

# Atomic Updates



This page exists for the Solr Community to share Tips, Tricks, and Advice about [Atomic Updates](#).

Reference material previously located on this page has been migrated to the [Official Solr Reference Guide](#). If you need help, please consult the Ref Guide for the version of Solr you are using for the specific details about using [this feature](#).

If you'd like to share information about how you use this feature, please [add it to this page](#).

*/\* cwikimigrated \*/*

Solr4.0

## Atomic Updates

Atomic Updates is a feature added in Solr 4.0 that allows you to update on a field level rather than on a document level (In previous versions). This means that you can update individual fields without having to send the entire document to Solr with the un-updated fields values. Internally Solr re-adds the document to the index with the updated fields.

- [Atomic Updates](#)
- [Caveats and Limitations](#)
  - [Update Log](#)
  - [DistributedUpdateRequestProcessorFactory](#)
    - [Using Update Processors that Modify Documents](#)
    - [Disabling DistributedUpdateRequestProcessorFactory](#)
- [References](#)

## Caveats and Limitations

### Update Log

An `<updateLog/>` must be configured in your `solrconfig.xml` in order for atomic document updates to be used. This is necessary to ensure that the update instructions are applied to the most recently indexed version of the document – even if that version has not yet been committed (just like [RealTimeGet](#)).

### DistributedUpdateRequestProcessorFactory

To improve performance in typical installations, atomic document updates are processed during execution of the `DistributedUpdateRequestProcessor`. By default all [update chains](#) have the `DistributedUpdateRequestProcessorFactory` inserted into them just prior to the `RunUpdateProcessorFactory` and most users won't be affected by this implementation detail, but there are two situations where it might affect you...

### Using Update Processors that Modify Documents

If you use [UpdateRequestProcessors](#) to modify document values when indexing, these should be placed *after* the `DistributedUpdateRequestProcessorFactory` in your update chain to ensure they see the fully populated `SolrInputDocument` with all field values when an atomic document update request is received.

### Disabling [DistributedUpdateRequestProcessorFactory](#)

If you have disabled `DistributedUpdateRequestProcessorFactory` by adding your own `DistributingUpdateProcessorFactory` (or `NoOpDistributingUpdateProcessorFactory`) to your update chain, then atomic document updates are not supported

## References

- <https://issues.apache.org/jira/browse/SOLR-139>
- <http://yonik.com/solr/atomic-updates/>