

GoldenGate 安装配置手册

目录

1. GoldenGate 介绍.....	4
1.1. GoldenGate 技术架构.....	4
1.2. GoldenGate 拓扑结构.....	9
1.3. GoldenGate 复制的可靠性.....	9
1.4. GGSCI.....	10
2. GoldenGate 安装.....	12
2.1. 安装.....	12
2.2. 环境变量.....	13
3. OGG 数据库配置说明.....	14
3.1. 源数据库配置.....	14
3.2. GoldenGate 管理用户.....	17
3.3. GoldenGate 子目录配置.....	18
3.4. 配置管理进程.....	19
3.5. 配置进程检查点 (checkpoint).....	21
3.6. 提取进程参数说明.....	23
3.7. 应用进程参数说明.....	25
3.8. GLOBALS 参数配置.....	26
3.9. 参考文档.....	26
4. GoldenGate 配置示例.....	28
4.1. 报表系统配置实例.....	28
4.2. 实时数据分发配置.....	30
4.3. 实时数据仓库配置.....	31
4.4. 实时备用数据库.....	32
4.5. 双活高可用配置.....	33
5. 目标数据库初始化.....	35
6. GoldenGate 配置示例.....	36
6.1. DML 数据同步简单配置.....	36
6.2. 使用 DATA PUMP 方式进行同步.....	40
6.3. DDL 同步.....	44
7. 异构数据库.....	51

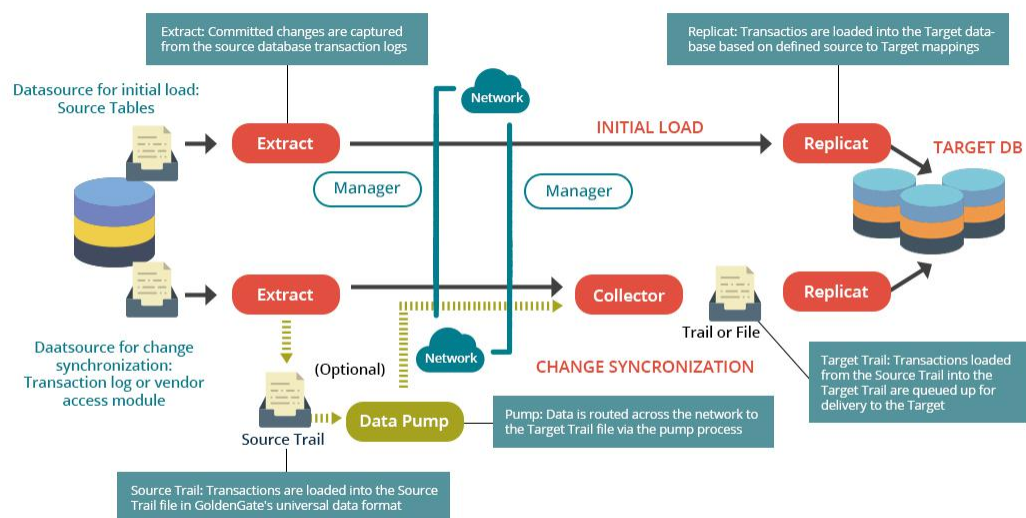
7.1. MySQL.....	51
8. 常见命令说明.....	52
8.1. 登陆 GoldenGate 控制台.....	52
8.2. 登陆数据库.....	52
8.3. 查看历史执行命令.....	53
8.4. 存储常用命令.....	53
8.5. 通过 SHELL 执行脚本.....	54
8.6. 配置文件中变量.....	54
8.7. 通配符说明.....	55
8.8. 查看进程状态.....	55
8.9. 查看队列信息.....	57
8.10. 队列加密.....	57
8.11. 用户口令加密.....	57
8.12. 查看参数文件.....	58
8.13. 编辑参数文件.....	59
8.14. 启动进程.....	59
8.15. 停止进程.....	59
8.16. 强制终止.....	59
8.17. 查看进程统计信息.....	60
8.18. 查看进程报告.....	60
9. 常用文件目录说明.....	62
9.1. ggserr.log.....	62
9.2. dirrpt 目录.....	62
9.3. dirprm 目录.....	63
9.4. dirdat 目录.....	63
9.5. dirpcs 目录.....	63
9.6. dirchk 目录.....	63
9.7. dirtmp 目录.....	63
10. 常用错误处理.....	64
10.1. OGG-00037 DISCARDFILE file ./dirrpt/repsrcdb.dis already exists.....	64

1. GoldenGate 介绍

1.1. GoldenGate 技术架构

Oracle GoldenGate 软件是一种基于 log(日志)的结构化数据复制软件，它通过解析源数据库在线 log 或归档 log 获得数据的增、删、改变化，再将这些变化应用到目标数据库，实现源数据库与目标数据库同步、双活。Oracle GoldenGate 软件可以在异构的 IT 基础结构(包括几乎所有常用操作系统平台和数据库平台)之间实现大量数据亚秒级的实时复制，

其复制过程简图如下：



由此可见，Oracle GoldenGate 是一种基于软件的数据复制方式，它从数据库的 log 解析数据的变化。Oracle GoldenGate 将数据变化转化为自己的格式，直接通过 TCP/IP 网络传输，无需依赖于数据库自身的传递方式，而且可以通过高达 9:1 的压缩率对数据进行压缩，可以大大降低带宽需求。在目标端，Oracle GoldenGate 可以通过交易重组，分批加载等技术手段大大加快数据投递的速度和效率，降低目标系统的资源占用，可以在亚秒级实现大量数据的复制，并且目标端数据库是活动的。

以下是 OracleGoldenGate 的技术架构：

1.1.1. Manager 进程

Manager 进程是 GoldenGate 的控制进程，运行在源端和目标端上。它主要作用有以下几个方面：启动、监控、重启 GoldenGate 的其他进程，报告错误及事件，分配数据存储

空间，发布阈值报告等。在目标端和源端有且只有一个 **manager** 进程，其运行状态为 **running**。在 **windows** 系统上，**manager** 进程作为一个服务来启动，在 **Linux/Unix** 系统上则是一个系统进程。

GGSCI (rac11g1) 3> **INFO MANAGER**

Manager is running (IP port rac11g1.7801).

1.1.2. Extract 进程

Extract 运行在数据库源端，负责从源端数据表或者日志中捕获数据。在早期的 **GoldenGate** 版本中，被称为 **Collect** 进程。

Extract 的作用可以按照时间来划分：

- ❖ 初始时间装载阶段：在初始数据装载阶段，**extract** 进程直接从源端的数据表中抽取数据。
- ❖ 同步变化捕获阶段：初始数据同步完成以后，**extract** 进程负责捕获源端数据的变化 (**DML** 和 **DDL**)。

GoldenGate 并不是对所有的数据库都支持 **DDL** 操作。

Extract 进程会捕获所有已配置的需要同步的对象变化，但只会将已提交的事务发送到远程的 **trail** 文件用于同步。当事务提交时，所有和该事务相关的日志记录被以事务为单元顺序的记录到 **trail** 文件中。**extract** 进程利用其内在的 **checkpoint** 机制，周期性的记录其读写的位置，这种机制是为了保证 **extract** 进程终止或操作系统宕机，重新启动 **extract** 后，**GoldenGate** 可以恢复到之前的状态，从上一个断点继续往下运行。通过上面的两个机制，就可以保证数据的完整性了。

多个 **extract** 进程可以同时对不同对象进行操作。例如，可以在一个 **extract** 进程抽取并向目标端发生事务数据的同时，利用另一个 **extract** 进程抽取的数据做报表。或者，两个 **extract** 进程可以利用两个 **trail** 文件，同时抽取并并行传输给两个 **replicat** 进程以减少数据同步的延时。

在进行初始化转载，或者批量同步数据时，**GoldenGate** 会生成 **extract** 文件来存储数据而不是 **trail** 文件。默认情况下，只会生成一个 **extract** 文件，但如果出于操作系统对单个文件大小限制或者其他因素的考虑，也可以通过配置生成多个 **extract** 文件。**extract** 文件不记录检查点。

extract 进程的状态包括 **Stopped**(正常停止)，**Starting**(正在启动)，**Running**(正在运行)，**Abended**(**Abnomal End** 的缩写，标示异常结束)。

1.1.3. Pump 进程

pump 进程运行在数据库源端，其作用是将源端产生的本地 **trail** 文件，把 **trail** 以数据块的形式通过 **TCP/IP** 协议发送到目标端，这通常也是推荐的方式。**pump** 进程本质是 **extract** 进程的一种特殊形式，如果不使用 **trail** 文件，那么 **extract** 进程在抽取完数据以后，直接投递到目标端，生成远程 **trail** 文件。

与 **Pump** 进程对应的叫 **Server Collector** 进程，这个进程不需要引起我的关注，因为在实际操作过程中，无需我们对其进行任何配置，所以对我们来说它是透明的。它运行在目标端，其任务就是把 **Extract/Pump** 投递过来的数据重新组装成远程 **trail** 文件。

注意：无论是否使用 **pump** 进程，在目标端都会生成 **trail** 文件。

pump 进程可以在线或者批量配置，他可以进行数据过滤，映射和转换，同时他还可以配置为“直通模式”，这样数据被传输到目标端时就可以直接生成所需的格式，无需另外操作。直通模式提高了 **data pump** 的效率，因为生成后的对象不需要继续进行检索。

在大多数情况下，**oracle** 都建议采用 **data pump**，原因如下：

- ❖ **为目标端或网络问题提供保障：**如果只在目标端配置 **trail** 文件，由于源端会将 **extract** 进程抽取的内容不断的保存在内存中，并及时的发送到目标端。当网络或者目标端出现故障时，由于 **extract** 进程无法及时的将数据发送到目标，**extract** 进程将耗尽内存然后异常终止。如果在源端配置了 **data pump** 进程，捕获的数据会被转移到硬盘上，预防了异常终止的情况。当故障修复，源端和目标端恢复连通性时，**data pump** 进程发送源端的 **trail** 文件到目标端。
- ❖ **可以支持复杂的数据过滤或者转换：**当使用数据过滤或者转换时，可以先配置一个 **data pump** 进程在目标端或者源端进行第一步的转换，利用另一个 **data pump** 进程或者 **replicat** 组进行第二部的转换。
- ❖ **有效的规划存储资源：**当从多个数据源同步到一个数据中心时，采用 **data pump** 的方式，可以在源端保存抽取的数据，目标端保存 **trail** 文件，从而节约存储空间。
- ❖ **解决单数据源向多个目标端传输数据的单点故障：**当从一个数据源发送数据到多个目标端时，可以为每个目标端分别配置不同的 **data pump** 进程。这样如果某个目标端失效或者网络故障时，其他的目标端不会受到影响可以继续同步数据。

1.1.4. Replicat 进程

Replicat 进程,通常我们也把它叫做应用进程。运行在目标端,是数据传递的最后一站,负责读取目标端 **trail** 文件中的内容,并将其解析为 **DML** 或 **DDL** 语句,然后应用到目标数据库中。

和 **Extract** 进程一样, **Replicat** 也有其内部的 **checkpoint** 机制,保证重启后可以从上次记录的位置开始恢复而无数据损失的风险。

Replicat 进程的状态包括 **Stopped**(正常停止),**Starting**(正在启动),**Running**(正在运行),**Abended**(**Abnomal End** 的缩写,标示异常结束)。

可以通过 **DEFERAPPLYINTERVAL** 参数控制应用进程延迟应用变化。

1.1.5. Trail 文件

为了更有效、更安全的把数据库事务信息从源端投递到目标端。**GoldenGate** 引进 **trail** 文件的概念。前面提到 **extract** 抽取完数据以后 **GoldenGate** 会将抽取的事务信息转化为一种 **GoldenGate** 专有格式的文件。然后 **pump** 负责把源端的 **trail** 文件投递到目标端,所以源、目标两端都会存在这种文件。**trail** 文件存在的目的旨在防止单点故障,将事务信息持久化,并且使用 **checkpoint** 机制来记录其读写位置,如果故障发生,则数据可以根据 **checkpoint** 记录的位置来重传。当然,也可以通过 **extract** 通过 **TCP/IP** 协议直接发送到目标端,生成远程 **trail** 文件。但这种方式可能造成数据丢失。

存放在本地系统的称为 **extract trail** (或 **local trail**)。存放在远程系统的被称为 **remote trail**。只有 **Extract** 进程才能写 **trail**。每个 **Extract** 进程都有对应的 **trail**。

Trail 文件默认为 **10MB**,通过 **ADD RMTTRAIL** 或 **ADD EXTTRAIL** 进行增加操作。以两个字符开始加上 **000000~999999** 的 **6** 位数字作为文件名。如 **c:/directory/tr000001**。默认情况下存储在 **GoldenGate** 的 **dirdat** 子目录中。可以为不同应用或者对象创建不同的 **trail** 文件。同一时刻,只会有一个 **extract** 进程处理一个 **trail** 文件。

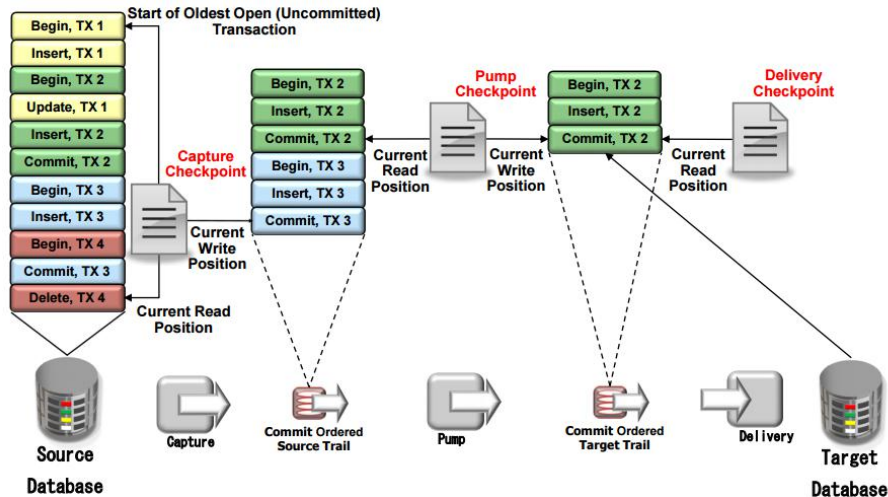
10.0 版本以后的 **GoldenGate**,会在 **trail** 文件头部存储包含 **trail** 文件信息的记录,而 **10.0** 之前的版本不会存储该信息。每个 **trail** 文件中的数据记录包含了数据头区域和数据区域。在数据头区域中包含事务信息,数据区域包含实际抽取的数据。

为了减小系统的 **I/O** 负载,抽取的数据通过大字节块的方式存储到 **trail** 文件中。同时为了提高兼容性,存储在 **trail** 文件中的数据以通用数据模式(一种可以在异构数据库之间进行快速而准确转换的模式)存储。当然,根据不同应用的需求,数据也可以存储为不同的模式。

默认情况下, **extract** 进程以追加的方式写入 **trail** 文件。当 **extract** 进程异常终止时, **trail** 文件会被标记为需要恢复。当 **extract** 重新启动时会追加 **checkpoint** 之后的数据追加到该 **trail** 文件中。在 **GoldenGate 10.0** 之前的版本, **extract** 进程采用的是覆盖模式。即当 **extract** 进程异常终止,则会将至上次完整写入的事务数据之后的数据覆盖现有 **trail** 文件中的内容。

1.1.6. checkpoint

checkpoint 用于抽取或复制失败后(如系统宕机、网络故障灯), 抽取、复制进程重新定位抽取或者复制的起点。在高级的同步配置中, 可以通过配置 checkpoint 多个 extract 或者 replicat 进程读取同个 trail 文件集。



extract 进程在数据源和 trail 文件中都会标识 checkpoint, Replicat 只会在 trail 文件中标识 checkpoint。

在批处理模式中, extract 和 replicat 进程都不会记录 checkpoint。如果批处理失败, 则整改批处理会重新进行。

checkpoint 信息会默认存储在 GoldenGate 的子目录 dirchk 中。在目标端除了 checkpoint 文件外, 我们也可以通过配置通过额外 checkpoint table 来存储 replicat 的 checkpoint 信息。

1.1.7. Group

我们可以通过为不同的 extract 和 replicat 进程进行分组来去区分不同进程之间的作用。例如, 当需要并行的复制不同的数据集时, 我们则可以创建两个或者多个复制进程。

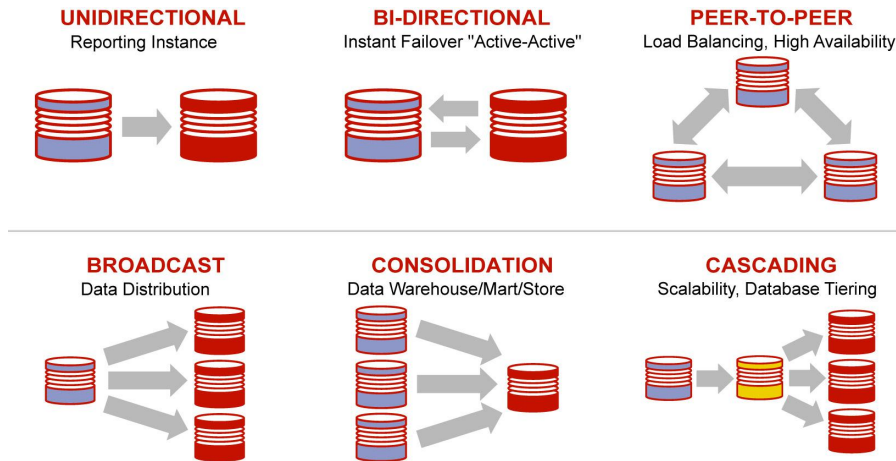
进程组中包含进程, 进程文件, checkpoint 文件和其他与进程相关的文件。对于 replicat 进程来说, 如果配置了 checkpoint table, 则不同组的都会包含 checkpoint table。

1.1.8. Commit Sequence Number

前文已经多次提到, Goldengate 是以事务为单位来保证数据的完整性的, 那么 GoldenGate 又是怎么识别事务的呢? 这里用到的是 Commit Sequence Number(CSN)。CSN 存储在事务日志中和 trail 文件中, 用于数据的抽取和复制。CSN 作为事务开始的标志被记录在 trail 文件中, 可以通过 @GETENV 字段转换函数或者 logdump 工具来查看。

1.2. GoldenGate 拓扑结构

Oracle GoldenGate 支持多种拓扑结构，为用户数据复制方案的灵活性、先进性奠定基础，如图所示：



- 单向数据复制：由一个源数据库复制到一个目的数据库，一般用于高可用性和容灾，为生产机保持一个活动的备用数据库，从而在发生灾难的时候迅速切换，减少数据丢失和停机时间；
- 双向数据复制：利用 Oracle GoldenGate 可以实现两个数据库之间数据的双向复制，任何一方的数据变化都会被传递到另一端，可以利用此模式开展双业务中心；
- 广播复制：由一个数据库向多个数据库复制，利用 Oracle GoldenGate 的数据过滤功能可以实现数据的有选择分发；
- 集中复制：由多个数据库向一个数据库复制，可以将分布的、跨平台或异构的多个数据库集中到一个数据库。此种模式广泛应用于 N+1 模式的容灾，通过将多个系统数据库集中到一起，可以充分利用备份中心的设施，大幅减少投资；另外也用于跨平台多系统的数据集成，为这些提供系统提供一个统一视图便于查询和统计数据。
- 多层复制：由 A 数据库向 B 复制，同时又由 B 向 C 复制，可以在以上几种模式基础上无限制扩展。

1.3. GoldenGate 复制的可靠性

Oracle GoldenGate 可以提供可靠的数据复制，主要体现在下面三点：

- 保证事务一致性

Oracle GoldenGate 在灾备数据库应用复制数据库交易的顺序与在生产中心数据库上的顺序相同，并且按照相同的事务环境提交，确保在目标系统上数据的完整性和读一致性，为实时查询和事务处理创造了条件。

- 检查点机制保障数据无丢失

Oracle GoldenGate 的抽取和复制进程使用检查点机制记录完成复制的位置。对于抽取进程，其检查点记录当前已经抽取 log 的位置和写队列文件的位置；对于投递进程，其检查点记录当前读取队列文件的位置。检查点机制可以保证在系统、网络或 Oracle GoldenGate 进程故障重启后数据无丢失。

➤ 可靠的数据传输机制

Oracle GoldenGate 用应答机制传输交易数据，只有在得到确认消息后才认为数据传输完成，否则将自动重新传输数据，从而保证了抽取出的所有数据都能发送到备用端。数据传输过程中支持 128 位加密和数据压缩功能；

1.4. GGSCI

GGSCI 是 GoldenGate Software Command Interface 的缩写，它提供了十分丰富的命令来对 GoldenGate 进行各种操作，如创建、修改、监控 GoldenGate 进程等等。

GGSCI (rac11g1) 8> [HELP](#)

GGSCI Command Summary:

Object:	Command:
SUBDIRS	CREATE
ER	INFO, KILL, LAG, SEND, STATUS, START, STATS, STOP
EXTRACT	ADD, ALTER, CLEANUP, DELETE, INFO, KILL, LAG, REGISTER, SEND, START, STATS, STATUS, STOP UNREGISTER
EXTTRAIL	ADD, ALTER, DELETE, INFO
GGSEVT	VIEW
MANAGER	INFO, SEND, START, STOP, STATUS
MARKER	INFO
PARAMS	EDIT, VIEW
REPLICAT	ADD, ALTER, CLEANUP, DELETE, INFO, KILL, LAG, SEND, START, STATS, STATUS, STOP
REPORT	VIEW
RMTTRAIL	ADD, ALTER, DELETE, INFO
TRACETABLE	ADD, DELETE, INFO
TRANDATA	ADD, DELETE, INFO

```
SCHEMATRANDATA  ADD, DELETE, INFO
```

```
CHECKPOINTTABLE  ADD, DELETE, CLEANUP, INFO
```

Commands without an object:

```
(Database)      DBLOGIN, LIST TABLES, ENCRYPT PASSWORD, FLUSH SEQUENCE
```

```
                MININGDBLOGIN
```

```
(DDL)           DUMPDDL
```

```
(Miscellaneous) FC, HELP, HISTORY, INFO ALL, OBEY, SET EDITOR, SHELL,
```

```
                SHOW, VERSIONS, ! (note: you must type the word
```

```
                COMMAND after the ! to display the ! help topic.)
```

```
                i.e.: GGSCI (sys1)> help ! command
```

For help on a specific command, type **HELP <command> <object>**

设置默认编辑器为 VIM。

```
GGSCI (rac11g1) 59> set editor vim
```

2. GoldenGate 安装

2.1. 安装

```
[root@gxoradb1 oracle]# mkdir GoldenGate
```

```
[root@gxoradb1 oracle]# chown oracle:oinstall ./GoldenGate/
```

```
[root@gxoradb1 oracle]# su - oracle
```

```
[oracle@gxoradb1 ~]$ ls
```

```
Oracle_GoldenGate_Monitor_Linux_x64_11.1.1.1.0.zip  
ogg112101_fbo_ggs_Linux_x64_ora11g_64bit.zip  oradiag_oracle
```

```
[oracle@gxoradb1 ~]$ unzip ogg112101_fbo_ggs_Linux_x64_ora11g_64bit.zip -d  
/oracle/GoldenGate/
```

```
Archive:  ogg112101_fbo_ggs_Linux_x64_ora11g_64bit.zip  
  inflating: /oracle/GoldenGate/fbo_ggs_Linux_x64_ora11g_64bit.tar  
  inflating: /oracle/GoldenGate/OGG_WinUnix_Rel_Notes_11.2.1.0.1.pdf  
  inflating: /oracle/GoldenGate/Oracle GoldenGate 11.2.1.0.1 README.txt  
  inflating: /oracle/GoldenGate/Oracle GoldenGate 11.2.1.0.1 README.doc
```

```
[oracle@gxoradb1 ~]$ cd /oracle/GoldenGate/
```

```
[oracle@gxoradb1 GoldenGate]$ ls -lrt
```

```
total 223764  
-rw-rw-r-- 1 oracle oinstall 228556800 Apr 23  2012 fbo_ggs_Linux_x64_ora11g_64bit.tar  
-rwxrwxrwx 1 oracle oinstall    220546 May  2  2012 OGG_WinUnix_Rel_Notes_11.2.1.0.1.pdf  
-rwxrwxrwx 1 oracle oinstall    93696 May  2  2012 Oracle GoldenGate 11.2.1.0.1 README.doc  
-rwxrwxrwx 1 oracle oinstall    24390 May  2  2012 Oracle GoldenGate 11.2.1.0.1 README.txt
```

```
[oracle@gxoradb1 GoldenGate]$ tar -xvf fbo_ggs_Linux_x64_ora11g_64bit.tar
```

```
[oracle@gxoradb1 GoldenGate]$ ll gg*
```

```
-rw-r----- 1 oracle oinstall 1257280 Apr 23  2012 ggMessage.dat  
-rwxr-x--- 1 oracle oinstall 6393899 Apr 23  2012 ggcmd  
-rwxr-x--- 1 oracle oinstall 13398159 Apr 23  2012 ggsci
```

```
[oracle@gxoradb1 GoldenGate]$ ./ggsci
```

```
Oracle GoldenGate Command Interpreter for Oracle  
Version 11.2.1.0.1 OGGCORE_11.2.1.0.1_PLATFORMS_120423.0230_FBO  
Linux, x64, 64bit (optimized), Oracle 11g on Apr 23 2012 08:32:14  
  
Copyright (C) 1995, 2012, Oracle and/or its affiliates. All rights reserved.  
  
GGSCI (gxoradb1) 1> help
```

2.2. 环境变量

在解压路径下执行：

```
[root@GG_HOME] > # ldd ggsci
```

将列出所有需要的 **lib** 和当前缺少的。**GoldenGate** 在 **Linux** 和 **Unix** 下安装，需要安装 **ORACLE** 的 **lib** 环境以及 **\$ORACLE_HOME/lib** 下的几个包，所以必须安装在 **Oracle** 之后，而且确保在环境变量中加入：

```
export LD_LIBRARY_PATH=$ORACLE_HOME/lib
```

环境变量设置实例：

```
export GG_HOME=/oracle/goldengate  
export PATH=.:$GG_HOME:$ORACLE_HOME/bin:$GRID_HOME/bin:/usr/sbin:/sbin:$PATH  
export LD_LIBRARY_PATH=$ORACLE_HOME/lib  
  
alias gg='cd $GG_HOME; ./ggsci'
```

3. OGG 数据库配置说明

3.1. 源数据库配置

如果需要进行双向同步，目标库也需要执行下面操作。

3.1.1. 打开数据库归档模式

```
SQL> alter system set  
log_archive_dest_1='LOCATION=/oracle/app/oracle/oradata/source/archive/';
```

```
SQL> shutdown immediate;
```

```
SQL> startup mount;
```

```
SQL> alter database archivelog;
```

```
SQL> alter database open;
```

```
SQL> archive log list;
```

Database log mode	Archive Mode
Automatic archival	Enabled
Archive destination	/oracle/app/oracle/oradata/source/archive/
Oldest online log sequence	17
Next log sequence to archive	19
Current log sequence	19

3.1.2. 打开强制日志

实际生产应用中，最好同时打开 ORACLE 的强制日志模式，以防止源数据库因直接路径加载忽略 redo 生成而导致这部分数据无法同步。

```
SQL> alter database force logging;
```

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

3.1.3. 开启附加日志

如果开启 DDL 复制做冗余灾备份，最好直接在数据库级别打开补全日志。

```
SQL> alter database add supplemental log data;
```

```
SQL> alter database add supplemental log data (primary key,unique,foreign key)
columns;
```

```
SQL> SELECT supplemental_log_data_min, supplemental_log_data_pk,
supplemental_log_data_ui FROM v$database;
```

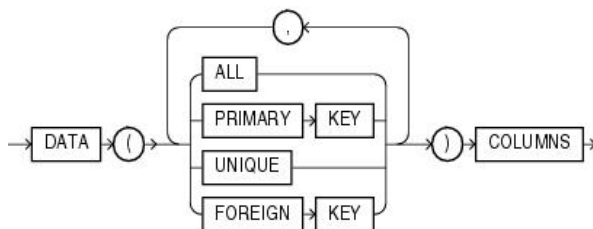
打开表级的补全日志,可以在 GoldenGate 中使用 `add trandata` 命令强制重做日志记录主键值,以保证在目标端能成功复制。

`SUPPLEMENTAL_LOG_DATA_MIN=YES` 和 `IMPLICIT` 都表示数据库启动了最小化的补充日志。`YES` 表示通过 `ALTER DATABASE ADD SUPPLEMENTAL LOG DATA` 语句启动的最小化补充日志;如果启动了对主键、唯一键、外键、`ALL` 的补充日志,默认情况下都会打开最小化的补充日志,`IMPLICIT` 表示的是通过启动对主键、唯一键、外键或者 `ALL` 的支持而打开的最小化补充日志。

```
GGSCI> dblogin userid ddw,password ddw --GoldenGate 中登录 OARCLE 数据库
```

```
GGSCI> add trandata ddw.<tablename>
```

```
GGSCI> add trandata coss3.per_test,nokey,cols(sampletime, objectid)
```



Oracle 补全日志可以在数据库级别设置,也可以在表级别设置。在数据库级别中,补全日志按补全的信息量,对应好几个级别:

- 1) 最小附加日志(Minimal supplemental logging): 是开启 logminer 的最低日志要求,提供了行链接(chained rows)和多种数据存储(比如聚簇表、索引组织表)的信息。在 Oracle 9.2 之后的版本中,默认都不开启。
- 2) 主键补全(Primary key supplemental logging): 在日志中补全所有主键列。如果表中无主键,则补全一个非空唯一索引列;如果非空唯一索引键也没,那么会补全除了 LOB 和 LONG 类型字段以外的所有列,这时就和下面的所有补全一样了。
- 3) 唯一键补全(Unique key supplemental logging): 当唯一键列或位图索引列被修改时,在日志中补全所有唯一键列或位图索引列。打开唯一键补全也会同时打开主键补全。注意这个级别是需要条件触发的。

- 4) 外键补全(Foreign Key supplemental logging): 当外键列被修改时, 将在日志中补全所有外键列。这个级别也是需要条件触发的。
- 5) 所有补全(All supplemental logging): 在日志中补全所有字段(排除 LOB 和 LONG 类型)。

数据库级别中的 5 个类型中, 除了最小附加日志级别, 都可以在表级进行设置。除此之外, 表级还可以明确指定需要补全的列。

Oracle 表级补全日志需要在最小补全日志打开的情况下才起作用, 即若一个数据库没有开最小补全日志或之前 `drop supplemental log data` 操作则即便指定了表级补全日志, 实际在重做日志输出的过程中描述的记录仍只记录 `rowid` 和相关列值。而要关闭最小补全日志, 也必须首先关闭数据库级别的其他补全级别后, 才能关闭。

注意:

如果对非常繁忙的数据库启动补充日志可能会带来性能问题和出现 **Hang** 住的情况。建议在数据库 **MOUNT** 状态下进行该操作。

Alter Database Add Supplemental Log Data Hangs (文档 ID 406498.1)

3.1.4. 11G Integrated Capture Mode 设置

从 Oracle GoldenGate11g 版本开始, oracle 引入了一种新的 capture mode, 称为 Integrated Capture Mode。传统的 capture mode 被称为 classic capture mode。

在 classic capture mode 中, GoldenGate extract 进程直接读取 oracle redo log, 捕获数据变化, 存为 GoldenGate 的 trail file 格式, 然后利用 pump 进程将这些 trail file 传输到目标数据库, 目标数据库上的 replicat 进程读取这些 trail file, 再利用 sql 将这些变化 apply 到目标数据库里。

而在新建的 integrated capture mode 中, GoldenGate extract 进程不再直接读取 oracle redo log, 而是通过与数据库 log mining server 整合来捕获数据变化: log mining server 负责以 LCR 的格式从数据库日志中捕获数据变化, 然后 extract 进程再将这些抓取的数据存成 trail file 的格式。

与 classic capture mode 相比, 这种 integrated capture mode 的主要差别就是 extract 不再直接读取 oracle redo log, 而交由数据库内部的 log mining 工具来完成。由于 extract 进程是由操作系统来管理的, 身处数据库系统之外, integrated capture mode 的这种改变所带来的主要好处体现在兼容性方面: 支持更多的数据类型和存储类型, 以及由于与数据库更紧密的整合, 不再需要为 Oracle RAC, ASM 和 TDE 作更多额外的配置工作。

integrated capture mode 带来的另一个重要变化就是: 开始支持 local 和 downstream 两种配置选项。而在 classic capture mode 中, 我们只有 local 这一种选择, 也就是 extract 进程必须配置在源数据库服务器上。

使用集成模式，建议安装对应的补丁。

Oracle GoldenGate -- Oracle RDBMS Server Recommended Patches (文档 ID 1557031.1)

ALTER SYSTEM SET ENABLE_GOLDENGATE_REPLICATION=TRUE SCOPE=BOTH;

从数据库 11.2.0.4 版本开始,我们需要在 source 和 target 数据库上同时 enable 该参数,这是 oracle 用来作 license 控制的。该参数是动态参数，可以在线进行修改。

3.1.5. 关闭回收站

在 GoldenGate 10g 中如果使用 DDL 复制，需要关闭回收站。在 11g 中不需要该操作。

SQL> **ALTER SYSTEM SET RECYCLEBIN=OFF DEFERRED;**

3.2. GoldenGate 管理用户

3.2.1. 源库和目标库创建用户 GGADMIN

该操作需要在源库和目标库同时进行。

➤ 源库

```
SQL> CREATE TABLESPACE GOLDENGATE DATAFILE  
'/oracle/app/oracle/oradata/source/goldengate.dbf' SIZE 100M;
```

➤ 目标库

```
SQL> CREATE TABLESPACE GOLDENGATE DATAFILE  
'/oracle/app/oracle/oradata/target/goldengate.dbf' SIZE 100M;
```

```
CREATE USER GGADMIN IDENTIFIED BY GGADMIN DEFAULT TABLESPACE  
GOLDENGATE TEMPORARY TABLESPACE TEMP;  
GRANT CONNECT, RESOURCE, ALTER SESSION TO GGADMIN;  
GRANT SELECT ANY DICTIONARY, SELECT ANY TABLE TO GGADMIN;  
GRANT ALTER ANY TABLE TO GGADMIN;  
GRANT EXECUTE ON UTL_FILE TO GGADMIN;  
GRANT FLASHBACK ANY TABLE TO GGADMIN;  
GRANT EXECUTE ON DBMS_FLASHBACK TO GGADMIN;
```

3.2.2. 源库 GGADMIN 增加权限(DDL)

```
GRANT CREATE SEQUENCE TO GGADMIN;
```

3.2.3. 目标库 GGADMIN 增加权限

如果需要双向同步，源库和目标库都需要进行配置。

```
GRANT INSERT ANY TABLE TO GGADMIN;
GRANT UPDATE ANY TABLE TO GGADMIN;
GRANT DELETE ANY TABLE TO GGADMIN;
GRANT UNLIMITED TABLESPACE TO GGADMIN;
GRANT CREATE ANY TABLE TO GGADMIN;      --DDL 需要该权限
GRANT CREATE ANY INDEX TO GGADMIN;       --DDL 需要该权限
GRANT DROP ANY TABLE TO GGADMIN;       --TRUNCATE 操作需要该权限
```

3.2.4. 11.2.0.4 后新增权限配置程序

```
EXEC DBMS_GOLDENGATE_AUTH.GRANT_ADMIN_PRIVILEGE( GRANTEE =>
'GGADMIN', PRIVILEGE_TYPE => 'CAPTURE', GRANT_SELECT_PRIVILEGES=>
TRUE, DO_GRANTS => TRUE);
```

3.3. GoldenGate 子目录配置

该操作需要在源库和目标库都进行。

GGSCI (gxradb1) 2> [CREATE SUBDIRS](#)

```
Creating subdirectories under current directory /oracle/GoldenGate

Parameter files           /oracle/GoldenGate/dirprm: already exists
Report files              /oracle/GoldenGate/dirrpt: created
Checkpoint files          /oracle/GoldenGate/dirchk: created
Process status files      /oracle/GoldenGate/dirpcs: created
SQL script files          /oracle/GoldenGate/dirsql: created
Database definitions files /oracle/GoldenGate/dirdef: created
extract data files        /oracle/GoldenGate/dirdat: created
Temporary files           /oracle/GoldenGate/dirtmp: created
Stdout files              /oracle/GoldenGate/dirout: created
```

这些目录不能调整，只能通过软连接的方式指向其他目录。

3.4. 配置管理进程

该操作需要在源库和目标库都进行。配置文件会自动存放在 **GoldenGate** 根目录下的 **dirprm** 目录中。

在参数文件中增加 **CHECKPARAMS** 参数。可以对参数文件内容进行校验。

3.4.1. 配置管理进程参数

GGSCI (rac11g1) 10> **EDIT PARAM MGR**

```
COMMENT 'SOURCE DATABASE MGR CONFIG'
PORT 7801
DYNAMICPORTLIST 7810-7850
AUTOSTART EXTRACT *
AUORESTART EXTRACT *, RETRIES 5, WAITMINUTES 3
STARTUPVALIDATIONDELAY 5
PURGEOLDEXTRACTS ./dirdat/*, usecheckpoints, minkeepdays 3
LAGREPORTHOURS 1
LAGINFOMINUTES 30
LAGCRITICALMINUTES 45
```

3.4.2. MANAGER 进程参数配置说明

➤ **PORT:**

PORT 7809

指定服务监听端口；这里以 **7839** 为例，默认端口为 **7809**；

➤ **DYNAMICPORTLIST:**

DYNAMICPORTLIST 7810-7850

动态端口：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，源端和目标段的 **Collector**、**Replicat**、**GGSCI** 进程通信也会使用这些端口；

➤ **AUTOSTART:**

AUTOSTART EXTRACT *

指定在管理进程启动时自动启动哪些进程：

➤ **AUTORESTART:**

```
AUTORESTART EXTRACT *, RETRIES 5, WAITMINUTES 3
```

自动重启参数设置：本处设置表示每 3 分钟尝试重新启动所有 EXTRACT 进程，共尝试 5 次；

➤ **STARTUPVALIDATIONDELAY 5**

启动时，延迟 5 秒进行校验；

➤ **PURGEOLDEXTRACTS:**

```
PURGEOLDEXTRACTS ./dirdat/*, usecheckpoints, minkeepdays 3
```

定期清理 trail 文件设置：本处设置表示对于超过 3 天的 trail 文件进行删除；

➤ **LAGREPORT、LAGINFO、LAGCRITICAL:**

```
LAGREPORTHOURS 1
```

```
LAGINFOMINUTES 30
```

```
LAGCRITICALMINUTES 45
```

定义数据延迟的预警机制：本处设置表示 MGR 进程每隔 1 小时检查 EXTRACT 的延迟情况，如果超过了 30 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 45 分钟，则把它作为警告写到错误日志中；

➤ **COMMENT:**

```
COMMENT 'SOURCE DATABASE MGR CONFIG'
```

注释行，也可以用--来代替；记录太长时，可以使用&进行换行操作。

3.4.3. 启动主控制进程

```
GGSCI (rac11g1) 13> START MGR
```

```
Manager started.
```

3.4.4. 通过操作系统命令启动管理进程

```
$ mgr paramfile <param file> [reportfile <report file>]
```

3.4.5. 停止主控制进程

```
GGSCI (rac11g1) 13> STOP MGR [!]
```

```
Manager process is required by other GGS processes.
```

```
Are you sure you want to stop it (y/n)? y
```

```
Sending STOP request to MANAGER ...
```

```
Request processed.
```

```
Manager stopped.
```

Where: ! stops Manager without user confirmation.

3.4.6. 查看进程状态

```
GGSCI (rac11g1) 15> INFO ALL
```

Program	Status	Group	Lag at Chkpt	Time Since Chkpt
MANAGER	RUNNING			

3.5. 配置进程检查点 (checkpoint)

检查点记录了进程读写的位置信息用以数据恢复，目的是为了防止进程因系统、网络崩溃而导致的数据丢失，对于 **GoldenGate** 保证数据同步过程中数据不丢失非常重要。

GoldenGate 的检查点由一个内部进程自动控制，与数据库检查点的概念类似。提取进程的检查点记录它在数据源中的读取位置和队列的写出位置，复制进程的检查点记录它读取队列的位置。每条提取或复制进程都有自己对应的检查点信息。当 **GoldenGate** 的进程重启时，由它所记录的检查点决定需要读取的队列位置。

GoldenGate 的检查点信息有两种存放方式：

3.5.1. 文件检查点

默认存放在 GGHOME\dirchk 下的文件中，一个进程对应一个文件。**提取进程只能使用这种模式**。不需要特殊配置。

3.5.2. 数据库心跳表 (只用于目标端)

存放在数据库指定的表中，需要进行如下配置：

```
GGSCI (rac11g2) 1> DBLOGIN USERID ggadmin@192.168.56.130:1521/TARGET,  
password GGADMIN
```

```
Successfully logged into database.
```

GGSCI (rac11g2) 2> [add checkpointtable ggadmin.checkpoint](#)

```
Successfully created checkpoint table ggadmin.checkpoint.
```

在新增复制应用进程时可以在添加时指定 `checkpointtable [<owner>.<table>]` 替代 `nodbcheckpoint`，使用数据库记录检查点信息。

➤ 指定心跳表

指定全局方式的心跳表

GGSCI (rac11g2) 22> [EDIT PARAMS ./GLOBALS](#)

```
CHECKPOINTTABLE GGADMIN.CHECKPOINT
```

可以在 `ADD REPLICAT` 命令中指定单独的心跳表，或者屏蔽全局心跳表

GGSCI (rac11g2) 26> [HELP ADD REPLICAT](#)

Syntax:

```
ADD REPLICAT <group name>
{
  , SPECIALRUN |
  , EXTFILE <full path name> |
  , EXTTRAIL <full path name>
}
[, BEGIN {NOW | yyyy-mm-dd:hh:mm[:ss[.cccccc]]} |
  , EXTSEQNO <seqno>, EXTRBA <rba>]
[, CHECKPOINTTABLE <owner, table> | NODBCHECKPOINT]
[, PARAMS <parameter file>]
[, REPORT <report file>]
[, DESC "<description>"]
```

ORACLE 官方文档中，比较推荐将复制应用进程的检查点信息存放到数据库表中进行管理，认为在某些情况下能促进数据恢复。并指出检查点信息量非常小，而且是进行记录更新而非记录插入，一个进程只对应一条记录，在它特殊的检查点处理机制下不会对数据库造成影响。个人猜想是当目标数据库崩溃还原后(特别是在不完全恢复的情况下)，检查点信息能同数据库一起还原，在数据上能利用数据库事务性与数据库保持一致，从而在数据库正常打开后能继续进行数据同步。但实际上并不必要，因为数据库故障的情况多种多样，就算检查点同步恢复后，也不能保证直接就能启动 GoldenGate 进程。

3.6. 提取进程

3.6.1. 配置提取进程

GGSCI (rac11g2) 28> [HELP ADD EXTRACT](#)

```
ADD EXTRACT <group name>
{
  SOURCEISTABLE |
    , TRANLOG [<bsds name>] |
    , INTEGRATED TRANLOG |
    , VAM |
    , EXTFILESOURCE <file name> |
    , EXTTRAILSOURCE <trail name> |
    , VAMTRAILSOURCE <VAM trail name>}
{
  BEGIN {NOW | yyyy-mm-dd[:hh:mi:[ss[.cccccc]]]} |
    , EXTSEQNO <seqno>, EXTRBA <relative byte address> |
    , EOF |
    , LSN <value> |
    , EXTRBA <relative byte address> |
    , EOF | LSN <value> |
    , PAGE <data page>, ROW <row> |
    , SEQNO <sequence number>
}
[ , THREADS <n>]
[ , PASSIVE]
[ , PARAMS <parameter file>]
[ , REPORT <report file>]
[ , DESC "<description>"]
```

3.6.2. 配置 TRAIL

trail 创建后，提取进程会自动创建文件。

```
ADD {RMTTRAIL | EXTTRAIL} <pathname>, EXTRACT <group name> [, MEGABYTES <n>]
```

- RMTTRAIL specifies a trail on a remote system.
- EXTTRAIL specifies a trail on the local system.

3.6.3. 提取进程参数说明

- **EXTRACT**
进程参数配置说明
- **SETENV**
配置系统环境变量
- **USERID/PASSWORD**
指定 OGG 连接数据库的用户名和密码;
- **COMMENT**
注释行, 也可以用--来代替;
- **TABLE**
定义需复制的表, 后面需以;结尾
- **TABLEEXCLUDE**
定义需要排除的表, 如果在 TABLE 参数中使用了通配符, 可以使用该参数指定排除掉得表。
- **GETUPDATEAFTERS | IGNOREUPDATEAFTERS**
是否在队列中写入后影像, 缺省复制
- **GETUPDATEBEFORES | IGNOREUPDATEBEFORES**
是否在队列中写入前影像, 缺省不复制
- **GETUPDATES | IGNOREUPDATES**
是否复制 UPDATE 操作, 缺省复制
- **GETDELETES | IGNOREDELETES**
是否复制 DELETE 操作, 缺省复制
- **GETINSERTS | IGNOREINSERTS**
是否复制 INSERT 操作, 缺省复制
- **GETTRUNCATES | IGNORETRUNCATES**
是否复制 TRUNCATE 操作, 缺省不复制;
- **RMTHOST**
指定目标系统及其 GoldenGate Manager 进程的端口号, 还用于定义是否使用压缩进行传输;
- **RMTTRAIL**
指定写入到目标端的哪个队列;
- **EXTTRAIL**
指定写入到本地的哪个队列;

➤ **SQLEXEC**

在 extract 进程运行时首先运行一个 SQL 语句;

➤ **PASSTHRU | NOPASSTHRU**

禁止 extract 进程与数据库交互, 适用于 Data Pump 传输进程;

➤ **REPORT**

定义自动定时报告;

➤ **STATOPTIONS**

定义每次使用 stat 时统计数字是否需要重置;

➤ **REPORTCOUNT**

报告已经处理的记录条数统计数字;

➤ **TRACE**

打开对于数据库日志的跟踪日志;

➤ **DISCARDFILE**

定义 discardfile 文件位置, 如果处理中记录出错会写入到此文件中;

➤ **DBOPTIONS**

指定对于某种特定数据库所需要的特殊参数;

➤ **TRANLOGOPTIONS**

指定在解析数据库日志时所需要的特殊参数, 例如:对于裸设备, 可能需要加入以下参数 rawdeviceoggset 0

➤ **WARNLONGTRANS**

指定对于超过一定时间的长交易可以在 gsserr.log 里面写入警告信息, 本处配置为每隔 3 分钟检查一次场交易, 对于超过 2 小时的进行警告;

3.7. 应用进程参数说明

➤ **REPLICAT**

进程参数配置说明

➤ **ASSUMETARGETDEFS**

假定两端数据结构一致使用此参数;

➤ **SOURCEDEFS**

假定两端数据结构不一致, 使用此参数指定源端的数据结构定义文件, 该文件需要由 GlodenGate 工具产生。

➤ **MAP**

用于指定源端与目标端表的映射关系;

➤ **MAPEXCLUDE**

用于使用在 MAP 中使用*匹配时排除掉指定的表；

➤ **REPERERROR**

定义出错以后进程的响应，一般可以定义为两种：

- **ABEND**，即一旦出现错误即停止复制，此为缺省配置；
- **DISCARD**，出现错误后继续复制，只是把错误的数据放到 **discard** 文件中。

➤ **DISCARDFILE**

定义 **discardfile** 文件位置，如果处理中记录出错会写入到此文件中；

```
DISCARDFILE ./dirrpt/rpee.dsc,PURGE
DISCARDROLLOVER AT 02:00
```

➤ **SQLEXEC**

在进程运行时首先运行一个 SQL 语句；

➤ **GROUPTRANSOPS**

将小交易合并成一个大的交易进行提交，减少提交次数，降低系统 IO 消耗。

➤ **MAXTRANSOPS**

将大交易拆分，每 XX 条记录提交一次。

3.8. GLOBALS 参数配置

GGSCI (rac11g1) 2> [EDIT PARAMS ./GLOBALS](#)

必须使用 ./ 指定目录。GLOBALS 文件不需存放在 **goldengate** 的根目录下。

3.9. 参考文档

How to Modify the Location for Subdirectories Created by CREATE SUBDIRS In GoldenGate? (Doc ID 1366597.1)

How to Setup Oracle GoldenGate - Oracle to Oracle Extract and Replicat (step by step Classic Extract & Replicat setup) and Tutorials for Heterogeneous Databases (Doc ID 1484793.1)

Oracle GoldenGate Best Practices: Instantiation from an Oracle Source Database (Doc ID 1276058.1)

Master Note - Oracle GoldenGate: Initial Load Techniques and References (Doc ID 1311707.1)

I Need Information About The Replicat Checkpoint Table (Doc ID 965698.1)

Master Note - OGG TCP Port Usage (文档 ID 1300117.1)

Master Note for Oracle GoldenGate Setup Sample Scripts (文档 ID 1433100.1)

Master Note for Oracle GoldenGate for Filtering and Transformation Data (文档 ID 1450495.1)

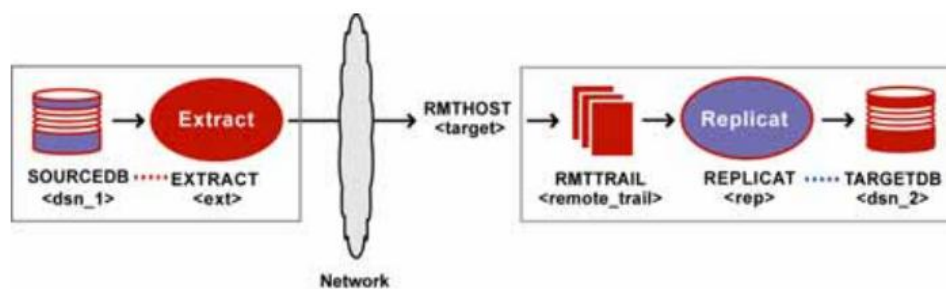
Master Note - Oracle GoldenGate (文档 ID 1298817.1)

Master Note for Oracle GoldenGate Core Product Patch Sets (文档 ID 1645495.1)

4. GoldenGate 架构示例

4.1. 报表系统配置实例

4.1.1. 标准配置



● Source system

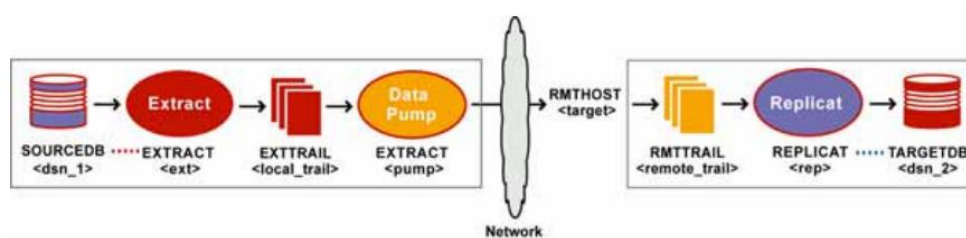
```
ADD EXTRACT <ext>, {TRANLOG | INTEGRATED TRANLOG}, BEGIN <time>[, THREADS <n>]
```

```
ADD RMTRAIL <remote_trail>, EXTRACT <ext>
```

● Target system

```
ADD REPLICAT <rep>, EXTRACT <remote_trail>, BEGIN <time>
```

4.1.2. data pump 模式配置



● Source system

```
ADD EXTRACT <ext>, {TRANLOG | INTEGRATED TRANLOG}, BEGIN <time>[, THREADS <n>]
```

```
ADD EXTTRAIL <local_trail>, EXTRACT <ext>
```

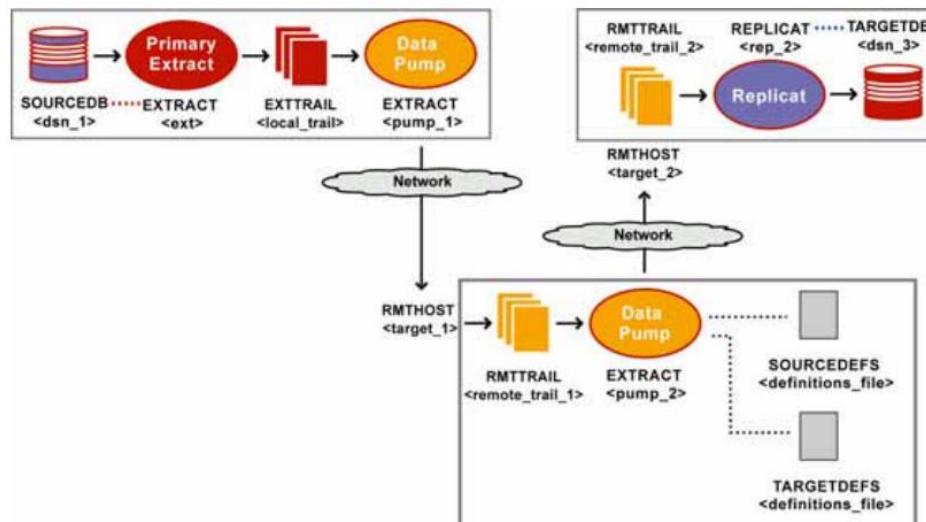
```
ADD EXTRACT <pump>, EXTTRAILSOURCE <local_trail>, BEGIN <time>
```

```
ADD RMTRAIL <remote_trail>, EXTRACT <pump>
```

- Target system

```
ADD REPLICAT <rep>, EXTTRAIL <remote_trail>, BEGIN <time>
```

4.1.3. data pump 中间系统配置



- Source system

```
ADD EXTRACT <ext>, {TRANLOG | INTEGRATED TRANLOG}, BEGIN <time>[, THREADS <n>]
```

```
ADD EXTTRAIL <local_trail>, EXTRACT <ext>
```

```
ADD EXTRACT <pump_1>, EXTTRAILSOURCE <local_trail>, BEGIN <time>
```

```
ADD RMTRAIL <remote_trail_1>, EXTRACT <pump_1>
```

- Intermediary system

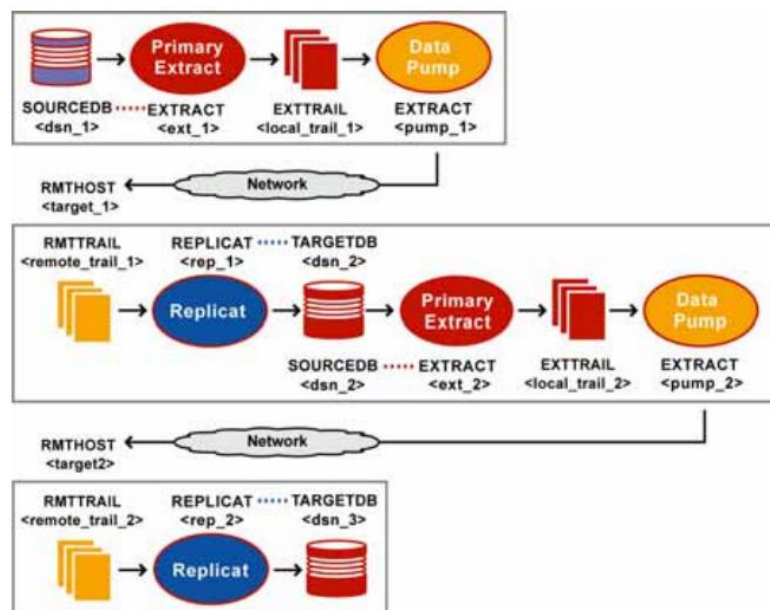
```
ADD EXTRACT <pump_2>, EXTTRAILSOURCE <local_trail_1>, BEGIN <time>
```

```
ADD RMTRAIL <remote_trail_2>, EXTRACT <pump_2>
```

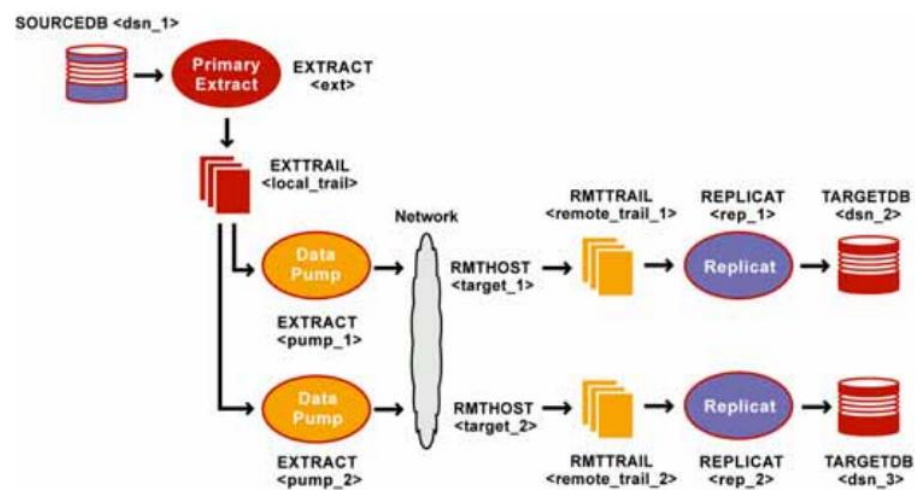
- Target system

```
ADD REPLICAT <rep>, EXTTRAIL <remote_trail_2>, BEGIN <time>
```

4.1.4. 级联配置



4.2. 实时数据分发配置



- **Source system**

```
ADD EXTRACT <ext>, {TRANLOG | INTEGRATED TRANLOG}, BEGIN <time>[, , THREADS]
```

```
ADD EXTTRAIL <local_trail>, EXTRACT <ext>
```

```
ADD EXTRACT <pump_1>, EXTTRAILSOURCE <local_trail>, BEGIN <time>
```

```
ADD RMTTRAIL <remote_trail_1>, EXTRACT <pump_1>
```

```
ADD EXTRACT <pump_2>, EXTTRAILSOURCE <local_trail>, BEGIN <time>

ADD RMTTRAIL <remote_trail_2>, EXTRACT <pump_2>
```

● Target systems

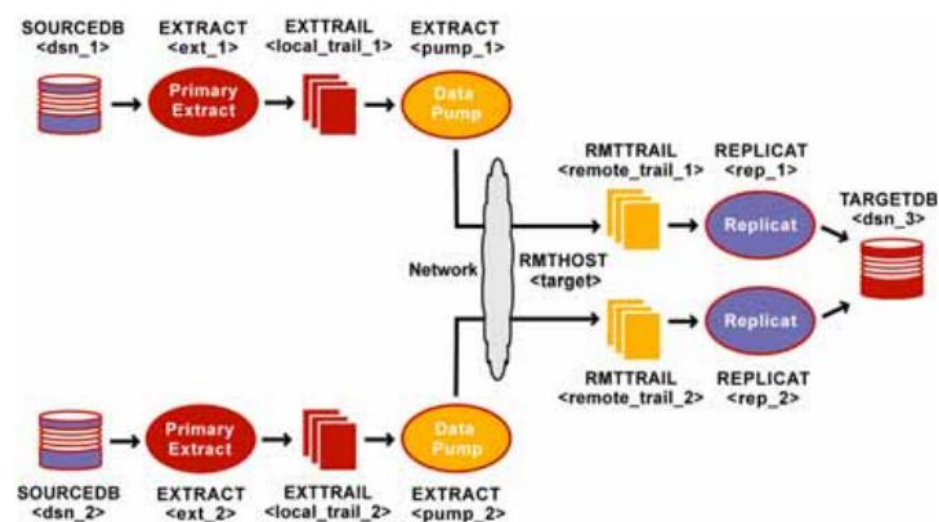
```
Target_1

ADD REPLICAT <rep_1>, EXTTRAIL <remote_trail_1>, BEGIN <time>

Target_2

ADD REPLICAT <rep_2>, EXTTRAIL <remote_trail_2>, BEGIN <time>
```

4.3. 实时数据仓库配置



● Source systems

```
Extract_1

ADD EXTRACT <ext_1>, {TRANLOG | INTEGRATED TRANLOG}, BEGIN <time>[, THREADS <n>]

ADD EXTTRAIL <local_trail_1>, EXTRACT <ext_1>

ADD RMTTRAIL <remote_trail_1>, EXTRACT <pump_1>

Data pump_1

ADD EXTRACT <pump_1>, EXTTRAILSOURCE <local_trail_1>, BEGIN <time>

Extract_2

ADD EXTRACT <ext_2>, {TRANLOG | INTEGRATED TRANLOG}, BEGIN <time>[, THREADS <n>]
```

```
ADD EXTTRAIL <local_trail_2>, EXTRACT <ext_2>
```

```
Data pump_2
```

```
ADD EXTRACT <pump_2>, EXTTRAILSOURCE <local_trail_2>, BEGIN <time>
```

```
ADD RMTTRAIL <remote_trail_2>, EXTRACT <pump_2>
```

● Target system

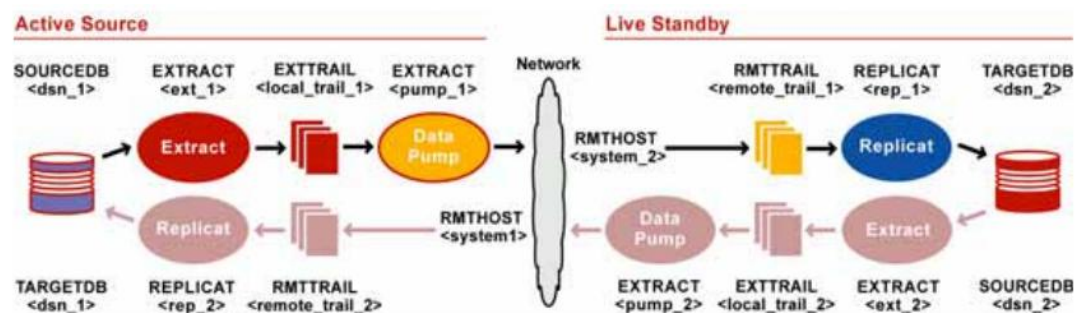
```
Replicat_1
```

```
ADD REPLICAT <rep_1>, EXTTRAIL <remote_trail_1>, BEGIN <time>
```

```
Replicat_2
```

```
ADD REPLICAT <rep_2>, EXTTRAIL <remote_trail_2>, BEGIN <time>
```

4.4. 实时备用数据库



● active source to standby

```
ADD EXTRACT <ext_1>, {TRANLOG | INTEGRATED TRANLOG}, BEGIN <time>[, THREADS <n>]
```

```
ADD EXTTRAIL <local_trail_1>, EXTRACT <ext_1>
```

```
ADD EXTRACT <pump_1>, EXTTRAILSOURCE <local_trail_1>, BEGIN <time>
```

```
ADD RMTTRAIL <remote_trail_1>, EXTRACT <pump_1>
```

```
ADD REPLICAT <rep_1>, EXTTRAIL <remote_trail_1>, BEGIN <time>
```

● standby to active source

```
ADD EXTRACT <ext_2>, {TRANLOG | INTEGRATED TRANLOG}, BEGIN <time>[, THREADS <n>]
```

```
ADD EXTTRAIL <local_trail_2>, EXTRACT <ext_2>
```

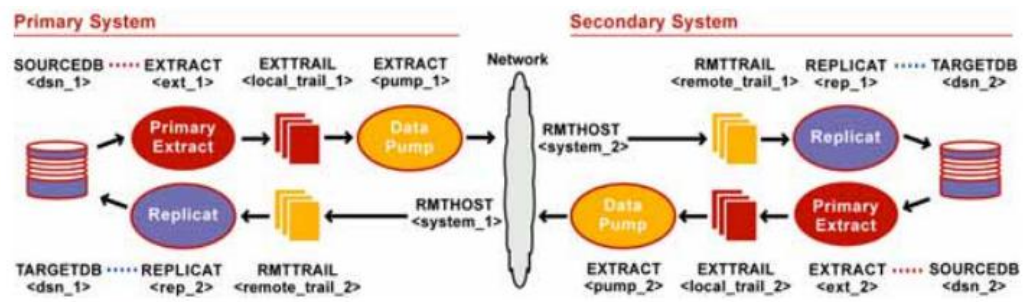


```
ADD EXTRACT <pump_2>, EXTRACTSOURCE <local_trail_2>, BEGIN <time>
```

```
ADD RMTTRAIL <remote_trail_2>, EXTRACT <pump_2>
```

```
ADD REPLICAT <rep_2>, EXTTRAIL <remote_trail_2>, BEGIN <time>
```

4.5. 双活高可用配置



5. 数据操作与映射

TABLE is used in the Extract parameter file, and MAP is used in the Replicat parameter file.

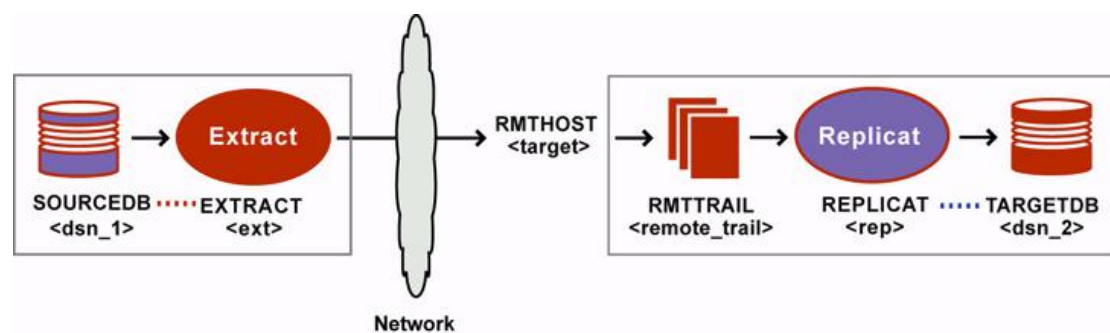
5.1. COLMAP

6. 目标数据库初始化

<http://www.itpub.net/forum.php?mod=viewthread&tid=1781501&highlight=goldengate>

7. GoldenGate 配置示例

7.1. DML 数据同步简单配置



7.1.1. 源库同步配置

- 源库配置提取进程 (extract process)

GGSCI (rac11g1) 3> `add extract srcdbext,tranlog,begin now`

extract added.

新增提取进程 `srcdbext`，基于日志方式，立即生效。需要注意的是，GoldenGate 中的进程名最长只能 8 个字符。

注意，如果是在 RAC 环境下，需要再加上 `THREADS <n>` 指定提取的日志 THREAD

GGSCI> `add extract srcdbext, tranlog,begin now, THREADS <n>`

- 创建提取进程远端队列

给这个提取进程创建远端队列，即图中的 RMTTRAIL:

GGSCI> `add rmttrail ./dirdat/rl, extract srcdbext`

需要注意的是，这里最后的名称 `rl`(可以自定义，字母或数字)是指以后生成的 `trail` 文件都会以 `rl` 开头，如 `rl000001`，这个文件头名称最长只能 2 个字符。如果需要在文件级别区分队列，建议将队列创建于不同的目录下。

➤ 编辑提取进程配置

GGSCI (rac11g1) 5> [edit param srcdbext](#)

```

extract srcdbext                                --抽取进程名
SETENV(NLS_LANG = "AMERICAN_AMERICA.ZHS16GBK")
SETENV(ORACLE_HOME = "/oracle/app/oracle/product/11.2.0/db_1")
SETENV(ORACLE_SID = "source")

userid ggadmin, password GGADMIN                --连接本机 DB 的帐号密码
rmthost 192.168.56.130, mgrport 7801            --目标数据库服务器地址和 GG 服务端口号
rmtrail /oracle/goldengate/dirdat/rl            --远程队列的位置(下一步建的东西)
dynamicresolution                               --优化参数，动态分析表结构
gettruncates                                    --抓取 truncate 数据
table sky.*;                                    --需要抽取的表，可以使用通配符

```

➤ 开启提取进程

GGSCI (rac11g1) 13> [START EXTRACT SRCDBEXT](#)

```

Sending START request to MANAGER ...
extract SRCDBEXT starting

```

➤ 查看进程状态

GGSCI (rac11g1) 14> [INFO ALL](#)

Program	Status	Group	Lag at Chkpt	Time Since Chkpt
MANAGER	RUNNING			
extract	RUNNING	SRCDBEXT	00:00:00	00:00:52

➤ 查看进程日志

GGSCI (gxoradb1) 15> [view report srcdbext](#)

7.1.2. 目标库同步配置

➤ 配置应用进程(replicat process)

新增复制进程，使用对应的队列 **rl**(该队列名称与源库创建的队列名称要一致 **add rmttrail ./dirdat/rl**)，即刻开始，使用文件检查点。

```
GGSCI> add replicat repsrcdb exttrail ./dirdat/rl, begin now, nodbcheckpoint
```

➤ 编辑应用进程参数

```
GGSCI (gxoradb2) 6> edit param repsrcdb
```

```
replicat repsrcdb                                --复制进程名
SETENV (NLS_LANG = "AMERICAN_AMERICA.ZHS16GBK")
SETENV (ORACLE_HOME = "/oracle/app/oracle/product/11.2.0/db_1")
SETENV (ORACLE_SID = "source")
userid ggadmin, password GGADMIN                  --连接本机 DB 的帐号密码
assumetargetdefs                                  --两台数据库数据结构一致则使用此参数
repreerror default,discard                        --如果复制数据出错则忽略
gettruncates                                       --复制 truncate 操作
map sky.t, target sky.t;                          --映射关系，注意 target 前必须留一个空格
```

➤ 启动复制进程

```
GGSCI (gxoradb2) 7> START REPLICAT REPSRCDB
```

7.1.3. 数据同步测试

➤ 主库数据录入

```
[oracle@rac11g1 archive]$ sqlplus sky/sky@source
```

```
SQL> insert into t values (1);
```

```
SQL> commit;
```

```
SQL> select * from t;
```

ID

```
-----
1
```

➤ 目标库数据查询

```
[oracle@rac11g2 admin]$ sqlplus sky/sky@target
```

```
SQL> select * from t;
```

```
ID
-----
1
```

7.1.4. 清理测试环境

➤ 停止目标库进程

```
GGSCI (rac11g2) 18> INFO ALL
```

Program	Status	Group	Lag at Chkpt	Time Since Chkpt
MANAGER	RUNNING			
REPLICAT	RUNNING	REPSRCDB	00:00:00	00:00:09

```
GGSCI (rac11g2) 19> STOP REPLICAT REPSRCDB
```

```
Sending STOP request to REPLICAT REPSRCDB ...
```

```
Request processed.
```

```
GGSCI (rac11g2) 20> DELETE REPLICAT REPSRCDB
```

```
Deleted REPLICAT REPSRCDB.
```

```
GGSCI (rac11g2) 18> INFO ALL
```

Program	Status	Group	Lag at Chkpt	Time Since Chkpt
MANAGER	RUNNING			

➤ 停止源库进程

```
GGSCI (rac11g1) 45> INFO ALL
```

Program	Status	Group	Lag at Chkpt	Time Since Chkpt
MANAGER	RUNNING			
EXTRACT	RUNNING	SRCDBEXT	00:00:00	00:00:07

GGSCI (rac11g1) 46> **STOP EXTRACT SRCDBEXT**

Sending STOP request to EXTRACT SRCDBEXT ...

Request processed.

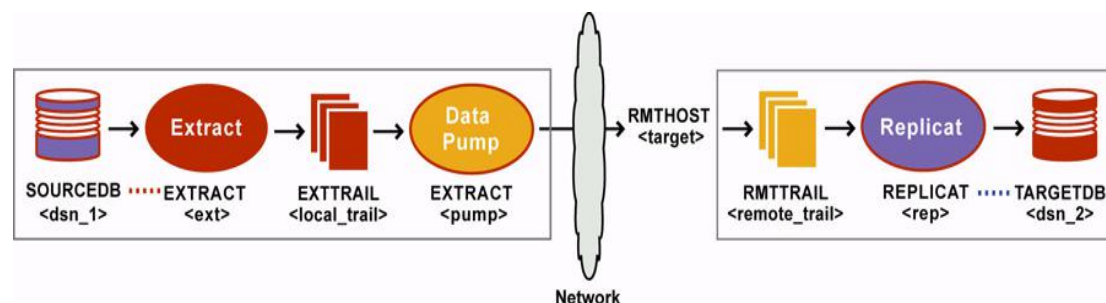
GGSCI (rac11g1) 47> **DELETE EXTRACT SRCDBEXT**

Deleted EXTRACT SRCDBEXT.

GGSCI (rac11g1) 45> **INFO ALL**

Program	Status	Group	Lag at Chkpt	Time Since Chkpt
MANAGER	RUNNING			

7.2. 使用 DATA PUMP 方式进行同步



这里的 DATAPUMP 与 ORACLE 10g 推出的数据泵不是一个概念。在 GoldenGate 中，DATAPUMP 相当于一个次级提取进程(secondary extract)。在上面演示的同步流程中，提取进程直接将提取的 redo 信息经过处理后放置到了目标端服务器上，当两者之间的网络出现故障时，会因无法生成 trail 文件而导致提取进程崩溃，错误提示类似如下：

```

2015-11-27 21:16:40 WARNING OGG-01223 TCP/IP error 111 (Connection refused).
2015-11-27 21:16:56 INFO OGG-01226 Socket buffer size set to 27985 (flush size 27985).
2015-11-27 21:16:56 INFO OGG-01230 Recovered from TCP error, host 192.168.56.130, port 7819.
2015-11-27 21:16:56 INFO OGG-01052 No recovery is required for target file
/oracle/goldengate/dirdat/rl000000, at RBA 0 (file not opened).
2015-11-27 21:16:56 INFO OGG-01478 Output file /oracle/goldengate/dirdat/rl is using
format RELEASE 11.2.

```


而加入 DATAPUMP 后，主提取进程(即第一个 **extract**)首先将 **trail** 生成在本地，然后 DATAPUMP 读取本地 **trail** 再发送到目标服务器，即便网络故障，主提取进程仍然能随着事务生成 **trail** 文件，而 DATAPUMP 则会暂时停止传输，等待网络通畅后在将堆积的本地 **trail** 文件发送至目标服务器，从而实现了断点传输的功能。在实际应用中，每一个同步流程都应该配置 DATAPUMP 以应对网络问题。

7.2.1. 源库同步配置

➤ 源库配置提取进程 (extract process)

GGSCI (rac11g1) 3> **add extract srdbext, tranlog, begin now**

```
extract added.
```

➤ 创建提取进程本地队列

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。

GGSCI> **add exttrail ./dirdat/et, extract srdbext**

➤ 编辑提取进程配置

GGSCI (rac11g1) 5> **edit param srdbext**

```
extract srdbext
SETENV (NLS_LANG = "AMERICAN_AMERICA.ZHS16GBK")
SETENV (ORACLE_HOME = "/oracle/app/oracle/product/11.2.0/db_1")
SETENV (ORACLE_SID = "source")
userid ggadmin, password GGADMIN
exttrail ./dirdat/et
dynamicresolution
gettruncates
table sky.*;
```

➤ 源库配置 PUMP 进程

在源端一个数据 Pump 进程是 secondary Extract 进程组，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，能稳定且没有差错的把数据投递到目标端。

Pump 进程需要指定本地的提取进程队列。

GGSCI (rac11g1) 3> **add extract pumsrc, exttrailsource ./dirdat/et**

```
extract added.
```

➤ 创建 PUMP 进程远程队列

给这个提取进程创建远程队列：

GGSCI> **add rmtrail ./dirdat/pt, extract pumsrc**

➤ 编辑提取进程配置

GGSCI (rac11g1) 5> **edit param pumsrc**

```
extract pumsrc
SETENV(NLS_LANG = "AMERICAN_AMERICA.ZHS16GBK")
SETENV(ORACLE_HOME = "/oracle/app/oracle/product/11.2.0/db_1")
SETENV(ORACLE_SID = "source")
userid ggadmin, password GGADMIN
rmthost 192.168.56.130, mgrport 7801
rmtrail ./dirdat/pt
passthru
gettruncates
table sky.*;
```

➤ 启动提取进程

GGSCI (rac11g1) 25> **START EXTRACT SRCDBEXT**

```
Sending START request to MANAGER ...
```

```
EXTRACT SRCDBEXT starting
```

GGSCI (rac11g1) 26> **START EXTRACT PUMSRC**

```
Sending START request to MANAGER ...
```

```
EXTRACT PUMSRC starting
```

7.2.2. 目标库同步配置

新增复制进程，使用对应的队列 **pt**(该队列名称与源库创建的 **DATAPUMP** 队列名称要一致 **add rmttrail ./dirdat/pt**)，即刻开始，使用文件检查点

GGSCI> **add replicat repsrcdb exttrail ./dirdat/pt, begin now, nodbcheckpoint**

➤ 编辑应用进程参数

GGSCI (gxoradb2) 6> **edit param repsrcdb**

```
replicat repsrcdb                                --复制进程名
SETENV(NLS_LANG = "AMERICAN_AMERICA.ZHS16GBK")
SETENV(ORACLE_HOME = "/oracle/app/oracle/product/11.2.0/db_1")
SETENV(ORACLE_SID = "target")
userid ggadmin, password GGADMIN                  --连接本机 DB 的帐号密码
assumetargetdefs                                  --两台数据库数据结构一致则使用此参数
repererror default,discard                        --如果复制数据出错则忽略
discardfile ./dirrpt/repsrcdb.dis, append
gettruncates                                       --复制 truncate 操作
map sky.t, target sky.t;                          --映射关系，注意 target 前必须留一个空格
```

➤ 启动复制进程

GGSCI (gxoradb2) 7> **START REPLICAT REPSRCDB**

7.2.3. 数据同步测试

➤ 主库数据录入

[oracle@rac11g1 archive]\$ **sqlplus sky/sky@source**

```
SQL> insert into t values (1);
```

```
SQL> commit;
```

```
SQL> select * from t;
```

ID
1

➤ 目标库数据查询

```
[oracle@rac11g2 admin]$ sqlplus sky/sky@target
```

```
SQL> select * from t;
```

ID
1

7.2.4. 清理测试环境

环境清理方式，参考[清理测试环境](#)。

7.3. DDL 同步

GoldenGate 的 DDL 同步只支持两边一致的数据库，限制条件较多(如不能进行字段映射、转换等)，具体可以参考官方文档。DDL 的抓取不是通过日志抓取来捕获的，而是通过触发器来实现，所以对源数据库的性能影响要比单纯的数据抓取要大很多。GoldenGate 不支持 ORACLE 保留模式(SYS/SYSTEM/PUBLIC 等模式)的 DDL 复制操作。

7.3.1. 初始化 DDL 支持环境

所有 DDL 环境初始化脚本，均存放在 GoldenGate 安装目录下。在进行 DDL 配置前，建议停止所有的提取和应用进程。

➤ SQL> @marker_setup.sql

```
Marker setup script
```

```
Enter Oracle GoldenGate schema name:GGADMIN
```

```
Marker setup table script complete, running verification script...

Please enter the name of a schema for the GoldenGate database objects:

Setting schema name to GGADMIN

MARKER TABLE
-----

OK

MARKER SEQUENCE
-----

OK

Script complete.
```

➤ SQL> @ddl_setup.sql

```
Oracle GoldenGate DDL Replication setup script

Enter Oracle GoldenGate schema name:GGADMIN

DDL replication setup script complete, running verification script...

Please enter the name of a schema for the GoldenGate database objects:

Setting schema name to GGADMIN

CLEAR_TRACE STATUS:

cript complete.
```

➤ SQL> @role_setup.sql

```
GGs Role setup script

Enter GoldenGate schema name:GGADMIN

Grant this role to each user assigned to the Extract, GGSCI, and Manager processes, by using the
following SQL command:

GRANT GGS_GGSUSER_ROLE TO <loggedUser>
```

where <loggedUser> is the user assigned to the GoldenGate processes.

➤ SQL> GRANT GGS_GGSUSER_ROLE TO GGADMIN;

➤ SQL> @ddl_enable.sql

Trigger altered.

➤ SQL> @marker_status.sql

Please enter the name of a schema for the GoldenGate database objects:

GGADMIN

Setting schema name to GGADMIN

MARKER TABLE

OK

MARKER SEQUENCE

OK

7.3.2. 源库配置全局参数

GGSCI (rac11g1) 34> EDIT PARAM ./GLOBALS

GGSCHEMA GGADMIN

7.3.3. 修改提取进程配置

GGSCI (rac11g1) 39> INFO ALL

Program	Status	Group	Lag at Chkpt	Time Since Chkpt
MANAGER	STOPPED			
EXTRACT	STOPPED	PUMSRC	00:00:00	00:05:28
EXTRACT	STOPPED	SRADBEXT	00:00:00	00:05:27

GGSCI (rac11g1) 40> [EDIT PARAM SRADBEXT](#)

```
extract sradbext
SETENV(NLS_LANG = "AMERICAN_AMERICA.ZHS16GBK")
SETENV(ORACLE_HOME = "/oracle/app/oracle/product/11.2.0/db_1")
SETENV(ORACLE_SID = "source")
userid ggadmin, password GGADMIN
exttrail ./dirdat/et
dynamicresolution
gettruncates
ddl include all
ddloptions addtrandata, report
table sky.*;
```

GGSCI (rac11g1) 42> [VIEW PARAM PUMSRC](#)

```
extract pumsrc
SETENV(NLS_LANG = "AMERICAN_AMERICA.ZHS16GBK")
SETENV(ORACLE_HOME = "/oracle/app/oracle/product/11.2.0/db_1")
SETENV(ORACLE_SID = "source")
userid ggadmin, password GGADMIN
rmthost 192.168.56.130, mgrport 7801
rmtrail ./dirdat/pt
passthru
gettruncates
ddl include all
ddloptions addtrandata, report
table sky.*;
```

注意，PUMP 队列不能够配置 DDL 参数，否则队列无法正常启动：

```
2015-12-04 17:27:02 ERROR OGG-00528 The DDL parameter is not supported for
this data source. Please remove this parameter and al
I DDL operations will be propagated in pass-through (PASSTHRU) mode.
```

7.3.4. 修改应用进程配置

GGSCI (rac11g2) 9> [INFO ALL](#)

Program	Status	Group	Lag at Chkpt	Time Since Chkpt
MANAGER	STOPPED			
REPLICAT	STOPPED	REPSRCDB	00:00:00	00:08:44

GGSCI (rac11g2) 10> [EDIT PARAM REPSRCDB](#)

```
replicat repsrcdb
SETENV(NLS_LANG = "AMERICAN_AMERICA.ZHS16GBK")
SETENV(ORACLE_HOME = "/oracle/app/oracle/product/11.2.0/db_1")
SETENV(ORACLE_SID = "target")
userid ggadmin, password GGADMIN
assumetargetdefs
repererror default,discard
discardfile ./dirrpt/repsrcdb.dis, append
gettruncates
ddl include mapped
ddloptions report
map sky.*, target sky.*;
```

7.3.5. 测试 DDL 同步

➤ 源库执行 DDL 操作

SQL> [CREATE TABLE SKY.DDL_TAB \(ID NUMBER PRIMARY KEY\);](#)

➤ 目标库进行表查询

SQL> [SELECT * FROM SKY.DDL_TAB;](#)

no rows selected

➤ 源库写入测试数据

```
SQL> INSERT INTO SKY.DDL_TAB VALUES (1);
```

```
1 row created.
```

```
SQL> COMMIT;
```

```
Commit complete.
```

➤ 目标库进行表查询

```
SQL> SELECT * FROM SKY.DDL_TAB;
```

```
ID
```

```
1
```

7.3.6. 清除 DDL 设置

```
SQL> @ddl_disable.sql
```

```
SQL> @ddl_remove.sql
```

```
SQL> @marker_remove.sql
```

7.3.7. 参考文档

Error In DDL Replication Setup (Doc ID 1679898.1)

1. 执行 `ddl_setup.sql` 时提示权限不足，该问题是 11.2.1.0.1 的 bug，建议升级到最新版本

```
Why running ddl_setup.sql fails with
```

```
-----  
ERROR at line 1:
```

```
ORA-01031: insufficient privileges
```

```
ORA-06512: at "GGUSER.INITIAL_SETUP", line 477
```

```
ORA-06512: at line 1  
-----
```

errors even when database user has the correct privileges?

8. 异构数据库

8.1. MySQL

How to Replicate Data Between Oracle and MySQL Database? (Doc ID 1605674.1)

9. 常见命令说明

9.1. 登陆 GoldenGate 控制台

```
[oracle@rac11g1 ~]$ cd /oracle/goldengate/
```

```
[oracle@rac11g1 goldengate]$ ggsci
```

```
Oracle GoldenGate Command Interpreter for Oracle
Version 11.2.1.0.1 OGGCORE_11.2.1.0.1_PLATFORMS_120423.0230_FBO
Linux, x64, 64bit (optimized), Oracle 11g on Apr 23 2012 08:32:14
Copyright (C) 1995, 2012, Oracle and/or its affiliates. All rights reserved.

GGSCI (rac11g1) 1>
```

如出现登陆失败,则可能因环境变量设置有误,请确认 ORACLE_HOME 以及 LIBPATH 已设置。

9.2. 登陆数据库

```
GGSCI (rac11g1) 65> dblogin userid ggadmin@source, password GGADMIN
```

```
Successfully logged into database.
```

```
GGSCI (rac11g1) 66> VERSIONS
```

```
Operating System:
Linux
Version #1 SMP Tue Aug 12 06:26:17 EDT 2014, Release 2.6.18-398.el5
Node: rac11g1
Machine: x86_64

Database:
Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production
PL/SQL Release 11.2.0.4.0 - Production
CORE      11.2.0.4.0      Production
```

```
TNS for Linux: Version 11.2.0.4.0 - Production
NLSRTL Version 11.2.0.4.0 - Production
```

9.3. 查看历史执行命令

GGSCI (rac11g1) 6> **HISTORY**

GGSCI Command History

```
1: info all
2: start mgr
3: info all
4: history
5: history
6: HISTORY
```

！ 执行上一次执行的命令

FC 编辑上一个命令，并执行

9.4. 存储常用命令

[oracle@rac11g1 goldengate]\$ **vi SOURCE.OB**

```
START MGR
INFO ALL
```

GGSCI (rac11g1) 2> **OBEY SOURCE.OB**

GGSCI (rac11g1) 3> **START MGR**

Manager started.

GGSCI (rac11g1) 4> **INFO ALL**

Program	Status	Group	Lag at Chkpt	Time Since Chkpt
MANAGER	RUNNING			
EXTRACT	STARTING	PUMSRC	00:00:00	25:57:58

EXTRACT	STARTING	SRCDTEXT	00:00:00	09:05:57
---------	----------	----------	----------	----------

9.5. 通过 SHELL 执行脚本

```
[oracle@rac11g1 goldengate]$ ./ggsci < SOURCE.OB
```

命令格式: ggsci < <input_file>

注意: 使用脚本方式停止 **Manager** 进程时, 需要添加 ! 参数。

```
Oracle GoldenGate Command Interpreter for Oracle
Version 11.2.1.0.31 20976570 OGGCORE_11.2.1.0.OGGBP_PLATFORMS_150530.0925_FBO
Linux, x64, 64bit (optimized), Oracle 11g on May 30 2015 13:39:31
Copyright (C) 1995, 2015, Oracle and/or its affiliates. All rights reserved.

GGSCI (rac11g1) 1> MGR is already running.

GGSCI (rac11g1) 2>

Program      Status      Group      Lag at Chkpt  Time Since Chkpt
-----
MANAGER      RUNNING
EXTRACT      ABENDED     PUMSRC     00:00:00      26:02:32
EXTRACT      RUNNING     SRCDTEXT   00:00:00      00:00:07
```

9.6. 配置文件中变量

```
SOURCEISFILE
EXTFILE ?EXTFILE
MAP ?TABNAME, TARGET ACCOUNT_TARG;
```

```
$ export EXTFILE=/ggs/extfile
```

```
$ export TABNAME=PROD.ACCOUNTS
```

```
$ replicat paramfile ggs/dirprm/parmfl
```

9.7. 通配符说明

9.7.1. TARGET 配置

```
TABLE HQ.T_*, TARGET RPT.T_*;  
MAP HQ.T_*, TARGET RPT.T_*;
```

会被映射为

```
HQ.T_TEST1 maps to RPT.T_T_TEST1  
HQ.T_TEST2 maps to RPT.T_T_TEST2
```

因此 TARGET 只需指定 * 即可。

```
MAP HQ.T_*, TARGET RPT.*;
```

9.7.2. NAMEMATCHIGNORECASE

通过 NAMEMATCHIGNORECASE 参数，可以忽略映射过程中的名称大小写。

9.8. 查看进程状态

GGSCI (rac11g1) 3> [INFO ALL](#)

Program	Status	Group	Lag at Chkpt	Time Since Chkpt
MANAGER	RUNNING			
EXTRACT	ABENDED	SRADBEXT	00:00:00	02:41:50

- 依次为进程类型、运行状态、进程名字、时间延迟、检查点频率
- 运行状态三种 **RUNNING**、**STOPPED**、**ABENDED**，分别为正常运行、手动停止以及异常终止
- 正常情况下 **Lag** 以及 **Time Since Chkpt** 均小于 10s
- **MGR** 进程为管理进程
- 管理进程外的进程名字是自定义的，通常含 **EX** 为抽取进程、**DP** 为投递进程、**RE** 为复制进程

GGSCI (rac11g1) 6> [INFO EXTRACT SRADBEXT,SHOWCH](#)

```

EXTRACT    SRCDBEXT  Last Started 2015-11-29 01:53   Status RUNNING
Checkpoint Lag      00:00:00 (updated 02:43:59 ago)
Log Read Checkpoint Oracle Redo Logs
                  2015-11-28 23:07:33   Seqno 27, RBA 13896704
                  SCN 0.333142 (333142)

```

Current Checkpoint Detail:

Read Checkpoint #1

Oracle Redo Log

Startup Checkpoint (starting position in the data source):

Thread #: 1

Sequence #: 27

RBA: 13896208

Timestamp: 2015-11-28 23:07:33.000000

SCN: 0.333142 (333142)

Redo File: /oracle/app/oracle/oradata/source/redo03.log

GGSCI (rac11g2) 1> [INFO REPLICAT *](#)

GGSCI (rac11g1) 1> [INFO EXTRACT *](#)

```

EXTRACT    PUMSRC     Last Started 2015-12-03 13:40   Status RUNNING
Checkpoint Lag      00:00:00 (updated 00:00:07 ago)
Log Read Checkpoint File ./dirdat/et000001
                  2015-12-03 14:35:12.000000   RBA 2083

```

```

EXTRACT    SRCDBEXT  Last Started 2015-12-03 13:39   Status RUNNING
Checkpoint Lag      00:00:00 (updated 00:00:01 ago)
Log Read Checkpoint Oracle Redo Logs
                  2015-12-03 15:04:43   Seqno 30, RBA 31800320
                  SCN 0.455256 (455256)

```


9.9. 查看队列信息

GGSCI (rac11g1) 59> **INFO EXTTRAIL ***

GGSCI (rac11g1) 51> **INFO RMTTRAIL ***

```
Extract Trail: ./dirdat/pt
```

```
Extract: PUMSRC
```

```
Seqno: 1
```

```
RBA: 2220
```

```
File Size: 100M
```

```
Extract Trail: ./dirdat/et
```

```
Extract: SRCDBEXT
```

```
Seqno: 1
```

```
RBA: 2083
```

```
File Size: 100M
```

9.10. 队列加密

ENCRYPTTRAIL 参数在 EXTTRAIL, RMTTRAIL, EXTFILE, 或 RMTFILE 之前添加

ENCRYPTTRAIL [{AES128 | AES192 | AES256} KEYNAME <keyname>]

DECRYPTTRAIL [{AES128 | AES192 | AES256} KEYNAME <keyname>]

9.11. 用户口令加密

**ENCRYPT PASSWORD <password> <algorithm> ENCRYPTKEY {<keyname> |
DEFAULT}**

9.11.1. 使用 KEYGEN 命令生成 KEY

[oracle@rac11g1 goldengate]\$ **keygen 128 1**

0x78DB79565B161766437DFE32B839B509

9.11.2. 手工编辑 ENCKEYS 文件

文件的格式为

Key name	Key value
----------	-----------

[oracle@rac11g1 goldengate]\$ vi ENCKEYS

```
sourceogg 0x78DB79565B161766437DFE32B839B509
```

9.11.3. 生成口令密码

GGSCI (rac11g1) 1> ENCRYPT PASSWORD GGADMIN AES128 ENCRYPTKEY
sourceogg

Encrypted password:

```
AADAAAAAAAAAAHAIHBAEJTBQBIGXIBEMIUIINHCBGCGJFMDHEUGLCOICIEQETBOHTCDFKARGXIIJIAC
```

Algorithm used: AES128

9.11.4. 编辑参数文件

编辑参数文件之前需要停止进程，修改参数后再重新启动进程。

```
userid ggadmin, password GGADMIN
```

替换为:

```
userid ggadmin, PASSWORD
```

```
AADAAAAAAAAAAHAIHBAEJTBQBIGXIBEMIUIINHCBGCGJFMDHEUGLCOICIEQETBOHTCDFKARGXIIJIAC, &  
AES128, ENCRYPTKEY sourceogg
```

9.12. 查看参数文件

GGSCI (rac11g1) 15> VIEW PARAM SRCDBEXT

语法: VIEW PARAMS <group name>

```
extract srdbext
```

```
userid system@source,password oracle
```

```
rmthost 192.168.56.130, mgrport 7801
```

```
rmtrail /oracle/goldengate/dirdat/rl
```

```
dynamicresolution
```

```
gettruncates
```

```
table sky.*;
```

9.13. 编辑参数文件

GGSCI (rac11g1) 16> [EDIT PARAM SRCDBEXT](#)

在进行参数编辑之前，必须停掉对应的进程。

```
extract srcdbext
userid system@source,password oracle
rmthost 192.168.56.130, mgrport 7801
rmttrail /oracle/goldengate/dirdat/rl
dynamicresolution
gettruncates
table sky.*;
```

9.14. 启动进程

GGSCI (rac11g1) 19> [START EXTRACT SRCDBEXT](#)

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

9.15. 停止进程

GGSCI (rac11g1) 20> [STOP EXTRACT SRCDBEXT](#)

```
Sending STOP request to EXTRACT SRCDBEXT ...
Request processed.
```

9.16. 强制终止

GGSCI (rac11g1) 18> [KILL EXTRACT SRCDBEXT](#)

```
Sending KILL request to MANAGER ...
Killed process (3894) for EXTRACT SRCDBEXT
```

9.17. 查看进程统计信息

GGSCI (rac11g1) 9> [STATS SRCDBEXT, TABLE SKY.T, TOTAL](#)

Sending STATS request to EXTRACT SRCDBEXT ...

Start of Statistics at 2015-12-01 21:07:27.

Output to /oracle/goldengate/dirdat/rl:

Extracting from SKY.T to SKY.T:

*** Total statistics since 2015-12-01 21:07:25 ***

Total inserts	1.00
Total updates	0.00
Total deletes	0.00
Total discards	0.00
Total operations	1.00

End of Statistics.

9.18. 查看进程报告

GGSCI (rac11g1) 12> [VIEW REPORT SRCDBEXT](#)

Oracle GoldenGate Capture for Oracle

Version 11.2.1.0.1 OGGCORE_11.2.1.0.1_PLATFORMS_120423.0230_FBO

Linux, x64, 64bit (optimized), Oracle 11g on Apr 23 2012 08:42:16

Copyright (C) 1995, 2012, Oracle and/or its affiliates. All rights reserved.

Starting at 2015-12-01 21:05:29

Operating System Version:

Linux

Version #1 SMP Tue Aug 12 06:26:17 EDT 2014, Release 2.6.18-398.el5

Node: rac11g1

Machine: x86_64

	soft limit	hard limit
Address Space Size :	unlimited	unlimited
Heap Size :	unlimited	unlimited
File Size :	unlimited	unlimited
CPU Time :	unlimited	unlimited

Process id: 3894

10. 常用文件目录说明

下面提到的文件以及目录默认均在 **GoldenGate** 进程的安装目录下。

GGSCI (rac11g1) 72> **SHOW**

```
Parameter settings:
SET SUBDIRS      ON
SET DEBUG        OFF
Current directory: /oracle/goldengate
Using subdirectories for all process files

Editor:  vi

Reports (.rpt)           /oracle/goldengate/dirrpt
Parameters (.prm)        /oracle/goldengate/dirprm
Stdout (.out)            /oracle/goldengate/dirout
Replicat Checkpoints (.cpr) /oracle/goldengate/dirchk
Extract Checkpoints (.cpe) /oracle/goldengate/dirchk
Process Status (.pcs)     /oracle/goldengate/dirpcs
SQL Scripts (.sql)        /oracle/goldengate/dirsql
Database Definitions (.def) /oracle/goldengate/dirdef
```

10.1. ggserr.log

ggserr.log 记录各 **GoldenGate** 进程的信息。该文件存放在 **GoldenGate** 安装目录的根目录下。

由于 **dirrpt** 目录下报错的历史 **report** 记录数量有限，要查询更早期的 **GG ERROR** 等信息，只能通过 **ggserr.log** 文件查看。

通常此文件比较大，请用 **tail** 命令先选去末尾行。

10.2. dirrpt 目录

dirrpt 目录里面存储各进程运行报告以及丢弃文件记录，除了当前那一份外，还有十份历史记录。

这些记录为分析 GoldenGate 状况的重要依据。

10.3. dirprm 目录

dirprm 目录里面存储各进程的参数文件。

10.4. dirdat 目录

dirdat 为默认的队列文件存放位置，队列文件中存放库表变化信息。

10.5. dirpcs 目录

dirpcs 目录里面存储各进程的状态信息。

10.6. dirchk 目录

dirchk 目录里面存储各进程的检查点文件，此文件十分重要，如果丢失，进程将无法启动。

10.7. dirtmp 目录

dirtmp 为临时文件目录，一般情况下为空。

11. 常用错误处理

11.1. OGG-00037 DISCARDFILE file ./dirrpt/repsrcdb.dis already exists.

```
2015-12-07 09:22:45 ERROR OGG-00037 DISCARDFILE file ./dirrpt/repsrcdb.dis already exists.  
2015-12-07 09:22:45 ERROR OGG-01668 PROCESS ABENDING.
```

应用进程启动失败，出现 OGG-00037 错误，这个是由于 REPLICAT 进程中的 DISCARDFILE 没有文件处理策略有关，添加 APPEND | PURGE 即可。

```
DISCARDFILE <file name>  
[, APPEND | PURGE]  
[, MAXBYTES <n> | MEGABYTES <n>]
```