

数据复制

1	触发器方式非实时单项复制.....	2
1.1	Topology	2
1.1.1	数据服务器.....	2
1.1.2	数据源和目标物理架构.....	2
1.1.3	上下文.....	2
1.1.4	数据源和目标逻辑架构.....	2
1.1.5	代理.....	3
1.2	模型和知识模块导入.....	3
1.2.1	模型沿用 ODI Tutorial 中创建的 oracle_sh 和 oracle_tgtwh	3
1.2.2	数据存储需要增加.....	3
1.2.3	沿用 ODI Tutorial 中导入的知识模块	3
1.3	配置和启动 CDC 以及 Subscriber.....	3
1.3.1	针对数据源模型配置日记记录（选择合适的 JKM）	3
1.3.2	将需要进行 CDC 的数据存储添加到 CDC.....	4
1.3.3	添加订阅.....	5
1.3.4	启动日记.....	6
1.4	接口.....	7
1.4.1	新建一个接口如 “TGT_CHANNELS_CDC_Int”.....	7
1.4.2	关系图.....	7
1.4.3	流.....	8
1.4.4	控件.....	8
1.5	包.....	9
1.5.1	新建包如 “TGT_CHANNELS_CDC_PKG”	9
1.5.2	关系图.....	9
1.6	生成方案.....	9
1.7	配置计划.....	9
1.8	启动 Schedule Agent.....	9
1.9	模拟更改数据源数据，检测结果.....	9
2	触发器方式实时单项复制.....	9
2.1	包.....	10
2.1.1	新建包如 TGT_CHANNNLES_CDC_RT.....	10
2.1.2	关系图.....	10
2.2	无需配置包的计划，直接执行包就可以了	12
3	删除日记（CDC）	12
4	LogMiner 方式非实时单项复制.....	12
4.1	使用 LogMiner 异步方式的前提.....	12
4.1.1	数据库必须启用 ARCHIVELOG.....	12
4.1.2	为进行 CDC 操作的用户授权.....	12
4.2	订户和日记配置和启动.....	12
4.2.1	针对数据源模型配置日记记录（选择合适的 JKM）	13
4.2.2	将需要进行 CDC 的数据存储添加到 CDC.....	14

4.2.3	添加订阅.....	14
4.2.4	启动日记.....	14
4.3	接口.....	14
4.4	包.....	14
4.4.1	新建包如 TGT_CHANNLES_LOGMINER_PKG.....	14
4.4.2	关系图.....	14
4.5	其余步骤参照 1 触发器方式非实时单项复制.....	18
5	LogMiner 方式实时单项复制.....	18
5.1	参照 4LogMiner 方式非实时单项复制设置 JKM.....	18
5.2	接口.....	18
5.3	包.....	18
5.3.1	新建包如 TGT_CHANNLES_LOGMINER_RT.....	18
5.3.2	关系图.....	18

1 触发器方式非实时单项复制

1.1 Topology

1.1.1 数据服务器

延用 ODI Tutorial 中创建的 Oracle 的数据服务器 orcl_local

1.1.2 数据源和目标物理架构

延用 ODI Tutorial 中创建的数据源物理架构 orcl_local.sh

延用 ODI Tutorial 中创建的目标物理架构 orcl_local.tgt_wh

1.1.3 上下文

延用 ODI Tutorial 中的上下文 global

1.1.4 数据源和目标逻辑架构

延用 ODI Tutorial 中创建的数据源逻辑架构 orcl_src_sh

延用 ODI Tutorial 中创建的目标逻辑架构 orcl_tgt_wh

1.1.5 代理

延用 ODI Tutorial 中创建的物理代理 local_agent

延用 ODI Tutorial 中创建的逻辑代理 local_logicalagent

1.2 模型和知识模块导入

1.2.1 模型沿用 ODI Tutorial 中创建的 oracle_sh 和 oracle_tgtwh

1.2.2 数据存储需要增加

反向导入一个数据源数据存储 CHANNELS

反向导入一个目标数据存储 TGT_CHANNELS

1.2.3 沿用 ODI Tutorial 中导入的知识模块

1.3 配置和启动 CDC 以及 Subscriber

1.3.1 针对数据源模型配置日记记录（选择合适的 JKM）

某个数据源模型如 oracle_sh 的定义框，选择“日记记录”标签，进行如下配置：

- 日记记录模式

一致性适用于多表之间有关系（如主外键等），对多表进行 CDC。

简单适用于单表或者虽然是多表，但是表之间的 CDC 没有顺序问题。

此处我们选择简单

- JKM

选择 JKM Oracle Simple（触发器方式）



1.3.2 将需要进行 CDC 的数据存储添加到 CDC

选中需要添加到 CDC 的数据存储如 CHANNELS 添加到 CDC。



数据存储添加到 CDC 成功后，“刷新”Designer，可以发现相应的数据存储前面加了一个黄色的钟。



1.3.3 添加订阅

选中数据存储如 CHANNELS，右键，已更改数据捕获->订户->订阅
输入订户名称如 **subscriber_sh**（在后面接口中条件设置上要使用），添加，确定。



确定，运行订户。

1.3.4 启动日记

数据存储如 CHANNELS，右键，“已更改数据捕获”->“启动日记”，启动日记成功后，刷新 Designer，数据存储前面的钟变为绿色。



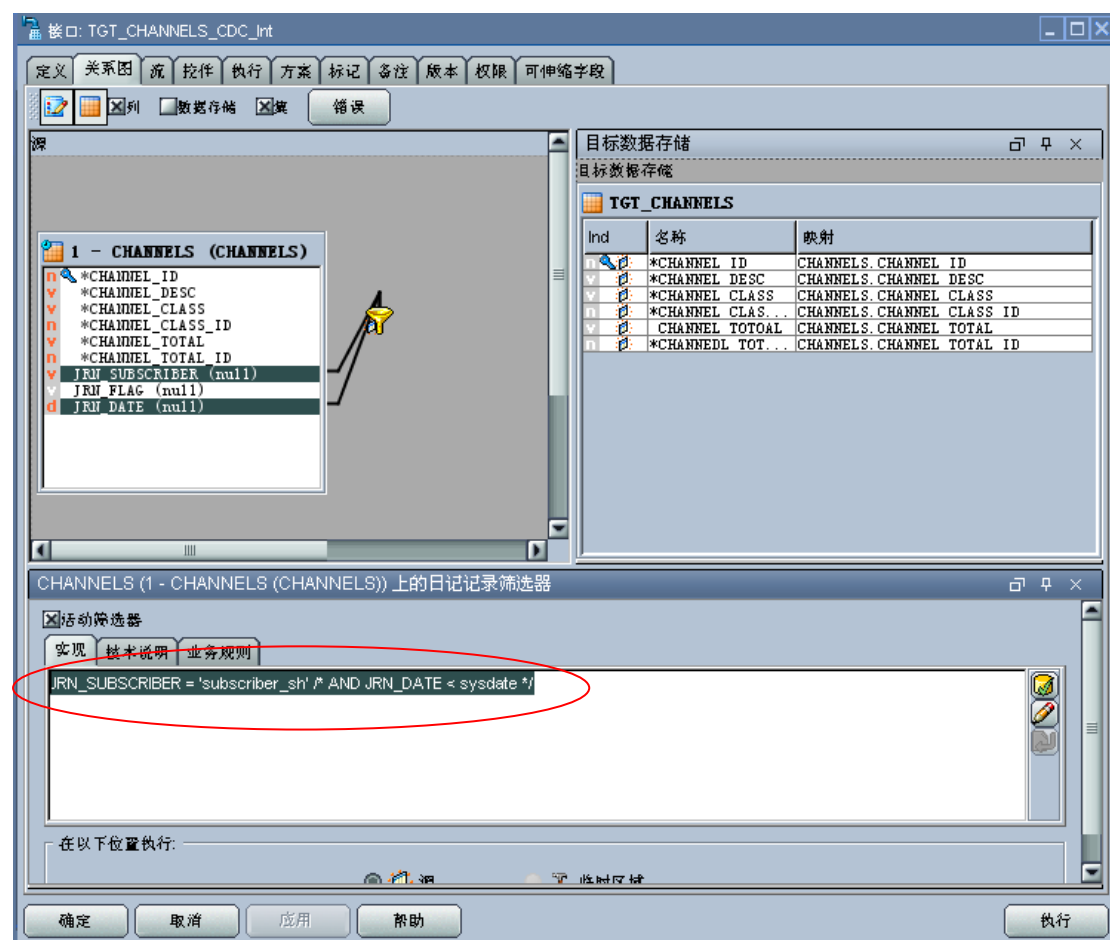
1.4 接口

1.4.1 新建一个接口如 “TGT_CHANNELS_CDC_Int”

1.4.2 关系图

从模型托拽数据源数据存储如 TGT_CHANNELS 到数据源，注意选择“仅已进行日记记录的数据”

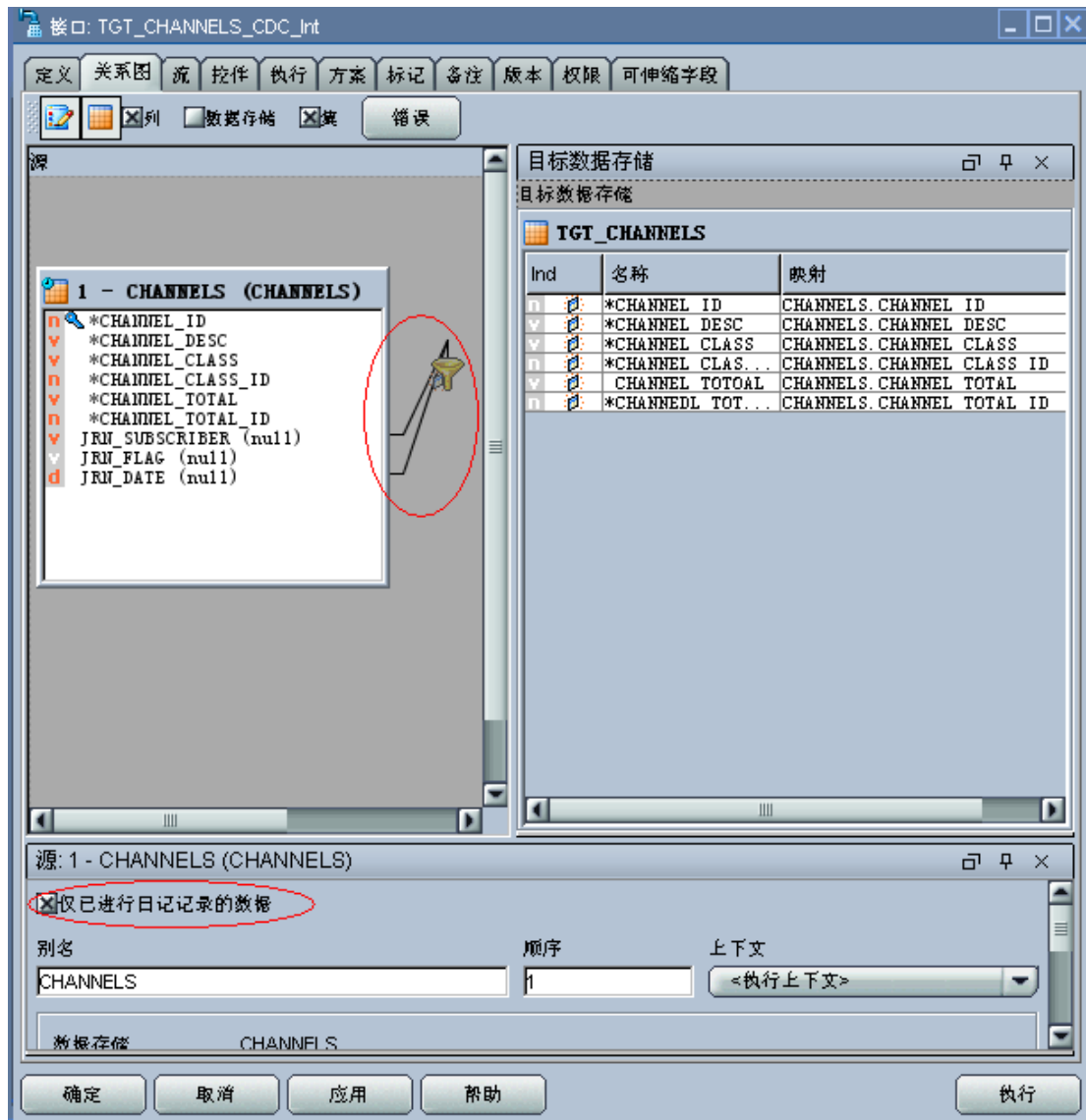
注意，源中自动添加的条件需要进行编辑：JRN_SUBSCRIBER = 'subscriber_sh' /* AND JRN_DATE < sysdate */。其中的 JRN_SUBSCRIBER 需要是前面设置的订户名称。



托拽目标数据存储如 TGT_CHANNELS 到目标数据存储。

进行字段自动匹配。

注意对于目标数据存储，需要设定键 **CHANNEL_ID**



1.4.3 流

选择缺省的 IKM Oracle Incremental Update，选项为缺省值。

1.4.4 控件

选择缺省的 CKM Oracle，选项为缺省值。

1.5 包

1.5.1 新建包如 “TGT_CHANNELS_CDC_PKG”

1.5.2 关系图

包中只包含一个步骤：接口 TGT_CHANNELS_CDC_Int

1.6 生成方案

针对包 TGT_CHANNELS_CDC_PKG 生成方案。

1.7 配置计划

配置计划，如每小时运行一次。

1.8 启动 Schedule Agent

1.9 模拟更改数据源数据，检测结果

利用 sh 用户进入 SQLPLUS，
插入、删除和更新 CHANNELS 的数据。
然后到 Operator 中查看包是否运行，如果已经运行
再进入 SQLPLUS，以 TGT_WH 用户连接，
查看 TGT_CHANNELS 表中是否已经有了更新数据。

2 触发器方式实时单项复制

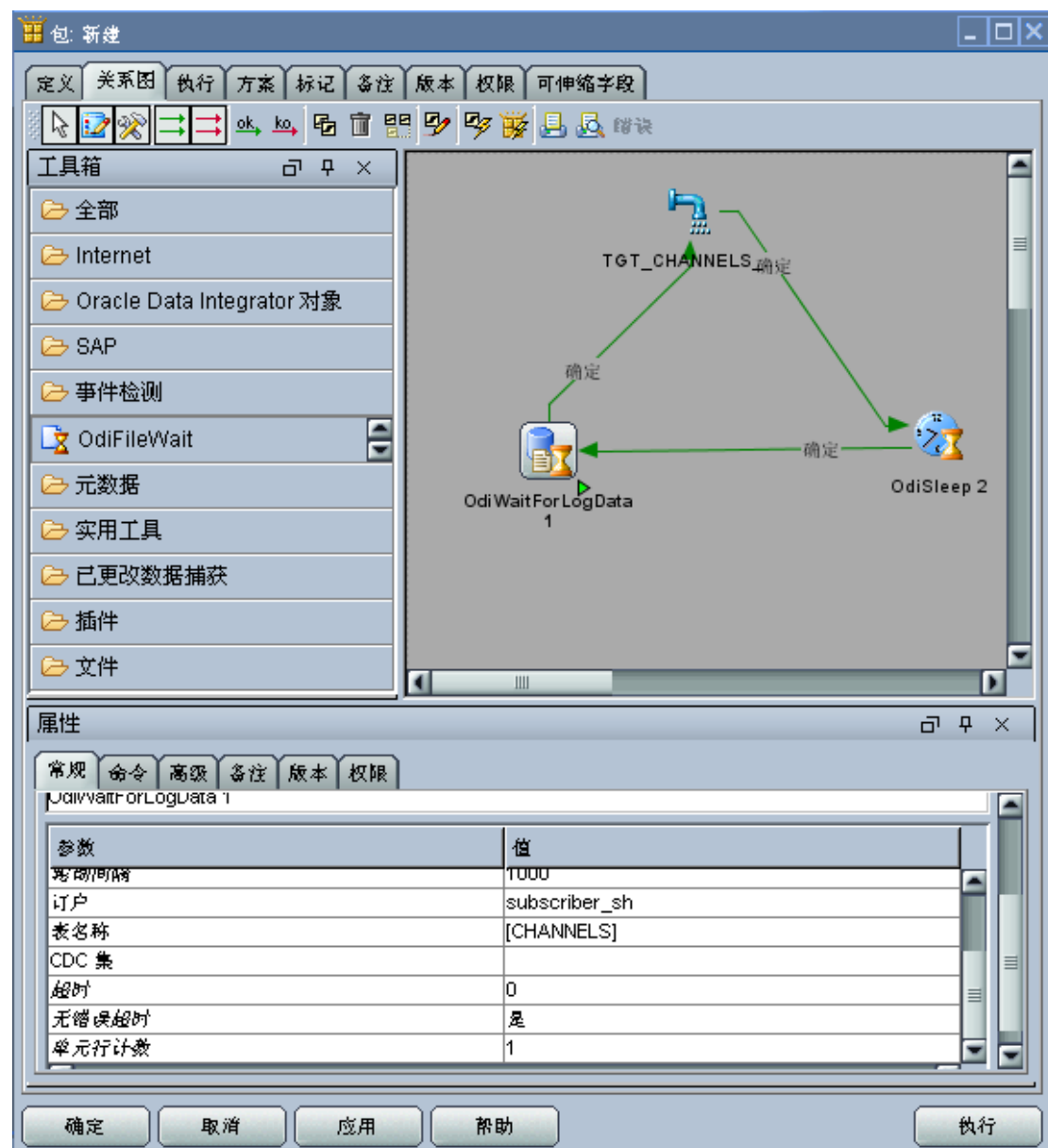
只有包的设计与 1 触发器方式非实时单项复制不同，其他相同。下面只是说明如何设计包。

2.1 包

2.1.1 新建包如 TGT_CHANNELS_CDC_RT

2.1.2 关系图

- 创建的包中，包含三个步骤：OdiWaitForLogData（在工具箱的已更改数据捕获中），实现单项复制的接口，OdiSleep（在工具箱的事件检测中）。如下图：



要注意：需要将 OdiWaitForLogData 设置成“第一步”

- 配置 OdiWaitForLogData

OdiWaitForLogData 将轮询数据源，发现变化数据。如果发现了变化数据则会进入下一步。

要配置的属性主要包括：

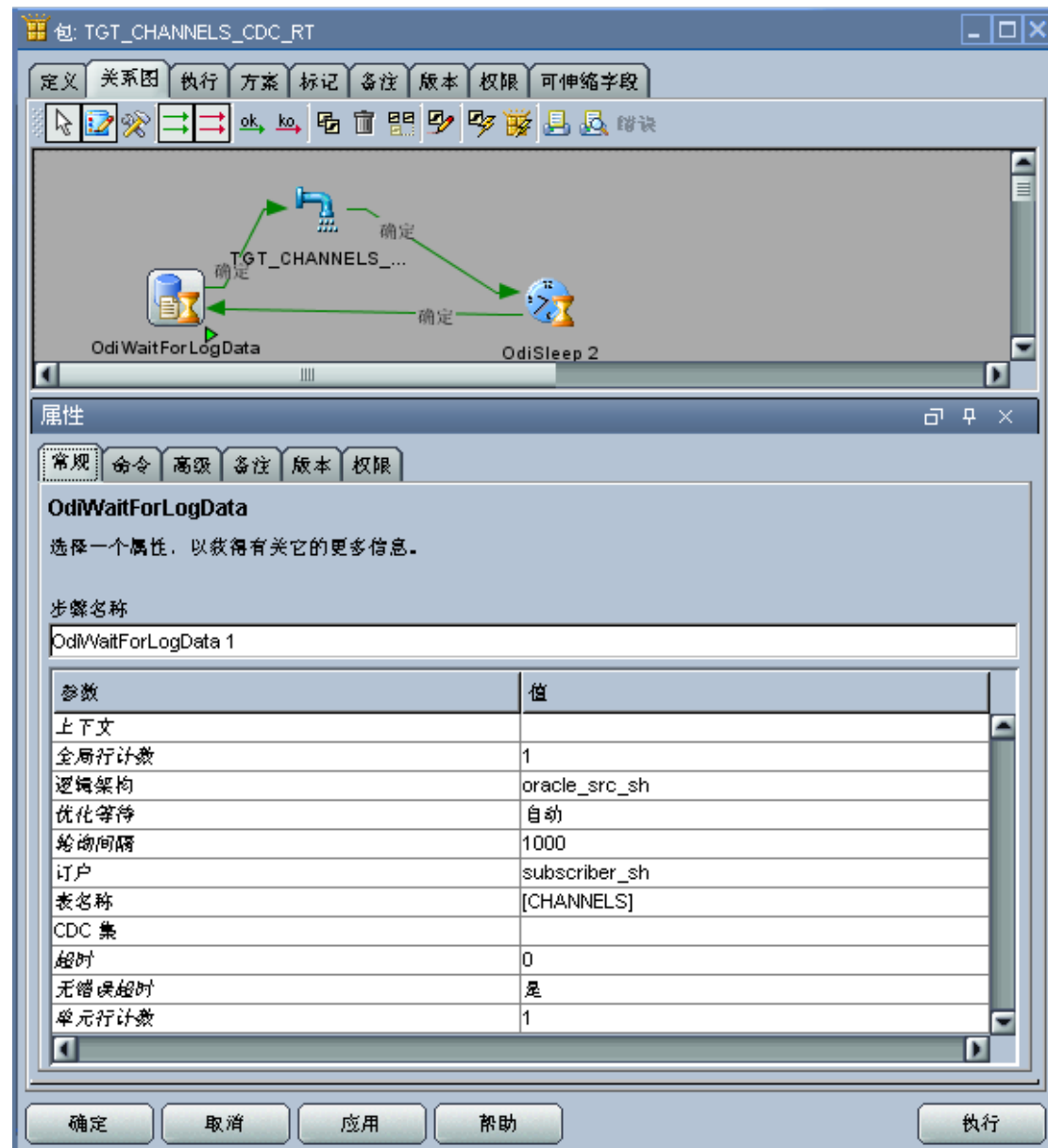
逻辑架构：数据源表所对应的逻辑架构

轮询间隔：每隔多长时间轮询一次。例子中间隔是 5000 毫秒。

订户：CDC 的订户名称。

表名称：数据源表。这个表应该启动了 CDC 日志

“单元行计数”和“全局行计数”：可以定义必须要发生多少行变化才进行同步。在例子中，一旦有变化就同步。



● 配置 OdiSleep

OdiSleep 使得执行完接口之后可以等待一段时间再开始轮询。主要配置的参数有：

等待延迟：等待的毫秒数。例如 5000 毫秒，或者 180000 表示 30 分钟。

2.2 无需配置包的计划，直接执行包就可以了

3 删除日记（CDC）

为了下一步操作，删除日记。

Designer->模型->某个数据源模型如 oracle_sh，右键->已更改数据捕获->删除日记。

4 LogMiner 方式非实时单项复制

与 1 触发器方式非实时单项复制的唯一区别是模型的日记记录的配置不同，也就是选择不同 JKM 实现变化数据捕获。

4.1 使用 LogMiner 异步方式的前提

4.1.1 数据库必须启用 ARCHIVELOG

4.1.2 为进行 CDC 操作的用户授权

进行 CDC 操作的用户需要具有下面的权限：

```
GRANT SELECT ON CHANGE_SETS TO :user;
```

```
GRANT EXECUTE on DBMS_CDC_PUBLISH TO :user;
```

```
GRANT CREATE SESSION TO :user;
```

```
GRANT CREATE TABLE TO :user;
```

```
GRANT CREATE TABLESPACE TO :user;
```

```
GRANT UNLIMITED TABLESPACE TO :user;
```

```
GRANT SELECT_CATALOG_ROLE TO :user;
```

```
GRANT EXECUTE_CATALOG_ROLE TO :user;
```

```
GRANT CREATE SEQUENCE TO :user;
```

```
GRANT DBA TO :user;
```

```
GRANT EXECUTE on DBMS_CDC_PUBLISH TO :user;
```

```
EXECUTE DBMS_STREAMS_AUTH.GRANT_ADMIN_PRIVILEGE(GRANTEE =>
:user');
```

4.2 订户和日记配置和启动

“JKM Oracle 10g Consistent (LOGMINER)” 利用 Oracle10g 内置的 CDC 实现变化数据捕获。 Oracle10g 的 CDC 支持同步和异步捕获两种模式。其中的异步模式是通过

LOGMINER 实现数据捕获的, 相对于 TRIGGER 的方式而言, 对数据源系统性能影响较小。
请按照下面的步骤来使用这个 JKM。

4.2.1 针对数据源模型配置日记记录（选择合适的 JKM）

某个数据源模型如 oracle_sh 的定义框, 选择“日记记录”标签, 进行如下配置:

- 日记记录模式

一致性适用于多表之间有关系（如主外键等），对多表进行 CDC。

简单适用于单表或者虽然是多表，但是表之间的 CDC 没有顺序问题。

此处我们选择一致性

- 选择“JKM Oracle 10g Consistent (LOGMINER)”作为 JKM，然后将 **ASYNCHRONOUS_MODE** 配置为“是”。如下图所示：

模型: oracle_sh

已进行日记记录的表 标记 服务 备注 版本 权限 可伸缩字段

定义 反向 选择性反向 控制 日记记录

日记记录模式

☒ 一致性设置 ☐ 简单

日记记录 KM

选择 KM

JKM Oracle 10g Consistent (LOGMINER).Training_Project

选项	值
ASYNCHRONOUS_MODE	是
AUTO_CONFIGURATION	<默认>: 否
JOURNAL_TABLE_OPTIONS	<默认>:

If selected, the JKM will minimize the overhead on the source system by using the Asynchronous HotLog method in Oracle Change Data Capture.

Caution: using this option requires setting Oracle parameters including archivelog and force logging on the database. Please carefully review Oracle documentation on this topic before using these features (Data Warehousing Guide - Change Data Capture).

订户列表

确定 取消 应用 帮助 反向

4.2.2 将需要进行 CDC 的数据存储添加到 CDC

选中需要添加到 CDC 的数据存储如 CHANNELS 添加到 CDC。

4.2.3 添加订阅

选中数据存储如 CHANNELS，右键，已更改数据捕获->订户->订阅

输入订户名称如 **subscriber_sh**（在后面接口中条件设置上要使用），添加，确定。

4.2.4 启动日记

数据存储如 CHANNELS，右键，“已更改数据捕获”->“启动日记”，

启动日记成功后，刷新 Designer，数据存储前面的钟变为绿色。



4.3 接口

参照 1 触发器方式非实时单项复制中接口创建和设计既可

4.4 包

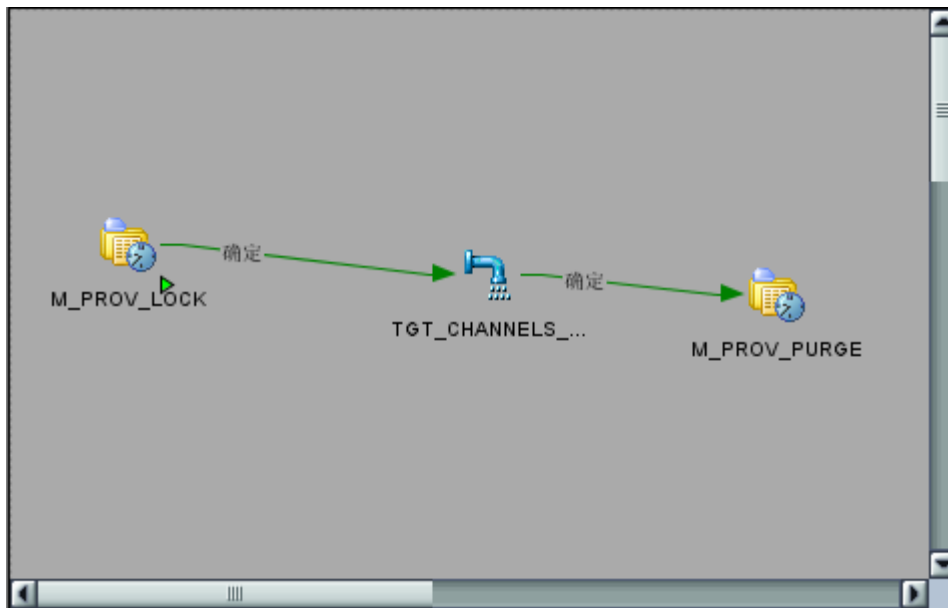
利用 Log Miner 方式实现的包和触发器方式相比有所不同，主要是添加了两个步骤：

一个步骤完成数据扩展和订户锁定，第二个步骤完成清除日记和取消订户锁定。具体操作步骤如下：

4.4.1 新建包如 TGT_CHANNELS_LOGMINER_PKG

4.4.2 关系图

PACKAGE 的内容如下图：



主要包括三个步骤：

- M_PROV_LOCK 是将需要同步的模型如 oracle_sh 拖到包中来。用来扩展窗口和锁定订户。下图是 M_PROV_LOCK 的例子。

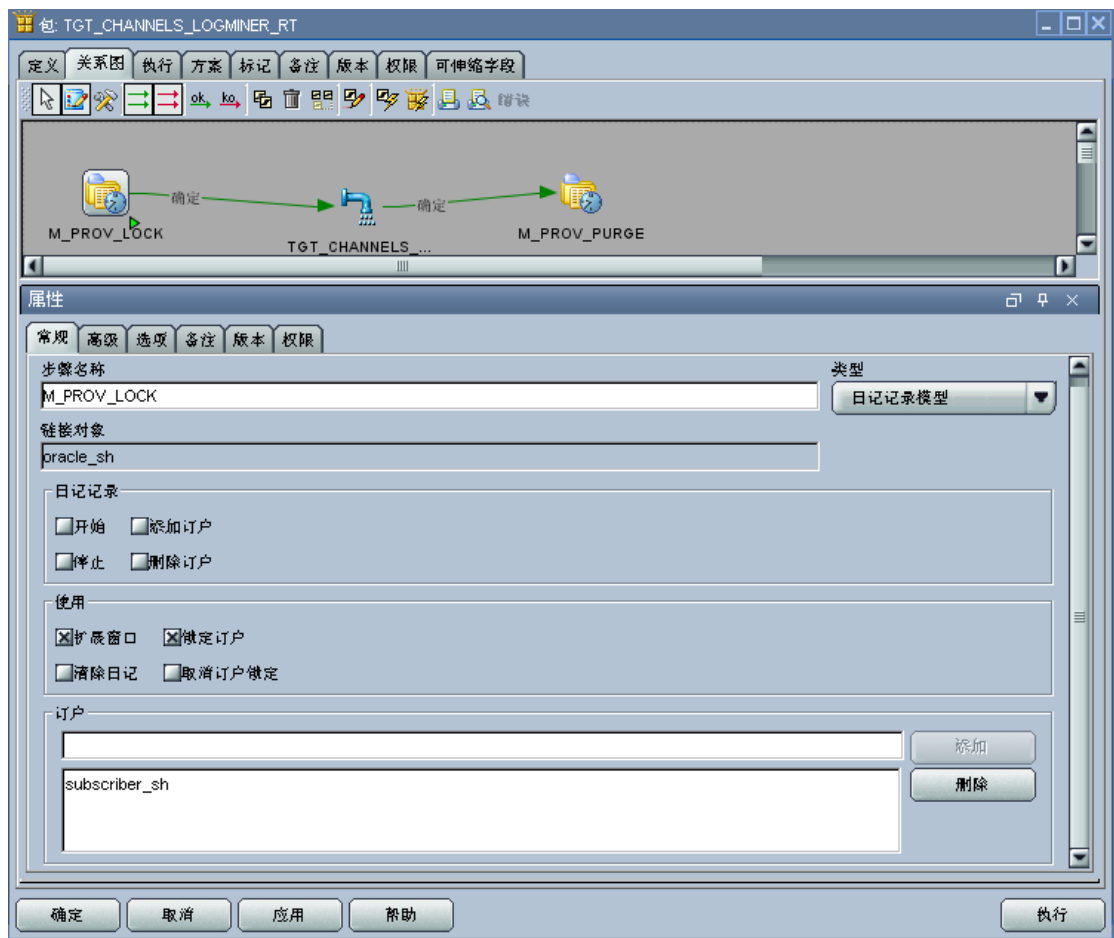
主要配置了下面的信息：

步骤名称：M_PROV_LOCK

类型：选择“日记记录模型”

使用：选上扩展窗口和锁定订户。

订户：添加要同步的订户。



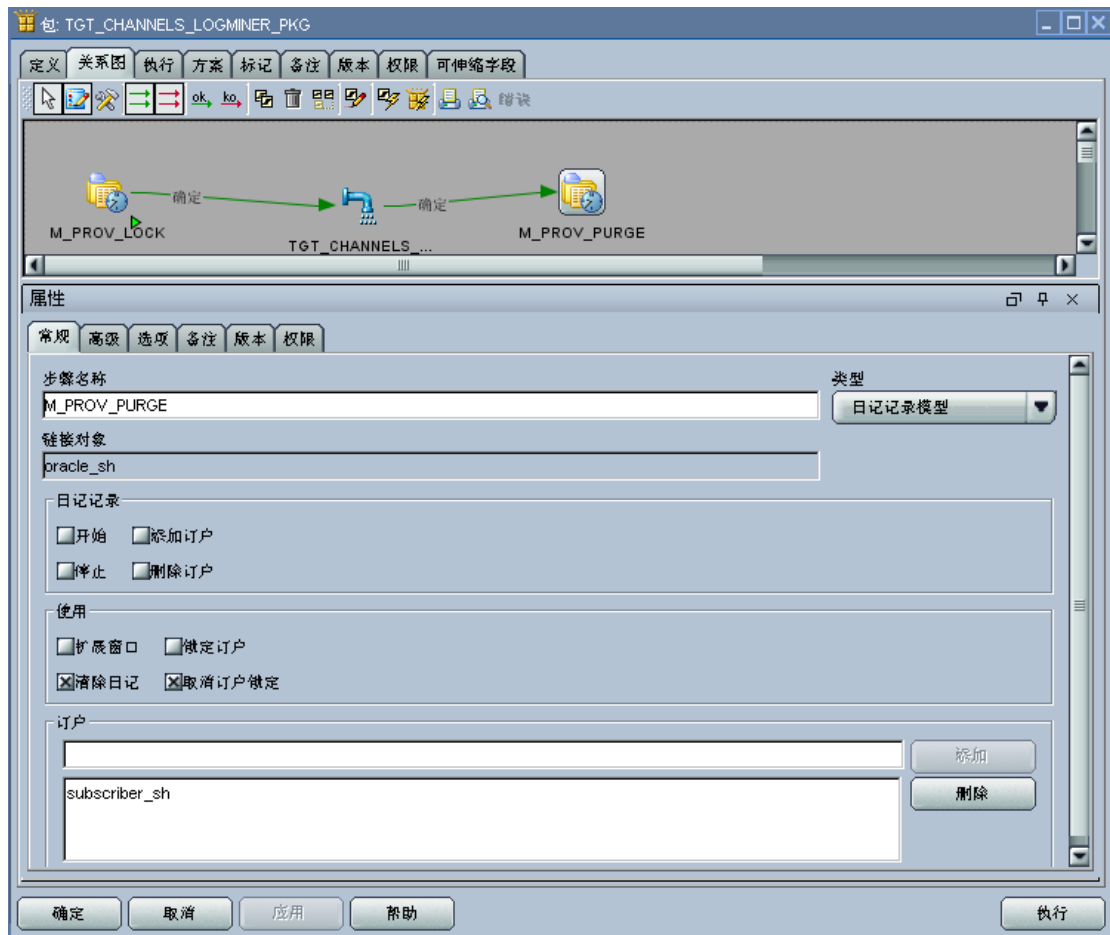
- 将变化数据同步到目标表的接口如 TGT_CHANNELS_CDC_Int
- M_PROV_PURGE 是将需要同步的模型如 oracle_sh 拖到包中来。用来清除日记和取消订户锁定。下图是 M_PROV_PURGE 的例子。

主要配置了下面的信息：

类型：选择“日记记录模型”

使用：选上清除日记和取消订户锁定。

订户：添加要同步的订户。



-
- WaitData-City1 和 WaitData-City2 分别用来判断是否存在 City1 和 City2 的变化数据。是工具箱中的 OdiWaitForData 命令。下面是一个例子。

主要配置了下面的信息：

全局行计数：等待多少行变化数据

逻辑架构：模型所属的逻辑架构的名称

轮询间隔：隔多长时间查询一次。例子中代表 50 毫秒

SQL 筛选器：WHERE 子句。

对象类型：选择 V(代表视图)

表名称：日志视图的名称。可以从 SNP_CDC_OBJECTS 查询到

超时：50 毫秒

无错误超时：否

如果成功则调用接口进行数据复制，如果失败则不调用接口。

4.5 其余步骤参照 1 触发器方式非实时单项复制

5 LogMiner 方式实时单项复制

5.1 参照 4LogMiner 方式非实时单项复制设置 JKM

5.2 接口

参照 1 触发器方式非实时单项复制中接口创建和设计既可

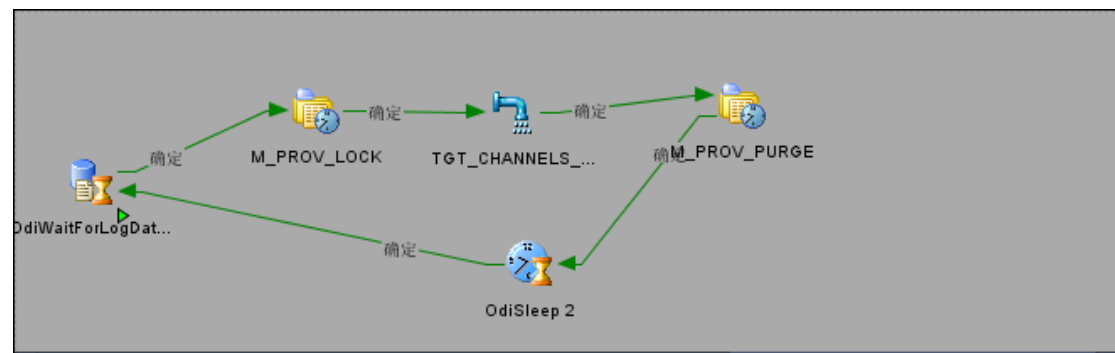
5.3 包

结合 2 触发器方式实时单项复制和 4Log Miner 方式非实时单项复制，可以得到 Log Miner 方式实时单项复制的包设计。

5.3.1 新建包如 TGT_CHANNELS_LOGMINER_RT

5.3.2 关系图

PACKAGE 的内容如下图：



● 配置 OdiWaitForLogData

OdiWaitForLogData 将轮询数据源，发现变化数据。如果发现了变化数据则会进入下一步。

要配置的属性主要包括：

逻辑架构：数据源表所对应的逻辑架构

轮询间隔：每隔多长时间轮询一次。例子中间隔是 5000 毫秒。

订户：CDC 的订户名称。

表名称：数据源表。这个表应该启动了 CDC 日志

“单元行计数”和“全局行计数”：可以定义必须要发生多少行变化才进行同步。在例子中，一旦有变化就同步。

- 配置 M_PROV_LOCK

M_PROV_LOCK 是将需要同步的模型如 oracle_sh 拖到包中来。用来扩展窗口和锁定订户。

下图是 M_PROV_LOCK 的例子。

主要配置了下面的信息：

类型：选择“日记记录模型”

使用：选上扩展窗口和锁定订户。

订户：添加要同步的订户。

- 将变化数据同步到目标表的接口如 TGT_CHANNELS_CDC_Int

- M_PROV_PURGE 是将需要同步的模型如 oracle_sh 拖到包中来。用来清除日记和取消订户锁定。下图是 M_PROV_PURGE 的例子。

主要配置了下面的信息：

类型：选择“日记记录模型”

使用：选上清除日记和取消订户锁定。

订户：添加要同步的订户。

- 配置 OdiSleep

OdiSleep 使得执行完接口之后可以等待一段时间再开始轮询。主要配置的参数有：

等待延迟：等待的毫秒数。例如 5000 毫秒，或者 180000 表示 30 分钟。