## How to fix the damaged node indexes?

**Unindexed Solr Transactions:**

You can check the status of the Solr index to identify the nodes to a transaction that failed to index.

To generate a general report for Solr 4, including the last transaction indexed and the time, use:

`https://localhost:8443/solr4/admin/cores?action=REPORT&wt=xml`

To generate a general report for Solr 6, including the last transaction indexed and the time, use:

`http://localhost:8983/solr/admin/cores?action=REPORT&wt=xml`

The `REPORT` parameter compares the database with the index and generates an overall status report with the following details:

`Count of duplicate nodes in the index`: indicates the number of duplicate docs or folders in the index. ***The value of this parameter should be zero. If not, there is an issue with the index***.

`Count of duplicated transactions in the index`: indicates the number of transactions that appear more than once in the index. ***The value of this parameter should be zero. If not, there is an issue with the index.***

`Count of duplicated acl transactions in the index`: indicates the number of ACL transactions that appear more than once in the index. ***The value of this parameter should be zero. If not, there is an issue with the index.***

`Count of transactions in the index but not the database`: indicates the number of transactions in the index but not in the database. This count includes empty transactions that have been purged from the database. ***The value of this parameter should be zero. If not, there might be an issue with the index.***

`Count of acl transactions in the index but not the DB`: indicates the number of ACL transactions in the index but not in the database. ***The value of this parameter should be zero. If not, there is an issue with the index***. Note that empty ACL transactions are not purged from the database.

`Count of missing transactions from the Index`: indicates the number of transactions in the database but not in the index. ***The value of this index should be zero when the index is up-to-date.***

`Count of missing acl transactions from the Index`: indicates the number of ACL transactions in the database but not in the index. ***The value of this index should be zero when the index is up-to-date.***

`DB transaction count`: indicates the transaction count on the database

`DB acl transaction count`: indicates the ACL transaction count on the database

`Index transaction count`: indicates the number of transactions in the index.

`Index acl transaction count`: indicates the number of ACL transactions in the index.

`Index unique transaction count`: indicates the number of unique transactions in the index.

`Index unique acl transaction count`: indicates the number of unique ACL transactions in the index.

`Index leaf count`: indicates the number of docs and folders in the index.

`Last indexed transaction commit time`: indicates the time stamp for the last transaction added to the index. It also indicates that transactions after this time stamp have not yet been indexed.

`Last indexed transaction commit date`: indicates the time stamp as date for the last transaction added to the index. It also indicates that transactions after this date have not yet been indexed.

`Last TX id before holes`: indicates that transactions after this ID will be checked again to make sure they have not been missed. This is computed from the index at start up time. By default, it is set an hour after the last commit time found in the index. Solr tracking, by default, goes back an hour from the current time to check that no transactions have been missed.

`First duplicate`: indicates if there are duplicate transactions in the index. It returns the ID of the first duplicate transaction.

`First duplicate acl tx`: indicates if there are duplicate ACL transactions in the index. It returns the ID of the first duplicate ACL transaction.

`First transaction in the index but not the DB`: if the related count is > 0, it returns the ID of the first offender.

`First acl transaction in the index but not the DB`: if the related count is > 0, it returns the ID of the first offender.

`First transaction missing from the Index`: if the related count is > 0, it returns the ID of the first offender.

`First acl transaction missing from the Index`: if the related count is > 0, it returns the ID of the first offender.

`First duplicate leaf in the index`: if the related count is > 0, it returns the ID of the first offender.

To generate a summary report for Solr 4, use:

```
https://localhost:8443/solr4/admin/cores?action=SUMMARY&wt=xml
```

To generate a summary report for Solr 6, use:

```
http://localhost:8983/solr/admin/cores?action=SUMMARY&wt=xml
```

With multi-threaded tracking, you can specify additional tracking details and tracking statistics:

`detail=true`: provide statistics per tracking thread

`hist=true`: provides a histogram of the times taken for tracking operations for each tracking thread

`reset=true`: resets all tracking statistics

`values=true`: reports (by default) the last 50 values recorded for each tracking operation for each thread

The `SUMMARY` parameter provides the status of the tracking index and reports the progress of each tracking thread. It generates a report with the following details:

`Active`: indicates the tracker for the core active.

`Last Index Commit Time`: indicates the time stamp for the last transaction that was indexed.

`Last Index Commit Date`: indicates the time stamp as a date for the last transaction that was indexed. Changes made after this time are not yet in the index.

`Lag`: indicates the difference in seconds between the last transaction time stamp on the server and the time stamp for the last transaction that was indexed.

`Duration`: indicates the time lag as an XML duration.

`Approx transactions remaining`: indicates the approximate number of transactions to index in order to bring the index up-to-date. It is calculated as the last transaction ID on the server minus the last transaction ID indexed. It includes all the missing and empty transactions.

`Approx transaction indexing time remaining`: it is based on `Approx transactions remaining`, the average number of nodes per transaction and the average time to index a node (how long the index will take to be up-to-date). The estimate is in the most appropriate scale, for example, seconds, minutes, hours and days.

`Model sync times (ms)`: indicates summary statistics for model sync time. It supports additional information with `&detail=true`, `&hist=true` and `&value=true`.

`Acl index time (ms)`: indicates summary statistics for ACL index time. It supports additional information with `&detail=true`, `&hist=true` and `&value=true`.

`Node index time (ms)`: indicates summary statistics for node index time. It supports additional information with `&detail=true`, `&hist=true` and `&value=true`.

`Acl tx index time (ms)`: indicates the summary statistics for ACL transaction index time. It supports additional information with `&detail=true`, `&hist=true` and `&value=true`.

`Tx index time (ms)`: indicates summary statistics for transaction index time. It specifies the estimated time required to bring the index up-to-date.

`Docs/Tx`: indicates summary statistics for the number of documents per transaction. It supports additional information with `&detail=true`, `&hist=true` and `&value=true`.

`Doc Transformation time (ms)`: indicates summary statistics for document transformation time. It supports additional information with `&detail=true`, `&hist=true` and `&value=true`.

**SOLR Re-Index (encode the value of query):**

http://localhost:8080/solr4/admin/cores?action=**REINDEX**&query=TYPE:"cm:content" AND @cm\:created:[2016-06-01T00:00:00 TO 2017-02-12T00:00:00]

**Reindex a given folder:**

http://localhost:8983/solr/admin/cores?action=REINDEX&query=TYPE:"cm:folder" AND @cm\:name:"Folder_To_Reindex"

To repair an unindexed or failed transaction (as identified by the `REPORT` option), run the following report:

**http://localhost:8080/solr4/admin/cores?action=FIX**

The `FIX` parameter compares the database with the index and identifies any missing or duplicate transactions. It then updates the index by either adding or removing transactions.
Use the `PURGE` parameter to remove transactions, acl transactions, nodes and acls from the index. It can also be used for testing wrong transactions and then to fix them.

**http://localhost:8080/solr4/admin/cores?action=PURGE&txid=1&acltxid=2&nodeid=3&aclid=4**

Use the `REINDEX` parameter to reindex a transaction, acl transactions, nodes and acls.

```
http://localhost:8080/solr4/admin/cores?action=REINDEX&txid=1&acltxid=2&nodeid=3&ac
lid=4
```

Use the `INDEX` parameter to create entries in the index. It can also be used to create duplicate index entries for testing.

```
http://localhost:8080/solr4/admin/cores?action=INDEX&txid=1&acltxid=2&nodeid=3&acli
d=4
```

Use the `RETRY` parameter to retry indexing any node that failed to index and was skipped. In other words, it enables the users to attempt to fix documents that failed to index in the past and appear in the solr report (`http://localhost:8080/solr4/admin/cores?action=REPORT&wt=xml`) with the field **Index error count**.

```
http://localhost:8080/solr4/admin/cores?action=RETRY
```

Use the following setting to specify an option core for the report. If it is absent, a report is produced for each core. For example:

```
&core=alfresco
&core=archive
```

You can also fix index issues, check the index cache and backup individual indexes by using JMX. The status of the index can be checked using the JMX client on the **JMX MBeans > Alfresco > solrIndexes > <store>** tabs. The default view is the Solr core summary. The operations run the same consistency checks that are available by URL.

**Reindex by Node DBID:**
http://localhost:8983/solr/admin/cores?action=reindex&nodeid=608175

**Reindex by type:**
http://localhost:8983/solr/admin/cores?action=reindex&core=alfresco&query=demo:whitePaper

**Reindex by property value:**
http://localhost:8983/solr/admin/cores?action=reindex&core=alfresco&query=demo:author:John

**Reindex by folder path:**
http://localhost:8983/solr/admin/cores?action=reindex&core=alfresco&query=PATH:"/app:com
pany_home/st:sites/cm:testsite/cm:documentLibrary/cm:Draft_x0020_Reports/cm:Whitepapers//
*"

**The following SOLR Query will return the number of SOLR Documents with a different value in those *_CONTENT_VERSION_ID fields:**

http://localhost:8983/solr/alfresco/select?q=*&fq={!frange l=1 u=1
v=$equals}&equals=if(not(eq(LATEST_APPLIED_CONTENT_VERSION_ID,LAST_INCOM
ING_CONTENT_VERSION_ID)),1,0)&indent=on&wt=json

**If you want to know the status of the nodes pending to be indexed or updated, you can add those \*_CONTENT_VERSION_ID fields to the "fl" parameter:**

http://localhost:8983/solr/alfresco/select?q=*&fl=[cached]LATEST_APPLIED_CONTENT_VERSION_ID,LAST_INCOMING_CONTENT_VERSION_ID&fq={!frange l=1 u=1 v=$equals}&equals=if(not(eq(LATEST_APPLIED_CONTENT_VERSION_ID,LAST_INCOMING_CONTENT_VERSION_ID)),1,0)&indent=on&wt=json

**How enable/disable indexing:**

**http://localhost:8983/solr/admin/cores?action=disable-indexing**

**http://localhost:8983/solr/admin/cores?action=enable-indexing**

More details: https://docs.alfresco.com/search-services/latest/admin/restapi/#disable-indexing
**how-many-content-remains-to-be-indexed**

**Search Services Admin REST API can be used to get this information.**
**http://localhost:8983/solr/admin/cores?action=summary&core=alfresco**

**Search Service 2.x**
```
<lst name="FTS">
  <long name="Node count whose content is in sync">170</long>
  <long name="Node count whose content needs to be updated">7</long>
</lst>
```

**Search Service 1.x:**

```
<lst name="FTS">
 <long name="Node count with FTSStatus Clean">125</long>
 <long name="Node count with FTSStatus Dirty">3</long>
 <long name="Node count with FTSStatus New">152</long>
</lst>
```
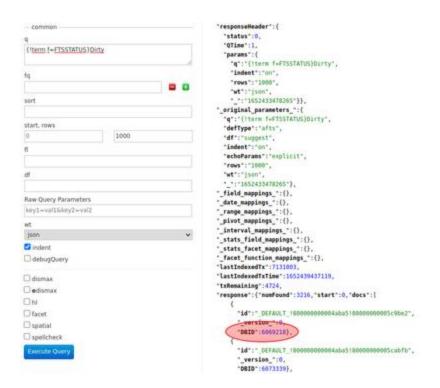
**Additional information for ASS 1.4:**

**In case you would like to know which documents are in the FTSStatus "Dirty" state**

**http://localhost:8983/solr/#/alfresco/query?q=%7B!term%20f%3DFTSSTATUS%7DDirty**

You can use the **DBID**

and fire up a query in the Alfresco Node Browser (Admin Console older Share Administration Tools) with

@sys:node-dbid:<DBID> to get more information about the document in the "dirty" state.