

Multiple Indexes

Running Multiple Indexes

There are various strategies to take when you want to manage multiple "indexes" of different types of documents in a Single Servlet Container

- [Running Multiple Indexes](#)
 - [MultiCore](#)
 - [Flattening Data Into a Single Index](#)
 - [Multiple Solr Ports](#)
 - [Multiple Solr Webapps](#)

MultiCore

⚠ [Solr1.3](#) added support for multiple "Solr Cores" in a single deployment of Solr – each Solr Core has its own index. For more information please see [Core Admin](#).

In this situation, the individual indexes are still fairly isolated, but you can manage them as a single application, create new indexes on the fly by spinning up new SolrCores, and even make one SolrCore replace another SolrCore without ever restarting your Servlet Container.

⚠ There is an open issue regarding opening multiple cores, see [LotsOfCores](#) for more information. Be aware that each core opened will take at least one thread and each thread will open at least one file handler.

Flattening Data Into a Single Index

In many cases where it feels like multiple indexes are needed, the same effects can be accomplished in a single index with an additional field to select what type you are searching for. Consider the case where you want to index books and movies.

For a book you may need:

- title
- author
- release (date)

For a movie you would need:

- title
- director
- release (date)

You can design a single [SchemaXml](#) to handle both object types.

```
<!-- this will be "book" or "movie" -->
<field name="type" ... />

<field name="title" ... />
<field name="author" ... />
<field name="director" ... />
<field name="release " ... />
```

To search for just books or movies add "+type:book" to your query.

Using the **fq** (filter query) parameter gives you a performance improvement:

```
&fq=type:book&q=...
```

This type of approach can be particularly well suited to situations where you need to "blend" results from conceptually distinct sets of documents.

Multiple Solr Ports

You can have separate instances of Solr, running in separate ports in isolation. This can be advantageous if you want to manage resources for each Solr instance completely independently.

Multiple Solr Webapps

You can put each index as an independent instance of the Solr webapp, configuring them to have separate Solr Home directories using JNDI. This can be useful if you want the indexes to "pool" resources in one Servlet Container (ie: ram) but be otherwise isolated.

The following documents have configuration tips and examples for various Servlet Containers...

- [Tomcat](#)
- [Jetty](#)
- [Resin](#)