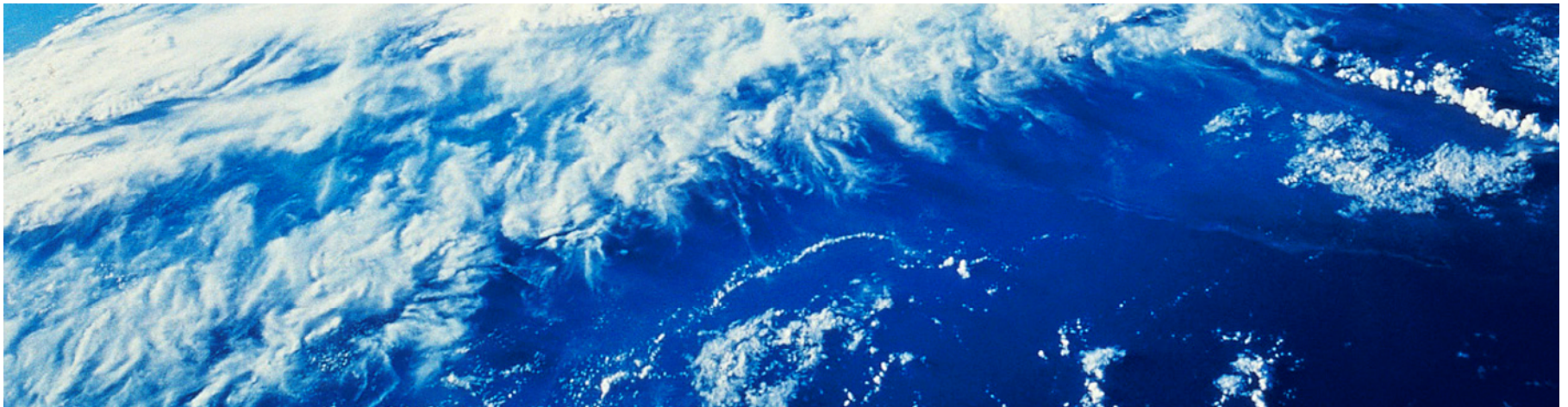


IBM WebSphere Commerce V7 FEP8

Advanced Filter for StagingProp



Melissa Singh

- Introduction
- StagingProp by Scope and Filter
- Solution: Advanced Filtering
- References

Currently:

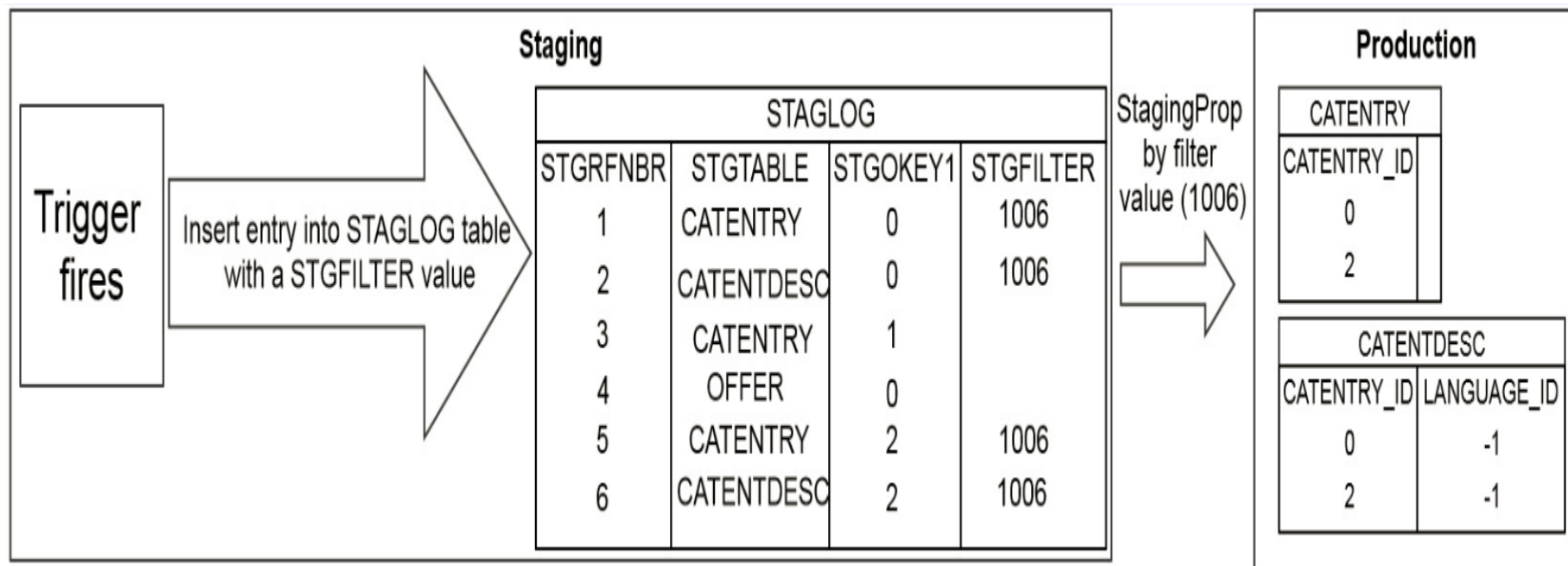
- Propagate all changes to production since the previous StagingProp was performed
- Typically performed during off-peak time so that impact to the site is not intrusive

What about customers who have a global instance and don't necessarily have an off-peak time?

As an administrator, I want to be able to perform a stagingprop to publish only a subset of content from staging to production as required by my business teams needs such as per store.

- Store based / locale based stagingprop:with global customers that have independent content for each country/store/geography.
- Schedule based changes - customers being able to send content to production that is scheduled to go live rather than all content.
- Component based changes - customers want to send specific data - promotions, catalog, marketing at different intervals.
- Combination of the above use cases.

- StagingProp by **scope**: Propagates all changes for particular tables
- StagingProp by **filter**: Propagates specific changes in a table by populating the STAGLOG.STGFILTER column for each change.
 - Use custom triggers to populate the column



StagingProp by scope:

- All changes for a table are propagated instead of just a subset of the changed records

StagingProp by filter:

- The use of triggers for all operations (insert, update and delete) can be problematic
- Need to update/maintain a lot of triggers
- Custom triggers can be overwritten by FP/FEP updates
- Customers require extensive knowledge of the table structure.

Create a staging filter configuration file:

- Customers can supply SQL conditions to reduce the set of records processed per table
- It's actually filtering the amount of records processed from the STAGLOG table
- Each SQL condition is based on the operation (Insert, Update, delete)
- Customers can supply another SQL condition to mark specific STAGLOG records as processed
- Additionally, the SQL condition can be used in conjunction with the existing StagingProp by filter option

Query 1: Filters the STAGLOG records to be propagated

STGRFNBR	STGTABLE	STGKEYNAME1	STGOKEY1	STGPROCESSED
1	offer	offer_id	0	0
2	offer	offer_id	1	0
3	offerprice	offer_id	2	0
4	offerprice	offer_id	3	0

Query 2: Filters the STAGLOG records to be updated as processed

STGRFNBR	STGTABLE	STGKEYNAME1	STGOKEY1	STGPROCESSED
1	offer	offer_id	0	1
2	offer	offer_id	1	0
3	offerprice	offer_id	2	1
4	offerprice	offer_id	3	0

Step 1: Configuration of filter XML file

src/com/ibm/commerce/staging/xml/stagingFilter.xml

Stage by store for the ATTR table

```
<wc:FilterDefinition tableName="ATTR className="">
  <wc:StagLogFilter>
    <wc:Filter action="INSERT">
      <![CDATA[T.storeent_id = {customfilter1}]]>
    </wc:Filter>
    <wc:Filter action="UPDATE">
      <![CDATA[T.storeent_id = {customfilter1}]]>
    </wc:Filter>
  </wc:StagLogFilter>
</wc:FilterDefinition>
```

Stage by store for the ATTRDESC table

```
<wc:FilterDefinition tableName="ATTRDESC className="">
  <wc:StagLogFilter>
    <wc:Filter action="INSERT">
      <![CDATA[SL.STGRFNBR in ( select B.STGRFNBR from
        attr A, staglog B where B.stgtable='attrdesc' AND
        B.stgokey1=A.attr_id AND B.stgprocessed=0 AND
        A.storeent_id = {customfilter1} )]]>
    </wc:Filter>
    <wc:Filter action="UPDATE">
      <![CDATA[SL.STGRFNBR in ( select B.STGRFNBR from
        attr A, staglog B where B.stgtable='attrdesc' AND
        B.stgokey1=A.attr_id AND B.stgprocessed=0 AND
        A.storeent_id = {customfilter1} )]]>
    </wc:Filter>
  </wc:StagLogFilter>
</wc:FilterDefinition>
```

```
stagingprop -dbtype DB2 -scope _all_ -sourcedb mall  
-sourcedb_user db2admin -sourcedb_passwd xxxxxxxx -destdb  
mallprod -destdb_user wcsadmin -destdb_passwd xxxxxxxx  
-customfilter1 10051 -filterconfigfile  
src/com/ibm/commerce/staging/xml/stagingFilter.xml
```

- The customfilter1 parameter can be the store_id and will be substituted into each xml configuration that declares {customfilter1}

For filtering of **deleted data**, the data to query may no longer exist.

For example:

Let's say we want to propagate deleted catalog entries for a given store. We must query the store id from another table – STORECENT – which holds the catalog-store relationship.

If data has already been deleted (or is missing) from the STORECENT table then we can't query it for the appropriate store id.

Solution: Use an updated staging trigger for delete operations to update the STAGLOG.FILTER value whenever a catalog entry is deleted from the CATENTRY table. The corresponding store id value can be inserted into the STAGLOG table at the time of deletion.

- Using the StagingProp log for troubleshooting will be the same as FEP 7
- **Primary key collisions:** If used correctly, advanced filtering will not allow for primary key collisions.
However, there is potential for **misuse**:
 - Let's say a custom query is used to select the staglog records to be marked as processed. If the query isn't correct, then not all of the targeted STAGLOG entries may be marked as 'processed'. Thus, on the next iteration of stagingProp, some data that was already processed may get propagated.
- With a detailed trace level, can use the log to identify the queries used to filter the STAGLOG entries to be propagated as well as the STAGLOG entries to be marked as processed

- Filtering data for the stagingprop utility to propagate
http://www-01.ibm.com/support/knowledgecenter/SSZLC2_7.0.0/com.ibm.commerce.admin.doc/tasks/tssfiltering.htm?lang=en
- Creating a staging filter configuration XML file
http://www-01.ibm.com/support/knowledgecenter/SSZLC2_7.0.0/com.ibm.commerce.admin.doc/tasks/tsscreatefilterconfigxml.htm?lang=en
- Example: Propagating filtered data to the production database
http://www-01.ibm.com/support/knowledgecenter/SSZLC2_7.0.0/com.ibm.commerce.admin.doc/refs/rsspropfilterdataex.htm?lang=en

Thank You!

Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, Coremetrics, DB2, PowerVM, Rational, WebSphere, and z/VM are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "

Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>

Other company, product, or service names may be trademarks or service marks of others.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2014. All rights reserved.