

oracle 10g 研究ORACLE_HOME rdbms admin 下的脚本的功能 (2) a0900010.sql

```
oracle 10g 研究ORACLE_HOME rdbms admin 下的脚本的功能 (2) a0900010.sql
#upgrade from 9.0.1 to 9.2.0
#upgrade from 9.2.0 to the new release

oracle 10g 研究ORACLE_HOME rdbms admin 下的脚本的功能 (2) a0900010.sql
Rem
Rem $Header: a0900010.sql 07-apr-2003.10:51:23 nbhatt Exp $
Rem
Rem a0900010.sql
Rem
Rem Copyright (c) 1999, 2003, Oracle Corporation. All rights reserved.
Rem
Rem NAME
Rem a0900010.sql - additional ANONYMOUS BLOCK dictionary upgrade.
Rem Upgrade Oracle RDBMS from 9.0.1 to the new release
Rem
Rem
Rem DESCRIPTION
Rem Additional upgrade script to be run during the upgrade of an
Rem 9.0.1 database to the new release.
Rem
Rem This script is called from u0900010.sql and a0801070.sql
Rem
Rem Put any anonymous block related changes here.
Rem Any dictionary create, alter, updates and deletes
Rem that must be performed before catalog.sql and catproc.sql go
Rem in c0900010.sql
Rem
Rem The upgrade is performed in the following stages:
Rem STAGE 1: steps to upgrade from 9.0.1 to 9.2.0
Rem STAGE 2: upgrade from 9.2.0 to the new release
Rem
Rem NOTES
Rem * This script must be run using SQL*PLUS.
Rem * You must be connected AS SYSDBA to run this script.
Rem
Rem MODIFIED (MM/DD/YY)
Rem nbhatt 04/07/03 -
Rem nbhatt 03/07/03 -
Rem nbhatt 03/06/03 - fwm merge upgrade changes
Rem aramacha 12/05/02 - AQ - upgrade message for rule subs.
Rem nbhatt 09/25/02 - upgrade bugs
Rem rburns 02/13/02 - call 9.2.0 script
Rem nbhatt 02/08/02 - transformation upgrade changes
Rem ksurlake 02/22/02 - recreate subscriber view
Rem nbhatt 02/14/02 - add ruleset and ev context to old queue tables
Rem rburns 12/06/01 - cleaup comments.
Rem rburns 11/13/01 - rename registry package
Rem skaluska 11/02/01 - add rule engine upgrade script
Rem nbhatt 11/02/01 - subscriber downgrade
Rem najain 11/01/01 - fix bugs
Rem nbhatt 11/01/01 - subscriber enhancements
Rem eehrsam 10/11/01 - Merged eehrsam_lrg75925
Rem rburns 08/22/01 - populate component registry
Rem rburns 06/07/01 - Merged rburns_setup_901_upgrade
```

```

Rem      rburns      06/04/01 - created
Rem

Rem =====
Rem BEGIN STAGE 1: upgrade from 9.0.1 to 9.2.0
Rem =====

Rem Insert PL/SQL blocks here

Rem=====
Rem Upgrade the transformations metadata
Rem=====

Rem populate new columns for all the existing transformations
declare
    trans_cursor      INTEGER;
    rows_processed    INTEGER;
    prs_stmt          VARCHAR2(2000);
    CURSOR  get_txfmts IS
        SELECT transformation_id, owner, name, from_toid, to_toid
          FROM transformations$;
    trans_row          get_txfmts%ROWTYPE;
    src_schema         VARCHAR2(30);
    src_name           VARCHAR2(30);
    dest_schema        VARCHAR2(30);
    dest_name          VARCHAR2(30);
    fetch_type         VARCHAR2(300);

begin
    trans_cursor := dbms_sql.open_cursor;
    fetch_type := 'SELECT u.name, o.name FROM obj$ o, user$ u WHERE ' ||
        ' u.user# = o.owner# AND o.oid$ = :1';

    FOR table_row IN get_txfmts
    LOOP
        BEGIN
            EXECUTE IMMEDIATE fetch_type INTO src_schema, src_name
              USING table_row.from_toid;
        EXCEPTION
            WHEN no_data_found THEN
                src_schema := NULL;
                src_name := NULL;
            WHEN OTHERS THEN
                dbms_system.ksdwrt(1, 'exception when upgrading transformation : ' ||
                    table_row.owner || '.' || table_row.name);
        END;

        BEGIN
            EXECUTE IMMEDIATE fetch_type INTO dest_schema, dest_name
              USING table_row.from_toid;
        EXCEPTION
            WHEN no_data_found THEN
                dest_schema := NULL;
                dest_name := NULL;
            WHEN OTHERS THEN
                dbms_system.ksdwrt(1, 'exception when upgrading transformation : ' ||
                    table_row.owner || '.' || table_row.name);
        END;
    END LOOP;
END;

```

```

UPDATE transformations$ t
SET t.from_schema = src_schema, t.from_type = src_name,
    t.to_schema = dest_schema, t.to_type = dest_name
WHERE t.transformation_id = table_row.transformation_id;

END LOOP;

end;

/

Rem =====
Rem upgrade rules engine objects
Rem =====

begin
    dbms_rule_compatible_90.upgrade_rule_objects;

end;

/

DECLARE

    qt_schema    VARCHAR2(30);
    qt_name      VARCHAR2(30);
    qt_flags     NUMBER;

    CURSOR find_qt_c IS SELECT schema, name, flags, objno
                        FROM system.aq$queue_tables;

    subtab_sql   VARCHAR2(1024);
    add_col_sql  VARCHAR2(300);

    sel_queues   VARCHAR2(300);
    qcur         INTEGER;
    ignore       INTEGER;
    q_name       VARCHAR2(30);

BEGIN

    -- statement to select normal queues of each queue table
    sel_queues := 'SELECT name FROM system.aq$queues ' ||
                  ' WHERE table_objno = :a1 AND usage = 0 ';

    qcur := dbms_sql.open_cursor;
    dbms_sql.parse(qcur, sel_queues, dbms_sql.v7);

    FOR q_rec IN find_qt_c LOOP          -- iterate all queue tables

        qt_schema := q_rec.schema;      -- get queue table schema
        qt_name   := q_rec.name;        -- get queue table name
        qt_flags  := q_rec.flags;       -- get queue table flags

        IF ((bitand(qt_flags, 8) = 8) AND (bitand(qt_flags, 1) = 1)) THEN
            subtab_sql := 'update ' || qt_schema || '.AQ$_' || qt_name || '_S a' ||
                          ' set a.subscriber_type = 8 + 64 + 128 ' ||
                          ' where a.subscriber_type = 8';

            execute immediate subtab_sql;

            subtab_sql := 'update ' || qt_schema || '.AQ$_' || qt_name || '_S a' ||
                          ' set a.subscriber_type = 1 + 64 ' ||
                          ' where a.subscriber type = 1';
        END IF;
    END LOOP;

END;

/

```

```

execute immediate subtab_sql;

add_col_sql := 'ALTER TABLE '
               || qt_schema || '.' || 'AQ$_' || qt_name || '_S'
               || ' ADD (ruleset_name VARCHAR2(61))';

BEGIN
    EXECUTE IMMEDIATE add_col_sql;
EXCEPTION
    WHEN OTHERS THEN
        RAISE;
END;

-- create table evaluation context
dbms_prvtaqis.create_qtab_evctx(qt_schema, qt_name);

dbms_prvtaqis.upgrade_90_92(qt_schema, qt_name);

-- drop the old subscriber view
dbms_prvtaqis.drop_subscriber_view(qt_schema, qt_name, TRUE);
-- create the new subscriber view
dbms_prvtaqis.create_subscriber_view(qt_schema, qt_name);

-- drop the old rules view
dbms_prvtaqis.drop_rules_view(qt_schema, qt_name, TRUE);
-- create the new rules view
dbms_prvtaqis.create_rules_view(qt_schema, qt_name);

-- for each normal queue
dbms_sql.define_column(qcur, 1, q_name, 31);
dbms_sql.bind_variable(qcur, 'a1', q_rec.objno);
ignore := DBMS_SQL.EXECUTE(qcur);

-- for all normal queues in the 81 queue table create a rule set
LOOP
    IF DBMS_SQL.FETCH_ROWS(qcur) > 0 THEN
        dbms_sql.column_value(qcur, 1, q_name);
        dbms_prvtaqis.create_queue_rule_set(
            qt_schema||'.'||q_name||'_R', qt_schema, qt_name);
    ELSE
        EXIT;
    END IF;
END LOOP;

END IF;
END LOOP;

-- close the queue cursor
dbms_sql.close_cursor(qcur);

END;
/

Rem Upgrade unconsumed messages for rule subscribers
Rem using Sub Names, to '92 Single Message Format'.
BEGIN
    dbms_prvtaqis.upgrade_rulesub_msgs;
END;
```

/

```
Rem =====
Rem Populate component registry based on old database contents
Rem =====

execute dbms_registry_sys.populate;

Rem =====
Rem END STAGE 1: upgrade from 9.0.1 to 9.2.0
Rem =====

Rem =====
Rem BEGIN STAGE 2: upgrade from 9.2.0 to the new release
Rem =====
Rem

@@a0902000

Rem =====
Rem END STAGE 2: upgrade from 9.2.0 to the new release
Rem =====

Rem *****
Rem END a0900010.sql
Rem *****
```