <INFO>
org.apache.catalina                      <INFO>
org.apache.coyote                        <INFO>
org.apache.jasper                        <INFO>
org.eclipse.persistence.session          <INFO>
org.jvnet.hk2.osgiadapter                <INFO>
Command list-log-levels executed successfully.

#设置日志级别
asadmin> set-log-levels
javax.enterprise.system.container.web=WARNING:javax.enterprise.system.container.ejb=SEVERE
javax.enterprise.system.container.web package set with log level WARNING.
javax.enterprise.system.container.ejb package set with log level SEVERE.
These logging levels are set for server.
Command set-log-levels executed successfully.

```

### 30.3 滚动日志

rotate-log 子命令滚动日志，使用时间戳命名，`server.log_date-and-time`，并创建新的日志文件。更改动态生效，即不需要服务器重启。

日志队列的大小可以通过 `logging.properties` 文件进行配置。日志轮换基于文件大小或时间间隔。在某些情况下，队列可能会被填满，特别是如果日志级别设置为FINEST并且服务器上有大量活动。在这种情况下，会使用 rotate-log 子命令立即轮换服务器日志。

此命令仅在远程模式下支持。

### 30.3.1 概要

```
rotate-log [--target=target]
```

### 30.3.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--target
```

指定目标。server、instance、cluster。如果省略此选项，将为默认server设置属性。

### 30.3.3 示例

示例1 滚动日志

该示例执行滚动日志操作。

```
asadmin> rotate-log
Command rotate-log executed successfully.
```

## 30.4 收集日志

`collect-log-files` 子命令用于收集域管理服务器（DAS）、指定集群或指定服务器实例的所有可用日志文件，并创建日志文件的单个ZIP存档。如果指定了集群，ZIP存档还包含DAS的日志文件。

此命令仅在远程模式下支持。

### 30.4.1 概要

```
collect-log-files [--help] [--target target]
[--retrieve={false|true}] [retrievefilepath]
```

### 30.4.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--retrieve
```

指定是否在默认目录以外的其他目录中创建ZIP归档文件。

默认情况下，ZIP存档是在 `DOMAIN_HOME/cometed-logs` 目录中创建的。ZIP文件名由指定的目标和时间戳构成，如下所示

```
log_yyyy-mm-dd_hh-min-sec.zip
```

可以设置为以下值：

- false

ZIP存档将在默认目录中创建。如果省略，则 `--retrieve` 选项默认为false。

- true

ZIP存档将在检索文件路径属性值指定的目录中创建。如果省略检索文件路径，则将在默认目录中创建ZIP存档。

```
--target
```

指定目标。server、instance、cluster。如果省略此选项，将为默认server设置属性。

```
retrievefilepath
```

将在其中创建ZIP存档的目录的名称。如果省略此属性值，则将在默认目录中创建ZIP存档。如果 `--retrieve` 选项为false，则忽略此操作。

### 30.4.3 示例

#### 示例1 收集日志

该示例直接执行收集日志命令。

```
asadmin> collect-log-files
Created Zip file under /opt/ApusicAS/aas/domains/mydomain/collected-logs/log_2025-05-16_11-12-13.zip.
Command collect-log-files executed successfully.
```

#### 示例2 收集指定集群日志

该示例收集指定的集群，cluster1日志。

```
asadmin> collect-log-files --target cluster1
Log files are downloaded for instance1.
Log files are downloaded for instance2.
Created Zip file under /opt/ApusicAS/aas/domains/mydomain/collected-logs/log_2025-05-15_15-54-06.zip.
Command collect-log-files executed successfully.
```

## 31 安全管理

### 31.1 开启安全管理

`enable-secure-admin` 子命令的作用是让域管理服务器（DAS）和域中的实例使用SSL证书来加密它们之间发送的消息。此子命令还允许域管理服务器接受来自远程管理客户端（如 `asadmin` 实用工具和集成开发环境（IDEs））的管理消息。

简单来说，这个命令增强了Apusic应用服务器域的安全性，通过启用SSL加密来保护管理通信，防止未经授权的访问和数据泄露。当启用此功能后，所有管理相关的通信都会通过加密的SSL连接进行，提高了系统的整体安全性。

当启用安全管理员功能时，默认情况下，域管理服务器（DAS）和实例会使用SSL证书作为安全“主体”来相互认证，并授权管理员访问。`--asadminalias` 值指示DAS应使用哪个SSL证书来向实例标识自己。`--instancealias` 值则决定了实例应使用哪个SSL证书来向DAS标识自己。

如果域中任何管理用户的密码为空，则 `enable-secure-admin` 子命令会失败。

另外，你也可以使用 `enable-secure-admin-internal-user` 子命令，让服务器使用安全管理员用户名和密码来进行身份验证。

注意：在启用或禁用安全管理员功能后，您必须重新启动域中所有正在运行的服务器。如果只有DAS（Domain Administration Server）在运行，启用或禁用安全管理功能会更简单，然后重启DAS，再启动任何其他实例。

#### 31.1.1 概要

```
enable-secure-admin [--help]
[--adminalias=alias]
[--instancealias=alias]
```

#### 31.1.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--adminalias
```

在DAS（部署管理服务）上引用SSL/TLS证书的别名。此别名由DAS用于向实例标识自身。默认值为kaas。

```
--instancealias
```

实例上引用SSL/TLS证书的别名。此别名由实例用于向DAS标识自身。默认值为aas-instance。

#### 31.1.3 示例

示例1：为域启用安全管理

以下示例展示了如何使用管理员别名adtest和实例别名intest为域启用安全管理。

```
asadmin> enable-secure-admin --adminalias adtest --instancealias intest
Command enable-secure-admin executed successfully.
```

### 31.2 禁用安全管理

`disable-secure-admin` 子命令用于在已启用安全管理员功能的情况下禁用该功能。

注意：在启用或禁用安全管理员功能后，您必须重启域中的任何正在运行的服务器。最简单的方法是，仅在域管理服务器（DAS）运行时启用或禁用安全管理员功能，然后重启DAS，之后再启动任何其他实例。

#### 31.2.1 概要

```
disable-secure-admin [--help]
```

### 31.2.2 选项

```
--help  
-?
```

显示子命令的帮助文本。

### 31.2.3 示例

示例1: 为域禁用安全管理

以下示例展示了如何禁用安全管理。

```
asadmin> disable-secure-admin  
Command disable-secure-admin executed successfully.
```

## 32 list子命令

### 32.1 list子命令

`list` 子命令列出了 {product---name} 的可配置和可监视属性。`list` 子命令的输出是表示单个服务器组件和子系统的以点分隔的名称列表。例如，`applications.application`。了解特定组件或子系统后，您可以使用 `get` 子命令访问任何属性，使用 `set` 子命令修改可配置属性。以下规则适用于 `list` 子命令中的带点名称：

- 任何带有以点分隔的名称且后面没有通配符 (\*) 的 `list` 子命令都会列出当前节点的直接子节点。例如，以下命令列出了属于服务器节点的所有直接子节点：

```
asadmin> list server
```

- 任何带有以点分隔的名称后跟通配符 (\*) 的 `list` 子命令都会列出当前节点的子节点层次树。例如，以下命令列出了应用程序的所有子节点及其后续子节点，以此类推：

```
asadmin> list applications.*
```

- 任何 `list` 子命令，如果其点名称前面或后面有 `*dotted name` 或 `dottedname*` 形式的通配符 (\*)，则会列出与提供的匹配模式创建的正则表达式匹配的所有节点及其子节点。

有关带点名称的详细信息，请参阅 `dotted-name (5ASC)` 帮助页面。

#### 32.1.1 概要

```
list [--help]
[--monitor={false| true}]
[dotted-parent-attribute-name]
```

#### 32.1.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--monitor
-m
```

默认为false。如果设置为false，则返回可配置的属性值。如果设置为true，则返回可监视的属性值。

```
dotted-parent-attribute-name
```

可配置的属性名称。

#### 32.1.3 示例

示例1 获取所有的属性

此示例获取当前所有的可配置值。

```
asadmin> list *
```

示例2 列出可监视对象

此示例列出了启用监视的可监视对象的名称。

```
asadmin> list --monitor *
```

## 33 get子命令

### 33.1 get子命令

get 子命令获取可配置或可监控属性的值。

get子命令使用点名称来获取{product---name}元素的`可配置或可监视属性`的名称和值。您可以使用 `list` 子命令显示代表单个服务器组件和子系统的以点分隔的名称。例如，带点名称可能是 `applications.application`。监控层次结构中的属性是只读的，但可以使用 `set` 子命令修改配置属性。有关以点分隔的名称的更多详细信息，请参阅 `dotted-names (5ASC)` 帮助页面。

对shell或命令解释器具有特殊含义的字符，如\*（星号），应根据需要在shell中引用或转义，例如将参数括在引号中。在多模式中，只有包含空格、引号或反斜杠的参数才需要引号。

以下列表显示了带有\*（星号）的get子命令的常见用法：

```
get *
```

获取所有带点名称前缀上的所有值。

```
get domain.* 或 get domain*.*
```

获取以domain开头的带点名称上的所有值。

```
get *config.*
```

获取与 `config.` 匹配的带点名称上的所有值。

```
get configs.*.http-listener-1.*
```

获取 `configs` 开头包含 `http-listener-1` 的所有值。

#### 33.1.1 概要

```
get [--help]
    [--monitor={true|false}]
    [--aggregatedataonly={true|false}]
    (dotted-attribute--name)+
```

#### 33.1.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
--target
```

指定目标。server、instance、cluster。如果省略此选项，将为默认server设置属性。

```
--monitor
-m
```

默认为false。如果设置为false，则返回可配置的属性值。如果设置为true，则返回可监视的属性值。

```
--aggregatedataonly
-c
```

集群中所有 {product---name} 实例的监控数据。默认值为false。

```
dotted-attribute-name
```

以虚线表示法标识属性名称。至少需要一个点名称属性。虚线符号是用于访问可配置实体属性的语法。

### 33.1.3 示例

示例1 获取所有的属性

此示例获取当前所有的可配置或可监控属性的值。

```
asadmin> get *
```

示例2 获取以configs开头的所有可配置或监控的值。

```
asadmin> get configs.*
```

示例3 获取包括http-listener-1的所有可配置或监控的值。

```
asadmin> get *http-listener-1.*
```

## 34 set子命令

### 34.1 set子命令

set 子命令使用以点分隔的名称来修改一个或多个可配置属性的值。监控层次结构中的属性是只读的，但可以修改配置属性。您可以使用 list 子命令显示代表单个服务器组件和子系统的以点分隔的名称。例如，以点分隔的名称可能是 applications.application。发现特定组件或子系统后，可以使用 get 子命令访问属性。有关以点分隔的名称的更多详细信息，请参阅 dotted-names 部分

通过修改属性，您可以启用和禁用服务，并自定义现有元素的功能。应用服务器提供了 asadmin 子命令来更新某些元素；例如，更新密码别名，update-password-alias。

但是，要更新其他元素，必须使用 set 命令。例如，可以使用 create-jdbcconnection-pool 子命令创建JDBC连接池。如果需要更改属性设置，请使用 set 命令。使用 asadmin 实用程序子命令或管理控制台所做的任何更改都会自动应用于关联的 {product---name} 配置文件。

注意：对shell或命令解释器具有特殊含义的字符，如\*（星号），应根据需要在shell中引用或转义，例如将参数括在引号中。在多模式中，只有包含空格、引号或反斜杠的参数才需要引号。

#### 34.1.1 概要

```
set [--help] attribute-name=value
```

#### 34.1.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
attribute-name
```

日志记录属性的全范围名称。list-log-attributes 子命令可用于列出当前定义的所有属性的名称。

```
attribute-value
```

标识属性名称的完整的以点分隔的名称及其值。

#### 34.1.3 示例

示例1 设置线程池参数

此示例设置线程池 http-thread-pool 最大线程池大小为300。通常情况下，如果不清楚属性的全称，可以使用 get 子命令获取属性，再通过 set 子命令设置。

##### #查看相关属性

```
asadmin> get *http-thread-pool.*
configs.config.default-config.thread-pools.thread-pool.http-thread-
pool.classname=org.Apusic.grizzly.threadpool.GrizzlyExecutorService
configs.config.default-config.thread-pools.thread-pool.http-thread-pool.idle-thread-timeout-seconds=900
configs.config.default-config.thread-pools.thread-pool.http-thread-pool.max-queue-size=4096
configs.config.default-config.thread-pools.thread-pool.http-thread-pool.max-thread-pool-size=250
configs.config.default-config.thread-pools.thread-pool.http-thread-pool.min-thread-pool-size=5
configs.config.default-config.thread-pools.thread-pool.http-thread-pool.name=http-thread-pool
configs.config.default-config.thread-pools.thread-pool.http-thread-pool.thread-priority=5
configs.config.server-config.thread-pools.thread-pool.http-thread-
pool.classname=org.Apusic.grizzly.threadpool.GrizzlyExecutorService
configs.config.server-config.thread-pools.thread-pool.http-thread-pool.idle-thread-timeout-seconds=900
configs.config.server-config.thread-pools.thread-pool.http-thread-pool.max-queue-size=4096
configs.config.server-config.thread-pools.thread-pool.http-thread-pool.max-thread-pool-size=250
configs.config.server-config.thread-pools.thread-pool.http-thread-pool.min-thread-pool-size=5
configs.config.server-config.thread-pools.thread-pool.http-thread-pool.name=http-thread-pool
```

```
configs.config.server-config.thread-pools.thread-pool.http-thread-pool.thread-priority=5
Command get executed successfully.
```

#### #设置属性

```
asadmin> set configs.config.server-config.thread-pools.thread-pool.http-thread-pool.max-thread-pool-
size=300
configs.config.default-config.thread-pools.thread-pool.http-thread-pool.max-thread-pool-size=300
Command set executed successfully.
```

#### 示例2 设置JDBC连接池参数

此示例设置JDBC连接池 mydata 的初始和最小线程池大小为10。通常情况下，如果不清楚属性的全称，可以使用 get 子命令获取属性，再通过 set 子命令设置。

#### #查看相关属性

```
asadmin> get *.mydata.*
Enter admin password for user "admin">
resources.jdbc-connection-pool.mydata.property.port=5236
resources.jdbc-connection-pool.mydata.property.password=
resources.jdbc-connection-pool.mydata.property.user=DATATEST
resources.jdbc-connection-pool.mydata.property.url=jdbc:dm://127.0.0.1:5236/DATATEST
resources.jdbc-connection-pool.mydata.property.databaseName=DATATEST
resources.jdbc-connection-pool.mydata.allow-non-component-callers=false
resources.jdbc-connection-pool.mydata.associate-with-thread=false
resources.jdbc-connection-pool.mydata.connection-creation-retry-attempts=0
resources.jdbc-connection-pool.mydata.connection-creation-retry-interval-in-seconds=10
resources.jdbc-connection-pool.mydata.connection-leak-reclaim=false
resources.jdbc-connection-pool.mydata.connection-leak-timeout-in-seconds=0
resources.jdbc-connection-pool.mydata.connection-validation-method=auto-commit
resources.jdbc-connection-pool.mydata.deployment-order=100
resources.jdbc-connection-pool.mydata.driver-classname=dm.jdbc.driver.DmDriver
resources.jdbc-connection-pool.mydata.fail-all-connections=false
resources.jdbc-connection-pool.mydata.idle-timeout-in-seconds=300
resources.jdbc-connection-pool.mydata.is-connection-validation-required=false
resources.jdbc-connection-pool.mydata.is-isolation-level-guaranteed=true
resources.jdbc-connection-pool.mydata.lazy-connection-association=false
resources.jdbc-connection-pool.mydata.lazy-connection-enlistment=false
resources.jdbc-connection-pool.mydata.match-connections=false
resources.jdbc-connection-pool.mydata.max-age-time-in-seconds=0
resources.jdbc-connection-pool.mydata.max-connection-usage-count=0
resources.jdbc-connection-pool.mydata.max-pool-size=32
resources.jdbc-connection-pool.mydata.max-wait-time-in-millis=60000
resources.jdbc-connection-pool.mydata.name=mydata
resources.jdbc-connection-pool.mydata.non-transactional-connections=false
resources.jdbc-connection-pool.mydata.object-type=user
resources.jdbc-connection-pool.mydata.ping=false
resources.jdbc-connection-pool.mydata.pool-resize-quantity=2
resources.jdbc-connection-pool.mydata.pooling=true
resources.jdbc-connection-pool.mydata.res-type=java.sql.Driver
resources.jdbc-connection-pool.mydata.secret-level=SECERT
resources.jdbc-connection-pool.mydata.statement-cache-size=0
resources.jdbc-connection-pool.mydata.statement-cache-type=
resources.jdbc-connection-pool.mydata.statement-leak-reclaim=false
resources.jdbc-connection-pool.mydata.statement-leak-timeout-in-seconds=0
resources.jdbc-connection-pool.mydata.statement-timeout-in-seconds=-1
resources.jdbc-connection-pool.mydata.steady-pool-size=8
```

```
resources.jdbc-connection-pool.mydata.validate-atmost-once-period-in-seconds=0  
resources.jdbc-connection-pool.mydata.wrap-jdbc-objects=true  
Command get executed successfully.
```

#### #设置属性

```
asadmin> set resources.jdbc-connection-pool.mydata.steady-pool-size=10  
resources.jdbc-connection-pool.mydata.steady-pool-size=10  
Command set executed successfully.
```

## 35 export子命令

### 35.1 export子命令

`export` 子命令标记一个变量名，以便在多模式下自动导出到后续命令的环境中。在多模式中，`export` 子命令标记一个环境变量，以便自动导出到后续命令的环境中。除非退出多模式，否则所有后续命令都使用指定的变量名值，或者使用 `unset` 子命令取消设置变量。如果只指定了变量名，则显示该变量名的当前值。如果使用导出子命令时没有任何参数，则会显示所有导出变量及其值的列表。在调用 `asadmin` 实用程序之前设置的导出shell环境变量会自动导入，并在 `asadmin` 中设置为导出变量。`asadmin` 实用程序无法读取未导出的环境变量。此子命令仅在本地模式下受支持。

#### 35.1.1 概要

```
export [--help]
[variable-name=value [variable-name=value]*]
```

#### 35.1.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
variable-name
```

变量名称和值，用于自动导出到环境中供后续命令使用。

#### 35.1.3 示例

示例1 列出所有已经设置的环境变量

```
asadmin> export
AS_ADMIN_USER = admin
AS_ADMIN_PORT = 6848
Command export executed successfully
```

## 36 unset子命令

### 36.1 unset子命令

`unset` 子命令删除为多模式环境设置的一个或多个环境变量。删除后，变量及其相关值不再适用于多模式环境。要列出已设置的环境变量，请使用不带选项的 `export` 子命令。如果 `export` 子命令没有列出环境变量，则不会设置环境变量。此子命令仅在本地模式下受支持。

#### 36.1.1 概要

```
unset [--help]
variable-list
```

#### 36.1.2 选项

```
--help
-?
```

显示子命令的帮助文本。

```
variable-list
```

以空格分隔的要删除的环境变量列表。

#### 36.1.3 示例

示例1 删除已设置的环境变量

此示例先列出已经设置的环境变量，再删除环境变量 `AS_ADMIN_PREFIX`。

```
asadmin> export
AS_ADMIN_USER = admin
AS_ADMIN_HOST = bluestar
AS_ADMIN_PREFIX = server1.jms-service
AS_ADMIN_PORT = 8000
Command export executed successfully

asadmin> unset AS_ADMIN_PREFIX
Command unset executed successfully
```

## 37 enable子命令

### 37.1 enable子命令

enable 子命令启用指定的已部署组件。如果组件已经启用，则重新启用。如果尚未部署，则返回错误消息。此子命令在远程模式下受支持。

#### 37.1.1 概要

```
enable [--help] [--target target_name] component_name
```

#### 37.1.2 选项

```
--help  
-?
```

显示子命令的帮助文本。

```
--target
```

指定目标。server、域名、集群名称或实例名称。如果省略此选项，将为默认server设置属性

```
component_name
```

要启用的组件的名称。

#### 37.1.3 示例

示例1 启动已部署的应用

此示例启动已部署的应用test1.

```
asadmin> enable test1  
Command enable executed successfully
```

## 38 disable子命令

### 38.1 disable子命令

disabled 子命令禁用指定的已部署组件。如果尚未部署，则返回错误消息。此子命令在远程模式下受支持。

#### 38.1.1 概要

```
disable [--help] [--target target_name] component_name
```

#### 38.1.2 选项

```
--help  
-?
```

显示子命令的帮助文本。

```
--target
```

指定目标。server、域名、集群名称或实例名称。如果省略此选项，将为默认server设置属性

```
component_name
```

要禁用的组件的名称。

#### 38.1.3 示例

示例1 禁用已启用的应用

此示例禁用已启用的应用test1.

```
asadmin> disable test1  
Command disable executed successfully
```

## 39 角色管理

### 39.1 管理角色子命令

`save-role-with-db` 子命令用于管理角色信息，新建或编辑角色信息都可用该命令。需要使用安全管理员用户设置。

#### 39.1.1 概要

```
save-role-with-db [--help] [--roleDeveloper={false|true}] --resources resources roleName
```

#### 39.1.2 选项

```
--help  
-?
```

显示子命令的帮助文本。

```
--roleDeveloper
```

是否为开发者角色。如果为true，则该角色没有修改配置、资源和配置模块的部分权限。默认为false。

```
resources
```

该角色可以访问的资源权限。多个用分号 `:` 区分。

```
roleName
```

定义角色名称。

#### 39.1.3 示例

示例1 新建角色

该示例新建角色 `myrole`，设置可访问的资源为 首页和域，`common_index:sysadmin_domain`。

```
save-role-with-db --resources common_index:sysadmin_domain:sysadmin_applications:sysadmin_monitor myrole
```

## 40 用户管理

### 40.1 管理用户子命令

`save-user-with-db` 子命令用于管理角色信息，新建或编辑用户信息都可用该命令。需要使用安全管理员用户设置。

#### 40.1.1 概要

```
save-user-with-db [--help] [--role roleName] --pw password userName
```

#### 40.1.2 选项

```
--help  
-?
```

显示子命令的帮助文本。

```
--role
```

为该用户指定角色。

```
--pw
```

设置用户密码。

```
userName
```

定义用户名称。

#### 40.1.3 示例

示例1 新建用户

该示例新建用户，名称为 `myuser`，密码为 `myuser123`，角色为 `myrole`。

```
save-user-with-db --role myrole --pw myuser123 myuser
```

## 41 附件

### 41.1 dotted-names

`dotted-names` 表示使用句点分隔name元素的语法。

带点名称是特定(`product---name`)元素的标识符, 例如可配置或可监控对象。点名称使用英文句点 (.) (称为点) 作为分隔符来分隔元素名称的各个部分。带点名称中的句点类似于斜线 (/) 字符, 用于分隔UNIX文件系统中文件绝对路径名中的级别。asadmin实用程序的子命令使用带点名称, 如下所示:

- `list` 子命令提供管理组件属性的完全限定名称。
- `get` 子命令提供对属性的访问。
- `set` 子命令允许您修改可配置属性和设置属性。

配置层次结构松散地基于域的模式文档, 属性是可修改的。监控层次结构的属性是只读的。以下格式用于配置带点名称:

```
config-name.config-element-name.primary-key.attribute-name
instance-name.config-element-name.primary-key.attribute-name
```

以下格式用于资源带点名称:

```
server-name.resource-name.primary-key.attribute-name
domain.resources.resource-name.primary-key.attribute-name
```

以下规则适用于形成点名称:

- 顶层是配置、服务器或域名。例如, `server-config (default-config)`、`server (default-server)` 或 `domain1 (default domain)`。
- 点 (.) 是分隔名称的两个连续部分。
- 名称的一部分通常标识服务器子系统或其特定实例。例如, `web-container`、`log-service`、`thread-pool-1`。
- 如果名称本身的任何部分包含点 (.), 则必须用前导 \ (反斜杠) 转义该点, 以便 . (点) 的作用不像分隔符。
- 星号 (\*) 字符可以在带点名称中的任何位置使用, 其作用类似于正则表达式中的通配符。此外, 一个 (\*) 可以折叠带点名称的所有部分。 (\*) 字符可以在带点名称中的任何位置使用, 其作用类似于正则表达式中的通配符。此外, 一个 (\*) 可以折叠带点名称的所有部分。例如, 像 `this.is.really.long.hierarchy` 这样的长点名称可以缩写为 `th*.hierarchy`。这个 . (点) 是分隔带点名称的各个部分。
- `get` 和 `list` 子命令的 `--monitor` 选项选择监视或配置层次结构。如果子命令指定 `--monitor=false` (默认值), 则选择配置层次结构。如果子命令指定 `--monitor=true`, 则选择监视层次结构。

全国统一服务热线  
4008-555-800



金蝶天燕云计算股份有限公司(简称“金蝶天燕云”)成立于2000年,前身为“金蝶中间件公司”,是金蝶集团旗下新一代软件基础云平台服务商,云计算国家标准制定企业,国家信创产业核心软件企业。金蝶天燕是国家863重点研发计划与核高基重大专项承接企业,也是“两网一站四库十二金”国家重点工程的基础平台提供商,产品广泛应用于政府、军工、金融、能源等关键行业,累计服务客户总数超过10万家。

**Apusic**  
金蝶天燕

云计算国家标准制定企业  
金蝶集团旗下基础软件企业  
信息技术应用创新核心企业  
官网: [www.apusic.com](http://www.apusic.com)

