Process Portal Search Index Comprehensive

SETH MERRISS

Introduction

The search index in IBM Business Process Manager Process Portal is used to allow users to search for tasks and process instances and is used to generate various charts and data throughout Process Portal including in the Team Performance and Process Performance dashboards. Unfortunately, the search index can cause problems for BPM end users when the search index data falls out of sync with the operational BPM data.

The purpose of this article is to help BPM Administrators have a deeper understanding of the Process Portal Search Index, of how the indexing process works, what are the configurations available, how to understand that issues exists, and how to resolve common issues related to the search index. Until now the search indexing process has largely been a “black box” operating behind the scenes in BPM Process Portal. BPM Administrators are largely unaware that the search index may be experiencing issues (such as being out of sync) until BPM users report discrepancies in the data they observe in Process Portal.

Understanding the search indexing process will help BPM Admins to troubleshoot the indexing process and determine the root cause by being able to identify the “point of failure” in the process. This document will also provide details for best practices for configuring, maintaining, cleaning up, and resolving common issues that may arise with the Process Portal search index.

How it Works

A Brief Description

The Process Portal search index is a file-based Lucene index. By default, the search index is enabled in Process Portal and by default only human tasks are indexed. System tasks can be indexed by updating a setting. Each node in a BPM topology maintains its own search index. In some cases, it may be appropriate to configure a shared search index. There are three search index tracking database tables which are involved in the process of search indexing. They are the following:

- BPM_TASK_INDEX – keeps records of BPM tasks, reflection of the operational LSW_TASK data
- BPM_INSTANCE_INDEX – keeps records of BPM process instances, reflection of the operational LSW_BPD_INSTANCE data
- BPM_TASK_INDEX_JOB – keeps records of each indexing “job” for each node

There are four main phases of the search indexing process. Each phase is described in detail below. Please note that the following details document the indexing process after APARs JR55539 and JR55895 are applied.

Phase I – Operational Data Changes Trigger Runtime Events

Operational data is stored in the LSW_TASK and LSW_BPD_INSTANCE tables. When operational data is created, updated, or deleted the runtime engine fires events. The events correspond to starting processes, updating process instance data, completing a process instance, tasks starting, tasks being assigned or reassigned, tasks completing, etc.

Phase II – Search Event Point Listener Subscribed to the Events Triggers Updates to Index Tracking Tables

When the listener picks up that a task or process instance is created a new row is added to the BPM_TASK_INDEX or BPM_INSTANCE_INDEX table. At this time a field in the table named "MAJOR_EVENT_DATETIME" is set to a time stamp, which is one day before epoch time of "1969-12-31T00:00:00", creating a virtual queue. Similarly, when a process instance or a task are updated the "MAJOR_EVENT_DATETIME" field is updated to this same timestamp. For each of these operations a single row is inserted or updated at a time.

When process instance data and tasks are deleted from BPM using the wsadmin commands the SQL query updates up to 50,000 rows at once in the corresponding tracking tables. This SQL query operation updates the DELETED_DATETIME field from NULL to the current database timestamp.

Phase III – De-queueing the Task and Process Instances to be updated

Another thread called *ProcessIndexQueueDaemon* runs continuously in parallel which updates the MAJOR_EVENT_DATETIME field from the epoch time to the current database time. This occurs in batches of 20 rows at a time. If this SQL transaction takes longer than 1 second to complete then a warning is logged in the server systemout.log file:

```
CWLLG3231W: Task Index entries {LIST OF TASK IDS} took longer than 1 second to commit, retrying.
```

```
CWLLG3230W: Instance Index entries {LIST OF INSTANCE IDS} took longer than 1 second to commit, retrying.
```

This process acts as a dequeuer for the tasks or process instances allowing the indexing job to consume the updated records in order to update the corresponding documents in the search index.

Phase III – Updating the Lucene Search Index

At this point the index job thread, *ProcessIndexUpdaterDaemon*, finds records from the index tracking tables that have a MAJOR_EVENT_DATETIME that falls within the indexing window (last index job timestamp to the current indexing job timestamp) and updates the Lucene search index documents accordingly.

The purge process for deleting documents that have been deleted from the operational tables also occurs in this thread. This thread checks for records in the tracking tables that have a DELETED_DATETIME value that is newer than the last index purge time.

Each indexing job generates new records in the BPM_TASK_INDEX_JOB database table. This thread also deletes records of indexing jobs that are older than 7 days. The records are checked hourly to verify if any job records should be deleted.

Additionally, the *ProcessIndexUpdatedDaemon* thread is also responsible for recreating the Lucene search index during a manual rebuild or when running the *processIndexFullReIndex* command. This is completed by directly parsing the operational LSW_TASK and LSW_BPD_INTANCE tables which creates the corresponding documents in the Lucene index as well as creating or updating rows in the search index tracking tables BPM_TASK_INDEX and BPM_INSTANCE_INDEX.

Configuring the Process Portal Search Index

There are a number of configurable properties and configuration changes that can be made to the search index.

Configurable Properties

The following configurable properties can be updated using the 100Custom file and then restarting the server. To verify that the property changes persisted you can check the TeamworksConfiguration.running.xml file to confirm the new value.

When making changes to any of the following properties use the following template and update the value accordingly:

```
<search-index merge="mergeChildren">
    <task-index-enabled merge=”replace”>true</task-index-enabled>
    <task-index-update-interval merge=”replace”>5</task-index-update-interval>
    <task-index-update-completed-tasks merge=”replace”>false</task-index-update-completed-tasks>
    <task-index-store-fields merge=”replace”>false</task-index-store-fields>
    <task-index-work-manager merge=”replace”>wm/default</task-index-work-manager>
    <task-index-include-system-tasks merge=”replace”>false</task-index-include-system-tasks>
    <process-index-instance-completion-best-effort merge=”replace”>false</process-index-instance-
completion-best-effort>
</search-index>
```

<search-index-enabled>

By default, the value is *true*. Updating the value to *false* will disable the search index on the node in which the change was made. BPM end users will observe that the search filter field is no longer available in Process Portal. Users will observe the change in the Work, Processes, Team Performance, and Process Performance dashboard. Users will observe that Quick Stats will also be unavailable.

Please note that if you have multiple nodes and disable indexing on only some nodes but allow users to continue to work with process instances and tasks on these nodes then the search index will fall out of sync for all nodes. The nodes with indexing disabled will not register the runtime events and thus not trigger the updates to the index tracking tables.

<task-index-update-interval>

The default value for this property is 5 seconds. This property controls the amount of time between index updates. This specifies the window that the indexer uses to look for tasks and process instances to update. The indexer uses the latest INDEX_END_TIME from the BPM_TASK_INDEX_JOB table for the node in which the indexer is occurring then adds the value of this property to look for all task and process instance records whose MAJOR_EVENT_DATETIME falls in that timeframe.

Increasing this property will cause the indexer thread to have a longer period of times between processing records waiting to be indexed. It will decrease the load of the indexer. Users may also observe that tasks and process instances require a longer period of time before being available for searches and seeing Quick Stats updated in the dashboards.

<task-index-update-completed-tasks>

The default value is *false*. The indexer only updates the Lucene index with tasks that are open or just completed. When this property is set to *true* completed tasks will be updated in the Lucene search index when instance level data such as business variable values are updated. Searches will be based on the most recent values of the instance level data. Setting this property to *true* will increase the load of the indexer as it will require processing more data.

<task-index-store-fields>

The default value is *false*. This property determines if the value of the fields that are stored are stored as separate fields.

<task-index-work-manager>

The default value is the string *wm/default*. This property sets the work manager that is used by the indexing process to update the search index. You improve the performance of the search index creation by changing the work manager to a new dedicated work manager that has a higher number of available threads. You can create the new work manager in the WAS Admin Console.

<task-index-include-system-tasks>

By default, this value is *false*. This property can be updated if business requirements determine that system tasks should be available for search queries and for inclusion in the Gantt Chart generated in Process Performance dashboard. Generally, system tasks do not need to be indexed as they are handled entirely without human interaction. Indexing system tasks also greatly increases the load on the system.

<process-index-instance-completion-best-effort>

The default setting is *false*. This property controls whether completion dates are created for instances migrated from previous versions of BPM. If the value is *true*, the latest completion date of the associated tasks is used for the instance completion date. If there are not any associated tasks then the last modified time stamp of the instance is used.

<process-index-queue-update-size>

The default value for this property is 20. This property is added with APAR JR55895 to introduce batch sets to the indexing process with updating the Lucene search index. The property limits the number of records per batch to ensure that the commit transaction to update the search index completes in less than 1 second.

Using a Shared Search Index

Each node maintains its own search index. In some environment topologies with multiple nodes it becomes difficult for each search indexes to stay up-to-date when all search indexes are trying to access the index tracking database tables. Competing for resources can be a challenge when there are more than 4 nodes configured. It can be beneficial to set up a shared search index between the nodes.

The index can be maintained in only one cluster member at a time. It is enforced using a locking strategy on the index.

To set up a shared search index you can use a shared network storage solution for your index. Using WAS Admin Console navigate to the environment variables via Environment > WebSphere variables. Update the value of the cell scoped `BPM_SEARCH_TASK_INDEX_ROOT` variable to point to the common location for each cell.

Maintaining the Process Portal Search Index

Stay Current with the Latest Fixes

Be aware of the latest BPM Maintenance Strategy. It is important to stay up to date with the latest version of BPM and cumulative fixes to ensure you have the latest fixes applied to your environments.

Beyond that you will want to be aware for the following search index related APARs. Currently the most important search index related APAR is APAR JR55539. The BPM developers consider this a major fix. This fix supersedes almost all search index related APARs that were delivered before it. The fix also made major improvements to the BPM search index code to reduce the likelihood of out of sync conditions and to prevent high CPU and Out Of Memory Errors.

Here is a list of the current fixes available to improve the Process Portal Search Index:

APAR	DESCRIPTION	VERSION AVAILABLILITY
JR55539	THE IBM PROCESS PORTAL INDEX CAN FALL OUT OF SYNC WITH BPM TASK AND INSTANCE DATA LEADING TO WSVR0605W OR OUTOFMEMORYERROR	8.5.0.1, 8.5.0.2, 8.5.5, 8.5.6 CF2, 8.5.7 CF2016.06
JR55895	INTERIM FIX JR55539 LIMITS IBM PROCESS PORTAL INDEXING THROUGHPUT	8.5.0.1, 8.5.0.2, 8.5.7 CF2016.06
JR56801	TASK INDEXER FAILS WITH A NULLPOINTEREXCEPTION WHEN IT CAN'T FIND TASK PARTICIPANT INFORMATION IN THE IBM BPM DATABASE	8.5.0.1, 8.5.6 CF2, 8.5.7 CF2016.09
JR57183	THE MESSAGE STATING THAT THE FULL REINDEX COMPLETED	8.5.7 CF2017.03

	SUCCESSFULLY IS CONFUSING	
JR57261	WHEN REBUILDING A LUCENE INDEX THE TASKS IMPLEMENTED BY AN EXTERNAL ACTIVITY ARE NOT INDEXED	8.5.6 CF2, 8.5.7 CF2017.03
JR57297	TASK INDEXER FAILS WITH A NULLPOINTEREXCEPTION WHEN CREATING AN INSTANCE DOCUMENT	8.5.6 CF2, 8.5.7 CF2017.03
JR57607	OVERLAPPINGFILELOCKEXCEPTION MIGHT CAUSE THE LUCENEINDEXER TO STOP INDEXING DATA ERROR	8.5.0.2, 8.5.5, 8.5.7 CF2017.06
JR57349	WHEN THE LUCENE INDEXER RUNS AGAINST AN IBM DB2 DATABASE IT CREATES TOO MANY LOCKS ON THE BPM_TASK_INDEX TABLE	8.5.7 CF2017.03
JR57388	WHILE REINDEXING THE PROCESS PORTAL SEARCH INDEX, THE RETRIEVED LIST OF INSTANCES AND TASKS TO INDEX MIGHT BE INCOMPLETE	8.5.7 CF2017.03

Index Diagnostic Utility Tool V2.0

The Index Diagnostic tool was created by a team of BPM Developers. It is packaged as a .war file and can be deployed in the AppCluster as a WAS application. The tooling is quick and easy to install and access. It does not require stopping or restarting the server and takes only minutes to install. The tool comes with a .pdf file that provides in-depth detail for installation instructions and instructions for generating reports and interpreting the results. The tool also provides the ability to re-sync the search index tracking tables and the Lucene search index directly in the tool. More in-depth information is provided about the Diagnostic tool in next section.

Clean up the Search Index Tracking Database Tables

One issue with the search indexing process can occur over a long period of time. When operational BPM data is deleted using the appropriate wsadmin commands it is deleted from the operational database tables and the Process Portal search index but the corresponding records are not automatically deleted from the index tracking tables, BPM_TASK_INDEX and BPM_INSTANCE_INDEX. Instead only the DELETED_DATETIME field gets updated in these tables. Over time as records build up in these tables issues can occur such as database lock timeouts, high CPU usage, etc.

Verify the number of records in your tracking tables that represent deleted tasks and process instances by using the following queries, substituting <schema_name> where indicated.

```
SELECT COUNT(1) FROM <schema_name>.BPM_TASK_INDEX WHERE DELETE_DATETIME IS NOT NULL;
```

```
SELECT COUNT(1) FROM <schema_name>.BPM_INSTANCE_INDEX WHERE DELETE_DATETIME IS NOT NULL;
```

If you find that this number is large then there are two options you can pursue to clean up the tables. One is to open a PMR with IBM BPM Support and request a customer stored procedure that can be used to run regularly on your environment to clean up the tables. If you pursue this option please also provide the following:

- BPM version info report. Run the **`versionInfo -fixpacks -ifixes`** command from the bin directory of your IBM Business Process Manager install and redirect the output to a file.
- Database type and version
- Results of the COUNT SQL queries listed above

The second option requires stopping the environment, truncating the three index tracking tables (BPM_TASK_INDEX_JOB is the third), removing or deleting the search index directories, and then restarting the environment – bringing up one node at a time, waiting for the node to have rebuilt its search index before starting another one (this is not relevant if you have a shared search index configured). This will rebuild the tracking database tables based on the operational data in LSW_TASK and LSW_BPD_INSTANCE tables. You will no longer have any records where DELETED_DATETIME is not NULL. Detailed steps to rebuild the index tracking tables are provided in the section "*How to Rebuild the Search Index – Manually Rebuilding the Search Index Tracking Tables*".

In future releases of BPM a wsadmin command will be provided to clean up the index tracking tables so that the stored procedure or manual cleanup is no longer necessary.

Maintain and Tune the Database

In some cases it is important to tune the properties related to locking and perform tuning and upkeep of the DB environment.

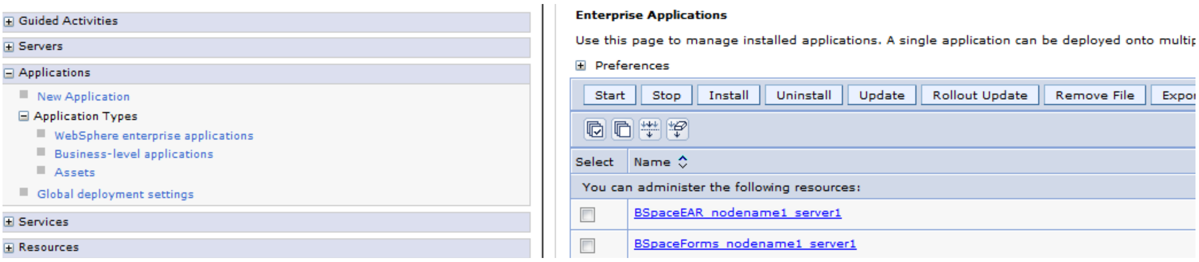
Search Index Diagnostic Utility Tool

This tool is useful for generating reports on the health of the index tracking tables and the search index. It can also be used to resync the tracking tables and the search index if out of sync conditions are found. For more in-depth information consult the IndexDiagnostic.pdf file provided for the tool which contains detailed installation instructions and usage documentation.

Installing the Tool

The following provides brief instructions to install the Diagnostic tool. The tool takes just a few minutes to install and does not require stopping nor restarting the BPM environment.

1. Open WebSphere Administrative Console and navigate to the Enterprise Application page



Enterprise Applications

Use this page to manage installed applications. A single application can be deployed onto multip

Select	Name
You can administer the following resources:	
□	BSpaceEAR_nodename1_server1
□	BSpaceForms_nodename1_server1

2. Click Install and select the index-diagnostic.ear file
3. Click on Next
4. Select Fast Path and click Next
5. Click on Step 4 then click Finish
6. Click on Save when the application has finished installing
7. Click on the checkbox next to the index-diagnostic application and click Start

Accessing the Tool

The tool is deployed to the default secure port of the application server. The base URL is /IndexDiagnostic. You should be able to access the tool using the following URL where 9443 is the default secure port.

https://<*bpm_host_name*>:9443/IndexDiagnostic

Analyzing the Reports

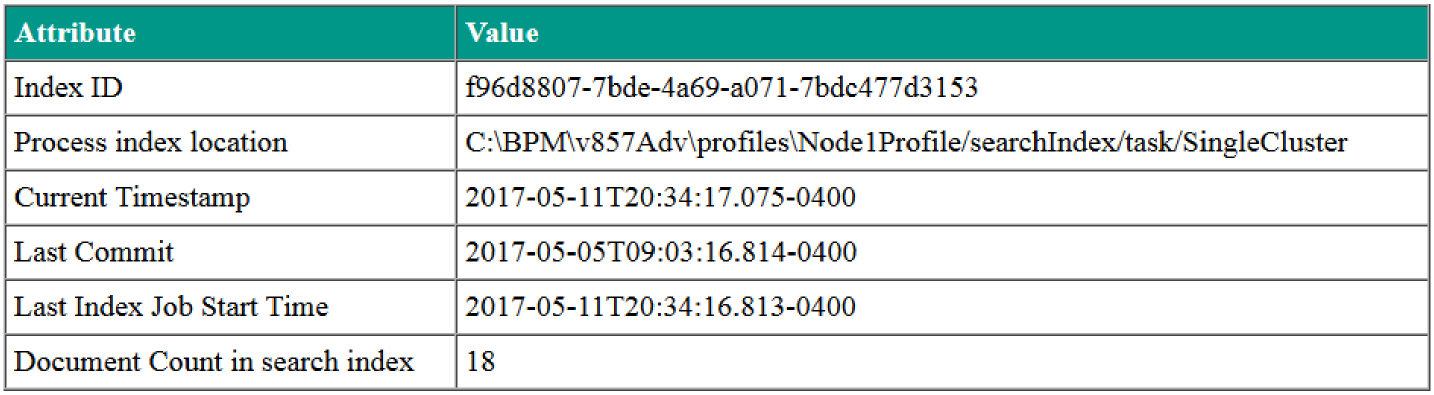
The following sections discuss each report that the Diagnostic tool generates and how to determine the condition of the search index and if any action is required to improve the health of the search index.

Index Summary Report

This report provides an overall view of the search index. Each section of this report is discussed in detail below.

Search Index Summary

This section provides a quick overview of the condition of the search indexing jobs.

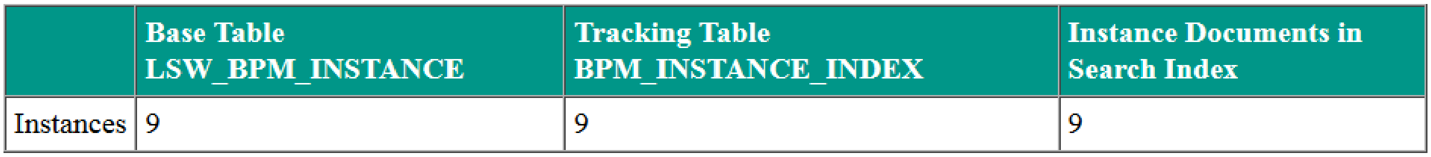


Attribute	Value
Index ID	f96d8807-7bde-4a69-a071-7bdc477d3153
Process index location	C:\BPM\v857Adv\profiles\Node1Profile/searchIndex/task/SingleCluster
Current Timestamp	2017-05-11T20:34:17.075-0400
Last Commit	2017-05-05T09:03:16.814-0400
Last Index Job Start Time	2017-05-11T20:34:16.813-0400
Document Count in search index	18

1. Index ID – each node maintains its own search index and this attribute provides the search index ID
2. Process index location – provides the directory of the Lucene search index
3. Current Timestamp – the current time of the database
4. Last Commit – the timestamp of the latest indexing job where tasks or process instances were committed to the index
5. Last Index Job Start Time – the timestamp of when the latest indexing job started (INDEX_START_TIME from BPM_TASK_INDEX_JOB table)
6. Document Count in search index – the total number of documents in the search index including tasks and process instances

Instance Summary Comparison

This table provides a comparison of the process instance data between the operational database table (LSW_BPD_INSTANCE), the index tracking table (BPM_INSTANCE_INDEX), and the Lucene search index. All three numbers should match in a perfectly synced system.



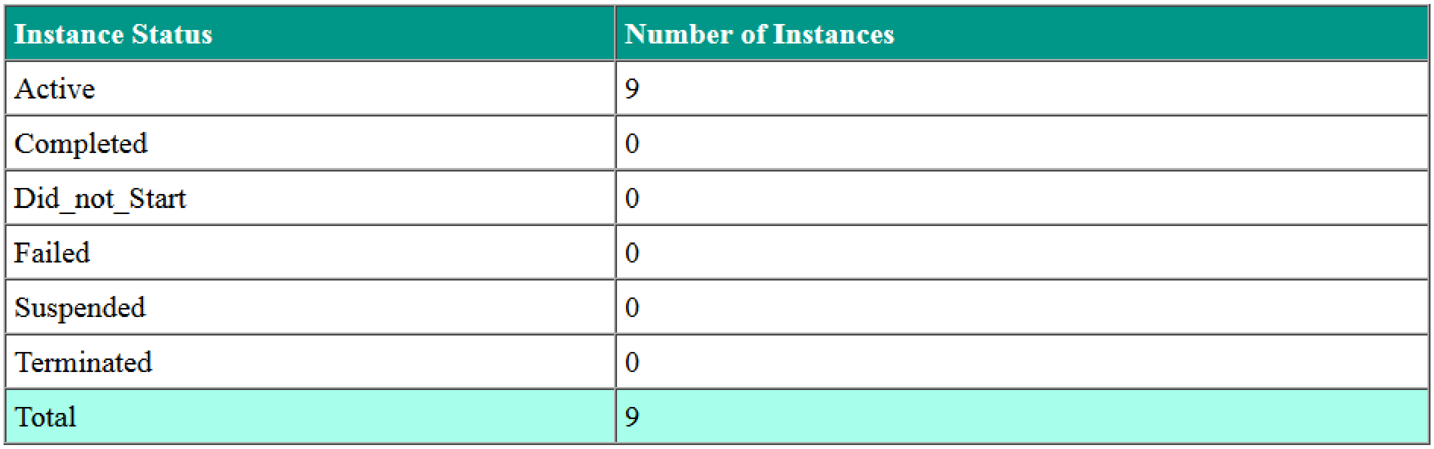
	Base Table LSW_BPM_INSTANCE	Tracking Table BPM_INSTANCE_INDEX	Instance Documents in Search Index
Instances	9	9	9

In a high traffic environment you may find that the numbers vary by a small amount. In this case a small variation does not indicate that there is an issue with the system as the process instance records may still be waiting to be processed by the indexer. Small discrepancies may fix themselves over a short period of time or can be fixed by using the Instance Reconciliation Report.

If there is a large discrepancy between any of the three values then there may be a larger issue with the search index.

Instance Breakdown from base table LSW_BPD_INSTANCE

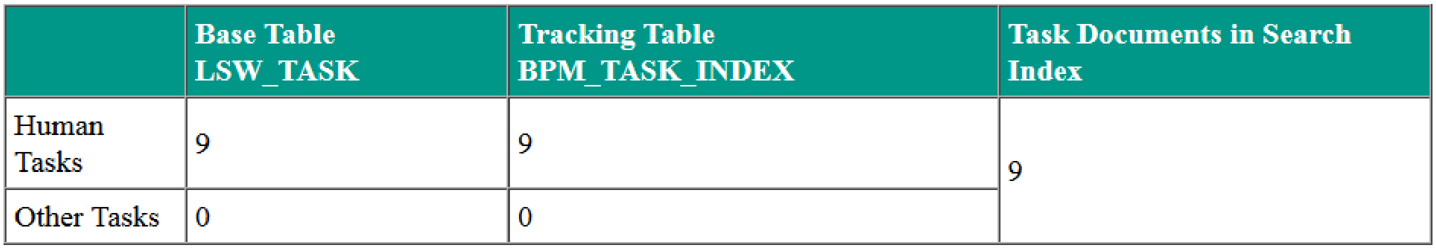
This table provides an overview of the process instances from the operational table, sorted by status.



Instance Status	Number of Instances
Active	9
Completed	0
Did_not_Start	0
Failed	0
Suspended	0
Terminated	0
Total	9

Task Summary Comparison

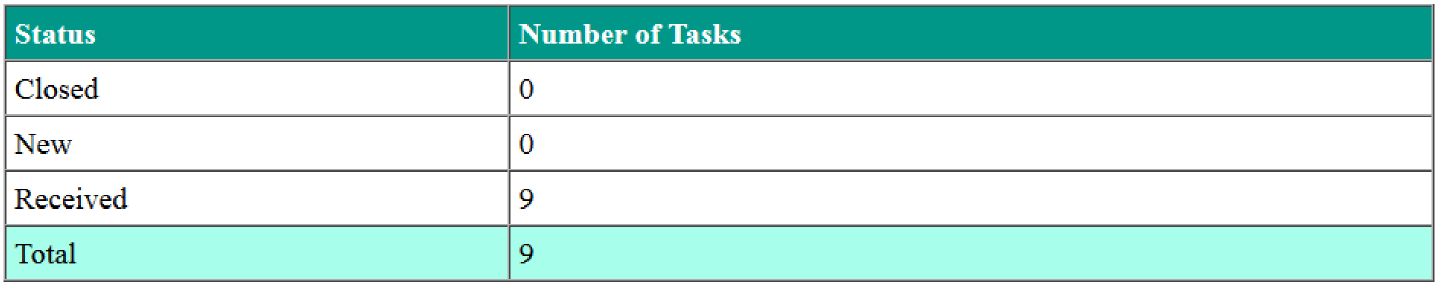
This section provides the number of tasks reported from the operational table (LSW_TASK), tracking table (BPM_TASK_INDEX), and the number of task documents in the search index. It provides both data for both human tasks and "other tasks" (system tasks). The number of Task Documents in the Search Index should match the sum total of human tasks and other tasks. Any major discrepancies found here may indicate a problem with the search indexing process. The Task Reconciliation Report can be used to try to correct any discrepancies reported here. Note that if system tasks are not indexed (<task-index-include-system-task> set to *false*) then the number reported for Other Tasks should be 0.



	Base Table LSW_TASK	Tracking Table BPM_TASK_INDEX	Task Documents in Search Index
Human Tasks	9	9	9
Other Tasks	0	0	

Task Breakdown from the base table LSW_TASK

This table shows the number of tasks reported in the LSW_TASK table categorized by status.



Status	Number of Tasks
Closed	0
New	0
Received	9
Total	9

Indexing Attributes Summary

The Attribute table provides the values of the configurable search index properties.

The Federated Index Enabled table indicates if the search index has been setup to enable indexing on Process Federation Server. This is only needed if the environment federated for use with Process Federation Server.

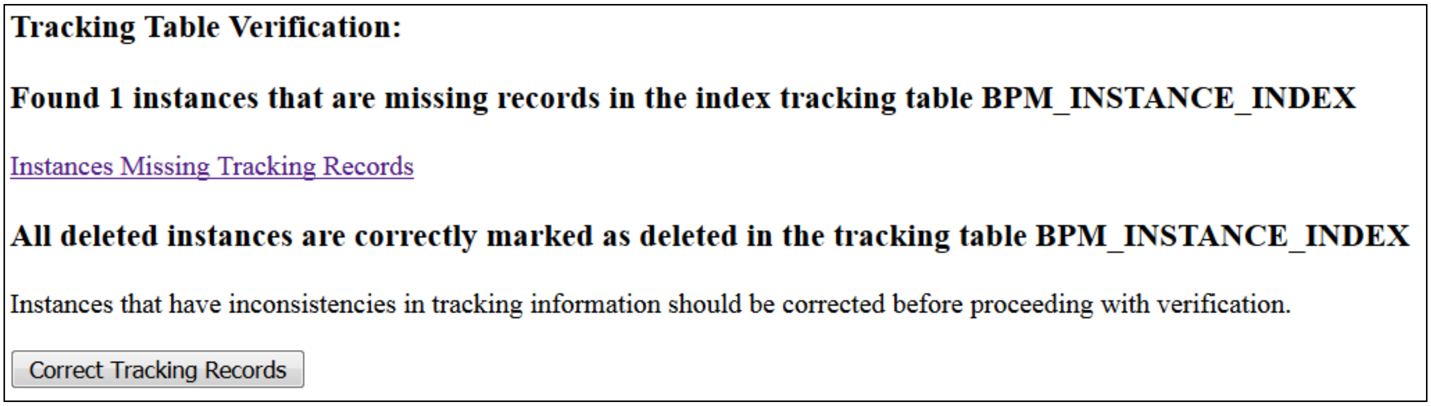
The last table notes if the listed PFS database tables exists. This is only needed if the environment federated for use with Process Federation Server.

Instance Reconciliation Report (Instance Verification)

This report can be used to run reconciliation to resolve discrepancies found for process instance data.

Tracking Table Verification

This section will show if any process instances are reported in the operational table LSW_BPD_INSTANCE but missing from the BPM_INSTANCE_INDEX index tracking table. This section also checks if all deleted instances are correctly marked as deleted in the index tracking table.



Tracking Table Verification:

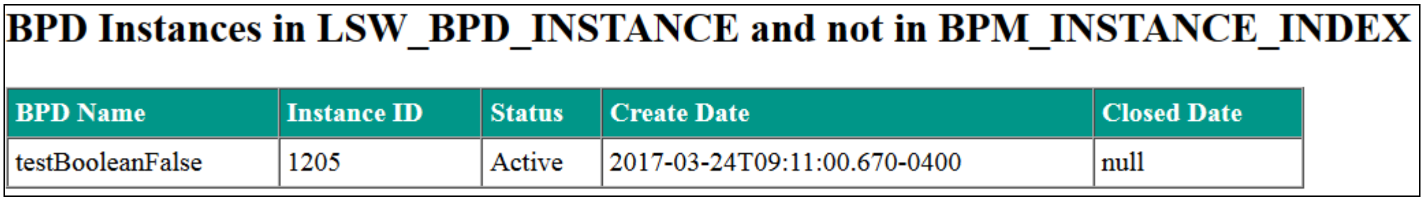
Found 1 instances that are missing records in the index tracking table BPM_INSTANCE_INDEX

Instances Missing Tracking Records

All deleted instances are correctly marked as deleted in the tracking table BPM_INSTANCE_INDEX

Instances that have inconsistencies in tracking information should be corrected before proceeding with verification.

If any discrepancies are found then a link is generated that will create an .html document that lists the missing process instance data.

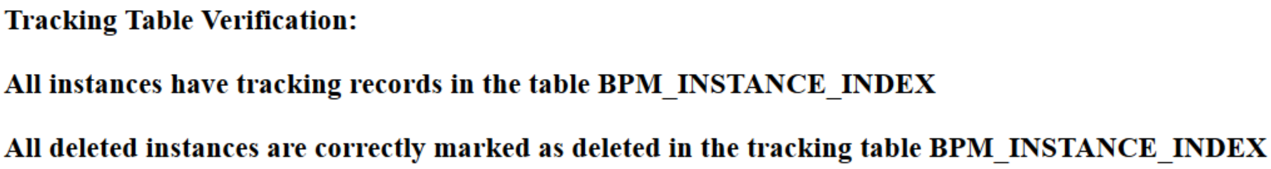


BPD Instances in LSW_BPD_INSTANCE and not in BPM_INSTANCE_INDEX

BPD Name	Instance ID	Status	Create Date	Closed Date
testBooleanFalse	1205	Active	2017-03-24T09:11:00.670-0400	null

Clicking the “Correct Tracking Records” button will automatically reconcile the reported process instances, updating the BPM_INSTANCE_INDEX table.

If the BPM_INSTANCE_INDEX table is synced with the operational LSW_BPD_INSTANCE table then you will see the following:



Tracking Table Verification:

All instances have tracking records in the table BPM_INSTANCE_INDEX

All deleted instances are correctly marked as deleted in the tracking table BPM_INSTANCE_INDEX

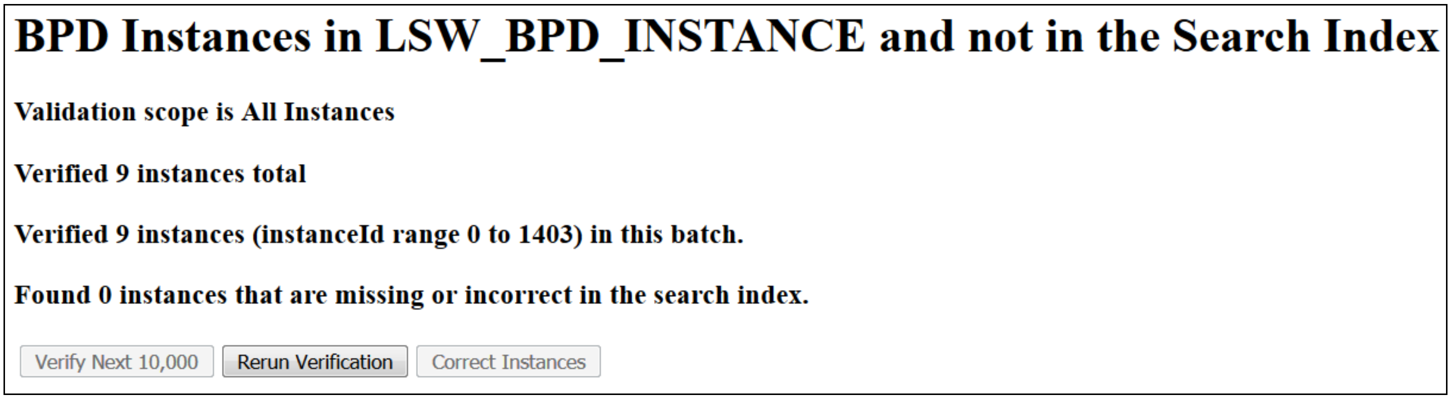
Search Index Verification

This section will allow reconciling any discrepancies between the process instance index tracking table and the search index. This section behaves differently than the Tracking Table Verification section. Discrepancies will not be automatically reported.

If this is the first time running this report the user should choose “Reconcile All Instances” and click the Continue button. Now the report will verify instances in the search index in batches of 10,000. If any instances are reported missing or incorrect in the search index then the user can click the Correct Instances button to update the search index.

The user can choose to cycle through all process instance documents in the search index by clicking Verify Next 10,000 button until all documents are verified.

Once this initial undertaking is complete then the user can choose the option to “Reconcile Instances that have been updated in the last:” and select an appropriate time frame. Users can run this reconciliation tool regularly as a best practice periodically to ensure their process instance data is synced.



BPD Instances in LSW_BPD_INSTANCE and not in the Search Index

Validation scope is All Instances

Verified 9 instances total

Verified 9 instances (instanceId range 0 to 1403) in this batch.

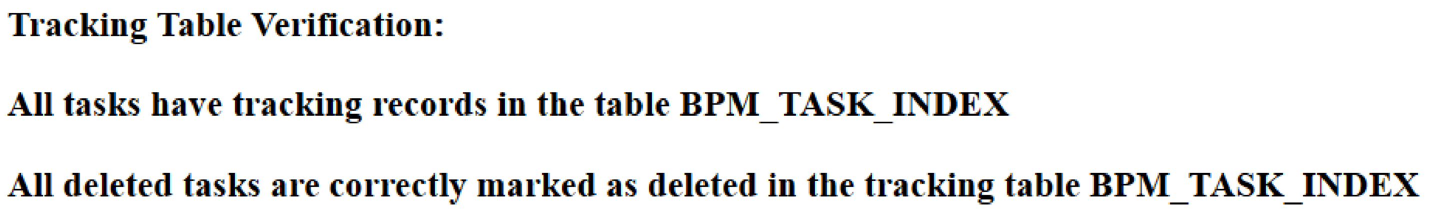
Found 0 instances that are missing or incorrect in the search index.

Task Reconciliation Report (Task Verification)

This report can be used to run reconciliation to resolve discrepancies found for task data.

Tracking Table Verification

This section will show if any task records from the LSW_TASK table (operational data) are missing from the corresponding index tracking table BPM_TASK_INDEX. If any records are found missing a link is provided by the report to generate an .html document which will list the task record. Similarly this section also reports any task records that have been deleted from the LSW_TASK table but are not marked as deleted in the BPM_TASK_INDEX table. A link to a list of such records is also provided if any such task records exist.



Tracking Table Verification:

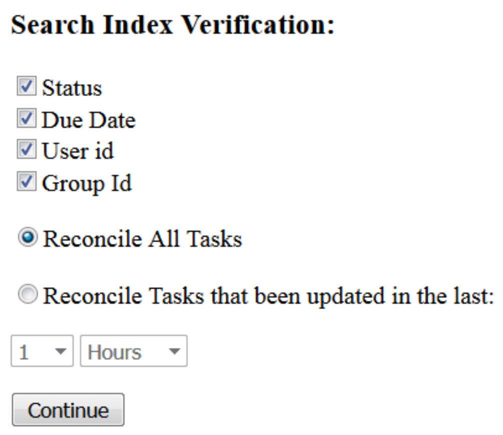
All tasks have tracking records in the table BPM_TASK_INDEX

All deleted tasks are correctly marked as deleted in the tracking table BPM_TASK_INDEX

If any task records are found missing or not appropriately marked as deleted a button will also be displayed in this section to “Correct Tracking Records”.

Search Index Verification

This section will allow reconciling of any task records that are not synced between the tracking table and the Lucene search index. This section behaves differently than the Tracking Table Verification section. Discrepancies will not be automatically reported.



Search Index Verification:

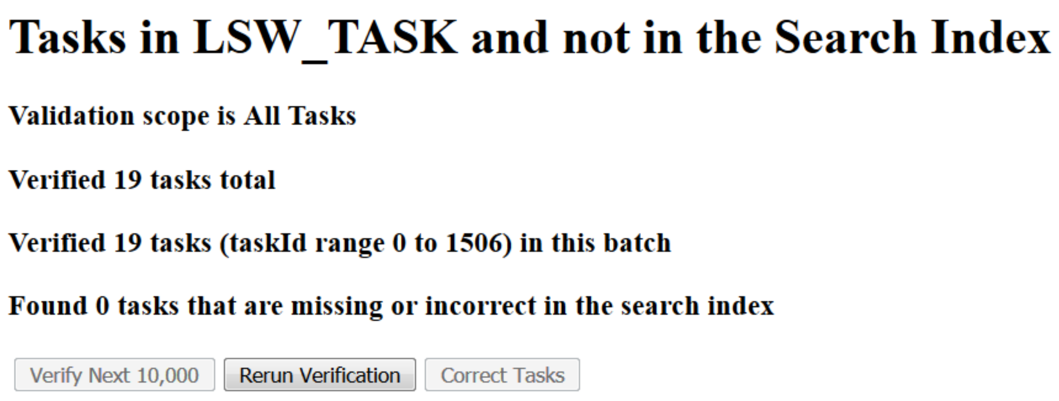
☑ Status
☑ Due Date
☑ User id
☑ Group Id

● Reconcile All Tasks

○ Reconcile Tasks that been updated in the last:

1 Hours

If this is the first time running this report the user should choose “Reconcile All Tasks” and click the Continue button. Now the report will verify tasks in the search index in batches of 10,000. If any tasks are reported missing or incorrect in the search index then the user can click the Correct Tasks button to update the search index.



Tasks in LSW_TASK and not in the Search Index

Validation scope is All Tasks

Verified 19 tasks total

Verified 19 tasks (taskId range 0 to 1506) in this batch

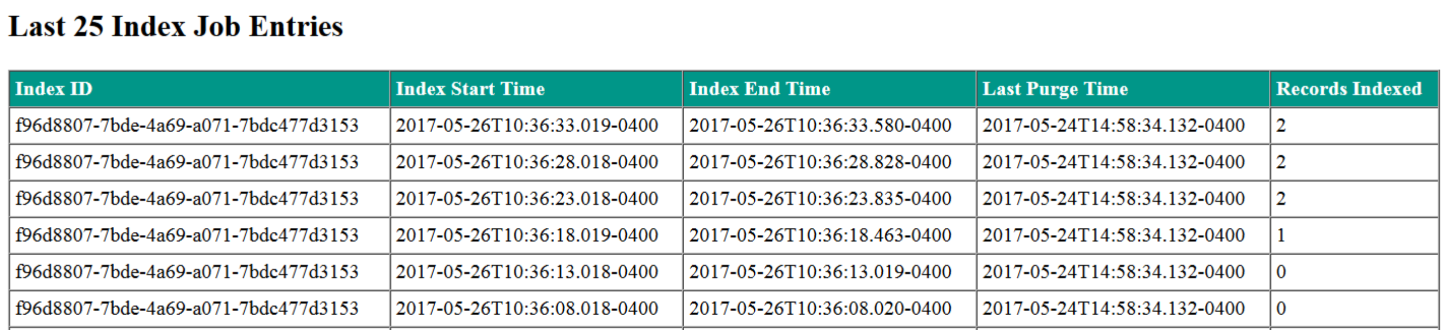
Found 0 tasks that are missing or incorrect in the search index

The user can choose to cycle through all task documents in the search index by clicking Verify Next 10,000 button until all documents are verified.

Once this initial undertaking is complete then the user can choose the option to “Reconcile Tasks that have been updated in the last:” and select an appropriate time frame. Users can run this reconciliation tool regularly as a best practice periodically to ensure their task data is synced.

List Latest Index Jobs

This report will list the latest 25 indexing jobs across all nodes. This report is useful to verify that the indexing jobs are running on each node which can be determined by checking the Index ID value as each node’s search index has a unique ID to identify the particular search index. The report is also useful in determining that the indexer on each node is running during the regularly expected interval as determined by the *process-index-update-interval* property which by default is 5 seconds.

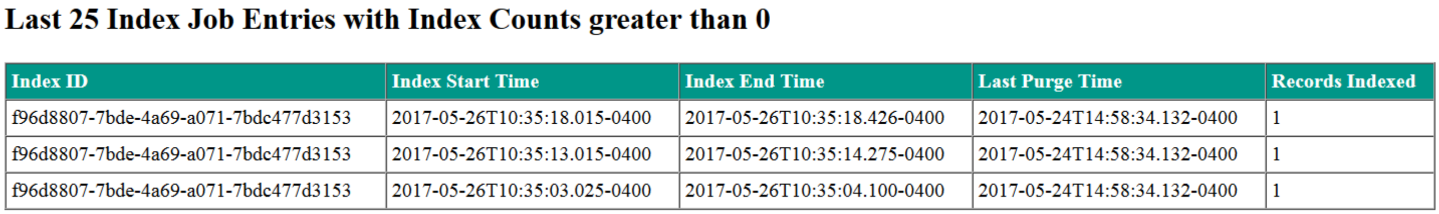


Last 25 Index Job Entries

Index ID	Index Start Time	Index End Time	Last Purge Time	Records Indexed
f96d8807-7bde-4a69-a071-7bdc477d3153	2017-05-26T10:36:33.019-0400	2017-05-26T10:36:33.580-0400	2017-05-24T14:58:34.132-0400	2
f96d8807-7bde-4a69-a071-7bdc477d3153	2017-05-26T10:36:28.018-0400	2017-05-26T10:36:28.828-0400	2017-05-24T14:58:34.132-0400	2
f96d8807-7bde-4a69-a071-7bdc477d3153	2017-05-26T10:36:23.018-0400	2017-05-26T10:36:23.835-0400	2017-05-24T14:58:34.132-0400	2
f96d8807-7bde-4a69-a071-7bdc477d3153	2017-05-26T10:36:18.019-0400	2017-05-26T10:36:18.463-0400	2017-05-24T14:58:34.132-0400	1
f96d8807-7bde-4a69-a071-7bdc477d3153	2017-05-26T10:36:13.018-0400	2017-05-26T10:36:13.019-0400	2017-05-24T14:58:34.132-0400	0
f96d8807-7bde-4a69-a071-7bdc477d3153	2017-05-26T10:36:08.018-0400	2017-05-26T10:36:08.020-0400	2017-05-24T14:58:34.132-0400	0

List Latest Index Jobs with Indexed Entries

This report will provide the latest 25 indexing jobs where tasks or process instances were indexed such that for the indexing job the number of records indexed is a non-zero value, displayed in descending chronological order.

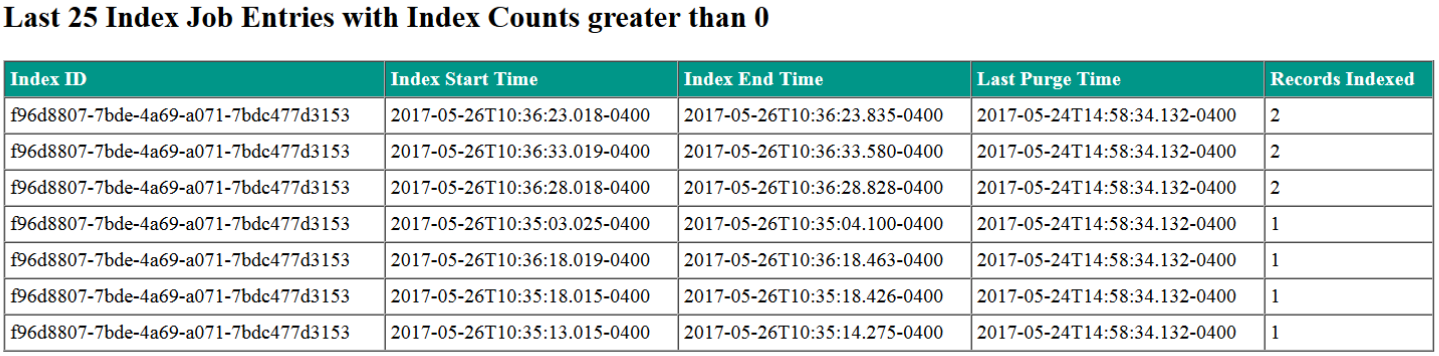


Last 25 Index Job Entries with Index Counts greater than 0

Index ID	Index Start Time	Index End Time	Last Purge Time	Records Indexed
f96d8807-7bde-4a69-a071-7bdc477d3153	2017-05-26T10:35:18.015-0400	2017-05-26T10:35:18.426-0400	2017-05-24T14:58:34.132-0400	1
f96d8807-7bde-4a69-a071-7bdc477d3153	2017-05-26T10:35:13.015-0400	2017-05-26T10:35:14.275-0400	2017-05-24T14:58:34.132-0400	1
f96d8807-7bde-4a69-a071-7bdc477d3153	2017-05-26T10:35:03.025-0400	2017-05-26T10:35:04.100-0400	2017-05-24T14:58:34.132-0400	1

List Latest Index Jobs with Indexed Entries Ordered By Record Count

This section shows the latest 25 indexing jobs where the number of records indexed are greater than zero and they are displayed in descending order of the number of records indexed.

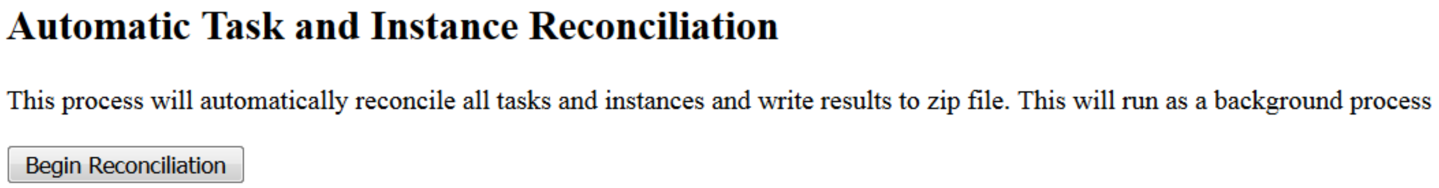


Last 25 Index Job Entries with Index Counts greater than 0

Index ID	Index Start Time	Index End Time	Last Purge Time	Records Indexed
f96d8807-7bde-4a69-a071-7bdc477d3153	2017-05-26T10:36:23.018-0400	2017-05-26T10:36:23.835-0400	2017-05-24T14:58:34.132-0400	2
f96d8807-7bde-4a69-a071-7bdc477d3153	2017-05-26T10:36:33.019-0400	2017-05-26T10:36:33.580-0400	2017-05-24T14:58:34.132-0400	2
f96d8807-7bde-4a69-a071-7bdc477d3153	2017-05-26T10:36:28.018-0400	2017-05-26T10:36:28.828-0400	2017-05-24T14:58:34.132-0400	2
f96d8807-7bde-4a69-a071-7bdc477d3153	2017-05-26T10:35:03.025-0400	2017-05-26T10:35:04.100-0400	2017-05-24T14:58:34.132-0400	1
f96d8807-7bde-4a69-a071-7bdc477d3153	2017-05-26T10:36:18.019-0400	2017-05-26T10:36:18.463-0400	2017-05-24T14:58:34.132-0400	1
f96d8807-7bde-4a69-a071-7bdc477d3153	2017-05-26T10:35:18.015-0400	2017-05-26T10:35:18.426-0400	2017-05-24T14:58:34.132-0400	1
f96d8807-7bde-4a69-a071-7bdc477d3153	2017-05-26T10:35:13.015-0400	2017-05-26T10:35:14.275-0400	2017-05-24T14:58:34.132-0400	1

Automatically Reconcile All Tasks Instances – Background Task

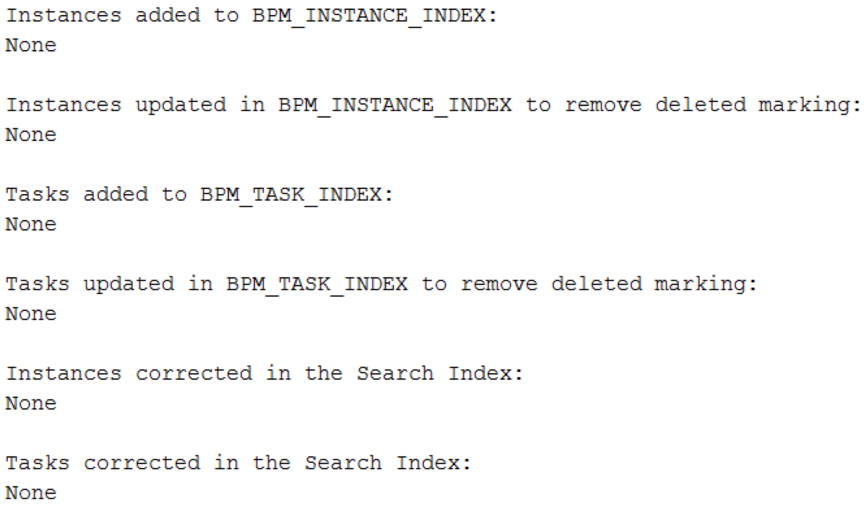
This feature of the report can be run to quickly reconcile all task and process instances in the tracking tables and the search indexes rather than performing the reconciliation activities separately in the other Reconciliation reports.



Automatic Task and Instance Reconciliation

This process will automatically reconcile all tasks and instances and write results to zip file. This will run as a background process

Once this task is executed a .zip file will be generated containing a .txt file providing a summary of how many records were reconciled.



```
Instances added to BPM_INSTANCE_INDEX:
None

Instances updated in BPM_INSTANCE_INDEX to remove deleted marking:
None

Tasks added to BPM_TASK_INDEX:
None

Tasks updated in BPM_TASK_INDEX to remove deleted marking:
None

Instances corrected in the Search Index:
None

Tasks corrected in the Search Index:
None
```

Troubleshooting the Process Portal Search Index

This section will help you determine when problems arise with the Process Portal Search Index and how to resolve them.

How to Determine an Issue Exist

It may not always be obvious when there is a problem with the search indexer. The following methods can be used to definitively determine if there is a major problem with the health of the Search Index.

Process Portal Observations

Any number of discrepancies may be reported by BPM end users which can indicate out of sync conditions within the search index. The search index stores many different fields of data for tasks and process instances including status, owner, team, exposed business data, task and instance IDs, etc. The Process Portal Search Index is used to generate search results, task lists, and other data throughout Process Portal. Users may report that a closed task is appearing as a result in the search against their WORK task list with an Open status. Users may report that there is are incorrect numbers reported for the Quick Stats views in the out of the box Process Portal dashboards. Users may be able to identify these discrepancies by comparing the data to another data point in Process Portal. Screenshots are a useful way to prove these conditions exists.

Index Diagnostic Reports

As discussed previously, the Index Diagnostic reports provide snapshots of the health of the Search Index. You can review these reports to determine if the search index tracking tables and the Lucene search index are in sync with the operational data by checking the Index Summary Report. You can also check that the search indexing process is running on every interval to ensure that there are not any issues with the indexing jobs. You may find that they are not running as expected indicating a problematic condition has occurred preventing the jobs from running efficiently and successfully as expected.

Server Logs

Frequently check the systemout.log files and the FFDC files for exceptions or messages related to the search index. Search indexing activity exceptions can be identified with the string “`com.ibm.bpm.search`” and “`com.ibm.bpm.server.search.eventpoints.listeners`”. Oftentimes exceptions will be written in the FFDC files providing more robust details of the error that has occurred. Other search terms you can use to verify warnings or exceptions are related to the search index are “`ProcessIndexU`”, “`ProcessIndexQ`”, “`ProcessIndexB`”, or “`SearchEventPo`”.

Database Deadlocks & Lock Escalations

Contact your DBA to determine if any deadlocks or locking escalations are preventing the indexing process from running successfully. Your DBA will be able to point to any locking issues with the search index tracking tables BPM_TASK_INDEX, BPM_INSTANCE_INDEX, or the BPM_TASK_INDEX_JOB tables.

Javacores & Heapdumps

If you are experiencing high CPU usage and/or OutOfMemoryError exceptions then you should analyze any javacore files or heapdumps that are generated. If the analysis reveals that "com/ibm/bpm/search/…/…/…" is the culprit then you can determine that the search index is involved in the issue. You may also find hung threads reported in the server logs indicating "com.ibm.bpm.search…" which reveals hung threads with the search indexer, most likely due to deadlocks in the database.

How to Resolve Common Issues

This section will describe common issues the IBM BPM Support team has encountered frequently and steps to resolve them. Be sure to have APAR JR55539 applied to your environment to avoid many of the issues discussed below.

Simple Out of Sync Scenarios

Out of sync conditions can be verified easily by BPM Administrators via the Index Diagnostic tool reports. The report will show the counts of tasks and process instance listed in the operational database tables versus the index tracking tables versus the Lucene search index itself to quickly determine if an out of sync condition exist. If an out of sync condition exists there are a number of methods to get back in sync.

Check the Search Index Configuration Properties for each Node

Check the search index configuration properties by accessing the Index Diagnostic Utility tool – Index Summary Report or by reviewing the <*search-index*> section in the TeamworksConfiguration.running.xml file.

Ensure indexing is enabled. By default system tasks are not indexed so if you expect Process Portal to return system tasks in search results against tasks then you must configure system tasks to be included in the indexing process.

If you have multiple nodes in your environment and need search indexing enabled to utilize quick search in Process Portal then you must have search indexing enabled on all nodes. If your configuration has indexing disabled on some nodes but enabled on others with users performing activities with tasks on all nodes then the tasks and processes that are updated, created, or closed will fall out of sync on the search indexes maintained on the other nodes. Activities on the nodes with indexing disabled will not trigger updated to the index tracking tables shared by all nodes and not be persisted to the other active search indexes. If you must disable indexing on one node then you must also bring the node offline so that users are not able to perform activities against tasks and process instances on that node.

Index Diagnostic Utility Tool Reconciliation Feature

The easiest way to re-sync the index tracking tables to the operational database and to sync the Lucene search index to the index tracking tables is to run the reconciliation features in the Index Diagnostic tool. The tool will report when discrepancies exists and identify the task and process instances records or documents that are out of sync. The reconciliation is executed with the click of the button inside the report.

Run the Search Index Rebuild Command

The *processIndexFullReIndex* command rebuilds the search index. This command must be run against each node in the environment. While this command is executing the search filters usually available in Process Portal will be temporarily disabled and unavailable to users. This command can be run while users are working in the environment.

To monitor the progress and completion of the tool check the systemout.log file for the following messages:

CWLLG0763I: The IBM BPM process search full re-index job was successfully requested. Process indexing will begin on the next scheduled process index update interval.

CWLLG0764I: The BPM process search full re-index job was successfully started.

CWLLG0765I: The IBM BPM process search full re-index job was successfully completed. 5 instances and 5 tasks were successfully indexed. If some process instances or tasks were created or updated during the full re-index job, the indexer still needs some time to fully synchronize with the IBM BPM database.

When the command has completed the CWLLG0765I message will indicate how many tasks and process instance documents were updated in the search index.

High CPU Usage & OutofMemoryError Exceptions due to TaskIndexVO

This issue can occur when a large volume of tasks and/or process instances are cleaned up from the environment. The search indexer tries to update many records of the search index tracking tables leading to this issue. Originally this issue was resolved with JR54004 but has since been superseded by JR55539.

As a workaround to this issue to relieve the high CPU usage you can rebuild both the search indexes and the index tracking tables. Steps to perform these actions are listed below in the section "*How to Rebuild the Search Index*".

Search Index Job Commits Frequently Failing

You may find that the search index is failing to commit updates to the Lucene search index indicated by a high volume of the following message generated in the systemout.log file:

```
ProcessIndexQ W com.ibm.bpm.search.process.spi.impl.ProcessIndexQueueDaemon updateTaskQueueItems() Task Index [1935094] took longer than 1 second to commit, retrying.
```

This is an indication that the throughput of the number of tasks and instances attempted to be updated in one indexing job is too high and needs to be regulated. The fix APAR JR55895 resolves this issue by introducing batching to the indexing process. By default each batch being updated at one time is 20 records. APAR JR55895 introduces a new configurable property, *<process-index-queue-update-size>*, which changes the number of records that can be updated in one batch.

If you apply JR55895 and continue to see this message occur frequently in the systemout.log file then you should perform the following steps to resolve the issue:

1. Tune the *<process-index-queue-update-size>* property. Decreasing the value will reduce the time it takes to run
2. Tune the database to ensure allocated resources are sufficient and run REORG & RESTATS on tracking tables

Database Deadlocks & Lock Escalations

If you do not maintain the health of the tracking tables by cleaning up records corresponding to delete tasks and process instances then you may find that deadlocks occur during the indexing process preventing the search index from syncing successfully and causing high CPU usage in the database server.

If you have a DB2 database you should apply APAR JR57349 which reduces the amount of locking executed on the BPM_TASK_INDEX table.

To resolve these issues you may need to manually rebuild the search index tracking tables BPM_TASK_INDEX, BPM_INSTANCE_INDEX, and the BPM_TASK_INDEX_JOB tables as described below in the section "*How to Rebuild the Search Index*". The rebuild of the search index tracking tables will only create records corresponding to records in the operational database tables, thus the tasks and process instances that have been deleted previously will not be included in the rebuild.

You may also open a PMR with IBM Support and request to receive a stored procedure which can be run to clean up the records no longer needed in the tracking tables. The stored procedure is a temporary measure until a wsadmin command to clean up the search index tracking tables can be delivered in a future release.

It may also be beneficial to configure a shared search index to be used between all the nodes which will alleviate the amount of locking required since each node's search index attempts to lock the table to scan for records that need to be updated.

NullPointerException Errors Reported in the Server Logs

The server logs may indicate NPE errors related to "*com.ibm.bpm.search…*" Ensure you have JR55539 applied. Post-JR55539 there are two additional APARs that resolve NPEs related to the search index. They are APAR JR56801 and APAR JR57297. If you apply these fixes and still experience NPEs related to the search index open a PMR with IBM BPM Support and upload the server logs as well as the Index Diagnostic Reports for Support.

How to Rebuild the Search Index

This section will provide detailed steps to rebuild the search index to resolve out of sync situations and other issues related to the search index. There are iterations described below where further steps may be required. For each iteration the conditions for which completing the steps are appropriate are provided.

In many cases performing the below steps may provide temporary relief until a fix can be applied to permanently resolve the root cause of the search indexing issue.

Running the Search Index Rebuild Command

This process involves running the *processIndexFullReIndex* command script from the /bin directory of your BPM node profile.

The conditions for when to run the rebuilding command are as follows:

- When end users report discrepancies in search results for tasks and process instances
- When the Index Diagnostic Report indicates the Lucene search index is out of sync with the tracking tables
- After applying a search index related APAR

The steps to run this command are provided in the Knowledge Center – Updating the Process Portal Index. Note that this command is only executed against one node. You should typically run this command for all nodes unless you have identified a particular node with an out of sync search index. This command

Monitor the systemout.log file for the following messages to see the progress of the command:

CWLLG0763I: The IBM BPM process search full re-index job was successfully requested. Process indexing will begin on the next scheduled process index update interval.

CWLLG0764I: The BPM process search full re-index job was successfully started.

CWLLG0765I: The IBM BPM process search full re-index job was successfully completed. 5 instances and 5 tasks were successfully indexed. If some process instances or tasks were created or updated during the full re-index job, the indexer still needs some time to fully synchronize with the IBM BPM database.

Manually Rebuilding the Lucene Search Index

Manually rebuilding the Lucene search index requires deleting the search index directory or re-naming it so that a new one is built. This process should only be performed during non-peak hours when the environment has very low activity being performed on it.

The conditions for when manually rebuilding the Lucene search index:

- Rebuilding the search index using the *processIndexFullReIndex* command did not sync the search index to the tracking tables
- To be performed in parallel when rebuilding the search index tracking tables

Steps to manually rebuild the Lucene search index on each node:

1. Stop the BPM environment
2. Locate the search index directory for each node

    a. *<BPM_install_root>/profiles/<profile_name>/searchIndex*
3. Remove or rename the */searchIndex* directory
4. Repeat steps 2 – 3 for all nodes on which you intend to rebuild the Lucene search index
5. Start BPM environment, bringing up one node at a time
6. To verify when a node's search index has completely rebuilt monitor the logs for the following:
    a. *CWLLG0765I: The IBM BPM process search full re-index job was successfully completed. 5 instances and 5 tasks were successfully indexed. If some process instances or tasks were created or updated during the full re-index job, the indexer still needs some time to fully synchronize with the IBM BPM database.*

Manually Rebuilding the Search Index Tracking Tables

Rebuilding the three search index tracking tables requires truncating/deleting the tables in parallel with rebuilding the Lucene search index. This step can be performed in a Production environment but should only be performed during non-peak hours so that no activity is being performed on the environment. It is safe to rebuild these tables without fear of losing operational data for tasks and process instances as the data stored in the index tracking tables are only references to the operational data and the tables will be rebuilt on start up to reflect the operational data.

It can be useful to perform this activity when any of the following conditions exist:

- High CPU usage and OutofMemoryError exceptions due to TaskIndexVO
- Deadlocks and lock escalations due to SQL activity against the BPM_TASK_INDEX or BPM_INSTANCE_INDEX
- You have a quick need to purge the BPM_TASK_INDEX and/or BPM_INSTANCE_INDEX of records that correspond to deleted task or process instance data (rather than using the stored procedure provided by IBM BPM Support)

Steps to rebuilding the search index tracking tables are as follows:

1. Stop the BPM environment
2. Truncate the following BPMN database tables:
    a. BPM_TASK_INDEX
    b. BPM_INSTANCE_INDEX
    c. BPM_TASK_INDEX_JOB

    Note: this step can be performed by running the SQL command `DELETE FROM <schema_name>.<table_name>;`
3. Perform the steps listed above to remove/delete the search index directories for each node
4. Start up the BPM environment, bringing up only one node. While starting the first node the three index tracking tables will be rebuilt based on the records in the LSW_TASK and LSW_BPD_INSTANCE tables
5. Start up each remaining node one at a time, monitoring when the search index is fully rebuilt before proceeding

Providing Mustgathers for IBM BPM Support

When opening a PMR with IBM BPM Support to address search index related issues please provide the following:

1. Problem Description
    a. What problems are observed in the BPM environment?
    b. What are the indications that the issue is related to the search index?
2. Business Impact Statement
    a. How are end users impacted?
    b. How are business requirement impacted?
    c. Are there any deadlines impacted – if yes, when is the deadline?
3. What type of environment is the issue occurring within? DEV / TEST / PROD?
4. If the BPM users observe problems with search results or other discrepancies in areas of Process Portal that indicate that an issue exist with the search index please upload screenshots of these discrepancies such as search conditions and the search results.
5. Provide the TeamworksConfiguration.running.xml and 100Custom files for each node
6. Use the Index Diagnostic tool to *Create Diagnostic Summary Report for Support* and upload the .zip file
7. Upload the entire /logs directory including FFDC files
8. Provide the results of the following SQL queries:

```
a. SELECT COUNT(1) FROM <schema_name>.BPM_TASK_INDEX WHERE DELETE_DATETIME IS
   NOT NULL;
b. SELECT COUNT(1) FROM <schema_name>.BPM_INSTANCE_INDEX WHERE DELETE_DATETIME
   IS NOT NULL;
```

Downloadable Resources

Download this article as a pdf file

Download the Index Diagnostic Tool

Related Topics and Resources

Knowledge Center: Administering the Process Portal index

Knowledge Center: Updating the Process Portal index

Knowledge Center: Configuring the Process Portal index

Knowledge Center: Default indexed fields

dwAnswer: How to investigate and solve a Portal index out of sync situation?

dwAnswer: Updating Schema of Search Index Database Tables for Oracle Database

IBM Redbooks – IBM Business Process Manager V8.5 Performance Tuning and Best Practices