



# 目 录

目 录.....	2
<b>第 3 章 Linux 平台 Oracle RAC-Oracle Standalone 复制.....</b>	<b>3</b>
3.1 目标概述.....	3
3.2 GoldenGate 在 Linux 平台的安装.....	4
3.2.1 安装前准备工作.....	4
3.2.1.1 创建 GoldenGate 操作系统用户.....	4
3.2.1.2 准备集群文件系统.....	4
3.2.1.3 创建安装目录分配存储空间.....	4
3.2.1.4 设置 GoldenGate 用户的环境变量.....	4
3.2.1.5 安装 GoldenGate.....	4
3.2.1.6 创建 GoldenGate 数据库用户 ggs 及授 dba 权限.....	5
3.2.1.7 配置 ASM 的连通性.....	5
3.2.2 使用 Oracle clusterware 管理 GoldenGate.....	6
3.2.2.1 添加一个应用程序 VIP 资源.....	6
3.2.2.2 创建一个 action 程序.....	7
3.2.2.3 创建一个应用程序 profile.....	9
3.2.2.4 启动应用程序.....	10
3.2.2.5 管理应用程序.....	10
3.2.2.6 清理应用程序.....	10
3.2.2.7 使用 ASM 存放 trail 文件注意事项.....	11
3.2.3 配置源端数据库.....	11
3.3 配置源端进程组.....	13
3.3.1 配置 MGR 进程组.....	13
3.3.1.1 动态端口参数设置.....	13
3.3.1.2 自动启动参数设置.....	13
3.3.1.3 trail 文件维护.....	14
3.3.2 配置 Extract 进程组.....	14
3.3.3 配置 Pump 进程组.....	15
3.4 配置目标端进程.....	16
3.4.1 配置目标端 MGR 进程组.....	16
3.4.2 配置目标端 Replicat 进程组.....	16
3.5 DML 测试.....	17

## 第 3 章 Linux 平台 Oracle RAC-Oracle Standalone 复制

### 3.1 目标概述

在 Vmware 虚拟机模拟环境下利用 GoldenGate 实现数据同步，GoldenGate 可以复制 DML 和 DDL 操作。

#### 1. Oracle RAC 环境:

节点 1	节点 2
操作系统: Red Hat Linux 5.4	操作系统: Red Hat Linux 5.4
数据库 : Oracle 10g	数据库 : Oracle 10g
IP 地址: eth0:192.168.0.101 eth1:10.10.17.201	IP 地址: eth0:192.168.0.102 eth1:10.10.17.202
网关: 192.168.0.168	网关: 192.168.0.168
DNS: 192.168.0.168	DNS: 192.168.0.168
内存: 512G	内存: 512G
网卡: 2 块	网卡: 2 块
存储管理: Oracle ASM	

表 3-1

#### 2. Oracle Standalone 环境:

操作系统:	Red Hat Linux 5.4
数据库:	Oracle 10g
IP 地址:	192.168.0.104
网关:	192.168.0.168
DNS:	192.168.0.168

内存:	512G
网卡:	1 块

表 3-2

## 3.2 GoldenGate 在 Linux 平台的安装

### 3.2.1 安装前准备工作

到 <http://edelivery.oracle.com> 下载 GoldenGate For Linux 版本。

#### 3.2.1.1 创建 GoldenGate 操作系统用户

这里直接使用 Oracle 用户安装 GoldenGate，而不创建新的用户。

#### 3.2.1.2 准备集群文件系统

这里把 GoldenGate 安装到每一个 RAC 节点，GoldenGate 在每一个节点使用相同的目录结构，然后把 checkpoint 文件和 trail files 放到 ASM 文件系统来实现高可用性。

#### 3.2.1.3 创建安装目录分配存储空间

/ggs 目录为 GoldenGate 的安装目录

#### 3.2.1.4 设置 GoldenGate 用户的环境变量

在两个 RAC 节点 GoldenGate 用户的 .bash\_profile 文件分别加入把 GoldenGate 的目录加入到 PATH 中，另外

```
export PATH=/u01/ggs/10.0:$PATH
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/u01/ggs/10.0:$LD_LIBRARY_PATH
```

图 3-1

#### 3.2.1.5 安装 GoldenGate

把适合操作系统版本的 GoldenGate 上传到每个 RAC 节点，使用 unzip 命令解压，这样得到一个 tar 包：

```
[oracle@node1 10.0]$ unzip V18428-01\32bit\*.zip
Archive:  V18428-01(32bit).zip
  inflating: ggs_redhatAS50_x86_ora10g_32bit_v10.4.0.19_002.tar
```

图 3-2

使用命令 `tar -xvof <filename>.tar` 解压 GoldenGate 可以看到很多文件：

```

[oracle@node1 10.0]$ tar -xvof ggs_redhatAS50_x86_oral0g_32bit_v10.4.0.
r
mgr
ggsci
ggMessage.dat
help.txt
bencrypt.txt
  
```

图 3-3

在一个新的目录运行 GGSCI 会出现如下界面，这里在/u01/ggs 目录运行：

```

[oracle@node1 ggs]$ ggsci

Oracle GoldenGate Command Interpreter for Oracle
Version 10.4.0.19 Build 002
Linux, x86, 32bit (optimized), Oracle 10 on Sep 25 2009 12:49:31

Copyright (C) 1995, 2009, Oracle and/or its affiliates. All rights reserved.

GGSCI (node1) 1>
  
```

图 3-4

输入命令 create subdirs 创建 GoldenGate 的工作目录：

```

GGSCI (node1) 1> create subdirs

Creating subdirectories under current directory /u01/ggs

Parameter files           /u01/ggs/dirprm: created
Report files              /u01/ggs/dirrpt: created
Checkpoint files         /u01/ggs/dirchk: created
Process status files     /u01/ggs/dirpcs: created
SQL script files         /u01/ggs/dirsql: created
Database definitions files /u01/ggs/dirdef: created
Extract data files       /u01/ggs/dirdat: created
Temporary files          /u01/ggs/dirtmp: created
Veridata files           /u01/ggs/dirver: created
Veridata Lock files      /u01/ggs/dirver/lock: created
Veridata Out-Of-Sync files /u01/ggs/dirver/oos: created
Veridata Out-Of-Sync XML files /u01/ggs/dirver/oosxml: created
Veridata Parameter files /u01/ggs/dirver/params: created
Veridata Report files    /u01/ggs/dirver/report: created
Veridata Status files    /u01/ggs/dirver/status: created
Veridata Trace files     /u01/ggs/dirver/trace: created
Stdout files             /u01/ggs/dirout: created
  
```

图 3-5

### 3.2.1.6 创建 GoldenGate 数据库用户 ggs 及授 dba 权限

```
SQL> create user ggs identified by ggs default tablespace users temporary tablespace temp;

User created.

SQL> grant dba to ggs;

Grant succeeded.

SQL>
```

图 3-6

### 3.2.1.7 配置 ASM 的连通性

如果使用 Oracle 的 ASM 作为存储管理软件，那么我们就需要确保 GoldenGate 能够同样也能连接到 ASM 实例：

- 1、 确保 ASM 实例已经添加到 tnsnames.ora 文件；
- 2、 确保监听可以监听到连接 ASM 实例的请求，listener.ora 需要加入如下的内容：

```
SID_LIST_LISTENER_DARAN =
(SID_LIST =
(SID_DESC =
(SID_NAME = PLSExtProc)
(ORACLE_HOME = /u01/app/oracle/product/10.0/db_1)
(PROGRAM = extproc)
)
(SID_DESC =
(ORACLE_HOME = /u01/app/oracle/product/10.0/db_1)
(SID_NAME = +ASM1)
)
)
```

示例 3-1

## 3.2.2 使用 Oracle clusterware 管理 GoldenGate

### 3.2.2.1 添加一个应用程序 VIP 资源

我们通过使用 CRS 来管理 GoldenGate 的资源组，并且使用 RAC 的 vip 连接到 GoldenGate，一旦数据库的某一个节点宕掉，Oracle clusterware 将自动切换到另一个可用的节点。

下面是创建应用程序 vip 资源的步骤：

1. 使用安装 GoldenGate 的用户登录；
2. 为 GoldenGate VIP 资源创建一个 profile：

```
$ORA_CRS_HOME/bin/crs_profile create ggatevip \
-t application \
-a $ORA_CRS_HOME/bin/usrvip \
-o oi=eth0,ov=192.168.0.23,on=255.255.255.0
```

示例 3-2

命令解析:

- \$ORA\_CRS\_HOME : 是 clusterware 的 HOME 目录
- Ggatevip : 是你创建的应用程序 vip 的名字
- oi=eth0 : 是制定公用网卡为 eth0
- ov=192.168.0.23 : 设置虚拟 IP 为 192.168.0.23
- on=255.255.255.0: 设置虚拟 IP 的子网掩码为 255.255.255.0, 需要和公网 ip 的子网掩码相同。

你还可以设置更多的参数, 详情请查看 Oracle clusterware 文档。

```
[oracle@node1 ~]$ $ORA_CRS_HOME/bin/crs_profile -create ggatevip -t application  
-a $ORA_CRS_HOME/bin/usrvip -o oi=eth0,ov=192.168.0.23,on=255.255.255.0  
[oracle@node1 ~]$
```

图 3-7

3. 把这个资源注册到 CRS:

运行命令: \$ORA\_CRS\_HOME/bin/crs\_register ggatevip

```
[oracle@node1 ~]$ $ORA_CRS_HOME/bin/crs_register ggatevip  
[oracle@node1 ~]$
```

图 3-8

4. 把 vip 的所有权给 root, 执行命令:

在 root 下执行, [root@node1 bin]# ./crs\_setperm ggatevip -o root

```
[root@node1 bin]# ./crs_setperm ggatevip -o root  
[root@node1 bin]#
```

图 3-9

5. 为 Oracle 用户分配启动这个资源的权限:

```
[root@node1 bin]# ./crs_setperm ggatevip -u user:oracle:r-x  
[root@node1 bin]#
```

图 3-10

6. 通过 Oracle 用户启动这个资源:

```
[root@node1 bin]# su - oracle  
[oracle@node1 ~]$ $ORA_CRS_HOME/bin/crs_start ggatevip  
Attempting to start 'ggatevip' on member 'node2'  
Start of 'ggatevip' on member 'node2' succeeded.  
[oracle@node1 ~]$
```

图 3-11

7. 查看这个资源的状态。比如运行状态以及在那个节点上运行:

```
[oracle@node1 ~]$ $ORA_CRS_HOME/bin/crs_stat ggatevip -t
Name                Type           Target         State         Host
-----
ggatevip            application    ONLINE        ONLINE        node2
[oracle@node1 ~]$
```

图 3-12

### 3.2.2.2 创建一个 action 程序

action 程序必须放在每个节点的相同的目录结构下面（也可以放到共享磁盘上），必须可以接受三个参数：start, stop 和 check

- start 和 stop: 返回 0 成功, 1 失败;
- check: 返回 0 表示 GoldenGate 在运行, 1 表示不运行。

```
[oracle@node1 ggs]$ pwd
/u01/ggs
[oracle@node1 ggs]$ ls
10.0  dirdat  dirout  dirprm  dirsql  dirver
dirchk  dirdef  dirpcs  dirrpt  dirtmp  goldengate_action.scr
[oracle@node1 ggs]$
```

图 3-13

下面是这个示例程序的内容:

```
#!/bin/sh
#GoldenGate_action.scr
GGS_HOME=/ggate/10.0
start_delay_secs=5
#Include the GoldenGate home in the library path to start GGSCI
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:${GGS_HOME}
#Set the oracle home to the database to ensure GoldenGate will get the
#right environment settings to be able to connect to the database
export ORACLE_HOME=/u01/oracle/ora111
#check_process validates that a manager process is running at the PID
#that GoldenGate specifies
check_process () {
if ( [ f
"${GGS_HOME}/dirpcs/MGR.pcm" ] )
then
pid=`cut f8
"${GGS_HOME}/dirpcs/MGR.pcm"`
if [ ${pid} = `ps e
|grep ${pid} |grep mgr |cut d
" " f1`
]
then
#manager process is running on the PID - exit success
exit 0
else
```

```
#if the manager process has a low PID then the cut should use f2
if [ ${pid} = `ps e
|grep ${pid} |grep mgr |cut d
" " f2`
]
then
#manager process is running on the PID - exit success
exit 0
else
#manager process is not running on the PID
exit 1
fi
fi
else
#manager is not running because there is no PID file
exit 1
fi
}
#call_ggsci is a generic routine that executes a GGSCI command
call_ggsci () {
ggsci_command=$1
ggsci_output=`${GGS_HOME}/GGSCI << EOF
${ggsci_command}
exit
EOF`
}
case $1 in
'start')
call_ggsci 'start manager'
#there is a small delay between issuing the start manager command
#and the process being spawned on the OS - wait before checking
sleep ${start_delay_secs}
#check whether manager is running and exit accordingly
check_process
;;
'stop')
#attempt a clean stop for all nonmanager
processes
call_ggsci 'stop er *'
#ensure everything is stopped
call_ggsci 'stop er *!'
#stop manager without (y/n) confirmation
call_ggsci 'stop manager!'
#exit success
exit 0
;;
'check')
```

```
check_process
;;
Esac
```

示例 3-3

### 3.2.2.3 创建一个应用程序 profile

应用程序 profile 是一个键-值对文本文件，可以使用 \$ORA\_CRS\_HOME/bin/crs\_profile 来创建和操作这个文件。

#### 1. 创建 profile

使用 Oracle 用户执行下面的命令：

```
$ORA_CRS_HOME/bin/crs_profile
-create GoldenGate_app
-t application
-r ggatevip
-a /u01/ggs/GoldenGate_action.scr
-o ci=10
```

示例 3-4

```
[oracle@node1 ~]$ $ORA_CRS_HOME/bin/crs_profile -create goldengate_app -t applic
ation -r ggatevip -a /u01/ggs/goldengate_action.scr -o ci=10
```

图 3-14

命令解析：

- -create GoldenGate\_app: 应用程序的名字是 GoldenGate\_app
- -r ggatevip: ggatevip 必须在 GoldenGate 启动之前运行
- -a /u01/ggs/ GoldenGate\_action.scr: 指定 action 脚本的位置，在每一个实例必须可用
- -o ci=10: 检查的时间间隔设置为 10

#### 2. 注册应用程序，用 Oracle 用户执行：

```
[oracle@node1 public]$ $ORA_CRS_HOME/bin/crs_register goldengate_app
[oracle@node1 public]$
```

图 3-15

#### 3. 修改 GoldenGate\_app 应用程序的所有权，用 root 用户执行：

```
[root@node1 bin]# ./crs_setperm goldengate_app -o oracle
[root@node1 bin]#
```

图 3-16

#### 4. 允许 Oracle 用户启动应用程序 GoldenGate\_app:

如果你的 GoldenGate 用户和 clusterware 是一个用户，则无需执行下面的语句！

```
[root@node1 bin]# ./crs_setperm goldengate_app -u user:oracle:r-x
```

图 3-17

### 3.2.2.4 启动应用程序

在 Oracle 用户下，使用命令 `$ORA_CRSHOME/bin/crs_start GoldenGate_app`，可以启动应用程序。

```
[oracle@node1 ~]$ $ORA_CRSHOME/bin/crs_start goldengate_app
```

图 3-18

### 3.2.2.5 管理应用程序

在 GoldenGate 运行的时候，你想让 GoldenGate 在另一个服务器运行，你可以使用 `./crs_relocate -f GoldenGate_app` 接口使它强行漂移：

```
[oracle@node1 ~]$ $ORA_CRSHOME/bin/crs_relocate -f goldengate_app
```

图 3-19

停止 GoldenGate，使用 Oracle 用户执行：

```
[oracle@node1 ~]$ $ORA_CRSHOME/bin/crs_stop goldengate_app
```

图 3-20

### 3.2.2.6 清理应用程序

如果你想停止 Oracle clusterware 管理 GoldenGate，并且清理已经注册的资源，需要进行以下操作：

1. 以 Oracle 用户，停止 GoldenGate 应用程序：

```
[oracle@node1 ~]$ $ORA_CRSHOME/bin/crs_stop goldengate_app
```

图 3-21

2. 使用 Oracle 用户，停止 vip 资源：

```
[oracle@node1 ~]$ $ORA_CRSHOME/bin/crs_stop ggatevip
```

图 3-22

3. 取消注册 GoldenGate\_app 应用程序资源：

需要用 GoldenGate\_app 用户执行(这里使用的是 Oracle)。

```
[oracle@node1 ~]$ $ORA_CRSHOME/bin/crs_unregister goldengate_app
```

图 3-23

4. 取消注册 VIP 资源：(以 root 用户执行)

```
[root@node1 bin]# pwd
/u01/app/oracle/product/10.2.0/crs_1/bin
[root@node1 bin]# ./crs_unregister ggatevip
```

图 3-24

5. 到你创建 GoldenGate\_app 文件的节点，删除文件：（以 Oracle 用户执行）

```
[oracle@node1 ~]$ $ORA_CRS_HOME/bin/crs_profile -delete goldengate_app
```

图 3-25

6. 到你创建 vip 文件的节点，删除文件：（以 Oracle 用户执行）

```
[oracle@node1 ~]$ $ORA_CRS_HOME/bin/crs_profile -delete ggatevip
```

图 3-26

### 3.2.2.7 使用 ASM 存放 trail 文件注意事项

GoldenGate 支持从 ASM 中捕获数据，但如果使用 ASM 存储重做日志文件和/或者归档日志文件，有一些注意事项：

- 3、 GoldenGate 需要通过监听连接到 ASM 实例读取日志，所以需要配置监听文件
- 4、 为了连接到监听，需要添加连接描述符到 tnsnames.ora 文件
- 5、 GoldenGate 的 Extract 参数文件必须包括下面的内容：

```
TRANLOGOPTIONS ASUSER <user>@<asm>, ASMPASSWORD <pwd>, ENCRYPTKEY <key>
```

示例 3-5

### 3.2.3 配置源端数据库

由于 GoldenGate 的原理是根据 Oracle 的日志进行抽取复制的，所以 Oracle 的 redo 对于 GoldenGate 非常重要。因为 redo 会频繁的切换，日志组是有限的，并且是循环使用的。

1. 为了让 GoldenGate 不丢失任何的日志，需要打开数据库的归档：

在数据库 mount 阶段，使用命令：

```
alter database archivelog
```

示例 3-6

打开数据库的归档，如果使用了 Flash Recovery Area，系统会自动归档到 db\_recovery\_file\_dest。

当然也可以使用以下命令手动指定归档路径：

```
alter system set log_archive_dest_1=' location=/temp/arch' ;
```

示例 3-7

然后用命令。

```
SQL> archive log list
```

示例 3-8

可以查看数据库是否归档：

```
SQL> archive log list
Database log mode           Archive Mode
Automatic archival         Enabled
Archive destination        +RAC_DISK/racdb/
Oldest online log sequence 24
Next log sequence to archive 25
Current log sequence        25
SQL>
```

图 3-27

2. GoldenGate 需要需要数据库开启 supplemental log:

GoldenGate 需要准确的知道源端的数据修改了那些列，需要更为详细的日志，所以需要数据库开启 supplemental log。

如果没有开启，则根据下面的操作开启：

```
SQL> select supplemental_log_data_min from v$database;

SUPPLEME
-----
NO

SQL> alter database add supplemental log data;

Database altered.
```

图 3-28

3. 关闭 recyclebin:

这个选项是为了支持 GoldenGate DDL 复制的。关掉 recyclebin 的操作为：

```
SQL> alter system set "recyclebin"=off;

System altered.
```

图 3-29

4. 在目标端添加 checkpoint 表:

这是为了保证源和目标在传数据的时候不会重复或者少传，添加的步骤为在目标机器上编辑 GLOBALS 文件，添加一行，checkpoint ggs.checkpoint。然后 dblogin 数据库，添加 checkpoint 表:

```
GGSCI (linux4) 1> edit params ./GLOBALS

checkpoint ggs.checkpoint
~
```

图 3-30

```
GGSCI (linux4) 2> dblogin userid ggs,password ggs
Successfully logged into database.

GGSCI (linux4) 3> add checkpointtable ggs.checkpoint

Successfully created checkpoint table GGS.CHECKPOINT.

GGSCI (linux4) 4>
```

图 3-31

### 3.3 配置源端进程组

#### 3.3.1 配置 MGR 进程组

为了编辑和运行 GoldenGate，在源端和目标端都必须运行一个 MGR 进程组，它负责启动 GoldenGate 进程，以及启动动态进程，管理 trail 文件，以及错误信息报告等等。

创建动态进程组的步骤为：

进入安装 GoldenGate 的目录，运行 GGSCI 程序来打开 GoldenGate 命令接口。

- 在 GGSCI，输入 `edit params mgr` 命令编辑参数文件；
- 在 GGSCI，输入 `port <port_number>` 指定管理端口；
- 在 GGSCI，还可以配置一些其他的参数，然后输入保存文件。

```
[oracle@node1 10.0]$ ggsci

Oracle GoldenGate Command Interpreter for Oracle
Version 10.4.0.19 Build 002
Linux, x86, 32bit (optimized), Oracle 10 on Sep 25 2009 12:49:31

Copyright (C) 1995, 2009, Oracle and/or its affiliates. All rights reserved.

GGSCI (node1) 1> edit params mgr
```

图 3-32

##### 3.3.1.1 动态端口参数设置

可以制定最大 256 个可用端口的动态列表，当指定的端口不可用时，管理进程将会从列表中选择一个可用的端口，源端和目标端的 Collector, Replicat, GGSCI 进程将会使用这些端口。

指定动态端口的命令：

```
DYNAMICPORTLIST {<port> | <port>-<port>} [, ...]
```

示例 3-9

参数 `DYNAMICPORTREASSIGNDELAY <seconds>` 可以指定多长时间重新制定一次。

##### 3.3.1.2 自动启动参数设置

当管理进程启动的时候使用 autostart 参数设置 Extract 和 Replicat 进程自动启动。也可以使用 autostart 设置当 Extract 和 Replicat 进程异常终止时来自动启动。

这个功能在网络临时中断、进程异常终止、数据库死锁等情况下比较有用，它会节省大量的工作量。

下面为这个命令的格式：

```
AUTORESTART {ER | EXTRACT | REPLICAT} {group name | wildcard}  
[, RETRIES <max retries>]  
[, WAITMINUTES <wait minutes>]  
[, RESETMINUTES <reset minutes>]
```

示例 3-10

```
GGSCI (node1) 2> view params mgr  
  
port 7809  
dynamicportlist 7800-8000  
autorestart extract *,waitminutes 2,resetminutes 5  
  
GGSCI (node1) 3>
```

图 3-33

这里 autorestart 行表示每 2 分钟尝试重新启动所有的 Extract 进程，一共尝试 2 次（默认 entries 大小为 2，如果需要设置尝试的次数，可以设置 entries 参数），以后 5 分钟清零，然后再按照没 2 分钟尝试一共清理 2 次。

启动 MGR 进程：

```
GGSCI (node1) 3> start mgr  
  
Manager started.  
  
GGSCI (node1) 4>
```

图 3-34

### 3.3.1.3 trail 文件维护

当 GoldenGate 应用完成 trail 文件以后，可以设置定期的自动清理 trail 文件，释放磁盘空间。在参数文件中使用参数 PURGEOLDEXTRACTS，管理进程就会自动定期的清除 trail 文件。

注意：默认情况下 GoldenGate 并不会删除过期的 trail 文件，这样的话，trail 占用的磁盘空间会随着时间的积累越来越大，如不及时清理，会导致对应的文件系统占满，相应的进程会 abend，甚至还会引起操作系统的一些问题。

## 3.3.2 配置 Extract 进程组

抽取进程组在源端运行，负责抓取需要传输的数据。

下例为创建一个名为 eora 的 Extract 进程组：

1. edit params eora, 然后添加需要的参数内容:

```
GGSCI (node1) 62> view params eora

extract eora
dynamicresolution
userid ggs,password ggs
exttrail /u01/ggs/10.0/dirdat/et
table scott.*;
```

图 3-35

2. 在源端用 add extract 命令创建一个主 Extract 组  
(因为测试中已经添加, 所以这里用的 alter 示意)

```
GGSCI (node1) 10> alter extract eora,tranlog,begin now
EXTRACT altered.
```

图 3-36

3. 用 add exttrail 命令创建本地 trail 文件:

主 Extract 组负责写这部分文件, Pump 负责把 trail 文件从源端投递到目标端(因为测试中已经添加, 所以这里用的 alter 示意)

```
GGSCI (node1) 11> alter exttrail /u01/ggs/10.0/dirdat/et,extract eora
EXTTRAIL altered.
```

图 3-37

4. 使用 GGSCI 命令管理 Extract:

在 GGSCI 命令中, 可以使用 add, alter, cleanup, delete, info, kill 命令管理 Extract 进程。

```
GGSCI (node1) 19> start extract eora

Sending START request to MANAGER ...
EXTRACT EORA starting
```

图 3-38

### 3.3.3 配置 Pump 进程组

在源端一个数据 Pump 进程是 secondary Extract 进程组, 前面介绍过, 如果没有 Pump 进程, 则需要配置 Extract 进程把抽取来的数据投递到目标端, 但是配置 Pump 进程的好处是可以保证当网络不稳定的时候, 能稳定没有差错的把数据投递到目标端。

创建一个名为 pump\_so 的 Pump 进程组:

1. 使用 edit params pump\_so, 添加内容保存即可:

```
GGSCI (node1) 63> view params pump_so

extract pump_so
dynamicresolution
passthru
rmthost 192.168.0.104,mgrport 7809,compress
rmtrail /u01/ggs/10.0/dirdat/pt
table scott.*;
--table ggs.test;
```

图 3-39

2. 用 add extract 指定本地 trail 文件  
(因为测试中已经添加, 所以这里用的 alter 示意)

```
GGSCI (node1) 18> alter extract pump_so,exttrailsource /u01/ggs/10/0/dirdat/et
EXTRACT altered.
```

图 3-40

3. 用 add rmttrail 指定远程 trail 文件:  
(因为测试中已经添加, 所以这里用的 alter 示意)

```
GGSCI (node1) 20> alter rmttrail /u01/ggs/10.0/dirdat/pt,extract pump_so
RMTTRAIL altered.
```

图 3-41

### 3.4 配置目标端进程

#### 3.4.1 配置目标端 MGR 进程组

使用 edit params mgr 编辑 mgr 参数文件然后保存即可:

```
GGSCI (linux4) 1> view params mgr

port 7809
dynamicportlist 7800-8000
autostart er *
autorestart extract *,waitminutes 2,retries 5
lagreporhours 1
laginfominutes 3
lagcriticalminutes 5
purgeoldextracts /u01/ggs/10.0/rt*,usecheckpoints,minkeepdays 3
```

图 3-42

这里使用了延迟的预警机制: MGR 进程每隔 1 个小时检查 Extract 的延迟情况, 如果超过了 3 分钟就把延迟作为信息记录到错误日志中, 如果延迟超过了 5 分钟, 则把它作为警告写到错误日志中。

### 3.4.2 配置目标端 Replicat 进程组

Replicat 进程在目标端运行，他负责读源端投递过来的文件数据，然后把变化应用到目标端，下面我们来配置 Replicat：

1. 我们使用 `edit params` 命令创建一个名为 `repl` 的参数文件：

```
GGSCI (linux4) 53> view params repl

replicat repl
userid ggs,password ggs
assumtargetdefs
reperror default,discard
discardfile ./dirrpt/repl.dsc,append,megabytes 50
dynamicresolution
--map ggs.test, target ggs.test;
map scott.*, target scott.*;
```

图 3-43

2. 在目标端用 `add replicat` 添加一个 `repl` 进程
3. （因为测试中已添加，所以这里用 `alter`）

```
GGSCI (linux4) 7> alter replicat repl,extrail /u01/ggs/10.0/dirdat/pt
REPLICAT altered.
```

图 3-44

### 3.5 DML 测试

测试按上面的参数配置，是否可以实现数据同步。

1. 我们首先启动源端的所有需要的进程：

```
GGSCI (node1) 26> info all
```

Program	Status	Group	Lag	Time Since Chkpt
MANAGER	RUNNING			
EXTRACT	RUNNING	EORA	00:00:00	00:00:08
EXTRACT	RUNNING	PUMP_S0	00:00:00	00:00:08

图 3-45

2. 同样启动目标端的所有进程：

```
GGSCI (linux4) 10> info all
```

Program	Status	Group	Lag	Time Since Chkpt
MANAGER	RUNNING			
REPLICAT	RUNNING	REP1	00:00:00	00:00:01

图 3-46

3. 在源端数据库，插入 `scott.dept` 一行数据：

```
SQL> select * from dept;

DEPTNO DNAME          LOC
-----
10 ACCOUNTING      NEW YORK
20 RESEARCH        DALLAS
30 SALES           CHICAGO
40 OPERATIONS      BOSTON

SQL> insert into dept values(50,'RENXIAODONG','NIHAO');

1 row created.

SQL> commit;

Commit complete.

SQL> select * from dept;

DEPTNO DNAME          LOC
-----
50 RENXIAODONG    NIHAO
10 ACCOUNTING      NEW YORK
20 RESEARCH        DALLAS
30 SALES           CHICAGO
40 OPERATIONS      BOSTON
```

图 3-47

4. 在目标端查看:

```
SQL> select * from dept;

DEPTNO DNAME          LOC
-----
10 ACCOUNTING      NEW YORK
20 RESEARCH        DALLAS
30 SALES           CHICAGO
40 OPERATIONS      BOSTON

SQL> select * from dept;

DEPTNO DNAME          LOC
-----
10 ACCOUNTING      NEW YORK
20 RESEARCH        DALLAS
30 SALES           CHICAGO
40 OPERATIONS      BOSTON
50 RENXIAODONG    NIHAO
```

图 3-48

可见我们上面的配置是成功的。