

# PreAnalyzedUpdateProcessorFactory

An update processor that parses configured fields of any document being added using [PreAnalyzedField](#) with the configured format parser.

Fields are specified using the same patterns as in [FieldMutatingUpdateProcessorFactory](#). They are then checked whether they follow a pre-analyzed format defined by parser. Valid fields are then parsed. The original [SchemaField](#) is used for the initial creation of Lucene Field, which is then modified to add the results from parsing (token stream value and/or string value) and then it will be directly added to the final Lucene Document to be indexed.

Fields that are declared in the patterns list but are not present in the current schema will be removed from the input document.

## Implementation details

This update processor uses [PreAnalyzedParser](#) to parse the original field content (interpreted as a string value), and thus obtain the stored part and the token stream part. Then it creates the "template" Field-s using the original [SchemaField.createFields\(Object, float\)](#) as declared in the current schema. Finally it sets the pre-analyzed parts if available (string value and the token stream value) on the first field of these "template" fields. If the declared field type does not support stored or indexed parts then such parts are silently discarded. Finally the updated "template" Field-s are added to the resulting [SolrInputField](#), and the original value of that field is removed.

## Example configuration

In the example configuration below there are two update chains, one that uses the `simple` parser (`SimplePreAnalyzedParser`) and one that uses the `json` parser (`JsonPreAnalyzedParser`). Field "nonexistent" will be removed from input documents if not present in the schema. Other fields will be analyzed and if valid they will be converted to Lucene Field-s or if they are not in a valid format that can be parsed with the selected parser they will be passed as-is. Assuming that `ssto` field is stored but not indexed, and `sind` field is indexed but not stored: if `ssto` input value contains the indexed part then this part will be discarded and only the stored value part will be retained. Similarly, if `sind` input value contains the stored part then it will be discarded and only the token stream part will be retained.

```
<updateRequestProcessorChain name="pre-analyzed-simple">
  <processor class="solr.PreAnalyzedUpdateProcessorFactory">
    <str name="fieldName">title</str>
    <str name="fieldName">nonexistent</str>
    <str name="fieldName">ssto</str>
    <str name="fieldName">sind</str>
    <str name="parser">simple</str>
  </processor>
  <processor class="solr.RunUpdateProcessorFactory" />
</updateRequestProcessorChain>

<updateRequestProcessorChain name="pre-analyzed-json">
  <processor class="solr.PreAnalyzedUpdateProcessorFactory">
    <str name="fieldName">title</str>
    <str name="fieldName">nonexistent</str>
    <str name="fieldName">ssto</str>
    <str name="fieldName">sind</str>
    <str name="parser">json</str>
  </processor>
  <processor class="solr.RunUpdateProcessorFactory" />
</updateRequestProcessorChain>
```