

Oracle Technet: [Oracle Streams](#)

Making Data Flow

By Sanjay Mishra

Traduzione di Andrea Salzano

Usa Oracle Streams per far sì che i dati si muovano da soli.

Oggi giorno una delle sfide degli ambienti di business distribuiti è la condivisione di informazioni tra un insieme eterogeneo di applicazioni e di database. Nel momento in cui il numero di applicazioni cresce, si finisce con l'utilizzare molti database, magari di vendors differenti. La condivisione dell'informazione in ambienti così complessi ed eterogenei può essere dispendiosa. Oracle Streams fornisce un'infrastruttura flessibile che può essere usata per semplificare la condivisione.

Cosa è Streams?

Oracle Streams cattura le modifiche del database, le mette in un'area per poterle applicare su più istanze, le propaga ad uno o più database di destinazione e le applica. Utilizzando Oracle Streams, sistemi enterprise possono catturare, propagare ed applicare le informazioni come segue:

- All'interno di un database Oracle
- Tra due database Oracle
- Tra più database Oracle
- Tra un database Oracle ed un database non-Oracle

Come Streams lavora

Oracle Streams inizia con il catturare i cambiamenti. Tali cambiamenti (dati, tabelle, etc) che hanno luogo in un database, sono registrati nei redo log files.

Il processo di "Streams capture" estrae questi cambiamenti dai redo log e li formatta in un "logical change record" (LCR). Gli LCR vengono storicizzati in una coda (staged). Successivamente, Streams propaga gli LCR da una coda (la "producer queue") all'altra (la "consumer queue") ed applica (o "consume") gli LCR dalla "consumer queue" al database di destinazione.

Oracle Streams possono essere utilizzati per eseguire compiti come:

- Data replication
- Estrazione e caricamento di Data warehouse
- Notifica degli eventi
- Message queuing
- Migrazione di piattaforma del database
- Upgrade di database ed applicazioni

Quest'articolo mostra come è possibile costruire uno "Streams data replication" da un database Oracle ad un altro (sempre Oracle).

Costruire Streams tra due database Oracle

L'esempio in questo articolo illustra la replicazione tra un database sorgente, TEST10G1, ed uno di destinazione, TEST10G2 (e' fatta ipotesi che i due database risiedano su due macchine separate, ma nulla vieta di usare gli stessi passi se i due db risiedono sulla stessa macchina). E' possibile seguire i seguenti passi per costruire un ambiente di "Streams replication":

1. Impostare il db in ARCHIVELOG mode,
2. Creare lo "Streams administrator",
3. Impostare i parametri di inizializzazione,
4. Creare un database link,
5. Costruire la coda sorgente e quella di destinazione,
6. Costruire il logging supplementare nel database sorgente,
7. Configurare il processi di "cattura" nel database sorgente,
8. Configurare il processo di propagazione,
9. Creare la tabella di destinazione,
10. Fornire i privilegi,
11. Istanziamento del system change number (SCN),
12. Configurazione dei processi di "apply" al database di destinazione,
13. Avvio dei processi di cattura e di apply

Le seguenti sezioni descrivono i passi sopra indicati ed includono codice d'esempio ove possibile.

1. Impostazione del db in ARCHIVELOG mode. Il processo di "Streams capture" dai redo log files del database. Per assicurare che l'informazione nei redo log files è disponibile per il "capture", occorre lanciare il database sorgente in ARCHIVELOG mode. In questo esempio, si fa ipotesi che il database sia così impostato. Far riferimento a "Oracle Database Administrator's Guide" per eventuali istruzioni su come impostare il db in ARCHIVELOG mode.
2. Creazione dello "Streams administrator". Tutto l'ambiente Streams (Streams environment) è gestito da un utente database-amministrativo. Per gestire lo Streams environment, lo Streams administrator necessita di alcuni specifici privilegi e deve creare alcune tabelle per storicizzare informazioni Streams-related. Per ogni database partecipante allo Streams data sharing environment, è necessario creare un utente e designare questo account come Streams administrator. Non utilizzare questo account per un qualsiasi altro scopo e non utilizzare gli utenti SYS o SYSTEM come Streams administrator. Lo Streams administrator crea alcune tabelle nel suo tablespace di default. Occorre specificare un tablespace diverso da SYSTEM per questo scopo. E' possibile utilizzare il tablespace USERS o un altro per storicizzare le tabelle proprietarie di Streams administrator, ma per facilità di gestione, è raccomandato utilizzare un tablespace separato per Streams. Utilizzare il seguente comando per creare il tablespace per lo Streams administrator:

```
CREATE TABLESPACE streams_tbs DATAFILE
'/u01/app/oracle/oradata/TEST10G1/streams_tbs.dbf'
SIZE 25M;
```

Ora occorre creare l'utente Streams administrator nel database, come segue:

```
CREATE USER strmadmin IDENTIFIED BY strmadmin
DEFAULT TABLESPACE streams_tbs
TEMPORARY TABLESPACE temp
QUOTA UNLIMITED ON streams_tbs;
```

Ora fornire i grant dei ruoli di CONNECT, RESOURCE e DBA allo Streams administrator:

```
GRANT CONNECT, RESOURCE, DBA TO strmadmin;
```

Utilizzare la procedura GRANT_ADMIN_PRIVILEGE nel package DBMS_STREAMS_AUTH per fornire i privilegi richiesti allo Streams administrator:

```
BEGIN
    DBMS_STREAMS_AUTH.GRANT_ADMIN_PRIVILEGE (
        grantee => 'strmadmin',
        grant_privileges => true);
END;
/
```

E' da notare che Streams non richiede l'uso dei seguenti ruoli ma fornirli può aiutare l'amministratore:

```
GRANT SELECT_CATALOG_ROLE TO strmadmin;
GRANT SELECT ANY DICTIONARY TO strmadmin;
```

Esegui i precedenti passi per costruire un amministratore per ognuno dei database partecipanti all'ambiente di Streams data sharing. In questi esempi, i passi necessitano di essere eseguiti su entrambi i database TESTG1 TESTG2.

3. Impostazione dei parametri di inizializzazione. Impostare i specifici parametri di inizializzazione nei database partecipanti allo Streams data sharing. Table 1 descrive tali parametri ed i valori a cui devono essere impostati.

Parameter	Source (TEST10G1)	Destination (TEST10G2)
GLOBAL_NAMES	TRUE	TRUE
COMPATIBLE	10.1.0	10.1.0
JOB_QUEUE_PROCESSES	2 (or higher)	N/A
STREAMS_POOL_SIZE	200MB (minimum recommended)	200MB (minimum recommended)

Tabella 1: Parametri di inizializzazione e loro valori

Il parametro GLOBAL_NAMES deve essere impostato a TRUE sia nel database sorgente che in quello di destinazione. Questo influenza la creazione del database link al Passo 4¹.

Affinché Streams lavori, il parametro COMPATIBLE deve essere settato a 9.2.0 o superiore in tutti i database. Per questo esempio, viene utilizzato il valore 10.1.0, poiché stiamo costruendo Streams replication in un database Oracle 10g.

Il parametro JOB_QUEUE_PROCESSES deve essere settato a 2 o ad un valore superiore nei database sorgenti. In questo esempio occorre impostarlo a 2 o ad un valore superiore, nel database sorgente (TEST10G1).

Streams utilizza system global area (SGA) memory per immagazzinare le informazioni richieste per il “capture” e l’“apply”. Per allorare memoria nella SGA utilizzata da Streams, imposta uno “Streams pool” specificando il parametro STREAMS_POOL_SIZE. Il valore di default per questo parametro è 0, il che significa che Streams usa memoria dallo shared pool.

Se si lascia STREAMS_POOL_SIZE al suo valore di default (0), Streams utilizzerà fino al 10% dello SHARED_POOL_SIZE, così Oracle raccomanda l’impostazione di STREAMS_POOL_SIZE almeno di 200MB. Sia il database sorgente che quello di destinazione devono tener conto di questa memoria aggiuntiva nella SGA.

4. Creazione di un database link. E’ necessaria la creazione di un database link dal database sorgente al database di destinazione. In questo esempio, verrà creato un database link da TEST10G1 a TEST10G2.

```
CONNECT strmadmin/strmadmin@TEST10G1
CREATE DATABASE LINK TEST10G2
CONNECT TO strmadmin
IDENTIFIED BY strmadmin
USING 'TEST10G2' ;
```

5. Costruzione di una coda sorgente ed una di destinazione. I dati si spostano dal database sorgente a quello di destinazione attraverso code. Utilizzare la procedura SET_UP_QUEUE nel package DBMS_STREAMS_ADM per costruire le code. Di

¹ GLOBAL_NAMES specifica se un database link debba avere lo stesso nome del database a cui si connette.

default questa procedura crea una “queue table” il cui nome è streams_queue_table ed una coda il cui nome è “streams_queue”. E’ possibile sovrascrivere questi nome specificando i parametri “queue_table” e “queue_name” della procedura SET_UP_QUEUE. I nomi di default vanno bene, a meno che si voglia creare code multiple e multiple “queue tables”. Per gli scopi di esempio di replicazione in questo articolo, si accetteranno i nomi di default, eseguendo la procedura come segue:

```
EXEC DBMS_STREAMS_ADM.SET_UP_QUEUE ( ) ;
```

Eseguire questa procedura su entrambi i database. Quando eseguita sul database sorgente (TEST10G1), viene creata la coda sorgente, e quando eseguita sul database di destinazione (TEST10G2), viene creata la coda di destinazione. Poiché abbiamo deciso di accettare i nomi di default, i nomi delle code sorgente e destinazione saranno gli stessi. Successivamente assoceremo queste due code ad un processo di propagazione.

6. Costruzione di logging supplementare nel database sorgente. Prima di iniziare a catturare i cambiamenti nel database sorgente, è necessario aggiungere un logging supplementare sulle tabelle che verranno modificate. Il logging supplementare mette informazioni aggiuntive nei redo logs utili durante il processo di “apply”. Per più informazioni su logging supplementare, far riferimento a Oracle Streams Replication Administrator's Guide.

Per aggiungere logging supplementare, connettersi al database sorgente come lo schema proprietario (SCOTT, in questo esempio) ed eseguire il seguente statement:

```
ALTER TABLE emp
ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY, UNIQUE)
COLUMNS ;
```

7. Configurazione del processo di cattura nel database sorgente. Il processo di “capture” inizia la replicazione catturando i cambiamenti nel database sorgente e formattando ognuno di essi in un LCR (Logical Change Record) ed accodandoli. Occorre creare un “capture process” (processo di cattura) per estrarre i cambiamenti dai redo logs. E’ possibile configurare capture process lanciandolo sul database sorgente – chiamato “local capture” – o in remoto su un altro database - chiamato “downstream capture”.

Mentre si crea un capture process, vengono aggiunte regole per specificare quali cambiamenti catturare e quali scartare. Le regole sono combinate in insiemi di regole. Il capture process può avere un insieme di regole positive ed uno di regole negative. Per il lavoro di un processo di “replication capture”, è necessario aggiungere un insieme di regole positive che specifichino che i cambiamenti DML e DDL siano catturati. Occorre fare questo utilizzando la procedura ADD_TABLE_RULES del package DBMS_STREAMS_ADM:

```

BEGIN  DBMS_STREAMS_ADM.ADD_TABLE_RULES(
        table_name      => 'scott.emp',
        streams_type    => 'capture',
        streams_name    => 'capture_stream',
        queue_name      => 'strmadmin.streams_queue',
        include_dml     => true,
        include_ddl     => true,
        inclusion_rule  => true);
END;
/

```

Questa chiamata alla procedura crea il processo locale di “Streams capture” chiamato capture_stream. Nota che è possibile utilizzare il parametro opzionale source_database per specificare un’altro database per uno “downstream capture”. Se questo parametro viene omesso (come in questo esempio) o impostato a NULL, la procedura creerà un processo di cattura locale.

Il parametro streams_type indica che questa procedura creerà un capture process. (Nota che la stessa procedura sarà utilizzata successivamente per creare un apply process). Il parametro inclusion_rule con il valore impostato a true indica che questa procedura creerà un insieme positivo di regole per il processo di cattura. Il valore true per il parametro include_dml indica che una regola sarà creata per i cambiamenti DML, ed il valore true per il parametro include_ddl indica che una regola sarà creata per i cambiamenti DDL. Il parametro table_name indicata che queste regole sono associate alla tabella EMP nello schema SCOTT. E’ possibile costruire regole per uno schema o per l’intero database, utilizzando le procedure ADD_SCHEMA_RULES e ADD_GLOBAL_RULES rispettivamente. Vedere il manuale PL/SQL Packages and Types Reference per maggiori informazioni su questi. La coda, streams_queue, specificata in questa procedura era stata creata dalla procedura SET_UP_QUEUE al Passo 5.

8. Configurare il processo di creazione. Una volta che i cambiamenti sono catturati ed accodati, occorre propagarli al database di destinazione. Per fare questo, si deve creare un processo di propagazione ed associare la coda sorgente con la coda di destinazione. Sono state create le code sorgente e destinazione (ognuna col nome di default “streams_queue”) nel database sorgente ed in quello di destinazione, rispettivamente, al Passo5 chiamando la procedura SET_UP_QUEUE.

La procedura nel Listato 1 (visibile alla in calce a questo documento N.d.T) The procedure in [Listing 1](#) crea un processo di propagazione ed aggiunge delle regole ad un insieme di regole positive per questo processo.

Questa procedura crea un processo di propagazione, chiamato TEST10G1_to_TEST10G2, per il quale la coda sorgente è streams_queue nel database TEST10G1 e la coda di destinazione è streams_queue nel database TEST10G2. Questa procedura aggiunge anche regole DML e DDL all’insieme di regole positive.

9. Creazione della tabella di destinazione. Occorre creare la tabella di destinazione. Prima di iniziare a replicare le modifiche nelle DDL e le DML da una tabella

sorgente a una di destinazione, la tabella deve esistere nel database di destinazione. Se la tabella non esiste ancora nel database di destinazione, si hanno diverse opzioni per creare l'oggetto, come usare il Data Pump, export/import, RMAN, transportable tablespaces, e così via. E' possibile consultare l'articolo [Moving Data Faster](#)² per ulteriori informazioni sulle opzioni disponibili.

10. Fornire i privilegi. Sul database di destinazione, lo Streams administrator applica i cambiamenti catturati dal database sorgente. Per essere capace di applicare i cambiamenti alle tabelle di destinazione, lo Streams administrator deve avere opportuni privilegi sugli oggetti in gioco. Per esempio, se i cambiamenti devono essere applicati alla tabella EMP dello schema SCOTT, è necessario eseguire quanto segue dopo essersi connessi al database di destinazione TEST10G2:

```
GRANT ALL ON scott.emp TO strmadmin;
```

Poiché in questo esempio è stato costruito lo Streams administrator con i ruoli da DBA (Passo 2), è possibile scegliere di saltare questo passo. E' tuttavia una buona idea esplicitamente fornire grant appropriati agli oggetti dello Streams administrator.

11. Istanza del System Change Number (SCN). E' necessario istanziare l'SCN per la tabella dal database sorgente che si vuol replicare. Questo assicura che i cambiamenti nella tabella sorgente, catturati prima di istanziare l'SCN non saranno applicati al database di destinazione. La procedura nel Listato 2, istanzia l'SCN per la tabella EMP nel database di destinazione, prendendo l'SCN corrente dal database sorgente. Eseguire la procedura nel Listato 2 connettendosi al database sorgente (TEST10G1) come Streams administrator.

Il Listato 2 mostra come la procedura GET_SYSTEM_CHANGE_NUMBER del package DBMS_FLASHBACK restituisce il corrente SCN del database sorgente. Questo SCN è utilizzato per istanziare l'SCN per la tabella EMP dello schema SCOTT nel database di destinazione. E' da notare che la procedura SET_TABLE_INSTANTIATION_SCN del package DBMS_APPLY_ADM è chiamata attraverso il database link TEST10G2, che implica che, sebbene questa procedura venga chiamata mentre c'è la connessione a TEST10G1, verrà eseguita nel DB TEST10G2.

Se la tabella non esisteva precedentemente nel database di destinazione e si è utilizzato export/import, Data Pump, o transportable tablespaces per copiarla, l'istanza dell'SCN sarà impostata automaticamente per la tabella.

12. Configurare l'"apply process" nel database di destinazione. Occorre creare un proceso di "apply" ed associargli la coda di destinazione. Inoltre, per tale processo, occorre aggiungere anche delle regole. Si realizza questo chiamando la procedura ADD_TABLE_RULES del package DBMS_STREAMS_ADM:

² Technet: <http://www.oracle.com/technology/oramag/oracle/04-sep/o54data.html>

```

BEGIN
    DBMS_STREAMS_ADM.ADD_TABLE_RULES(
        table_name      => 'scott.emp',
        streams_type    => 'apply',
        streams_name    => 'apply_stream',
        queue_name      => 'strmadmin.streams_queue',
        include_dml     => true,
        include_ddl     => true,
        source_database => 'TEST10G1',
        inclusion_rule  => true);
END;
/

```

Questa procedura crea un “apply process” (come indicato dal parametro streams_type) per la tabella EMP nello schema SCOTT. L’“apply process”, chiamato apply_stream, è associate con streams_queue. Il processo di apply aggiunge anche regole di DML e di DDL rules all’insieme di regole positive (come indicato dal parametro inclusion_rule).

E’ da notare che precedentemente è stato usato la procedura DBMS_STREAMS_ADM.ADD_TABLE_RULES per creare (ed aggiungere regole) un processo di cattura (Passo 7).

13. Start del processo di “capture” e di “apply”. Ora che sono stati configurati tutti gli oggetti ed i processi configurati, ciò che è necessario è far partire il processo di “capture” ed “apply”. Per far partire il processo di “capture”, connettersi al database sorgente ed eseguire la procedura START_CAPTURE del package DBMS_CAPTURE_ADM:

```

BEGIN
    DBMS_CAPTURE_ADM.START_CAPTURE(
        capture_name => 'capture_stream');
END;
/

```

Allo stesso modo, per far partire il processo di “apply”, occorre connettersi al database di destinazione ed eseguire la procedura START_APPLY del package DBMS_APPLY_ADM. Comunque, prima di fare questo, è raccomandato impostare il parametro disable_on_error del processo di apply ad n, cosicché tale processo continuerà anche se incontra errori.

```

BEGIN
    DBMS_APPLY_ADM.SET_PARAMETER(
        apply_name => 'apply_stream',
        parameter  => 'disable_on_error',
        value      => 'n');
END;
/

```

```

BEGIN
    DBMS_APPLY_ADM.START_APPLY(
        apply_name => 'apply_stream');
END;
/

```

Ora l'ambiente di Streams replication è pronto, e I cambiamenti DML e DDL nella tabella SCOTT.EMP nel database sorgente sarà replicato nella corrispondente in the source database will be replicated in the corresponding table in the destination database.

Conclusioni

Oracle Streams può catturare, propagare ed applicare i cambiamenti in un database, automaticamente, includendo sia modifiche DML che DDL. Applicazioni che richiedono replicazione, data warehouses, database migrations, and database upgrades possono beneficiare di Oracle Streams.

Listato 1: Creazione delle regole e del processo di propagazione

```

BEGIN
    DBMS_STREAMS_ADM.ADD_TABLE_PROPAGATION_RULES(
        table_name           => 'scott.emp',
        streams_name         => 'TEST10G1_to_TEST10G2',
        source_queue_name    => 'strmadmin.streams_queue',
        destination_queue_name => 'strmadmin.streams_queue@TEST10G2',
        include_dml          => true,
        include_ddl          => true,
        source_database       => 'TEST10G1',
        inclusion_rule        => true);
END;
/

```

Listato 2: Ottenere ed impostare l'istanza dell' SCN

```

DECLARE
    source_scn  NUMBER;
BEGIN
    source_scn := DBMS_FLASHBACK.GET_SYSTEM_CHANGE_NUMBER();
    DBMS_APPLY_ADM.SET_TABLE_INSTANTIATION_SCN@TEST10G2(
        source_object_name      => 'scott.emp',
        source_database_name    => 'TEST10G1',
        instantiation_scn       => source_scn);
END;
/

```

Sanjay Mishra (smishra_tech@yahoo.com) has coauthored four Oracle books, including the recent *Mastering Oracle SQL*, Second Edition, published by O'Reilly & Associates (2004).

Copyright © [2004], Oracle. All rights reserved.