

# SuperSync 用戶操作指南

## 法律声明

- 迪思杰集团拥有本产品及相关文档的全部版权，受法律保护。
- 未经本公司书面许可，任何单位及个人不得以任何方式或理由对本产品的任何部分进行复制、抄录和传播或与其他产品捆绑使用、销售。
- 本文档中所提及的其他商标或者产品名称均为各自持有者所拥有的商标或产品名称，版权归相应的持有者所有。

## 联系我们

迪思杰集团欢迎您通过尽可能多的渠道向我们提供信息，您的意见和问题都会得到我们的重视和妥善处理，请将反馈信息投递到下述地址：

迪思杰集团北京总部	
地址：	北京市西城区新外大街 28 号办公楼 B 座 210 室
邮编：	100088
电话：	010-82051581(82/84/85/86/87)
传真：	010-82051590
E-mail：	Market@dsgdata.com
上海办事处	
地址：	上海市徐汇区漕宝路 86 号光大会展中心 F 座 2103 室
电话：	021-64320873
广州办事处	
地址：	广州市天河区侨林街 63 号桥林苑 B 座 502 室
电话：	020-38827900
成都办事处	
地址：	成都市武侯区桐梓林北路二号凯莱帝景花园 B 座 15 楼 B 室
电话：	028-61540211
沈阳办事处	
地址：	沈阳市沈河区奉天街 373-1 号领秀 e 家 C2 座 10 楼 3 号
电话：	024-88525233

南京办事处	
地址:	南京市鼓楼区汉中路 8 号金轮国际广场 1820 室
电话:	025-84773839
杭州办事处	
地址:	杭州市西湖区文三路 477 号华星科技大厦 430 室
电话:	0571-56883636
长沙办事处	
地址:	湖南省长沙市雨花区芙蓉中路二段 459 号芙蓉公馆 A 栋 1 单元 2312 室
电话:	0731-85860224
福州办事处	
地址:	福州市井大路 113 号七星佳寓 1210 房
电话:	0591-87400366
武汉办事处	
地址:	武汉市东湖新技术开发区鲁巷光谷街尚都酒店二栋 2208 室
电话:	027-87644355
西安办事处	
地址:	西安市雁塔区高新开发区高新一路 5 号正信大厦 A 座 306 室
电话:	029-83151908
济南办事处	
地址:	济南市高新区新泺大街 1299 号鑫盛大厦 1 号楼 14B
电话:	0531-88813281

# 目录

---

前言.....	1
<b>1 SuperSync 概述.....</b>	<b>2</b>
1.1 SuperSync 原理.....	2
1.2 SuperSync 复制目录结构.....	3
1.2.1 SuperSync 源端复制目录结构.....	3
1.2.2 SuperSync 目标端复制目录结构.....	6
<b>2 启动/停止 SuperSync 进程.....</b>	<b>11</b>
2.1 启动/停止源端进程.....	11
2.1.1 启动源端进程.....	11
2.1.2 停止源端进程.....	13
2.2 启动/停止目标端进程.....	14
2.2.1 启动目标端进程.....	14
2.2.2 停止目标端进程.....	18
2.3 单独启动/停止代理进程.....	18
2.3.1 单独启动/停止源端 Vagentd 进程.....	18
2.3.2 单独启动/停止目标端 Vagent 进程.....	19
2.4 单独启动/停止源端日志分析进程.....	20
2.5 单独启动/停止源端发送进程.....	21
2.6 单独启动/停止目标端首次同步装载进程.....	22
2.7 单独启动/停止目标端实时装载进程.....	23
<b>3 首次初始化.....</b>	<b>25</b>
3.1 Oracle 数据库间首次全同步.....	25
3.2 异种数据库间首次全同步.....	28
<b>4 配置复制.....</b>	<b>32</b>

4.1	单独同步数据表 .....	32
4.1.1	单独同步单张表 .....	32
4.1.2	单独同步多张表 .....	34
4.1.3	单独同步新增的多张表 .....	36
4.2	取消实时同步单张表 .....	38
4.3	确认同步表 .....	39
<b>5</b>	<b>打开源端/目标端 DSG 备份文件 .....</b>	<b>41</b>
5.1	打开源端 vagentd 备份文件 .....	41
5.2	打开源端 sender 备份文件 .....	42
5.3	打开目标端 loader -s 备份文件 .....	43
5.4	打开目标端 loader -r 备份文件 .....	44
<b>6</b>	<b>修改 vman .....</b>	<b>46</b>
6.1	修改 vman 中的主机 IP .....	46
6.2	修改 vman 中的数据库 DSG 用户密码 .....	49
<b>7</b>	<b>SuperSync 装载分发 .....</b>	<b>53</b>
7.1	全同步装载分发 .....	53
7.2	实时同步装载分发 .....	55
7.2.1	首次同步过程中分发实时装载 .....	55
7.2.2	实时同步过程中分发实时装载 .....	57
<b>8</b>	<b>DDL 过滤 .....</b>	<b>64</b>
8.1	对所有表进行 DDL 过滤 .....	64
8.2	对特定表进行 DDL 过滤 .....	67
<b>9</b>	<b>查看性能统计分析 .....</b>	<b>72</b>
9.1	查看源端性能统计分析 .....	72
9.2	查看目标端性能统计分析 .....	75
<b>10</b>	<b>系统监控 .....</b>	<b>78</b>
10.1	源端监控 .....	78
10.2	目标端监控 .....	79
<b>11</b>	<b>数据比对 .....</b>	<b>82</b>

<b>12</b>	<b>xf1 文件操作</b>	<b>89</b>
12.1	查看 xf1 文件	89
12.1.1	查看 xf1 文件的 Rowid 信息	89
12.1.2	查看 xf1 文件的十六进制信息	90
12.1.3	查看 xf1 文件的 SQL 语句	92
12.2	删除 xf1 文件中的某个 objn 交易	94
12.3	删除 xf1 文件中的某类操作的交易	95
12.4	删除 xf1 文件中的某条 Rowid 记录	96
<b>13</b>	<b>查看 SuperSync 日志</b>	<b>98</b>
13.1	查看 SuperSync 源端日志	98
13.1.1	查看源端 log.vagentd 日志	98
13.1.2	查看源端 log.sender 日志	99
13.1.3	查看源端 archivelog 日志	100
13.2	查看 SuperSync 目标端日志	100
13.2.1	查看目标端 log.vagentd 日志	100
13.2.2	查看目标端 log.s* 日志	101
13.2.3	查看目标端 log.r* 日志	102
13.2.4	查看目标端 archivelog 日志	103
<b>14</b>	<b>查看 SuperSync 版本信息</b>	<b>105</b>

# 前言

## 概述

本文档介绍 SuperSync 软件日常操作步骤，帮助用户了解和掌握 SuperSync 软件。

## 产品版本

产品	版本
SuperSync	V6.0
Oracle	8i、9i、10g、11g、12c

## 读者对象

本文档主要适用于以下工程师：

- 现场工程师
- 系统管理员

适用本文档应该具备的技能条件：

- 熟悉 Linux 基本命令
- 熟悉 SQL 基本命令
- 掌握 Oracle 原理

# 1 SuperSync 概述

介绍 SuperSync，使用户了解 SuperSync 产品。

本章节包含下列主题：

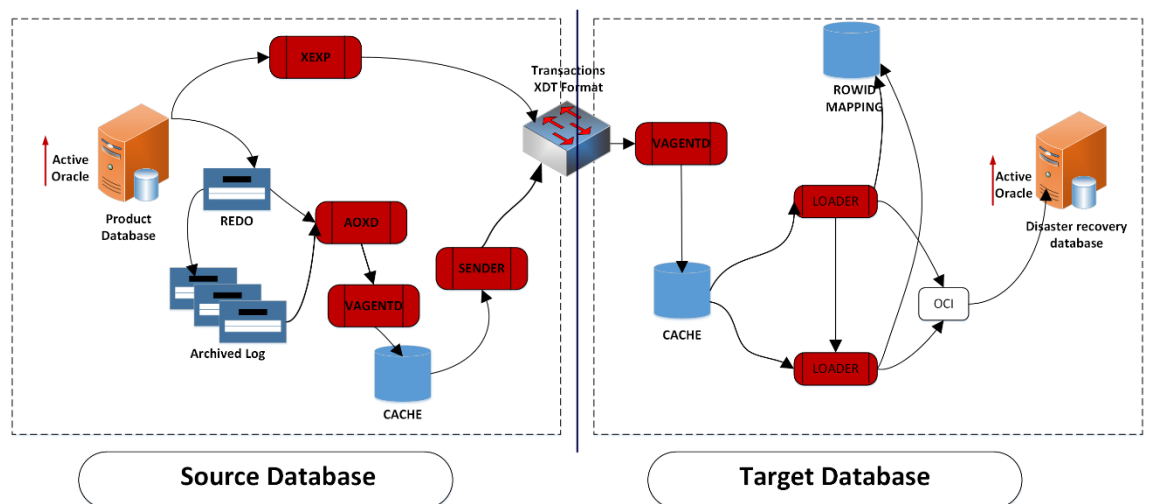
- SuperSync 原理
- SuperSync 复制目录结构

## 1.1 SuperSync 原理

SuperSync 是 DSG 公司推出的数据库异构复制热容灾解决方案，实现数据的高效复制及容灾。SuperSync 的设计目的是为企业系统间的信息流通和共享，提升企业信息价值。企业只需要根据自己的业务规则指定信息的获取或发布策略，即可轻易实现信息流转。

SuperSync 设计用于为 Oracle 数据库进行数据复制。与传统的数据复制技术不同，SuperSync 复制软件是针对数据库提供基于逻辑的交易复制方式。该方式通过直接捕获源数据库的交易，将数据库的改变逻辑复制到目标系统数据库中，实现源系统和目标系统数据的一致性。SuperSync 的工作原理如图 1 所示。

图 1 SuperSync 工作原理



SuperSync 在源端安装 AOXD 和 Agent 进程，在数据库目标端安装 Agent 进程，源端的 AOXD 进程对 Oracle 日志进行监控，发现改变则通过 Agent 及时对目标数据库进行更新。

当应用系统在源端向数据库进行任何操作时，这些信息都将在 Redo Log 中保存，SuperSync 的 AOXD 通过对实时获取的 Log 日志进行分析，将 Oracle 日志以最快的速度转换为 XDT 格式输出，并实时通过网络传送到目标端系统中。



目标端系统的 Agent 接收数据库包，经过校验码检查，确认正确的数据库包后，调用 Oracle、SQL Server、DB2 等函数按照交易的先后顺序在目标端系统中执行该交易。

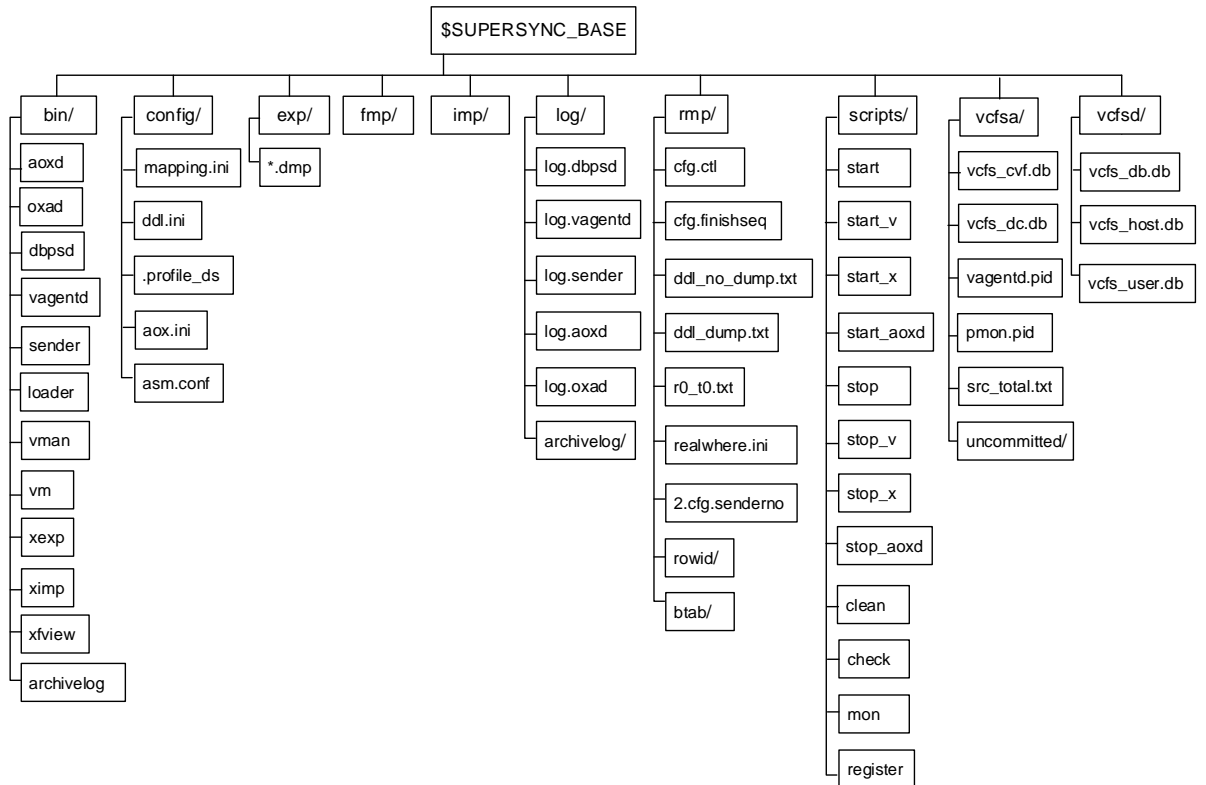
## 1.2 SuperSync 复制目录结构

SuperSync 安装完成后，将在源端和目标端生成复制目录结构，SuperSync 的操作运行都是在这些目录结构中进行的。

### 1.2.1 SuperSync 源端复制目录结构

SuperSync 源端复制目录结构如图 2 所示。

图 2 SuperSync 源端复制目录结构



SuperSync 源端复制目录的具体说明如表 1 所示。

表 1 源端 SuperSync 复制目录说明

<b>\$SUPERSYNC_BASE:</b> 为 SuperSync 源端安装时指定的目录	
<b>bin 目录:</b> SuperSync 执行文件存放目录	
aoxd	SuperSync aoxd 程序，用于并发日志分析
oxad	SuperSync oxad 程序
dbpsd	SuperSync Server 程序
vagentd	SuperSync Vagentd 程序

sender	SuperSync xf1 传送程序
loader	SuperSync xf1 接收程序
vman	SuperSync 管理配置工具
vm	vman 的配置文件
xexp	SuperSync 逻辑导出工具
ximp	SuperSync 逻辑导入工具
xfview	xf1 文件查看工具
archivelog	归档日志分析工具
config 目录: SuperSync 配置文件存放目录	
mapping.ini	SuperSync 数据同步配置文件
ddl.ini	SuperSync DDL 过滤配置文件
.profile_ds	SuperSync 源端配置文件
.profile_dt	SuperSync 目标端配置文件
aox.ini	aoxd 配置文件
asm.conf	ASM 配置文件
exp 目录: SuperSync 数据导出目录。	
*.dmp	SuperSync 导出文件
fmp 目录: SuperSync 交易缓存目录, 该目录在源端为空	
imp 目录: SuperSync 数据导入目录, 该目录在源端为空	
log 目录: 存放日志的目录	
log.aoxd	Aoxd 启动运行日志
log.oxad	Oxad 启动运行日志
log.dbpsd	dbpsd 启动运行日志
log.vagentd	Vagentd 启动运行日志

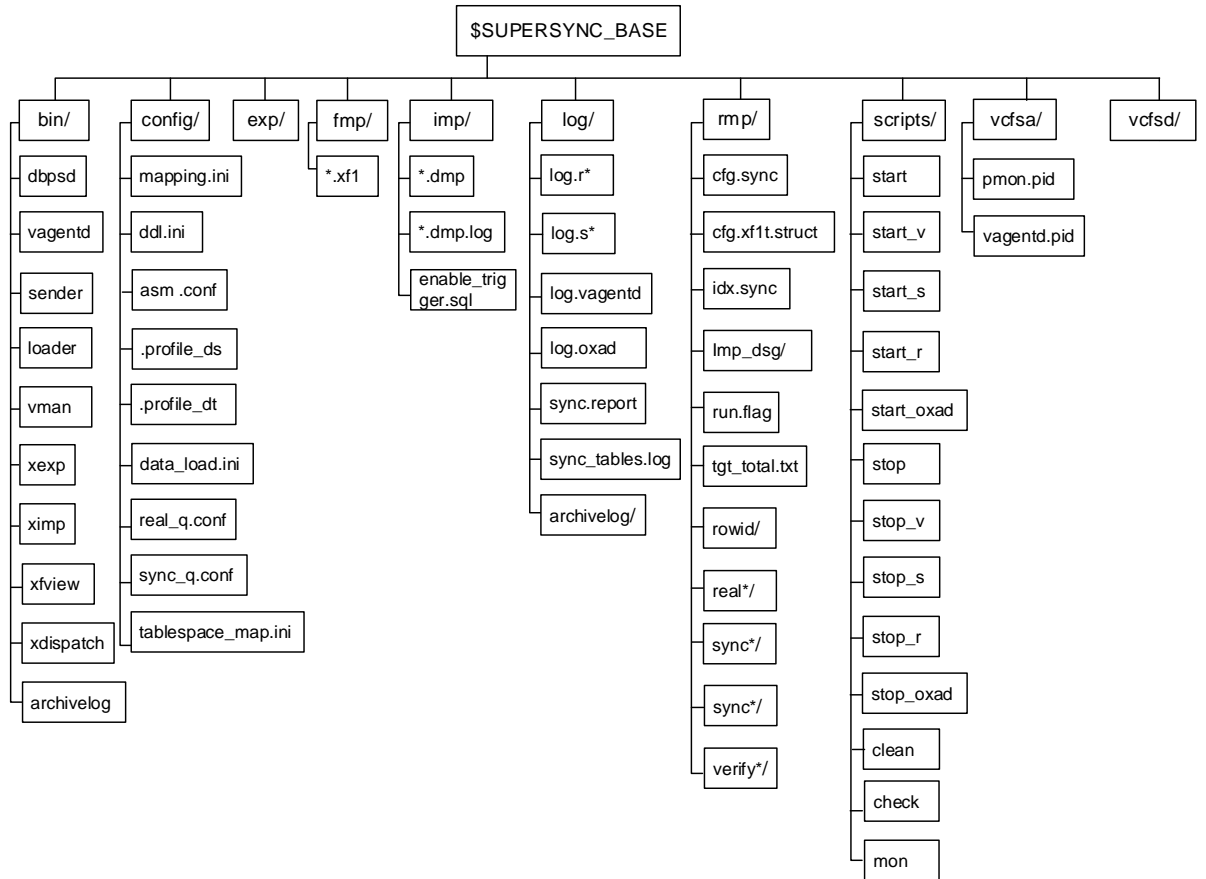
log.sender	源端发送 xf1 文件的运行日志
archivelog 目录	源端存放 log.vagentd、log.sender 的归档日志的目录
rmp 目录：存放 SuperSync 操作映射文件目录	
cfg.ctl	SuperSync 复制操作控制文件
cfg.finishseq	记录最近分析完的日志的 sequence 号的文件
ddl_no_dump.txt	对不需要复制的 DDL 操作的分析记录
ddl_dump.txt	对需要复制的 DDL 操作的分析记录
r0_t0.txt	记录当前分析 Redo log 日志的位置的文件
realwhere.ini	实时复制范围的记录文件
2.cfg.senderno	SuperSync 编号控制文件，记录 vagentd 和 sender 进程的维护信息
rowid 目录	存储源端数据库的 rowid 信息
btab 目录	SuperSync 使用频繁 DDL 方式所需要用到信息的目录
scripts 目录：存放执行脚本目录	
start	启动 SuperSync dbps、vagentd、sender 及日志归档进程脚本
start_v	单独启动 vagentd 进程脚本
start_x	单独启动 sender 进程脚本
start_aoxd	单独启动 aoxd 进程脚本
stop	停止 SuperSync dbps、vagentd、sender 及日志归档进程脚本
stop_v	单独停止 vagentd 进程脚本
stop_x	单独停止 sender 进程脚本
stop_aoxd	单独停止 aoxd 进程脚本
clean	清除缓存目录脚本
check	检查 SuperSync dbps、vagentd、sender 及日志归档进程脚本
mon	系统监控脚本
register	SuperSync 注册程序
vcfsa 目录：存放进程和读取日志状态的目录	

pmon.pid	监控 vagentd 进程号
src_total.txt	源端传送内容文件
uncommitted 目录	未提交缓存目录
vagentd.pid	vagentd 进程号
vcfs_cvf.db	记录当前没有提交的事务的统计（索引）信息
vcfs_dc.db	复制配置信息 db 文件
vcfsd: 存放配置信息目录，该目录只有运行 dbpsd 进程时才使用该目录。	
vcfs_db.db	supersync 数据库配置 db 文件
vcfs_host.db	supersync 主机配置 db 文件
vcfs_user.db	supersync 用户配置 db 文件

## 1.2.2 SuperSync 目标端复制目录结构

SuperSync 目标端复制目录结构如图 3 所示。

图 3 SuperSync 目标端复制目录结构



SuperSync 目标端复制目录的具体说明如表 2 所示。

表 2 目标端 SuperSync 复制目录说明

<b>\$SUPERSYNC_BASE:</b> 为 SuperSync 目标端安装时指定的目录	
<b>bin 目录:</b> SuperSync 执行文件存放目录	
dbpsd	SuperSync Server 程序
vagentd	SuperSync Vagentd 程序
sender	SuperSync xf1 发送程序
loader	SuperSync xf1 装载程序
vman	SuperSync 管理配置工具
xexp	SuperSync 逻辑导出工具
ximp	SuperSync 逻辑导入工具
xfview	xf1 文件查看工具
xdispatch	xf1 任务拆分工具
archive_log	归档日志分析工具
<b>config 目录:</b> SuperSync 配置文件存放目录	

mapping.ini	SuperSync 数据同步配置文件
ddl.ini	SuperSync DDL 过滤配置文件
asm.conf	ASM 配置文件
.profile_ds	SuperSync 源端配置文件
.profile_dt	SuperSync 目标端配置文件
data_load.ini	数据装载配置文件
real_q.conf	实时多并发时控制 vagentd 分发交易的配置文件
sync_q.conf	全同步多并发时控制 vagentd 分发交易的配置文件
tablespace_map.ini	表空间映射配置文件
exp 目录: SuperSync 数据导出目录, 该目录在目标端为空	
fmp 目录: SuperSync 交易缓存目录	
*.xf1	高级双活复制部署中 loader 生成的 rowmapping 映射同步文件
imp 目录: SuperSync 数据导入目录	
*.dmp	SuperSync 导入文件
*.dmp.log	SuperSync 导入文件日志
enable_trigger.sql	trigger 允许修改文件
log 目录: 存放日志的目录	
log.r*	多个进程增量同步的日志。*为数字, 代表第几个装载的日志
log.s*	多个进程首次同步的日志。*为数字, 代表第几个装载的日志
log.vagentd	Vagentd 启动运行日志
sync.report	数据同步报告, 该文件中包含同步的表信息及同步和装载的行数
log.oxad	oxad 启动运行日志
sync_tables.log	表全同步的日志
archivelog 目录	定期归档 log.vagentd、log.r* 日志的目录

rmp 目录：存放 SuperSync 操作映射文件目录	
cfg.sync	从源端接收 xf1 文件个数的文件
cfg.xf1t.struct	从源端接收 xf1 文件的结构
idx.sync	从源端同步的索引在目标端中关联的个数
imp_dsg 目录	该目录下存放着源端数据库的 rowmap 信息及源端数据字典
run.flag	SuperSync 运行阶段文件
tgt_total.txt	接收源端 xf1 件目标端装载的详细记录
rowid 目录	行链接或行迁移表的缓存信息
real*目录	实时同步装载目录，*为数字，表示第几个 loader -r 进程装载的交易
sync*目录	首次同步装载目录，*为数字，表示第几个 loader -s 进程装载的交易
verify 目录	复制本地用于校验的 rmp 目录
scripts 目录：存放执行脚本目录	
start	启动 SuperSync vagentd、loader -s、loader -r 及日志归档进程脚本
start_v	单独启动 vagentd 进程脚本
start_s	单独启动 loader -s 进程脚本
start_r	单独启动 loader -r 进程脚本
start_oxad	单独启动 oxad 进程脚本
stop	停止 SuperSync vagentd、loader -s、loader -r 及日志归档进程脚本
stop_v	单独停止 vagentd 进程脚本
stop_s	单独停止 loader -s 进程脚本
stop_r	单独停止 loader -r 进程脚本
stop_oxad	单独停止 oxad 进程脚本
clean	清除缓存目录脚本
check	检查 SuperSync vagentd、loader -s、loader -r 及日志归档进程脚本
mon	系统监控脚本

vcfsa 目录：存放进程和读取日志状态的目录	
pmon.pid	监控 vagentd 进程号
vagentd.pid	vagentd 进程号
vcfsd：存放配置信息目录，该目录只有运行 dbpsd 进程时才使用该目录。	



## 2 启动/停止 SuperSync 进程

---

介绍 SuperSync 源端和目标端进程启动和停止步骤。

本章节包含下列主题：

- 启动/停止源端进程
- 启动/停止目标端进程
- 单独启动/停止代理进程
- 单独启动/停止源端的发送进程
- 单独启动/停止目标端首次同步装载进程
- 单独启动/停止目标端的实时装载进程

### 2.1 启动/停止源端进程

介绍数据库源端启动和停止 SuperSync 进程步骤。

#### 2.1.1 启动源端进程

介绍启动源端 SuperSync 进程。

##### 前提条件

- 源端服务器已安装 SuperSync 软件，具体操作请参见《SuperSync 安装配置指南》。
- 源端数据库 SuperSync 进程未启动。
- 使用 dsG 用户进行操作。

##### 操作步骤

1. 进入源端服务器 SuperSync 复制队列的 scripts 目录中，启动源端复制进程和 AOXD 进程。

```
$ cd SuperSync 源端复制目录/scripts
```

```
$ ./start
```

```
$ ./start_aoxd
```

2. 正常启动后，系统将显示下列启动进程项。

```

oracle10 21794      1  0 15:39 pts/4    00:00:00
/dsg/supersync/ds/bin/dbpsd 54010

oracle10 21788      1  0 15:39 pts/4    00:00:00 /dsg/supersync/ds/bin/
vagentd 54011

oracle10 21789 21788  1 15:39 pts/4    00:00:00 /dsg/supersync/ds/bin/
vagentd 54011

oracle10 21792      1  0 15:39 pts/4    00:00:00
/dsg/supersync/ds/bin/sender -tseq 1

oracle10 21801 21792  0 15:39 pts/4    00:00:00
/dsg/supersync/ds/bin/sender -tseq 1

oracle10 21795 21757  0 15:39 pts/4    00:00:00
/dsg/supersync/ds/bin/arch_vagentd_ds /dsg/supersync/ds/log/archive/
log.vagentd_archlog 43200 5G

oracle10 21797 21757  0 15:39 pts/4    00:00:00
/dsg/supersync/ds/bin/arch_vagentd_ds /dsg/supersync/ds/log/log.sender
/dsg/supersync/ds/log/archive/ log.sender_archlog 43200 5G

oracle10  18203      1  0 Apr28 ?          00:00:00
/dsg/supersync/ds/bin/aoxd -parfile /dsg/supersync/ds/config/aox.ini

oracle10  18207 18203  0 Apr28 ?          00:00:00
/dsg/supersync/ds/bin/aoxd -parfile /dsg/supersync/ds/config/aox.ini

oracle10  17667      1  0 Apr28 ?          00:00:00
/dsg/supersync/ds/bin/oxad -startup -n 127.0.0.1:7705 -blen 30M

```

### 📖 说明

- oracle10 21794 1 0 15:39 pts/4 00:00:00  
/dsg/supersync/ds/ bin/dbpsd 54010 是 SuperSync 源端的 dbpsd 进程，启动一个是正常的，它的启动日志在 log 下的 log.dbpsd。
- oracle10 21788 1 0 15:39 pts/4 00:00:00  
/dsg/supersync/ds/ bin/ vagentd 54011  
  
oracle10 21789 21788 1 15:39 pts/4 00:00:00  
/dsg/supersync/ds/ bin/ vagentd 54011  
  
是 SuperSync 源端的 vagentd 进程，启动为 2 个一对是正常的，它的启动日志在 log 下的 log.vagentd。如果 vagentd 启动为一个进程，表示不正常，请查看 log.vagentd 日志确定是否复制正常。
- oracle10 21792 1 0 15:39 pts/4 00:00:00  
/dsg/supersync/ds/ bin/ sender -tseq 1  
  
oracle10 21801 21792 0 15:39 pts/4 00:00:00  
/dsg/supersync/ds/ bin/ sender -tseq 1  
  
是 SuperSync 源端的 sender 进程，启动为 2 个一对是正常的，它的启动日志在 log 下的 log.sender。如果 sender 启动进程为一个进程，表示不正常，请查看 log.sender 日志确定是否复制正常。
- oracle10 21795 21757 0 15:39 pts/4 00:00:00  
/dsg/supersync/ds/ bin/arch\_vagentd\_ds  
/dsg/supersync/ds/log/archive/log.vagentd\_ archlog 43200 5G 是

SuperSync 源端的归档 log.vagentd 日志的进程，启动一个为正常。log.vagentd 归档日志在 log/archivelog 目录下，此目录下的日志都是打了时间戳的。

该进程中：43200 表示每 43200 秒对 log.vagentd 日志归档一次，同时启停进程时也会进行归档，并在 log/archivelog 目录的归档文件 log.vagentd\_archlog.xxxxxx\_xxxxxx 后面打上归档时的时间戳；5G 表示 log/archivelog 归档目录的最大空间为 5G，如果该目录下的归档日志文件大于 5G，则删除时间最早的文件，直到所有文件总大小小于 5G 为止。

- oracle10 21797 21757 0 15:39 pts/4 00:00:00  
/dsg/supersync/ds/ bin/arch\_ vagentd\_ds  
/dsg/supersync/ds/log/log.sender /dsg/supersync/ ds/log/archivelog/  
log.sender \_archlog 43200 5G 是 SuperSync 源端的归档 log.sender 日志的进程，启动一个为正常，它的归档日志在 log/archivelog 下，此目录下的日志都是打了时间戳的。

该进程中：43200 表示每 43200 秒对 log.sender 日志归档一次，同时起停进程时也会进行归档，并在 log/archivelog 目录的归档文件 log.sender\_archlog.xxxxxx\_xxxxxx 后面打上归档时的时间戳；5G 表示 log/archivelog 归档目录的最大空间为 5G，如果该目录下的归档日志文件大于 5G，则删除时间最早的文件，直到所有文件总大小小于 5G 为止。

- oracle10 18203 1 0 Apr28 ? 00:00:00  
/dsg/supersync/ds/bin/aoxd -parfile /dsg/supersync/ds/config/aox.ini
- oracle10 18207 18203 0 Apr28 ? 00:00:00  
/dsg/supersync/ds/bin/aoxd -parfile /dsg/supersync/ds/config/aox.ini

是 SuperSync 源端的 aoxd 进程，启动为 2 个一对是正常的，它的启动日志在 log 下的 log.aoxd。如果 aoxd 启动为一个进程，表示不正常，请查看 log.aoxd 日志确定是否复制正常。

- oracle10 17667 1 0 Apr28 ? 00:00:00  
/dsg/supersync/ds/bin/oxad -startup -n 127.0.0.1:7705 -blen 30M

是 SuperSync 源端的 oxad 进程，启动一个是正常的，它的启动日志在 log 下的 log.oxad。

## 2.1.2 停止源端进程

介绍源端 SuperSync 停止进程操作。

### 前提条件

- 源端 SuperSync 进程已启动，具体操作请参见 [2.1.1 启动源端进程](#)。
- 使用 dsg 用户进行操作。

### 操作步骤

1. 进入源端服务器 SuperSync 复制队列的 scripts 目录执行停止源端复制和 AOXD 进程操作。

**\$ cd SuperSync 源端复制目录/scripts**

```
$ ./stop
```

```
$ ./stop_aoxd
```

- 系统将停止 SuperSync 进程，检查进程。

```
$ ./check
```

如果没有显示进程，则说明进程已经停掉。

## 2.2 启动/停止目标端进程

介绍目标端服务器中启动和停止 SuperSync 复制进程的操作方法。

### 2.2.1 启动目标端进程

介绍目标端进程启动操作。

#### 前提条件

- 目标端服务器已安装配置 SuperSync 软件，具体操作请参见《SuperSync 安装配置指南》。
- 目标端 SuperSync 进程未启动。
- 使用 dsg 用户进行操作。

#### 操作步骤

- 目标端为 Oracle 数据库时，目标端启动进程操作：
  1. 进入目标端服务器 SuperSync 复制队列的 scripts 目录，执行启动进程操作。

```
$ cd SuperSync 目标端复制目录/scripts
```

```
$ ./start
```

2. 启动后系统显示下列启动进程项。

```
oracle10 31584      1  0 16:24 pts/2    00:00:00 /dsg/supersync/dt/bin/
vagentd 54021
oracle10 31585 31584  0 16:24 pts/2    00:00:00 /dsg/supersync/dt/bin/
vagentd 54021
oracle10 31596      1  0 16:24 pts/2    00:00:00
/dsg/supersync/dt/bin/loader -r -qno 0 1
oracle10 31597 31596  0 16:24 pts/2    00:00:00
/dsg/supersync/dt/bin/loader -r -qno 0 1
oracle10 31595      1  0 16:24 pts/2    00:00:00
/dsg/supersync/dt/bin/loader -s -qno 0 1
oracle10 31598 31595  0 16:24 pts/2    00:00:00
```

```

/dsg/supersync/dt/bin/loader -s -qno 0 1

oracle10 31580 31556 0 16:24 pts/2 00:00:00
/dsg/supersync/dt/bin/archivelog /dsg/supersync/dt/log/log.vagentd
/dsg/supersync/dt/log/archivelog/log.vagentd_archlog
/dsg/supersync/dt/log/archivelog/log.vagentd_errlog err 43200 5G

oracle10 31590 31556 0 16:24 pts/2 00:00:00 /dsg/supersync/dt/bin/
archivelog /dsg/supersync/dt/log/log.r0 /dsg/supersync/dt/log/archivelog
/log.r0_archlog /dsg/supersync/dt/log/archivelog/log.r0_errlog rowid
43200 5G
    
```

 **说明**

- 1) oracle10 31584 1 0 16:24 pts/2 00:00:00  
/dsg/supersync/dt/bin/vagentd 54021  
  
oracle10 31585 31584 0 16:24 pts/2 00:00:00  
/dsg/supersync/dt/bin/vagentd 54021

是 SuperSync 目标端的 vagentd 进程，启动一对 2 个为正常，它的启动日志在 log 下的 log.vagentd。如果 vagentd 启动为一个进程，表示不正常，请查看 log.vagentd 日志确定是否复制正常。

- 2) oracle10 31596 1 0 16:24 pts/2 00:00:00  
/dsg/supersync/dt/bin/loader -r -qno 0 1  
  
oracle10 31597 31596 0 16:24 pts/2 00:00:00  
/dsg/supersync/dt/bin/loader -r -qno 0 1

是 SuperSync 目标端实时同步装载 loader - r 进程。-qno 0 1 里的 0 表示第几个 loader - r 进程，每个正常启动时有 1 对 2 个 loader - r 进程。

- 3) oracle10 31595 1 0 16:24 pts/2 00:00:00  
/dsg/supersync/dt/bin/loader -s -qno 0 1  
  
oracle10 31598 31595 0 16:24 pts/2 00:00:00  
/dsg/supersync/dt/bin/loader -s -qno 0 1

是 SuperSync 目标端首次同步装载 loader - s 进程。-qno 0 1 里的 0 表示第几个 loader -s 进程，每个正常启动时有 1 对 2 个 loader - s 进程。

- 4) oracle10 31580 31556 0 16:24 pts/2 00:00:00  
/dsg/supersync/dt/bin/archivelog  
/dsg/supersync/dt/log/log.vagentd /dsg/supersync/dt/log/  
archivelog /log.vagentd\_archlog  
/dsg/supersync/dt/log/archivelog/log.vagentd\_errlog err 43200 5G

是 SuperSync 目标端归档 log.vagentd 日志的进程，正常启动为一个，它的归档日志在 log/archivelog 下。此目录下的日志都是打了时间戳的。该进程中：43200 表示每 43200 秒对 log.vagentd 日志归档一次，同时启停进程时也会进行归档，并在 log/archivelog 目录的归档文件 log.vagentd\_archlog.xxxxxx\_xxxxxx 后面打上归档时的时间戳；5G 表示 log/archivelog 归档目录的最大空间为 5G，如果该目录下的归档日志文件大于 5G，则删除时间最早的文件，直到所有文件总大小小于 5G 为止。

```
5) oracle10 31590 31556 0 16:24 pts/2 00:00:00
/dsg/supersync/dt/ bin/ archivelog /dsg/supersync/dt/log/log.r0
/dsg/supersync/dt/log/ archivelog/log.r0_archlog
/dsg/supersync/dt/log/archivelog/log.r0_errlog rowid 43200 5G
```

是 SuperSync 目标端归档 log.r\* 日志的进程，正常启动为一个，其归档日志在 log/archivelog 下，此目录下的日志都是打了时间戳的。该进程中：每 43200 秒对 log. r\* 日志归档一次，同时启停进程时也会进行归档，并在 log/archivelog 目录的归档文件 log.rx\_archlog.xxxxxx\_xxxxxx 后面打上归档时的时间戳；5G 表示 log/archivelog 归档目录的最大空间为 5G，如果该目录下的归档日志文件大于 5G，则删除时间最早的文件，直到所有文件总大小小于 5G 为止。

- 目标端数据库为 MySQL、DBOne 等其他数据库时，目标端进程启动操作：
  1. 进入目标端服务器 SuperSync 复制队列的 scripts 目录，启动目标端复制进程、数据库对应的接口进程\*xad 和数据加载进程 yloader。

**\$ cd SuperSync 目标端复制目录/scripts**

**\$ ./start**

**\$ ./start\_\*xad**

**\$ ./start\_yloader**

#### 说明

其中，目标端不同的数据库对应不同的接口进程\*xad：

- mysql 对应 yxad
- dbone 对应 ixad
- kdb 对应 kxad
- postgresql 对应 pxad
- cdb 对应 cxad
- SAP hana 对应 hxad
- qcubic 对应 qxad

2. 启动后，查看系统已启动的进程项。

**\$ ./check**

```
dsg      1613      1  0 Aug03 ?      00:00:00 /dsg/bin/xagentd
-max_out_xf1 200,25 91001

dsg      1614 1613  0 Aug03 ?      00:00:00 /dsg/bin/xagentd
-max_out_xf1 200,25 91001

dsg      1856      1  0 Aug03 ?      00:00:00 /dsg/bin/yloader -f
/dsg/config/yloader.ini -startup -v0 -v3 -flog /dsg/log/log.yloader

dsg      1857 1856  0 Aug03 ?      00:00:00 /dsg/bin/yloader -f
/dsg/config/yloader.ini -startup -v0 -v3 -flog /dsg/log/log.yloader

dsg      1600      1  0 Aug03 ?      00:00:00 /dsg/bin/archivelog
```

```

/dsg/log/log.vagentd /dsg/log/archivelog/log.vagentd_archlog
/dsg/log/archivelog/log.vagentd_errlog Err err 43200 5G

dsg      1648      1  0 Aug03 ?          00:00:00 /dsg/bin/archivelog
/dsg/log/log.r0 /dsg/log/archivelog/log.r0_archlog
/dsg/log/archivelog/log.r0_errlog rowid ORA- Err err 43200 5G

dsg      1786      1  0 Aug03 ?          00:00:00 /dsg/bin/yxad
-startup -n 127.0.0.1:81000 -v0 -v3 -home /dsg/mysql -flog
/dsg/log/log.yxad
    
```

 **说明**

1) dsg 1613 1 0 Aug03 ? 00:00:00  
/dsg/bin/xagentd -max\_out\_xf1 200,25 91001

dsg 1614 1613 0 Aug03 ? 00:00:00  
/dsg/bin/xagentd -max\_out\_xf1 200,25 91001

是 SuperSync 目标端的 xagentd 进程，启动一对 2 个为正常，它的启动日志在 log 下的 log.xagentd。如果 xagentd 启动为一个进程，表示不正常，请查看 log.xagentd 日志确定是否复制正常。

2) dsg 1856 1 0 Aug03 ? 00:00:00  
/dsg/bin/yloader -f /dsg/config/yloader.ini -startup -v0 -v3 -flog  
/dsg/log/log.yloader

dsg 1857 1856 0 Aug03 ? 00:00:00  
/dsg/bin/yloader -f /dsg/config/yloader.ini -startup -v0 -v3 -flog  
/dsg/log/log.yloader

是 SuperSync 目标端的数据加载 yloader 进程，启动一对 2 个为正常，它的启动日志在 log 下的 log.yloader。如果 yloader 启动为一个进程，表示不正常，请查看 log.yloader 日志确定是否复制正常。

3) dsg 1600 1 0 Aug03 ? 00:00:00  
/dsg/bin/archivelog /dsg/log/log.vagentd  
/dsg/log/archivelog/log.vagentd\_archlog  
/dsg/log/archivelog/log.vagentd\_errlog Err err 43200 5G

是 SuperSync 目标端归档 log.vagentd 日志的进程，正常启动为一个，它的归档日志在 log/archivelog 下。此目录下的日志都是打了时间戳的。该进程中：43200 表示每 43200 秒对 log.vagentd 日志归档一次，同时启停进程时也会进行归档，并在 log/archivelog 目录的归档文件 log.vagentd\_archlog.xxxxxx\_xxxxxx 后面打上归档时的时间戳；5G 表示 log/archivelog 归档目录的最大空间为 5G，如果该目录下的归档日志文件大于 5G，则删除时间最早的文件，直到所有文件总大小小于 5G 为止。

4) dsg 1648 1 0 Aug03 ? 00:00:00  
/dsg/bin/archivelog /dsg/log/log.r0  
/dsg/log/archivelog/log.r0\_archlog  
/dsg/log/archivelog/log.r0\_errlog rowid ORA- Err err 43200 5G

是 SuperSync 目标端归档 log.r\* 日志的进程，正常启动为一个，其归档日志在 log/archivelog 下，此目录下的日志都是打了时间戳的。该进程中：每 43200 秒对 log.r\* 日志归档一次，同时启停进程时也会进行归档，并在 log/archivelog 目录的归档文件 log.rx\_archlog.xxxxxx\_xxxxxx 后面打上归档时的时间戳；5G 表示



log/archivelog 归档目录的最大空间为 5G，如果该目录下的归档日志文件大于 5G，则删除时间最早的文件，直到所有文件总大小小于 5G 为止

```
5) dsg          1786      1  0 Aug03 ?          00:00:00
   /dsg/bin/yxad -startup -n 127.0.0.1:81000 -v0 -v3 -home
   /dsg/mysql -flog /dsg/log/log.yxad
```

是 SuperSync 目标端的数据库对应的接口进程\*xad, 启动一个正常, 它的启动日志在 log 下的 log.\*xad。

## 2.2.2 停止目标端进程

介绍目标端停止 SuperSync 复制进程操作。

### 前提条件

- 目标端已启动 SuperSync 进程，具体操作请参见 [2.2.1 启动目标端进程](#)。
- 使用 dsg 用户进行操作。

### 操作步骤

1. 进入目标端服务器 SuperSync 复制队列的 scripts 目录，执行停止进程操作。

```
$ cd SuperSync 目标端复制目录/scripts
```

```
$ ./stop
```

3. 系统将停止 SuperSync 进程，检查进程。

```
$ ./check
```

如果没有进程显示，则说明进程已经停掉。

## 2.3 单独启动/停止代理进程

用户在源端或目标端服务器中修改了 SuperSync 配置，如果需要应用修改后的配置，可以单独启动 Vagent 代理进程。

### 2.3.1 单独启动/停止源端 Vagentd 进程

介绍源端启动和停止 SuperSync Vagent 进程操作。

#### 前提条件

- 源端 SuperSync Vagent 进程未启动，或源端所有进程均未启动。
- 使用 dsg 用户进行操作。

#### 操作步骤



- 启动源端 Vagent 进程
  1. 进入源端 SuperSync 复制队列的 scripts 目录，执行启动 Vagent 进程操作。

**\$ cd SuperSync 源端复制目录/scripts**

**\$ ./start\_v**

2. 系统启动 Vagent 进程，并显示下列进程信息。

```
oracle10 30536      1  0 10:40 pts/0    00:00:00 /dsg/supersync/ds/bin/
vagentd 54011

oracle10 30543 30536  0 10:40 pts/0    00:00:00 /dsg/supersync/ds/bin/
vagentd 54011
```

#### 说明

SuperSync 源端的 vagentd 进程，启动为 2 个一对是正常的，它的启动日志在 log 下的 log.vagentd。如果 vagentd 启动为一个进程，表示不正常，请查看 log.vagentd 日志确定是否复制正常。

- 停止源端 Vagent 进程
  1. 进入源端 SuperSync 复制队列的 scripts 目录，执行停止 Vagent 进程操作。

**\$ cd SuperSync 源端复制目录/scripts**

**\$ ./stop\_v**

3. 系统将停止 Vagent 进程，检查进程。

**\$ ./check**

如果没有 Vagent 进程显示，则停止成功。

## 2.3.2 单独启动/停止目标端 Vagent 进程

介绍单独启动和停止目标端 SuperSync Vagent 进程。

### 前提条件

- 目标端 SuperSync Vagent 进程未启动，或目标端所有进程均未启动。
- 使用 dsg 用户进行操作。

### 操作步骤

- 启动目标端 SuperSync Vagent 进程
  1. 进入目标端 SuperSync 复制队列的 scripts 目录，执行启动 Vagent 进程操作。

**\$ cd SuperSync 目标端复制目录/scripts**

**\$ ./start\_v**

2. 系统启动 Vagent 进程，并显示下列进程信息。

```
oracle10 4350      1  0 11:02 pts/2    00:00:00 /dsg/supersync/dt/bin/
vagentd 54021

oracle10 4351 4350  0 11:02 pts/2    00:00:00 /dsg/supersync/dt/bin/
vagentd 54021
```

### 说明

SuperSync 目标端的 vagentd 进程，启动为 2 个一对是正常的，它的启动日志在 log 下的 log.vagentd。如果 vagentd 启动为一个进程，表示不正常，请查看 log.vagentd 日志确定是否复制正常。

- 停止目标端 SuperSync Vagent 进程
  1. 进入目标端 SuperSync 复制队列的 scripts 目录，执行停止 Vagent 进程操作。

**\$ cd SuperSync 目标端复制目录/scripts**

**\$ ./stop\_v**

3. 系统将停止 Vagent 进程，检查进程。

**\$ ./check**

如果没有 Vagent 进程显示，则停止成功。

## 2.4 单独启动/停止源端日志分析进程

SuperSync 提供多并发进程分析日志功能，该功能需要用户配置参数后，启动日志分析进程后实现。

### 前提条件

- 已配置并发日志分析进程文件 aox.ini，具体操作请参见《SuperSync 安装指南》的“配置 AOXD”章节。
- 使用 dsg 用户进行操作。

### 操作步骤

- 启动日志分析进程
  1. 进入源端 SuperSync 复制队列的 scripts 目录，执行启动并发日志分析进程操作。

**\$ cd SuperSync 源端复制目录/scripts**

**\$ ./start\_aoxd**

2. 系统启动 aoxd 和 oxad 进程，并显示下列进程信息。

```
oracle10 18203      1  0 Apr28 ?          00:00:00
/dsg/supersync/ds/bin/aoxd -parfile /dsg/supersync/ds/config/aox.ini

oracle10 18207 18203  0  Apr28 ?          00:00:00
/dsg/supersync/ds/bin/aoxd -parfile /dsg/supersync/ds/config/aox.ini
```

```
oracle10 17667 1 0 Apr28 ? 00:00:00
/dsg/supersync/ds/bin/oxad -startup -n 127.0.0.1:7705 -blen 30M
```

### 说明

SuperSync 启动 aoxd 和 oxad 进程，aoxd 进程启动 2 个，oxad 进程启动 1 个，启动日志在 log 下的 log.aoxd 和 log.oxad。如果 aoxd 启动 1 个进程，表示不正常，请查看 log.aoxd 日志确定是否复制正常。

- 停止日志分析进程
  1. 进入源端 SuperSync 复制队列的 scripts 目录，执行停止并发日志分析进程操作。

**\$ cd SuperSync 源端复制目录/scripts**

**\$ ./stop\_aoxd**

2. 系统将停止 aoxd 和 oxad 进程，检查进程。

**\$ ./check**

如果没有 aoxd 和 oxad 进程，则停止成功。

## 2.5 单独启动/停止源端发送进程

如果用户不需要启动或停止 SuperSync 所有进程，仅需要对源端的发送进程（即 sender 进程）进行启动或停止，则可以根据本节内容对发送进程进行操作。

### 前提条件

- 源端 Vagent 进程已启动，具体操作请参见 [2.3.1 单独启动/停止源端 Vagent 进程](#)。
- 使用 dsg 用户进行操作。

### 操作步骤

- 启动源端 SuperSync sender 进程
  1. 进入源端 SuperSync 复制队列的 scripts 目录，执行启动 sender 进程操作。

**\$ cd SuperSync 源端复制目录/scripts**

**\$ ./start\_x**

2. 系统启动 sender 进程，并显示启动项信息。

```
oracle10 9072 1 0 11:28 pts/2 00:00:00 /dsg/supersync/ds/bin/
vagentd 54021
oracle10 9073 9072 0 11:28 pts/2 00:00:00
/dsg/supersync/ds/bin/ vagentd 54021
oracle10 7115 1 0 11:28 pts/0 00:00:00 /dsg/supersync/ds/bin/
sender -tseq 1
oracle10 7116 7115 0 11:28 pts/0 00:00:00 /dsg/supersync/ds/bin/
```

```
sender -tseq 1
```

### 📖 说明

- oracle10 9072 1 0 11:28 pts/2 00:00:00  
/dsg/supersync/ds/bin/vagentd 54021
- oracle10 9073 9072 0 11:28 pts/2 00:00:00  
/dsg/supersync/ds/bin/vagentd 54021 是 SuperSync 源端已启动的 Vagent 进程，正常启动为 2 个一对。
- oracle10 7115 1 0 11:28 pts/0 00:00:00  
/dsg/supersync/ds/bin/sender -tseq 1
- oracle10 7116 7115 0 11:28 pts/0 00:00:00  
/dsg/supersync/ds/bin/sender -tseq 1

是 SuperSync 源端的 sender 进程，启动为 2 个一对是正常的，它的启动日志在 log 下的 log.sender。如果 sender 启动进程为一个进程，表示不正常，请查看 log.sender 日志确定是否复制正常。

- 停止源端 SuperSync sender 进程
  1. 进入源端 SuperSync 复制队列的 scripts 目录，执行停止 sender 进程操作。

```
$ cd SuperSync 源端复制目录/scripts
```

```
$ ./stop_x
```

3. 系统将停止 sender 进程，检查进程。

```
$ ./check
```

如果没有 sender 进程显示，则停止成功。

## 2.6 单独启动/停止目标端首次同步装载进程

在 SuperSync 首次全同步时，如果数据复制发生错误或其他操作错误时，用户可以单独停止目标端全同步装载进程，对错误进行处理后，再单独启动全同步装载进程，完成数据首次全同步。

### 前提条件

- 目标端 Vagent 进程已启动，具体操作请参见 [2.3.2 单独启动/停止目标端 Vagent 进程](#)。
- 使用 dsg 用户进行操作。

### 操作步骤

- 启动目标端 loader -s 首次全同步装载进程
  1. 进入目标端服务器 SuperSync 复制队列的 scripts 目录，执行启动进程操作。

```
$ cd SuperSync 目标端复制目录/scripts
```

```
$ ./start_s
```

2. 启动后系统显示下列启动进程项。

```
oracle10 9072      1  0 11:28 pts/2    00:00:00 /dsg/supersync/dt/bin/
vagentd 54021

oracle10 9073 9072  0 11:28 pts/2    00:00:00 /dsg/supersync/dt/bin/
vagentd 54021

oracle10 9132      1  0 11:28 pts/2    00:00:00 /dsg/supersync/dt/bin/
loader -s -qno 0 1

oracle10 9133 9132  2 11:28 pts/2    00:00:00 /dsg/supersync/dt/bin/
loader -s -qno 0 1
```

### 说明

- oracle10 9072 1 0 11:28 pts/2 00:00:00 /dsg/supersync/dt/bin/vagentd 54021  
oracle10 9073 9072 0 11:28 pts/2 00:00:00 /dsg/supersync/dt/bin/vagentd 54021 是 SuperSync 的已启动的 Vagent 进程，正常启动为 2 个一对。
- oracle10 9132 1 0 11:28 pts/2 00:00:00 /dsg/supersync/dt/bin/loader -s -qno 0 1  
oracle10 9133 9132 2 11:28 pts/2 00:00:00 /dsg/supersync/dt/bin/loader -s -qno 0 1  
是 SuperSync 目标端首次同步装载 loader -s 进程。-qno 0 1 里的 0 表示第几个 loader -s 进程，每个正常启动时有 1 对 2 个 loader -s 进程。

- 停止目标端 loader -s 首次全同步装载进程
  1. 进入目标端服务器 SuperSync 复制队列的 scripts 目录，执行停止进程操作。

**\$ cd SuperSync 目标端复制目录/scripts**

**\$ ./stop\_s**

3. 系统将停止 loader -s 进程，检查进程。

**\$ ./check**

如果没有 loader -s 进程显示，则停止成功。

## 2.7 单独启动/停止目标端实时装载进程

在 SuperSync 增量同步过程中，如果数据实时复制发生错误或其他操作错误时，用户可以单独停止目标端实时装载进程，对错误进行处理后，在单独启动实时装载进程，完成数据增量同步。

### 前提条件

- 目标端 Vagent 进程已启动，具体操作请参见 [2.3.2 单独启动/停止目标端 Vagent 进程](#)。

- 使用 dsg 用户进行操作。

## 操作步骤

- 启动目标端 loader -r 实时同步装载进程
  1. 进入目标端服务器 SuperSync 复制队列的 scripts 目录, 执行启动进程操作。

**\$ cd SuperSync 目标端复制目录/scripts**

**\$ ./start\_r**

2. 启动后系统显示下列启动进程项。

```
oracle10 9072      1  0 11:28 pts/2    00:00:00 /dsg/supersync/dt/bin/
vagentd 54021

oracle10 9073 9072  0 11:28 pts/2    00:00:00 /dsg/supersync/dt/bin/
vagentd 54021

oracle10 8184      1  0 14:03 pts/0    00:00:00 /dsg/supersync/dt/bin/
loader -r -qno 0 1

oracle10 8185 8184  0 14:03 pts/0    00:00:00 /dsg/supersync/dt/bin/
loader -r -qno 0 1
```

### 说明

- oracle10 9072 1 0 11:28 pts/2 00:00:00 /dsg/supersync/dt/bin/vagentd 54021  
oracle10 9073 9072 0 11:28 pts/2 00:00:00 /dsg/supersync/dt/bin/vagentd 54021 是 SuperSync 的已启动的 Vagent 进程, 正常启动为 2 个一对。

- oracle10 8184 1 0 14:03 pts/0 00:00:00 /dsg/supersync/dt/bin/loader -r -qno 0 1  
oracle10 8185 8184 0 14:03 pts/0 00:00:00 /dsg/supersync/dt/bin/loader -r -qno 0 1

是 SuperSync 目标端实时同步装载 loader - r 进程。-qno 0 1 里的 0 表示第几个 loader - r 进程, 每个正常启动时有 1 对 2 个 loader - r 进程。

- 停止目标端 loader -r 实时同步装载进程
  1. 进入目标端服务器 SuperSync 复制队列的 scripts 目录, 执行停止进程操作。

**\$ cd SuperSync 目标端复制目录/scripts**

**\$ ./stop\_r**

3. 系统将停止 loader -r 进程, 检查进程。

**\$ ./check**

如果没有 loader -r 进程显示, 则停止成功。

## 3 首次初始化

用户在使用 SuperSync 进行数据复制前,可根据自己需求对 SuperSync 进行初始化,确定首次全同步的数据,初始化后, SuperSync 将根据配置完成数据复制。

### 3.1 Oracle 数据库间首次全同步

介绍源端 Oracle 数据库到目标端 Oracle 数据库的全同步操作。

#### 前提条件

- 源端和目标端服务器已安装配置 SuperSync 软件,具体操作请参见《SuperSync 安装配置指南》。
- 数据库正常运行。
- 使用 dsG 用户进行操作。

#### 操作步骤

1. 在 SuperSync 安装目录中,修改源端 start 启动脚本。

```
$ cd SuperSync 源端复制目录/scripts
```

```
$ vi start
```

2. 在脚本中添加配置参数。

- 如果 Oracle 版本为 8i~11g,则请添加下列参数:

```
export AOX_HOST=127.0.0.1
```

```
export AOX_PORT=79911
```

- 如果 Oracle 版本为 12c,则请根据实际复制情况添加下列参数:

```
export AOX_HOST=127.0.0.1
```

```
export AOX_PORT=79911
```

```
export ORA_PDB=db12pdb1
```

```
export ORA_CDB=1
```

#### 说明

- 1) AOX\_HOST: 为 AOXD 所在的服务器 IP 地址,默认为 127.0.0.1。
- 2) AOX\_PORT: 为 AOXD 端口号。

- 3) **ORA\_PDB**: 需要复制的 Oracle 12c 的 PDB 名称。如果将源端中的 PDB 复制到目标端中的其他 PDB 中, 则在目标端 start 脚本中必须配置 ORA\_PDB 参数, 参数值为相应的 PDB 名称。

如将源端名称为 db12pdb1 的 PDB 复制到目标端名称为 db12pdb3 的 PDB 中, 则:

源端 start 脚本中的 PDB 配置	目标端 start 脚本中的 PDB 配置
export ORA_PDB=db12pdb1	export ORA_PDB=db12pdb3

- 4) **ORA\_CDB**: 需要复制的 Oracle 12c 的 CDB 序号, 默认为 1,2。
- 如果将源端的 CDB 复制到目标端中的其他 CDB 中, 则在目标端 start 脚本中必须配置 ORA\_CDB 参数, 参数值为相应的 CDB 序号。

如将源端序号为 1 的 CDB 复制到目标端序号为 2 的 CDB 中, 则:

源端 start 脚本中的 CDB 配置	目标端 start 脚本中的 CDB 配置
export ORA_CDB=1	export ORA_CDB=2

- 如果将源端的 CDB 复制到目标端中的 PDB 中, 则在目标端 start 脚本中必须配置 ORA\_PDB 参数, 参数值为相应的 PDB 名称。

如将源端序号为 1 的 CDB 复制到目标端名称为 db12pdb3 的 PDB 中, 则:

源端 start 脚本中的 CDB 配置	目标端 start 脚本中的 PDB 配置
export ORA_CDB=1	export ORA_PDB=db12pdb3

3. 启动源端复制进程和 AOXD 进程。

```
$ ./start
```

```
$ ./start_aoxd
```

4. 进入目标端 SuperSync 安装目录中。

```
$ cd SuperSync 目标端复制目录/scripts
```

5. 如果目标端中 Oracle 版本为 12c, 则需要修改 start 脚本, 添加复制参数。

```
$ vi start
```

根据实际复制情况添加下列参数:

```
export ORA_PDB=db12pdb1
```

```
export ORA_CDB=1
```



 **说明**

- 1) ORA\_PDB: 需要复制到目标端 Oracle 12c 的 PDB 名称。
  - 2) ORA\_CDB: 需要复制到目标端 Oracle 12c 的 CDB 序号, 默认为 1,2。
6. 启动目标端复制进程。

**\$ ./start**

7. 进入源端 config 目录下, 编辑 mapping.ini。

**\$cd SuperSync 源端复制目录/config**

**\$vi mapping.ini**

8. 请根据用户的实际复制需求在 mapping.ini 文件中添加复制条件, 例如:

```
DSG DSG
real_where=
        u.name in ('DSG')
full_where=
        u.name in ('DSG')
```

 **说明**

- DSG、DSG 分别为源端和目标端用户名。
  - real\_where=u.name in('DSG')表示 SuperSync 实时复制的源端对象为 DSG, 将 DSG 用户下的所有数据实时复制到目标端。如果源端 Oracle 为 12c, 则 DSG 必须为要复制的 CDB 或 PDB 下的用户。
  - full\_where=u.name in('DSG')表示 SuperSync 全同步复制的源端对象为 DSG, 将 DSG 用后下的所有数据全同步到目标端。如果源端 Oracle 为 12c, 则 DSG 必须为要复制的 CDB 或 PDB 下的用户。
  - 数据必须进行全同步后, 才能进行增量实时复制, 所以首次初始化时, 建议 full\_where 和 real\_where 值相同, 或者也可以根据用户的具体需求赋值, 但必须保证 real\_where 的赋值范围小于 full\_where。
9. 查看 VM 脚本。

**\$cd SuperSync 源端复制目录/bin**

**\$cat vm**

10. 系统将显示 vm 脚本内容。

例如显示:

```
connect localhost:50000
user root/dbps
#set ds 1 -ctf -itvl 3 -state start
set ds 1 -itvl 5 -state start
```

```
set dt 1.1 -tgt db 2 -xf1 y -state start
#set dm 1.1 BILL.TABLE SAM.TABLE -sync ftcq -rule r
set dm 1.1 -cfg -sync ftcqIM1
```

#### 说明

- 50000 是复制源端的 DBPS 端口，请检查是否与安装时设置的一致。
- set ds、set dt、set dm 请根据需要进行配置，命令具体使用方法请查看 help。

11. 在源端发起首次全同步。

```
$/vman
```

```
VMAN> @vm
```

#### 说明

- 可以通过 log 目录下的 log.vagentd 文件查看全同步进度。  
**\$ grep Summ log.vagentd**
- 首次全同步之后，SuperSync 可自动进行增量同步。

## 3.2 异种数据库间首次全同步

介绍源端 Oracle 数据库到 MySQL、Sybase、DB2、DBOne 等其他数据库的全同步操作。

### 前提条件

- 源端和目标端服务器已安装配置 SuperSync 软件，具体操作请参见《SuperSync 安装配置指南》。
- 数据库正常运行。
- 使用 dsg 用户进行操作。

### 操作步骤

1. 进入源端 SuperSync 安装目录/scripts 中，修改源端 start 启动脚本。

```
$ cd SuperSync 源端安装目录/scripts
```

```
$ vi start
```

2. 在脚本中添加配置参数。

- 如果 Oracle 版本为 8i~11g，则请添加下列参数：

```
export AOX_HOST=127.0.0.1
```

```
export AOX_PORT=79911
```

- 如果 Oracle 版本为 12c，则请根据实际复制情况添加下列参数：

```
export AOX_HOST=127.0.0.1
export AOX_PORT=79911
export ORA_PDB=db12pdb1
export ORA_CDB=1
```

 **说明**

- 1) AOX\_HOST: 为 AOXD 所在的服务器 IP 地址, 默认为 127.0.0.1。
  - 2) AOX\_PORT: 为 AOXD 端口号。
  - 3) ORA\_PDB: 需要复制的 Oracle 12c 的 PDB 名称。
  - 4) ORA\_CDB: 需要复制的 Oracle 12c 的 CDB 序号, 默认为 1,2。
3. 启动源端复制进程和 AOXD 进程。

```
$ ./start
```

```
$ ./start_aoxd
```

4. 进入目标端 SuperSync 安装目录/scripts 中, 启动目标端复制进程、数据库对应的接口进程\*xad 和数据加载进程 yloader。

```
$ cd SuperSync 目标端安装目录/scripts
```

```
$ ./start
```

```
$ ./start_*xad
```

```
$ ./start_yloader
```

 **说明**

其中, 目标端不同的数据库对应不同的接口进程\*xad:

- MySQL 对应 yxad
- DBOne 对应 ixad
- K-DB 对应 kxad
- PostgreSQL 对应 pxad
- CDB 对应 cxad
- SAP HANA 对应 hxad
- Qcubic 对应 qxad

5. 进入源端安装路径下, 编辑 mapping.ini。

```
$cd SuperSync 源端安装目录/config
```

```
$vi mapping.ini
```

6. 请根据用户的实际复制需求在 mapping.ini 文件中添加复制条件。例如:

**DSG DSG**

```
real_where=
    u.name in ('DSG')
full_where=
    u.name in ('DSG')
```

#### 说明

- DSG、DSG 分别为源端和目标端用户名。
- real\_where=u.name in('DSG')表示 SuperSync 实时复制的源端对象为 DSG, 将 DSG 用户下的所有数据实时复制到目标端。如果源端 Oracle 为 12c, 则 DSG 必须为要复制的 CDB 或 PDB 下的用户。
- full\_where=u.name in('DSG')表示 SuperSync 全同步复制的源端对象为 DSG, 将 DSG 用后下的所有数据全同步到目标端。如果源端 Oracle 为 12c, 则 DSG 必须为要复制的 CDB 或 PDB 下的用户。
- 数据必须进行全同步后, 才能进行增量实时复制, 所以首次初始化时, 建议 full\_where 和 real\_where 值相同, 或者也可以根据用户的具体需求赋值, 但必须保证 real\_where 的赋值范围小于 full\_where。

7. 查看 VM 脚本。

```
$cd SuperSync 源端安装目录/bin
```

```
$cat vm
```

8. 系统将显示 vm 脚本内容。

本例中将显示:

```
connect localhost:7700
user root/dbps
#set ds 1 -ctf -itvl 3 -state start
set ds 1 -itvl 5 -state start
set dt 1.1 -tgt db 2 -xf1 y -state start
#set dm 1.1 BILL.TABLE SAM.TABLE -sync ftcq -rule r
set dm 1.1 -cfg -sync ftcqIM1 -rule r
#exit
```

#### 说明

- 7700 是复制源端的 DBPS 端口, 请检查是否与安装时设置的一致。
- set ds、set dt、set dm 请根据需要进行配置, 命令具体使用方法请查看 help。

9. 在源端发起首次全同步。

```
$/vman
```

**VMAN> @vm**

 **说明**

- 可以通过 log 目录下的 log.vagentd 文件查看全同步进度。

**\$ grep Summ log.vagentd**

- 首次全同步之后，SuperSync 可自动进行增量同步。

## 4 配置复制

介绍 SuperSync 对 Oracle 数据表配置复制的操作方法。

本章节包含下列主题：

- 单独同步数据表
- 取消实时同步单张表
- 确认同步表

### 4.1 单独同步数据表

在数据复制过程中，如果一个或多个数据表未复制到目标端或复制到目标端的数据出现错误不一致时，可以通过单独同步表的方式，将数据表再次复制到目标端。

#### 4.1.1 单独同步单张表

在数据复制过程中，某张表由于同步到目标端的数据不准确或由于目标端无法装载而未同步该表时，需要把此表的数据单独进行同步以保证数据一致。

##### 前提条件

- SuperSync 已完成首次初始化，具体操作请参见 [3 首次初始化](#)。
- 源端和目标端 SuperSync 进程均启动，且运行正常。
- 使用 dsg 用户进行操作。

##### 操作步骤

1. 进入 SuperSync 源端复制任务的 config 目录下，编辑 mapping.ini 文件。

```
$ cd SuperSync 源端复制目录/config
```

```
$ vi mapping.ini
```

2. 在 mapping.ini 文件的 full\_where 参数中增加需要同步的数据表信息，例如同步的数据表名称是 dsg 用户下的 test 表，则编辑后的 mapping.ini 内容为：

```
DSG DSG  
real_where=  
u.name in ('DSG')
```

```
full_where=
u.name in ('DSG') and o.name in ('TEST')
```

 说明

- DSG、DSG 分别为源端和目标端用户名。
- real\_where=u.name in ('DSG')表示 SuperSync 实时复制的源端对象为 DSG，将 DSG 用户下的所有数据实时复制到目标端。
- full\_where=u.name in ('DSG') and o.name in ('TEST')表示 SuperSync 同步复制 DSG 用户下的 test 表。

3. 退出并保存 mapping.ini 文件。

```
$ :wq!
```

4. 为保证新复制的表不存在之前的交易中，请在 SuperSync 目标端 config 目录中编辑 data\_load.ini 文件，修改参数 del\_real\_data 值为 yes。

```
$ cd SuperSync 目标端复制目录/config
```

```
$ vi data_load.ini
```

```
del_real_data = yes
```

5. 退出并保存 data\_load.ini 文件。

```
$ :wq!
```

6. 在 SuperSync 源端 bin 目录中，通过 vm 脚本，创建 one.vm 脚本。

```
$cd SuperSync 源端复制目录/bin
```

```
$cp vm one.vm
```

7. 编辑 one.vm 脚本。

```
$ vi one.vm
```

8. 修改 one.vm 脚本，修改后的脚本内容如：

```
connect localhost:50000
user root/dbps
#set ds 1 -ctf -itvl 3 -state start
set ds 1 -itvl 5 -state start
set dt 1.1 -tgt db 2 -xf1 y -state start
#set dm 1.1 BILL.TABLE SAM.TABLE -sync ftcqI -rule r
set dm 1.1 -cfg -sync ftcqI
```

 说明

只需要修改最后一行 `set dm 1.1 -cfg -sync ftcql`，修改后的 `ftcql` 不加 `M` 的并发数。

9. 在源端发起单表同步。

```
$/vman
```

```
VMAN> @one.vm
```

#### 说明

同步过程中，可以查看下列日志和文件，掌握数据同步进程：

- 查看源端 `log` 目录下的 `log.vagentd` 日志，查看全同步进度。
- 查看目标端 `log` 目录下的 `log.s*` 日志，查看目标端首次同步装载情况。
- 查看目标端 `rmp` 目录下的 `cfg.sync` 文件，查看目标端首次装载还剩下的文件总数。
- 查看目标端目录下的 `log.r*` 日志，查看目标端实时同步装载情况。

## 4.1.2 单独同步多张表

在数据复制过程中，某些表由于同步到目标端的数据不准确或由于目标端无法装载而未同步这些表时，需要对这些表的数据进行单独同步以保证数据一致。

### 前提条件

- SuperSync 已完成首次初始化，具体操作请参见 [3 首次初始化](#)。
- 源端和目标端 SuperSync 进程均启动，且运行正常。
- 使用 `dsg` 用户进行操作。

### 操作步骤

1. 进入 SuperSync 源端复制任务的 `config` 目录下，编辑 `mapping.ini` 文件。

```
$ cd SuperSync 源端复制目录/config
```

```
$ vi mapping.ini
```

2. 在 `mapping.ini` 文件的 `full_where` 参数中增加需要同步的数据表信息，例如同步的数据表名称是 `dsg` 用户下的 `test_%` 表，则编辑后的 `mapping.ini` 内容为：

```
DSG DSG  
real_where=  
u.name in ('DSG')  
full_where=  
u.name||'|' ||o.name in (select upper(u||'|' ||t) from dsg.full_table)
```

#### 说明



- DSG、DSG 分别为源端和目标端用户名。
- `real_where=u.name in('DSG')`表示 SuperSync 实时复制的源端对象为 DSG，将 DSG 用户下的所有数据实时复制到目标端。
- `full_where= u.name||'.'||o.name in (select upper(u)||'.'||t) from dsg.full_table)`表示 SuperSync 同步复制 `dsg.full_table` 表。

3. 退出并保存 `mapping.ini` 文件。

```
$ :wq!
```

4. 为保证新复制的表不存在之前的交易中，请在 SuperSync 目标端 `config` 目录中编辑 `data_load.ini` 文件，修改参数 `del_real_data` 值为 `yes`。

```
$ cd SuperSync 目标端复制目录/config
```

```
$ vi data_load.ini
```

```
del_real_data = yes
```

5. 退出并保存 `data_load.ini` 文件。

```
$ :wq!
```

6. 用 `dsg` 用户登陆到源端数据库上建立 `dsg.full_table` 表。

```
$ sqlplus dsg/dsg
```

```
SQL> create table dsg.full_table (u varchar2(20),t  
varchar2(30));
```

7. 将需要复制的多个表插入到 `dsg.full_table` 表中，并保存。例如需要复制 DSG 用户下的 `test_%`表包含：`test`、`test1`、`test2`，则需要将这三张表分别插入到 `dsg.full_table` 中：

```
SQL> insert into dsg.full_table values('DSG','TEST');
```

```
SQL> insert into dsg.full_table values('DSG','TEST1');
```

```
SQL> insert into dsg.full_table values('DSG','TEST2');
```

```
SQL> commit;
```

8. 在源端 SuperSync 复制任务的 `bin` 目录中发起同步。

```
$ cd SuperSync 源端复制目录/bin
```

```
$ ./vman
```

```
VMAN> @vm
```

### 说明

同步过程中，可以查看下列日志和文件，掌握数据同步进程：

- 查看源端 `log` 目录下的 `log.vagentd` 日志，查看全同步进度。
- 查看目标端 `log` 目录下的 `log.s*` 日志，查看目标端首次同步装载情况。
- 查看目标端 `rpm` 目录下的 `cfg.sync` 文件，查看目标端首次装载还剩的文件总数。

- 查看目标端目录下的 log.r\* 日志，查看目标端实时同步装载情况。

### 4.1.3 单独同步新增的多张表

首次同步完成后，可以新增实时同步某些表。

#### 前提条件

- SuperSync 已完成首次初始化，具体操作请参见 [3 首次初始化](#)。
- 源端和目标端 SuperSync 进程均启动，且运行正常。
- 使用 dsq 用户进行操作。

#### 操作步骤

1. 进入 SuperSync 源端复制任务的 config 目录下，编辑 mapping.ini 文件。

```
$ cd SuperSync 源端复制目录/config
```

```
$ vi mapping.ini
```

2. 在 mapping.ini 文件的 full\_where 和 real\_where 参数中都增加需要同步的数据表信息，例如同步的数据表名称是 dsq 用户下的 test\_% 表，则编辑后的 mapping.ini 内容为：

```
DSG DSG
real_where=
u.name||'|'||o.name in (select upper(u)||'|t) from dsq.real_table)
full_where=
u.name||'|'||o.name in (select upper(u)||'|t) from dsq.full_table)
```

#### 说明

- DSG、DSG 分别为源端和目标端用户名。
  - real\_where=u.name||'|'||o.name in (select upper(u)||'|t) from dsq.real\_table) 表示 SuperSync 实时同步复制 dsq.real\_table 表。
  - full\_where= u.name||'|'||o.name in (select upper(u)||'|t) from dsq.full\_table) 表示 SuperSync 同步复制 dsq.full\_table 表。
  - 由于是新增复制的表，所以必须即改动 full\_where 让其先初始化，还要改动 real\_where 让其实时变化的东西也抓取到。如果只改动 full\_where，不把这些表加入到 real\_where 里，则这些表的实时变化将无法抓取到。
3. 退出并保存 mapping.ini 文件。
- ```
$ :wq!
```
4. 为保证新复制的表不存在之前的交易中，请在 SuperSync 目标端 config 目录中编辑 data\_load.ini 文件，修改参数 del\_real\_data 值为 yes。

```
$ cd SuperSync 目标端复制目录/config
```

```
$ vi data_load.ini
```

```
del_real_data = yes
```

- 退出并保存 data\_load.ini 文件。

```
$ :wq!
```

- 用 dsg 用户登陆到源端数据库上建立 dsg.full\_table 表和 dsg.real\_table 表。

```
$ sqlplus dsg/dsg
```

```
SQL> create table dsg.full_table (u varchar2(20),t  
varchar2(30) );
```

```
SQL> create table dsg.real_table (u varchar2(20),t  
varchar2(30) );
```

- 将需要复制的多个表插入到 dsg.full\_table 表中，并保存。例如需要复制的 DSG 用户下的 test\_%表包含：test、test1、test2，则需要将这三张表分别插入到 dsg.full\_table 中：

```
SQL> insert into dsg.full_table values('DSG','TEST');
```

```
SQL> insert into dsg.full_table values('DSG','TEST1');
```

```
SQL> insert into dsg.full_table values('DSG','TEST2');
```

```
SQL> commit;
```

- 将步骤 7 中插入到 dsg.full\_table 中的表再插入到 dsg.real\_table 表中，并保存。例如：

```
SQL> insert into dsg.real_table values('DSG','TEST');
```

```
SQL> insert into dsg.real_table values('DSG','TEST1');
```

```
SQL> insert into dsg.real_table values('DSG','TEST2');
```

```
SQL> commit;
```

- 在源端 SuperSync 复制任务的 bin 目录中发起同步。

```
$ cd SuperSync 源端复制目录/bin
```

```
$ ./vman
```

```
VMAN> @vm
```

### 说明

同步过程中，可以查看下列日志和文件，掌握数据同步进程：

- 查看源端 log 目录下的 log.vagentd 日志，查看全同步进度。
- 查看目标端 log 目录下的 log.s\*日志，查看目标端首次同步装载情况。
- 查看目标端 rmp 目录下的 cfg.sync 文件，查看目标端首次装载还剩下的文件总数。
- 查看目标端目录下的 log.r\*日志，查看目标端实时同步装载情况。

## 4.2 取消实时同步单张表

当 SuperSync 已经同步了某些表，但用户不想再实时抓取其中的某张表时，可以通过单张表不同步功能实现。

### 前提条件

- SuperSync 已完成首次初始化或数据表同步，具体操作请参见 [3 首次初始化](#)或 [4.1 单独同步数据表](#)。
- 源端和目标端 SuperSync 进程均启动，且运行正常。
- 使用 dsg 用户进行操作。

### 操作步骤

1. 进入 SuperSync 源端复制任务的 config 目录下，编辑 mapping.ini 文件。

```
$ cd SuperSync 源端复制目录/config
```

```
$ vi mapping.ini
```

2. 在 mapping.ini 文件中，对 full\_where 和 real\_where 参数进行修改。

- a. 假设 SuperSync 已经全同步 dsg 用户下的 test\_%表，并实时抓取这些表，具体操作请参见 [4.1.3 单独同步新增的多张表](#)，mapping.ini 文件内容如下：

```
DSG DSG
real_where=
u.name||'|'||o.name in (select upper(u||'|'||t) from dsg.real_table)
full_where=
u.name||'|'||o.name in (select upper(u||'|'||t) from dsg.full_table)
```

- b. 如果希望不实时同步 dsg 用户下的 test 表，则需要先修改 full\_where 参数，在 u.name||'|'||o.name in (select upper(u||'|'||t) from dsg.full\_table)前添加#，使全同步操作失效。
- c. 退出并保存 mapping.ini 文件。

```
$ :wq!
```

3. 用 dsg 用户登陆到源端数据库上，并删除 DSG 用户下的 test 表。

```
$ sqlplus dsg/dsg
```

```
SQL> delete from dsg.real_table where u='DSG' and t='TEST';
```

```
SQL>commit;
```

4. 进入 SuperSync 源端复制任务的 bin 目录下，发起同步。

```
$ cd SuperSync 源端复制目录/bin
```

```
$ ./vman
```

```
VMAN> @vm
```

5. SuperSync 同步过程中将不再实时同步 DSG 用户下的 test 表。

## 4.3 确认同步表

用户可以通过确认同步表方式查看数据库将要全同步或实时同步的表，保证同步的准确性。

### 前提条件

- SuperSync 已完成首次初始化或数据表同步，具体操作请参见 [3 首次初始化](#)或 [4.1 单独同步数据表](#)。
- 源端和目标端 SuperSync 进程均启动，且运行正常。
- 使用 dsg 用户进行操作。

### 操作步骤

1. 进入 SuperSync 源端复制任务的 config 目录下，查看 mapping.ini 文件。

```
$ cd SuperSync 源端复制目录/config
```

```
$ vi mapping.ini
```

2. 用 dsg 用户登陆到源端数据库上，并输入 sql 语句。

```
$ sqlplus dsg/dsg
```

```
SQL> select to_char(t.obj#), u.name, o.name
```

```
from sys.user$ u,sys.ts$ ts,sys.tab$ t,sys.obj$ o,
```

```
sys.seg$ s,dba_tables dt,dba_objects dob
```

```
where o.owner# = u.user#
```

```
and o.obj# = t.obj# and s.file#(+) = t.file#
```

```
and s.block#(+) = t.block# and s.ts#(+) = t.ts#
```

```
and bitand(t.property, 1) = 0
```

```
and t.ts# = ts.ts#
```

```
and o.name not like 'MLOG$%'
```

```
and o.name not like 'RUPD$%'
```

```
and o.name not like 'BIN$%'
```

```
and o.name not like 'SYS_IOT%'
```

```
and bitand(t.property, 192) = 0
```

```
and bitand(t.property, 512) = 0
```

```
and dt.OWNER = u.NAME and dt.TABLE_NAME = o.NAME
```

```
and dt.TEMPORARY='N'
```

```

and dob.OWNER = u.NAME and dob.OBJECT_NAME
=o.NAME and dob.OBJECT_TYPE='TABLE'

and (u.name, o.name) not in (select sowner, tname from
sys.snap$)

and (查询条件)

order by o.obj#;

```

 说明

“查询条件”请根据实际情况进行修改：

- 如果需要查询全同步的表，则请在此处修改为 mapping.ini 文件中 full\_where=后面的参数值；
- 如果需要查询实时同步的表，则请在此处修改为 mapping.ini 文件中 real\_where=后面的参数值。

3. 系统将显示查询的同步表信息，如：

| TO_CHAR(T.OBJ#)         | NAME |
|-------------------------|------|
| 280528                  | DSG  |
| STD_CFM_JOUR            |      |
| 280587                  | DSG  |
| STD_COUNTER_FUND_DETAIL |      |

 说明

- 280528 和 280587 是要同步的对象编号；
- STD\_CFM\_JOUR 和 STD\_COUNTER\_FUND\_DETAIL 为将要全同步或实时同步的表名称；
- DSG 为复制的用户名称。

## 5 打开源端/目标端 DSG 备份文件

介绍如何打开源端和目标端的 DSG 备份文件。

本章节包含下列主题：

- 打开源端 vagentd 备份文件
- 打开源端 sender 备份文件
- 打开目标端 loader -s 备份文件
- 打开目标端 loader -r 备份文件

### 5.1 打开源端 vagentd 备份文件

介绍打开源端 vagentd 备份文件操作。

#### 前提条件

- SuperSync 已完成首次初始化或数据表同步，具体操作请参见 [3 首次初始化](#)或 [4.1 单独同步数据表](#)。
- 源端和目标端 SuperSync 进程均启动，且运行正常。
- 使用 dsq 用户进行操作。

#### 操作步骤

1. 进入 SuperSync 源端 scripts 目录下，编辑 start 文件。

```
$ cd SuperSync 源端复制目录/scripts
```

```
$ vi start
```

2. 在 start 文件中找到含 vagentd 和端口行：

```
$DBPS_HOME/bin/vagentd $LOCAL_VAGENTD_PORT >> $DBPS_HOME/log/log.vagentd 2>&1 &
```

3. 在该行中添加-xf1back 参数：

```
$DBPS_HOME/bin/vagentd $LOCAL_VAGENTD_PORT -xf1back >> $DBPS_HOME/log/log.vagentd 2>&1 &
```

4. 保存并退出。

```
$ :wq!
```

5. 停止 SuperSync 进程后，再启动 SuperSync 进程，使修改后的 start 文件生效。

```
$ ./stop
```

```
$ ./start
```

6. 系统将显示启动的进程项，启动后的 vagentd 进程项都带有-xf1back。

```
oracle10 19408      1  0 13:54 pts/16   00:00:00 /dsg/supersync/ds/bin/
vagentd 54011 -xf1back
oracle10 19409 19408  1 13:54 pts/16   00:00:00 /dsg/supersync/ds/bin/
vagentd 54011 -xf1back
```

7. 到源端 rmp 目录中查看备份生成的文件。

```
$ cd SuperSync 源端复制目录/rmp
```

```
$ ls
```

系统将显示如：seq1.925.xdt、seq1.925.xf1、seq1.925.log 文件。

## 5.2 打开源端 sender 备份文件

介绍打开源端 vagentd 备份文件操作。

### 前提条件

- SuperSync 已完成首次初始化或数据表同步，具体操作请参见 [3 首次初始化](#)或 [4.1 单独同步数据表](#)。
- 源端和目标端 SuperSync 进程均启动，且运行正常。
- 使用 dsg 用户进行操作。

### 操作步骤

1. 进入 SuperSync 源端 scripts 目录下，编辑 start 文件。

```
$ cd SuperSync 源端复制目录/scripts
```

```
$ vi start
```

2. 在 start 文件中找到含 sender 行：

```
$DBPS_HOME/bin/sender -tseq 1 >> $DBPS_HOME/log/log.sender 2>&1 &
```

3. 在该行中添加-xf1back 参数：

```
$DBPS_HOME/bin/sender -tseq 1 -xf1back >> $DBPS_HOME/log/log.sender
2>&1 &
```

4. 保存并退出。

```
$ :wq!
```



5. 停止 SuperSync 进程后，再启动 SuperSync 进程，使修改后的 start 文件生效。

```
$ ./stop
```

```
$ ./start
```

6. 系统将显示启动的进程项，启动后的 sender 进程项都带有-xf1back。

```
oracle10 21502      1  0 14:38 pts/17   00:00:00
/dsg/supersync/ds/bin/sender -tseq 1 -xf1back

oracle10 21503 21502  1 14:38 pts/17   00:00:00
/dsg/supersync/ds/bin/sender -tseq 1 -xf1back
```

7. 到源端 rmp 目录中查看备份生成的文件。

```
$ cd SuperSync 源端复制目录/rmp
```

```
$ ls
```

系统将显示如：2.165.xf1.bak、2.165.xf1.snd\_filter0\_ after、2.165.xf1.snd\_filter0\_ before 文件。其中：2.165.xf1.snd\_filter0\_ after 是经过 sender 发送后的文件，2.165.xf1.snd\_filter0\_ before 是未经过 sender 发送前的文件。

## 5.3 打开目标端 loader -s 备份文件

介绍打开目标端 loader -s 备份文件操作。

### 前提条件

- SuperSync 已完成首次初始化或数据表同步，具体操作请参见 [3 首次初始化](#)或 [4.1 单独同步数据表](#)。
- 源端和目标端 SuperSync 进程均启动，且运行正常。
- 使用 dsg 用户进行操作。

### 操作步骤

1. 进入 SuperSync 目标端 scripts 目录下，编辑 start 文件。

```
$ cd SuperSync 目标端复制目录/scripts
```

```
$ vi start
```

2. 在 start 文件中找到含 loader -s 所有行：

```
$DBPS_HOME/bin/loader -s -qno 0 1 >> $DBPS_HOME/log/log.s0 2>&1 &
```

### 说明

如果有多个 loader -s 进程，则 start 文件中有相应的 loader -s 行，本例中仅以 1 个 loader -s 进程为例进行介绍，如果有多个进程，请对所有 loader -s 进行操作。

- 在 loader -s 所有行中添加-xf1back 参数:

```
$DBPS_HOME/bin/loader -s -qno 0 1 -xf1back >> $DBPS_HOME/log/log.s0
2>&1 &
```

- 保存并退出。

```
$ :wq!
```

- 停止 SuperSync 进程后, 再启动 SuperSync 进程, 使修改后的 start 文件生效。

```
$ ./stop
```

```
$ ./start
```

- 系统将显示启动的进程项, 启动后的 loader -s 进程项都带有-xf1back。

```
oracle10 24870      1  0 14:57 pts/17   00:00:00
/dsg/supersync/dt/bin/loader -s -qno 0 1 -xf1back

oracle10 24871 24870  0 14:57 pts/17   00:00:00
/dsg/supersync/dt/bin/loader -s -qno 0 1 -xf1back
```

- 到目标端 rmp 目录中 sync0 中查看备份生成的文件。

```
$ cd SuperSync 目标端复制目录/rmp/sync0
```

```
$ ls
```

系统将显示.xf1.bak 文件。

## 5.4 打开目标端 loader -r 备份文件

介绍打开目标端 loader -r 备份文件操作。

### 前提条件

- SuperSync 已完成首次初始化或数据表同步, 具体操作请参见 [3 首次初始化](#)或 [4.1 单独同步数据表](#)。
- 源端和目标端 SuperSync 进程均启动, 且运行正常。
- 使用 dsg 用户进行操作。

### 操作步骤

- 进入 SuperSync 目标端 scripts 目录下, 编辑 start 文件。

```
$ cd SuperSync 目标端复制目录/scripts
```

```
$ vi start
```

- 在 start 文件中找到含 loader -r 所有行:

```
$DBPS_HOME/bin/loader -r -qno 0 1 >> $DBPS_HOME/log/log.r0 2>&1 &
```

 **说明**

如果有多个 loader -r 进程，则 start 文件中有相应的 loader -r 行，本例中仅以 1 个 loader -r 进程为例进行介绍，如果有多个进程，请对所有 loader -r 进行操作。

3. 在 loader -r 所有行中添加-xf1back 参数：

```
$DBPS_HOME/bin/loader -r -qno 0 1 -xf1back >> $DBPS_HOME/log/log.r0  
2>&1 &
```

4. 保存并退出。

```
$ :wq!
```

5. 停止 SuperSync 进程后，再启动 SuperSync 进程，使修改后的 start 文件生效。

```
$ ./stop
```

```
$ ./start
```

6. 系统将显示启动的进程项，启动后的 loader -r 进程项都带有-xf1back。

```
oracle10 14173      1  0 10:44 pts/11    00:00:00  
/dsg/supersync/dt/bin/loader -r -qno 0 1 -xf1back  
oracle10 14174 14173  2 10:44 pts/11    00:05:54  
/dsg/supersync/dt/bin/loader -r -qno 0 1 -xf1back
```

7. 到目标端 rmp 目录中 real0 中查看备份生成的文件。

```
$ cd SuperSync 目标端复制目录/rmp/real0
```

```
$ ls
```

系统将显示.xf1.bak 文件。

## 6 修改 vman

用户可以根据需要修改 SuperSync vman 中的主机 IP 和 DSG 用户密码等信息。

本章节包含下列主题：

- 修改 vman 中的主机 IP
- 修改 vman 中的数据库 DSG 用户密码

### 6.1 修改 vman 中的主机 IP

当部署了多个复制目标端时，如果用户需要对目标端服务器进行切换时，可以通过 SuperSync vman 工具修改目标端的主机 IP 地址，已完成目标端切换。

#### 前提条件

使用 dsg 用户进行操作。

#### 操作步骤

1. 进入源端服务器 SuperSync 复制队列的 scripts 目录中，查看源端进程是否运行正常。

```
$ cd SuperSync 源端复制目录/scripts
```

```
$ ./check
```



显示的启动项信息是否正常，请根据 [2.1.1 启动源端进程](#) 中的描述进行检查。

2. 进入 SuperSync 源端复制队列的 bin 目录中，调用 vman 程序。

```
$ cd SuperSync 源端复制目录/bin
```

```
$ ./vman
```

3. 连接源端主程序 DBPS。

```
VMAN> connect : DBPS 端口
```

例如：connect : 50000

4. 输入进入 dbps 主程序的用户名和密码。

```
DBPS> user 用户名/密码
```



SuperSync 的 dbps 主程序的用户名默认为“root”，密码默认为“dbps”。

5. 列出 dbps 主程序的功能菜单。

**SYNC:/> menu**

6. 系统将显示 dbps 功能项，请按照红色字体输入。

```

1. System
2. Scheduler
s. List current sessions
q. Return to command mode
===> 1
    1. User
    2. Host
    4. Database
    q. quit
===> 2
    1. List all
    2. List one in detail
    3. Add
    4. Delete
    5. Edit
    6. Detect and update hardware/os_type/vagentd
    q. quit
===> 5
Host Name: 192.168.1.154
Host Name: [192.168.1.154] 192.168.1.156
    Port: [54011]
    Enabled: [Y]
    HostID: [2831194625]
    Hardware: []
    OS Type: []
    Vagentd: []
Host [192.168.1.56] modified
    1. List all
    2. List one in detail
    3. Add

```

```

4. Delete
5. Edit
6. Detect and update hardware/os_type/vagentd
q. quit
===> 1
  Hid   Port Enabl  Host Name   Host ID     Hardware
  OS Type vagentd
-----
1 : 54011 Y           localhost 2831194625
2 : 54021 Y    192.168.1.156 2831194625
Press return to continue ...
1. List all
2. List one in detail
3. Add
4. Delete
5. Edit
6. Detect and update hardware/os_type/vagentd
q. quit
===> q
1. User
2. Host
4. Database
q. quit
===> q
1. System
2. Scheduler
s. List current sessions
q. Return to command mode
===> q
SYNC:/> exit

```

### 说明

- 第一个 Host Name 输入原始目标端主机 IP，本例为 192.168.1.154。
- 第二个 HostName 输入需要切换的目标端主机 IP 地址，本例中输入的 IP 地址为 192.168.1.156。

- Port、Enabled、HostID、Hardware、OS Type、Vagentd 参数请根据实际需要修改，如果没有变更，请直接回车即可。
  - 选择 1. List all 后，系统显示修改后的源端和目标端信息，用户可以通过查看这些信息，确认主机 IP 是否已修改成功。
7. 进入源端服务器 SuperSync 复制队列的 config 目录中，编辑 mapping.ini 文件。

```
$ cd SuperSync 源端复制目录/config
```

```
$ vi mapping
```

8. 根据复制需要，修改 mapping 文件。

#### 说明

根据复制需求的不同，mapping.ini 文件的具体修改方法，请参见 [3.首次初始化](#) 和 [4.1 单独同步数据表](#)。

9. 进入源端 bin 目录，执行 ./vman，使配置生效。

```
$ cd SuperSync 源端复制目录/bin
```

```
$ ./vman
```

```
VMAN>@vm
```

## 6.2 修改 vman 中的数据库 DSG 用户密码

用户可以通过 vman 程序修改数据库的 DSG 用户和密码。

### 前提条件

使用 dsg 用户进行操作。

### 操作步骤

1. 进入源端服务器 SuperSync 复制队列的 scripts 目录中，查看源端进程是否运行正常。

```
$ cd SuperSync 源端复制目录/scripts
```

```
$ ./check
```

#### 说明

显示的启动项信息是否正常，请根据 [2.1.1 启动源端进程](#) 中的描述进行检查。

2. 进入 SuperSync 源端复制队列的 bin 目录中，调用 vman 程序。

```
$ cd SuperSync 源端复制目录/bin
```

```
$ ./vman
```

3. 连接源端主程序 DBPS。

```
VMAN> connect : DBPS 端口
```

例如：connect : 50000

4. 输入进入 dbps 主程序的用户名和密码。

**DBPS> user** 用户名/密码

 **说明**

SuperSync 的 dbps 主程序的用户名默认为“root”，密码默认为“dbps”。

5. 列出 dbps 主程序的功能菜单。

**SYNC:/> menu**

6. 系统将显示 dbps 功能项，请按照红色字体输入。

```

1. System
2. Scheduler
s. List current sessions
q. Return to command mode
===> 1
    1. User
    2. Host
    4. Database
    q. quit
===> 4
        1. List all
        2. List one in detail
        3. Add
        4. Delete
        5. Edit
        q. quit
===> 5
        DBID: 1
Host Name: [localhost]
DB name: [dsg]
DB user: [dsg]
DB pwd: [*****] dsg
OS Owner: []
OS Group: []
SID: [dsg]
TNS: []
DB_HOME: []

```



```

TNS_ADMIN: []
DB [dsg] modified
    1. List all
    2. List one in detail
    3. Add
    4. Delete
    5. Edit
    q. quit
    ==> 1
DB Type  DB User      Hostname:DB Home
oracle   dsg localhost: dsg.1
oracle   dsg 192.168.1.156: dsg.2
Press return to continue ...
    1. List all
    2. List one in detail
    3. Add
    4. Delete
    5. Edit
    q. quit
    ==> q
    1. User
    2. Host
    4. Database
    q. quit
    ==> q
    1. System
    2. Scheduler
    s. List current sessions
    q. Return to command mode
    ==> q
SYNC:/> exit

```

 说明

- DBID 是数据库编号，1 表示源端，2 表示目标端，请根据需要输入。本例为源端数据库。

- DB pwd 为数据库的 dsg 用户密码，用户根据需要修改。本例中修改后的密码为 dsg。
  - 其他参数信息建议保持默认。
  - 选择 1. List all 后，系统显示源端和目标端修改后的数据库配置信息。
7. 进入源端服务器 SuperSync 复制队列的 config 目录中，编辑 mapping.ini 文件。

```
$ cd SuperSync 源端复制目录/config
```

```
$ vi mapping
```

8. 根据复制需要，修改 mapping 文件。

#### 说明

根据复制需求的不同，mapping.ini 文件的具体修改方法，请参见 [3.首次初始化](#) 和 [4.1 单独同步数据表](#)。

9. 进入源端 bin 目录，执行 ./vman，使配置生效。

```
$ cd SuperSync 源端复制目录/bin
```

```
$ ./vman
```

```
VMAN>@vm
```

# 7 SuperSync 装载分发

SuperSync 支持多路并发装载，用户可以在复制过程中增加全同步装载和实时同步装载进程，将表分发到多路装载进程中实现并发装载。

本章节包含下列主题：

- 全同步装载分发
- 实时同步装载分发

## 7.1 全同步装载分发

在首次全同步过程中，用户可以增加全同步装载进程，并将需要复制的表分配到装载进程中，实现 SuperSync 多路并发装载。

### 前提条件

- 源端和目标端 SuperSync 进程均启动，且运行正常。
- 使用 dsq 用户进行操作。

### 操作步骤

1. 进入 SuperSync 目标端复制任务的 scripts 目录下，停止目标端进程，并清理目标端缓存。

```
$ cd SuperSync 目标端复制目录/scripts
```

```
$ ./stop
```

```
$ ./clean
```

2. 进入目标端 config 目录下，编辑 SuperSync 配置文件.profile\_dt。

```
$ cd SuperSync 目标端复制目录/config
```

```
$ vi .profile_dt
```

3. 修改.profile\_dt 文件中的 LOCAL\_LOADER\_S\_PARALLEL 参数，增加多个装载进程。

例如：如果需要 3 个并发进程用于全同步装载，则请将 LOCAL\_LOADER\_S\_PARALLEL 后的参数值修改为 3：

```
LOCAL_LOADER_S_PARALLEL=3
```

### 说明

.profile\_dt 文件修改完成并保存后，可以通过以下操作查看设置是否成功：

- a) 在 script 目录下执行启动 SuperSync 进程，查看启动的进程中是否有 3 个 loader -s 进程：

**\$ cd SuperSync 目标端复制目录/scripts**

**\$ ./start**

- b) 如果进程中 loader -s 显示如下，则表示 3 条 loader -s 进程设置成功。

```
oracle10 12583      1  1 10:45 pts/2    00:00:00
/dsg/supersync/dt/ bin/loader -s -qno 0 1

oracle10 12595 12583  0 10:45 pts/2    00:00:00
/dsg/supersync/dt/ bin/loader -s -qno 0 1

oracle10 12584      1  0 10:45 pts/2    00:00:00
/dsg/supersync/dt/ bin/loader -s -qno 1 1

oracle10 12597 12584  0 10:45 pts/2    00:00:00
/dsg/supersync/dt/ bin/loader -s -qno 1 1

oracle10 12586      1  0 10:45 pts/2    00:00:00
/dsg/supersync/dt/ bin/loader -s -qno 2 1

oracle10 12587 12586  0 10:45 pts/2    00:00:00
/dsg/supersync/dt/ bin/loader -s -qno 2 1
```

- c) 停止 SuperSync 进程。

**\$ ./stop**

4. 在目标端 config 目录下，创建 sync\_q.conf 文件。

**\$ cd SuperSync 目标端复制目录/config**

**\$ vi sync\_q.conf**

5. 在 sync\_q.conf 文件中添加需要分发装载的表。

例如：将 dsg 用户下的表 test1、test2、test3 由全同步装载进程 0、1、2 进行分发，则请在 sync\_q.conf 文件中添加下列内容。

```
0 DSG:TEST1
1 DSG:TEST2
2 DSG:TEST3
```

#### 说明

- 0 DSG:TEST1 表示 dsg 用户下的表 test1 分发给 loader -s -qno 0 1 进程。
- 1 DSG:TEST2 表示 dsg 用户下的表 test2 分发给 loader -s -qno 1 1 进程。
- 2 DSG:TEST3 表示 dsg 用户下的表 test3 分发给 loader -s -qno 2 1 进程。
- 其他没有定义的表将默认分发给 loader -s -qno 0 1 进程。

- 如果某张表有多个分区，用户也可以对这张表的多个分区进行分发，只需要在 `sync_q.conf` 文件中的表后添加分区即可，如对表 TEST1 的分区进行分发：

```
0 DSG:TEST1: P_101
```

```
1 DSG:TEST1: P_111
```

```
2 DSG:TEST1: P_121
```

6. 保存并退出。

```
$ :wq!
```

7. 进入目标端 `scripts` 目录下，启动 SuperSync 进程。

```
$ cd SuperSync 目标端复制目录/scripts
```

```
$ ./start
```



说明  
用户可以在目标端 `log` 目录下查看 `log.vagentd` 和 `log.s*` 文件，保证 SuperSync 启动正常。

8. 在源端服务器中进行全同步操作，具体操作请参见 [3 首次初始化](#) 中源端的操作步骤。



说明  
全同步过程中，SuperSync 将对表的全同步进行并发装载。

## 7.2 实时同步装载分发

SuperSync 支持实时同步装载分发，用户可以将需要同步的表通过并发进程快速复制装载到目标端。

### 7.2.1 首次同步过程中分发实时装载

在首次全同步过程中，用户可以增加实时装载进程，并将需要复制的表分配到装载进程中，实现 SuperSync 多路并发装载。

#### 前提条件

- 源端和目标端 SuperSync 进程均启动，且运行正常。
- 使用 `dsg` 用户进行操作。

#### 操作步骤

1. 进入 SuperSync 目标端复制任务的 `scripts` 目录下，停止目标端进程，并清理目标端缓存。

```
$ cd SuperSync 目标端复制目录/scripts
```

```
$ ./stop
```

**\$ ./clean**

2. 进入目标端 config 目录下，编辑 SuperSync 配置文件.profile\_dt。

**\$ cd SuperSync 目标端复制目录/config**

**\$ vi .profile\_dt**

3. 修改.profile\_dt 文件中的 LOCAL\_LOADER\_R\_PARALLEL 参数，增加多个装载进程。

例如：如果需要 3 个并发进程用于实时装载，则请将 LOCAL\_LOADER\_R\_PARALLEL 后的参数值修改为 3:

```
LOCAL_LOADER_R_PARALLEL=3
```

### 说明

.profile\_dt 文件修改完成并保存后，可以通过以下操作查看设置是否成功：

- a) 在 script 目录下执行启动 SuperSync 进程，查看启动的进程中是否有 3 个 loader -r 进程：

**\$ cd SuperSync 目标端复制目录/scripts**

**\$ ./start**

- b) 如果进程中 loader -r 显示如下，则表示 3 条 loader -r 进程设置成功。

```
oracle10 17628      1  0 11:19 pts/20    00:00:00
/dsg/supersync/dt/ bin/loader -r -qno 0 1

oracle10 17629 17628  0 11:19 pts/20    00:00:00
/dsg/supersync/dt/ bin/loader -r -qno 0 1

oracle10 17622      1  0 11:19 pts/20    00:00:00
/dsg/supersync/dt/ bin/loader -r -qno 1 1

oracle10 17623 17622  0 11:19 pts/20    00:00:00
/dsg/supersync/dt/ bin/loader -r -qno 1 1

oracle10 17638      1  0 11:19 pts/20    00:00:00
/dsg/supersync/dt/ bin/loader -r -qno 2 1

oracle10 17639 17638  0 11:19 pts/20    00:00:00
/dsg/supersync/dt/ bin/loader -r -qno 2 1
```

- c) 停止 SuperSync 进程。

**\$ ./stop**

4. 在目标端 config 目录下，创建 real\_q.conf 文件。

**\$ cd SuperSync 目标端复制目录/config**

**\$ vi real\_q.conf**

5. 在 real\_q.conf 文件中添加需要分发装载的表。

例如：将 dsg 用户下的表 test1、test2、test3 由实时装载进程 0、1、2 进行分发，则请在 real\_q.conf 文件中添加下列内容。

```
0 DSG:TEST1
1 DSG:TEST2
2 DSG:TEST3
```

#### 说明

- 0 DSG:TEST1 表示 dsg 用户下的表 test1 分发给 loader -r -qno 0 1 进程。
- 1 DSG:TEST2 表示 dsg 用户下的表 test2 分发给 loader -r -qno 1 1 进程。
- 2 DSG:TEST3 表示 dsg 用户下的表 test3 分发给 loader -r -qno 2 1 进程。
- 其他没有定义的表将默认分发给 loader -r -qno 0 1 进程。
- 如果某张表有多个分区，用户也可以对这张表的多个分区进行分发，只需要在 real\_q.conf 文件中的表后添加分区即可，如对表 TEST1 的分区进行分发：

```
0 DSG:TEST1: P_101
1 DSG:TEST1: P_111
2 DSG:TEST1: P_121
```

6. 保存并退出。

```
$ :wq!
```

7. 进入目标端 scripts 目录下，启动 SuperSync 进程。

```
$ cd SuperSync 目标端复制目录/scripts
```

```
$ ./start
```

#### 说明

用户可以在目标端 log 目录下查看 log.vagentd 和 log.s\* 文件，保证 SuperSync 启动正常。

8. 在源端服务器中进行全同步操作，具体操作请参见 [3 首次初始化](#) 中源端的操作步骤。

#### 说明

全同步过程中，SuperSync 将实现对表的实时并发装载。

## 7.2.2 实时同步过程中分发实时装载

在数据实时同步中，SuperSync 也可以对复制进行实时多路分发装载。

### 前提条件

- 源端和目标端 SuperSync 进程均启动，且运行正常。

- 使用 dsg 用户进行操作。

## 操作步骤

1. 进入 SuperSync 目标端复制任务的 config 目录下，编辑配置文件 .profile\_dt。

```
$ cd SuperSync 目标端复制目录/config
```

```
$ vi .profile_dt
```

2. 修改 .profile\_dt 文件中的 LOCAL\_LOADER\_R\_PARALLEL 参数，增加多个装载进程。

例如：如果需要 3 个并发进程用于实时装载，则请将 LOCAL\_LOADER\_R\_PARALLEL 后的参数值修改为 3：

```
LOCAL_LOADER_R_PARALLEL=3
```

### 说明

.profile\_dt 文件修改完成并保存后，可以通过以下操作查看设置是否成功：

- a) 在 script 目录下执行启动 SuperSync 进程，查看启动的进程中是否有 3 个 loader -r 进程：

```
$ cd SuperSync 目标端复制目录/scripts
```

```
$ ./start
```

- b) 如果进程中 loader -r 显示如下，则表示 3 条 loader -r 进程设置成功。

```
oracle10 17628      1  0 11:19 pts/20    00:00:00
/dsg/supersync/dt/ bin/loader -r -qno 0 1

oracle10 17629 17628  0 11:19 pts/20    00:00:00
/dsg/supersync/dt/ bin/loader -r -qno 0 1

oracle10 17622      1  0 11:19 pts/20    00:00:00
/dsg/supersync/dt/ bin/loader -r -qno 1 1

oracle10 17623 17622  0 11:19 pts/20    00:00:00
/dsg/supersync/dt/ bin/loader -r -qno 1 1

oracle10 17638      1  0 11:19 pts/20    00:00:00
/dsg/supersync/dt/ bin/loader -r -qno 2 1

oracle10 17639 17638  0 11:19 pts/20    00:00:00
/dsg/supersync/dt/ bin/loader -r -qno 2 1
```

3. 用 dsg 用户登陆到源端数据库。

```
$ sqlplus dsg 用户名/密码
```

4. 执行 sql 语句，生成 cfg.objs 接口文件。

- a、如果复制的表没有分区，则请输入下列 sql 语句：

```
set heading off
```

```
set pagesize 50000
```

```
set echo off
```



```

select o1.obj# as objn,
       o.obj# as objp,
       o.dataobj# as objd,
       u.name||'.'||o.name||'.'||o.subname as OBJ_NAME
from sys.user$ u, sys.obj$ o, sys.obj$ o1, dba_tables dt,
dba_objects dob
where o.owner# = u.user#
     and o1.owner# = u.user#
     and o1.name=o.name
     and u.name not in ('SYS','SYSTEM')
     and o.type# in ( 2, 19, 34, 39, 40 )
     and o.dataobj# is not null
     and o1.type# = 2
     and dt.OWNER = u.NAME and dt.TABLE_NAME = o.NAME
and dt.TEMPORARY = 'N'
     and dob.OWNER = u.NAME and dob.OBJECT_NAME =
o.NAME and dob.OBJECT_TYPE = 'TABLE'
and (用户条件) order by objn,objp;

```

b、如果复制的表有分区表或子分区表，则执行下列 sql 语句：

- 对子分区表进行操作：

```

select tcp.bo# as objn,
       o.obj# as objp,
       o.dataobj# as objd,

u.name||'.'||op.name||'.'||op.subname||'.'||o.subnam
e as OBJ_NAME

from sys.user$ u, sys.obj$ o, sys.obj$ op,
sys.tabsubpart$ tsp, sys.tabcompart$ tcp
where u.user#=o.owner# --and u.user#=op.owner#
     and tsp.obj#=o.obj#    -- subpartition objn/objd
     and tsp.pobj#=tcp.obj# -- subpartition --> partition
     and tsp.pobj#=op.obj#
     and (用户条件)order by objn,objp;

```

- 对分区表进行操作：

```

--partition
select t.bo# as objn,

```

```

o.obj# as objp,
o.dataobj# as objd,
u.name||'.'||o.name||'.'||o.subname as
OBJ_NAME
from sys.user$ u, sys.obj$ o, sys.tabpart$ t
where u.user#=o.owner# and t.obj#=o.obj#
and (用户条件)order by objn,objp;

```

- 对分区表和子分区表同时进行操作：

```

select * from (
select t.bo# as objn,--partition
o.obj# as objp,
o.dataobj# as objd,
u.name||'.'||o.name||'.'||o.subname as
OBJ_NAME
from sys.user$ u, sys.obj$ o, sys.tabpart$ t
where u.user#=o.owner# and t.obj#=o.obj#
and (用户条件)
union
select tcp.bo# as objn, -- subpartition
o.obj# as objp,
o.dataobj# as objd,
u.name||'.'||op.name||'.'||op.subname||'.'||o.subnam
e as OBJ_NAME
from sys.user$ u, sys.obj$ o, sys.obj$ op,
sys.tabsubpart$ tsp, sys.tabcompart$ tcp
where u.user#=o.owner#
and tsp.obj#=o.obj# -- subpartition objn/objd
and tsp.pobj#=tcp.obj# -- subpartition --> partitioin
and tsp.pobj#=op.obj#
and (用户条件)
)
order by objn,objp;

```

#### 说明

“用户条件”请修改成 mapping.ini 文件中 real\_where=后面的参数值，mapping.ini 在 SuperSync 源端的 config 目录下。

5. 系统生成接口文件，格式如：

```

          OBJN      OBJP      OBJD
-----
OBJ_NAME
-----
          280528      280529      280529
DSG.STD_CFM_JOUR.MKT_HA

          280528      280530      280530
DSG.STD_CFM_JOUR.MKT_SA

          280528      280531      280531
DSG.STD_CFM_JOUR.MKT_HB

          280528      280532      280532
DSG.STD_CFM_JOUR.MKT_SB

```

6. 进入 SuperSync 目标端 config 目录下，创建 real\_q.conf 文件。

```
$ cd SuperSync 目标端复制目录/config
```

```
$ vi real_q.conf
```

7. 在 real\_q.conf 文件中添加需要分发装载的表。

例如：将 dsg 用户下的表 test1、test2、test3 由实时装载进程 0、1、2 进行分发，则请在 real\_q.conf 文件中添加下列内容。

```

0 DSG:TEST1
1 DSG:TEST2
2 DSG:TEST3

```

#### 说明

- 0 DSG:TEST1 表示 dsg 用户下的表 test1 分发给 loader -r -qno 0 1 进程。
- 1 DSG:TEST2 表示 dsg 用户下的表 test2 分发给 loader -r -qno 1 1 进程。
- 2 DSG:TEST3 表示 dsg 用户下的表 test3 分发给 loader -r -qno 2 1 进程。
- 其他没有定义的表将默认分发给 loader -r -qno 0 1 进程。

- 如果某张表有多个分区，用户也可以对这张表的多个分区进行分发，只需要在 `real_q.conf` 文件中的表后添加分区即可，如对表 TEST1 的分区进行分发：

```
0 DSG:TEST1: P_101
```

```
1 DSG:TEST1: P_111
```

```
2 DSG:TEST1: P_121
```

8. 进入 SuperSync 源端 `scripts` 目录中，停止源端进程。

```
$ cd SuperSync 源端复制目录/scripts
```

```
$ ./stop
```

9. 进入 SuperSync 目标端 `scripts` 目录中，停止目标端进程。

```
$ cd SuperSync 目标端复制目录/scripts
```

```
$ ./stop
```

10. 启动目标端进程，使多路实时装载生效。

```
$ ./start
```

11. 设置环境变量。

```
$ export XLDR_HOME=SuperSync 目标端复制目录/rmp
```

```
$ export VCFS_HOME=SuperSync 目标端复制目录/vcfsa
```

```
$ DBPS_HOME=SuperSync 目标端复制目录
```

#### 说明

例如 SuperSync 目标端复制目录为 `/dsg/supersync/dt`，则设置环境变量的命令为：

```
$ export XLDR_HOME=dsg/supersync/dt/rmp
```

```
$ export VCFS_HOME=dsg/supersync/dt/vcfsa
```

```
$ DBPS_HOME=dsg/supersync/dt
```

12. 进入 SuperSync 目标端 `bin` 目录中，执行交易任务拆分。

```
$ cd SuperSync 目标端复制目录/bin
```

```
$ ./xdispatch -p SuperSync 目标端复制目录/rmp/real0 -n xf1 起始  
编号 xf1 终止编号
```

#### 说明

- 目标端 `rmp/real0` 目录下存放着 SuperSync 的交易文件，该文件以 `xf1` 为后缀，`.xf1` 前的编号是以 `0,1,2.....` 数字顺序排列。
- `xf1` 起始编号和 `xf1` 终止编号表示要拆分的 `xf1` 交易文件的编号范围，例如要拆分 `0.xf1` 到 `20.xf1` 之间的所有交易文件，则拆分命令为：

```
$ ./xdispatch -p /dsg/supersync/dt/rmp/real0 -n 0 20
```

13. 系统进行拆分，并显示拆分过程，如：

```

CHECK_MEMLEAK is turned off
NLS_LANG=SIMPLIFIED CHINESE_CHINA.ZHS16GBK
stmt: alter session set skip_unusable_indexes =TRUE
To dispatch /dsg/supersync/dt/rmp/real0/0.xf1
2014-02-18:16:19:12: loading 2 items from /dsg/supersync/dt/config/
real_q.conf ...
In multiple loader mode.
2014-02-18:16:19:12: real, q# 0, XF1 31 ... (2.xf1) -twrite 31 bytes to 2.xf1 of
queue 0
elapsed 3ms
2014-02-18:16:19:12: real, q# 1, XF1 18 ... (0.xf1) -twrite 18 bytes to 0.xf1 of
queue 1
elapsed 14ms
2014-02-18:16:19:12: real, q# 2, XF1 18 ... (0.xf1) -twrite 18 bytes to 0.xf1 of
queue 2
elapsed 37ms
To dispatch /dsg/supersync/dt/rmp/real0/1.xf1
2014-02-18:16:19:12: real, q# 0, XF1 31 ... (3.xf1) -twrite 31 bytes to 3.xf1 of
queue 0
elapsed 2ms
2014-02-18:16:19:12: real, q# 1, XF1 18 ... (1.xf1) -twrite 18 bytes to 1.xf1 of
queue 1
elapsed 1ms
2014-02-18:16:19:12: real, q# 2, XF1 18 ... (1.xf1) -twrite 18 bytes to 1.xf1 of
queue 2
elapsed 1ms

```

### 说明

拆分过程中，用户也可以查看目标端 log.vagentd 日志，查看拆分队列情况：

```
$ cd SuperSync 目标端复制目录/log
```

```
$ tail -f log.vagentd
```

14. 系统拆分完成后，目标端 rmp 目录中，将会根据步骤 2 中设置的装载进程数，生成对应数目的 real\* 目录。

例如有 3 个装载进程，则在 rmp 目录中将存在 real0、real1、real2 三个目录。

15. 进入 SuperSync 源端 scripts 目录，启动源端进程。

```
$ cd SuperSync 源端复制目录/scripts
```

```
$ ./start
```

16. 系统将把源端累积的 xf1 文件，根据拆分队列发送到目标端。

## 8 DDL 过滤

SuperSync 提供过滤 DDL 操作功能，当生产系统上由于误操作执行了 drop、truncate 等 DDL 操作后，通过该功能使容灾系统可以对这些危险的 DDL 操作进行过滤，从而保证一旦在主库上进行了误操作后，还可以将容灾库的数据恢复到备库中去。

本章节包含下列主题：

- 对所有表进行 DDL 过滤
- 对特定表进行 DDL 过滤

### 8.1 对所有表进行 DDL 过滤

用户可以对数据库中的所有表进行 ddl 过滤，当源端数据库发生误操作执行了某些 DDL 操作时，保证这些操作不会复制到目标端中。

#### 前提条件

- 源端和目标端 SuperSync 进程均启动，且运行正常。
- 使用 dsg 用户进行操作。

#### 操作步骤

1. 进入 SuperSync 源端复制队列的 config 目录下，编辑 ddl.ini 文件。

```
$ cd SuperSync 源端复制目录/config
```

```
$ vi ddl.ini
```

2. 系统显示 ddl.ini 文件内容。

```
# PART 1: database level filter (...)  
grant=n  
revoke=n  
alter_database=n  
create_tablespace=n  
alter_tablespace=n  
drop_tablespace=n  
create_database_link=n  
drop_database_link=n
```

```
alter_user=n
create_user=n
drop_user=n
create_profile=n
alter_profile=n
drop_profile=n
create_role=n
alter_role=n
drop_role=n
create_type=n
create_type_body=n
create_directory=n
drop_directory=n

# PART 2: user level filter (...)
# table
create_table=y
drop_table=y
alter_table=y
analyze_table=y
truncate_table=y

# index
create_index=y
drop_index=y
alter_index=y
analyze_index=y

# sequence
create_sequence=y
alter_sequence=y
drop_sequence=y

# synonym
create_synonym=y
```

```
drop_synonym=y
alter_synonym=y

# view
create_view=y
drop_view=y
alter_view=y

# procedure
create_procedure=y
alter_procedure=y
drop_procedure=y

# trigger
create_trigger=y
alter_trigger=y
drop_trigger=y

# function
create_function=y
alter_function=y
drop_function=y

# package
create_package=y
alter_package=y
drop_package=y

# package body
create_package_body=y
alter_package_body=y
drop_package_body=y

# PART 3: table level filter (...)
```



 **说明**

初始的 ddl.ini 中，数据库级别即 PART 1: database level filter (...) 下方的 ddl 默认为 n，表级别即 # PART 2: user level filter (...) 下方的 ddl 默认为 y。n 表示过滤不进行复制，y 表示不过滤进行复制。

3. 根据需要对参数进行修改，将参数值修改为 n 或 y，使其进行 ddl 过滤或取消过滤。

例如：对 sequence 进行 create 操作过滤，则将 sequence 下方的 create\_sequence 参数值修改成 n。

```
# sequence
create_sequence=n
alter_sequence=y
drop_sequence=y
```

4. 退出并保存 ddl.ini 文件。

```
$ wq!
```

5. 进入 SuperSync 源端 scripts 目录，停止 SuperSync 进程。

```
$ cd SuperSync 源端复制目录/scripts
```

```
$ ./stop
```

6. 重新启动进程，使 ddl.ini 修改生效。

```
$ ./start
```

7. 当源端数据库中执行步骤 3 中过滤的操作时，系统阻止该操作复制到目标端中。用户可以通过源端 log 目录中的 log.vagentd 日志查看过滤情况。

如源端数据库执行 create\_sequence 操作时，log.vagent 中将显示下列过滤信息：

```
DDL filtered out: create sequence dff
OP:3.0 0x000001cd CREATE SEQUENCE DSG.DFF
opc 0xd.ffff fmt 0x0, flg 0 2 255 255, obj 289053/0, pg 1/1
xid 0x0001.017.00018e26 scn 0x0000.75504f36 .01
SQL by DSG/DSG 2014-02-20:14:22:07
create sequence dff
```

## 8.2 对特定表进行 DDL 过滤

用户可以对数据库中的特定的表进行 ddl 过滤，当源端数据库对这些表进行误操作执行了某些 DDL 操作时，保证这些操作不会复制到目标端中。

### 前提条件

- 源端和目标端 SuperSync 进程均启动，且运行正常。

- 使用 dsg 用户进行操作。

## 操作步骤

1. 进入 SuperSync 源端复制队列的 config 目录下，编辑 ddl.ini 文件。

```
$ cd SuperSync 源端复制目录/config
```

```
$ vi ddl.ini
```

2. 系统显示 ddl.ini 文件内容。

```
# PART 1: database level filter (...)
grant=n
revoke=n
alter_database=n
create_tablespace=n
alter_tablespace=n
drop_tablespace=n
create_database_link=n
drop_database_link=n
alter_user=n
create_user=n
drop_user=n
create_profile=n
alter_profile=n
drop_profile=n
create_role=n
alter_role=n
drop_role=n
create_type=n
create_type_body=n
create_directory=n
drop_directory=n

# PART 2: user level filter (...)
# table
create_table=y
drop_table=y
alter_table=y
```

```
analyze_table=y
truncate_table=y

# index
create_index=y
drop_index=y
alter_index=y
analyze_index=y

# sequence
create_sequence=y
alter_sequence=y
drop_sequence=y

# synonym
create_synonym=y
drop_synonym=y
alter_synonym=y

# view
create_view=y
drop_view=y
alter_view=y

# procedure
create_procedure=y
alter_procedure=y
drop_procedure=y

# trigger
create_trigger=y
alter_trigger=y
drop_trigger=y

# function
```

```
create_function=y
alter_function=y
drop_function=y

# package
create_package=y
alter_package=y
drop_package=y

# package body
create_package_body=y
alter_package_body=y
drop_package_body=y

# PART 3: table level filter (...)
```

#### 说明

初始的 ddl.ini 中，数据库级别即 PART 1: database level filter (...) 下方的 ddl 默认为 n，表级别即 # PART 2: user level filter (...) 下方的 ddl 默认为 y。n 表示过滤不进行复制，y 表示不过滤进行复制。

3. 在 # PART 3: table level filter (...) 下方添加需要过滤的表及 DDL 信息。

例如：对 dsg 用户下的 test 表过滤 drop\_table 和 truncate\_table 操作。

```
# PART 3: table level filter (...)
table=dsg.TEST
drop_table=n
truncate_table=n
```

4. 退出并保存 ddl.ini 文件。

```
$ wq!
```

5. 进入 SuperSync 源端 scripts 目录，停止 SuperSync 进程。

```
$ cd SuperSync 源端复制目录/scripts
```

```
$ ./stop
```

6. 重新启动进程，使 ddl.ini 修改生效。

```
$ ./start
```

7. 当源端数据库执行步骤 3 中设置的表的 DDL 操作时，系统将阻止该操作复制到目标端中。用户可以通过源端 log 目录中的 log.vagentd 日志查看过滤情况。

如源端数据库对 test 表进行 drop\_table 和 truncate\_table 操作时，系统将显示下列过滤信息：

```
DDL filtered out: drop table test
OP:3.0 0x000001d1 DROP TABLE DSG.TEST
      opc 0xc.1 fmt 0x0, flg 0 2 1 255, obj 159491/159491, pg 1/1
      xid 0x00a8.015.00037186 scn 0x0836.a6f57c11 .00
      SQL by DSG/DSG 2014-02-24:14:47:51
      drop table test

DDL filtered out:
OP:3.0 0x000001d5 TRUNCATE DSG.TEST
      opc 0x55.1 fmt 0x0, flg 0 0 1 255, obj 159491/159491, pg 1/1
      xid 0x00a5.027.0006360a scn 0x0836.a6f22cb7 .00
      SQL by DSG/DSG 2014-2-24:15:30:15
      truncate table test
```

## 9 查看性能统计分析

SuperSync 提供性能统计分析，用户可以通过打开查看这些统计信息，了解数据复制过程中 SuperSync 执行的操作及操作使用的时间。

本章节包含下列主题：

- 查看源端性能统计分析
- 查看目标端性能统计分析

### 9.1 查看源端性能统计分析

介绍源端性能统计分析查看方法。

#### 前提条件

- 源端 SuperSync 进程均启动，且运行正常。
- 使用 dsg 用户进行操作。

#### 操作步骤

1. 进入源端服务器 SuperSync 复制队列的 config 目录中，编辑 profile\_ds 文件。

```
$ cd SuperSync 源端复制目录/config
```

```
$ vi .profile_ds
```

2. 在 profile\_ds 文件的最上方将 export REPORT\_PERFORMANCE=ON 和 export REPORT\_PERFORMANCE\_DETAIL=ON 前的 # 删除，使这两个环境变量生效。

```
#0 For some useful variables
export REPORT_PERFORMANCE=ON
export REPORT_PERFORMANCE_DETAIL=ON
#export XLOGGER=Y
#export IN_MEM_XF1_SIZE=67108864
```

3. 保存并退出编辑状态。

```
$ :wq!
```

4. 进入到 scripts 目录中，启动 SuperSync 进程，使配置文件 profile\_ds 修改生效。

```
$ cd SuperSync 源端复制目录/scripts
```

```
$ ./start
```

5. 进入 log 目录中，查看 log.vagentd 日志。

```
$ cd SuperSync 源端复制目录/log
```

```
$ tail -f log.vagentd
```

6. 系统将显示 SuperSync 的性能统计分析信息。

```
2014-01-22:14:23:30: *** PERFORMANCE STATISTICS (elapsed 427ms) ***
```

```
log read          1 ms (0.23%)
extract to XDT    1 ms (0.23%)
XDT to XF1       17 ms (3.98%)
CACHE LOB        0 ms (0.00%)
rollback         0 ms (0.00%)
datalink         0 ms (0.00%)
OB_push         0 ms (0.00%)
OB_filter        0 ms (0.00%)
XF1 x_add        1 ms (0.23%)
XF1 x_add xtag   0 ms (0.00%)
XF1 x_add dmln   0 ms (0.00%)
XF1 x_add dump   0 ms (0.00%)
XF1 x_add DDL    0 ms (0.00%)
XF1 write       180 ms (42.15%)
misc.           227 ms (53.16%)
```

```
-----
total           427 ms
```

```
2014-01-22:14:23:30: *** DETAIL PERFORMANCE STATISTICS ***
```

```
XDT2XF1         17 ms (100.00%)
  OPH_01        0 ms (0.00%)
  OPH_0B        0 ms (0.00%)
  OPH_05        0 ms (0.00%)
  e             0 ms (nan%)
  x             0 ms (nan%)
  OPH_03       17 ms (100.00%)
```

```
-----
X_ADD           0 ms (0.00%)
SEARCH         0 ms (nan%)
```

```

APPEND 0 ms (nan%)
-----
total      17 ms
    
```

 说明

- \*\*\* PERFORMANCE STATISTICS (elapsed 427ms) \*\*\*下方显示的是 SuperSync 正在执行的交易的性能。
- \*\*\* DETAIL PERFORMANCE STATISTICS \*\*\*下方显示的是 SuperSync 交易的详细性能信息。
- 如果用户不需要查看详细的性能统计信息，则步骤 2 中只需要添加 export REPORT\_PERFORMANCE=ON 环境变量即可。

7. 用户也可以查看 log.sender 日志，查看发送进程的统计信息。

**\$ tail -f log.sender**

8. 系统将显示 sender 进程的统计信息。

```

2014-01-22:14:46:08: *** DETAIL PERFORMANCE STATISTICS ***
XF1Filter          33 ms (100.00%)
MEM PREPARE        33 ms (100.00%)
LOB PREPARE         0 ms (0.00%)
CACHE DEAL         0 ms (0.00%)
WRITE FILE         0 ms (0.00%)
  CHAINED ROW GET  0 ms (nan%)
  CHAINED_ROW_MALLOC 0 ms (nan%)
  GETTING LOB      0 ms (nan%)
WRITE OTHER        0 ms (nan%)
LAST DEAL          0 ms (0.00%)
-----
total              33 ms

2014-01-22:14:46:08: /dsg/supersync/ds/rmp/2.13.xf1(230) (PID-31789
PPID-31788) ...

2014-01-22:14:46:08: *** PERFORMANCE STATISTICS ***
read file          0 ms (0.00%)
xf1 filter         48 ms (97.96%)
send file          0 ms (0.00%)
misc.              1 ms (2.04%)
-----
    
```



```
total          49 ms
```

## 9.2 查看目标端性能统计分析

介绍目标端性能统计分析查看方法。

### 前提条件

- 目标端 SuperSync 进程均启动，且运行正常。
- 使用 dsG 用户进行操作。

### 操作步骤

1. 进入目标端服务器 SuperSync 复制队列的 config 目录中，编辑 .profile\_dt 文件。

```
$ cd SuperSync 目标端复制目录/config
```

```
$ vi .profile_dt
```

2. 在 .profile\_dt 文件的最上方将 export REPORT\_PERFORMANCE=ON 和 export REPORT\_PERFORMANCE\_DETAIL=ON 前的 # 删除，使这两个环境变量生效。

```
#0 For some useful variables
export REPORT_PERFORMANCE=ON
export REPORT_PERFORMANCE_DETAIL=ON
#export XLOGGER=Y
#export IN_MEM_XF1_SIZE=67108864
```

3. 保存并退出编辑状态。

```
$ :wq!
```

4. 进入目标端 scripts 目录，启动 SuperSync 进程，使 .profile\_dt 文件修改生效。

```
$ cd SuperSync 目标端复制目录/scripts
```

```
$ ./start
```

5. 进入目标端服务器 SuperSync 复制队列的 log 目录中，查看 log.r0 日志。

```
$ cd SuperSync 目标端复制目录/log
```

```
$ tail -f log.r0
```

6. 系统将显示 SuperSync 的性能统计分析信息。

```
2014-01-22:15:01:29: *** PERFORMANCE STATISTICS (elapsed 1ms) ***
insert          0 ms (0.00%)
delete          0 ms (0.00%)
update          0 ms (0.00%)
```

```

ddl                0 ms (0.00%)
rowmap             0 ms (0.00%)
commit            1 ms (100.00%)
get dict          0 ms (0.00%)
  other           0 ms (0.00%)
  table list      0 ms (0.00%)
  from src db     0 ms (0.00%)
  from memory     0 ms (0.00%)
  from file       0 ms (0.00%)
  select struct  0 ms (0.00%)
check struct      0 ms (0.00%)
uncompress        0 ms (0.00%)
save dict         0 ms (0.00%)
hint              0 ms (0.00%)
dpi close         0 ms (0.00%)
misc.             0 ms (0.00%)
-----
total             1 ms
  
```

2014-01-22:15:01:29: \*\*\* DETAIL PERFORMANCE STATISTICS \*\*\*

```

INSERT            0 ms (0.00%)
  MEM ALLOCATE    0 ms (nan%)
  BUF POPULATE    0 ms (nan%)
  INS PREPARE     0 ms (nan%)
  OCI OPERATION   0 ms (nan%)
  RID MAPPING     0 ms (nan%)
-----
UPDATE            0 ms (0.00%)
  MEM ALLOCATE    0 ms (nan%)
  BUF POPULATE    0 ms (nan%)
  GET ROWID       0 ms (nan%)
  OCI BIND        0 ms (nan%)
  OCI EXEC        0 ms (nan%)
  RID MAPPING     0 ms (nan%)
-----
  
```

```

DELETE                0 ms (0.00%)
    MEM ALLOCATE      0 ms (nan%)
    BUF POPULATE     0 ms (nan%)
    OCI OPERATION    0 ms (nan%)
    RID MAPPING      0 ms (nan%)
-----
COMMIT                1 ms (100.00%)
    DBI DIRPATH      0 ms (0.00%)
    DBI NORMAL       1 ms (100.00%)
    RMP COMMIT       0 ms (0.00%)
    RMP COMMIT1     0 ms (0.00%)
-----
total                 1 ms

```

 说明

- **\*\*\* PERFORMANCE STATISTICS (elapsed 1ms) \*\*\***下方显示的是 SuperSync 正在执行的交易的性能。
- **\*\*\* DETAIL PERFORMANCE STATISTICS \*\*\***下方显示的是 SuperSync 交易的详细性能信息。
- 如果用户不需要查看详细的性能统计信息，则步骤 2 中只需要添加 `export REPORT_PERFORMANCE=ON` 环境变量即可。

# 10 系统监控

SuperSync 提供系统监控功能，用户可以通过该功能简单查看源端和目标端进程运行是否正常及交易的延迟情况。

本章节包含下列主题：

- 源端监控
- 目标端监控

## 10.1 源端监控

介绍 SuperSync 源端系统监控操作方法。

### 前提条件

- 源端 SuperSync 进程均启动，且运行正常。
- 使用 dsG 用户进行操作。

### 操作步骤

1. 进入源端服务器 SuperSync 复制队列的 scripts 目录中，执行监控操作。

```
$ cd SuperSync 源端复制目录/scripts
```

```
$ ./mon
```

2. 系统显示进程监控信息。

```
=====
Monitor Ds_Vagentd : 2014-01-22:15:53:31
=====

DBPS_HOME      Component      errCount  lastAnalyzingTime
lastTransactionTime  AnalyzeDelayTime(s)  MonitringTime
/dsg/supersync/ds      Ds_vagentd      0          2014-01-22:15:53:28
2014-1-22:15:53:22      6              2014-01-22:15:53:31
=====

Monitor Sender : 2014-01-22:15:53:31
=====

DBPS_HOME      Component      errCount  warningCount
ParsedNum      SendedNum      SenderDelayNum  lastSendedTime
```

```
SenderDelayTime(s)
/dsg/supersync/ds      Sender      5      0      24
24      0      2014-01-22:15:52:59 32
=====
```

 **说明**

1) vagentd 分析延迟监控:

- DBPS\_HOME: 为源端复制目录。
- Component: 监控的 SuperSync 进程, 源端监控 vagentd 和 sender 进程。
- errCount: 当前分析日志的最近 300 行中包含错误关键字的行数, 错误关键字包括: ORA-、err、Err 等。日志中包含这些错误关键字的记录并不完全都代表 DSG 进程有误。
- lastAnalyzingTime: 最近一次分析日志的时间。
- lastTransactionTime: 最后一次分析日志中交易的时间。
- AnalyzeDelayTime(s)=lastAnalyzingTime- lastTransactionTime, 分析日志相关交易的延迟时间。
- MonitringTime: 监控时间。

2) sender 发送延迟监控:

- DBPS\_HOME: 为源端复制目录。
- Component: 监控的 SuperSync 进程, 源端监控 vagentd 和 sender 进程。
- errCount: 当前分析日志的最近 300 行中包含错误关键字的行数, 错误关键字包括: ORA-、err、Err 等。日志中包含这些错误关键字的记录并不完全都代表 DSG 进程有误。
- warningCount: 当前分析日志的最近 300 行中包含告警关键字的行数, 告警关键字包括: warning。
- ParsedNum: vagentd 分析日志解析的交易文件数量, 即 sender 要发送交易文件数量。
- SendedNum: sender 已经发送交易文件数量。
- SenderDelayNum=ParsedNum- SendedNum, sender 发送交易文件个数的延迟。
- lastSendedTime: sender 最近一次发送交易文件的时间。
- SenderDelayTime(s)= MonitringTime- lastSendedTime, sender 发送交易文件的时间延迟。

## 10.2 目标端监控

介绍 SuperSync 目标端系统监控操作方法。

## 前提条件

- 目标端 SuperSync 进程均启动，且运行正常。
- 使用 dsg 用户进行操作。

## 操作步骤

1. 进入目标端服务器 SuperSync 复制队列的 scripts 目录中，执行监控操作。

```
$ cd SuperSync 目标端复制目录/scripts
```

```
$ ./mon
```

2. 系统显示进程监控信息。

```
=====
Monitor Loader : 2014-01-22:15:53:31
=====
DBPS_HOME      Component      errCount  ReceivedNum
LoadedNum  LoadDelayNum  lastTranTime  LoadDelayTime
/dsg/supersync/dt  Loader_r0      82013     82012     1
2014-01-22:15:53:31      11
=====
Monitor Dt_Vagentd : 2014-01-22:15:53:31
=====
DBPS_HOME      Component      errCount  lastReceivedTime
MonitringTime  NoRecived(s)  MapDelayNum
/dsg/supersync/dt  Dt_Vagentd      0
2014-01-22:15:52:59
2014-01-22:15:53:00      1      Unknown
=====
```

### 说明

1) loader 加载延迟监控:

- DBPS\_HOME: 为目标端复制目录。
- Component: 监控的 SuperSync 进程, 目标端监控 vagentd 和 loader 进程。
- errCount: 当前分析日志的最近 300 行中包含错误关键字的行数, 错误关键字包括: ORA-、err、Err 等。日志中包含这些错误关键字的记录并不完全都代表 DSG 进程有误。
- ReceivedNum: vagentd 接收到源端发送的交易文件数量, 即 loader 要加载的交易文件数量。

- **LoadedNum**: loader 已经加载的交易文件数量。
- **LoadDelayNum=ReceivedNum-LoadedNum**, loader 加载积压, 交易文件数量的延迟。
- **lastTranTime**: loader 最近一次加载的交易文件中交易发生的时间。
- **LoadDelayTime(s)=MonitoringTime-lastTranTime**, loader 加载交易文件的延迟时间。

2) vagentd 接收延迟:

- **DBPS\_HOME**: 为目标端复制目录。
- **Component**: 监控的 SuperSync 进程, 目标端监控 vagentd 和 loader 进程。
- **errCount**: 当前分析日志的最近 300 行中包含错误关键字的行数, 错误关键字包括: ORA-、err、Err 等。日志中包含这些错误关键字的记录并不完全都代表 DSG 进程有误。
- **lastReceivedTime**: vagentd 最近一次接受交易文件的时间。
- **MonitoringTime**: 进行监控的时间。
- **NoReceived(s)= monitoringTime-lastReceivedTime**, vagentd 有多长时间没有接收交易文件的时间。
- **MapDelayNum**: 单向复制拓扑中无意义 (显示为 unknown), 双活容灾复制环境下, 目标端 rmp\_sender 发送映射关系文件的数量延迟, 也可以是源端 rmp\_loader 加载映射关系文件的数量延迟。

# 11 数据比对

---

DSG 单独提供 Monitor 工具，用于复制完成后对源端和目标端的数据进行对比，保证复制数据的准确性。

## 前提条件

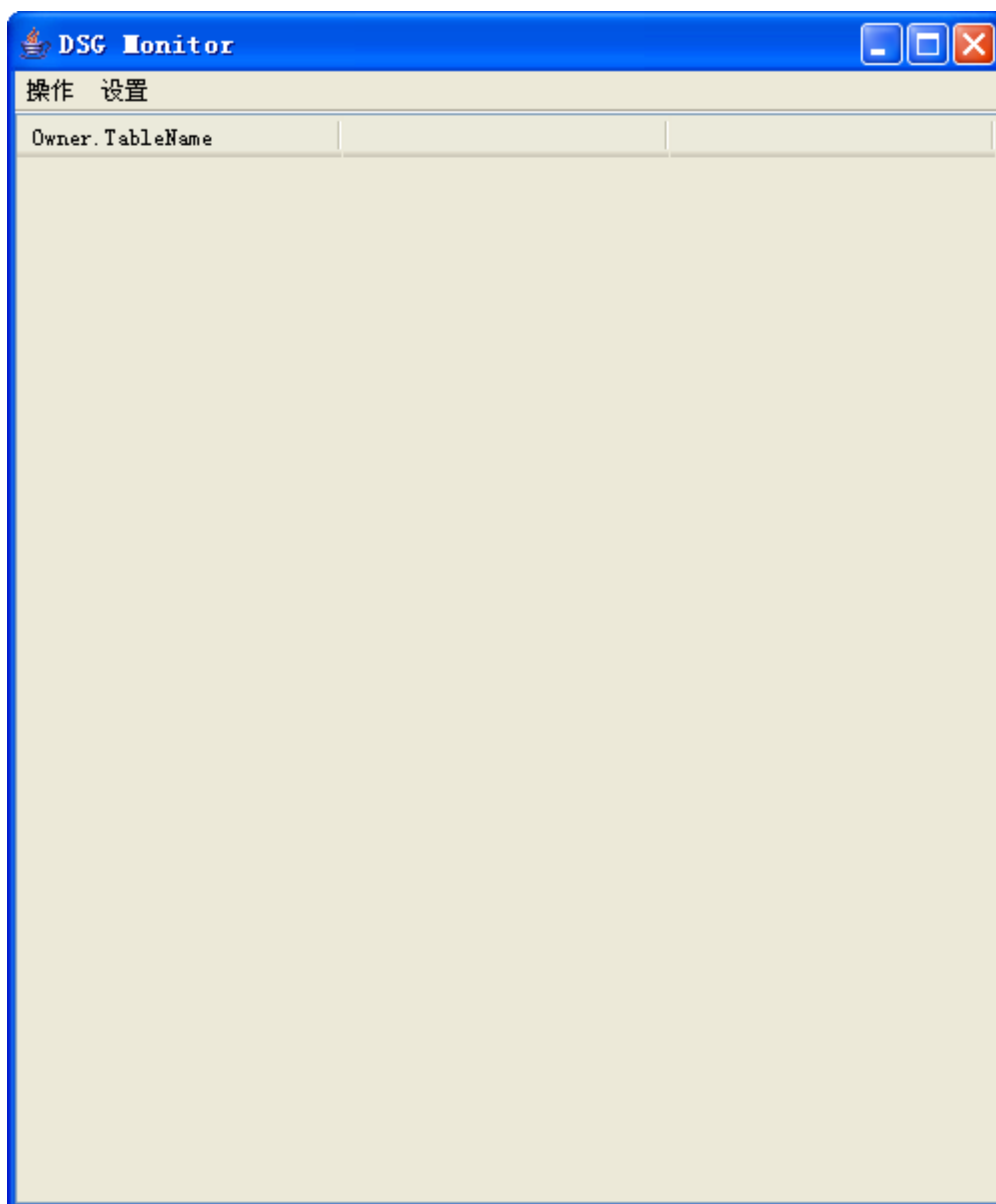
- SuperSync 复制任务已完成。
- 已获取 DSG Monitor 工具。

## 操作步骤

1. 双击 DSG Monitor 客户端应用程序 DSGMonitor.exe，打开 Monitor 操作界面，如图 4 所示。

图 4 DSG Monitor 操作界面





2. 选择菜单“操作>添加主机”，打开“添加主机”窗口，如图 5 所示。

图 5 添加主机

- 在“添加主机”窗口下方先选择添加的是源端主机还是目标端主机，源端主机请选择“SRC”，目标端主机请选择“TGT”；在“主机 IP”输入框中输入主机 IP 地址；在“SID”输入框中输入数据库实例名；在“连接用户名”输入框中输入数据库用户名，默认为“dsg”；在“密码”输入框中输入数据库用户密码，即 dsg 用户的密码；在“端口”输入框中输入 Oracle 数据库监听端口，默认为 1521，如图 6 所示。

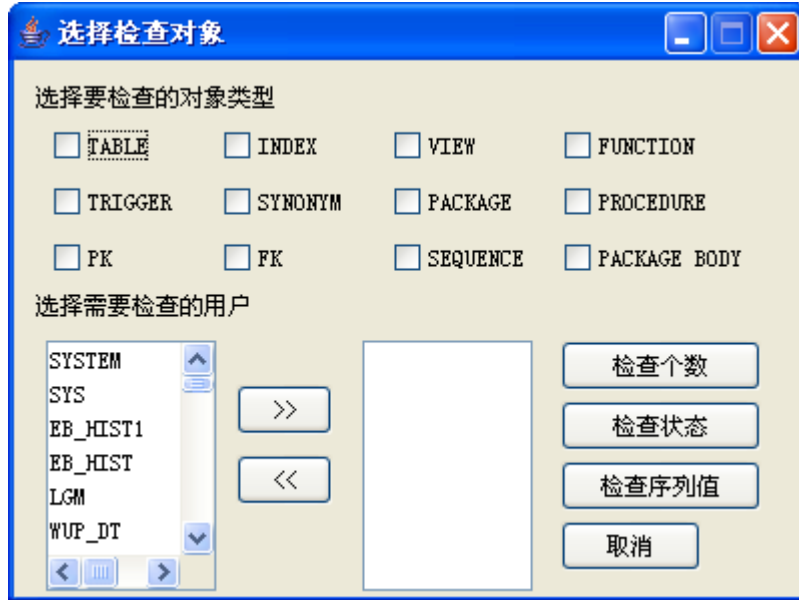
图 6 添加主机信息

- 设置完成后，单击“测试”，系统对主机进行连接测试，并弹出提示框，如图 7 所示则表示主机设置正确，连接成功。

图 7 连接成功

- 单击“确定”，退出提示框口，在“添加主机”窗口中单击“确定”，主机设置完成。
- 源端和目标端主机设置完成后，在 Monitor 操作界面中，选择“操作>检查对象”，系统弹出“选择检查对象”对话框，如图 8 所示。

图 8 选择检查对象



- 在“选择要检查的对象类型”下方勾选对象，在“选择需要检查的用户”下方将需要检查的用户从左侧列表框中添加到右侧后，单击“检查个数”，系统弹出“检查用户对象”窗口，如图 9 所示。

图 9 检查用户对象

| Owner.ObjectType | 192.168.1.154_db10_SRC | 192.168.1.156_db10_TGT |
|------------------|------------------------|------------------------|
| SYSTEM.TABLE     | 200                    | 200                    |
| SYSTEM.INDEX     | 223                    | 223                    |
| SYSTEM.VIEW      | 12                     | 11                     |
| SYSTEM.FUNCTION  | 1                      | 1                      |
| SYS.TABLE        | 649                    | 649                    |
| SYS.INDEX        | 807                    | 807                    |
| SYS.VIEW         | 2840                   | 2838                   |
| SYS.FUNCTION     | 59                     | 59                     |
| LGM.TABLE        | 60                     | 60                     |
| LGM.INDEX        | 27                     | 27                     |
| LGM.VIEW         | 0                      | 0                      |
| LGM.FUNCTION     | 0                      | 0                      |

- 在“检查用户对象”窗口中查看源端和目标端数据个数比对结果，通过比对判断数据复制的准确度。
- 用户还可以比对源端和目标端数据表的行数，在 Monitor 操作界面中，选择“操作>比对行数”，系统弹出“选择监控表”对话框，如图 10 所示。

图 10 选择监控表




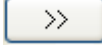
10. 在“选择监控表”窗口最左侧选择用户，则中间列表框中将显示该用户下的所有数据表，通过  按钮将需要对比的数据表添加到右侧列表框中；或在“筛选表”上方的输入框中输入数据表关键字，由系统在中间列表框中列出包含该关键字的所有数据表，用户再通过  按钮将需要对比的数据表添加到右侧列表框中。设置完成后，单击“确定”，系统将在 Monitor 操作界面中显示源端和目标端数据表的行数信息，如图 11 所示。

图 11 表行数比对

DSG Monitor Thu Feb 13 11:13:45 CST 2014

操作 设置

| Owner.TableName        | 192.168.1.154_db10_SRC | 192.168.1.156_db10_TGT |
|------------------------|------------------------|------------------------|
| DSG.LIUCY              | 2                      | 0                      |
| DSG.PARTB              | 9                      | 1                      |
| DSG.TEST               | 8                      | 0                      |
| DSG.AUDIT_ACTIONS      | 0                      | 0                      |
| DSG.BB                 | 0                      | 0                      |
| DSG.BBBB               | 0                      | 0                      |
| DSG.BBTMP              | 0                      | 0                      |
| DSG.BF_TZFCZGDQX       | 1590                   | 1590                   |
| DSG.BIGCOLTAB          | 0                      | 0                      |
| DSG.BIGT1              | 1                      | 1                      |
| DSG.BIGT2              | 2                      | 2                      |
| DSG.BLOB_TEST1         | 0                      | 0                      |
| DSG.C1                 | 2996                   | 2996                   |
| DSG.C2                 | 2991                   | 2991                   |
| DSG.C3                 | 2991                   | 2991                   |
| DSG.C4                 | 3000                   | 3000                   |
| DSG.CHAINEDROWSTEST    | 0                      | 0                      |
| DSG.CHANGE             | 0                      | 0                      |
| DSG.A1                 | 0                      | 0                      |
| DSG.AATMP              | 0                      | 0                      |
| DSG.ABC                | 2                      | 2                      |
| DSG.ADSLSERVICE        | 0                      | 0                      |
| DSG.CHECK_LOB_ERR      | 0                      | 0                      |
| DSG.CHECK_TAB          | 0                      | 0                      |
| DSG.CHECK_TAB_BAK      | 0                      | 0                      |
| DSG.CHECK_TAB_HIS      | 0                      | 0                      |
| DSG.CON_FASCICULE_INFO | 0                      | 0                      |
| DSG.DDD                | 0                      | 0                      |
| DSG.DEPARTMENT         | 0                      | 0                      |
| DSG.DP_TAB_ROWcnt_BF   | 1738                   | 1738                   |
| DSG.DQMAPPING          | 4                      | 4                      |
| DSG.DSGSCHMAP          | 1                      | 1                      |
| DSG.DSG_CHECK_HIS      | 1832                   | 1832                   |

 说明

在检查过程中，如果发现源端和目标端数据比对差异很大，请联系 DSG 技术人员。

# 12 xf1 文件操作

xf1 文件为 SuperSync 存储交易的文件，用户可以对 xf1 文件进行查看、删除等操作。

本章节包含下列主题：

- 查看 xf1 文件
- 删除 xf1 文件中的某个 objn 交易
- 删除 xf1 文件中的某类操作的交易
- 删除 xf1 文件中的某条 Rowid 记录

## 12.1 查看 xf1 文件

用户可以对 xf1 文件进行查看，可根据需要查看 xf1 的 Rowid 信息、十六进制和 SQL 语句信息。

### 12.1.1 查看 xf1 文件的 Rowid 信息

用户可以根据需要查看 xf1 文件的 Rowid 信息，Rowid 信息包含 Oracle 的 bojn、objd、dba 和 slot 信息。

#### 前提条件

- SuperSync 已完成首次初始化，具体操作请参见 [3 首次初始化](#)。
- 源端和目标端 SuperSync 进程均启动，且运行正常。
- 使用 dsG 用户进行操作。

#### 操作步骤

1. 进入 SuperSync 源端或目标端 bin 目录中，执行查看 xf1 文件命令。

```
$ cd SuperSync 源端或目标端复制目录/bin
```

```
$ ./xfview -show 1 xf1 文件存放目录/*.xf1
```

#### 说明

- 源端 xf1 文件存放在源端 rmp 目录中，目标端 xf1 文件存放在目标端 rmp/real\* 目录中。
- 当 SuperSync 只有一路复制装载进程时，在目标端 rmp 目录中只有 real0 文件夹。

- 当 SuperSync 有多路复制装载进程时，在目标端 rmp 目录中，将存在多个 real\*文件夹，如 SuperSync 有 3 路装载进程，则 rmp 目录中将存在 real0、real1、real2 三个文件夹。xf1 交易文件根据要求分别存放在这三个文件夹中。

- 例如查看目标端 real0 目录下的 2.xf1 文件，则命令为：

```
$ cd /dsg/supersync/dt/bin
$ ./xfview -show 1
/dsg/supersync/dt/rmp/real0/2.xf1
```

2. 系统显示 xf1 文件的 Rowid 信息，如：

```
2.xf1
CHECK_MEMLEAK is turned off
      : 00(NIL)
      : 000441BF.000441FB.0103CE26.0000 B3(DRP) --H-FL-- cc:3
len:22
      : 54(CMT), SCN 0x74fe.0a110001, Time 2014-01-26 16:01:45
-- Time: 0 seconds
```

#### 说明

- 00(NIL)表示内容为空。
- 000441BF.000441FB.0103CE26.0000 表示 Oracle 的 objn.objd.dba.slot 信息。
- B3(DRP)为操作符，B3 代表 delete。xf1 文件中常用的操作符有：B2、B3、B4、B5，B2 代表 insert，B4 代表 lock，B5 代表 update。
- 54(CMT), SCN 0x74fe.0a110001, Time 2014-01-26 16:01:45 表示 SCN 为 0x74fe.0a110001 的 commit 标记及时间，其中 CMT 表示 commit。

## 12.1.2 查看 xf1 文件的十六进制信息

用户可以根据需要查看 xf1 文件的十六进制信息，用于与源端 Oracle 数据进行比较。

### 前提条件

- SuperSync 已完成首次初始化，具体操作请参见 [3 首次初始化](#)。
- 源端和目标端 SuperSync 进程均启动，且运行正常。
- 使用 dsg 用户进行操作。

### 操作步骤

1. 进入 SuperSync 源端或目标端 bin 目录中，执行查看 xf1 文件命令。

```
$ cd SuperSync 源端或目标端复制目录/bin
```



**\$ ./xfview -show 2 xf1 文件存放目录/\*.xf1**

**📖 说明**

- 源端 xf1 文件存放在源端 rmp 目录中，目标端 xf1 文件存放在目标端 rmp/real\*目录中。
- 当 SuperSync 只有一路复制装载进程时，在目标端 rmp 目录中只有 real0 文件夹。
- 当 SuperSync 有多路复制装载进程时，在目标端 rmp 目录中，将有多多个 real\*文件夹，如 SuperSync 有 3 路装载进程，则 rmp 目录中将存在 real0、real1、real2 三个文件夹。xf1 交易文件根据要求分别存放在这三个文件夹中。
- 例如查看目标端 real0 目录下的 2.xf1 文件，则命令为：

**\$ cd /dsg/supersync/dt/bin**

**\$ ./xfview -show 2  
/dsg/supersync/dt/rmp/real0/2.xf1**

2. 系统显示 xf1 文件的十六进制信息，如：

```

2.xf1
CHECK_MEMLEAK is turned off
    Len [ 4]: 0000 000D
    OPHL [ 1]: 00
    Context [ 8]: 0003 0100 0000 003A .....

    Len [ 4]: 0000 0029
    OPHL [ 1]: B3
    Objn [ 4]: 0004 41BF
    Rowid [ 10]: 0004 41FB 0103 CE26-0000
    Flag Byte [ 1]: 2C
    CC [ 1]: 03
    COL_001 [ 5]: 0003 4C59 58 ..LYX
    COL_002 [ 7]: 0005 C402 2751 18 ..!Q.
    COL_003 [ 4]: 0002 C20B ....

    Len [ 4]: 0000 0011
    OPHL [ 1]: 54
    SCN [ 8]: 0000 74FE 0A11 0001
    SCN Time [ 4]: 31F1 02E9
    
```

```
-- Time: 0 seconds
```

### 12.1.3 查看 xf1 文件的 SQL 语句

用户可以查看 xf1 的 SQL 语句。

#### 前提条件

- SuperSync 已完成首次初始化，具体操作请参见 [3 首次初始化](#)。
- 源端和目标端 SuperSync 进程均启动，且运行正常。
- 使用 dsg 用户进行操作。

#### 操作步骤

- 源端查看 xf1 文件
  1. 进入 SuperSync 源端 bin 目录中，执行查看 xf1 文件命令。

```
$ cd SuperSync 源端复制目录/bin
```

```
$ ./xfview -show 3 -dict ../rmp/s_dict.xf1 ../rmp/*.xf1
```

#### 说明

s\_dict.xf1 为 SuperSync 的数据字典，在执行查询命令前，请确定该数据字典已放置在源端 rmp 目录中，如果 rmp 目录中没有该文件，请从 SuperSync 目标端的 rmp/imp\_dsg 中拷贝该文件。

2. 系统显示 xf1 文件的 SQL 语句，如：

```
2.xf1
CHECK_MEMLEAK is turned off
<sdict used xf1>
CREATE INDEX "DSG"."IDX_STD_CFM_JOUR" ON
"DSG"."STD_CFM_JOUR" ("BUSI_CODE") TABLESPACE "USERS"
LOCAL PARALLEL 4;

-- Time: 1 seconds
```

- 目标端查看 xf1 文件
  1. 进入 SuperSync 目标端 bin 目录中，执行查看 xf1 文件命令。

```
$ cd SuperSync 目标端复制目录/bin
```

```
$ ./xfview -show 3 ../rmp/real*/*.xf1
```

#### 说明

- 当 SuperSync 只有一路复制装载进程时,在目标端 rmp 目录中只有 real0 文件夹。
  - 当 SuperSync 有多路复制装载进程时,在目标端 rmp 目录中,将有多多个 real\*文件夹,如 SuperSync 有 3 路装载进程,则 rmp 目录中将存在 real0、real1、real2 三个文件夹。xf1 交易文件根据要求分别存放在这三个文件夹中。
2. 系统显示 xf1 文件显示的信息,如:

```

2.xf1
CHECK_MEMLEAK is turned off
    No dictionary available, translation will be not possible
    Len [ 4]: 0000 0029
    OPHL [ 1]: B3
    Objn [ 4]: 0004 41BF
    Rowid [ 10]: 0004 41FB 0103 CE26-0000
    Flag Byte [ 1]: 2C
    CC [ 1]: 03
    COL_001 [ 5]: 0003 4C59 58          ..LYX
    COL_002 [ 7]: 0005 C402 2751 18      .'Q.
    COL_003 [ 4]: 0002 C20B             ....

commit; -- (0x74fe.0a110001 0x0000.000.00000000 2014-01-26:16:01:45)
-- Time: 0 seconds
    
```

 说明

- Len 表示该条交易的长度。
- OPHL 表示操作类型, B3 代表 delete 操作。xf1 文件中常用的操作符有: B2、B3、B4、B5, B2 代表 insert, B4 代表 lock, B5 代表 update。
- Objn 表示 Oracle 的 object id。
- Rowid 表示 Oracle 的 Rowid 信息,包含 Oracle 的 objd、dba 和 slot 信息。
- Flag Byte 表示程序标记,默认值为 2C。
- CC 表示该交易有多少列。
- COL\_001、COL\_002、COL\_003 表示第一列、第二列和第三列的内容。
- commit; -- (0x74fe.0a110001 0x0000.000.00000000 2014-01-26:16:01:45 表示该 SCN 的 commit 标记及时间。

- 对 xf1 文件中的 SQL 信息进行翻译，使其成为可阅读的 sql 语句。

```
$ ./xfview -dict ../rmp/imp_dsg/s_dict.xf1 ../rmp/real*/  
*.xf1
```

- 系统显示 sql 语句，如步骤 2 中显示的内容翻译后为：

```
2.xf1  
CHECK_MEMLEAK is turned off  
DELETE FROM "DSG"."TEST" WHERE "TEL"=1388023 /*  
AABEH7AAEAAA84mAAA */;  
commit; -- (0x74fe.0a110001 0x0000.000.00000000 2014-01-26:16:01:45)  
-- Time: 5 seconds
```

## 12.2 删除 xf1 文件中的某个 objn 交易

用户可以在 xf1 文件中删除某个 objn 交易，使其不复制装载到目标端中。

### 前提条件

- SuperSync 已完成首次初始化，具体操作请参见 [3 首次初始化](#)。
- 源端和目标端 SuperSync 进程均启动，且运行正常。
- 使用 dsg 用户进行操作。

### 操作步骤

- 使用 dsg 用户进入源端数据库，查出要删除表的 objn 号。

```
$ sqlplus dsg 用户名/密码
```

```
SQL> select object_id from dba_objects where owner='dsg 用  
户名' and object_name='表名';
```

#### 说明

“表名”处修改成需要删除的表的名称。例如查询 dsg 用户下 test 表：

```
SQL> select object_id from dba_objects where  
owner='DSG ' and object_name='TEST';
```

- 系统显示查询表的 objn 号，如：

```
OBJECT_ID  
-----  
280529
```

- 进入 SuperSync 目标端 rmp/real\* 目录下，执行删除操作。

```
$ cd SuperSync 目标端复制目录/rmp/real*
```

```
$ ../../bin/xfview -delobj objn *.xf1
```

#### 说明

objn 处请修改成要删除的表的 object 编号，\*.xf1 修改成该表所存储的 xf1 文件，real\* 表示该\*.xf1 文件存放在的目录。

例如需要删除 objn 为 280529，该交易存放在 dsg/supersync/dt/rmp/real0 目录下的 0.xf1 文件中，则命令应修改为：

```
$ cd dsg/supersync/dt/rmp/real0
```

```
$ ../../bin/xfview -delobj 280529 0.xf1
```

4. 系统执行删除操作，并显示下面信息：

```
CHECK_MEMLEAK is turned off
Deleted: IRP-0 URP-0 DRP-0 QMI-1 OTher-0
```

#### 说明

显示信息中的 QMI-1 表示已完成交易删除。

## 12.3 删除 xf1 文件中的某类操作的交易

用户可以对 xf1 文件中的某类操作（如 delete、insert、update 等操作）的交易进行删除，使其不复制装载到目标端中。

### 前提条件

- SuperSync 已完成首次初始化，具体操作请参见 [3 首次初始化](#)。
- 源端和目标端 SuperSync 进程均启动，且运行正常。
- 使用 dsg 用户进行操作。

### 操作步骤

1. 进入 SuperSync 目标端 rmp/real\* 目录下，查看需要进行操作的\*.xf1 文件的操作类型。

```
$ cd SuperSync 目标端复制目录/rmp/real*
```

```
$ ../../bin/xfview -show 1 *.xf1
```

#### 说明

- 当 SuperSync 只有一路复制装载进程时，在目标端 rmp 目录中只有 real0 文件夹。
- 当 SuperSync 有多路复制装载进程时，在目标端 rmp 目录中，将有多多个 real\* 文件夹，如 SuperSync 有 3 路装载进程，则 rmp 目录中将存在 real0、real1、real2 三个文件夹。xf1 交易文件根据要求分别存放在这三个文件夹中。

2. 系统将显示该 xf1 文件的操作信息，如：

```
2.xf1
CHECK_MEMLEAK is turned off
      : 00(NIL)
      : 000441BF.000441FB.0103CE26.0000 B3(DRP) --H-FL-- cc:3
len:22
      : 54(CMT), SCN 0x74fe.0a110001, Time 2014-01-26 16:01:45
-- Time: 0 seconds
```

 说明

其中 B3(DRP)显示交易的操作类型，常用的操作符有：B2 代表 insert，B3 代表 delete，B4 代表 lock，B5 代表 update。

3. 执行删除命令，将该 xf1 文件中需要删除的操作类型进行删除。

```
$ ../../bin/xfview -delopc 0xb* *.xf1
```

 说明

请将 0xb\*中的\*改成操作类型对应的数字，如需要删除 B3 操作类型的交易，则将 0xb\*改成 0xb3。

4. 系统执行删除操作，并显示下面信息：

```
Deleted: IRP-1 URP-0 DRP-0 OTher-0
```

 说明

显示信息中的 IRP-1 表示已完成交易删除。

## 12.4 删除 xf1 文件中的某条 Rowid 记录

用户可以对 xf1 文件中某条 Rowid 记录进行删除。

### 前提条件

- SuperSync 已完成首次初始化，具体操作请参见 [3 首次初始化](#)。
- 源端和目标端 SuperSync 进程均启动，且运行正常。
- 使用 dsG 用户进行操作。

### 操作步骤

1. 进入 SuperSync 目标端 rmp/real\*目录下，查看需要进行操作的\*.xf1 文件的 Rowid 信息。

```
$ cd SuperSync 目标端复制目录/rmp/real*
```

```
$ ../../bin/xfview -show 1 *.xf1
```

 说明

- 当 SuperSync 只有一路复制装载进程时,在目标端 rmp 目录中只有 real0 文件夹。
- 当 SuperSync 有多路复制装载进程时,在目标端 rmp 目录中,将有多多个 real\*文件夹,如 SuperSync 有 3 路装载进程,则 rmp 目录中将存在 real0、real1、real2 三个文件夹。xf1 交易文件根据要求分别存放在这三个文件夹中。

2. 系统将显示该 xf1 文件的 Rowid 信息, 如:

```
2.xf1
CHECK_MEMLEAK is turned off
      : 00(NIL)
      : 000441BF.000441FB.0103CE26.0000 B3(DRP) --H-FL-- cc:3
len:22
      : 54(CMT), SCN 0x74fe.0a110001, Time 2014-01-26 16:01:45
-- Time: 0 seconds
```

 说明

000441BF.000441FB.0103CE26.0000 为 Oracle 的 objn.objd.dba.slot 信息,其中 objn=000441BF, objd=000441FB, dba=0103CE26, slot=0000。

3. 执行删除命令, 将该 xf1 文件中需要删除的 Rowid 记录进行删除。

```
$ ../../bin/xfview -delrowid objd dba slot *.xf1
```

 说明

在输入命令时, 请根据实际情况, 将 objd dba slot 处改成其对应的值, 如删除 2.xf1 文件中 objd=000441FB, dba=0103CE26, slot=0000 的交易记录, 则删除命令为:

```
$ ../../bin/xfview -delrowid 000441FB 0103CE26 0000
2.xf1
```

4. 系统执行删除操作, 并显示下面信息:

```
Deleted: IRP-1 URP-0 DRP-0 OTher-0
```

 说明

显示信息中的 IRP-1 表示已完成交易删除。

# 13 查看 SuperSync 日志

SuperSync 源端有代理日志 log.vagentd、发送日志 log.sender 和归档日志 archivelog；目标端有代理日志 log.vagentd、首次全同步装载日志 log.s\*、实时同步装载日志 log.r\* 及归档日志 archivelog。通过查看源端和目标端日志对分析数据复制过程及定位问题有着重要的作用。

本章节包含下列主题：

- 查看 SuperSync 源端日志
- 查看 SuperSync 目标端日志

## 13.1 查看 SuperSync 源端日志

介绍 SuperSync 源端 log.vagentd、log.sender 和 archivelog 日志的查看方法。

### 13.1.1 查看源端 log.vagentd 日志

查看源端 log 目录下的 log.vagentd 文件，可以查看源端分析到什么时候的交易。

#### 前提条件

使用 dsg 用户进行操作。

#### 操作步骤

1. 进入源端服务器 SuperSync 复制队列的 log 目录中，查看 log.vagentd 日志。

```
$ cd SuperSync 源端复制目录/log
```

```
$ tail -f log.vagentd
```

2. 系统将显示 log.vagentd 日志内容，如：

```
grp# size seq# bsz nab flg dup TH Next-SCN 2014/01/15
15:10:07
3 204800 891 512 174996 8 1 1 0xffffffff 891.174861.0
2014-01-15:15:10:07: processing rba 891.174861.0 (ckp: 891.174996.x) ...
2014-01-15:15:10:07: processed returned on rba 891.174996.0 (ckp:
891.174996.x)
SCN 0x67fb.1bd80001 in 2014-1-15:15:10:06 (U:0:0,C:0:0)
```



 **说明**

- 1) 2014-01-15:15:10:07: processing rba 891.174861.0 (ckp: 891.174996.x) ...  
  
2014-01-15:15:10:07: processed returned on rba 891.174996.0 (ckp: 891.174996.x)  
  
其中 2014-01-15:15:10:07 表示源端分析此交易的时间。
- 2) SCN 0x67fb.1bd80001 in 2014-1-15:15:10:06 (U:0:0,C:0:0) 中  
2014-1-15:15:10:06 表示源端数据库交易操作时间。
- 3) 源端分析交易时间-源端数据库交易时间=源端分析滞后的时间。
- 4) 日志不断更新，源端分析交易时间和源端数据库交易时间持续增加，则表示日志正常。
- 5) 如果日志报错、日志不继续更新或日志为空则表示分析可能有问题，请检查或报告 DSG 技术人员。

## 13.1.2 查看源端 log.sender 日志

查看源端 log 目录下的 log.sender 文件，查看源端日常传送的进度。

### 前提条件

使用 dsg 用户进行操作。

### 操作步骤

1. 进入源端服务器 SuperSync 复制队列的 log 目录中，查看 log.sender 日志。

```
$ cd SuperSync 源端复制目录/log
```

```
$ tail -f log.sender
```

2. 系统将显示 log.sender 日志内容，如：

```
2014-01-15:15:01:43: /dsg/supersync/ds/rmp/2.0.xf1(18) (PID-20974  
PPID-20972) ...
```

 **说明**

- 1) 2014-01-15:15:01:43 表示源端日常传送此 xf1 文件的系统时间。
- 2) /dsg/supersync/ds/rmp/2.0.xf1 表示源端传送此 xf1 文件的具体编号。
- 3) (18) 表示源端发送此 xf1 文件的大小。
- 4) 日志不断更新，并且传送的文件数字不断增加，没有出现重试的错误信息表示源端传送 xf1 正常。
- 5) 如果日志报错或为空表示可能传送有问题，请检查或报告 DSG 技术人员。

### 13.1.3 查看源端 archive log 日志

源端 log/archive log 目录中存放着 log.vagentd 和 log.sender 的归档日志，用户可以查看该目录下的归档日志，用于查看或分析已完成的数据复制过程。

#### 前提条件

使用 dsg 用户进行操作。

#### 操作步骤

1. 进入源端服务器 SuperSync 复制队列的 log/archive log 目录中，查看该目录下已存在的归档文档。

```
$ cd SuperSync 源端复制目录/log/archive log
```

```
$ ls
```

2. 系统显示该目录下的归档日志。

```
log.sender_archlog.131223_105739 log.vagentd_archlog.131223_114648
```

#### 说明

- log.sender\_archlog.131223\_105739 为 log.sender 的归档日志，131223\_105739 为归档日志时间戳，表示归档时间为 2013 年 12 月 23 日 10 点 57 分 39 秒。
  - log.vagentd\_archlog.131223\_114648 为 log.vagentd 归档日志，131223\_114648 为归档日志时间戳，表示归档时间为 2013 年 12 月 23 日 11 点 46 分 48 秒。
3. 查看归档日志。

```
$ vi 归档日志
```

#### 说明

例如查看的归档日志是 log.sender\_archlog.131223\_105739，则应输入查看命令：

```
$ vi log.sender_archlog.131223_105739
```

## 13.2 查看 SuperSync 目标端日志

SuperSync 目标端 log 目录下包含 log.vagentd、log.s\*、log.r\* 和归档日志 archive log，通过这些日志，可以实时查看目标端日常数据接收进度、数据首次全同步装载和实时装载进度等。

### 13.2.1 查看目标端 log.vagentd 日志

查看目标端 log 目录下的 log.vagentd 文件，查看目标端日常接收的进度。

## 前提条件

使用 dsg 用户进行操作。

## 操作步骤

1. 进入目标端服务器 SuperSync 复制队列的 log 目录中，查看 log.vagentd 日志。

```
$ cd SuperSync 目标端复制目录/log
```

```
$ tail -f log.vagentd
```

2. 系统显示 log.vagentd 日志内容，如：

```
2014-01-15:15:03:05: full, q# 0, XF1 523 ... (303.xf1) elapsed 466ms
2014-01-15:15:03:07: full, q# 0, XF1 21 ... (304.xf1) elapsed 159ms
2014-01-15:15:03:07: full, q# 0, XF1 21 ... (305.xf1) elapsed 332ms
2014-01-15:15:03:12: real, q# 0, XF1 18 ... (0.xf1) elapsed 1960ms
2014-01-15:15:03:16: real, q# 0, XF1 18 ... (1.xf1) elapsed 511ms
```

### 说明

- 1) full 表示全同步复制，real 表示实时复制。
- 2) 2014-01-15:15:03:05 表示目标端日常接收此 xf1 文件的系统时间。
- 3) q# 0 表示目标端日常接收此 xf1 文件的所在队列，多队列分 0, 1, 2 等。
- 4) XF1 523 后面的 523 表示目标端日常接收此 xf1 文件的大小。
- 5) elapsed 466ms 表示目标端日常接收此 xf1 文件花费的时间。
- 6) log.vagentd 日志不断更新，并且时间持续增加表示目标端接收 xf1 正常。
- 7) log.vagentd 日志不更新、报错或日志内容为零则表示目标端接收 xf1 可能有问题，请检查或报告 DSG 技术人员。

## 13.2.2 查看目标端 log.s\* 日志

查看目标端 log 目录下的首次同步装载日志 log.s\* 文件，如果目标端仅为 1 路同步装载，则只需要查看 log.s0，如果目标端为多路并发装载，则需要同时查看 log.s0、log.s1 等多个文件。

## 前提条件

使用 dsg 用户进行操作。

## 操作步骤

1. 进入目标端服务器 SuperSync 复制队列的 log 目录中，查看 log.s\* 日志。

```
$ cd SuperSync 目标端复制目录/log
```

```
$ tail -f log.s*
```

 说明

- 如果目标端为 1 路同步装载，则只需要查看 log.s0 文件：

**\$ tail -f log.s0**

- 如果目标端为多路并发装载，则需要查看多个日志文件，如 3 路并发装载，则需要查看 log.s0、log.s1、log.s2。

**\$ tail -f log.s0**

**\$ tail -f log.s1**

**\$ tail -f log.s2**

2. 系统显示 log.s\* 日志信息，当系统显示下列信息时，表示目标端首次同步装载文件装载完成。

```
Loader: 2001.xf1 2014-02-12:15:32:36: 0-I 0-D 0-U 0-DDL(0 ms) (0 s)
to compile drop constraint
compile num=0
compile try 1 ...
all compile finish
```

 说明

compile num=0 表示显示的是第一路装载，如果查看 log.s1 文件时，该参数值将为 1。

## 13.2.3 查看目标端 log.r\* 日志

查看目标端 log 目录下的实时装载日志 log.r\* 文件，如果目标端仅为 1 路实时装载，则只需要查看 log.r0，如果目标端为多路实时装载，则需要同时查看 log.r1、log.r2 等多个文件。

### 前提条件

使用 dsg 用户进行操作。

### 操作步骤

1. 进入目标端服务器 SuperSync 复制队列的 log 目录中，查看 log.r\* 日志。

**\$ cd SuperSync 目标端复制目录/log**

**\$ tail -f log.r\***

 说明

- 如果目标端为 1 路实时装载，则只需要查看 log.r0 文件：

**\$ tail -f log.r0**

- 如果目标端为多路实时装载，则需要查看多个日志文件，如 3 路实时装载，则需要查看 log.r0、log.r1、log.r2。

```
$ tail -f log.r0
```

```
$ tail -f log.r1
```

```
$ tail -f log.r2
```

2. 系统显示 log.r\* 日志信息，如：

```
Loader: 1.xf1 real sync data in first sync loading
2014-02-12:15:33:39: 19556-I 0-D 0-U 0-DDL(2086 ms) (2 s)

Loader: 2.xf1 real sync data in first sync load finish
2014-02-12:15:33:39: 19556-I 0-D 0-U 0-DDL(2086 ms) (2 s)
```

#### 说明

- 1) Loader: 1.xf1 和 Loader: 2.xf1 表示目标端装载相应 xf1 文件的编号。
- 2) 2014-02-12:15:33:39 表示目标端装载该 xf1 文件的系统时间。
- 3) 19556-I 0-D 0-U 0-DDL 表示该 xf1 文件的具体操作类型及对应个数。
- 4) (2086 ms) (2 s) 表示装载该 xf1 花费的时间。
- 5) 日志不断更新，并且装载的时间不断增加，没有出现错误信息则表示目标端装载 xf1 正常。
- 6) 如果日志不更新、报错或为空表示可能装载有问题，请检查或报告 DSG 技术人员。

## 13.2.4 查看目标端 archivelog 日志

目标端 log/archivelog 目录中存放着 log.vagentd、log.s\* 和 log.r\* 的归档日志，用户可以查看该目录下的归档日志，用于查看或分析已完成的数据复制过程。

### 前提条件

使用 dsg 用户进行操作。

### 操作步骤

1. 进入目标端服务器 SuperSync 复制队列的 log/archivelog 目录中，查看该目录下已存在的归档文档。

```
$ cd SuperSync 目标端复制目录/log/archivelog
```

```
$ ls
```

2. 系统显示该目录下的归档日志。

```
log.r0_archlog.140114_162456 log.r0_errlog.140130_034136
```

```
log.s0_archlog.140116_101032 log.s0_errlog.140130_102014
log.vagentd_archlog.140124_034135 log.vagentd_errlog.140122_110612
```

### 说明

- log.r0\_archlog.140114\_162456 是 log.r0 的归档日志，140114\_162456 为归档日志时间戳，表示归档时间为 2014 年 1 月 14 日 16 点 24 分 56 秒。
- log.r0\_errlog.140130\_034136 是 log.r0 的错误信息归档日志，140130\_034136 为归档时间戳。
- log.s0\_archlog.140116\_101032 是 log.s0 的归档日志，140116\_101032 为归档时间戳。
- log.s0\_errlog.140130\_102014 是 log.s0 的错误信息归档日志，140130\_102014 为归档时间戳。
- log.vagentd\_archlog.140124\_034135 为 log.vagentd 归档日志，140124\_034135 为归档日志时间戳。
- log.vagentd\_errlog.140122\_110612 为 log.vagentd 的错误信息归档日志，140122\_110612 为归档日志时间戳。

### 3. 查看归档日志。

#### **\$ vi** 归档日志

### 说明

例如查看的归档日志是 log.r0\_archlog.140114\_162456，则应输入查看命令：

```
$ vi log.r0_archlog.140114_162456
```

## 14 查看 SuperSync 版本信息

用户可以查看 SuperSync 版本信息。

### 前提条件

使用 dsg 用户进行操作。

### 操作步骤

- 源端查看 SuperSync 版本信息
  1. 进入 SuperSync 源端复制目录的 bin 目录中，执行查看操作。  
**\$ cd SuperSync 源端复制目录/bin**  
**\$ ./vagentd -v**
  2. 系统显示 SuperSync 版本信息，如：

```

RealSync agent 6.2.3.129 (11.2.0.1.0)(2015072910)
RealSync for oracle version 11.2.0.1.0

Module: Analyzer      RealSync 6.2.3.129
Module: Synthesizer  RealSync 6.2.3.129
Module: Loader        RealSync 6.2.3.129
Module: DDL           RealSync 1.0
Module: RealTimer     RealSync 6.2.3.129

Support for LOB(CLOB,BLOB) datatype. (LOB_SUPPORT)
Support for PK/UK MODE. (UK_MODE_SUPPORT)
Support for LOB data dynamic analyzing. (REDO_LOB)
Support for limit max open 100 files. (LIMIT_MAX_OPEN_FILE)
  
```

### 说明

- RealSync agent6.2.3.129 (11.2.0.1.0)(2015072910)指 SuperSync 的版本为 6.2.3.129, 支持的 Oracle 版本为 11.2.0.1.0, 2015072910 为 SuperSync 版本生成的时间是 2015 年 7 月 29 日 10 点。
- Module: Analyzer RealSync 6.2.3.129

Module: Synthesizer RealSync 6.2.3.129

Module: Loader RealSync 6.2.3.129

Module: DDL RealSync 1.0

Module: RealTimer RealSync 6.2.3.129

显示 SuperSync 中具体模块的版本号信息, Analyzer 为日志分析模块、Synthesizer 为交易合成模块、Loader 为交易装载模块、DDL 为 ddl 过滤分析模块、RealTimer 为复制时间模块。

- Support for LOB(CLOB,BLOB) datatype. (LOB\_SUPPORT)
- Support for PK/UK MODE. (UK\_MODE\_SUPPORT)
- Support for LOB data dynamic analyzing. (REDO\_LOB)
- Support for limit max open 100 files. (LIMIT\_MAX\_OPEN\_FILE)

表示该版本 SuperSync 支持 CLOB 和 BLOB 数据类型、主键和唯一键复制模式、LOB 数据动态分析及最大可以打开 100 个文件功能。

- 目标端查看 SuperSync 版本信息
  1. 进入 SuperSync 目标端复制目录的 bin 目录中, 执行查看操作。

**\$ cd SuperSync 目标端复制目录/bin**

**\$ ./vagentd -v**

2. 系统显示 SuperSync 版本信息, 如:

```
RealSync agent 6.2.3.129 (11.2.0.1.0)(2015072910)
RealSync for oracle version 11.2.0.1.0

Module: Analyzer RealSync 6.2.3.129
Module: Synthesizer RealSync 6.2.3.129
Module: Loader RealSync 6.2.3.129
Module: DDL RealSync 1.0
Module: RealTimer RealSync 6.2.3.129

Support for LOB(CLOB,BLOB) datatype. (LOB_SUPPORT)
Support for PK/UK MODE. (UK_MODE_SUPPORT)
Support for LOB data dynamic analyzing. (REDO_LOB)
Support for ASM. (ASM_SUPPORT)
Support for limit max open 25 files. (LIMIT_MAX_OPEN_FILE)
```





- RealSync agent6.2.3.129 (11.2.0.1.0)(2015072910)指 SuperSync 的版本为 6.2.3.129, 支持的 Oracle 版本为 11.2.0.1.0, 2015072910 为 SuperSync 版本生成的时间是 2015 年 7 月 29 日 10 点。
- Module: Analyzer RealSync 6.2.3.114.ASM  
Module: Synthesizer RealSync 6.2.3.114.ASM  
Module: Loader RealSync 6.2.3.114.ASM  
Module: DDL RealSync 1.0  
Module: RealTimer RealSync 6.2.3.114.ASM

显示 SuperSync 中具体模块的版本号信息, Analyzer 为日志分析模块、Synthesizer 为交易合成模块、Loader 为交易装载模块、DDL 为 ddl 过滤分析模块、RealTimer 为复制时间模块。

- Support for LOB(CLOB,BLOB) datatype. (LOB\_SUPPORT)  
Support for PK/UK MODE. (UK\_MODE\_SUPPORT)  
Support for LOB data dynamic analyzing. (REDO\_LOB)  
Support for ASM. (ASM\_SUPPORT)  
Support for limit max open 25 files. (LIMIT\_MAX\_OPEN\_FILE)

表示该版本 SuperSync 支持 CLOB 和 BLOB 数据类型、主键和唯一键复制模式、LOB 数据动态分析、支持 Oracle 的 ASM 及最大可以打开 25 个文件功能。