

工具和模块 {#Tools_and_Modules}

工具和模块描述了与Apache Geode相关的工具和模块。

- **gfsh**

`gfsh`（发音为"jee-fish"）提供了一个强大的命令行界面，您可以从中启动，管理和监控Geode进程，数据和应用程序。

- **Gemcached**

Gemcached是一个Geode适配器，允许Memcached客户端与Geode服务器集群通信，就像服务器是memcached服务器一样。Memcached是一个开源缓存解决方案，它使用分布式内存中的哈希映射来存储字符串或对象数据的键值对。

- **HTTP Session Management Modules**

Apache Geode HTTP会话管理模块为HTTP服务器提供快速，可扩展且可靠的会话复制，而无需更改应用程序。

- **Geode Pulse**

Geode Pulse是一个Web应用程序，它提供了一个图形仪表板，用于监视Geode集群，成员和区域的重要实时运行状况和性能。

- **Geode Redis Adapter**

Geode Redis适配器允许Geode作为Redis数据存储的直接替代品，让Redis应用程序可以利用Geode的扩展功能而无需更改其客户端代码。Redis客户端使用IP地址和端口号以与连接Redis服务器相同的方式连接到Geode服务器。

- **Apache Lucene® Integration**

ApacheLucene®集成使用户能够创建Lucene索引并对存储在Geode中的数据执行Lucene搜索。

gfsh {#gfsh}

`gfsh`（发音为"jee-fish"）提供了一个强大的命令行界面，您可以从中启动，管理和监控Geode进程，数据和应用程序。

- **What You Can Do with gfsh]{#What You Can Do with gfsh {#What_You_Can_Do_with_gfsh}}**

`gfsh` 支持Apache Geode进程和应用程序的管理，调试和部署。

- [Startinggfsh](#)

在开始 `gfsh` 之前，请确认您已设置JAVA_HOME，并且您的PATH变量包含 `gfsh` 可执行文件。

- **Configuring the gfsh Environment**

`gfsh.bat` 和 `gfsh` bash脚本自动将所需的Apache Geode和JDK的 `.jar` 库附加到现有的CLASSPATH。您可以为安全性，环境变量，日志记录和故障排除设置用户可配置的属性。

- **Useful gfsh Shell Variables**

您可以在脚本中使用内置的 `gfsh` shell变量。

- **Basic Shell Features and Command-Line Usage**

gfsh 实用程序为shell环境提供了有用的功能，包括命令自动完成，保留的命令历史记录和多行命令的分隔。上下文相关的帮助可通过命令和主题获得。

- **Specifying JSON within Command-Line Options**

一些 gfsh 命令允许在命令行选项中使用JSON规范。

- **Tutorial—Performing Common Tasks with gfsh**

本主题将指导您在启动 gfsh 后执行的典型任务序列。

- **Quick Reference of gfsh Commands by Functional Area**

此快速参考将所有命令分类到功能区域。

- **gfsh Command Help**

本节提供按字母顺序列出的所有 gfsh 命令的帮助和用法信息。

- **Creating and Running gfsh Command Scripts**

gfsh 提供了几种在脚本环境中运行命令的方法。

- **Running gfsh Commands on the OS Command Line**

- **Mapping of cache.xml Elements to gfsh Configuration Commands.**

您可以使用cache.xml文件配置Geode集群，也可以使用 gfsh 和集群配置服务来配置集群。本节将cache.xml元素映射到配置和管理集群的 gfsh 命令。

你可以用gfsh做什么 {#What_You_Can_Do_with_gfsh}

gfsh 支持Apache Geode进程和应用程序的管理，调试和部署。

使用 gfsh，您可以：

- 启动和停止Apache Geode进程，例如定位器和缓存服务器
- 启动和停止网关发送方和网关接收方进程
- 部署应用程序
- 创建和销毁区域
- 执行函数
- 管理磁盘存储
- 导入和导出数据
- 监控Apache Geode进程
- 启动Apache Geode监控工具

gfsh 命令行界面使开发人员可以花更少的时间配置缓存实例XML，属性，日志和统计信息。gfsh命令生成报告；捕获集群范围的统计数据；并支持导出统计信息，日志和配置。与Spring Roo一样，gfsh具有命令完成功能（因此您无需了解语法），上下文相关帮助，脚本以及使用简单API从应用程序内调用任何命令的功能。gfsh接口使用 JMX/RMI 与Apache Geode进程通信。

您可以使用HTTP协议将gfsh连接到远程集群。请参阅[使用gfsh通过HTTP或HTTPS管理远程集群](#)。

默认情况下，集群配置服务会在您使用gfsh创建Apache Geode对象时保存Apache Geode集群的配置。您可以导出此配置并将其导入另一个Apache Geode集群。请参见[集群配置服务概述](#)。

启动gfsh {#Starting_gfsh}

在启动gfsh之前，请确认您已设置JAVA_HOME并且您的PATH变量包含gfsh可执行文件。

注意:在Windows上，必须正确设置JAVA_HOME环境变量，才能对定位器和服务器使用start，stop和status命令。

要启动gfsh命令行界面，请在当前随Apache Geode一起安装的任何计算机上的提示符处执行以下命令：

Start gfsh on Windows:

```
1 | <product_directory>\bin\gfsh.bat
```

其中<product_directory>对应于安装Apache Geode的位置。

Start gfsh on Unix:

```
1 | <product_directory>/bin/gfsh
```

其中<product_directory>对应于安装Apache Geode的位置。执行时，gfsh脚本将所需的Apache Geode和JDK的jar库附加到现有的CLASSPATH。

如果您已成功启动gfsh，则会出现gfsh启动画面和提示。

```
1 | c:\Geode\Latest>gfsh.bat
2 |
3 |  _____
4 |  /  _  /  _  /  _  /  _  /
5 | /  _  /  _  /  _  /  _  /
6 | /  _  /  _  /  _  /  _  /
7 |
8 | Monitor and Manage Geode
9 | gfsh>
```

您也可以直接在终端中运行一些gfsh命令，而无需输入gfsh提示。例如，在Unix/Linux上你可以输入：

```
1 | $ gfsh start server --name=server1
```

或在Windows上：

```
1 | prompt> gfsh start server --name=server1
```

有关详细信息，请参阅[创建和运行gfsh命令脚本](#)。

配置gfsh环境 {#Configuring the gfsh Environment}

gfsh.bat 和 gfsh bash脚本自动将所需的Apache Geode和JDK jar 库附加到现有的CLASSPATH。您可以为安全性，环境变量，日志记录和故障排除设置用户可配置的属性。

CLASSPATH中的JAR库

当你启动gfsh时，它会自动加载已经打包在gfsh-dependencies.jar文件中的所需JAR文件。您无需修改CLASSPATH即可运行gfsh。

JAR文件打包在lib目录的安装目录中。

机器主机名

在某些操作系统上，您可能需要确保在系统主机文件中配置了计算机的主机名。例如，在macOS上，您可能需要将机器的主机名映射到 `/etc/hosts` 文件中的IP地址，以便 gfsh 和Pulse正常运行。

配置gfsh安全性

由于 gfsh 必须连接到JMX Manager成员才能运行某些命令（即管理和监视其他成员的那些命令），因此JMX Manager配置属性可能会影响 gfsh 安全性。在 `gemfire.properties` 中，以下Geode属性可以影响JMX Manager的 gfsh connection设置：

- `jmx-manager-ssl`
- `jmx-manager-port`
- `jmx-manager-password-file`
- `jmx-manager-access-file`

您可能还需要验证端口是否可用并对客户端连接开放。有关这些安全属性的详细信息，请参阅[配置JMX Manager](#)。

配置gfsh环境变量

此外，您可以使用 `set variable` 命令设置特定于gfsh的预设SHELL变量。例如，您可以将 gfsh 设置为以安静模式运行。并非所有 gfsh 变量都是可修改的。用户可配置的变量包括：

- `APP_FETCH_SIZE`
- `APP_QUIET_EXECUTION`

有关详细信息，请参阅[有用的gfsh Shell变量](#)。

配置gfsh会话日志记录

默认情况下，禁用 gfsh 会话日志记录。要启用gfsh日志记录，必须设置Java系统属性 `-Dgfsh.log-level=desired_log_level`，其中`desired_log_level`是以下值之一：`severe`, `warning`, `info`, `config`, `fine`, `finest`。例如，在Linux中：

```
1 | export JAVA_ARGS=-Dgfsh.log-level=info
```

然后，启动 gfsh。

gfsh生成一个名为`gfsh-%u_%g.log`的日志文件。此日志文件记录单个gfsh会话的事件。它包括环境信息，例如java和系统信息，以及详细的命令执行。变量替换如下：

- `%u` - 解决冲突的唯一编号
- `%g` - 用于区分旋转日志的世代号

gfsh使用JDK Logger生成gfsh会话日志文件。有关如何在命名日志文件时使用变量的说明，请参见<http://docs.oracle.com/javase/7/docs/api/java/util/logging/FileHandler.html>。无法更改生成的gfsh日志文件的默认名称。

默认情况下，日志文件将写入您执行 gfsh 或 gfsh.bat 脚本的当前工作目录。要修改写入日志文件的目录位置，请使用 `gfsh.log-dir` Java系统属性。例如：

```
1 | export JAVA_ARGS="-Dgfsh.log-level=info -Dgfsh.log-dir=/machinename/logs"
```

然后，启动 gfsh。

另外，gfsh在`${SYS_USER_HOME}/.geode/.gfsh.history`文件中记录了命令的历史记录，您可以使用它来创建脚本或查看过去的命令。

成员日志文件

gfsh为通过gfsh启动的任何成员写入几个日志文件。 有用的成员日志文件包括:

- **<locator_name>.log.** 详细说明定位器的配置 (包括所有 `gemfire.properties`) 以及启动后定位器上发生的所有活动。 此日志文件将写入以定位符命名的目录。 例如, 如果启动名为locator1的定位器, 则该文件将在 `<product_dir>/locator1` 目录中写为 `locator1.log`。
- **vf.gf.locator.pid.** 包含定位器的进程ID。 您可以使用PID来停止或查看此定位器的状态。 此文件将写入与定位器日志文件相同的目录位置。
- **<server_name>.log.** 详细说明服务器的配置 (包括所有 `gemfire.properties`) 以及启动后服务器上发生的所有活动。 此日志文件将写入以服务器命名的目录。 例如, 如果启动名为server1的服务器, 则该文件将在 `<product_dir>/server1` 目录中写为 `server1.log`。 如果停止并以相同的名称启动服务器, 则较旧的日志文件将保留在同一目录中, 但会重命名以进行版本控制。
- **vf.gf.server.pid.** 包含服务器的进程ID。 您可以使用PID来停止或查看此服务器的状态。 此文件将写入与服务器日志文件相同的位置。

查看标准输出和标准错误

默认情况下, Geode不会将应用程序写入的消息显示为标准输出和标准错误。 要允许将这些消息分别写入定位器和服务器日志文件, 请使用gfsh `start locator` 或 `start server` 命令指定 `--redirect-output` 选项。 例如, 以下命令会将 `stdout` 和 `stderr` 消息写入 `locator1.log` 文件:

```
1 | gfsh> start locator --name=locator1 --redirect-output
```

Tab自动补全

本节仅适用于UNIX安装

从UNIX bash shell运行gfsh命令时, 可以通过运行以下命令在shell中启用自动tab-completion:

```
1 | source <gemfire-install-directory>/bin/gfsh-completion.bash
```

运行此命令后, 您可以在从bash shell运行gfsh命令时使用自动完成。

请参阅[使用TAB自动补全](#)。

命令历史和gfsh.history

已成功执行的命令历史记录在运行gfsh的用户的主目录下的 `.gemfire` 目录中的 `.gfsh.history` 文件中。 您还可以使用 `history --file=your_file_name` 命令导出历史文件。

JMX Manager更新率和系统监控

当您执行数据操作 (例如put) 然后监视系统状态 (例如使用gfsh `show metrics` 命令或Geode Pulse) 时, 受监视的系统可能不会立即反映最近的操作。 例如, 如果执行put操作然后立即执行 `show metrics` gfsh命令, 则可能看不到该区域中正确的条目数。 管理层每2秒更新一次。 执行操作活动后等待几秒钟以查看最准确的结果。

您可以修改 `gemfire.properties` 中的 `jmx-manager-update-rate` 属性, 以增加或减少将更新推送到JMX Manager的速率 (以毫秒为单位)。 此属性设置应大于或等于 `statistic-sample-rate`。 如果遇到性能问题, 您可能希望提高此间隔; 但是, 将此值设置得太高会导致在 gfsh 和Geode Pulse中看到陈旧值。

格式化结果

本节仅适用于UNIX安装

诸如 `query` 之类的 `gfsh` 命令产生具有宽列的输出，这些列可能会变得不对齐并需要手动重新格式化以查看输出。如果输出无法适应终端的可用宽度，`gfsh` 会自动修剪列宽以适应。您可以通过将 `gfsh` 环境变量 `GFSH.TRIMSCRWIDTH` 设置为 `false` 来禁用此行为。

您可以将 `APP_RESULT_VIEWER` 变量设置为 `external`，以便使用 `UNIX less` 命令查看输出。

请参阅[配置gfsh环境变量](#)。

有用的gfsh Shell变量 {#Useful_gfsh_Shell_Variables}

您可以在脚本中使用内置的 `gfsh` shell 变量。

您还可以使用 `set variable` 命令修改 shell 行为或定义自己的变量。

要查看所有 `gfsh` shell 变量及其当前值的列表，请使用以下命令：

```
1 | gfsh>echo --string=$*
```

要获取现有变量的当前值，请使用以下命令语法（该变量必须括在大括号中）：

```
1 | gfsh>echo --string=${VARIABLE}
```

例如：

```
1 | gfsh>echo --string=${SYS_CLASSPATH}
```

系统变量

<code>SYS_CLASSPATH</code>	<code>gfsh</code> JVM的CLASSPATH（只读）。
<code>SYS_GEMFIRE_DIR</code>	已安装Geode的产品目录（只读）。
<code>SYS_HOST_NAME</code>	启动 <code>gfsh</code> 的主机（只读）。
<code>SYS_JAVA_VERSION</code>	使用的Java版本（只读）。
<code>SYS_OS</code>	操作系统名称（只读）。
<code>SYS_OS_LINE_SEPARATOR</code>	在编写 <code>gfsh</code> 脚本时可以使用行分隔符（\或^）变量。（只读）。
<code>SYS_USER</code>	用户名（只读）。
<code>SYS_USER_HOME</code>	用户的主目录（只读）。

GFSH环境变量

APP_FETCH_SIZE	查询时要使用的提取大小。 值:0 - 2147483647.默认值为100.
APP_LAST_EXIT_STATUS	上次命令退出状态。 与\$相似? (Unix) 和%errorlevel% (Windows)。 值:0 (成功) , 1 (错误) , 2 (崩溃) (只读) .
APP_LOGGING_ENABLED	Whether gfsh logging is enabled. Default: false (read only). You can enable gfsh logging by setting the <code>gfsh.log.level</code> Java system property to a supported Java log level .
APP_LOG_FILE	当前gfsh日志文件的路径和名称 (只读) .
APP_NAME	应用程序的名称 - "gfsh" (只读) .
APP_PWD	启动gfsh的当前工作目录 (只读) .
APP_QUERY_RESULTS_DISPLAY_MODE	切换显示模式以返回查询结果。 值: <code>table</code> or <code>catalog</code> (表或编目)。 默认值为 <code>table</code> .
APP_QUIET_EXECUTION	执行是否应该处于安静模式。 值 (不区分大小写) :true, false。 默认值为false.
APP_RESULT_VIEWER	仅限Unix。 将此变量设置为 <code>external</code> 以使用 UNIX <code>less</code> 命令查看输出。 默认值为basic (gfsh) .

基本Shell功能和命令行用法

{#Basic_Shell_Features_and_Command_Line_Usage}

`gfsh` 实用程序为shell环境提供了有用的功能，包括命令自动完成，保留的命令历史记录和多行命令的分隔。 上下文相关的帮助可通过命令和主题获得。

要查看可用的gfsh命令列表，请在空提示符下按Tab键。

您看到的命令列表取决于您是否连接到Geode群集。 如果未连接，则会看到可用的本地命令列表。

使用hint命令获取有关特定主题的信息。

`hint`命令显示指定主题的单行描述和关联命令。 例如，**`hint data`**返回数据主题的描述以及可用于数据的所有可能操作:

```

1 gfsh>hint Data
2 User data as stored in regions of the Geode distributed system.
3
```



```

4 create index : Create an index that can be used when executing queries.
5 destroy index : Destroy/Remove the specified index.
6 export data : Export user data from a region to a file.
7 get : Display an entry in a region. If using a region whose key
and value classes have been set, then specifying --key-class and --value-
class is unnecessary.
8 import data : Import user data from a file to a region.
9 list indexes : Display the list of indexes created for all members.
10 locate entry : Identifies the location, including host, member and region,
of entries that have the specified key.
11 put : Add/Update an entry in a region. If using a region whose
key and value classes have been set, then specifying --key-class and --
value-class is unnecessary.
12 query : Run the specified OQL query as a single quoted string and
display the results in one or more pages. Limit will default to the value
stored in the "APP_FETCH_SIZE" variable. Page size will default to the
value stored in the "APP_COLLECTION_LIMIT" variable.
13 rebalance : Rebalance partitioned regions. The default is for all
partitioned regions to be rebalanced.
14 remove : Remove an entry from a region. If using a region whose key
class has been set, then specifying --key-class is unnecessary.

```

要查看提示主题列表，请键入 `hint`。

使用 `help` 命令获取有关特定命令的信息。

根据命令，键入 `help <command-name>` 将显示该特定命令的使用信息或与该命令相关的命令列表。

例如，键入 `help start` 以显示带有简短描述的启动命令列表。说明指示命令是否可用，并取决于 `gfsh` 的连接状态。以下示例假定 `gfsh` 当前未连接（通过连接到 MX Manager 节点），因此某些启动命令不可用。

```

1 gfsh>help start
2 start gateway-receiver (Not Available)
3     Start the Gateway Receiver on a member or members.
4 start gateway-sender (Not Available)
5     Start the Gateway Sender on a member or members.
6 start jconsole (Available)
7     Start the JDK's JConsole tool in a separate process. JConsole will be
8     launched, but connecting to GemFire must be done manually.
9 start jvisualvm (Available)
10    Start the JDK's Java VisualVM (jvisualvm) tool in a separate process.
    Java
11    VisualVM will be launched, but connecting to GemFire must be done
    manually.
12 start locator (Available)
13    Start a Locator.
14 start pulse (Available)
15    Open a new window in the default web browser with the URL for the Pulse
16    application.
17 start server (Available)
18    Start a GemFire Cache Server.

```

使用 Tab 键自动完成命令或触发可能的完成。

输入部分命令以触发命令的自动完成或可能的命令完成列表后，使用 Tab 键。例如，在键入 `hint` 后按 Tab 键会显示所有可用主题：


```

1 gfsh>hint
2
3 Configuration          Data          Debug-Utility
4 Disk Store             Function Execution  GFSH
5 GemFire                Help          JMX
6 Lifecycle              Locator       Management-Monitoring
7 Manager                Region        Server
8 Statistics

```

键入 `hint d` 后按Tab键会显示以 `d` 开头的可用主题:

```

1 gfsh>hint d
2
3 data          debug-Utility  disk Store

```

自动完成还为命令提供可用参数和参数; 例如, 在键入 `start` 后点击TAB键将列出可以启动的所有服务。

```

1 gfsh>start
2
3 start data-browser  start jconsole      start jvisualvm
4 start locator      start pulse        start server

```

在 `start locator` 之后点击TAB键将在命令后填充 `--name` 参数。(如果您没有指定成员名称, `gfsh` 将自动选择一个随机名称。这对自动化非常有用。)

使用向上箭头访问命令历史记录.

通过使用向上箭头滚动以前的命令来访问shell历史记录中的命令。

使用反斜杠分隔多行命令.

输入长命令时, 可以使用反斜杠字符 (`\`) 作为分隔符来中断命令行。例如:

```

1 gfsh>create region --name=region1 \
2 --type=REPLICATE_PERSISTENT \
3 --cache-writer=org.apache.geode.examples.MyCachewriter \
4 --group=Group1 --disk-store=DiskStore1

```

用单引号或双引号括起包含空格或逗号的字符串

在 `gfsh` 命令shell中执行 `gfsh` 命令时, 请在单引号中包含任何包含空格或逗号的字符串。例如:

```

1 gfsh>start locator --name='locator 1'
2 start locator --name=locator1 --port=9009 --mcast-port=0\
3 --J='-Dgemfire.remote-locators=192.0.2.0[9009],192.0.2.1[9009]'

```

当您从操作系统shell在一行中执行多个 `gfsh` 命令时, 请将 `gfsh` 命令用双引号括起来。在双引号内, 包含任何包含带单引号的空格或逗号的字符串。例如:

```

1 $ gfsh -e "start locator --name='locator 1'" -e "start server --name=server1"

```

在命令行选项中指定JSON

{#Specifying_JSON_within_Command_Line_Options}

一些 `gfsh` 命令允许在某些选项中使用JSON规范。这些JSON字符串以键/值对的形式指定初始化属性。

对于那些实现 `Declarable` 接口的类，JSON字符串被附加到完全限定的类名。接口的 `init` 方法的参数是一个包含初始化属性的对象。解析JSON字符串并将其分解为一组定义对象的属性名称和值。

`gfsh create region` 命令中的缓存加载器类的规范可能具有JSON规范。例如，

```
1 gfsh>create region --name=region1 --type=PARTITION \  
2   --cache-loader=com.example.Setup{'k1':'v1','k2':'v2','k3':'v3'}
```

此示例中的JSON字符串是

```
1 {'k1':'v1','k2':'v2','k3':'v3'}
```

JSON字符串由大括号括起，并且用逗号分隔。

每对由冒号分隔。定义一对的两个字符串中的每一个都用单引号或双引号字符括起来。在使用单引号或双引号字符时，在整个JSON字符串中保持一致。

空格字符分隔 `gfsh` 命令行选项，并且在解析JSON字符串之前进行选项的标记化。如果命令行选项中的值包含空格，请使用双引号将其分隔。

例如，假设有一对，并且该对的值是 `value with space`。 `--cache-loader` 示例选项的右侧变为

```
1 gfsh>create region --name=region2 --type=PARTITION \  
2   --cache-loader="com.example.Setup{'k1':'value with space'}"
```

要在值中包含单引号字符，请使用两个反斜杠字符对其进行转义。例如，

```
1 gfsh>create region --name=region3 --type=PARTITION \  
2   --cache-loader="com.example.Setup{'k1':'property\\'s value'}"
```

value的值是 `property's value`。

教程 - 使用gfsh执行常见任务 {#Tutorial}

本主题将指导您在启动 `gfsh` 后执行的典型任务序列。

步骤1:创建临时工作目录并切换到该目录。例如:

```
1 $ mkdir gfsh_tutorial  
2 $ cd gfsh_tutorial
```

Step 1: 启动gfsh提示符.

```
1 $ gfsh  
2 _____  
3  / ____/ ____/ ____/ ____/ /  
4  / /  _/  _/  _/  _/  _/  _/  
5  / /_/_/  _/  _/  _/  _/  _/  
6  /____/_/  _/  _/  _/  _/  _/  
7  
8 Monitor and Manage Geode  
9 gfsh>
```

有关详细信息，请参阅[启动gfsh](#)。

Step 2: 启动定位器。 输入以下命令：

```
1 | gfsh>start locator --name=locator1
```

出现以下输出：

```
1 | gfsh>start locator --name=locator1
2 | .....
3 | Locator in /home/username/gfsh_tutorial/locator1 on 192.0.2.0[10334]
4 | as locator1 is currently online.
5 | Process ID: 67666
6 | Uptime: 6 seconds
7 | Geode Version: 1.7
8 | Java Version: 1.8.0_121
9 | Log File: /home/username/gfsh_tutorial/locator1.log
10 | JVM Arguments: -Dgemfire.enable-cluster-configuration=true
11 | -Dgemfire.load-cluster-configuration-from-dir=false
12 | -Dgemfire.launcher.registersSignalHandlers=true
13 | -Djava.awt.headless=true -
14 | Dsun.rmi.dgc.server.gcInterval=9223372036854775806
15 | Class-Path: /home/username/geode/geode-assembly/build/install/apache-
16 | geode/lib
17 | /geode-core-1.2.0.jar:/home/username/geode/geode-
18 | assembly/build/install/apache-geode
19 | /lib/geode-dependencies.jar
20 | Successfully connected to: JMX Manager [host=192.0.2.0, port=1099]
21 | Cluster configuration service is up and running.
```

如果你从gfsh运行 `start locator` 而没有指定成员名，gfsh会自动选择一个随机成员名。这对自动化很有用。

在文件系统中，检查执行 gfsh 的文件夹位置。请注意，`start locator` 命令已自动创建了一个工作目录（使用定位器的名称），并在该工作目录中创建了一个日志文件，一个状态文件和一个 `.pid`（包含定位器的进程ID）对于这个定位器。

此外，由于还没有其他JMX Manager存在，请注意 gfsh 已在定位器内的端口1099上自动启动嵌入式JMX Manager，并已将您连接到该JMX Manager。

Step 3: 检查现有的gfsh连接。

在当前shell中，键入以下命令：

```
1 | gfsh>describe connection
```

如果已连接到在步骤2中启动的定位器中启动的JMX Manager，则会显示以下输出：

```
1 | gfsh>describe connection
2 | Connection Endpoints
3 | -----
4 | ubuntu.local[1099]
```

请注意，JMX Manager位于1099，而定位器的默认端口为10334。

Step 4: 从不同的终端连接到相同的 locator/JMX Manager。

此步骤说明如何连接到 locator/JMX Manager。打开第二个终端窗口，然后启动第二个 gfsh 提示符。在第二个提示中键入与步骤3中相同的命令：

```
1 | gfsh>describe connection
```

这一次，请注意您未连接到JMX Manager，并显示以下输出：

```
1 | gfsh>describe connection
2 | Connection Endpoints
3 | -----
4 | Not connected
```

在第二个 gfsh 终端中键入以下命令：

```
1 | gfsh>connect
```

该命令将连接到您在步骤2中启动的当前运行的本地定位器。

```
1 | gfsh>connect
2 | Connecting to Locator at [host=localhost, port=10334] ..
3 | Connecting to Manager at [host=ubuntu.local, port=1099] ..
4 | Successfully connected to: [host=ubuntu.local, port=1099]
```

请注意，如果您在启动定位器时使用了自定义 --port，或者您从另一个成员的 gfsh 提示符连接时，您还需要在连接时指定 --locator=hostname[port] 到群集。例如（如果你想尝试下一个命令，首先键入 disconnect）：

```
1 | gfsh>connect --locator=localhost[10334]
2 | Connecting to Locator at [host=localhost, port=10334] ..
3 | Connecting to Manager at [host=ubuntu.local, port=1099] ..
4 | Successfully connected to: [host=ubuntu.local, port=1099]
```

将 gfsh 连接到集群的另一种方法是直接连接到定位器内运行的JMX Manager。例如（如果你想尝试下一个命令，首先键入 disconnect）：

```
1 | gfsh>connect --jmx-manager=localhost[1099]
2 | Connecting to Manager at [host=localhost, port=1099] ..
3 | Successfully connected to: [host=localhost, port=1099]
```

此外，您还可以通过HTTP连接到远程群集。请参阅[使用gfsh通过HTTP或HTTPS管理远程群集](#)。

Step 5: 断开并关闭第二个终端窗口。 键入以下命令以断开连接并退出第二个 gfsh 提示符：

```
1 | gfsh>disconnect
2 | Disconnecting from: localhost[1099]
3 | Disconnected from : localhost[1099]
4 | gfsh>exit
```

关闭第二个终端窗口。

Step 6: 启动服务器. 返回到第一个终端窗口，然后启动使用您在步骤2中启动的定位器的缓存服务器。键入以下命令：

```
1 | gfsh>start server --name=server1 --locators=localhost[10334]
```

如果服务器成功启动，则会显示以下输出：

```
1 | gfsh>start server --name=server1 --locators=localhost[10334]
2 | Starting a Geode Server in
   | /home/username/gfsh_tutorial/server1/server1.log...
3 | ...
4 | Server in /home/username/gfsh_tutorial/server1 on 192.0.2.0[40404] as
   | server1
5 | is currently online.
6 | Process ID: 68375
7 | Uptime: 4 seconds
8 | Geode Version: 1.7
9 | Java Version: 1.8.0_121
10 | Log File: /home/username/gfsh_tutorial/server1/server1.log
11 | JVM Arguments: -Dgemfire.locators=localhost[10334]
12 | -Dgemfire.use-cluster-configuration=true -Dgemfire.start-dev-rest-
   | api=false
13 | -XX:OnOutOfMemoryError=kill -KILL %p
14 | -Dgemfire.launcher.registerSignalHandlers=true
15 | -Djava.awt.headless=true -
   | Dsun.rmi.dgc.server.gcInterval=9223372036854775806
16 | Class-Path: /home/username/geode/geode-assembly/build/install/apache-
   | geode/lib
17 | /geode-core-1.2.0.jar:/home/username/geode/geode-assembly/build/install
18 | /apache-geode/lib/geode-dependencies.jar
```

如果从gfsh运行 `start server` 而不指定成员名称，gfsh将自动选择一个随机成员名称。这对自动化很有用。

在文件系统中，检查执行gfsh的文件夹位置。请注意，就像 `start locator` 命令一样，`start server` 命令已经自动创建了一个工作目录（以服务器命名），并且在该工作目录中，它创建了一个日志文件和一个 `.pid`（包含服务器的进程ID）此缓存服务器。此外，它还编写了日志文件。

Step 7: 列出成员. 使用 `list members` 命令查看刚刚创建的集群的当前成员。

```
1 | gfsh>list members
2 |   Name      | Id
3 | ----- | -----
4 | Coordinator: | ubuntu(locator1:5610:locator)<ec><v0>:34168
5 | locator1    | ubuntu(locator1:5610:locator)<ec><v0>:34168
6 | server1     | ubuntu(server1:5931)<v1>:35285
```

Step 8: 通过执行 `describe member` 命令查看成员详细信息.

```
1 | gfsh>describe member --name=server1
2 | Name      : server1
3 | Id        : ubuntu(server1:5931)<v1>:35285
4 | Host      : ubuntu.local
5 | Regions   :
6 | PID       : 5931
```

```

7 Groups      :
8 Used Heap   : 12M
9 Max Heap    : 239M
10 Working Dir : /home/username/gfsh_tutorial/server1
11 Log file    : /home/username/gfsh_tutorial/server1/server1.log
12 Locators    : localhost[10334]
13
14 Cache Server Information
15 Server Bind      :
16 Server Port      : 40404
17 Running          : true
18 Client Connections : 0

```

请注意，尚未为此成员分配任何区域。

Step 9: 创建您的第一个区域。 键入以下命令，然后键入Tab键:

```

1 gfsh>create region --name=region1 --type=

```

将显示可能的区域类型列表，后跟您输入的部分命令:

```

1 gfsh>create region --name=region1 --type=
2
3 PARTITION
4 PARTITION_REDUNDANT
5 PARTITION_PERSISTENT
6 PARTITION_REDUNDANT_PERSISTENT
7 PARTITION_OVERFLOW
8 PARTITION_REDUNDANT_OVERFLOW
9 PARTITION_PERSISTENT_OVERFLOW
10 PARTITION_REDUNDANT_PERSISTENT_OVERFLOW
11 PARTITION_HEAP_LRU
12 PARTITION_REDUNDANT_HEAP_LRU
13 REPLICATE
14 REPLICATE_PERSISTENT
15 REPLICATE_OVERFLOW
16 REPLICATE_PERSISTENT_OVERFLOW
17 REPLICATE_HEAP_LRU
18 LOCAL
19 LOCAL_PERSISTENT
20 LOCAL_HEAP_LRU
21 LOCAL_OVERFLOW
22 LOCAL_PERSISTENT_OVERFLOW
23 PARTITION_PROXY
24 PARTITION_PROXY_REDUNDANT
25 REPLICATE_PROXY
26
27 gfsh>create region --name=region1 --type=

```

使用您要创建的区域类型完成命令。 例如，创建一个本地区域:

```

1 gfsh>create region --name=region1 --type=LOCAL
2 Member | Status
3 -----|-----
4 server1 | Region "/region1" created on "server1"

```

由于此时集群中只有一台服务器，因此该命令会在server1上创建本地区域。

Step 10: 启动其他服务器。 这次使用不同的服务器端口指定 `--server-port` 参数，因为您在同一主机上启动了缓存服务器进程。

```
1 gfsh>start server --name=server2 --server-port=40405
2 Starting a Geode Server in /home/username/gfsh_tutorial/server2...
3 ...
4 Server in /home/username/gfsh_tutorial/server2 on 192.0.2.0[40405] as
5 server2 is currently online.
6 Process ID: 68423
7 Uptime: 4 seconds
8 Geode Version: 1.7
9 Java Version: 1.8.0_121
10 Log File: /home/username/gfsh_tutorial/server2/server2.log
11 JVM Arguments: -Dgemfire.default.locators=192.0.2.0[10334]
12 -Dgemfire.use-cluster-configuration=true -Dgemfire.start-dev-rest-
13 api=false
14 -XX:OnOutOfMemoryError=kill -KILL %p -
15 Dgemfire.launcher.registersSignalHandlers=true
16 -Djava.awt.headless=true -
17 Dsun.rmi.dgc.server.gcInterval=9223372036854775806
18 Class-Path: /home/username/geode/geode-assembly/build/install/apache-geode
19 /lib/geode-core-1.2.0.jar:/home/username/geode/geode-assembly/build/install
20 /apache-geode/lib/geode-dependencies.jar
```

Step 11: 创建复制区域。

```
1 gfsh>create region --name=region2 --type=REPLICATE
2 Member | Status
3 -----|-----
4 server1 | Region "/region2" created on "server1"
5 server2 | Region "/region2" created on "server2"
```

Step 12: 创建分区区域。

```
1 gfsh>create region --name=region3 --type=PARTITION
2 Member | Status
3 -----|-----
4 server1 | Region "/region3" created on "server1"
5 server2 | Region "/region3" created on "server2"
```

Step 13: 创建一个复制的持久区域。

```
1 gfsh>create region --name=region4 --type=REPLICATE_PERSISTENT
2 Member | Status
3 -----|-----
4 server1 | Region "/region4" created on "server1"
5 server2 | Region "/region4" created on "server2"
```

Step 14: 列出区域。 将显示您刚刚创建的所有区域的列表。


```
1 gfsh>list regions
2 List of regions
3 -----
4 region1
5 region2
6 region3
7 region4
```

Step 15: 通过执行describe member命令再次查看成员详细信息.

```
1 gfsh>describe member --name=server1
2 Name      : server1
3 Id        : ubuntu(server1:5931)<v1>:35285
4 Host      : ubuntu.local
5 Regions   : region4
6            region3
7            region2
8            region1
9 PID       : 5931
10 Groups    :
11 Used Heap : 14M
12 Max Heap  : 239M
13 Working Dir : /home/username/gfsh_tutorial/server1
14 Log file   : /home/username/gfsh_tutorial/server1/server1.log
15 Locators   : localhost[10334]
16
17 Cache Server Information
18 Server Bind      :
19 Server Port      : 40404
20 Running          : true
21 Client Connections : 0
```

请注意，您现在创建的所有区域都显示在成员说明的“Regions”部分中。

```
1 gfsh>describe member --name=server2
2 Name      : server2
3 Id        : ubuntu(server2:6092)<v2>:17443
4 Host      : ubuntu.local
5 Regions   : region4
6            region3
7            region2
8            region1
9 PID       : 6092
10 Groups    :
11 Used Heap : 14M
12 Max Heap  : 239M
13 Working Dir : /home/username/gfsh_tutorial/server2
14 Log file   : /home/username/gfsh_tutorial/server2/server2.log
15 Locators   : 192.0.2.0[10334]
16
17 Cache Server Information
18 Server Bind      :
19 Server Port      : 40405
20 Running          : true
21 Client Connections : 0
```

请注意，即使您在创建第一个区域（region1）后启动了第二个服务器，第二个服务器仍会列出 region1，因为它从群集配置服务中选择了其配置。

Step 16: 将数据放进本地区域. 输入以下put命令:

```
1 gfsh>put --key=('123') --value=('ABC') --region=region1
2 Result      : true
3 Key Class   : java.lang.String
4 Key         : ('123')
5 Value Class : java.lang.String
6 Old Value   : <NULL>
```

Step 17: 将数据放在复制区域中. 输入以下put命令:

```
1 gfsh>put --key=('123abc') --value=('Hello world!!') --region=region2
2 Result      : true
3 Key Class   : java.lang.String
4 Key         : ('123abc')
5 Value Class : java.lang.String
6 Old Value   : <NULL>
```

Step 18: 检索数据. 您可以使用 locate entry, query 或 get 来返回刚刚放入该区域的数据。

例如，使用 get 命令:

```
1 gfsh>get --key=('123') --region=region1
2 Result      : true
3 Key Class   : java.lang.String
4 Key         : ('123')
5 Value Class : java.lang.String
6 Value       : ('ABC')
```

例如，使用 locate entry 命令:

```
1 gfsh>locate entry --key=('123abc') --region=region2
2 Result      : true
3 Key Class   : java.lang.String
4 Key         : ('123abc')
5 Locations Found : 2
6
7
8 MemberName | MemberId
9 ----- | -----
10 server2    | ubuntu(server2:6092)<v2>:17443
11 server1    | ubuntu(server1:5931)<v1>:35285
```

请注意，由于条目已放入复制区域，因此该条目位于两个集群成员上。

例如，使用 query 命令:

```

1  gfsh>query --query='SELECT * FROM /region2'
2
3  Result      : true
4  startCount  : 0
5  endCount    : 20
6  Rows        : 1
7
8  Result
9  -----
10 ('Hello world!!')
11
12 NEXT_STEP_NAME : END

```

Step 19: 导出您的数据. 要保存区域数据，可以使用 `export data` 命令。

例如:

```

1  gfsh>export data --region=region1 --file=region1.gfd --member=server1

```

您可以稍后使用 `import data` 命令将该数据导入另一个成员的同区域。

Step 20: 关闭群集.

```

1  gfsh>shutdown --include-locators=true

```

按功能区快速参考gfsh命令

{#Quick_Reference_of_gfsh_Commands_by_Functional_Area}

此快速参考将所有命令分类到功能区域。

单击命令可查看其他信息，包括语法，选项列表和示例。

- **Basic Geode gfsh Commands**
- **Configuration Commands**
- **Data Commands**
- **Deployment Commands**
- **Disk Store Commands**
- **Durable CQ and Client Commands**
- **Function Execution Commands**
- **Gateway (WAN) Commands**
- **Geode Monitoring Commands**
- **Index Commands**
- **JMX Connection Commands**
- **Locator Commands**
- **Lucene Commands**
- **PDX Commands**
- **Region Commands**
- **Server Commands**

基本Geode gfsh命令 {#Basic_Geode_gfsh_Commands}

命令	描述	可用性
debug	在 gfsh 中启用或禁用调试输出.	online, offline
echo	回显给定文本，其中可能包括系统和用户变量.	online, offline
exit	退出gfsh shell。您也可以使用 <code>quit</code> 退出shell.	online, offline
help	如果参数是gfsh命令，则显示该命令的语法和用法信息。如果没有参数，则显示所有可用命令的列表.	online, offline
hint	显示有关主题的信息以及与主题关联的命令列表.	online, offline
history	显示或保存命令历史记录.	online, offline
run	执行一组GFSH命令.	online, offline
set variable	在GFSH环境中设置变量.	online, offline
sh	执行操作系统（OS）命令.	offline, online
sleep	延迟 gfsh 命令执行.	online, offline
version	显示产品版本信息.	online, offline

配置命令 {#Configuration_Commands}

命令	描述	可用性
alter runtime	在一个或多个成员运行时更改特定成员的配置属性.	online
change loglevel	更改指定服务器上的日志记录级别.	online
configure pdx	此命令可更改所有高速缓存的群集范围的PDX配置设置.	online 注意: 必须在启动数据成员之前运行此命令才能强制执行配置设置.
describe config	显示成员的配置.	online
export config	导出配置, 数据, 日志和堆栈跟踪.	online
export cluster-configuration	导出共享配置ZIP文件, 其中包含配置和操作集群所需的cache.xml文件, gemfire.properties文件和JAR文件.	online
import cluster-configuration	导入导出的配置.	online
status cluster-config-service	报告群集配置服务器的状态.	online

数据命令 {#Data_Commands}

命令	描述	可用性
export data	将用户数据从区域导出到文件.	online
get	显示区域中的条目.	online
import data	将用户数据从文件导入区域.	online
locate entry	在成员上找到区域条目.	online
put	添加或更新区域条目.	online
query	对Geode区域运行查询.	online
remove	从区域中删除条目.	online

部署命令 {#Deployment_Commands}

命令	描述	可用性
deploy	将JAR打包的应用程序部署到一个或多个成员.	online
list deployed	显示使用 dep1oy 命令部署到成员的JAR列表.	online
undeploy	取消部署部署使用 dep1oy 命令部署在成员或群体的JAR文件.	online

磁盘存储命令 {#Disk_Store_Commands}

命令	描述	可用性
alter disk-store	修改现有的Geode资源.	online
backup disk-store	将所有成员的持久数据备份到指定目录.	online
compact disk-store	压缩在线磁盘存储.	online
compact offline-disk-store	压缩脱机磁盘存储.	online, offline
create disk-store	定义一个或多个磁盘存储的池, 可供区域和客户端订阅队列使用.	online
describe disk-store	显示有关成员磁盘存储的信息.	online
describe offline-disk-store	显示有关脱机成员磁盘存储的信息	online, offline
destroy disk-store	删除磁盘存储以及磁盘存储使用的磁盘上的所有文件。 先前使用此磁盘存储的封闭区域的数据将丢失.	online
list disk-stores	列出Geode集群中的所有可用磁盘存储.	online
revoke missing-disk-store	指示集群成员停止等待磁盘存储可用.	online
show missing-disk-stores	显示群集中当前缺少的磁盘存储的摘要.	online
validate offline-disk-store	验证脱机磁盘存储.	online, offline

持久的CQ和客户端命令 {#Durable_CQ_and_Client_Commands}

命令	描述	可用性
list durable-cqs	列出与指定的持久客户端ID关联的持久客户端CQ.	online
close durable-cq	关闭持久客户端注册的持久CQ, 并从订阅队列中为持久CQ保留排除事件.	online
close durable-client	试图关闭持久客户端。 客户端必须断开连接.	online
show subscription-queue-size	显示订阅队列中的事件数。 如果提供CQ名称, 则它计算指定CQ的订阅队列中的事件数.	online

函数执行命令 {#Function_Execution_Commands}

命令	描述	可用性
destroy. function	Destroy or unregister a function. The default is for the function to be unregistered from all members.	online
execute function	Execute the function with the specified ID. By default, executes on all members.	online
list functions	Display a list of registered functions. The default is to display functions for all members.	online

网关(WAN)命令 {#GatewayWANCommands}

命令	描述	可用性
create async-event-queue	创建异步事件队列.	online
create gateway-receiver	在一个或多个成员上创建网关接收器.	online
create gateway-sender	在一个或多个成员上创建网关发件人.	online
destroy gateway-sender	销毁一个或多个成员上的网关发件人.	online
list async-event-queues	显示所有成员的异步事件队列列表.	online
list gateways	显示一个或多个成员的网关发件人和收件人.	online
load-balance gateway-sender	使指定的网关发送方关闭其当前连接并以更平衡的方式重新连接到远程网关接收方.	online
pause gateway-sender	暂停网关发件人.	online
resume gateway-sender	恢复已暂停的所有网关发件人.	online
start gateway-receiver	在给定的成员或成员组上启动网关接收器.	online
start gateway-sender	在一个或多个成员上启动网关发件人.	online
status gateway-receiver	显示指定网关接收器的状态.	online
status gateway-sender	显示指定网关发件人的状态.	online
stop gateway-receiver	停止一个或多个成员的网关接收器.	online
stop gateway-sender	在指定成员或指定成员组成员上具有给定ID的网关发件人.	online

Geode 异步事件队列命令 {#Geode_AsyncEventQueue_Commands}

命令	描述	可用性
create async-event-queue	创建异步事件队列.	online
list async-event-queues	显示所有成员的异步事件队列列表.	online

Geode 监控命令 {#Geode_Monitoring_Commands}

命令	描述	可用性
describe client	显示指定客户端的详细信息.	online
describe member	显示具有给定名称/ID的成员的详细信息.	online
export logs	将日志导出/转储到给定目录.	online
export stack-traces	导出一个或多个成员的堆栈跟踪.	online
gc	强制成员或成员的垃圾回收.	online
list clients	显示已连接客户端的列表.	online
list members	显示全部或部分成员.	online
netstat	通过“netstat”操作系统命令报告网络信息和统计信息.	online
show dead-locks	显示死锁, 日志, 指标和丢失的磁盘存储.	online
show log	显示成员的日志.	online
show metrics	显示或导出整个群集, 成员或区域的度量标准.	online
shutdown	关闭所有具有缓存的成员.	online
start jconsole	在单独的进程中启动JDK JConsole监视应用程序。JConsole会自动连接到正在运行的JMX Manager节点（如果有）	online, offline
start jvisualvm	在单独的进程中启动JDK的Java VisualVM监视应用程序.	online, offline
start pulse	在用户的默认系统浏览器中启动Geode Pulse监控仪表板工具.	online, offline

索引命令 {#Index_Commands}

命令	描述	可用性
clear defined indexes	清除所有已定义的索引.	online, offline
create defined indexes	创建所有已定义的索引.	online
create index	创建可在执行查询时使用的索引.	online
define index	定义可在执行查询时使用的索引。然后，您可以一次创建多个索引.	online, offline
destroy index	销毁或删除指定的索引.	online
list indexes	显示为所有成员创建的索引列表.	online

JMX连接命令 {#JMX_Connection_Commands}

命令	描述	可用性
connect	直接或通过定位器连接到jmx-manager.	offline
describe connection	显示连接信息详细信息.	online, offline
disconnect	关闭所有活动连接.	online

定位器命令 {#Locator_Commands}

命令	描述	可用性
start locator	启动定位器。该命令创建一个以定位符命名的子目录和日志文件。如果定位器检测到不存在其他JMX Manager，则定位器将自动启动嵌入式JMX Manager并将当前的 <code>gfsh</code> 会话连接到JMX Manager.	online, offline
status locator	显示指定定位器的状态。	online, offline
stop locator	停止定位器.	online, offline

Lucene 命令 {#Lucene_Commands}

命令	描述	可用性
create lucene index	创建Lucene索引.	online
describe lucene index	描述一个Lucene索引.	online
destroy lucene index	销毁Lucene索引.	online
list lucene indexes	列出为所有成员创建的Lucene索引。 可选的 <code>--with-stats</code> 限定符显示索引上的活动.	online
search lucene	搜索Lucene索引.	online

PDX 命令 {#PDX_Commands}

命令	描述	可用性
configure pdx	为群集中的所有缓存配置Portable Data eXchange.	online, offline
pdx rename	在脱机磁盘存储中重命名PDX类型.	online, offline

区域 命令 {#Region_Commands}

命令	描述	可用性
alter region	改变区域的配置.	online
create region	创建和配置区域.	online
describe region	显示区域的属性和关键信息.	online
destroy region	销毁或删除某个地区.	online
list regions	显示成员或成员的区域。 如果未指定参数，则列出群集中的所有区域.	online
rebalance	重新平衡分区区域.	online

服务器命令 {#Server_Commands}

命令	描述	可用性
start server	启动Geode缓存服务器进程.	online, offline
status server	显示指定的Geode缓存服务器的状态.	online, offline
stop server	停止Geode缓存服务器.	online, offline

gfsh命令帮助 {#gfsh_Command_Help}

本节提供按字母顺序列出的所有 `gfsh` 命令的帮助和用法信息。

- **alter**
修改现有的Geode资源。
- **backup disk-store**
将所有成员的持久数据备份到指定目录。
- **change loglevel**
更改指定成员的日志记录级别。
- **clear defined indexes**
清除所有已定义的索引。
- **close**
关闭持久客户CQ和持久客户。
- **compact**
压缩在线和离线磁盘存储。
- **configure**
为群集中的所有缓存配置Portable Data eXchange。
- **connect**
直接或通过定位器连接到jmx-manager。
- **create**
创建异步事件队列，磁盘存储，网关接收器，网关发件人，索引和区域。
- **debug**
在 `gfsh` 中启用或禁用调试输出。
- **define index**
定义可在执行查询时使用的索引。然后，您可以使用 `create defined indexes` 执行单个命令以一次创建多个索引。
- **deploy**
将JAR打包的应用程序部署到一个或多个成员。
- **describe**
显示成员配置，shell连接，磁盘存储，成员或区域的详细信息。
- **destroy**
删除或取消注册功能，删除索引，磁盘存储和区域。
- **disconnect**
关闭所有活动连接。
- **echo**
回显给定文本，其中可能包括系统和用户变量。
- **execute function**
在成员或区域上执行功能。
- **exit**
退出 `gfsh` shell。您也可以使用 `quit` 退出shell。

- **export**
导出配置，数据，日志和堆栈跟踪。
- **gc**
对一个或多个成员强制GC（垃圾收集）。
- **get**
显示区域中的条目。
- **help**
显示所有可用命令的语法和用法信息。
- **hint**
显示有关主题的信息以及与主题关联的命令列表。
- **history**
显示或保存命令历史记录。
- **import**
您可以将数据导入区域或将现有群集配置导入群集。
- **list**
列出现有的Geode资源，例如已部署的应用程序，磁盘存储，功能，成员，服务器和区域。
- **load-balance gateway-sender**
使指定的网关发送方关闭其当前连接并以更平衡的方式重新连接到远程网关接收方。
- **locate entry**
在成员上找到区域条目。
- **netstat**
通过“netstat”操作系统命令报告网络信息和统计信息。
- **pause gateway-sender**
暂停网关发件人。
- **pdx rename**
在脱机磁盘存储中重命名PDX类型。
- **put**
添加或更新区域条目。
- **query**
对Geode区域运行查询。
- **rebalance**
重新平衡分区区域。
- **remove**
从区域中删除条目。
- **resume gateway-sender**
恢复已暂停的所有网关发件人。
- **revoke missing-disk-store**
指示群集的成员停止等待磁盘存储可用。
- **run**

执行一组GFSH命令。

- **set variable**

在GFSH环境中设置变量。

- **sh**

执行操作系统命令。

- **show**

显示死锁，日志，指标和丢失的磁盘存储。

- **shutdown**

停止所有成员。

- **sleep**

延迟 `gfsh` 命令执行。

- **start**

启动服务器，定位器，网关发送器和网关接收器以及监视工具。

- **status**

检查群集配置服务和Geode成员进程的状态，包括定位器，网关接收器，网关发件人和服务器。

- **stop**

停止网关接收器，网关发送器，定位器和服务器。

- **undeploy**

取消使用 `deploy` 命令部署在成员或组上部署的JAR文件。

- **validate offline-disk-store**

验证脱机磁盘存储。

- **version**

显示产品版本信息。

alter {#alter}

Modify an existing Geode resource.

- **alter async-event-queue**

修改异步事件队列的属性

- **alter disk-store**

修改或删除脱机磁盘存储中的区域。

- **alter region**

改变区域的配置。

- **alter runtime**

在成员或成员运行时更改所有成员或成员子集的配置属性。

alter async-event-queue {#alter_async_event_queue}

更改指定的异步事件队列的属性。必须重新启动托管指定异步事件队列的每个服务器，新属性设置才能在该服务器上生效。

可用性: Online. 必须在`gfsh`中连接到JMX Manager成员才能使用此命令。

句法:

```
1 alter async-event-queue --id=value [--batch-size=value] [--batch-time-  
  interval=value]  
2     [--max-queue-memory=value] [--if-exists(=value)]
```

必需的选项 `--id` 标识要更改的异步事件队列。

Parameters, alter async-event-queue

名称	描述
<code>--id</code>	<i>Required.</i> 要更改的异步事件队列的ID
<code>--batch-size</code>	批处理可以包含的最大事件数
<code>--batch-time-interval</code>	在批次交付之前可以经过的最长时间（以毫秒为单位）
<code>--max-queue-memory</code>	在溢出到磁盘之前队列可以消耗的最大内存量（以兆字节为单位）
<code>--if-exists</code>	如果指定的异步事件队列不存在，gfsh将响应该消息。如果此参数为true，则响应的前缀为"Skipping:"标签。对脚本测试很有用。默认值（如果未指定参数）:false。默认值（如果指定的参数没有值）:true。

示例命令:

```
1 alter async-event-queue --id=myAsyncEventQueue --batch-size=50 --if-exists
```

alter disk-store {#alter_disk_store}

修改或删除脱机磁盘存储中的区域。

修改区域的配置时，通常会使该区域脱机并使用新配置重新启动。您可以使用 `alter disk-store` 命令更改磁盘存储中存储的区域的配置，以匹配您在重新启动时使用的配置。

可用性: Offline.

句法:

```
1 alter disk-store --name=value --region=value --disk-dirs=value(,value)*  
2     [--compressor(=value)] [--concurrency-level=value]  
3     [--enable-statistics=value] [--initial-capacity=value] [--load-  
  factor=value]  
4     [--lru-algorithm=value] [--lru-action=value] [--lru-limit=value]  
5     [--off-heap(=value)] [--remove(=value)]
```

三个必需的选项 `--name`，`--region` 和 `--disk-dirs` 标识要更改的磁盘存储区和区域。如果未指定其他选项，`gfsh` 将显示当前配置而不进行任何更改。

Parameters, alter disk-store

名称	描述
--name	<i>Required.</i> 将更改其内容的磁盘存储的名称。
--region	<i>Required.</i> 使用磁盘存储区域的名称（包括路径）。
--disk-dirs	<i>Required.</i> 先前已写入磁盘存储数据的目录。
--compressor	压缩区域条目值时要使用的压缩程序的完全限定类名。值为 <code>none</code> 会删除压缩器。
--concurrency-level	估计将同时访问区域条目的最大应用程序线程数。与 <code>--initial-capacity</code> 和 <code>--load-factor</code> 一起，在用于存储区域条目的底层 <code>java.util.ConcurrentHashMap</code> 上设置参数。此属性不适用于分区区域。
--enable-statistics	启用 <code>--region</code> 选项指定的区域的统计信息。有效值为 <code>true</code> 或 <code>false</code> 。如果指定的参数没有值，则使用 <code>true</code> 值。
--initial-capacity	与 <code>--concurrency-level</code> 和 <code>--load-factor</code> 一起，在用于存储区域条目的底层 <code>java.util.ConcurrentHashMap</code> 上设置参数。
--load-factor	与 <code>--concurrency-level</code> 和 <code>--initial-capacity</code> 一起，在用于存储区域条目的底层 <code>java.util.ConcurrentHashMap</code> 上设置参数。这必须是0到1之间的浮点数，包括0和1。
--lru-action	驱逐该地区的条目时要采取的行动。有效值为: <code>none</code> , <code>overflow-to-disk</code> , <code>local-destroy</code>
--lru-algorithm	最近最少使用逐出算法。有效类型是: <code>none</code> , <code>lru-entry-count</code> , <code>lru-heap-percentage</code> , <code>lru-memory-size</code>
--lru-limit	驱逐前该地区允许的条目数量。
--off-heap	指定区域值是在堆内存还是堆外内存中。如果为 <code>true</code> ，则区域值位于堆外内存中。如果指定的参数没有值，则使用 <code>true</code> 值。
--remove	指定是否从磁盘存储中删除该区域。如果指定的参数没有值，则使用 <code>true</code> 值。 注意: <code>--remove</code> 删除该区域的所有持久数据。如果您可能希望在以后检索数据，请考虑在使用此选项之前将磁盘存储文件复制到备份。

示例命令:

```

1 alter disk-store --name=DiskStore1 --region=region1 --disk-
  dirs=/Disks/DiskStore1 --off-heap
2 alter disk-store --name=DiskStore1 --region=region1 --disk-
  dirs=/Disks/DiskStore1 --remove

```

alter region {#alter_region}

改变区域的配置。

有关语法详细信息，请参阅[在命令行选项中指定JSON](#)。

可用性: Online. 必须在gfsh中连接到JMX Manager成员才能使用此命令。

句法:

```
1 alter region --name=value [--groups=value(,value)*]
2   [--entry-idle-time-expiration=value]
3   [--entry-idle-time-expiration-action(=value)?]
4   [--entry-time-to-live-expiration=value]
5   [--entry-time-to-live-expiration-action(=value)?]
6   [--entry-idle-time-custom-expiry=value] [--entry-time-to-live-custom-
  expiry=value]
7   [--region-idle-time-expiration=value]
8   [--region-idle-time-expiration-action(=value)?]
9   [--region-time-to-live-expiration=value]
10  [--region-time-to-live-expiration-action(=value)?]
11  [--cache-listener=value(,value)*] [--cache-loader=value]
12  [--cache-writer=value] [--async-event-queue-id=value(,value)*]
13  [--gateway-sender-id=value(,value)*] [--enable-cloning(=value)?]
14  [--eviction-max(=value)?]
```

Parameters, alter region

名称	描述	默认值
--async-event-queue-id	将用于后写操作的异步事件队列的ID。	
--cache-listener	要实例化的插件的完全限定类名，用于接收对区域及其条目的更改的事件后通知。可以配置任意数量的缓存侦听器。完全限定的类名可以附加一个JSON规范，该规范将被解析成为实现 <code>Declarable</code> 接口的类的 <code>init()</code> 方法的参数字段。	
--cache-loader	要实例化的插件的完全限定类名，用于接收区域中缓存未命中的通知。最多可以在该区域的每个成员中定义一个缓存加载器。对于分布式区域，可以从具有定义区域的其他成员远程调用缓存加载器。完全限定的类名可以附加一个JSON规范，该规范将被解析成为实现 <code>Declarable</code> 接口的类的 <code>initialize()</code> 方法的参数字段。	

名称	描述	默认值
--cache-writer	要实例化的插件的完全限定类名，用于接收区域及其条目更改的事件前通知。插件可能会取消该事件。最多可以在该区域的每个成员中定义一个缓存写入器。完全限定的类名可以附加一个JSON规范，该规范将被解析为实现 <code>Declarable</code> 接口的类的 <code>init()</code> 方法的参数字段。	
--enable-cloning	确定 <code>fromDelta</code> 如何将增量应用于本地缓存以进行增量传播。如果为 <code>true</code> ，则将更新应用于值的克隆，然后将克隆保存到缓存中。如果为 <code>false</code> ，则在缓存中就地修改该值。	<code>false</code>
--entry-idle-time-expiration	区域或条目到期之前的秒数。指定 <code>-1</code> 表示此类型没有到期	<code>-1</code>
--entry-idle-time-expiration-action	应在区域或条目到期时发生的操作。选择以下过期操作之一： <code>local-destroy</code> ，从本地缓存中删除区域或条目，但不将删除操作分发给远程成员。您不能对分区区域条目使用此操作。 <code>destroy</code> ，从缓存中完全删除区域或条目。根据区域的分布设置分发销毁操作。当群集中的任何应用程序不再需要区域或条目时，请使用此选项。使默认到期操作无效。将区域中的条目或所有条目标记为无效。根据区域范围分配失效。当区域或条目不再对群集中的任何应用程序有效时，这是正确的选择。 <code>local-invalidate</code> 将区域中的条目或所有条目标记为无效，但不分发操作。您不能对分区区域条目使用此操作。仅对未配置为复制区域的区域支持本地区域失效。	<code>invalidate</code>
--entry-time-to-live-expiration	在区域或项过期前的秒数。指定 <code>-1</code> 表示该类型没有过期。	<code>-1</code>
--entry-time-to-live-expiration-action	应在区域或条目到期时执行的操作。选择以下过期操作之一： <code>local-destroy</code> 从本地缓存中删除区域或条目，但不将删除操作分发给远程成员。您不能对分区区域条目使用此操作。 <code>destroy</code> 从缓存中完全删除区域或条目。根据区域的分布设置分发销毁操作。当群集中的任何应用程序不再需要区域或条目时，请使用此选项。 <code>invalidate</code> 默认到期操作。将区域中的条目或所有条目标记为无效。根据区域范围分配失效。当区域或条目不再对群集中的任何应用程序有效时，这是正确的选择。 <code>local-invalidate</code> 将区域中的条目或所有条目标记为无效，但不分发操作。您不能对分区区域条目使用此操作。仅对未配置为复制区域的区域支持本地区域失效。	<code>invalidate</code>
--entry-idle-time-custom-expiry	为入口空闲时间实现 <code>CustomExpiry</code> 的类的名称。为初始化属性附加JSON字符串。	

名称	描述	默认值
--entry-time-to-live-custom-expiry	实现CustomExpiry以进入生存时间的类的名称。 为初始化属性附加JSON字符串。	
--eviction-max	驱逐算法用于确定何时执行其驱逐操作的驱逐属性的最大值。 最大值的单位由驱逐算法确定。	0
-gateway-sender-id	数据路由的网关发件人的ID。	
-groups	该地区将被改变的成员组。	
-name	Required. 该地区的名称（包括路径）。	
-region-idle-time-expiration	区域或条目到期之前的秒数。 如果未指定超时，则默认为零（这意味着没有到期）。	-1
-region-idle-time-expiration-action	应在区域或条目到期时执行的操作。选择以下过期操作之一： <code>local-destroy</code> 从本地缓存中删除区域或条目，但不将删除操作分发给远程成员。您不能对分区区域条目使用此操作。 <code>destroy</code> 从缓存中完全删除区域或条目。根据区域的分布设置分发销毁操作。当群集中的任何应用程序不再需要区域或条目时，请使用此选项。 <code>invalidate</code> 默认到期操作。将区域中的条目或所有条目标记为无效。根据区域范围分配失效。当区域或条目不再对群集中的任何应用程序有效时，这是正确的选择。 <code>local-invalidate</code> 将区域中的条目或所有条目标记为无效，但不分发操作。您不能对分区区域条目使用此操作。仅对未配置为复制区域的区域支持本地区域失效。	<code>invalidate</code>
-region-time-to-live-expiration	区域或条目到期之前的秒数。 如果未指定超时，则默认为零（这意味着没有到期）。	-1
-region-time-to-live-expiration-action	应在区域或条目到期时执行的操作。选择以下过期操作之一： <code>local-destroy</code> 从本地缓存中删除区域或条目，但不将删除操作分发给远程成员。您不能对分区区域条目使用此操作。 <code>destroy</code> 从缓存中完全删除区域或条目。根据区域的分布设置分发销毁操作。当群集中的任何应用程序不再需要区域或条目时，请使用此选项。 <code>invalidate</code> 默认到期操作。将区域中的条目或所有条目标记为无效。根据区域范围分配失效。当区域或条目不再对群集中的任何应用程序有效时，这是正确的选择。 <code>local-invalidate</code> 将区域中的条目或所有条目标记为无效，但不分发操作。您不能对分区区域条目使用此操作。仅对未配置为复制区域的区域支持本地区域失效。	<code>invalidate</code>

示例命令:

```
1 | alter region --name=region1 --eviction-max=5000 [-group=all]
```

样本输出:

```
1 | gfsh>alter region --name=customer --eviction-max=5000
2 | Member | Status
3 | ----- | -----
4 | server1 | Region "/customer" altered on "server1"
```

alter runtime {#alter_runtime}

在成员或成员运行时更改所有成员或成员子集的配置属性。

有关这些配置属性的更多信息，请参阅[cache.xml](#) 和配置参数参考。

可用性: Online. 您必须在 gfsh 中连接到 MX Manager 成员才能使用此命令。

句法:

```
1 | alter runtime [--members=value(,value)*] [--groups=value(,value)*]
2 |     [--archive-disk-space-limit=value]
3 |     [--archive-file-size-limit=value] [--log-disk-space-limit=value]
4 |     [--log-file-size-limit=value] [--log-level=value]
5 |     [--statistic-archive-file=value] [--statistic-sample-rate=value]
6 |     [--enable-statistics=value] [--copy-on-read(=value)?] [--lock-
7 | lease=value]
   |     [--lock-timeout=value] [--message-sync-interval=value] [--search-
   | timeout=value]
```

Parameters, alter runtime

名称	描述	默认值
-members	要在运行时更改其配置的成员的名称或ID。如果未指定此参数，则会使用群集配置服务为所有群集成员修改配置属性。	如果未指定，则使用群集配置服务的所有成员
-groups	要更改其成员的运行时配置的组的名称。如果未指定此参数，则会使用群集配置服务为所有群集成员修改配置属性。	如果未指定，则使用群集配置服务的所有成员
-archive-disk-space-limit	归档磁盘空间限制。组合的所有非活动统计归档文件的最大大小（以兆字节为单位）。如果超出此限制，则会删除非活动归档文件，最先删除，直到总大小在限制范围内。如果设置为零，则磁盘空间使用不受限制。有效值为（以兆字节为单位）:0 - 1000000。	0
-archive-file-size-limit	存档文件大小限制。单个统计存档文件的最大大小（以兆字节为单位）。超过此限制后，将创建新的统计存档文件，并且当前存档文件将变为非活动状态。如果设置为零，则文件大小不受限制。有效值为（以兆字节为单位）:0 - 1000000。	0
-log-disk-space-limit	记录磁盘空间限制。所有非活动日志文件的最大大小（以兆字节为单位）。如果超出此限制，则会删除非活动日志文件，这是最早的，直到总大小在限制范围内。如果设置为零，则磁盘空间使用不受限制。有效值为（以兆字节为单位）:0 - 1000000。	0
-log-file-size-limit	日志文件大小限制。日志文件关闭之前的最大大小（兆字节），并且日志记录将滚动到新的（子）日志文件。如果设置为零，则禁用日志滚动。有效值为（以兆字节为单位）:0 - 1000000。	0
-loglevel	新的日志级别。此选项是必需的，您必须指定一个值。有效值包括:ALL, TRACE, DEBUG, INFO, WARN, ERROR, FATAL, OFF。	INFO
-statistic-archive-file	正在运行的系统成员写入统计样本的文件。例如:“StatisticsArchiveFile.gfs”。必须定义为将归档存储到文件。将.gz后缀添加到文件名会导致它被压缩。请参阅 统计 。	not set
-statistic-sample-rate	统计抽样率。有效值为（以毫秒为单位）:100 - 60000.请参阅 统计 。	1000
-enable-statistics	是否应启用统计抽样。指定--statistic-archive-file将统计信息存储到文件中。有效值为:true和false。请参阅 统计 。	false
-copy-on-read	对或错。设置高速缓存读取操作的“读取时复制”功能。请参阅 安全条目修改 。	false
-lock-lease	设置此缓存获取的分布式锁定租约的长度（以秒为单位）。请参阅 设置缓存超时 。	120
-lock-timeout	设置高速缓存操作在超时之前可等待以获取分布式锁定租约的秒数。请参阅 设置缓存超时 。	60
-message-sync-interval	设置主缓存服务器节点将消息发送到所有辅助缓存服务器节点的频率（以秒为单位），以删除已从队列调度的事件。请参阅 更改服务器队列同步率 。	1

名称	描述	默认值
-search-timeout	设置缓存获取操作可用于搜索值的秒数。请参阅 设置缓存超时 。	300

示例命令:

```
1 | alter runtime --members=server1 --loglevel=WARN --enable-statistics=true
```

样本输出:

```
1 | gfsh>alter runtime --members=server1 --loglevel=WARN --enable-statistics=true
2 | Runtime configuration altered successfully for the following member(s)
3 | 192.0.2.0(server1:240)<v1>:64871
```

backup disk-store {#backup_disk_store}

将所有成员的持久数据备份到指定目录。

指定的目录必须存在于所有成员上，但它可以是每台计算机上的本地目录。此命令可确保备份文件不会被并发操作损坏。建议不要使用操作系统复制命令备份正在运行的系统。

您还可以使用此命令执行增量备份。有关增量备份的详细信息，请参阅[为系统恢复和操作管理创建备份](#)。

可用性: Online. 您必须在 gfsh 中连接到 JMX Manager 成员才能使用此命令。

句法:

```
1 | backup disk-store --dir=value [--baseline-dir=value]
```

名称	描述
--dir	<i>Required.</i> 写入备份文件的目录。
--baseline-dir	包含用于在增量备份期间进行比较的基准备份的目录。增量备份操作会备份 --baseline-dir 中指定的目录中不存在的任何数据。如果该成员找不到以前备份的数据，或者先前备份的数据已损坏，则该命令会对该成员执行完全备份。

Table 1. 备份磁盘存储参数

示例命令:

```
1 | backup disk-store --dir=data/backups
2 |
3 | backup disk-store --dir=data/backup/disk-store --
  | baselineDir=data/backups/2012-09-24-17-08-50
```

样本输出:

```
1 gfsh>backup disk-store --dir=data/backups
2 The following disk stores were backed up successfully
3
4 Member | UUID | Directory
5 -----|-----|-----
6 server2 | a6bb11f0-0baa-45c9-b23e-64876d02a586 | c:\PivotalGemFire70\Latest\server2\ | 192.0.2.0
7 server1 | 8dc365bd-c086-4af4-99d0-86b0b521aa04 | c:\PivotalGemFire70\Latest\server1\ | 192.0.2.0
```

change loglevel {#change_loglevel}

更改指定成员的日志记录级别。

可用性: Online. 您必须在 gfsh 中连接到JMX Manager成员才能使用此命令。

句法:

```
1 change loglevel --loglevel=value [--members=value(,value)*] [--groups=value(,value)*]
```

Name	Description	Default Value
--members	Name or ID of one or more member(s) whose logging level you want to change.	
--groups	One or more group names. The logging level changes for all members of these groups.	
--loglevel	<i>Required.</i> Log level to change. Valid options are: ALL, TRACE, DEBUG, INFO, WARN, ERROR, FATAL, OFF.	

Table 1. 更改Loglevel参数

示例命令:

```
1 gfsh>change loglevel --loglevel=DEBUG --members=server1
```

样本输出:

```
1 gfsh>change loglevel --loglevel=DEBUG --members=server1
2
3 Summary
4
5 Member | Changed log-level
6 -----|-----
7 192.0.2.0(server1:3060)<v1>:24653 | true
```

clear defined indexes {#clear_defined_indexes}

清除所有已定义的索引。

索引定义本地存储在 `gfsh` 客户端上。如果要创建一组新索引，或者如果一个或多个索引创建失败，则可能需要清除定义

另请参见[define index](#)。

可用性: Online or offline.

句法:

```
1 | clear defined indexes
```

示例命令:

```
1 | gfsh> clear defined indexes
```

样本输出:

```
1 | gfsh>clear defined indexes
2 | Index definitions successfully cleared
```

close {#close}

关闭持久客户CQ和持久客户。

- **close durable-client**
试图关闭一个持久的客户端。必须断开客户端才能使此命令起作用。
- **close durable-cq**
关闭持久客户端注册的持久连续查询（CQ），并从订阅队列中排除为持久CQ保留的事件。

close durable-client {#close_durable_client}

试图关闭一个持久的客户端。必须断开客户端才能使此命令起作用。

可用性: Online. 您必须在 `gfsh` 中连接到JMX Manager成员才能使用此命令。

句法:

```
1 | close durable-client --durable-client-id=value [--members=value(,value)*]
2 | [--groups=value(,value)*]
```

名称	描述
--durable-client-id	<i>Required.</i> 持久客户端的ID.
--members	要关闭持久客户端的成员的名称或ID.
--groups	持久客户将被关闭的成员组.

Table 1. 关闭持久客户端参数

示例命令:

```
1 | close durable-client --durable-client-id=client1
```

样本输出:

```
1 | gfsh>close durable-client --durable-client-id=client1
2 | Closed the durable client : "client1". on following members.
3 | 1.server4
4 | 2.server3
```

错误消息:

```
1 | gfsh>close durable-cq --durable-cq-name=cq1 --durable-client-id=client1
2 |
3 | Could not close the durable-cq : "cq1" for the durable-client-id :
4 | "client1" due to following reasons.
5 |
6 | CacheClientProxy: Could not drain cq cq1 because client proxy id client1 is
7 | connected.
8 | Occurred on members
9 | 1.server4
10 | 2.server3
11 |
12 | No client found with client-id : client1
13 | Occurred on members
14 | 1.server1
```

close durable-cq {#close_durable_cq}

关闭持久客户端注册的持久连续查询（CQ），并从订阅队列中排除为持久CQ保留的事件。

可用性: Online. You must be connected in `gfsh` to a JMX Manager member to use this command.

句法:

```
1 | close durable-cq --durable-client-id=value --durable-cq-name=value
2 | [--members=value(,value)*] [--groups=value(,value)*]
```

名称	描述
--durable-client-id	Required. 持久客户端的ID.
--durable-cq-name	Required. 要关闭的CQ的名称.
--members	注册持久客户端的成员的名称或ID以及要关闭的持久CQ.
--groups	持久客户注册的成员组和关闭的持久CQ.

Table 2. 关闭Durable-CQ参数

示例命令:

```
1 close durable-cq --durable-client-id=client1 --durable-cq-name=cq1
```

示例输出:

```
1 gfsh>close durable-cq --durable-cq-name=cq1 --durable-client-id=client1
2 Closed the durable cq : "cq1" for the durable client : "client1". on
  following members.
3 1.server4
4 2.server3
```

错误消息:

```
1 gfsh>close durable-client --durable-client-id=client1
2 Unable to close the durable client : "client1" due to following reasons.
3
4 Cannot close a running durable client : client1
5 Occurred on members
6 1.server4
7 2.server3
8
9 No client found with client-id : client1
10 Occurred on members
11 1.server1
```

compact {#compact}

压缩在线和离线磁盘存储。

- **compact disk-store**
使用该磁盘存储在所有成员上压缩磁盘存储。
- **compact offline-disk-store**
压缩脱机磁盘存储。

compact disk-store {#compact_disk_store}

使用该磁盘存储在所有成员上压缩磁盘存储。

此命令使用每个成员为其磁盘存储配置的压缩阈值。磁盘存储必须将 `allow-force-compaction` 属性设置为 `true`。

有关详细信息，请参阅[在磁盘存储日志文件上运行压缩](#)。

可用性: Online. 您必须在 gfsH 中连接到 JMX Manager 成员才能使用此命令。

句法:

```
1 compact disk-store --name=value [--groups=value(,value)*]
```

名称	描述
--name	<i>Required.</i> 要压缩的磁盘存储的名称。
--groups	执行磁盘压缩的成员组。如果未指定组，则磁盘存储将由所有成员压缩。

Table 1. 压缩磁盘存储参数

示例命令:

```
1 compact disk-store --name=Disk1
2 compact disk-store --name=Disk1 --group=MemberGroup1,MemberGroup2
```

错误消息:

```
1 "Disk store \"{0}\" does not exist.";
2 " for group(s) \"{0}\"";
3 "No members found in the specified group(s) \"{0}\".";
4 "Compaction was attempted but nothing to compact.";
5 "Error occurred while doing compaction. Reason: \"{0}\"";
```

compact offline-disk-store {#compact_offline_disk_store}

压缩脱机磁盘存储。

如果磁盘存储很大，则可能需要使用 `-J=-XmxNNNm` 参数为进程分配额外的内存。

有关详细信息，请参阅[在磁盘存储日志文件上运行压缩](#)。

注意: 不要在增量备份的基线目录上执行脱机压缩。

可用性: Online or offline.

句法:

```
1 compact offline-disk-store --name=value --disk-dirs=value(,value)*
2 [--max-oplog-size=value] [--J=value(,value)*]
```

名称	描述	默认值
--name	<i>Required.</i> 要压缩的脱机磁盘存储的名称。	
--disk-dirs	<i>Required.</i> 一个或多个目录，其中先前已写入磁盘存储的数据。用逗号分隔目录。	
--max-oplog-size	压缩创建的oplog的最大大小（以兆字节为单位）。	-1
--J	传递给Java虚拟机的参数在磁盘存储上执行压缩操作。例如： <code>-J=-Xmx1024m</code> 。	

Table 2. 压缩离线磁盘存储参数

示例命令:

```
1 compact offline-disk-store --name=Disk2 --disk-dirs=/Disks/Disk2
2 compact offline-disk-store --name=Disk2 --disk-dirs=/Disks/Disk2 --max-oplog-size=512 -J=-Xmx1024m
```

configure {#configure}

为集群中的所有缓存配置Portable Data eXchange。

configure pdx

为集群中的所有缓存配置Geode的Portable Data eXchange。此命令不会影响系统中正在运行的成员。此命令将使用群集配置服务在定位器中保留pdx配置。

注意: 应在启动任何数据成员之前发出此命令。

可用性: Online.

句法:

```
1 configure pdx [--read-serialized=value] [--ignore-unread-fields=value]
2               [--disk-store=value] [--auto-serializable-classes=value(,value)*]
3               [--portable-auto-serializable-classes=value(,value)*]
```

名称	描述	默认值
--read-serialized	设置为true以使PDX反序列化生成PdxInstance而不是域类的实例。	false
--ignore-unread-fields	控制PDX是否忽略反序列化期间未读取的字段。默认设置是在序列化期间保留未读字段包括其数据。但是，如果将缓存配置为忽略未读字段，则在序列化期间它们的数据将丢失。如果您知道此成员将只读取缓存数据，则应仅将此属性设置为true。在此用例中，您不需要支付保留未读字段的成本，因为您永远不会重新序列化PDX数据。	false
--disk-store	命名磁盘存储，其中将存储PDX类型数据	none

--disk-store	即磁盘存储，其中存储PDX安全数据。	none
名称 --auto-serializable-classes	描述 将ReflectionBasedAutoSerializer配置为成员类的PDX序列化程序。与域类名称匹配的模式，以确定它们是否应自动序列化。不检查类是否可以移植到非Java语言（相当于 check-portability=false）。	默认值 none
--portable-auto-serializable-classes	将ReflectionBasedAutoSerializer配置为成员类的PDX序列化程序。与域类名匹配的模式，以确定是否应序列化它们。如果这些类的对象不能移植到非Java语言（相当于 check-portability=true），PDX autoserIALIZER完成的序列化将引发异常。	none

Table 1. Configure PDX Parameters

示例命令:

```
1 | gfsh>configure pdx --read-serialized=true
```

示例输出:

```
1 | gfsh>configure pdx --read-serialized=true
2 | persistent = false
3 | read-serialized = true
4 | ignore-unread-fields = false
5 |
6 | gfsh>configure pdx --disk-store=/home/username/server4/DEFAULT.drf
7 | persistent = true
8 | disk-store = /home/username/server4/DEFAULT.drf
9 | read-serialized = false
10 | ignore-unread-fields = false
```

错误消息:

```
1 | "Failed to persist the configuration changes due to this command, Revert the
   | command to maintain consistency.
2 | Please use "status cluster-config-service" to determining whether Cluster
   | configuration service is RUNNING."
```

connect {#connect}

直接或通过定位器连接到JMX管理器。

如果您通过定位器进行连接，并且JMX管理器尚不存在，则定位器将启动一个。

gfsh作为发现客户端连接到定位器服务并询问JMX Manager的位置。定位器知道何时没有当前配置为JMX管理器的成员，只需在其自身内启动JMX管理器服务。gfsh作为JMX客户端连接到定位器的JMX RMI 端口。

您还可以使用HTTP协议连接到远程定位器，如下面的第二个示例所示。

可用性: Offline. 如果您已连接，您将收到“已连接到：主机[端口]”的通知。

句法:


```
1 connect [--locator=value] [--jmx-manager=value] [--use-http(=value)?] [--  
url=value]  
2     [--user=value][--password=value] [--key-store=value] [--key-store-  
password=value]  
3     [--trust-store=value] [--trust-store-password=value] [--ciphers=value]  
4     [--protocols=value] [--security-properties-file=value] [--use-  
ssl(=value)?]  
5     [--skip-ssl-validation(=value)?]
```

名称	描述	默认值
--locator	定位器的网络地址格式为： host [port]。	localhost[10334]
--jmx-manager	JMX管理器的网络地址格式为： host [port]。	
--use-http	使用HTTP协议连接到JMX管理器HTTP服务。	如果未指定参数： false 如果指定的参数没有值： true
--url	用于连接到JMX管理器的HTTP服务的URL	http://localhost:8080/gemfire/v1
--user	连接到JMX管理器时用于身份验证的凭据的用户名。如果指定，如果未指定 --password 选项， gfsh 将提示输入密码。	
--password	连接到JMX管理器时用于身份验证的凭据的密码部分。	
--key-store	包含此应用程序的证书和私钥的Java密钥库文件。如果未指定 --key-store-password 参数， gfsh 会提示操作员输入密码。	
--key-store-password	用于从 --key-store 指定的密钥库文件访问私钥的密码。	
--trust-store	包含此应用程序信任的CA证书集合的Java密钥库文件。如果未指定 --trust-store-password 参数， gfsh 会提示操作员输入密码。	
--trust-store-password	用于解锁 --trust-store 指定的密钥库文件的密码。	
--ciphers	加密连接时使用的SSL/TLS密码。默认值为“any”。	
--protocols	加密连接时启用的SSL/TLS协议版本。默认值为“any”。	
--security-properties-file	gfsecurity.properties 文件，用于配置gfsh以连接到Locator/Manager。文件路径可以是当前gfsh目录的绝对路径或相对路径。	
--use-ssl	是否使用SSL与Locator和/或JMX Manager进行通信。如果设置为 true，则connect命令也会读取 gfsecurity.properties。SSL选项优先于属性文件中设置的值。如果未指定，则使用默认值。	如果未指定参数： false 如果指定的参数没有值： true

名称	描述	默认值
--skip-ssl-validation	启用SSL通信并指定此选项或赋值为true时，此gfsh客户端接受任何SSL证书，允许此gfsh客户端验证它所连接的任何定位器或服务器。此选项的存在是为了便于测试，而不适用于生产系统。	false

Table 1. 连接参数

示例命令:

如果未指定定位器或JMX管理器，则 gfsh 将连接到默认端口上localhost上的定位器。

```
1 gfsh>connect
```

示例输出:

```
1 gfsh>connect
2 Connecting to Locator at [host=localhost, port=10334] ..
3 Connecting to Manager at [host=GemFireStymon, port=1099] ..
4 Successfully connected to: [host=GemFireStymon, port=1099]
```

通过HTTP连接到远程定位器的示例:

```
1 gfsh>connect --use-http=true --
  url="http://myLocatorHost.example.com:8080/gemfire/v1"
2 Successfully connected to: GemFire Manager's HTTP service @
3 http://myLocatorHost.example.com:8080/gemfire/v1
```

错误消息:

```
1 "Locator could not find a JMX Manager";
2 "jmx password must be specified.";
3 "Could not connect to : {0}. {1}";
4 "Could not find a GemFire jmx-manager service running at {0}.";
5 "Could not connect to GemFire Locator service at {0}."
```

create {#create}

创建异步事件队列，磁盘存储，网关接收器，网关发件人，索引和区域。

- **create async-event-queue**
在网关发件人传递事件之前，为批处理事件创建异步事件队列。
- **create defined indexes**
创建所有已定义的索引。
- **create disk-store**
定义一个或多个磁盘存储池，可供区域和客户机订阅队列使用，以及网关发送方队列用于WAN分发。
- **create gateway-receiver**

创建网关接收器。每个成员只能有一个网关接收器，与网关发送器不同，您不需要为网关接收器指定标识符。

- **create gateway-sender**

在群集的一个或多个成员上创建网关发件人。

- **create index**

创建可在执行查询时使用的索引。

- **create jndi-binding**

创建一个JNDI绑定，指定描述JDBC连接的资源属性。

- **create lucene index**

创建具有给定路径和配置的区域。

- **create region**

创建具有给定路径和配置的区域。

create async-event-queue {#create_async_event_queue}

在网关发件人传递事件之前，为批处理事件创建异步事件队列。

请参阅[配置多站点 \(WAN\) 事件队列](#)。

可用性: Online. 您必须在 `gfs` 中连接到JMX Manager成员才能使用此命令。

句法:

```
1 create async-event-queue --id=value --listener=value [--  
  groups=value(,value)*]  
2   [--parallel(=value)?] [--enable-batch-conflation(=value)?] [--batch-  
  size=value]  
3   [--batch-time-interval=value] [--persistent(=value)?] [--disk-  
  store=value]  
4   [--disk-synchronous(=value)?] [--max-queue-memory=value]  
5   [--dispatcher-threads=value] [--order-policy=value]  
6   [--gateway-event-filter=value(,value)*]  
7   [--gateway-event-substitution-filter=value]  
8   [--listener-param=value(,value)*]
```

Parameters, create async-event-queue:

名称	描述	默认值
-id	<i>Required.</i> 异步事件队列的ID	
-groups	队列是在组的所有成员上创建的。如果未指定组，则会在所有成员上创建队列。	
-parallel	指定队列是否并行。	false
-enable-batch-conflation	启用批处理混合。	false
-batch-size	批处理可以包含的最大消息数。	100
-batch-time-interval	在批处理之前可以经过的最长时间（以毫秒为单位）。	5
-persistent	确定Geode是否持久保留此队列的布尔值。	false如果指定了一个值，则默认为true。
-disk-store	命名磁盘存储，用于存储队列溢出或持久化队列。如果指定值，则必须存在指定的磁盘存储。如果指定空值，Geode将使用默认磁盘存储来进行溢出和队列持久性。	
-disk-synchronous	指定磁盘写入是否同步。	true
-max-queue-memory	在溢出到磁盘之前队列可以消耗的最大内存量（兆字节）。	100
-dispatcher-threads	用于发送事件的线程数。	5
-order-policy	当 <code>-dispatcher-threads is > 1</code> 时调度事件的策略。可能的值是 <code>THREAD</code> , <code>KEY</code> , <code>PARTITION</code> 。	KEY

名称	描述	默认值
-gateway-event-filter	此队列的GatewayEventFilters的完全限定类名列表。 这些类在分派到远程服务器之前过滤事件。	
-gateway-event-substitution-filter	此队列的 GatewayEventSubstitutionFilter 的完全限定类名。	
-listener	<i>Required.</i> 此队列的Async Event Listener的完全限定类名	
-listener-param	要传递给Async Event Listener类的参数名称和值。 （可选）您可以通过使用 # 字符和值跟随参数名称来指定值。 例如： <code>-listener-param=myParam#24</code>	
-forward-expiration-destroy	允许将到期销毁操作转发到AsyncEventListener实例。 如果指定没有值，则此参数设置为“false”。	false

示例命令:

```
1 | create async-event-queue --id=myAEQ --listener=myApp.myListener
```

create defined indexes {#create_defined_indexes}

创建所有已定义的索引。

另请参阅[define index](#)和[clear defined indexes](#)。

可用性: Online. 必须在gfsh中连接到JMX Manager成员才能使用此命令。

句法:

```
1 | create defined indexes [--members=value(,value)*] [--groups=value(,value)*]
```

参数，创建定义的索引:

名称	描述	默认值
--members	将在其上创建索引的成员的名称/ID。	
--groups	将在成员组中的所有成员上创建索引。	

示例命令:

```
1 | create defined indexes
```

示例输出:

```
1 | gfsh>create defined indexes
2 | Indexes successfully created. Use list indexes to get details.
3 | 1. ubuntu(server1:17682)<v1>:27574
```

如果索引创建失败，您可能会在gfsh中收到类似于以下内容的错误消息：

```
1 gfsh>create defined indexes
2 Exception : org.apache.geode.cache.query.RegionNotFoundException ,
3 Message : Region ' /r3' not found: from /r30ccurred on following members
4 1. india(s1:17866)<v1>:27809
```

create disk-store {#create_disk_store}

定义一个或多个磁盘存储池，可供区域和客户机订阅队列使用，以及网关发送方队列用于WAN分发。

请参阅[磁盘存储](#)

可用性: Online. 必须在gfsh中连接到JMX Manager成员才能使用此命令。

句法:

```
1 create disk-store --name=value --dir=value(,value)* [--allow-force-
  compaction(=value)?]
2 [--auto-compact(=value)?] [--compaction-threshold=value] [--max-oplog-
  size=value]
3 [--queue-size=value] [--time-interval=value] [--write-buffer-size=value]
4 [--groups=value(,value)*]
5 [--disk-usage-warning-percentage=value] [--disk-usage-critical-
  percentage=value]
```

参数，创建磁盘存储:

名称	描述	默认值
--name	<i>Required.</i> 此磁盘存储的名称。	
--dir	<i>Required.</i> 写入磁盘存储文件的一个或多个目录名。可选地，目录名后面可以跟 # 和磁盘存储可以在目录中使用的最大兆字节数。例如：--dir=/data/ds1 --dir=/data/ds2#5000。如果指定的目录不存在，该命令将为您创建目录。	
--allow-force-compaction	设置为true以允许在此磁盘存储上强制执行磁盘压缩。	false
--auto-compact	设置为true以自动压缩磁盘文件。	true
--compaction-threshold	磁盘存储符合压缩条件之前允许的垃圾百分比。	50
--max-oplog-size	oplog文件的最大大小（以兆字节为单位）。当oplog文件达到此大小时，该文件将转到新文件。	1024
--queue-size	可以异步排队以写入磁盘的最大操作数。	0
--time-interval	将未写入的数据写入磁盘之前可以经过的毫秒数。	1000
--groups	磁盘存储在组的所有成员上创建。如果未指定组，则会在所有成员上创建磁盘存储。	
--write-buffer-size	此磁盘存储在将数据写入磁盘时使用的写入缓冲区的大小。较大的值可能会提高性能但会占用更多内存。磁盘存储分配一个此大小的直接内存缓冲区。	32768
--disk-usage-warning-percentage	磁盘使用率高于此阈值会生成警告消息。例如，如果阈值设置为90%，则在100 GB可用磁盘空间下的1 TB驱动器上会生成警告。设置为“0”（零）以禁用。	90
--disk-usage-critical-percentage	磁盘使用率高于此阈值会生成错误消息并关闭成员的缓存。例如，如果阈值设置为99%，则1 TB驱动器上的10 GB可用磁盘空间不足会生成错误并关闭缓存。设置为“0”（零）以禁用。	99

示例命令:

```
1 | create disk-store --name-store1 --dir=/data/ds1
```

示例输出:


```

1 gfsh>create disk-store --name-store1 --dir=/data/ds1
2 Member | Result
3 ----- | -----
4 server1 | Success

```

create gateway-receiver {#create_gateway_receiver}

创建网关接收器。每个成员只能有一个网关接收器，与网关发送器不同，您不需要为网关接收器指定标识符。

除非指定了 `--groups` 或 `--members` 选项，否则创建将在所有服务器上进行。

如果网关接收器创建成功至少一个成员，则此 `gfsh` 命令退出，退出代码表示成功。

输出每个成员的网关接收器的表格格式状态，与创建的成功或失败无关。

请参阅[网关接收器](#)。

可用性: Online. 您必须在 `gfsh` 中连接到 JMX Manager 成员才能使用此命令。

句法:

```

1 create gateway-receiver [--groups=value(,value)*] [--members=value(,value)*]
2   [--manual-start=value] [--start-port=value] [--end-port=value] [--bind-
   address=value]
3   [--maximum-time-between-pings=value] [--socket-buffer-size=value]
4   [--gateway-transport-filter=value(,value)*] [--hostname-for-senders=value]
5   [--if-not-exists=(value)?]

```

参数，创建网关接收器:

名称	描述	默认值
<code>--groups</code>	网关接收器是在组的成员上创建的。	
<code>--members</code>	要在其上创建网关接收器的成员的名称。为了向后兼容，如果指定了此选项并且启用了集群配置，则不会保留网关接收器配置。	
<code>--manual-start</code>	布尔值，指定是否需要手动启动网关接收器。如果提供空值，则网关接收器自动启动默认值为“false”。	true
<code>--start-port</code>	在指定可能的端口号范围时使用的起始端口号，此网关接收器将用于连接到其他站点中的网关发送器。Geode在指定的端口号范围内选择一个未使用的端口号来启动接收器。如果该范围内没有可用的端口号，则抛出异常。STARTPORT 值包含在内，而 ENDPOINT 值是独占的。例如，如果指定 STARTPORT = "50510" 和 ENDPOINT = "50520"，Geode 会选择50510到50519之间的端口值。	5000
<code>--end-port</code>	定义在指定此网关接收器将用于来自其他站点中的网关发件人的连接的可能端口号范围时要使用的上限端口号。Geode在指定的端口号范围内选择一个未使用的端口号来启动接收器。如果该范围内没有可用的端口号，则抛出异常。ENDPOINT 值是独占的，而 STARTPORT 值是包含的。例如，如果指定 STARTPORT="50510" 和 ENDPOINT="50520"，Geode会选择50510到50519之间的端口值。	5500
<code>--bind-address</code>	来自其他站点中的网关发件人的连接的网络地址。将地址指定为文字字符串。	

address	子字符串。	
名称	描述	默认值
--socket-buffer-size	一个整数值，用于设置此网关接收器的套接字连接的缓冲区大小（以字节为单位）。此值应与连接到此接收器的网关发件人的 socket-buffer-size 设置相匹配。	32768
--gateway-transport-filter	要添加到网关接收器的GatewayTransportFilter的完全限定类名。	
--maximum-time-between-pings	整数值，指定ping到连接的WAN站点之间使用的时间间隔（以毫秒为单位）。此值确定在将远程WAN站点视为脱机之前可以经过的最长时间。	60000
--hostname-for-senders	主机名或IP地址告知网关发件人作为他们连接的地址。定位器通知网关发件人此值。	
--if-not-exists	如果指定时未提供布尔值或指定并设置为true，则网关接收器（如果已存在）将不会创建。命令输出报告每次创建尝试的状态。	false

示例命令:

```
1 | gfsh>create gateway-receiver --members=server1
```

示例输出:

```
1 | gfsh>create gateway-receiver --members=server1
2 | Member | Status
3 | ----- | -----
4 | server1 | GatewayReceiver created on member "server1" and will listen on the port "0"
```

create gateway-sender {#create_gateway_sender}

在群集的一个或多个成员上创建网关发件人。

请参阅[网关发件人](#)。

注意: 在承载网关发件人的每个Geode成员上，特定发件人 id 的网关发件人配置必须相同。

可用性: Online. 您必须在 gfsh 中连接到JMX Manager成员才能使用此命令。

句法:

```
1 create gateway-sender --id=value --remote-distributed-system-id=value
2   [--groups=value(,value)*] [--members=value(,value)*] [--parallel=value]
3   [--manual-start=value] [--socket-buffer-size=value] [--socket-read-
4   timeout=value]
5   [--enable-batch-conflation=value] [--batch-size=value] [--batch-time-
6   interval=value]
7   [--enable-persistence=value] [--disk-store-name=value] [--disk-
8   synchronous=value]
9   [--maximum-queue-memory=value] [--alert-threshold=value] [--dispatcher-
10  threads=value]
11  [--order-policy=value] [--gateway-event-filter=value(,value)*]
12  [--gateway-transport-filter=value(,value)*]
```

参数，创建网关发件人：

名称	描述	默认值
--id	<i>Required.</i> 网关发件人的唯一标识符，通常是与物理位置关联的标识符。	
--remote-distributed-system-id	<i>Required.</i> 此网关发件人发送事件的远程群集的ID。	
--groups	网关发件人是在组成员上创建的。	
--members	要在其上创建网关发件人的成员的名称。	
--parallel	设置为true时，指定并行网关发件人。	false
--enable-batch-conflation	确定Geode是否应该混淆消息的布尔值。	false
--manual-start	不推荐使用 布尔值，指定是否需要手动启动网关发件人。如果提供空值，则使用默认值false，网关发件人自动启动。 <i>手动启动可能会导致数据丢失，因此不应在生产系统中使用手动启动。</i>	false
--socket-buffer-size	将消息发送到远程站点的套接字缓冲区的大小。此大小应与处理区域事件的远程网关接收器的 socket-buffer-size 属性的大小相匹配。	32768
--socket-read-timeout	网关发件人等待从远程站点接收确认的时间量（以毫秒为单位）。默认情况下，此值设置为0，表示没有超时。如果设置此超时，则必须将其设置为最小值30000（毫秒）。将其设置为较小的数字将生成错误消息并将值重置为默认值0。	0
--batch-size	批处理可以包含的最大消息数。	100
--batch-time-interval	发送批次之间可以经过的最大毫秒数。	1000
--enable-persistence	确定Geode是否持久保存网关队列的布尔值。	false
--disk-store-name	命名磁盘存储，用于存储队列溢出或持久化队列。如果指定值，则必须存在指定的磁盘存储。如果指定空值，Geode将使用默认磁盘存储来进行溢出和队列持久性。	
--disk-synchronous	对于写入磁盘的区域，boolean指定是否为该区域同步完成磁盘写入。	true
--maximum-queue-memory	在溢出到磁盘之前队列可以消耗的最大内存量（兆字节）。	100 MB
--alert-threshold	在Geode记录警报之前，区域事件可以保留在网关发件人队列中的最大毫秒数。	0
--dispatcher-threads	用于处理来自网关发件人队列或异步事件队列的区域事件的调度程序线程数。	5
--order-policy	当 dispatcher-threads 属性大于1时，order-policy 配置多个调度程序线程处理来自串行网关队列或串行异步事件队列的区域事件的方式。此属性可以具有以下值之一： key 从本地队列分发区域事件时，多个调度程序线程保留密钥更新的顺序。 thread 从本地队列分发区域事件时，多个调度程序线程保留给定线程将区域事件添加到队列的顺序。 partition 从本地队列分发区域事件时，多个调度程序线程保留区域事件添加到本地队列的顺序。对于分区区域，这意味着传递到特定分区的所有区域事件都以相同的顺序传递到远程Geode站点。对于分布式区域，这意味着传递到本地网关发送方队列的所有密钥更新都以相同的顺序分发到远程站点。您无法为并行事件队列配置 order-policy，因为并行队列无法保留事件订购地区。只能保留给定分区（或分布式区域的给定队列）中事件的顺序。	key
--gateway-event-filter	要与GatewaySender关联的GatewayEventFilters（由逗号分隔）的完全限定类名列表。这用作用户在调度到远程集群之前过滤掉事件的回调。例如： gateway-event-filter=com.user.filters.MyFilter1,com.user.filters.MyFilters2	
--gateway-transport-filter	要添加到GatewaySender的GatewayTransportFilter的完全限定类名。	

示例命令:

```
1 | gfsh>create gateway-sender --remote-distributed-system-id="2" --id="sender2"
```

示例输出:

```

1 gfsh>create gateway-sender --remote-distributed-system-id="2" --id="sender2"
2 Member | Status
3 ----- | -----
4 server1 | GatewaySender "sender2" created on "server1"

```

create index {#create_index}

创建可在执行查询时使用的索引。

可用性: Online. 您必须在 gfsh 中连接到 JMX Manager 成员才能使用此命令。

请参阅[使用索引](#)。

句法:

```

1 create index --name=value --expression=value --region=value
2 [--members=value(,value)*] [--type=value] [--groups=value(,value)*]

```

参数, 创建索引:

名称	描述	默认值
--name	<i>Required.</i> 要创建的索引的名称。	
--expression	<i>Required.</i> 索引引用的区域值的字段。	
--region	<i>Required.</i> 与查询中的“from”子句对应的区域的名称/路径。	
--members	将在其上创建索引的成员的名称/ID。	
--type	索引的类型。有效值为: <code>range</code> 和 <code>key</code> 。 (第三种类型, <code>hash</code> , 仍然被识别, 但不推荐使用哈希索引。)	<code>range</code>
--groups	将在组中的所有成员上创建索引。	

示例命令:

```

1 create index --name=myKeyIndex --expression=region1.Id --region=region1 --
  type=key

```

示例输出:

```

1 gfsh>create index --name=myKeyIndex --expression=region1.Id --region=region1
  --type=key
2 Index successfully created with following details
3 Name      : myKeyIndex
4 Expression : region1.Id
5 RegionPath : /region1
6 Members which contain the index
7 1. ubuntu(server1:17682)<v1>:27574
8
9 gfsh>create index --name=myIndex2 --expression=exp2 --region=/exampleRegion
10 Failed to create index "myIndex2" due to following reasons
11 Index "myIndex2" already exists. Create failed due to duplicate name.
12 Occurred on following members
13 1. ubuntu(server1:17682)<v1>:27574

```

create jndi-binding {#create_jndi_binding}

创建一个JNDI绑定，指定描述JDBC连接的资源属性。

可用性: Online. 必须在gfsh中连接到JMX Manager成员才能使用此命令。

句法:

```
1 create jndi-binding --name=value --type=value --jdbc-driver-class=value
2   --connection-url=value [--blocking-timeout-seconds=value]
3   [--conn-pooled-datasource-class=value] [--idle-timeout-seconds=value]
4   [--init-pool-size=value] [--login-timeout-seconds=value]
5   [--managed-conn-factory-class=value] [--max-pool-size=value] [--
password=value]
6   [--transaction-type=value] [--username=value] [--xa-datasource-class=value]
7   [--if-not-exists(=value)?] [--datasource-config-properties=value(,value)*]
```

参数, 创建jndi绑定:

名称	描述	默认值
--name	<i>Required.</i> 要创建的绑定的名称。	
--type	<i>Required.</i> XA数据源的类型。 其中之一： <code>MANAGED</code> , <code>SIMPLE</code> , <code>POOLED</code> 或 <code>XAPOOLED</code> 。	
--jdbc-driver-class	<i>Required.</i> JDBC驱动程序类的标准名称。	
--connection-url	<i>Required.</i> JDBC驱动程序连接URL字符串。 例如, <code>jdbc:hsqldb:hsqldb://localhost:1701</code> 。	
--blocking-timeout-seconds	指定在抛出异常之前等待连接时阻塞的最长时间（以秒为单位）。	
--conn-pooled-datasource-class	包含XA数据源连接的连接池实现的标准名称。	
--idle-timeout-seconds	指定连接在关闭之前可能处于空闲状态的时间（以秒为单位）。	
--init-pool-size	指定池应保留的初始连接数。	
--login-timeout-seconds	客户端线程因不活动而断开连接的秒数。	
--managed-conn-factory-class	连接工厂实现的完全限定名称。	
--max-pool-size	池中可以创建的最大连接数。	
--password	创建新连接时使用的默认密码。	
--transaction-type	交易类型。 一个是 <code>XATransaction</code> , <code>NoTransaction</code> 或 <code>LocalTransaction</code> 。	
--username	指定创建新连接时要使用的默认用户名。	
--xa-datasource-class	<code>javax.sql.XADataSource</code> 实现类的完全限定名。	
--if-not-exists	如果为true, 则如果已存在具有相同名称的jndi绑定, 则不会创建重复的jndi绑定。 如果为false, 则尝试创建重复的jndi绑定会导致错误。 如果指定的选项没有值, 则该选项设置为true。	false

名称	描述	默认值
--datasource-config-properties	自定义 XADatasource 驱动程序的属性。附加包含 (名称, 类型, 值) 元组的JSON字符串以设置任何属性。例如: <code>--datasource-config-properties={ 'name': 'name1', 'type': 'type1', 'value': 'value1' }, { 'name': 'name2', 'type': 'type2', 'value': 'value2' }</code>	

示例命令:

```

1 gfsh>create jndi-binding --name=jndi1 --type=SIMPLE \
2   --jdbc-driver-class=org.apache.derby.jdbc.EmbeddedDriver \
3   --connection-url="jdbc:derby:newDB;create=true"

```

create lucene index {#create_lucene_index}

创建Lucene索引。有关Lucene索引创建的详细信息, 请参阅[Apache Lucene Integration](#)。

有关Lucene相关的其他gfsh命令, 请参阅[describe lucene index](#), [destroy lucene index](#), [list lucene indexes](#) 和 [search lucene](#)。

可用性: Online. 必须在gfsh中连接到JMX Manager成员才能使用此命令。

句法:

```

1 create lucene index --name=value --region=value --field=value(,value)*
2   [--analyzer=value(,value)*] [--serializer=value] [--group=value(,value)*]

```

参数, 创建lucene索引:

名称	描述	默认值
--name	<i>Required.</i> 要创建的索引的名称.	
--region	<i>Required.</i> 要定义索引的区域的名称/路径.	
--field	<i>Required.</i> 索引引用的区域值的字段, 指定为以逗号分隔的列表. 要将整个值视为单个字段, 请指定 <code>__REGION_VALUE_FIELD</code> .	
--analyzer	分析器从文本中提取术语, 指定为以逗号分隔的列表. 如果未指定, 则默认分析器将用于所有字段. 如果指定, 分析器的数量必须与指定的字段数完全匹配. 列出分析器时, 对于将使用默认分析器的任何字段使用关键字 <code>DEFAULT</code> .	Lucene StandardAnalyzer
--serializer	与此索引一起使用的序列化程序的完全限定类名. 序列化器必须实现 <code>LuceneSerializer</code> 接口. 您可以使用内置的 <code>org.apache.geode.cache.lucene.FlatFormatSerializer</code> 来索引和搜索集合和嵌套字段. 如果未指定, 则使用简单的默认序列化程序, 该序列化程序仅索引和搜索区域对象的顶级字段.	simple serializer
--group	将在指定成员组中的所有成员上创建索引.	

示例命令:


```

1 gfsh>create lucene index --name=customerIndex --region=/Customer
2   --field=__REGION_VALUE_FIELD
3
4 gfsh>create lucene index --name=analyzerIndex --region=/Person
5   --field=name,email,address,revenue
6   --analyzer=DEFAULT,org.apache.lucene.analysis.core.KeywordAnalyzer,
7   examples.MyCharacterAnalyzer,DEFAULT

```

示例输出:

```

1 gfsh>create lucene index --name=testIndex --region=testRegion
2   --field=__REGION_VALUE_FIELD
3
4   Member | Status
5 ----- | -----
6 192.168.1.23(server505:17200)<v1>:1025 | Successfully created lucene index

```

create region {#create_region}

创建具有给定路径和配置的区域。

在创建区域时，必须为初始配置指定 `--type` 或 `--template-region`。指定 `--key-constraint` 和 `--value-constraint` 会在查询和索引期间使对象类型信息可用。

请参阅[区域数据存储和分发](#)。

有关语法详细信息，请参阅[在命令行选项中指定JSON](#)。

可用性: Online. 您必须在 gfsh 中连接到JMX Manager成员才能使用此命令。

句法:

```

1 create region --name=value [--type=value] [--template-region=value]
2   [--groups=value(,value)*] [--if-not-exists(=value)?]
3   [--key-constraint=value] [--value-constraint=value]
4   [--enable-statistics=value] [--entry-idle-time-expiration=value]
5   [--entry-idle-time-expiration-action=value]
6   [--entry-time-to-live-expiration=value]
7   [--entry-time-to-live-expiration-action=value]
8   [--entry-idle-time-custom-expiry=value] [--entry-time-to-live-custom-
9   expiry=value]
10  [--region-idle-time-expiration=value]
11  [--region-idle-time-expiration-action=value]
12  [--region-time-to-live-expiration=value]
13  [--region-time-to-live-expiration-action=value] [--disk-store=value]
14  [--enable-synchronous-disk=value] [--enable-async-conflation=value]
15  [--enable-subscription-conflation=value] [--cache-
16  listener=value(,value)*]
17  [--cache-loader=value] [--cache-writer=value]
18  [--async-event-queue-id=value(,value)*]
19  [--gateway-sender-id=value(,value)*] [--enable-concurrency-
20  checks=value]
21  [--enable-cloning=value] [--concurrency-level=value]
22  [--colocated-with=value] [--local-max-memory=value]
23  [--recovery-delay=value] [--redundant-copies=value]
24  [--startup-recovery-delay=value] [--total-max-memory=value]
25  [--total-num-buckets=value] [--compressor=value] [--off-heap(=value)]
26  [--partition-resolver=value] [--eviction-entry-count=value]

```

```
24  [--eviction-max-memory=value] [--eviction-action=value] [--eviction-  
    object-size=value]
```

参数, 创建区域:

名称	描述	默认值
--name	<i>Required.</i> 要创建的区域的名称/路径.	
--type	<i>Required</i> (如果未指定template-region。)要创建的区域类型。选项包括: PARTITION, PARTITION_REDUNDANT, REPLICATE, LOCAL等。要获取所有区域类型选项的列表, 请添加--type参数, 然后选择TAB键以显示完整列表.	
--template-region	<i>Required</i> (如果未指定type。)创建此区域时应重复其属性的区域的名称/路径.	
--groups	将在其上创建区域的成员组.	
--if-not-exists	如果已存在具有相同名称的区域, 则不会创建新区域。默认情况下, 尝试创建重复区域会报告为错误。如果指定此选项没有值或指定值为 <code>true</code> , 则gfsh会显示“跳过...”确认, 但不会引发错误.	false
--key-constraint	作为区域键允许的对象完全限定类名。确保区域条目的键都属于同一类.	
--value-constraint	允许作为区域值的对象完全限定类名。如果未指定, 则区域值可以是任何类.	
--enable-statistics	是否收集该地区的统计数据。必须为true才能在该地区使用expiration.	
--entry-idle-time-expiration	区域的条目可以在未被访问的情况下保留在缓存中多长时间.	no expiration
--entry-idle-time-expiration-action	对超过空闲到期的条目采取的操作。有效的到期操作包括destroy, local-destroy, invalidate (默认), local-invalidate。	
--entry-time-to-live-expiration	区域的条目可以在未被访问或更新的情况下保留在缓存中多长时间。默认值为此类型没有到期。	no expiration
--entry-time-to-live-expiration-action	对超过TTL到期的条目采取的措施。有效的到期操作包括destroy, local-destroy, invalidate (默认), local-invalidate。	
--entry-idle-time-custom-expiry	为入口空闲时间实现CustomExpiry的类的名称。为初始化属性附加JSON字符串。	
--entry-time-to-live-custom-expiry	实现CustomExpiry以进入生存时间的类的名称。为初始化属性附加JSON字符串。	
--region-idle-time-expiration	该区域可以在未被访问的情况下保留在缓存中多长时间。默认值为此类型没有到期。	
--region-idle-time-expiration-action	对超过空闲到期的区域采取的措施。有效的到期操作包括destroy, local-destroy, invalidate (默认), local-invalidate。	
--region-time-to-live-expiration	该区域可以在未被访问或更新的情况下保留在缓存中多长时间。默认值为此类型没有到期。	no expiration
--region-time-to-live-expiration-action	对超过TTL到期的区域采取的措施。有效的到期操作包括destroy, local-destroy, invalidate (默认), local-invalidate。	
--disk-store	此区域使用的磁盘存储。 list disk-stores 命令可用于显示现有磁盘存储。	

名称	描述	默认值
--enable-synchronous-disk	对于将数据持久保存到磁盘的区域，是否同步完成写入。	
--enable-async-conflation	是否允许聚合由该区域的生产者成员发送的异步TCP/IP消息。false值会导致所有异步消息单独发送。	
--enable-subscription-conflation	服务器是否应将其消息与客户端混淆。false值会导致所有服务器 - 客户端消息单独发送。	
--cache-listener	要实例化的插件的完全限定类名，用于接收对区域及其条目的更改的事件后通知。可以配置任意数量的缓存侦听器。完全限定的类名可以附加一个JSON规范，该规范将被解析成为实现Declarable接口的类的init()方法的参数字段。	
--cache-loader	要实例化的插件的完全限定类名，用于接收区域中缓存未命中的通知。最多可以在该区域的每个成员中定义一个缓存加载器。对于分布式区域，可以从具有定义区域的其他成员远程调用缓存加载器。完全限定的类名可以附加一个JSON规范，该规范将被解析成为实现Declarable接口的类的initialize()方法的参数字段。	
--cache-writer	要实例化的插件的完全限定类名，用于接收区域及其条目更改的事件前通知。插件可能会取消该事件。最多可以在该区域的每个成员中定义一个缓存写入器。完全限定的类名可以附加一个JSON规范，该规范将被解析成为实现Declarable接口的类的init()方法的参数字段。	
--async-event-queue-id	将用于后写操作的异步事件队列的ID。	
--gateway-sender-id	要将数据路由到的网关发件人的ID。	
--enable-concurrency-checks	是否实现了区域版本向量。区域版本向量是版本控制方案的扩展，有助于复制区域的同步。	
--enable-cloning	确定fromDelta如何将增量应用于本地缓存以进行增量传播。如果为true，则将更新应用于值的克隆，然后将克隆保存到缓存中。如果为false，则在缓存中就地修改该值。	
--concurrency-level	估计将同时访问区域条目的最大应用程序线程数。此属性不适用于分区区域。	
--colocated-with	与该地区相关的中部地区。	
--local-max-memory	此进程中区域使用的最大内存量（以兆字节为单位）。(默认值为可用堆的90%。)	
--recovery-delay	现有成员在成员崩溃之后等待的延迟（以毫秒为单位），然后在其余成员上恢复此区域的冗余。默认值（-1）表示在发生故障后不会恢复冗余。	
--redundant-copies	所需桶的额外副本数量。额外的副本允许面对VM离开（有意或无意）的高可用性和负载平衡读取操作。（允许值：0,1,2和3）	
--startup-recovery-delay	新成员在假定其共享级别冗余之前将等待的延迟（以毫秒为单位）。这允许在将冗余工作负载分配给新成员之前启动多个区域的时间。值-1表示添加新成员不会触发冗余恢复。	默认设置是在添加新成员时立即恢复冗余。
--total-max-memory	所有进程中该区域使用的最大内存量（以兆字节为单位）。	
--total-num-buckets	所有进程中该区域使用的哈希桶总数。	113

名称	描述	默认值
--compressor	实现区域压缩的Java类名称。您可以编写一个实现 <code>org.apache.geode.compression.Compressor</code> 的自定义压缩器，或者您可以指定与 Geode 捆绑在一起的 Snappy 压缩器 (<code>org.apache.geode.compression.SnappyCompressor</code>)。请参阅 区域压缩 。	no compression
--off-heap	指定区域值是存储在堆内存还是堆外内存中。如果为 true，则区域值位于堆外内存中。如果指定的参数没有值，则使用 true 值。	false
--partition-resolver	指定自定义分区解析程序的完整路径。指定 <code>org.apache.geode.cache.util.StringPrefixPartitionResolver</code> 以使用包含的字符串前缀分区解析程序。	
--eviction-entry-count	实现驱逐，驱逐政策基于该地区的条目数量。	
--eviction-max-memory	启用驱逐，其中驱逐策略基于区域消耗的内存量（以兆字节为单位）。	
--eviction-action	达到驱逐阈值时采取的行动。本地销毁当地销毁物品。请谨慎使用 - 可能导致不一致。 <code>overflow-to-disk</code> 溢出到磁盘。对于分区区域，这提供了整个区域中最可靠的读取行为。	
--eviction-object-sizer	指定 <code>ObjectSizer</code> 接口的实现，以测量区域中对象的大小。sizer 仅适用于基于堆和内存的驱逐。	

示例命令:

```

1 create region --name=region1 --type=REPLICATE_PERSISTENT \
2 --cache-writer=org.apache.geode.examples.MyCachewriter \
3 --group=Group1 --disk-store=DiskStore1
4
5 create region --name=region12 --template-region=/region1
6
7 create region --name=region2 --type=REPLICATE \
8 --cache-listener=org.apache.geode.examples.MyCacheListener1,\
9 org.apache.geode.examples.MyCacheListener2 \
10 --group=Group1,Group2
11
12 create region --name=region3 --type=PARTITION_PERSISTENT --redundant-
13 copies=2 \
14 --total-max-memory=1000 --startup-recovery-delay=5 --total-num-buckets=100
15 \
16 --disk-store=DiskStore2 --cache-
17 listener=org.apache.geode.examples.MyCacheListener3 \
18 --group=Group2
19
20 create region --name=region4 --type=REPLICATE_PROXY \
21 --cache-listener=org.apache.geode.examples.MyCacheListener1 --
22 group=Group1,Group2
23
24 create region --name=myRegion --type=REPLICATE --eviction-max-memory=100 \
25 --eviction-action=overflow-to-disk --eviction-object-
26 sizer=my.company.geode.MySizer
27
28 create region --name=r1 --type=PARTITION \
29 --cache-loader=org.example.myLoader{'URL':'jdbc:cloudscape:rmi:MyData'}
```

示例输出:

```

1 | gfsh>create region --name=myRegion --type=LOCAL
2 | Member | Status
3 | ----- | -----
4 | server1 | Region "/myRegion" created on "server1"

```

debug {#debug}

在 gfsh 中启用或禁用调试输出。

可用性: Online or offline.

句法:

```

1 | debug --state=value

```

名称	描述	默认值
--state	是否打开或关闭调试。有效选项包括：ON, OFF（不区分大小写）	OFF

Table 1. 调试参数

示例命令:

```

1 | debug --state=off
2 | debug --state=on

```

示例输出:

```

1 | gfsh>debug --state=on
2 | Debug is on

```

define index {#define_index}

定义可在执行查询时使用的索引。然后，您可以使用 `create defined indexes` 执行单个命令以一次创建多个索引。

可用性: Online or offline.

句法:

```

1 | define index --name=value --expression=value --region=value [--type=value]

```

名称	描述	默认值
--name	<i>Required.</i> 要定义的索引的名称。	
--expression	<i>Required.</i> 索引引用的区域值的字段。	
--region	<i>Required.</i> Name/Path of the region which corresponds to the "from" clause in a query.	
--type	Type of the index. Valid values are: range, key and hash.	range

Table 1. 定义索引参数

示例命令:

```

1 gfsh> define index --name=myIndex1 --expression=exp1 --region=/exampleRegion
2
3 gfsh> define index --name=myIndex2 --expression="c.exp2" --
  region="/exampleRegion e, e.collection1 c"
4
5 gfsh> define index --name=myIndex3 --expression=exp3 --region=/exampleRegion
  --type=hash
6
7 //then to create the indexes, execute:
8
9 gfsh> create defined indexes

```

示例输出:

```

1 gfsh>define index --name=myIndex1 --expression=exp1 --region=/exampleRegion
2 Index successfully defined with following details
3 Name      : myIndex1
4 Expression : exp1
5 RegionPath : /exampleRegion

```

deploy {#deploy}

将JAR打包的应用程序部署到一个或多个成员。

只能指定 `--jars` 或 `--dir` 中的一个。

可用性: Online. 您必须在 `gfsh` 中连接到JMX Manager成员才能使用此命令。

句法:

```

1 deploy [--groups=value(,value)*] [--jars=value(,value)*] [--dir=value]

```

名称	描述
--groups	将部署指定JAR的组。如果未指定此选项，则将在所有成员上进行部署。
--jars	要部署的JAR的路径。
--dir	从中部署JAR的目录。

Table 1. 部署参数

示例命令:

```
1 | deploy --jars=group1_functions.jar --groups=Group1
2 |
3 | deploy --dir=libs/group1-libs --groups=Group2
```

示例输出:

```
1 | gfsh> deploy --jars=group1_functions.jar --groups=Group1
2 |
3 | Member | Deployed JAR | Deployed JAR Location
4 | -----|-----|-----
5 | datanode1 | group1_functions.jar | /usr/local/gemfire/deploy/GF#group1_functions.jar#1
6 | datanode2 | group1_functions.jar | /usr/local/gemfire/deploy/GF#group1_functions.jar#1
7 |
8 |
9 | gfsh> deploy --dir=libs/group1-libs --groups=Group2
10 |
11 | Deploying files: group2_functions.jar, group2_dependencies.jar
12 | Total file size is: 0.64MB
13 |
14 | Continue? (Y/n): Y
15 |
16 | Member | Deployed JAR | Deployed JAR Location
17 | -----|-----|-----
18 | datanode3 | group2_functions.jar | /usr/local/gemfire/deploy/GF#group2_functions.jar#1
19 | datanode3 | group2_dependencies.jar | /usr/local/gemfire/deploy/GF#group2_dependencies.jar#1
20 | datanode4 | group2_functions.jar | /usr/local/gemfire/deploy/GF#group2_functions.jar#1
21 | datanode4 | group2_dependencies.jar | /usr/local/gemfire/deploy/GF#group2_dependencies.jar#1
```

describe {#describe}

显示成员配置，shell连接，磁盘存储，成员或区域的详细信息。

- describe client

显示指定客户端的详细信息。

- **describe config**
显示成员的配置。
- **describe connection**
显示连接信息详细信息
- **describe disk-store**
显示有关成员磁盘存储的信息。
- **describe jndi-binding**
显示有关JNDI绑定配置的信息。
- **describe lucene index**
显示有关Lucene索引的信息。
- **describe member**
显示具有给定名称/ID的成员的详细信息。
- **describe offline-disk-store**
显示有关脱机成员磁盘存储的信息。
- **describe region**
显示区域的属性和关键信息。

describe client {#describe_client}

显示指定客户端的详细信息。

可用性: Online. 您必须在 `gfsh` 中连接到JMX Manager成员才能使用此命令。

句法:

```
1 | describe client --clientID=value
```

参数，描述客户端:

名称	描述
--clientID	<i>Required.</i> 客户的ID。要查找客户端ID，可以使用 <code>list clients</code> 命令显示已连接客户端及其ID的列表。

示例命令:

```
1 | describe client --clientID=192.0.2.0(4987:loner):58922:7b3398cf
```

示例输出:

```
1 | gfsh>describe client --clientID=192.0.2.0(4987:loner):58922:7b3398cf
2 | -----
3 | Primary Servers           : 192.0.2.0(server1:5764)<v1>:15189
4 | Secondary Servers        : 192.0.2.0(server2:5891)<v2>:39082
5 | CPU                       : 0
6 | Number of Cache Listner Calls : 0
7 | Number of Gets            : 0
8 | Number of Misses          : 0
```

```

 9 | Number of Puts           : 0
10 | Number of Threads        : 0
11 | Process CPU Time (nanoseconds) : 0
12 | Queue size               : 1
13 | UP Time (seconds)        : 67
14 | Is Durable               : No

```

describe config {#describe_config}

显示成员的配置。

可用性: Online. 您必须在 gfsh 中连接到 JMX Manager 成员才能使用此命令。

句法:

```
1 | describe config --member=value [--hide-defaults(=value)?]
```

参数, 描述配置:

名称	描述	默认值
--member	要显示其配置的成员的名称或ID。	
--hide-defaults	是否使用默认值隐藏属性的配置信息。	true

示例命令:

```
1 | describe config --member=Member1;
```

示例输出:

```

1 | gfsh>describe config --member=server1
2 | Configuration of member : "server1"
3 |
4 |
5 | JVM command line arguments
6 | -----
7 | -Dgemfire.mcast-port=0
8 | -Dgemfire.locators=localhost[10334]
9 |
10 | GemFire properties defined using the API
11 | .....
12 | log-file           : vf.gf.server.log
13 | name              : server1
14 |
15 | GemFire properties defined at the runtime
16 | .....
17 | log-level          : finest
18 | statistic-sampling-enabled : true
19 |
20 | Cache attributes
21 | .....
22 | is-server         : true
23 |
24 | Cache-server attributes
25 | . bind-address    : localhost

```

describe connection {#describe_connection}

显示连接信息详细信息

可用性: Online. 您必须在 `gfsh` 中连接到 JMX Manager 成员才能使用此命令。

句法:

```
1 | describe connection
```

示例命令:

```
1 | describe connection
```

示例输出:

```
1 | gfsh>describe connection
2 | Connection Endpoints
3 | -----
4 | GemFireUser[1099]
```

describe disk-store {#describe_disk_store}

显示有关成员磁盘存储的信息。

可用性: Online. 您必须在 `gfsh` 中连接到 JMX Manager 成员才能使用此命令。

句法:

```
1 | describe disk-store --member=value --name=value
```

参数, 描述磁盘存储:

名称	描述
--member	<i>Required.</i> 具有要描述的磁盘存储的成员的名称/ID。
--name	<i>Required.</i> 要描述的磁盘存储的名称。

示例命令:

```
1 | describe disk-store --member=server1 --name=DiskStore1
```

示例输出:

```
1 | gfsh>describe disk-store --name=disk1 --member=server1
2 | Disk Store ID           : a531bc7b-5188-4510-85d7-de7de30c6671
3 | Disk Store Name        : disk1
4 | Member ID              : ubuntu(server1:7467)<v1>:35249
5 | Member Name            : server1
6 | Allow Force Compaction  : No
7 | Auto Compaction        : Yes
8 | Compaction Threshold   : 50
9 | Max Oplog Size         : 1024
10 | Queue Size             : 0
```

```

11 Time Interval                : 1000
12 Write Buffer Size            : 32768
13 Disk Usage Warning Percentage : 90
14 Disk Usage Critical Percentage : 99
15 PDX Serialization Meta-Data Stored : No
16
17
18      Disk Directory          | Size
19 -----|-----
20 /home/user/server1/DiskStore1 | 2147483647

```

describe jndi-binding {#describe_jndi_binding}

打印描述JDBC连接的配置信息。

可用性: Online. 您必须在 gfsh 中连接到JMX Manager成员才能使用此命令。

句法:

```
1 describe jndi-binding --name=value
```

参数, 描述jndi绑定:

名称	描述
--name	<i>Required.</i> 要描述的JNDI绑定的名称。

示例命令:

```
1 describe jndi-binding --name=jndi1
```

示例输出:

```

1 gfsh>describe jndi-binding --name=jndi1
2   Property      | value
3 -----|-----
4 type            | SimpleDataSource
5 jndi-name       | jndi1
6 jdbc-driver-class | org.apache.derby.jdbc.EmbeddedDriver
7 user-name       |
8 connection-url  | jdbc:derby:newDB

```

describe lucene index {#describe_lucene_index}

描述一个Lucene索引。

另请参阅[create lucene index](#), [destroy lucene index](#), [list lucene indexes](#) 和 [search lucene](#).

可用性: Online. 您必须在 gfsh 中连接到JMX Manager成员才能使用此命令。

句法:

```
1 describe lucene index --name=value --region=value
```

参数, 描述lucene索引:

名称	描述
--name	<i>Required.</i> 要描述的Lucene索引的名称
--region	<i>Required.</i> Lucene索引所在区域的名称和路径

示例命令:

```
1 | gfsh>describe lucene index --name=personIndex --region=/Person
```

示例输出:

```
1 | gfsh>describe lucene index --name=personIndex --region=/Person
2 | Index Name | Region Path | Indexed Fields |
   | Field Analyzer | Status | Query Executions | Updates | Commits |
   | Documents
3 | ----- | ----- | ----- | ----- | ----- |
   | ----- | ----- | ----- | ----- | ----- |
4 | personIndex | /Person | [name, email, address, streetAddress, revenue] |
   | {} | Initialized | 339 | 1008 | 962 | 1004
5 |
6 | gfsh>describe lucene index --name=analyzerIndex --region=/Person
7 | Index Name | Region Path | Indexed Fields | Field
   | Analyzer | Status | Query Executions | Updates | Commits |
   | Documents
8 | ----- | ----- | ----- | ----- | ----- |
   | ----- | ----- | ----- | ----- | ----- |
9 | analyzerIndex | /Person | [address, name, email] |
   | {address=MyCharacterAnalyzer, email.. | Initialized | 1695 | 1008
   | 962 | 1004
```

describe member {#describe_member}

显示具有给定名称/ID的成员的详细信息。

可用性: Online. 您必须在 gfsh 中连接到JMX Manager成员才能使用此命令。

句法:

```
1 | describe member --name=value
```

参数, 描述成员:

名称	描述
--name	<i>Required.</i> 显示有关成员的信息, 包括姓名, ID, 组, 区域等。

示例命令:

```
1 | describe member --name=server1
```

示例输出:

```

1  gfsh>describe member --name=server1
2  Name      : server1
3  Id        : GemFireUser(server1:240)<v1>:64871
4  Host      : 192.0.2.0
5  Regions   : region4
6             region5
7             region3
8             region2
9             region1
10
11  PID       : 240
12  Groups    :
13  Used Heap : 5M
14  Max Heap  : 123M
15  Working Dir : c:\PivotalGemFire70\Latest\server1
16  Log file   : C:\PivotalGemFire70\Latest\server1\vf.gf.server.log
17  Locators   : localhost[10334]
18
19  Server Bind      : localhost
20  Server Port      : 40404
21  Running          : true
22  Client Connections : 0

```

describe offline-disk-store {#describe_offline_disk_store}

显示有关脱机成员磁盘存储的信息。

可用性: Online. 您必须在 gfsh 中连接到 JMX Manager 成员才能使用此命令。

句法:

```

1  describe offline-disk-store --name=value --disk-dirs=value[,value)* [--
   pdx=value] [--region=value]

```

参数, 描述offline-disk-store:

名称	描述
--name	<i>Required.</i> 要描述的磁盘存储的名称。
--disk-dirs	<i>Required.</i> 包含磁盘存储文件的目录。
--pdx	如果设置（或设置为true），则显示存储在磁盘存储中的所有pdx类型。
--region	要描述的磁盘存储区中的区域的名称和路径。

示例命令:

```

1  describe offline-disk-store --name=DiskStore1\
2     --disk-dirs=/home/username/gemfire/mydiskStore1Dir
3
4  describe offline-disk-store --name=DiskStore1 --disk-dirs=/DiskDir1 --
   pdx=true

```

示例输出:

```


```

```

1  gfsh>describe offline-disk-store --name=DiskStore1 --disk-dirs=/DiskDir1 --
   pdx=true
2  Regions in the disk store:
3    /PdxTypes: -lru=none -concurrencyLevel=16 -initialCapacity=16 -
   loadFactor=0.75 -compressor=none -statisticsEnabled=false
4    /Region1: -lru=none -concurrencyLevel=16 -initialCapacity=16 -
   loadFactor=0.75 -compressor=none -statisticsEnabled=false
5  PDX Types:
6    com.app.data.PositionPdx: id=1
7      long avg20DaysVol;
8      String bondRating;
9      double convRatio;
10     String country;
11     double delta;
12     long industry;
13     long issuer;
14     double mktValue;
15     double qty;
16     String secId; // identity
17     String secIdIndexed;
18     String secLinks;
19     double sharesOutstanding;
20     String underlying;
21     long volatility;
22     int pid;
23     int portfolioId;
24     com.app.data.StockPdx: id=2
25       int ID; // identity
26       String pkid;
27       Object position1;
28       Object position2;
29       Object positions;
30       Object collectionHolderMap;
31       String type;
32       String status;
33       String[] names;
34       String description;
35       long createTime;
36       Object[] position3;
37       Object aDay;
38       Date date;
39   PDX Enums:
40     com.app.data.StockPdx$Day.Monday

```

describe region {#describe_region}

显示区域的属性和关键信息。

可用性: Online. 您必须在 gfsh 中连接到 JMX Manager 成员才能使用此命令。

句法:

```
1 | describe region --name=value
```

参数, 描述区域:

名称	描述
--name	<i>Required.</i> 要描述的区域名称/路径。

示例命令:

```
1 | describe region --name=region1
```

示例输出:

```
1 | gfsh>describe region --name=Region1
2 | .....
3 | Name : Region1
4 | Data Policy : persistent replicate
5 | Hosting Members : server-5
6 | server-4
7 | server-3
8 | server-2
9 |
10 | Non-Default Attributes Shared By Hosting Members
11 |
12 | Type | Name | Value
13 | ---- | -
14 | Region | data-policy | PERSISTENT_REPLICATE
15 | | disk-store-name | DiskStore1
16 | | size | 0
17 | | scope | distributed-ack
18 |
19 | .....
20 | Name : Region1
21 | Data Policy : empty
22 | Accessor Members : server-1
23 |
24 | Non-Default Attributes Shared By Accessor Members
25 |
26 | Type | Name | Value
27 | ---- | -
28 | Region | data-policy | EMPTY
29 | | size | 0
30 | | scope | distributed-ack
```

destroy {#destroy}

删除或取消注册功能，删除索引，磁盘存储和区域。

- **destroy async-event-queue**
销毁异步事件队列。
- **destroy disk-store**
删除磁盘存储以及磁盘存储使用的磁盘上的所有文件。
- **destroy function**
销毁或取消注册功能。

- **destroy gateway-sender**
销毁网关发件人。
- **destroy index**
销毁或删除指定的索引。
- **destroy jndi-binding**
销毁指定的JNDI绑定。
- **destroy lucene index**
销毁或删除指定的Lucene索引。
- **destroy region**
销毁或删除某个地区。

destroy async-event-queue {#destroy_async_event_queue}

销毁异步事件队列。

可用性: Online. 您必须在 `gfsh` 中连接到JMX Manager成员才能使用此命令。

句法:

```
1 | destroy async-event-queue --id=value [--groups=value(,value)*] [--if-exists=value]
```

参数, 销毁async-event-queue:

名称	描述
--id	<i>Required.</i> 要删除的异步事件队列的ID.
--groups	将销毁异步事件队列的成员组。如果未指定组, 则会在所有成员上销毁队列.
--if-exists	如果指定的异步事件队列不存在, gfsh将响应该消息。如果此参数为true, 则响应的前缀为“跳过:”标签。对脚本测试很有用。默认 (如果未指定参数) : false。默认值 (如果指定的参数没有值) : true。

示例命令:

```
1 | destroy async-event-queue --id=myAsyncEventQueue
```

destroy disk-store {#destroy_disk_store}

删除磁盘存储以及磁盘存储使用的磁盘上的所有文件。先前使用此磁盘存储的封闭区域的数据将丢失。

可用性: Online. 您必须在 `gfsh` 中连接到JMX Manager成员才能使用此命令。

句法:

```
1 | destroy disk-store --name=value [--groups=value(,value)*] [--if-exists=value]
```

参数, 销毁磁盘存储:

名称	描述
--name	<i>Required.</i> 要删除的磁盘存储的名称。
--groups	将销毁磁盘存储的成员组。如果未指定任何组，则会在所有成员上销毁磁盘存储。
--if-exists	如果指定的磁盘存储不存在，gfsh将响应该消息。如果此参数为true，则响应的前缀为“跳过：”标签。对脚本测试很有用。默认（如果未指定参数）：false。默认值（如果指定的参数没有值）：true。

示例命令:

```
1 | destroy disk-store --name=store1
```

示例输出:

```
1 | gfsh>destroy disk-store --name=store1
2 | Member | Result
3 | ----- | -----
4 | server1 | Success
```

destroy function {#destroy_function}

销毁或取消注册一个函数。

默认设置是从所有成员取消注册该函数。

可用性: Online. 您必须在 gfsh 中连接到JMX Manager成员才能使用此命令。

句法:

```
1 | destroy function --id=value [--groups=value(,value)*] [--member=value]
```

参数, 销毁函数:

名称	描述
--id	<i>Required.</i> 唯一函数标识符。使用 <code>list functions</code> 命令获取ID。
--groups	一个或多个成员组，从中取消注册此函数。
--member	将从中取消注册此函数的成员的名称或ID。

示例命令:

```
1 | (1) destroy function --id=InterestCalculations
2 | (2) destroy function --id=InterestCalculations --member=server1
3 | (3) destroy function --id=InterestCalculations --group=Group1
```

destroy gateway-sender {#destroy_gateway_sender}

销毁区域不再使用的网关发件人。

默认设置是在所有成员上销毁网关发件人。

没有区域可以附加到要销毁的网关发送方。如果仍然附加了某个区域，系统会发出类似于以下内容的错误消息：

```
1 | ERROR: The GatewaySender ParallelGatewaySender{id=1n,remoteDsId=2,isRunning
   | =false}
2 | could not be destroyed as it is still used by region(s).
```

使用类似于以下命令的命令从区域中删除网关发件人：

```
1 | gfsh>alter region --name=regionA --gateway-sender-id=""
```

可用性: Online. 您必须在 gfsh 中连接到 JMX Manager 成员才能使用此命令。

句法:

```
1 | destroy gateway-sender --id=value [--groups=value(,value)*]
2 | [--members=value(,value)*] [--if-exists=value]
```

参数，销毁网关发送者:

名称	描述
--id	<i>Required.</i> 唯一网关发件人标识 使用 <code>list gateways</code> 命令获取ID。
--groups	将从中销毁此网关发件人的一个或多个成员组。
--members	将从中销毁此网关发件人的成员的名称或ID。
--if-exists	如果指定的网关发送方不存在，gfsh将响应该消息。如果此参数为true，则响应的前缀为“跳过：”标签。对脚本测试很有用。默认（如果未指定参数）：false。默认值（如果指定的参数没有值）：true。

示例命令:

```
1 | destroy gateway-sender --id=SiteASender
```

destroy index {#destroy_index}

销毁或删除指定的索引。

可用性: Online. 您必须在 gfsh 中连接到 JMX Manager 成员才能使用此命令。

句法:

```
1 | destroy index [--name=value] [--region=value] [--members=value(,value)*]
2 | [--groups=value(,value)*] [--if-exists=value]
```

注意: 您必须至少指定一个参数选项。如果输入 `destroy index` 而没有任何参数，该命令将要求您指定至少一个选项。

参数，销毁索引:

名称	描述
--name	要删除的索引的名称。
--members	要删除索引的成员的ID。
--region	要从中销毁索引或所有索引的区域的名称。
--groups	将删除该组中所有成员的索引。
--if-exists	如果指定的索引不存在，gfsh将响应该消息。如果此参数为true，则响应的前缀为“跳过：”标签。对脚本测试很有用。默认（如果未指定参数）：false。默认值（如果指定的参数没有值）：true。

示例命令:

```
1 destroy index --members=server2
2 destroy index --name=MyKeyIndex
```

destroy jndi-binding {#destroy_jndi_binding}

销毁包含XA数据源配置的指定JNDI绑定。

可用性: Online. 您必须在 gfsh 中连接到JMX Manager成员才能使用此命令。

句法:

```
1 destroy jndi-binding --name=value [--if-exists=value]
```

参数, 破坏jndi绑定:

名称	描述
--name	<i>Required.</i> 要销毁的JNDI绑定的名称。
--if-exists	当指定的JNDI绑定不存在时，跳过destroy操作。如果没有此选项，则会因不存在的JNDI绑定的规范而导致错误。默认（如果未指定参数）：false。默认值（如果指定的参数没有值）：true。

示例命令:

```
1 destroy jndi-binding --name=jndi1
```

destroy lucene index {#destroy_lucene_index}

销毁或删除指定的Lucene索引。

参见 [create lucene index](#), [describe lucene index](#), [list lucene indexes](#) 和 [search lucene](#).

可用性: Online. 您必须在 gfsh 中连接到JMX Manager成员才能使用此命令。

句法:

```
1 destroy lucene index --region=value [--name=value]
```

参数, 销毁lucene索引:

名称	描述
--region	<i>Required.</i> 要从中删除索引的区域的名称。如果未指定 <code>--name</code> 选项，则销毁与该区域关联的所有索引。
--name	要删除的索引的名称。

示例命令:

```
1 | destroy lucene index --region=region1
2 | destroy lucene index --region=region1 --name=MyKeyIndex
```

destroy region {#destroy_region}

销毁或删除某个地区。

可用性: Online. 您必须在 `gfsh` 中连接到 JMX Manager 成员才能使用此命令。

句法:

```
1 | destroy region --name=value [--if-exists=value]
```

参数, 销毁区域:

名称	描述
--name	<i>Required.</i> 要删除的区域的名称和路径。
--if-exists	如果指定的区域不存在, <code>gfsh</code> 将响应该消息。如果此参数为 <code>true</code> , 则响应的前缀为“跳过:”标签。对脚本测试很有用。默认 (如果未指定参数): <code>false</code> 。默认值 (如果指定的参数没有值): <code>true</code> 。

示例命令:

```
1 | destroy region --name=region4
2 | destroy region --name=/region1/subregion1
```

示例输出:

```
1 | gfsh>destroy region --name=region1
2 | "region1" destroyed successfully.
```

disconnect {#disconnect}

关闭所有活动连接。

可用性: Online. 您必须在 `gfsh` 中连接到 JMX Manager 成员才能使用此命令。

句法:

```
1 | disconnect
```

示例命令:

```
1 | disconnect
```

示例输出:

```
1 | gfsh>disconnect
2 | Disconnecting from: Locator1[1099]
3 | Disconnected from : Locator1[1099]
```

错误消息:

```
1 | Error occurred while disconnecting: {0}
2 | Not connected!
```

echo {#echo}

回显给定文本，其中可能包括系统和用户变量。

该命令还可以回显gfsh环境属性（使用 ['set variable'](#)命令）如果变量名称 预先加上 `$` - 就像UNIX一样。

有关gfsh环境变量的列表，请参阅[有用的gfsh Shell变量](#)。

可用性: Online or offline.

句法:

```
1 | echo [--string=value]
```

名称	描述
--string	要回显的字符串。 例如， <code>SYS_USER</code> 变量设置为 <code>\${SYS_USER}</code> 。

Table 1. 回显参数

示例命令:

```
1 | echo --string="Hello world!"
2 | echo --string="Hello world! This is ${SYS_USER}"
3 | echo --string=${APP_FETCH_SIZE}
```

要查看shell中设置的所有变量:

```
1 | echo --string=$*
```

示例输出:

```
1 | gfsh>echo --string=${SYS_JAVA_VERSION}
2 | Post substitution: echo --string=1.8.0_60
3 | 1.8.0_60
```

execute function {#execute_function}

在成员或区域上执行函数。

执行函数

执行具有指定ID的函数。默认情况下，该函数在所有成员上执行。

可用性: Online. 您必须在 `gfsh` 中连接到JMX Manager成员才能使用此命令。

句法:

```
1 execute function --id=value [--groups=value(,value)*]
2 [--members=value(,value)*] [--region=value]
3 [--arguments=value(,value)*] [--result-collector=value] [--filter=value]
```

--id	<i>Required.</i> ID of the function to execute.
--groups	One or more groups of members on which this function should be executed.
--members	Name/ID of the member(s) on which the function will be executed.
--region	Region on which the data dependent function will be executed.
--arguments	Arguments to the function in comma separated string format.
--result-collector	Fully qualified class name of the <code>ResultCollector</code> to instantiate for gathering results.
--filter	Key list which causes the function to only be executed on members which have entries with these keys.

Table 1. 执行函数参数

示例命令:

```
1 execute function --id=InterestCalculations --region=/InterestRegion
2 execute function --id=InterestCalculations --members=server1
3 execute function --id=InterestCalculations --groups=Group1
```

exit {#exit}

退出 `gfsh` shell。您也可以使用 `quit` 退出shell。

退出gfsh shell并返回OS shell。

可用性: Online or offline.

句法:

```
1 | exit
```

示例命令:

```
1 | exit
```

export {#export}

导出配置，数据，日志和堆栈跟踪。

- **export cluster-configuration**
导出包含配置和操作集群所需的 `cache.xml` 文件，`gemfire.properties` 文件和应用程序JAR文件的集群配置ZIP文件。
- **export config**
导出成员或成员的配置属性。
- **export data**
将用户数据从区域导出到文件。
- **export logs**
将日志导出到给定目录。
- **export offline-disk-store**
将区域数据从脱机磁盘存储导出到gemfire快照文件。
- **export stack-traces**
导出一个或多个成员的堆栈跟踪。

export cluster-configuration {#export_cluster_configuration}

使用群集配置导出单个XML文件或ZIP文件，其中包含配置和操作群集所需的 `cache.xml` 文件，`gemfire.properties` 文件和应用程序JAR文件。

如果既未指定文件名也未指定ZIP文件名，则群集配置将写入标准输出。

可用性: Online. 您必须在 `gfsh` 中连接到JMX Manager成员才能使用此命令。

参见 [Overview of the Cluster Configuration Service](#).

句法:

```
1 | export cluster-configuration [--group(=value)?] [--xml-file=value]
2 |   [--zip-file-name=value]
```

导出群集配置参数:

名称	描述	默认值
--group	导出指定服务器组的配置。如果未指定组，请使用默认的 <code>cluster</code> 组。	
--xml-file	用于包含导出的群集配置的文件名。也可能包括绝对或相对路径。只指定 <code>--xml-file</code> 或 <code>--zip-file-name</code> 中的一个。	
--zip-file-name	包含导出的群集配置的ZIP文件的文件名。也可能包括绝对或相对路径。只指定 <code>--xml-file</code> 或 <code>--zip-file-name</code> 中的一个。	

示例命令:

```

1 gfsh>export cluster-configuration --zip-file-name=/group/shared-
  configs/devClusterConfig.zip
2 gfsh>export cluster-configuration --zip-file-name=my-
  configs/myClusterConfig.zip
3 gfsh>export cluster-configuration --zip-file-name=myClusterConfig.zip
4 gfsh>export cluster-configuration --xml-file=Cluster3Config.xml

```

示例输出:

```

1 gfsh>export cluster-configuration --zip-file-name=mySharedConfig.zip
2 Downloading cluster configuration : /home/username/gemfire/mySharedConfig.zip

```

export config {#export_config}

导出成员或成员的配置属性。

如果未指定任何参数，则将导出所有成员配置。

可用性: Online. 您必须在 `gfsh` 中连接到JMX Manager成员才能使用此命令。

句法:

```

1 export config [--members=value(,value)*] [--groups=value(,value)*]
2 [--dir=value]

```

导出配置参数:

名称	描述
--members	将导出其配置的成员的名称/ID。
--groups	将导出其配置的成员组。
--dir	将写入导出的配置文件的目录。

示例命令:

```

1 export config
2 export config --members=member1

```

示例输出:

```
1 gfsh>export config --members=member1
2 Downloading Cache XML file: c:\PivotalGemFire\Latest\.\member1-cache.xml
3 Downloading properties file: c:\PivotalGemFire\Latest\.\member1-gf.properties
```

export data {#export_data}

将用户数据从一个区域导出到一个或多个文件。

可用性: 线上。您必须在 gfsh 中连接到 JMX Manager 成员才能使用此命令。

句法:

要将数据从任何区域（已复制或已分区）按顺序导出到单个文件，请使用 `--file` 选项:

```
1 export data --region=value --member=value --file=value
```

在此方案中，来自复制和分区区域的数据将按顺序导出到指定成员上的单个文件中。

对于分区区域，可以使用 `--parallel` 选项加快导出过程:

```
1 export data --parallel --region=value --member=value --dir=value
```

在这种情况下，分区区域数据在所有托管节点上同时导出到这些相应节点的本地目录。

- 并行导出仅适用于分区区域。
- `--file` 和 `--dir` 选项是互斥的; `--file` 只能用于串行导出, `--dir` 可用于串行和并行导出。

导出数据参数:

名称	描述
<code>--region</code>	<i>Required.</i> 要从中导出数据的区域。
<code>--member</code>	<i>Required.</i> 托管该区域的成员的 Name/ID。在串行导出中，所有数据都将导出到运行该成员的主机上的指定文件中。在并行导出中，来自部分托管在此成员上的分区区域的数据将为每个分区导出到承载这些分区的节点上的文件。
<code>--file</code>	要写入导出数据的文件。该文件必须具有 <code>.gfd</code> 的扩展名。不能与 <code>--dir</code> 同时指定，不能与 <code>--parallel</code> 一起使用。
<code>--dir</code>	要写入导出数据的目录。如果 <code>--parallel</code> 为 true，则为必需。不能同时指定 <code>--file</code> 。
<code>--parallel</code>	将每个节点上的本地数据导出到该计算机上的目录。仅适用于分区区域。

示例命令:

```
1 export data --region=region2 --file=region2_20121001.gfd --member=server2
```

样本输出:

```
1 gfsh>export data --region=region2 --file=region2_20121001.gfd --
  member=server1
2 Data succesfully exported from region : region2 to file : C:\PivotalGemFire\
3 Latest\server1\region2_20121001.gfd on host : 192.0.2.0
```

export logs {#export_logs}

将日志导出到给定目录。

将导出具有指定时间范围内日志的所有文件。如果未指定时间范围，则将导出所有日志。

`--dir` 参数指定将写入日志文件的本地目录。仅在使用http连接导出日志时使用。如果通过http执行，则zip存档将保存在用户客户端计算机上的指定目录中。如果未指定，则将日志写入 `user.dir` 系统属性指定的位置。当命令在JMX上执行时，日志将在连接的定位器的工作目录中保存为 `exportedlogs_xxx.zip`。

可用性: Online. 您必须在 `gfsh` 中连接到JMX Manager成员才能使用此命令。

句法:

```
1 export logs [--dir=value] [--groups=value(,value)*] [--members=value(,value)*]
2 [--log-level=value] [--only-log-level=value] [--merge-log=value]
3 [--start-time=value] [--end-time=value] [logs-only(=value)?]
4 [--stats-only(=value)?] [--file-size-limit(=value)?]
```

导出日志参数:

名称	描述	默认值
<code>--dir</code>	使用http连接导出日志时将写入日志文件的本地目录。在JMX上执行命令时忽略。	
<code>--groups</code>	将导出其日志文件的成员组。	
<code>--members</code>	将导出其日志文件的成员的名称/ID。	
<code>--log-level</code>	要导出的最低日志条目级别。有效值包括: <code>OFF</code> , <code>FATAL</code> , <code>ERROR</code> , <code>WARN</code> , <code>INFO</code> , <code>DEBUG</code> , <code>TRACE</code> 和 <code>ALL</code> 。	<code>INFO</code>
<code>--only-log-level</code>	是否仅包含与指定的 <code>--log-level</code> 完全匹配的条目。	<code>false</code>
<code>--merge-log</code>	导出到目标目录后是否合并日志（不建议使用）。	<code>false</code>
<code>--start-time</code>	将导出此时间之后发生的日志条目。格式: <code>yyyy/MM/dd/HH/mm/ss/SSS/z</code> 或者 <code>yyyy/MM/dd</code>	<code>no limit</code>
<code>--end-time</code>	将导出在此时间之前发生的日志条目。格式: <code>yyyy/MM/dd/HH/mm/ss/SSS/z</code> 或者 <code>yyyy/MM/dd</code>	<code>no limit</code>
<code>--logs-only</code>	是否仅导出日志（不是统计信息）	如果未指定参数: <code>false</code> 。如果指定的参数没有值: <code>true</code>
<code>--stats-only</code>	是否仅导出统计信息（不是日志）	如果未指定参数: <code>false</code> 。如果指定的参数没有值: <code>true</code>
<code>--file-size-limit</code>	限制导出文件的总解压缩大小。指定0（零）无限制。默认值为兆字节，或者可以指定[k, m, g, t]。	如果未指定参数: <code>100m</code> 。如果指定的参数没有值: <code>0</code>

示例命令，显示输出:

```

1 | gfsh>export logs --dir=data/logs
2 | Logs exported to the connected member's file system: /my-
  | locator/data/logs/exportedLogs_1489513007261.zip
3 | gfsh>export logs --dir=data/logs --file-size-limit=1k
4 | Estimated exported logs expanded file size = 95599, file-size-limit = 1024.
5 | To disable exported logs file size check use option "--file-size-limit=0".
6 |
7 | gfsh>export logs --dir=data/logs --file-size-limit=99k
8 | Logs exported to the connected member's file system: /my-
  | locator/data/logs/exportedLogs_1489513007261.zip

```

export offline-disk-store {#export_offline_disk_store}

将区域数据从脱机磁盘存储导出到gemfire快照文件。

可用性: Online or offline.

句法:

```

1 | export offline-disk-store --name=value --disk-dirs=value(,value)* --dir=value

```

导出脱机磁盘存储参数:

名称	描述
--name	<i>Required.</i> 要导出的磁盘存储的名称。
--disk-dirs	包含磁盘存储文件的目录。
--dir	将快照文件导出到的目录。

示例命令:

```

1 | export offline-disk-store --name= Diskstore1 \
2 | --disk-dirs=/home/username/gemfire/mydiskStore1Dir --
  | dir=/home/username/gemfire/export

```

export stack-traces {#export_stack_traces}

导出一个或多个成员的堆栈跟踪。

可用性: Online. 您必须在 gfsh 中连接到JMX Manager成员才能使用此命令。

句法:

```

1 | export stack-traces --file=value [--members=value(,value)*] [--
  | groups=value(,value)*]

```

导出堆栈跟踪参数:

名称	描述
--file	<i>Required.</i> 要写入堆栈跟踪的文件名。
--members	将导出其日志文件的成员的名称或ID。
--groups	将导出其日志文件的成员组。

示例命令:

```
1 | export stack-traces --file=stack.txt
```

示例输出:

```
1 | gfsh>export stack-traces --file=stack.txt
2 | stack-trace(s) exported to file: C:\PivotalGemFire\Latest\locator1\stack.txt
3 | On host : GemFireStymon
```

gc {#gc}

对一个或多个成员强制GC（垃圾收集）。

默认设置是在所有缓存成员上进行垃圾收集。

可用性: Online. 您必须在 gfsh 中连接到JMX Manager成员才能使用此命令。

句法:

```
1 | gc [--groups=value(,value)*] [--member=value]
```

名称	描述
--groups	将强制进行垃圾收集的一个或多个成员组。
--member	将强制进行垃圾收集的成员的名称/ID。

Table 1. GC 参数

示例命令:

```
1 | gc --member=server1
2 | gc --groups=Group1
3 | gc
```

示例输出:

```
1 | gfsh>gc
2 | Sucessfully executed GC
```

get {#get}

显示区域中的条目。

可用性: Online. 您必须在 `gfsh` 中连接到 JMX Manager 成员才能使用此命令。

句法:

```
1 | get --key=value --region=value [--key-class=value] [--value-class=value]
```

名称	描述	默认值
--key	<i>Required.</i> 从中创建键的字符串或JSON文本。例如: "James", "100L" and "('id': '134s')".	
--region	<i>Required.</i> 从中获取条目的区域。	
--key-class	键类型的完全限定类名。	默认值是当前区域或String的键约束。
--value-class	值类型的完全限定类名。	默认值是当前区域或String的值约束。
--load-on-cache-miss	在为高速缓存未命中时检索指定Key的值时，显式启用或禁用任何已注册的CacheLoader。	true (enabled)

Table 1. Get 参数

示例命令:

```
1 | get --key=('id': '133abg124') --region=region1
2 |
3 | // Retrieving when key type is a wrapper(primitive)/String
4 | get --key=('133abg124') --region=/region1/region12 --value-
   | class=data.ProfileDetails
5 |
6 | get --key=('100L') --region=/region1/region12 --value-
   | class=data.ProfileDetails
7 | --key-class=java.lang.Long
```

示例输出:

```
1 | gfsh>get --key=('123') --region=region1
2 | Result      : true
3 | Key Class   : java.lang.String
4 | Key         : ('123')
5 | Value Class : java.lang.String
6 | Value       : ABC
```

help {#help}

显示所有可用命令的语法和用法信息。

在没有命令作为参数的情况下键入帮助会列出所有可用命令。

可用性: Online or offline.

句法:

```
1 | help [command]
```

Examples Commands:

```
1 | help
2 | help rebalance
```

示例输出:

```
1 | gfsh>help rebalance
2 | NAME
3 |     rebalance
4 | IS AVAILABLE
5 |     true
6 | SYNOPSIS
7 |     Rebalance partitioned regions. The default is for all partitioned
8 |     regions to be rebalanced.
9 | SYNTAX
10 |     rebalance [--include-region=value(,value)*] [--exclude-
11 |     region=value(,value)*] [--time-out=value] [--simulate(=value)?]
12 | PARAMETERS
13 |     include-region
14 |         Partitioned regions to be included when rebalancing.
15 |     Includes take precedence over excludes.
16 |         Required:false
17 |     exclude-region
18 |         Partitioned regions to be excluded when rebalancing.
19 |         Required:false
20 |     time-out
21 |         Time to wait (in seconds) before GFSH returns to a prompt
22 |         while rebalancing continues in the background. The default is to wait for
23 |         rebalancing to complete.
24 |         Required:false
25 |         Default if the parameter is not specified:-1
26 |     simulate
27 |         whether to only simulate rebalancing. The --time-out
28 |         parameter is not available when simulating.
29 |         Required:false
30 |         Default if no value for the parameter is given:true
31 |         Default if the parameter is not specified:false
```

hint {#hint}

显示有关主题的信息以及与主题关联的命令列表。

提供主题提示或列出所有可用主题（如果未指定主题）。

可用性: Online or offline.

句法:

```
1 | hint [topic]
```

示例命令:

```
1 | hint
2 | hint Server
```

示例输出:

```
1 | gfsh>hint
2 | Hints are available for the following topics. Use "hint <topic-name>" for a
   | specific
3 |   hint.
4 | Configuration
5 | Data
6 | Debug-Utility
7 | Disk Store
8 | Function Execution
9 | GFSH
10 | Help
11 | JMX
12 | Lifecycle
13 | Locator
14 | Management-Monitoring
15 | Manager
16 | Region
17 | Server
18 | Statistics
19 |
20 | gfsh>hint server
21 | A server is GemFire cluster member which holds a GemFire cache. Depending
   | on the
22 |   topology used it can refer to either a system that responds to client
   | requests
23 | or a system that is only a peer to other members.
24 |
25 | describe member : Display information about a member, including name, id,
   | groups
26 | , regions, etc.
27 | export logs      : Export the log files for a member or members.
28 | list members     : Display all or a subset of members.
29 | start server     : Start a GemFire Cache Server.
30 | status server    : Display the status of a GemFire Cache Server.
31 | stop server      : Stop a GemFire Cache Server..
```


history {#history}

显示或保存命令历史记录。

此历史记录可以保存到文件中，以后也可以用作脚本。

已成功执行的命令历史记录也记录在运行gfsh的用户的主目录中的 `.geode/.gfsh.history` 文件中。

可用性: Online or offline.

句法:

```
1 | history [--file=<history text file>]
```

名称	描述	默认值
--file	要保存历史记录的文件。	
--clear	设置为 true 时，清除gfsh命令的历史记录。	false

Table 1. History 参数

示例命令:

```
1 | history
2 | history --file=./mycommands.gfsh;
```

示例输出:

```
1 | gfsh>history --file=./mycommands.gfsh
2 | Wrote successfully to file ./mycommands.gfsh
```

import {#import}

您可以导入导出的群集配置以创建新群集或将数据导入区域。

- **import cluster-configuration**
导入群集配置。
- **import data**
将用户数据从文件导入区域。

import cluster-configuration {#import_cluster_configuration}

从ZIP文件或XML文件导入以前导出的群集配置。 在启动新群集时，此命令很有用。

- 在给定群集中，只有一个定位器需要执行导入。 该定位器与集群中所有其他连接的定位器共享导入的配置。
- 当服务器启动时，定位器与数据成员（服务器）共享导入的配置，或者最近启动了服务器，没有在其中定义区域，并且自启动以来没有给出任何其他配置更改。

要导入集群配置，请启动一个或多个定位器，然后运行 `gfsh`import cluster-configuration` 命令。

可用性: Online. 您必须在 `gfsh` 中连接到定位器才能使用此命令。

句法:

```
1 import cluster-configuration [--action=value] [--group(=value)?]
2 [--xml-file=value] [--zip-file-name=value]
```

Import Cluster-Configuration 参数:

名称	描述
--action	当值为 <code>APPLY</code> （默认值）时，配置将应用于没有配置的正在运行的服务器。如果任何服务器已配置，则命令失败。当值为 <code>STAGE</code> 时，配置将被覆盖，并将在将来的服务器创建期间使用；当前运行的服务器的配置不会更改。
--group	执行指定服务器组的导入。如果没有指定组，则隐含 <code>cluster</code> 。
--xml-file	用于导入群集配置的文件名。也可能包括绝对或相对路径。只指定 <code>--xml-file</code> 或 <code>--zip-file-name</code> 中的一个。
--zip-file-name	包含要导入的集群配置工件的ZIP文件的名称。只指定 <code>--xml-file</code> 或 <code>--zip-file-name</code> 中的一个。

示例命令:

```
1 gfsh>import cluster-configuration --zip-file-
  name=/home/username/myClusterConfig.zip
2 gfsh>import cluster-configuration --xml-file=configs/Cluster3Config.xml
```

示例输出:

```
1 gfsh>import cluster-configuration --zip-file-
  name=/home/username/myClusterConfig.zip
2 Cluster configuration successfully imported
```

import data {#import_data}

将用户数据从一个或多个文件导入区域。

可用性: 线上。您必须在 `gfsh` 中连接到MX Manager成员才能使用此命令。

句法:

如果数据按顺序导出到单个文件，请使用 `import` 命令的序列形式导入数据:

```
1 import data --region=value --file=value --member=value [--invoke-
  callbacks=value]
```

在此方案中，从指定成员上的单个文件顺序导入来自复制和分区区域的数据。

有关串行和并行数据导出之间差异的说明，请参阅[导出数据](#)。

如果数据是以并行格式从分区区域导出的，请使用 `--parallel` 选项导入数据：

```
1 import data --parallel --region=value --member=value --dir=value
2 [--invoke-callbacks=value]
```

在这种情况下，分区区域数据在所有托管节点上从这些相应节点的本地目录同时导入。

- 并行导入仅适用于以并行模式导出的分区区域。
- `--file` 和 `--dir` 选项是互斥的；`--file` 只能用于串行导入，`--dir` 可用于串行和并行导入。

导入数据参数：

名称	描述	缺省值
<code>--region</code>	<i>Required.</i> 将导入数据的区域.	
<code>--member</code>	<i>Required.</i> 托管该区域的成员的 Name/ID。在串行导入中，所有数据都是从运行该成员的主机上的指定文件导入的。在并行导入中，从托管这些分区的节点上的文件为每个分区导入部分托管在此成员上的分区区域的数据。	
<code>--file</code>	将从中读取导入数据的文件。该文件必须具有 <code>.gfd</code> 的扩展名，并且必须位于该成员的文件系统上（或通过NFS可访问该成员），该目标是导入。不能与 <code>--dir</code> 同时指定，不能与 <code>--parallel</code> 一起使用。	
<code>--dir</code>	从中导入数据的目录。如果 <code>--parallel</code> 为true，则为必需。不能同时指定 <code>--file</code> 。如果给定节点在指定目录中没有数据文件，则会以静默方式跳过该节点的导入操作。	
<code>--invoke-callbacks</code>	布尔值，当为true时，在数据导入期间调用回调。	false

示例命令：

```
1 import data --region=region2 --file=/mnt5/region2_20121001.gfd --
  member=server1
```

list {#list}

列出现有的Geode资源，例如已部署的应用程序，磁盘存储，功能，成员，服务器和区域。

- **list async-event-queues**
显示所有成员的异步事件队列列表。
- **list clients**
显示已连接客户端的列表。
- **list deployed**
显示使用deploy命令部署到成员的JAR列表。
- **list disk-stores**
列出Geode集群中的所有可用磁盘存储

- **list durable-cqs**

列出与指定的持久客户端ID关联的持久客户端CQ。

- **list functions**

显示已注册的函数列表。默认设置是显示所有成员的功能。

- **list gateways**

显示一个或多个成员的网关发件人和收件人。

- **list indexes**

Display the list of indexes created for all members.

- **list jndi-binding**

List all JNDI bindings, active and configured.

- **list lucene indexes**

列出为所有成员创建的Lucene索引。

- **list members**

显示全部或部分成员。

- **list regions**

显示成员或成员的区域。如果未指定参数，则列出群集中的所有区域。

list async-event-queues {#list_async_event_queues}

显示所有成员的异步事件队列列表。

可用性: Online. 您必须在 `gfsh` 中连接到JMX Manager成员才能使用此命令。

句法:

```
1 | list async-event-queues
```

示例命令:

```
1 | list async-event-queues
```

list clients {#list_clients}

显示已连接客户端的列表。

显示已连接客户端及其连接的服务器的列表。

可用性: Online. 您必须在 `gfsh` 中连接到JMX Manager成员才能使用此命令。

句法:

```
1 | list clients
```

示例命令:

```
1 | list clients
```

示例输出:

```

1 gfsh>list clients
2
3 ClientList
4
5         Client Name / ID          | Server Name / ID
6 -----|-----
7 192.0.2.0(4987:loner):58922:7b3398cf | member=server2,port=53508;
   member=server1,port=56806
8 192.0.2.0(5065:loner):39906:a6f598cf | member=server2,port=53508;
   member=server1,port=56806

```

list deployed {#list_deployed}

显示使用deploy命令部署到成员的JAR列表。

可用性: Online. 您必须在 gfsh 中连接到JMX Manager成员才能使用此命令。

句法:

```
1 list deployed [--groups=value(,value)*]
```

参数, 已部署列表:

名称	描述
--groups	将显示已部署JAR的成员组。如果未指定, 则显示所有成员的JAR。

示例命令:

```

1 list deployed
2 list deployed --groups=Group2

```

示例输出:

```

1 gfsh> list deployed --groups=Group2
2
3 Member | Deployed JAR | JAR Location
4 -----|-----|-----
5 datanode1 | group1_functions.jar |
   /usr/local/gemfire/deploy/vf.gf#group1_functions.jar#1
6 datanode2 | group1_functions.jar |
   /usr/local/gemfire/deploy/vf.gf#group1_functions.jar#1

```

错误消息:

```
1 No JAR Files Found
```

list disk-stores {#list_disk_stores}

列出Geode集群中的所有可用磁盘存储

该命令还使用磁盘存储列出已配置的磁盘目录以及任何区域，高速缓存服务器，网关，PDX序列化和异步事件队列，以便将信息溢出和/或保留到磁盘。使用 `describe disk-store` 命令查看特定磁盘存储的详细信息。

可用性: Online. 您必须在 `gfsh` 中连接到 JMX Manager 成员才能使用此命令。

句法:

```
1 | list disk-stores
```

示例命令:

```
1 | list disk-stores
```

示例输出:

```
1 gfsh> list disk-stores
2
3      Member Name      |                               Member Id                               | Disk Store
4      Name |          Disk Store ID
5      ----- | -----
6      consumerServer | 192.0.2.0(consumerServer:13825)<v5>:3545 | consumerData
7      | 4029af26-fd82-4997-bd6c-33382cdbb5e9
8      consumerServer | 192.0.2.0(consumerServer:13825)<v5>:3545 | observerData
9      | 7e0316ad-963c-49b0-9b01-8f59b8d9e29e
10     producerServer | 192.0.2.0(producerServer:13826)<v3>:53764 | producerData
11     | 4670e4eb-1c50-4465-b418-08ede3d5dbed
```

错误消息:

```
1 gfsh> list disk-stores
2 No Disk Stores Found
```

list durable-cqs {#list_durable_cqs}

列出与指定的持久客户端ID关联的持久客户端CQ。

可用性: Online. 您必须在 `gfsh` 中连接到 JMX Manager 成员才能使用此命令。

句法:

```
1 | list durable-cqs --durable-client-id=value
2 | [--members=value(,value)*] [--groups=value(,value)*]
```

参数, 列出 durable-cqs:

名称	描述
--durable-client-id	<i>Required.</i> 用于标识持久客户端的ID。
--members	注册持久客户端的成员的名称或ID以及将显示持久CQ。
--groups	将显示持久客户注册的成员组和持久CQ。

示例命令:

```
1 | list durable-cqs --durable-client-id=client1
```

样本输出:

```
1 | gfsh>list durable-cqs --durable-client-id=client1
2 | member | durable-cq-name
3 | ----- | -----
4 | server3 | cq3
5 |          | cq1
6 |          | cq2
7 | server4 | cq3
8 |          | cq1
```

错误消息:

```
1 | gfsh>list durable-cqs --durable-client-id=client1
2 |
3 | Unable to list durable-cqs for durable-client-id : "client1" due to following
4 | reasons.
5 | No client found with client-id : client1
6 | Occurred on members
7 | 1.server4
8 | 2.server1
9 | 3.server3
```

list functions {#list_functions}

显示已注册的函数列表。默认设置是显示所有成员的函数。

可用性: Online. 您必须在 gfsh 中连接到 JMX Manager 成员才能使用此命令。

句法:

```
1 | list functions [--matches=value] [--groups=value(,value)*]
2 | [--members=value(,value)*]
```

参数, 列表函数:

名称	描述
--matches	函数ID必须匹配才能包含的模式。使用Java模式匹配规则, 而不是UNIX。例如, 要多次匹配任何字符, 请使用“.”而不是“*”。
--groups	将显示函数的成员组。对多个组使用逗号分隔列表。
--members	将显示函数的成员的名称或ID。对多个成员使用逗号分隔列表。

示例命令:

```
1 | gfsh> list functions
2 | gfsh> list functions --matches=reconcile.*
```

示例输出:

```
1 list functions
2
3 Member | Function
4 -----|-----
5 camelot | loadDataFromExternalSource
6 camelot | reconcileWeeklyExpenses
7 excalibur | loadDataFromExternalSource
8 excalibur | reconcileDailyExpenses
9
10
11 Example of 'list function' with a "matches" filter:
12
13 gfsh> list functions --matches=reconcile.*
14
15 Member | Function
16 -----|-----
17 camelot | reconcileWeeklyExpenses
18 excalibur | reconcileDailyExpenses
19
20
21 Example of 'list functions' when no functions are found in Geode :
22
23 gfsh> list functions
24 No Functions Found.
```

list gateways {#list_gateways}

显示一个或多个成员的网关发件人和收件人。

可用性: Online. 您必须在 gfsh 中连接到 JMX Manager 成员才能使用此命令。

句法:

```
1 list gateways [--members=value(,value)*] [--groups=value(,value)*]
```

参数, 列出网关:

名称	描述
--members	网关发送者和接收者显示的成员。
--groups	将显示网关发件人和接收者的成员组。对多个组使用逗号分隔列表。

示例命令:

```
1 list gateways
```

list indexes {#list_indexes}

显示为所有成员创建的索引列表。

可用性: Online. 您必须在 gfsh 中连接到 JMX Manager 成员才能使用此命令。

句法:


```
1 | list indexes [--with-stats(=value)?]
```

参数，列表索引:

名称	描述	默认值
--with-stats	指定是否还应显示统计信息。	false

示例命令:

```
1 | list indexes
2
3 | list indexes --with-stats
```

示例输出:

```
1 | gfsh>list indexes
2 | Member Name      | Member ID      | Region
3 | Path | Name | Type | Indexed Expression | From Clause
4 | -----|-----|-----|-----|-----|-----
5 | - | - | - | - | - | -
6 | consumerServer | 192.0.2.0(consumerServer:13873):6317 | /consumers |
7 | cidIdx | KEY | id | /consumers
8 | consumerServer | 192.0.2.0(consumerServer:13873):6317 | /consumers |
9 | cnameIdx | RANGE | name | /consumers
10 | producerServer | 192.0.2.0(producerServer:13874):19198 | /producers |
11 | pidIdx | RANGE | id | /producers
12
13 | Example of 'list indexes' with stats printed:
14
15 | gfsh>list indexes --with-stats
16
17 | Member Name | Member ID | Region Path | Name | Type | Indexed
18 | Expression | From Clause | Uses | Updates | Update Time | Keys | Values
19 | -----|-----|-----|-----|-----|-----|-----
20 | - | - | - | - | - | - | -
21 | cs... | 192... | /consumers | cidIdx | KEY | id
22 | | /consumers | 2512 | 0 | 0 | 5020 | 5020
23 | cs... | 192... | /consumers | cnameIdx | RANGE | name
24 | | /consumers | 0 | 5020 | 421224000 | 0 | 5020
25 | ps... | 192... | /producers | pidIdx | RANGE | id
26 | | /producers | 0 | 5031 | 497872000 | 5031 | 5031
```

错误消息:

在Geode中找不到索引时的输出示例:

```
1 | gfsh> list indexes
2 | No Indexes Found
```

list jndi-binding {#list_jndi_binding}

列出所有JNDI绑定，活动和配置。活动绑定是绑定到服务器的JNDI上下文的绑定，也在群集配置中列出。已配置的绑定是群集配置中列出的绑定，但可能在服务器上不活动。

可用性: Online. 您必须在 `gfsh` 中连接到JMX Manager成员才能使用此命令。

句法:

```
1 | list jndi-binding
```

示例输出:

```
1 gfsh>list jndi-binding
2
3 Configured JNDI bindings:
4
5 Group Name | JNDI Name | JDBC Driver Class
6 ----- | ----- | -----
7 cluster    | jndi1     | org.apache.derby.jdbc.EmbeddedDriver
8
9 Active JNDI bindings found on each member:
10
11 Member | JNDI Name | JDBC Driver Class
12 ----- | ----- | -----
13
14 land-gifted-gun | java:UserTransaction |
   org.apache.geode.internal.jta.UserTransactionImpl
15 land-gifted-gun | java:TransactionManager |
   org.apache.geode.internal.jta.TransactionManagerImpl
```

list lucene indexes {#list_lucene_indexes}

显示为所有成员创建的Lucene索引列表。 可选的 `--with-stats` 限定符显示索引上的活动。

参见 [create lucene index](#), [describe lucene index](#), [destroy lucene index](#) and [search lucene](#).

可用性: Online. 您必须在 `gfsh` 中连接到JMX Manager成员才能使用此命令。

句法:

```
1 | list lucene indexes [--with-stats(=value)]
```

参数, 列出lucene索引:

Name	Description	Default Value
--with-stats	Specifies whether statistics should also be displayed.	false if not specified, true if specified

示例命令:

```
1 | list lucene indexes
```

示例输出:

```
1 gfsh>list lucene indexes --with-stats
2 Index Name | Region Path | Indexed Fields | Field Analy.. | Status
3 | Query Executions | Updates | Commits | Documents
4 ----- | ----- | ----- | ----- | -----
5 | ----- | ----- | ----- | ----- | -----
6 testIndex | /testRegion | [__REGION_VALUE_FIELD] | {__REGION_V.. | Defined
7 | NA | NA | NA | NA
8
9 gfsh>list lucene indexes
10 Index Name | Region Path | Indexed Fields
11 | Field Analy.. | Status
12 ----- | ----- | -----
13 analyzerIndex | /Person | [revenue, address, name, email]
14 | {revenue=St.. | Initialized
15 customerIndex | /Customer | [symbol, revenue, SSN, name, email, address,
16 __REGION_VALUE_FIELD] | {} | Initialized
17 pageIndex | /Page | [id, title, content]
18 | {} | Initialized
19 personIndex | /Person | [name, email, address, revenue]
20 | {} | Initialized
```

list members {#list_members}

显示全部或部分成员。

在输出中，列出了成员协调员。 <vN> 标识成员当前拥有的视图; N 将为零或正整数。 <ec> 表示哪些成员有资格成为会员协调员。

可用性: Online. 您必须在 gfsh 中连接到JMX Manager成员才能使用此命令。

句法:

```
1 list members [--group=value]
```

参数,列出成员:

名称	描述
--group	将显示成员的组名称。

示例命令:

```
1 list members
2 list members --group=Group1
```

示例输出:

```
1 gfsh>list members
2 Name | Id
3 ----- | -----
4 Coordinator: | 192.0.2.0(locator1:216:locator)<ec><v6>:33368
5 locator1 | 192.0.2.0(locator1:216:locator)<ec><v6>:33368
6 server1 | 192.0.2.0(server1:888)<v7>:10839
7 server2 | 192.0.2.0(server2:3260)<v8>:16721
```

list regions {#list_regions}

显示成员或成员的区域。如果未指定参数，则列出群集中的所有区域。

句法:

```
1 | list regions [--groups=value(,value)*] [--members=value(,value)*]
```

参数，列出区域:

名称	描述
--groups	将显示区域的成员组。
--members	将显示区域的成员的名称或ID。

示例命令:

```
1 | list regions
2 | list regions --groups=G1
3 | list regions --members=member1
```

示例输出:

```
1 | gfsh>list regions
2 | List of regions
3 | -----
4 | region1
5 | region2
```

load-balance gateway-sender {#load_balance_gateway_sender}

使指定的网关发送方关闭其当前连接并以更平衡的方式重新连接到远程网关接收方。

使用此命令可以平衡网关发件人与接收者之间的连接。例如，在远程站点添加新的网关接收器节点时，请执行此命令，以便新网关接收器可以从指定的网关发送器获取连接。调用此命令可在所有网关接收器之间更均匀地重新分配发送方的连接。

注意: 此命令对ping连接没有影响。

可用性: Online. 您必须在 gfsh 中连接到JMX Manager成员才能使用此命令。

句法:

```
1 | load-balance gateway-sender --id=value
```

名称	描述
--id	<i>Required.</i> 网关发件人的ID。

Table 1. 负载均衡网关 - 发件人参数

示例命令:

```
1 | load-balance gateway-sender --id=sender1-LN
```

示例输出:

```
1 | load-balance gateway-sender --id=ny
2 |
3 |           Member           | Result | Message
4 | -----|-----|-----
5 | boglesbymax(1n-1:88651)<v2>:48277 | OK      | GatewaySender ny is rebalanced
6 |   on member boglesbymax(1n-1:88651)<v2>:48277
7 | boglesbymax(1n-4:88681)<v5>:42784 | OK      | GatewaySender ny is rebalanced
8 |   on member boglesbymax(1n-4:88681)<v5>:42784
9 | boglesbymax(1n-3:88672)<v4>:43675 | OK      | GatewaySender ny is rebalanced
10|   on member boglesbymax(1n-3:88672)<v4>:43675
11| boglesbymax(1n-2:88662)<v3>:12796 | OK      | GatewaySender ny is rebalanced
12|   on member boglesbymax(1n-2:88662)<v3>:12796
```

locate entry {#locate_entry}

在成员上找到区域条目。

找到条目

使用指定的键在成员上找到给定的条目。使用分区区域时，此命令很有用。

可用性: Online. 您必须在 gfsh 中连接到 JMX Manager 成员才能使用此命令。

句法:

```
1 | locate entry --key=value --region=value [--key-class=value]
2 | [--value-class=value] [--recursive=value]
```

Name	Description	Default Value
--key	<i>Required.</i> String or JSON text from which to create a key. Examples include: "James", "100L" and "('id': '134s')".	
--region	<i>Required.</i> Region in which to locate values.	
--key-class	Fully qualified class name of the key's type.	java.lang.String
--value-class	Fully qualified class name of the value's type.	java.lang.String
--recursive	Whether to traverse regions and subregions recursively.	false

Table 1. 找到条目参数

示例命令:

```
1 locate entry --key=('id':'133abg124') --region=/region1
2 --key-class=data.ProfileKey --recursive=true;
```

示例输出:

```
1 gfsh>locate entry --key=('123abc') --region=region2
2 Result          : true
3 Key Class       : java.lang.String
4 Key             : ('123abc')
5 Locations Found : 2
6
7
8 MemberName | MemberId
9 ----- | -----
10 server1    | GemFireStymon(server1:3692)<v1>:13487
11 server2    | GemFireStymon(server2:2340)<v2>:11613
```

netstat {#netstat}

通过“netstat”操作系统命令报告网络信息和统计信息。

报告给定成员的重要网络使用信息/统计信息。

可用性: Online. 您必须在 gfsh 中连接到JMX Manager成员才能使用此命令。

句法:

```
1 netstat [--members=value(,value)*] [--group=value] [--file=value]
2 [--with-lsof(=value)?]
```

Name	Description	Default Value
--members	Name or ID of the member(s) on which to run the <code>netstat</code> command.	
--group	Group of members on which to run the <code>netstat</code> command.	
--file	Text file to which output from the <code>netstat</code> command will be written. A “.txt” extension will be added if it is not already part of the specified name.	
--with-lsof	Specifies whether <code>lsof</code> (list open files) command output should also be displayed. Not applicable for Microsoft Windows hosts.	false

Table 1. Netstat 参数

示例命令:

```
1 netstat
2 netstat --members=server1
3 netstat --members=server1 --file=server1_netstat.txt
```

示例输出:

```
1 gfsh>netstat
2 #####
3 Host: GemFireTest
4 OS: windows XP 5.1 x86
5 Member(s):
6   server2, locator1, server1
7 #####
8
9 Active Connections
10
11   Proto Local Address          Foreign Address         State
12   TCP    GemFireTest:epmap      GemFireTest:0           LISTENING
13   TCP    GemFireTest:microsoft-ds GemFireTest:0           LISTENING
14   TCP    GemFireTest:1034       GemFireTest:0           LISTENING
15   TCP    GemFireTest:1069       GemFireTest:0           LISTENING
16   TCP    GemFireTest:1099       GemFireTest:0           LISTENING
17   TCP    GemFireTest:1134       GemFireTest:0           LISTENING
18   TCP    GemFireTest:3389       GemFireTest:0           LISTENING
19   TCP    GemFireTest:8080       GemFireTest:0           LISTENING
20   TCP    GemFireTest:8081       GemFireTest:0           LISTENING
21   TCP    GemFireTest:10334      GemFireTest:0           LISTENING
22   TCP    GemFireTest:40404      GemFireTest:0           LISTENING
23   TCP    GemFireTest:40405      GemFireTest:0           LISTENING
24   TCP    GemFireTest:1025       GemFireTest:0           LISTENING
25   TCP    GemFireTest:5152       GemFireTest:0           LISTENING
26   TCP    GemFireTest:netbios-ssn GemFireTest:0           LISTENING
27   TCP    GemFireTest:1035       GemFireTest:0           LISTENING
28   TCP    GemFireTest:1035       GemFireTest:1081        ESTABLISHED
29   TCP    GemFireTest:1035       GemFireTest:1086        ESTABLISHED
30   TCP    GemFireTest:1035       GemFireTest:1147        ESTABLISHED
31   TCP    GemFireTest:1035       GemFireTest:1156        ESTABLISHED
32   TCP    GemFireTest:1046       GemFireTest:1099        ESTABLISHED
33   TCP    GemFireTest:1049       osdc-proxy-vip.vmware.com:http CLOSE_WA
34   TCP    GemFireTest:1050       osdc-proxy-vip.vmware.com:3128 CLOSE_WA
35   TCP    GemFireTest:1071       GemFireTest:0           LISTENING
36   TCP    GemFireTest:1071       GemFireTest:1077        ESTABLISHED
37   TCP    GemFireTest:1071       GemFireTest:1150        ESTABLISHED
38   TCP    GemFireTest:1071       GemFireTest:1157        ESTABLISHED
39   TCP    GemFireTest:1077       GemFireTest:1071        ESTABLISHED
40   TCP    GemFireTest:1078       GemFireTest:24400       ESTABLISHED
41   TCP    GemFireTest:1081       GemFireTest:1035        ESTABLISHED
42   TCP    GemFireTest:1086       GemFireTest:1035        ESTABLISHED
43   TCP    GemFireTest:1099       GemFireTest:1046        ESTABLISHED
44   TCP    GemFireTest:1136       GemFireTest:0           LISTENING
45   TCP    GemFireTest:1136       GemFireTest:1143        ESTABLISHED
46   TCP    GemFireTest:1136       GemFireTest:1151        ESTABLISHED
47   TCP    GemFireTest:1136       GemFireTest:1201        ESTABLISHED
48   TCP    GemFireTest:1141       GemFireTest:4247        ESTABLISHED
49   TCP    GemFireTest:1142       GemFireTest:48640       ESTABLISHED
50   TCP    GemFireTest:1143       GemFireTest:1136        ESTABLISHED
```

51	TCP	GemFireTest:1147	GemFireTest:1035	ESTABLISHED
52	TCP	GemFireTest:1150	GemFireTest:1071	ESTABLISHED
53	TCP	GemFireTest:1151	GemFireTest:1136	ESTABLISHED
54	TCP	GemFireTest:1156	GemFireTest:1035	ESTABLISHED
55	TCP	GemFireTest:1157	GemFireTest:1071	ESTABLISHED
56	TCP	GemFireTest:1201	GemFireTest:1136	ESTABLISHED
57	TCP	GemFireTest:1349	GemFireTest:10334	TIME_WAIT
58	TCP	GemFireTest:1350	GemFireTest:10334	TIME_WAIT
59	TCP	GemFireTest:1351	GemFireTest:10334	TIME_WAIT
60	TCP	GemFireTest:1352	GemFireTest:10334	TIME_WAIT
61	TCP	GemFireTest:1353	GemFireTest:10334	TIME_WAIT
62	TCP	GemFireTest:1354	GemFireTest:10334	TIME_WAIT
63	TCP	GemFireTest:4247	GemFireTest:0	LISTENING
64	TCP	GemFireTest:4247	GemFireTest:1141	ESTABLISHED
65	TCP	GemFireTest:24400	GemFireTest:0	LISTENING
66	TCP	GemFireTest:24400	GemFireTest:1078	ESTABLISHED
67	TCP	GemFireTest:48640	GemFireTest:0	LISTENING
68	TCP	GemFireTest:48640	GemFireTest:1142	ESTABLISHED
69	UDP	GemFireTest:microsoft-ds	*:*	
70	UDP	GemFireTest:isakmp	*:*	
71	UDP	GemFireTest:4500	*:*	
72	UDP	GemFireTest:ntp	*:*	
73	UDP	GemFireTest:1900	*:*	
74	UDP	GemFireTest:ntp	*:*	
75	UDP	GemFireTest:netbios-ns	*:*	
76	UDP	GemFireTest:netbios-dgm	*:*	
77	UDP	GemFireTest:1900	*:*	
78	UDP	GemFireTest:32270	*:*	
79	UDP	GemFireTest:42838	*:*	
80	UDP	GemFireTest:47727	*:*	

pause gateway-sender {#pause_gateway_sender}

暂停网关发件人。

暂停成员或成员上的网关发件人。有关暂停网关发件人的详细信息，请参阅[暂停网关发件人](#)。

可用性: Online. 您必须在 `gfs` 中连接到 JMX Manager 成员才能使用此命令。

句法:

```
1 pause gateway-sender --id=value [--groups=value(,value)*]
2 [--members=value(,value)*]
```

Name	Description
--id	<i>Required.</i> ID of the gateway sender.
--groups	Group(s) of members on which to pause the gateway sender.
--members	Name or ID of the member(s) on which to pause the gateway sender.

Table 1. 暂停 网关-发件人 参数

示例命令:

```
1 | pause gateway-sender --id=sender1
```

pdx rename {#pdx_rename}

在脱机磁盘存储中重命名PDX类型。

重命名的任何PDX类型都将在输出中列出。如果未完成重命名或磁盘存储在线，则此命令将失败。

可用性: Offline.

句法:

```
1 | pdx rename --old=value --new=value --disk-store=value --disk-  
  dirs=value(,value)*
```

Name	Description
--old	<i>Required.</i> If a PDX type's fully qualified class name has a word that matches this text then it will be renamed. Words are delimited by '.' and '\$'.
--new	<i>Required.</i> The text to replace the word that matched old.
--disk-store	<i>Required.</i> Name of the disk store to operate on.
--disk-dirs	<i>Required.</i> Directories which contain the disk store files.

示例命令:

将以“com.gemstone”开头的所有包更改为“com.pivotal”:

```
1 | gfsh>pdx rename --old=com.gemstone --new=com.pivotal --disk-store=ds1 --disk-  
  dirs=/diskDir1
```

将名为“MyClassName”的类更改为“YourClassName”:

```
1 | gfsh>pdx rename --old=MyClassName --new=YourClassName --disk-store=ds1 --  
  disk-dirs=/diskDir1
```

将FQCN“com.target.app1.OldClass”更改为“com.target.app2.NewClass”:

```
1 | gfsh>pdx rename --old=com.target.app1.OldClass --new=com.target.app2.NewClass  
  --disk-store=ds1 --disk-dirs=/diskDir1
```

示例输出:

```
1 | gfsh>pdx rename --old=PortfolioPdx --new=StockPdx --disk-store=DiskStore1 --  
  disk-dirs=/DiskDir1  
2 | Successfully renamed pdx types:  
3 |   com.app.data.StockPdx: id=2  
4 |   com.app.data.StockPdx$Day.Monday
```

错误消息:

如果没有类型匹配, 您可能会收到以下错误消息:

```
1 gfsh>pdx rename --old=gemstone --new=pivotal --disk-store=DiskStore1 --disk-
  dirs=/DiskDir1
2 Could not process command due to GemFire error. No Pdx types found to rename.
```

如果存储PDX类型的磁盘存储在线, 您将收到以下错误消息:

```
1 gfsh>pdx rename --old=StockPdx --new=PortfolioPdx --disk-store=DiskStore1 --
  disk-dirs=/DiskDir1
2 Could not process command due to GemFire error. Error renaming pdx types :
3 GemFireCache[id = 484629896; isClosing = false; isShutdownAll = false;
4 closingGatewayHubsByShutdownAll = false; created = Wed Jan 07 10:29:45 PST
  2015;
5 server = false; copyOnRead = false; lockLease = 120; lockTimeout = 60]: An
  open cache
6 already exists.
```

put {#put}

添加或更新区域中的条目。

可用性: Online. 您必须在 gfsh 中连接到 JMX Manager 成员才能使用此命令。

句法:

```
1 put --key=value --value=value --region=value [--key-class=value]
2 [--value-class=value] [--if-not-exists(=value)]
```

名称	描述	默认值
--key	<i>Required.</i> 从中创建键的字符串或JSON文本。例如: "James", "100L" 和 ("id": "134s")。	
--value	<i>Required.</i> 从中创建值的字符串或JSON文本。例如: "James", "100L" 和 ("id": "134s")。	
--region	<i>Required.</i> 条目将放入的区域。	
--key-class	键类型的完全限定类名。	java.lang.String
--value-class	值类型的完全限定类名。	java.lang.String
--if-not-exists	当具有相同键的条目已存在时, 跳过放置操作。	false

示例命令:

```

1 put --key=('id':'133abg125') --value=
  ('firstname':'James','lastname':'Gosling')
2 --region=/region1 --key-class=data.ProfileKey --value-
  class=data.ProfileDetails
3
4 put --key=('133abg124') --value=('Hello world!!') --region=/region2
5
6 put --key=('100F') --value=('2146547689879658564') --
  region=/region1/region12
7 --key-class=java.lang.Float --value-class=java.lang.Long

```

示例输出:

```

1 gfsh>put --key=('123abc') --value=('Hello world!!') --region=region2
2 Result      : true
3 Key Class   : java.lang.String
4 Key         : ('123abc')
5 Value Class : java.lang.String
6 Old Value   : <NULL>

```

错误消息:

```

1 "Region name is either empty or Null";
2 "Key is either empty or Null";
3 "Value is either empty or Null";
4 "Region <{0}> not found in any of the members";
5 "Region <{0}> Not Found";
6 "Key is not present in the region";

```

query {#query}

对Geode区域运行查询。

如果在查询中未设置限制结果大小的限制，则为[有用的gfsh Shell变量](#)中定义的gfsh环境变量 APP_FETCH_SIZE 的默认限制。

使用单引号围绕OQL查询。

注意: 如果被查询的对象包含循环引用，则不应从 gfsh 执行此命令。

可用性: Online. 您必须在 gfsh 中连接到JMX Manager成员才能使用此命令。

句法:

```

1 query --query=value [--step-name=value] [--file=path/to/results/file]

```

名称	描述
--query	<i>Required.</i> OQL 字符串.
--file	指定后，所有查询结果都将写入指定的文件。如果文件已存在，则会发出错误。

Table 1. 查询参数

示例输出:

```

1  gfsh>query --query='SELECT * FROM /region2'
2
3  Result      : true
4  startCount  : 0
5  endCount    : 20
6  Rows        : 1
7
8  Result
9  -----
10 ('Hello world!!')
11
12 NEXT_STEP_NAME : END

```

rebalance {#rebalance}

重新平衡分区区域。

默认情况下，所有分区区域都要重新平衡。

可用性: Online. 您必须在 `gfsh` 中连接到 JMX Manager 成员才能使用此命令。

句法:

```

1  rebalance [--include-region=value(,value)*]
2  [--exclude-region=value(,value)*] [--time-out=value] [--simulate(=value)?]

```

名称	描述	默认值
--include-region	为重新平衡操作包含的分区区域路径。包括优先于排除。	
--exclude-region	要为重新平衡操作排除的分区区域路径。	
--time-out	GFSH 返回到提示之前等待的时间（以秒为单位），同时在后台继续重新平衡。	-1 (等待重新平衡完成)
--simulate	是否只模拟重新平衡。模拟时 time-out 参数不可用。	false

Table 1. 重新平衡的参数

示例命令:

```

1  rebalance --include-region=/region3 --simulate=true

```

示例输出:

```

1  rebalance

```

```
2 1. server1 host1(3467):12435:12423
3 Row Rebalanced Stats | value
4 ---|-----|-----
5 1 TotalBucketCreateBytes | 0
6 2 TotalBucketCreateTime | 0
7 3 TotalBucketCreatesCompleted | 0
8 4 TotalBucketTransferBytes | 0
9 5 TotalBucketTransferTime | 0
10 6 TotalBucketTransfersCompleted | 0
11 7 TotalPrimaryTransferTime | 0
12 8 TotalPrimaryTransfersCompleted | 0
13 9 TotalTime | 56
14
15 Rebalance complete on host1(3467):12435:12423.
```

remove {#remove}

从区域中删除条目。

可用性: Online. 您必须在 `gfsh` 中连接到JMX Manager成员才能使用此命令。

句法:

```
1 | remove --region=value [--key=value] [--all(=value)?] [--key-class=value]
```

名称	描述	默认值
--key	将用于创建检索值的键的字符串或JSON文本。	
--key-class	键类型的完全限定类名。	当前区域或String的键约束
--region	<i>Required.</i> 要从中删除条目的区域。	
--all	一个布尔值，当为true时，通过删除所有条目来清除该区域。此选项不适用于分区区域。	false

示例命令:

```
1 | gfsh>remove --region=/region1 --key=('id': '133abg134')
2 | gfsh>remove --region=/region1 --key=('id': '133abg134') --key-
  | class=data.Profilekey
3 | gfsh>remove --region=/region1 --all=true
```

错误消息:

```

1  "Region name is either empty or Null"
2
3  "Key is either empty or Null"
4
5  "Value is either empty or Null"
6
7  "Region <{0}> not found in any of the members"
8
9  "Region <{0}> Not Found"
10
11 "Key is not present in the region"
12
13 "Option --all is not supported on partitioned region"

```

resume gateway-sender {#resume_gateway_sender}

恢复已暂停的所有网关发件人。

恢复成员或成员上的网关发件人。

可用性: Online. 您必须在 `gfsh` 中连接到 JMX Manager 成员才能使用此命令。

句法:

```

1 resume gateway-sender --id=value [--groups=value(,value)*]
2 [--members=value(,value)*]

```

名称	描述
--id	<i>Required.</i> 网关发件人的ID。
--groups	要恢复网关发件人的成员组。
--members	要恢复网关发件人的成员的名称/ID。

Table 1. 恢复网关发件人参数

示例命令:

```

1 resume gateway-sender --id=sender1-LN --groups=LN-Group1

```

revoke missing-disk-store {#revoke_missing_disk_store}

指示群集的成员停止等待磁盘存储可用。

仅在文件丢失时撤销磁盘存储，因为在启动撤销后它将不再可恢复。使用 `show missing-disk-store` 命令获取缺少磁盘存储的描述。

可用性: Online. 您必须在 `gfsh` 中连接到 JMX Manager 成员才能使用此命令。

句法:

```
1 | revoke missing-disk-store --id=value
```

名称	描述
--id	<i>Required.</i> 要撤消的丢失磁盘存储的ID。

Table 1. 撤消丢失磁盘存储参数

示例命令:

```
1 | revoke missing-disk-store --id=60399215-532b-406f-b81f-9b5bd8d1b55a
```

示例输出:

```
1 | gfsh>revoke missing-disk-store --id=60399215-532b-406f-b81f-9b5bd8d1b55a
2 | Missing disk store successfully revoked
```

错误消息:

无法找到磁盘存储时 `revoke missing-disk-store` 的示例:

```
1 | gfsh> revoke missing-disk-store --id=60399215-532b-406f-b81f
2 |
3 | Unable to find missing disk store to revoke
```

run {#run}

执行一组GFSH命令。

Commands that normally prompt for additional input will instead use default values.

可用性: Online or offline.

注意: 文件中指定的某些命令需要在线状态。

句法:

```
1 | run --file=value [--quiet(=value)?] [--continue-on-error(=value)?]
```

名称	描述	默认值
--file	<i>Required.</i> 使用 gfsh 识别的命令编写脚本的文件路径。路径应该是相对的或绝对的。	
--quiet	指定是否显示命令输出。 注意: 当使用 run 命令时, 脚本中的所有命令都以非交互方式运行。此选项不会更改该功能。	false
--continue-on-error	指定在执行其中一个命令失败时是否存在错误, 是否应继续执行脚本。	false

Table 1. Run 参数

示例命令:

```

1 run --file=create-regions.gfsh --quiet=true
2
3 (2) From command line:
4 prompt> /home/user1/gemfire70/bin/gfsh run ./create-regions.gfsh --quiet=true
5 prompt> /home/user1/gemfire70/bin/gfsh run ./create-regions.gfsh
6 --continue-on-error=true

```

示例输出:

```

1 gfsh>run --file=create-regions.gfsh
2 1. Executing - create region --name=region4 --type=REPLICATE
3
4 Member | Status
5 -----|-----
6 server2 | Region "/region4" created on "server2"
7 server1 | Region "/region4" created on "server1"
8
9 2. Executing - create region --name=region1/subregion1 --type=LOCAL
10
11 Parent region for "region1/subregion1" doesn't exist.

```

search lucene {#search_lucene}

搜索Lucene索引

参见 [create lucene index](#), [describe lucene index](#), [destroy lucene index](#) 和 [list lucene indexes](#).

可用性: Online.

句法:

```

1 search lucene --name=value --region=value --queryString=value --
  defaultField=value
2 [--limit=value] [--keys-only=value]

```

参数, 搜索lucene:

名称	描述	默认值
--name	<i>Required.</i> 要搜索的Lucene索引的名称。	
--region	<i>Required.</i> Lucene索引所在区域的名称/路径。	
--queryString	<i>Required.</i> 查询字符串以搜索Lucene索引。当字段为原始值时，使用 <code>__REGION_VALUE_FIELD</code> 作为查询字符串中的字段名称。用双引号括起一个字符串，以完成字符串的匹配。	
--defaultField	<i>Required.</i> 要搜索的默认字段。 <code>__REGION_VALUE_FIELD</code> 将字段标识为原始值。	
--limit	Number of search results needed.	如果未指定参数: -1
--keys-only	仅返回搜索结果的键。	如果未指定参数: false

示例命令:

```

1 gfsh> search lucene --name=testIndex --region=/testRegion --
  queryString=value1
2     --defaultField=__REGION_VALUE_FIELD
3
4
5 gfsh> search lucene --name=indexOfStrings --region=/stringTestRegion
6     --queryString='__REGION_VALUE_FIELD:"my exact string"'
7     --defaultField=__REGION_VALUE_FIELD

```

示例输出:

```

1 gfsh>search lucene --name=testIndex --region=/testRegion --
  queryString=value*
2     --defaultField=__REGION_VALUE_FIELD
3 key | value | score
4 --- | ----- | -----
5 3   | value3 | 1
6 2   | value2 | 1
7 1   | value1 | 1
8 gfsh>search lucene --region=/Person --name=analyzerIndex
9     --defaultField=addr --queryString="97763"
10 key | value
    | score
11 ----- | -----
12 key763 | Person{name='Kris Cat', addr='7 Ash St, Portland_OR_97763', emai..
    | 1.669657

```

set variable {#set_variable}

在GFSH环境中设置变量。

设置命令可以使用的GFSH变量。

您可以使用[echo](#)命令查看变量的值。例如，要查看所有环境变量及其当前值的列表，请使用以下命令：

```
1 | gfsh>echo --string=$*
```

有关预设环境变量的说明，请参阅[有用的gfsh Shell变量](#)。

可用性: Online or offline.

句法:

```
1 | set variable --name=value --value=value
```

名称	描述
--name	<i>Required.</i> 变量的名称。名称必须仅由字母，数字和“_”字符组成，并且不能以数字开头。
--value	<i>Required.</i> 变量将设置为的值。

Table 1. 设置变量的参数

示例命令:

```
1 | set variable --name=APP_COLLECTION_LIMIT --value=10
2 | set variable --name=FOO --value="foo"
3 | set variable --name=BAR --value="bar"
```

示例输出:

```
1 | gfsh>set variable --name=APP_COLLECTION_LIMIT --value=10
2 | Value for variable APP_COLLECTION_LIMIT is now: 10.
3 |
4 | gfsh>set variable --name=BAR --value="bar"
5 | Value for variable BAR is now: "bar".
```

sh {#sh}

执行操作系统命令。

执行操作系统（OS）命令。使用 `&` 立即返回 gfsh 提示符。

注意: 不支持将输出传递给另一个shell命令的命令。

句法:

```
1 | sh command [--use-console(=value)?]
```

名称	描述	默认值
--use-console	在UNIX系统上为需要控制台句柄的应用程序设置此参数。将“</dev/tty>/dev/tty”添加到指定的命令。	false

Table 1. Sh 参数

示例命令:

```
1 gfsh>sh ls -al
2 total 80
3 drwxrwxr-x. 10 username username 4096 Sep 3 15:10 .
4 drwxrwxr-x. 4 username username 4096 Sep 3 14:58 ..
5 drwx-----. 2 username username 4096 Sep 3 15:09 bin
6 drwx-----. 2 username username 4096 Sep 3 15:09 defaultConfigs
7 drwx-----. 3 username username 4096 Sep 3 15:09 docs
8 drwx-----. 2 username username 4096 Sep 3 15:09 dtd
9 -rwx-----. 1 username username 31830 Sep 3 15:09 EULA.txt
10 drwx-----. 2 username username 4096 Sep 3 15:09 lib
11 drwx-----. 6 username username 4096 Sep 3 15:09 sampleCode
12 drwx-----. 4 username username 4096 Sep 3 15:09 templates
13 drwx-----. 5 username username 4096 Sep 3 15:09 tools
```

show {#show}

显示死锁，日志，指标和丢失的磁盘存储。

- **show dead-locks**
显示群集中的所有死锁。
- **show log**
显示成员的日志。
- **show metrics**
显示或导出整个群集，成员或区域的度量标准。
- **show missing-disk-stores**
显示群集中当前缺少的磁盘存储的摘要。
- **show subscription-queue-size**
显示订阅队列中的事件数。

show dead-locks {#show_dead_locks}

显示群集中的所有死锁。

可用性: Online. 必须在gfsh中连接到MX Manager成员才能使用此命令。

句法:

```
1 | show dead-locks --file=value
```

名称	描述
--file	<i>Required.</i> 将写入成员之间依赖关系的文件的名称。

Table 1. 显示死锁参数

示例命令:

```
1 | show dead-locks --file=deadlocks.txt
```

示例输出:

```
1 | gfsh>show dead-locks --file=deadlocks.txt
2 | No dead lock detected.
3 |
4 | Please view the dependencies between the members in file : deadlocks.txt
```

show log {#show_log}

显示成员的日志。

可用性: Online. 您必须在 gfsh 中连接到 JMX Manager 成员才能使用此命令。

句法:

```
1 | show log --member=value [--lines=value]
```

名称	描述	默认值
--member	<i>Required.</i> 将显示其日志文件的成员的名称/ID。	
--lines	要显示的日志文件中的行数。最大值为100。	0

示例命令:

```
1 | show log --member=locator1 --lines=5
```

示例输出:

```

1 gfsh>show log --member=locator1 --lines=5
2 SystemLog:
3 [info 2012/09/25 14:04:51.340 PDT locator1 <RMI TCP
Connection(14)-192.0.2.0> tid=0x57] (tid=12 msgId=4) Parent region for
"region1/subregion1" doesnt exi
4 st.
5
6 [info 2012/09/25 14:04:51.372 PDT locator1 <RMI TCP
Connection(14)-192.0.2.0> tid=0x57] (tid=12 msgId=5) Error occurred while
executing "create region --n
7 ame=region1/subregion1 --type=LOCAL".
8
9 [info 2012/09/25 15:14:34.314 PDT locator1 <RMI TCP
Connection(159)-192.0.2.0> tid=0x68] (tid=13 msgId=6) Error occurred while
executing "show log --membe
10 r=server1 --lines=5".

```

show metrics {#show_metrics}

显示或导出整个群集，成员或区域的度量标准。

如果未给出命令行参数，则会显示集群，缓存，磁盘库和查询类别下的度量标准。

可用性: Online. 您必须在 gfsh 中连接到 JMX Manager 成员才能使用此命令。

句法:

```

1 show metrics [--member=value] [--region=value] [--file=value]
2 [--port=value] [--categories=value(,value)*]

```

名称	描述
--member	将显示/导出其指标的成员的名称/ID。
--region	将显示/导出其指标的区域的名称/路径。
--file	要写入度量标准的文件的名称。
--port	要显示/导出其度量标准的高速缓存服务器的端口号。这只能与 --member 参数一起使用。
--categories	基于指定的参数（上面列出的）可用的类别是： region specified: cluster, region, partition, diskstore, callback, eviction member specified: member, jvm, region, serialization, communication, function, transaction, diskstore, lock, eviction, distribution, offheap member and region specified: region, partition, diskstore, callback, eviction

Table 3. 显示度量标准参数

示例命令:

```

1 // Metrics for the entire system
2 show metrics
3 // Metrics for a region:
4 show metrics --region=region1
5 // Metrics for a given member
6 show metrics --member=server1
7 // Metrics for a region on a member
8 show metrics --region=region1 --member=server1
9 // Metrics for a member and the cacheserver it hosts
10 // NOTE: port option only work when used with --member option
11 show metrics --member=server1 --port=10334
12 // Export metrics for the entire system
13 show metrics --file=data/stats/system-stats.csv

```

示例输出:

```

1 gfsh>show metrics
2
3 Cluster-wide Metrics
4
5   Type      | Metric                      | Value
6   ----- | ----- | -----
7 cluster    | totalHeapSize               | 123
8 cache      | totalRegionEntryCount       | 0
9             | totalRegionCount            | 0
10            | totalMissCount              | 0
11            | totalHitCount               | 0
12 diskstore  | totalBytesOnDisk            | 0
13            | diskReadsRate               | 0
14            | diskWritesRate              | 0
15            | flushTimeAvgLatency         | 0
16            | totalBackupInProgress       | 0
17 query     | queryRequestRate            | 0

```

show missing-disk-stores {#show_missing_disk_stores}

显示群集中当前缺少的磁盘存储的摘要。

可用性: Online. 您必须在 gfsh 中连接到 JMX Manager 成员才能使用此命令。

句法:

```

1 show missing-disk-stores

```

示例命令:

```

1 show missing-disk-stores

```

示例输出:

```
1 gfsh> show missing-disk-stores
2
3           Disk Store ID           | Host |           Directory
4 -----|-----|-----
5 60399215-532b-406f-b81f-9b5bd8d1b55a | excalibur | /usr/local/gemfire/deploy/disk_store1
```

show subscription-queue-size {#show_subscription_queue_size}

显示订阅队列中的事件数。

如果提供CQ名称，则它计算指定CQ的订阅队列中的事件数。

可用性: Online. 您必须在 gfsh 中连接到JMX Manager成员才能使用此命令。

句法:

```
1 show subscription-queue-size --durable-client-id=value
2 [--members=value(,value)*] [--groups=value(,value)*]
```

名称	描述
--durable-client-id	<i>Required.</i> 用于标识持久客户端的ID。
--durable-cq-name	标识CQ的名称。
--members	要对订阅队列事件进行计数的成员的名称/ID。
--groups	要对订阅队列事件进行计数的成员组。

Table 4. 显示订阅 - 队列大小参数

示例命令:

```
1 show subscription-queue-size --durable-client-id=client1
```

示例输出:

```
1 gfsh>show subscription-queue-size --durable-client-id=client1
2 member | subcription-queue-size for durable-client : "client1".
3 -----|-----
4 server3 | 1
5 server4 | 0
```

错误消息:

```

1 gfsh>show subscription-queue-size --durable-client-id=client1
2
3 No client found with client-id : client1
4 Occurred on members
5 1.server4
6 2.server1
7 3.server3

```

shutdown {#shutdown}

停止所有成员。

要求所有具有缓存的成员关闭缓存并断开与系统的连接。如果指定了 `--include-locators` 参数，则该命令将逐个关闭所有正在运行的定位符。timeout参数允许您指定在超过时间后强制关闭系统。

可用性: Online. 您必须在 `gfsh` 中连接到JMX Manager成员才能使用此命令。

句法:

```

1 shutdown [--time-out=value] [--include-locators=value]

```

名称	描述	默认值
--time-out	等待（以秒为单位）正常关闭的时间。应该至少10秒。如果未指定，则默认值为10。	10
--include-locators	要关闭定位器，请将此选项指定为true。	false

Table 1. 停止参数

示例命令:

```

1 shutdown
2 shutdown --time-out=15
3 shutdown --include-locators=true

```

示例输出:

```

1 gfsh>shutdown
2 "As a lot of data in memory will be lost, including possibly events in
3 queues, do you really want to shutdown the entire distributed system? (Y/n):"
4 Y
5 Shutdown is triggered

```

sleep {#sleep}

延迟 `gfsh` 命令执行。

以秒为单位延迟指定的时间 - 允许浮点值。

可用性: Online of offline.

句法:

```
1 | sleep [--time=value]
```

名称	描述	默认值
--time	延迟的秒数。	3

Table 1. 延迟参数

示例命令:

```
1 | sleep
2 | sleep --time=60
```

示例输出:

```
1 | gfsh>sleep --time=60
2 |
3 | gfsh>
```

start {#start}

启动服务器，定位器，网关发送器和网关接收器以及监视工具。

- **start gateway-receiver**
在给定成员或成员组上启动网关接收器。
- **start gateway-sender**
在一个或多个成员上启动网关发件人。
- **start jconsole**
在单独的进程中启动JDK JConsole监视应用程序。
- **start jvisualvm**
在单独的进程中启动JDK的Java VisualVM监视应用程序。
- **start locator**
启动定位器。
- **start pulse**
在用户的默认系统浏览器中启动Geode Pulse监控仪表板工具，并将用户导航到登录页面（登录页面）。
- **start server**
启动Geode缓存服务器进程。

start gateway-receiver {#start_gateway_receiver}

在给定的成员或成员组上启动网关接收器。

请注意，每个成员只能有一个网关接收器，与网关发送器不同，您不需要为网关接收器指定标识符。

可用性: Online. 您必须在 gfsh 中连接到 JMX Manager 成员才能使用此命令。

句法:

```
1 | start gateway-receiver [--groups=value(,value)*] [--members=value(,value)*]
```

参数，启动网关-接收器

名称	描述
--members	启动网关接收器的成员的名称或ID。
--groups	要在其上启动网关接收器的成员组。

示例命令:

```
1 | start gateway-receiver
2 | start gateway-receiver --members=member1
```

示例输出:

```
1 | gfsh>start gateway-receiver
2 |      Member          | Result | Message
3 | -----|-----|-----
4 | pc13(2266)<v6>:56852   | OK     | GatewayReceiver is started on member
   | pc13(2266)<v6>:56852
5 | pc13(Manager:2242)<v5>:57631| Error  | GatewayReceiver is not available on
   | member pc13(Manager:2242)<v5>:57631
6 | pc13(2275)<v7>:47480   | OK     | GatewayReceiver is started on member
   | pc13(2275)<v7>:47480
7 | pc13(2293)<v8>:55472   | OK     | GatewayReceiver is started on member
   | pc13(2293)<v8>:55472
8 |
9 | gfsh>start gateway-receiver --members=pc13(2266)<v14>:36579
10 | GatewayReceiver is started on member pc13(2266)<v14>:36579
11 |
12 | gfsh>start gateway-receiver --group=RG1
13 |      Member          | Result | Message
14 | -----|-----|-----
15 | pc13(2275)<v23>:27484 | OK     | GatewayReceiver is started on member
   | pc13(2275)<v23>:27484
16 | pc13(2293)<v24>:55810 | OK     | GatewayReceiver is started on member
   | pc13(2293)<v24>:55810
17 | pc13(2266)<v22>:4522  | OK     | GatewayReceiver is started on member
   | pc13(2266)<v22>:4522
```

start gateway-sender {#start_gateway_sender}

在一个或多个成员上启动网关发件人。

有关如何配置网关发件人的信息，请参阅[配置网关发件人](#)。

注意: 默认情况下, 网关发件人配置为自动启动。 手动重启会带来数据丢失的风险; 它不适用于生产系统。

可用性: Online. 您必须在 `gfsh` 中连接到JMX Manager成员才能使用此命令。

句法:

```
1 start gateway-sender --id=value [--groups=value(,value)*] [--members=value(,value)*]
```

参数, 启动网关发件人

名称	描述
--id	<i>Required.</i> GatewaySender的ID。
--groups	启动Gateway Sender的成员组。
--members	启动Gateway Sender的成员

示例命令:

```
1 start gateway-sender --id=sender1-NY
2 start gateway-sender --id=sender1-NY --members=server1
3 start gateway-sender --id=sender1-NY --groups=MemberGroup1,MemberGroup2
```

示例输出:

```
1 gfsh>start gateway-sender --id=ln
2 Member | Result | Message
3 -----|-----|-----
4 pc13(30614)<v6>:63670 | OK | GatewaySender ln is started on
member pc13(30614)<v6>:63670
5 pc13(30621)<v7>:36015 | OK | GatewaySender ln is started on
member pc13(30621)<v7>:36015
6 pc13(30633)<v8>:13633 | OK | GatewaySender ln is started on
member pc13(30633)<v8>:13633
7 pc13(Manager:30588)<v5>:42792 | Error | GatewaySender ln is not available
on member pc13(Manager:30588)<v5>:42792
8
9 gfsh>start gateway-sender --id=ln --members=pc13(30614)<v14>:44519
10 GatewaySender ln is started on member pc13(30614)<v14>:44519
11 gfsh>start gateway-sender --id=ln --groups=SenderGroup1
12 Member | Result| Message
13 -----|-----|-----
14 pc13(30614)<v18>:15201 | OK | GatewaySender ln is started on member
pc13(30614)<v18>:15201
15 pc13(30621)<v19>:61437 | OK | GatewaySender ln is started on member
pc13(30621)<v19>:61437
16 pc13(30633)<v20>:22567 | OK | GatewaySender ln is started on member
pc13(30633)<v20>:22567
```

start jconsole {#start_jconsole}

在单独的进程中启动JDK JConsole监视应用程序。

JConsole会自动连接到正在运行的JMX Manager节点（如果有）。

请注意，您必须安装JDK（而不仅仅是JRE）并设置正确的PATH和JAVA_HOME环境变量。

有关将JConsole与Geode管理和监视系统一起使用的示例，请参阅[通过JConsole浏览Geode MBean](#)。

可用性: Online or offline.

句法:

```
1 start jconsole [--interval=<seconds>] [--notile] [--version]
2 [--J<jconsole JVM options>]
```

参数，启动jconsole

名称	描述	默认值
--interval	将更新间隔设置为n秒（默认为4秒）。（相当于JConsole的 -interval = n）	4
--notile	是否最初为两个或更多连接平铺窗口。此参数作为 -notile 传递给JConsole。	false
--pluginpath	搜索JConsole插件的目录或JAR文件。该路径应包含一个名为 META-INF/services/com.sun.tools.jconsole.JConsolePlugin 的提供者配置文件，每个插件包含一行，用于指定实现 com.sun.tools.jconsole.JConsolePlugin 的类的完全限定类名。	
--version	显示JConsole版本信息。此参数作为 -version 传递给JConsole。	false
-J	传递给JConsole运行的JVM的参数	

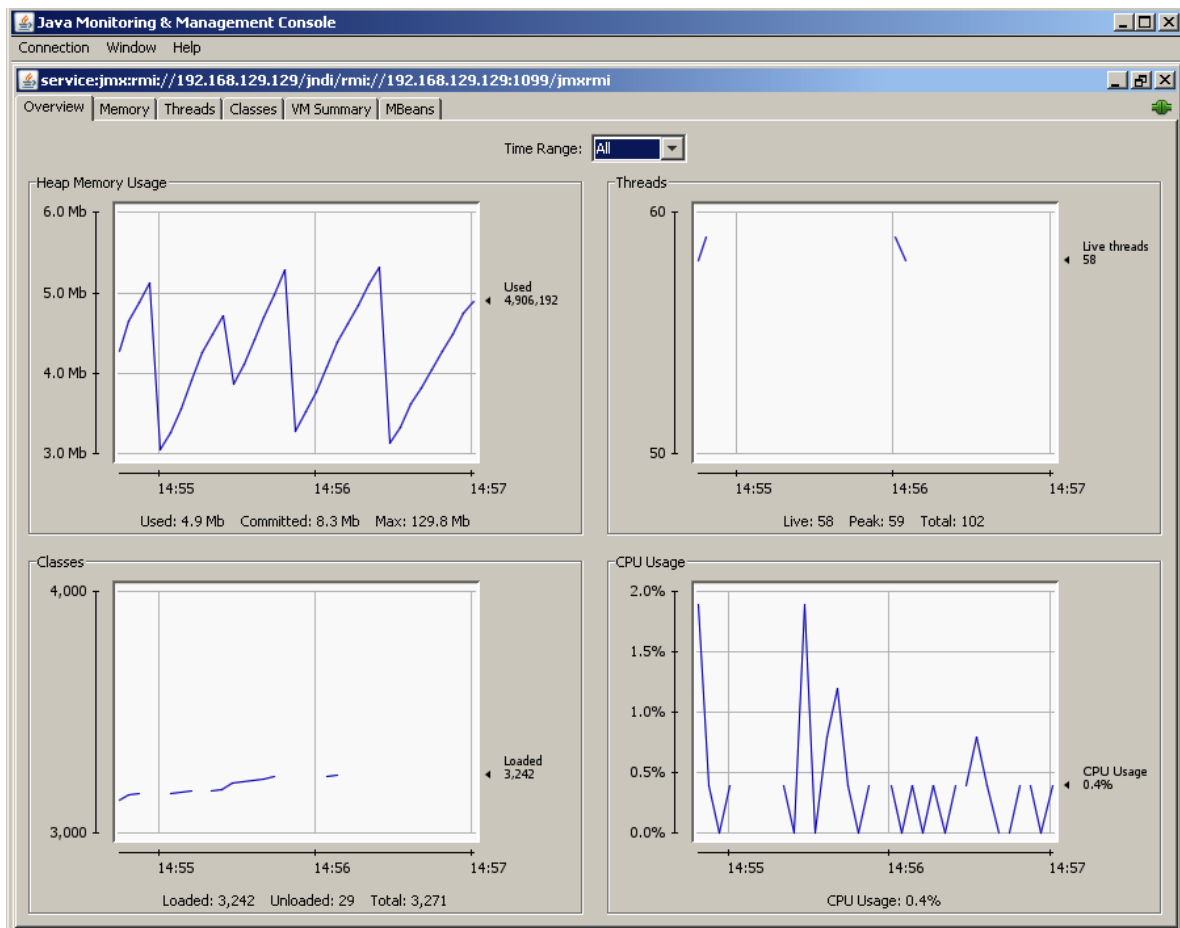
示例命令:

```
1 gfsh>start jconsole --interval=8 --notile;
2 Running JDK JConsole
3
4 gfsh>start jconsole --version;
5 JConsole version "1.8.0_31-b01-1"
6 Java(TM) SE Runtime Environment (build 1.8.0_31-b01-1-11)
7 Java HotSpot(TM) 64-Bit Server VM (build 20.6-b01-11, mixed mode)
```

示例输出:

```
1 gfsh>start jconsole
2 Running JDK JConsole
```

JConsole应用程序出现并自动连接到JMX Manager节点（如果有）：



错误消息:

```

1  An error occurred while launching JConsole = %1$s
2
3  Connecting by the Geode member's name or ID is not currently supported.
4  Please specify the member as '<hostname|IP>[PORT]'.
5
6  An IO error occurred while launching JConsole.
7  Please ensure that JAVA_HOME is set to the JDK installation
8  or the JDK bin directory is in the system PATH.
9
10 JConsole could not be found.\nPlease ensure that JAVA_HOME is set to the
11 JDK installation or the JDK bin directory is in the system PATH.

```

start jvisualvm {#start_jvisualvm}

在单独的进程中启动JDK的Java VisualVM监视应用程序。

可用性: Online or offline.

句法:

```
1 | start jvisualvm [--J=value(,value)*]
```

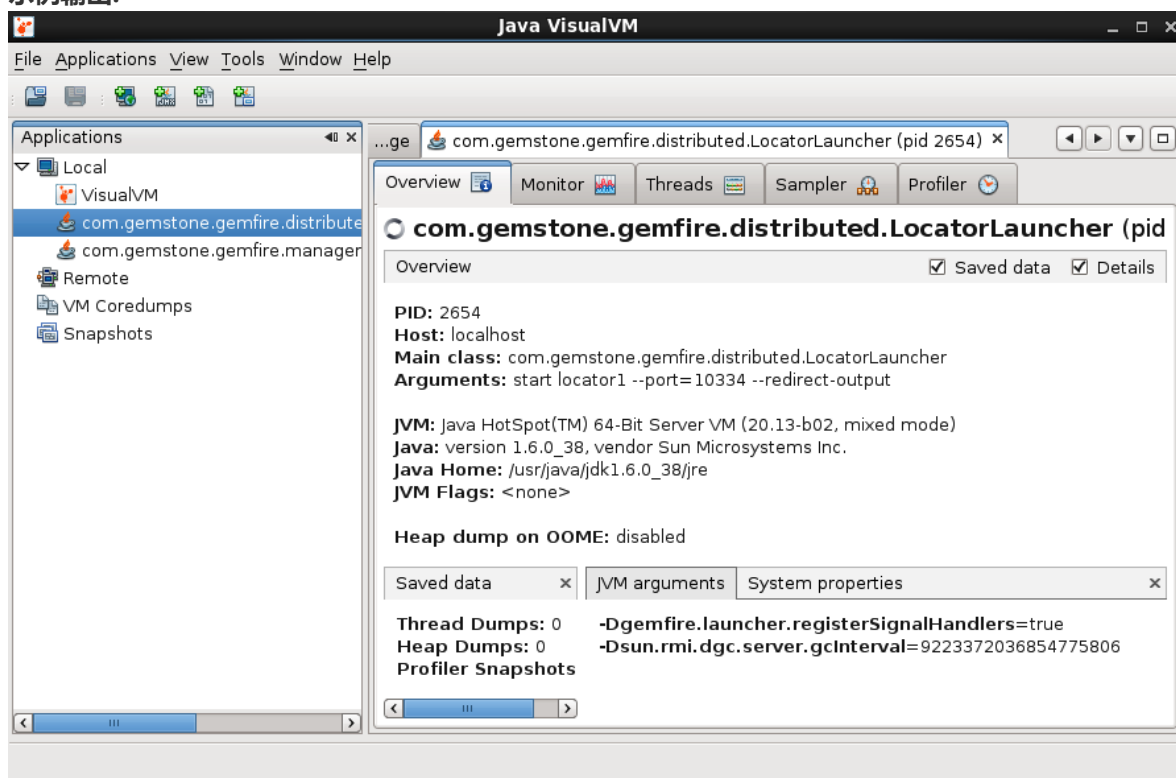
参数, 启动jvisualvm

名称	描述
--J	VM-option传递给生成的CacheServer VM。例如： <code>--J=-Dfoo.bar=true</code> 将foo.bar设置为`true`。

示例命令:

```
1 | start jvisualvm
```

示例输出:



start locator {#start_locator}

启动定位器。

该命令创建一个以定位符命名的子目录和日志文件。如果定位器检测到不存在其他JMX Manager，则定位器将自动启动嵌入式JMX Manager并将当前的 gfsh 会话连接到JMX Manager。

注意: 在启动gfsh之前必须设置 `JAVA_HOME` 才能使用此命令。

此外，如果gfsh尚未连接到JMX Manager，则gfsh控制台将自动连接到由新定位器启动的新嵌入式JMX Manager。

注意: 当在定位器启动期间指定 `--max-heap` 和 `--initial-heap` 时，Geode的资源管理器在内部指定其他GC参数。如果您不希望资源管理器设置其他默认GC属性，请使用 `-Xms` 和 `-Xmx` JVM选项。有关详细信息，请参阅[使用资源管理器控制堆使用](#)。

可用性: Online or offline.

句法:

```
1 start locator --name=value [--bind-address=value] [--force(=value)]
2   [--groups=value(,value)*] [--hostname-for-clients=value]
3   [--locators=value] [--log-level=value] [--mcast-address=value] [--mcast-
port=value] [--port=value] [--dir=value]
4   [--properties-file=value] [--security-properties-file=value] [--initial-
heap=value] [--max-heap=value]
5   [--connect(=value)] [--enable-cluster-configuration(=value)] [--load-
cluster-configuration-from-dir(=value)]
6   [--cluster-config-dir=value] [--redirect-output(=value)]
7   [--http-service-port=value] [--http-service-bind-address=value]
8   [--j=value(,value)*]
```

参数, 启动定位器

名称	描述	默认值
--name	用于此Geode定位器服务的名称。 如果未指定，gfsh将生成随机名称。	
--bind-address	定位器绑定的IP地址。	绑定到所有地址
--force	是否允许覆盖先前定位器运行的PID文件。	false
--groups	组定位器将成为其中的一部分。	
--hostname-for-clients	将发送到客户端的主机名或IP地址，以便它们可以连接到此定位器。	使用 <code>bind-address</code>
--locators	此定位器用于加入适当的Geode集群的定位器列表。	
--log-level	记录到定位器日志文件的输出级别。 日志级别的可能值包括：ALL，TRACE，DEBUG，INFO，WARN，ERROR，FATAL，OFF。	
--mcast-address	用于绑定UPD套接字以进行多播联网的IP地址或主机名，以便定位器可以定位Geode集群中的其他成员。 如果mcast-port为零，则忽略mcast-address。	
--mcast-port	用于多播网络的端口，因此定位器可以定位Geode集群的其他成员。 零值禁用mcast。	
--port	定位器的监听端口。	10334
--dir	将启动并运行定位器的目录。	<code>./<locator-member-name></code>
--properties-file	指定 <code>gemfire.properties</code> 文件以配置定位器的集群。 文件的路径应该是绝对的或相对于gfsh的工作目录。	
--security-properties-file	<code>gfsecurity.properties</code> 文件，用于在集群中配置Locator的安全配置。 文件的路径可以是gfsh的工作目录的绝对路径或相对路径。	
--initial-heap	Size的格式与 <code>-Xmx</code> / <code>-Xms</code> JVM选项相同。 注意: 如果使用 <code>-J-Xms</code> 和 <code>-J-Xmx</code> JVM属性而不是 <code>-initial-heap</code> 和 <code>-max-heap</code> ，则Geode不使用默认的JVM资源管理属性。 如果使用JVM属性，则必须手动为驱逐，垃圾收集，堆百分比等指定所有属性。	
--max-heap	Size的格式与 <code>-Xmx</code> / <code>-Xms</code> JVM选项相同。 注意: 如果使用 <code>-J-Xms</code> 和 <code>-J-Xmx</code> JVM属性而不是 <code>-initial-heap</code> 和 <code>-max-heap</code> ，则Geode不使用默认的JVM资源管理属性。 如果使用JVM属性，则必须手动为驱逐，垃圾收集，堆百分比等指定所有属性。	
--connect	当connect设置为false时，gfsh不会自动连接到使用此命令启动的定位器。	true
--enable-cluster-configuration	启用集群配置行为，其中定位器维护集群的所有成员的配置。 请参见 群集配置服务概述 。	true
--load-cluster-configuration-from-dir	已过时。使用gfsh import cluster-configuration实现此功能。 从shared-config目录加载集群配置。（当设置为false时，配置将从定位器用于持久保存配置的内部持久区域的磁盘存储区加载。）	false
--cluster-config-dir	群集配置服务用于将群集配置存储在文件系统上的目录	cluster-config
--redirect-output	如果为true，则将标准输出和标准错误重定向到定位器日志文件。 如果指定没有值，则该值设置为true。	false
--http-service-port	指定HTTP服务端口。	7070
--http-service-bind-address	指定HTTP服务绑定到的IP地址。	本地主机的地址
--j	参数传递给定位器将运行的JVM。 例如，指定 <code>--J=Dfoo.bar=true</code> 将属性“foo.bar”设置为“true”。 注意: 如果要传递的参数包含空格或逗号，请将选项括在单引号中。 例如： <code>start locator --name=locator1 --port=9009 --mcast-port=0\ --J='-Dgemfire.remote-locators=192.0.2.0[9009],192.0.2.1[9009]'</code>	none

示例命令:

```
1 | start locator --name=locator1
```

start pulse {#start_pulse}

在用户的默认系统浏览器中启动Geode Pulse监控仪表板工具，并将用户导航到登录页面（登录页面）。

有关Geode Pulse的更多信息，请参阅[Geode Pulse](#)。

可用性: Online or offline.

句法:

```
1 | start pulse [--url=value]
```

参数, 启动pulse

名称	描述	默认值
--url	Pulse Web应用程序的URL	http://localhost:7070/pulse

示例命令:

```
1 | start pulse
2 | start pulse --url=http://gemfire.example.com:7070/pulse
```

示例输出: 有关Pulse的示例，请参阅[Geode Pulse](#)。

start server {#start_server}

启动Geode缓存服务器进程。

注意: 如果在服务器启动期间指定了--max-heap和--initial-heap，则会代表您指定其他GC参数。如果您不希望设置其他默认GC属性，则使用 -Xms 和 -Xmx JVM选项来设置这些参数。有关详细信息，请参阅[使用资源管理器控制堆使用](#)。

可用性: Online or offline.

句法:

```
1 | start server --name=value [--assign-buckets(=value)] [--bind-address=value]
2 |   [--cache-xml-file=value] [--classpath=value] [--disable-default-
  server(=value)]
3 |   [--disable-exit-when-out-of-memory(=value)] [--enable-time-
  statistics(=value)]
4 |   [--force(=value)] [--include-system-classpath(=value)] [--properties-
  file=value]
5 |   [--security-properties-file=value] [--groups=value(,value)*]
6 |   [--locators=value] [--locator-wait-time=value] [--log-level=value]
7 |   [--mcast-address=value] [--mcast-port=value] [--memcached-port=value]
8 |   [--memcached-protocol=value] [--rebalance(=value)] [--server-bind-
  address=value]
9 |   [--server-port=value] [--spring-xml-location=value]
10 |  [--statistic-archive-file=value] [--dir=value] [--initial-heap=value]
11 |  [--max-heap=value] [--use-cluster-configuration(=value)] [--
  J=value(,value)*]
12 |  [--critical-heap-percentage=value] [--critical-off-heap-
  percentage=value]
13 |  [--eviction-heap-percentage=value] [--eviction-off-heap-
  percentage=value]
14 |  [--hostname-for-clients=value] [--max-connections=value]
15 |  [--message-time-to-live=value] [--max-message-count=value] [--max-
  threads=value]
```

```
16      [--socket-buffer-size=value] [--lock-memory=value] [--off-heap-memory-  
size=value]  
17      [--start-rest-api=value] [--redirect-output(=value)]  
18      [--http-service-port=value] [--http-service-bind-address=value]  
19      [--user=value] [--password=value]
```

参数, 启动服务器

名称	描述	默认值
--name	此服务器的成员名称。如果未指定，gfsh将生成随机名称。	
--assign-buckets	是否在服务器启动时将桶分配给缓存的分区区域。	false
--bind-address	服务器绑定的IP地址。	绑定到所有本地地址
--cache-xml-file	指定用于在创建缓存时初始化缓存的XML文件或资源的名称。	
--classpath	应用程序类在核心jar文件之后添加到服务器的CLASSPATH。有关详细信息，请参阅 设置CLASSPATH 。	
--include-system-classpath	如果为true，则在服务器的CLASSPATH上包含System CLASSPATH，因为默认情况下不包含System CLASSPATH。如果指定没有值，则该值设置为true。	false
--disable-default-server	W除此之外，默认情况下将启动缓存服务器。如果指定的参数没有值，则该值设置为true。如果设置为true，则缓存服务器充当对等方。	false
--disable-exit-when-out-of-memory	发生OutOfMemoryError时阻止JVM退出。	false
--enable-time-statistics	是否打开Geode操作收集其他基于时间的统计信息。	true
--properties-file	gemfire.properties 文件，用于配置服务器的集群。文件的路径可以是gfsh工作目录的绝对路径或相对路径。	
--security-properties-file	gfsecurity.properties 文件，用于在集群中配置服务器的安全配置。文件的路径可以是gfsh目录的绝对路径或相对路径。	
--groups	缓存服务器的组名称。	
--force	是否允许覆盖先前高速缓存服务器运行的PID文件。	false
--locators	设置高速缓存服务器用于加入适当的Geode集群的定位器列表。	
--locator-wait-time	设置服务器在放弃之前等待定位器在启动期间可用的秒数。	0
--log-level	设置记录到Cache Server日志文件的输出级别。日志级别的可能值包括：ALL，TRACE，DEBUG，INFO，WARN，ERROR，FATAL，OFF。	

名称	描述	默认值
--mcast-address	用于绑定UDP套接字以进行多播联网的IP地址或主机名，以便缓存服务器可以找到Geode集群中的其他成员。如果mcast-port为零，则忽略mcast-address。	
--mcast-port	设置用于多播联网的端口，以便缓存服务器可以找到Geode群集的其他成员。零值禁用mcast。	
--memcached-port	如果指定且非零，则设置嵌入式Gemcached服务器的端口号并启动Gemcached服务器。	
--memcached-protocol	设置嵌入式Gemcached服务器使用的协议。有效值为BINARY和ASCII。如果省略此属性，则使用ASCII协议。	
--server-bind-address	覆盖此服务器将侦听客户端连接的bind-address。在多宿主服务器环境中设置此选项以区分与客户端的通信。设置空字符串("")的值使用bind-address的值。	bind-address 的值
--server-port	服务器端口将侦听客户端连接。	40404
--spring-xml-location	指定用于引导和配置Geode Server的Spring XML配置文件的位置。此配置文件可以存在于CLASSPATH（默认）或Spring的Resource（Loader）位置说明符支持的任何位置（例如，classpath:，file: 等）。ResourceLoader在 Spring文档 中描述。	
--rebalance	是否在Geode集群中启动重新平衡。	false
--dir	指定运行服务器的目录。此目录将写入您启动gfsh的位置。	如果未指定，则以服务器命名该目录。
--statistic-archive-file	写入统计样本的文件。例如：“StatisticsArchiveFile.gfs”。必须定义为将归档存储到文件。空字符串（默认）禁用统计信息存档。	not set
--initial-heap	堆的初始大小，格式与JVM -Xms参数相同。 注意: 如果使用--J=-Xms 和 --J=-Xmx JVM属性而不是--initial-heap和--max-heap，那么Geode不使用默认的JVM资源管理属性。如果使用JVM属性，则必须手动为驱逐，垃圾收集，堆百分比等指定所有属性。	
--max-heap	堆的最大大小，格式与JVM -Xmx参数相同。 注意: 如果使用--J=-Xms 和 --J=-Xmx JVM属性而不是--initial-heap和--max-heap，那么Geode不使用默认的JVM资源管理属性。如果使用JVM属性，则必须手动为驱逐，垃圾收集，堆百分比等指定所有属性。	

名称	描述	默认值
--J	参数传递给运行缓存服务器的JVM。例如， <code>--J=-Dfoo.bar=true</code> 会将属性“foo.bar”设置为“true”。如果要传递的参数包含空格或逗号，请将选项括在单引号中。	
--use-cluster-configuration	指定服务器是否从定位器请求群集配置。请参见 群集配置服务概述 。	true
--critical-heap-percentage	设置堆的百分比等于或高于该百分比，由于垃圾收集暂停或内存不足异常，缓存被认为有可能无法运行。超过阈值，需要堆空间的操作将抛出一个 <code>LowMemoryException</code> 。此功能需要额外的VM标志才能正常执行; 你必须设置 <code>--initial-heap</code> 和 <code>--max-heap</code> 或相应的JVM属性来使用这个阈值。您还必须将 <code>--max-heap</code> 和 <code>--initial-heap</code> 设置为相同的值。	0 (没有强制执行关键堆阈值)
--critical-off-heap-percentage	由于内存不足异常，高速缓存被认为有可能无法运行的危险时使用的堆外内存百分比。超过阈值，需要堆空间的操作将抛出一个 <code>LowMemoryException</code> 。	0 (没有严格的堆外阈值强制执行)
--eviction-heap-percentage	设置应在HeapLRU驱逐配置的区域上开始驱逐的堆的百分比。更改此值可能会导致驱逐立即开始。在任何给定时间只允许对此属性或关键堆百分比进行一次更改，并且在允许下一次更改之前将完全实现其效果。此功能需要额外的VM标志才能正常执行; 你必须设置 <code>--initial-heap</code> 和 <code>--max-heap</code> 或相应的JVM属性来使用这个阈值。您还必须将 <code>--max-heap</code> 和 <code>--initial-heap</code> 设置为相同的值。	0，如果没有使用堆eviction配置区域，则将 <code>critical-heap-percentage</code> 设置为非零值，比该值小5%。如果未配置 <code>critical-heap-percentage</code> ，则为80%。
--eviction-off-heap-percentage	在为堆外和HeapLRU驱逐配置的区域上应该开始驱逐的堆外存储器的百分比。更改此值可能会导致驱逐立即开始。在任何给定时间，只允许对此属性或关键堆外百分比进行一次更改，并且在允许下一次更改之前将完全实现其效果。	0，如果没有区域配置堆驱逐如果 <code>critical-off-heap-percentage</code> 设置为非零值，则比该值小5%。80%，如果没有配置 <code>critical-off-heap-percentage</code> 。
--hostname-for-clients	设置定位器将为客户端提供的IP地址或主机名。客户端使用该地址连接到服务器。当客户端使用不同的地址与服务器连接而不是 <code>bind-address</code> 时，设置此值，因为这些客户端可能与私有云或多宿主环境中的服务器连接。不指定此选项或将此选项设置为空字符串 ("") 会导致将 <code>bind-address</code> 提供给客户端。	
--max-connections	设置允许的最大客户端连接数。达到最大值时，缓存服务器将停止接受连接	

名称	描述	默认值
--message-time-to-live	设置客户端队列中的消息将过期的时间（以秒为单位）。	
--max-message-count	设置可以在客户端队列中排队的最大消息数。	
--max-threads	设置此缓存服务器中允许的最大线程数，以便为客户端请求提供服务。默认值为0会导致缓存服务器为每个客户端连接专用一个线程。	
--socket-buffer-size	设置此CacheServer的套接字连接的缓冲区大小（以字节为单位）。默认值为32768字节。	
--lock-memory	（仅限Linux）如果为true，则成员的堆和堆外内存将锁定在RAM中，从而阻止它们被分页到磁盘。您必须增加相关的ulimit操作系统资源，以允许操作系统锁定足够大小的内存块。	false
--off-heap-memory-size	用于存储区域值的整数堆外堆内存。以千兆字节为单位，后缀为g，或者后缀为m的兆字节。例如，使用--off-heap-memory-size=2g分配2 GB的堆外空间。默认值0不使用任何堆外内存。	0
--start-rest-api	如果为true，则启动REST API服务。	false
--redirect-output	如果为true，则将标准输出和标准错误重定向到服务器日志文件。如果指定没有值，则该值设置为true。	false
--http-service-port	指定HTTP服务端口。	7070
--http-service-bind-address	指定HTTP服务绑定到的IP地址。	本地主机的地址
--user	用于对群集进行身份验证的凭据的用户名。如果指定，如果未指定--password选项，则gfsh将提示输入密码。	
--password	用于对群集进行身份验证的凭据的密码部分。	

例子

```

1 gfsh>start server --name=server1
2 gfsh>start server --name=server2 --server-port=40405

```

status {#status}

检查群集配置服务和Geode成员进程的状态，包括定位器，网关接收器，网关发件人和服务器。

- **status cluster-config-service**
显示集群配置服务的状态。
- **status gateway-receiver**
显示指定网关接收器的状态。
- **status gateway-sender**
显示指定网关发件人的状态。
- **status locator**
显示指定定位器的状态。
- **status server**
显示指定的Geode缓存服务器的状态。

status cluster-config-service {#status_cluster_config_service}

显示集群配置服务的状态。

显示enable-cluster-configuration设置为 `true` 的所有定位器上的集群配置服务的状态。

可用性: Online. 您必须在 `gfsh` 中连接到JMX Manager成员才能使用此命令。

句法:

```
1 | status cluster-config-service
```

示例命令:

```
1 | status cluster-config-service
```

示例输出:

```
1 | gfsh>status cluster-config-service
2 | Status of shared configuration on locators
3 |
4 |      Name      | Status
5 | ----- | -----
6 | locator8 | RUNNING
```

status gateway-receiver {#status_gateway_receiver}

显示指定网关接收器的状态。

可用性: Online. 您必须在 `gfsh` 中连接到JMX Manager成员才能使用此命令。

句法:

```
1 | status gateway-receiver [--groups=value(,value)*] [--members=value(,value)*]
```

名称	描述
--groups	用于显示状态的网关接收器组。
--members	要显示其状态的网关接收器的名称或ID。

Table 1. 状态网关 - 接收器参数

示例命令:

```
1 status gateway-receiver --groups=LN-Group1
2 status gateway-receiver --members=server1
```

示例输出:

```
1 gfsh>status gateway-receiver
2 Member          | Port  | Status
3 -----|-----|-----
4 pc13(8151)<v2>:26518 | 26837 | Running
5 pc13(8175)<v4>:53787 | 23753 | Running
6 pc13(8164)<v3>:24081 | 25457 | Running
7
8 Member          | Error
9 -----|-----
10 pc13(Manager:8124)<v1>:52410 | GatewayReceiver is not available or already
   stopped
11 pc13(8130):8180          | GatewayReceiver is not available or already
   stopped
12
13
14 gfsh>status gateway-receiver --members=pc13(8151)<v2>:50130
15 Member          | Port  | Status
16 -----|-----|-----
17 pc13(8151)<v2>:50130 | 28592 | Running
18
19 gfsh>status gateway-receiver --group=RG1
20 Member          | Port  | Status
21 -----|-----|-----
22 pc13(8151)<v2>:19450 | 27815 | Running
23 pc13(8175)<v4>:14139 | 27066 | Running
24 pc13(8164)<v3>:45150 | 29897 | Running
```

status gateway-sender {#status_gateway_sender}

显示指定网关发件人的状态。

可用性: Online. 您必须在 gfsh 中连接到 JMX Manager 成员才能使用此命令。

句法:

```
1 status gateway-sender --id=value [--groups=value(,value)*]
2 [--members=value(,value)*]
```


名称	描述
--id	<i>Required.</i> 网关发件人的ID。
--groups	要显示状态的网关发件人组。多个成员组的逗号分隔列表。
--members	要显示其状态的网关发件人的名称/ID。

Table 2. 状态网关 - 发件人参数

示例命令:

```
1 | status gateway-receiver receiver1-LN --groups=LN-Group1;
```

示例输出:

```
1 gfsh>status gateway-sender --id=ln_Serial
2 Member | Type | Runtime Policy | Status
3 -----|-----|-----|-----
4 pc13(8175)<v4>:21449 | Serial | Secondary | Not Running
5 pc13(8151)<v2>:40761 | Serial | Secondary | Not Running
6 pc13(8164)<v3>:33476 | Serial | Secondary | Not Running
7
8 Member | Error
9 -----|-----
10 pc13(8130):2365 | GatewaySender is not available
11 pc13(Manager:8124)<v1>:43821 | GatewaySender is not available
12
13 gfsh>status gateway-sender --id=ln_Serial --members=pc13(8151)<v2>:7411
14 Member | Type | Runtime Policy | Status
15 -----|-----|-----|-----
16 pc13(8151)<v2>:7411 | Serial | Secondary | Not Running
17
18 gfsh>status gateway-sender --id=ln_Serial --members=pc13(8151)<v2>:7411
19 Member | Type | Runtime Policy | Status
20 -----|-----|-----|-----
21 pc13(8151)<v2>:7411 | Serial | Primary | Running
22
23 gfsh>status gateway-sender --id=ln_Serial --groups=Serial_Sender
24 ==>
25 Member | Type | Runtime Policy | Status
26 -----|-----|-----|-----
27 pc13(8151)<v2>:44396 | Serial | Secondary | Not Running
28 pc13(8164)<v3>:29475 | Serial | Secondary | Not Running
29 Member | Error
30 -----|-----
31 pc13(8186)<v5>:45840 | GatewaySender is not available
```

status locator {#status_locator}

显示指定定位器的状态。

状态将是以下之一:

- started
- online
- offline

- not responding

可用性: Online or offline. 如果要在脱机时获取定位器的状态，请使用 `--dir` 选项。

句法:

```
1 | status locator [--name=value] [--host=value] [--port=value] [--dir=value]
```

名称	描述	默认值
--name	要显示其状态的定位器的名称/ ID。 您必须连接到JMX Manager才能使用此选项。 可用于获取远程定位器的状态。 请参阅 使用gfsh通过HTTP或HTTPS管理远程群集 。	
--host	运行定位器的主机名或IP地址。	
--port	定位器正在侦听的端口。	10334
--dir	启动定位器的目录。	当前目录

Table 3. 状态定位器参数

示例命令:

```
1 | status locator
2 | status locator --name=locator1
```

status server {#status_server}

显示指定的Geode缓存服务器的状态。

可用性: Online or offline. 如果要在脱机时获取服务器的状态，请使用 `--dir` 选项。

句法:

```
1 | status server [--name=value] [--dir=value]
```

名称	描述	默认值
--name	要显示其状态的高速缓存服务器的名称或ID。 您必须连接到JMX Manager才能使用此选项。 可用于获取远程服务器的状态。 请参阅 使用gfsh通过HTTP或HTTPS管理远程群集 。	
--dir	启动Geode缓存服务器的目录。	当前目录

Table 4. 状态服务器参数

示例命令:

```
1 status server
2 status server --name=server1
```

stop {#stop}

停止网关接收器，网关发送器，定位器和服务器。

- **stop gateway-receiver**
停止一个或多个成员的网关接收器。
- **stop gateway-sender**
停止指定成员或指定成员组成员上具有给定ID的网关发件人。
- **stop locator**
停止定位器。
- **stop server**
停止Geode缓存服务器。

stop gateway-receiver {#stop_gateway_receiver}

停止一个或多个成员的网关接收器。

可用性: Online. 您必须在 gfsh 中连接到JMX Manager成员才能使用此命令。

句法:

```
1 stop gateway-receiver [--groups=value(,value)*] [--members=value(,value)*]
```

名称	描述
--groups	停止Gateway接收器的成员组。对多个成员组使用逗号分隔列表。
--members	要停止网关接收器的成员的名称/ID。

Table 1. 停止网关 - 接收器参数

示例命令:

```
1 stop gateway-receiver --members=receiver1-LN
2 stop gateway-receiver --groups=LN-Group1
```

示例输出:

```
1 gfsh>stop gateway-receiver
2      Member              | Result | Message
3 -----|-----|-----
      -----|-----|-----
```

```
4 pc13(2266)<v6>:56852 | OK | GatewayReceiver is stopped on member
pc13(2266)<v6>:56852
5 pc13(Manager:2242)<v5>:57631| Error | GatewayReceiver is not available on
member pc13(Manager:2242)<v5>:57631
6 pc13(2275)<v7>:47480 | OK | GatewayReceiver is stopped on member
pc13(2275)<v7>:47480
7 pc13(2293)<v8>:55472 | OK | GatewayReceiver is stopped on member
pc13(2293)<v8>:55472
8
9 gfsh>stop gateway-receiver --members=pc13(2266)<v14>:36579
10 GatewayReceiver is stopped on member pc13(2266)<v14>:36579
11
12 gfsh>stop gateway-receiver --groups=RG1
13 Member | Result | Message
14 -----|-----|-----
15 pc13(2275)<v23>:27484| OK | GatewayReceiver is stopped on member
pc13(2275)<v23>:27484
16 pc13(2293)<v24>:55810| OK | GatewayReceiver is stopped on member
pc13(2293)<v24>:55810
17 pc13(2266)<v22>:4522 | OK | GatewayReceiver is stopped on member
pc13(2266)<v22>:4522
```

stop gateway-sender {#stop_gateway_sender}

停止指定成员或指定成员组成员上具有给定ID的网关发件人。

警告: 在并行网关发件人上使用 `stop gateway-sender` 命令（或等效的 `GatewaySender.stop()` API）要小心。我们建议关闭整个成员，以确保将分区区域事件正确地故障转移到其他网关发件人成员，而不是在成员上停止单个并行网关发件人。在单个并行网关上使用此命令发送方可能在事件丢失时发生。有关详细信息，请参阅[停止网关发件人](#)。

可用性: Online. 您必须在 `gfsh` 中连接到JMX Manager成员才能使用此命令。

句法:

```
1 stop gateway-sender --id=value [--groups=value(,value)*] [--
members=value(,value)*]
```

名称	描述
--id	<i>Required.</i> 网关发件人的ID。
--groups	需要停止的网关发件人的成员组。
--members	要停止Gateway Sender的成员的名称/ID。

Table 2. 停止网关 - 发件人参数

示例命令:

```
1 stop gateway-sender --id=1n --members=server1
```

示例输出:

```
1 gfsh>stop gateway-sender --id=ln
2 Member | Result | Message
3 -----|-----|-----
4 pc13(5184)<v7>:41914 | OK | GatewaySender ln is stopped on
member pc13(5184)<v7>:41914
5 pc13(5192)<v8>:25704 | OK | GatewaySender ln is stopped on
member pc13(5192)<v8>:25704
6 pc13(5174)<v6>:53996 | OK | GatewaySender ln is stopped on
member pc13(5174)<v6>:53996
7 pc13(Manager:5148)<v5>:64040 | Error | GatewaySender ln is not available
on member pc13(Manager:5148)<v5>:64040
8
9 gfsh>stop gateway-sender --id=ln --members=pc13(5174)<v14>:17819
10 GatewaySender ln is stopped on member pc13(5174)<v14>:17819
11
12 gfsh>stop gateway-sender --id=ln --groups=SenderGroup1
13
14 Member | Result | Message
15 -----|-----|-----
16 pc13(5174)<v18>:63332 | OK | GatewaySender ln is stopped on member
pc13(5174)<v18>:63332
17 pc13(5184)<v19>:20055 | OK | GatewaySender ln is stopped on member
pc13(5184)<v19>:20055
18 pc13(5192)<v20>:14622 | OK | GatewaySender ln is stopped on member
pc13(5192)<v20>:14622
```

stop locator {#stop_locator}

停止定位器。

注意: 必须指定命令行选项之一 `--name` 或 `--dir` 以标识要停止的定位符。

可用性: Online or offline. 如果要在脱机时停止定位器，请使用 `--dir` 选项。

句法:

```
1 | stop locator --name=value | --dir=value
```

名称	描述	默认值
--name	要停止的定位器的Geode成员名称。 您必须连接到MX Manager才能使用此选项。 可用于停止远程定位器。 请参阅 使用gfsh通过HTTP或HTTPS管理远程群集 。	
--dir	启动定位器的目录。	当前目录

Table 3. 停止定位器参数

示例命令:

```
1 | stop locator --name=locator3
```

示例输出:

```
1 | gfsh>stop locator --name=locator3
2 | Stopping Locator running in /Users/test/locator3 on 192.0.2.0[10334] as
   | locator3...
3 | Process ID: 71531
4 | Log File: /Users/test/locator3/locator3.log
5 | ...
6 | No longer connected to 192.0.2.0[1099].
7 | gfsh>stop locator --dir=loc2
8 | Stopping Locator running in /Users/test/loc2 on 192.0.2.0[10334] as loc2...
9 | Process ID: 71714
10 | Log File: /Users/test/loc2/loc2.log
11 | ...
12 | No longer connected to 192.0.2.0[1099].
```

stop server {#stop_server}

停止Geode缓存服务器。

可用性: Online or offline. 如果要在脱机时停止缓存服务器，请使用 --dir 选项。

句法:

```
1 | stop server [--name=value] [--dir=value]
```

名称	描述	默认值
--name	要停止的Geode缓存服务器的名称/ ID。 您必须连接到JMX Manager才能使用此选项。 可用于停止远程服务器。 请参阅 使用gfsh通过HTTP或HTTPS管理远程群集 。	
--dir	启动Geode缓存服务器的目录。	当前目录

Table 4. 停止服务器参数

示例命令:

```
1 | stop server --name=server1
2 | stop server --dir=server1
```

undeploy {#undeploy}

使用 `deploy` 命令取消部署在成员或组上部署的JAR文件。

如果未指定 `--jars`，该命令将取消部署所有已部署的JAR。如果未指定 `--groups`，则该命令适用于整个集群。请注意，此命令无法卸载在部署期间加载的类。成员应该重新启动。

可用性: Online. 您必须在 `gfsh` 中连接到JMX Manager成员才能使用此命令。

句法:

```
1 | undeploy [--jars=value(,value)*] [--groups=value(,value)*]
```

名称	描述	默认值
--groups	将取消部署指定JAR的组。	取消部署将在所有成员上发生
--jars	JAR或JAR将被取消部署。	所有JAR都将被取消部署

Table 1. 取消部署参数

示例命令:

```
1 | undeploy --jars=domain-objects.jar
2 | undeploy --groups=Group1
```

示例输出:

```
1 | gfsh>undeploy --jars=domain-objects.jar
2 |
3 | Member      | Un-Deployed JAR      | Un-Deployed From JAR Location
4 | -----
5 | datanode10 | domain-objects.jar | /usr/local/gemfire/deploy/GF#domain-objects#1
6 | datanode11 | domain-objects.jar | /usr/local/gemfire/deploy/GF#domain-objects#1
7 |
8 |
9 | gfsh> undeploy --groups=Group1
10 |
11 | Member      | Un-Deployed JAR      | Un-Deployed From JAR Location
12 | -----
13 | datanode1 | group1_functions.jar | /usr/local/gemfire/deploy/GF#group1_functions.jar#1
14 | datanode1 | group1_dependencies.jar | /usr/local/gemfire/deploy/GF#group1_dependencies.jar#1
15 | datanode2 | group1_functions.jar | /usr/local/gemfire/deploy/GF#group1_functions.jar#1
16 | datanode2 | group1_dependencies.jar | /usr/local/gemfire/deploy/GF#group1_dependencies.jar#1
```

错误消息:

```
1 | No JAR Files Found
```

validate offline-disk-store {#validate_offline_disk_store}

验证脱机磁盘存储。

可用性: Offline.

句法:

```
1 | validate offline-disk-store --name=value --disk-dirs=value(,value)*
```

名称	描述
--name	<i>Required.</i> 要验证的磁盘存储的名称。
--disk-dirs	<i>Required.</i> 先前已写入磁盘存储数据的目录。

Table 1. 验证脱机磁盘存储参数

示例命令:

```
1 | validate offline-disk-store --name=DiskStore2 --disk-dirs=data/dir3,data/dir4
```

version {#version}

显示产品版本信息。

可用性: Online or offline.

句法:

```
1 | version [--full]
```

名称	描述	默认值
--full	显示完整版本信息。	false

Table 1. 版本参数

示例命令:

```
1 | version
2 | version --full
```

示例输出:


```
1 gfsh>version
2 v8.0.0
3
4
5 gfsh>version --full
6 Java version: 8.0.0 build 48319 07/31/2014 17:26:09 PDT javac 1.8.0_1
7 Native version: native code unavailable
8 Source revision: 48319
9 Source repository: gemfire/branches/cedar_dev_Oct12
10 Running on: /192.0.2.0, 1 cpu(s), amd64 Linux 2.6.32-38-generic
```

Creating and Running gfsh Command Scripts

{#Creating_and_Running_gfsh_Command_Scripts}

gfsh提供了几种在脚本环境中运行命令的方法。

Running gfsh Scripts

您可以创建和运行包含要执行的gfsh命令的脚本。要执行该脚本，请使用gfsh [run](#)命令。例如：

```
1 gfsh run --file=mycommands.gfsh
```

注意：运行gfsh脚本时，将忽略交互式参数。您还可以将脚本设置为以安静模式运行以防止输出并指示脚本跳过它遇到的任何错误。

编写gfsh脚本时，命令历史记录文件会很有用。已成功执行的命令历史记录在运行gfsh的用户的主目录中的 `.gfsh.history` 文件中。您还可以使用 `history --file=your_file_name` 命令导出历史文件。

当用户从gfsh运行 `start server` 或 `start locator` 而不指定成员名称时，gfsh将自动选择一个随机成员名称。这对自动化很有用。

Running gfsh Commands on the OS Command Line

{#Running_gfsh_Commands_on_the_OS_Command_Line}

您可以通过在命令前加上 `gfsh` 直接从操作系统的提示符运行一些 `gfsh` 命令。这对于Unix shell或Windows批处理脚本非常有用。例如：

```
1 $ gfsh start locator --name=locator2 --port=10335
```

要在提示符下直接查看哪些gfsh命令可用：

```
1 $ gfsh help
```

在OS命令行上运行多个gfsh命令

要直接在命令行上运行多个命令，请在引号中使用 `-e` 选项，然后使用 `gfsh` 命令。例如：

```

1 prompt>gfsh -e "start locator --name=locator1" -e "start server --
   name=server1"
2
3 prompt>gfsh -e "start jconsole"
4
5 prompt>gfsh -e "connect --locator=remotehost[10334]" -e "rebalance" -e "gc"

```

Mapping cache.xml Elements to gfsh Configuration Commands

{#Mapping_cache_xml_Elements_to_gfsh_Configuration_Commands}

You can configure a Geode cluster using either cache.xml files, or you can use gfsh and the cluster configuration service to configure a cluster. This table maps `cache.xml` elements to the gfsh commands that configure and manage a cluster.

cache.xml 元素	gfsh 命令
,	start serverstatus serverstop serveralter runtime
	create async-event-queue list async-event-queues
	configure pdx
	create regionalter regiondestroy regiondescribe regionlist regionsrebalance
	create indexdestroy indexlist indexes
	create disk-storealter disk-storebackup disk-storecompact disk-storecompact offline-disk-storedescribe disk-storedescribe offline-disk-storedestroy disk-storelist disk-storesrevoke missing-disk-stores show missing-disk-storesvalidate offline-disk-store

Table 1. 将cache.xml元素迁移到gfsh命令