SPR (Smart Plant Review) Utility Work Flow

This document will give you the detailed overview of SPR utility which is used to sync the SPR MDB2 file with other application after the data exchange is completed.

Document Written by: Gagan Dhamija
SPR Utility Developer: Gagan Dhamija
Contact: gdhamija@bechtel.com
gagan.bechtel@gmail.com

SPR Utility Algorithm

1. Copy the label, label_names, label_values tables from Mdb2 in to SQL. (Time taken : 25 secs)

2. For each Commodity (spool, valve, weld etc.):
   a. Insert new labels in the label table (in both sql table and mdb table) if it exists then take the label id. (Time taken : few secs)
   b. Insert new labels with null value for all the linkages associated with the specified commodity in the temp table if new labels already exist for those linkages, copy the records with their values in the temp table so that we can update it in next step.
   c. This step 2(b) is done by the stored procedure named as UPDATE_LabelNames (Time taken : 30 secs)

   Procedure Explanation:

   i) Create the temp table as temp_GUID (guid was needed to create multiple temp tables in case multiple user execute the SPR utility)
   ii) Get the null value index from the label_value table if not found Insert the record with null value in the label_value table and get the null value index.
   iii) Parent Cursor (Loop through all the linkages of the specified commodity since the filter file is there to decide which commodity is which, this parent cursor is dynamically generated from C# code and passed it to this sp.)
   iv) Child Cursor (on list of new labels (i.e. const_status or WP))
   v) Insert null value on all the linkages for new labels if new label already exits get the existing record from the label table and copy in the temp table.

   d. Update all the records in the temp table based on proxy Views (Spool_view, weld_view etc.) This is done by stored procedure named as UPDATE_LabelValues

   For each new label (WP and CS), call UPDATE_LabelValues

   Procedure Explanation:

   i) Get the null value index and save in the variable.
   ii) For each row in proxy table, get tag and label value
      a) If the label value is null take null value index go to step c else step b.
      b) From label value get the value index if it exists in the label_value table else inserts the value in label_value table and get the value index.
      c) Update all the linkages of the specified tag with the new label value in the temp table (join query to get all linkages).

      (Time taken:

Commodity	Record count	Time
Hanger	6819	11 mins
Spool	8864	13 mins
Value	3820	6 mins
Weld	13193	22 mins)

3 Once the temp table is updated, delete the old records from label table and insert these new records from temp table. (Time taken : few secs)

4 Bulk copy labels and label_value tables from sql to mdb file. delete the old table first. (Time Taken: 30 secs)

5 Drop the cache tables from sql.

NOTE:

1. The time mentioned above is calculated by executing 25756-11.mdb2 file on Sherman db.
2. All the queries in the stored procedures are dynamic as we are producing dynamic cache tables based on GUID to support multiple utility run. And for this reason SQL is not able to cache the compiled query and it takes time to compile the query every time.
3. Most of the time was spent in step 2 (d) as update query takes time than insert or delete.
4. Earlier we were inserting the new labels with null values in label table directly and then updating those records as label table was very big so we introduced the temp table which significantly improved the processing time.