Field Type Definitions and Properties

A field type defines the analysis that will occur on a field when documents are indexed or queries are sent to the index.

A field type definition can include four types of information:

- The name of the field type (mandatory).
- An implementation class name (mandatory).
- If the field type is `TextField`, a description of the field analysis for the field type.
- Field type properties - depending on the implementation class, some properties may be mandatory.

Field Type Definitions in `schema.xml`

Field types are defined in `schema.xml`. Each field type is defined between `fieldType` elements. They can optionally be grouped within a `types` element. Here is an example of a field type definition for a type called `text_general`:

```
<fieldType name="text_general" class="solr.TextField" positionIncrementGap="100">
  <analyzer type="index">
    <tokenizer class="solr.StandardTokenizerFactory"/>
    <filter class="solr.StopFilterFactory" ignoreCase="true" words="stopwords.txt" />
    <!-- in this example, we will only use synonyms at query time
    <filter class="solr.SynonymFilterFactory" synonyms="index_synonyms.txt" ignoreCase="true"
expand="false"/>
    -->
    <filter class="solr.LowerCaseFilterFactory"/>
  </analyzer>
  <analyzer type="query">
    <tokenizer class="solr.StandardTokenizerFactory"/>
    <filter class="solr.StopFilterFactory" ignoreCase="true" words="stopwords.txt" />
    <filter class="solr.SynonymFilterFactory" synonyms="synonyms.txt" ignoreCase="true"
expand="true"/>
    <filter class="solr.LowerCaseFilterFactory"/>
  </analyzer>
</fieldType>
```

① The first line in the example above contains the field type name, `text_general`, and the name of the implementing class, `solr.TextField`.

② The rest of the definition is about field analysis, described in Understanding Analyzers, Tokenizers, and Filters.

The implementing class is responsible for making sure the field is handled correctly. In the class names in `schema.xml`, the string `solr` is shorthand for `org.apache.solr.schema` or `org.apache.solr.analysis`. Therefore, `solr.TextField` is really `org.apache.solr.schema.TextField`..

Field Type Properties

The field type `class` determines most of the behavior of a field type, but optional properties can also be defined. For example, the following definition of a date field type defines two properties, `sortMissingLast` and `omitNorms`.

```
<fieldType name="date" class="solr.TrieDateField"
           sortMissingLast="true" omitNorms="true"/>
```

The properties that can be specified for a given field type fall into three major categories:

- Properties specific to the field type's class.
- General Properties Solr supports for any field type.
- Field Default Properties that can be specified on the field type that will be inherited by fields that use this type instead of the default behavior.

General Properties

Property	Description	Values
name	The name of the fieldType. This value gets used in field definitions, in the "type" attribute. It is strongly recommended that names consist of alphanumeric or underscore characters only and not start with a digit. This is not currently strictly enforced.	
class	The class name that gets used to store and index the data for this type. Note that you may prefix included class names with "solr." and Solr will automatically figure out which packages to search for the class - so `solr.TextField` will work. If you are using a third-party class, you will probably need to have a fully qualified class name. The fully qualified equivalent for `solr.TextField` is `org.apache.solr.schema.TextField`.	
positionIncrementGap	For multivalued fields, specifies a distance between multiple values, which prevents spurious phrase matches	integer
autoGeneratePhraseQueries	For text fields. If true, Solr automatically generates phrase queries for adjacent terms. If false, terms must be enclosed in double-quotes to be treated as phrases.	true or false

Property	Description	Values
enableGraphQueries	For text fields, applicable when querying with `sow=false`. Use `true` (the default) for field types with query analyzers including graph-aware filters, e.g. Synonym Graph Filter and Word Delimiter Graph Filter. Use `false` for field types with query analyzers including filters that can match docs when some tokens are missing, e.g., Shingle Filter.	true or false
docValuesFormat	Defines a custom `DocValuesFormat` to use for fields of this type. This requires that a schema-aware codec, such as the `SchemaCodecFactory` has been configured in solrconfig.xml.	n/a
postingsFormat	Defines a custom `PostingsFormat` to use for fields of this type. This requires that a schema-aware codec, such as the `SchemaCodecFactory` has been configured in solrconfig.xml.	n/a

NOTE Lucene index back-compatibility is only supported for the default codec. If you choose to customize the `postingsFormat` or `docValuesFormat` in your schema.xml, upgrading to a future version of Solr may require you to either switch back to the default codec and optimize your index to rewrite it into the default codec before upgrading, or re-build your entire index from scratch after upgrading.

Field Default Properties

These are properties that can be specified either on the field types, or on individual fields to override the values provided by the field types.

The default values for each property depend on the underlying `FieldType` class, which in turn may depend on the `version` attribute of the `<schema/>`. The table below includes the default value for most `FieldType` implementations provided by Solr, assuming a `schema.xml` that declares `version="1.6"`.

Property	Description	Values	Implicit Default
indexed	If true, the value of the field can be used in queries to retrieve matching documents.	true or false	true
stored	If true, the actual value of the field can be retrieved by queries.	true or false	true
docValues	If true, the value of the field will be put in a column-oriented DocValues structure.	true or false	false

Property	Description	Values	Implicit Default
sortMissingFirst sortMissingLast	Control the placement of documents when a sort field is not present.	true or false	false
multiValued	If true, indicates that a single document might contain multiple values for this field type.	true or false	false
omitNorms	If true, omits the norms associated with this field (this disables length normalization for the field, and saves some memory). **Defaults to true for all primitive (non-analyzed) field types, such as int, float, data, bool, and string.** Only full-text fields or fields need norms.	true or false	*
omitTermFreqAndPositions	If true, omits term frequency, positions, and payloads from postings for this field. This can be a performance boost for fields that don't require that information. It also reduces the storage space required for the index. Queries that rely on position that are issued on a field with this option will silently fail to find documents. **This property defaults to true for all field types that are not text fields.**	true or false	*
omitPositions	Similar to `omitTermFreqAndPositions` but preserves term frequency information.	true or false	*
termVectors termPositions termOffsets termPayloads	These options instruct Solr to maintain full term vectors for each document, optionally including position, offset and payload information for each term occurrence in those vectors. These can be used to accelerate highlighting and other ancillary functionality, but impose a substantial cost in terms of index size. They are not necessary for typical uses of Solr.	true or false	false
required	Instructs Solr to reject any attempts to add a document which does not have a value for this field. This property defaults to false.	true or false	false
useDocValuesAsStored	If the field has docValues enabled, setting this to true would allow the field to be returned as if it were a stored field (even if it has `stored=false`) when matching "*" in an fl parameter.	true or false	true

Property	Description	Values	Implicit Default
large	Large fields are always lazy loaded and will only take up space in the document cache if the actual value is < 512KB. This option requires `stored="true"` and `multiValued="false"`. It's intended for fields that might have very large values so that they don't get cached in memory.	true or false	false

Field Type Similarity

A field type may optionally specify a `<similarity/>` that will be used when scoring documents that refer to fields with this type, as long as the "global" similarity for the collection allows it.

By default, any field type which does not define a similarity, uses `BM25Similarity`. For more details, and examples of configuring both global & per-type Similarities, please see Other Schema Elements.