

Statspackの取得について

- [Statspackの取得について](#)
 - [本資料について](#)
 - [Statspack利用上の注意点](#)
 - [Statspackレポートの一括出力](#)
 - [参照先](#)

本資料について

本資料は、Statspackを利用した性能分析に際しての取得手順をまとめたものです。

StatspackはOracleの性能分析をするためのレポートを作成するOracleデータベース標準のツールで、Oracle8iから登場しました。OracleデータベースのEnterprise Editionだけでなく、Standard Editionでも使用できるため、全ての環境で使用することができます。

Statspack利用上の注意点

- データベースを作成した直後のままではStatspackを使用することはできません。[事前にStatspackをインストールする](#)必要があります。
- Statspackのスナップショットは、デフォルトで1時間間隔でスナップショットが取得されています。
デフォルトでは1時間おきに実行されるようになっていきます。自動的なスナップショット取得の設定方法は以下になります。

```
SQL> @?/rdbms/admin/spauto
```

- SQL詳細やセグメントの情報を取得するために、`i_snap_level` は `Level 7` に設定します。
`i_snap_level` の設定方法は以下になります。

```
SQL> exec statspack.modify_statspack_parameter(i_snap_level => 7);
```

※ `Level 6` は `9.0.1` のバージョンから使用可能になり、`Level 7` は `9.2.0` のバージョンから使用可能になりました。パラメータを何も指定せずにデフォルトのまま実行すると `Level 5` になります。[お薦めはLevel 7です](#)。

Statspackレポートの一括出力

取得したStatspackのスナップショットからレポートを生成します。下記の `mkspreportscript.sql` スクリプトを `perfstat` ユーザで実行します。カレントのディレクトリにレポートが作成されます。

- 目的: 数日分のStatspackをレポートを一度に出力することを自動化するため。
- 使い方: `mkspreportscript.sql` に本日時点からさかのぼってレポートを出力したい日数を指定します。

```
SQL> @mkspreportscript <number of days> <=== 必ず日数を引数で指定する
```

- 実行例: 本日時点からさかのぼって過去5日間のレポートを一括出力する例

```
SQL> connect perfstat/*****
SQL> @mkspreportscript 5
SQL> @getspreport
```

実行後、`getspreport.sql` というファイルがカレントディレクトリに作成されるため、必要に応じてファイルを編集後、実行することで、Statspackレポートが出力される。

- 注意事項
 - 本日からn日前以降のレポートを全て出力する。(本日分も含まる)
 - レポート期間は1時間で連続するスナップショットIDを指定する。

`mkspreportscript.sql` の内容は以下になります。

```
-- Create a temporary SQL script to collect multiple Statspack reports
-- by specified number of days.
-- Usage : SQL> @mkspreportscript <number of days>
-- Sample: SQL> @mkspreportscript 5
set echo off
set feedback off
set verify off
set trimspool on
set serveroutput on
prompt Begin with getspreport.sql.
prompt =====
spool getspreport.sql replace

DECLARE
  num_day NUMBER := &1;
  has_record BOOLEAN := FALSE;
  cursor cur_instances is select distinct INSTANCE_NUMBER from STATS$SNAPSHOT;
BEGIN
  dbms_output.put_line('-- Temporary script created by mkspreportscript.sql');
  dbms_output.put_line('-- Used to create multiple Statspack reports between two
snapshots');
  dbms_output.put_line('-- Usage : SQL> @getspreport');
  dbms_output.put_line('-- ');

  for inst_id in cur_instances LOOP
    dbms_output.put_line('-- INSTANCE: ' || inst_id.INSTANCE_NUMBER);
    for rec in (
      select DBID,
             INSTANCE_NUMBER,
             to_char(round(SNAP_TIME, 'mi'), 'yyyymmdd') BEGIN_SNAP_TIME,
             SNAP_ID START_SNAP_ID,
             lead(SNAP_ID) over(order by SNAP_ID) END_SNAP_ID
      from STATS$SNAPSHOT
      where INSTANCE_NUMBER = inst_id.INSTANCE_NUMBER and
             SNAP_TIME > trunc(SYSDATE-num_day) and
```

```

        SNAP_TIME <= SYSDATE
    order by START_SNAP_ID
) LOOP
    continue when rec.END_SNAP_ID is null;
    has_record := TRUE;
    dbms_output.put_line('define begin_snap=' || rec.START_SNAP_ID);
    dbms_output.put_line('define end_snap=' || rec.END_SNAP_ID);
    dbms_output.put_line('define report_name=sp_' ||
        rec.BEGIN_SNAP_TIME || '_' ||
        rec.INSTANCE_NUMBER || '_' ||
        rec.START_SNAP_ID || '-' ||
        rec.END_SNAP_ID || '.lst');
    dbms_output.put_line('@?/rdbms/admin/spreport.sql');
END LOOP;
END LOOP;
IF (NOT has_record) THEN
    dbms_output.put_line('prompt NO RECORD. ABORT!');
ELSE
    dbms_output.put_line('undefine begin_snap;');
    dbms_output.put_line('undefine end_snap;');
    dbms_output.put_line('undefine report_name;');
    dbms_output.put_line('prompt DONE!');
END IF;
END;
/

spool off
-- @getspreport.sql
prompt =====
prompt Done with getspreport.sql.
prompt Next, continue to launch:
prompt
prompt @getspreport
prompt
prompt to generate Statspack text reports in the current folder.
prompt =====

set feedback on
set verify on
set echo on

```

参照先

- [StatspackとDiagnostics Packの概要と使用方法](#)
- [STATSPACKを使った自動情報収集](#)
- [Using Statspack](#)
- [STATSPACK の取得とレポートの一括出力](#)